# 1 PRODUCT OVERVIEW

## INTRODUCTION

This manual describes SAMSUNG's S3C2440X 16/32-bit RISC microprocessor. SAMSUNG's S3C24440X is designed to provide hand-held devices and general applications with low-power, and high-performance micro-controller solution in small die size. To reduce total system cost, the S3C2440X includes the following components separate 16KB Instruction and 16KB Data Cache, MMU to handle virtual memory management, LCD Controller (STN & TFT), NAND Flash Boot Loader, System Manager (chip select logic and SDRAM Controller), 3-ch UART, 4-ch DMA, 4-ch Timers with PWM, I/O Ports, RTC, 8-ch 10-bit ADC and Touch Screen Interface, Camera interface, IIC-BUS Interface, IIS-BUS Interface, USB Host, USB Device, SD Host & Multi-Media Card Interface, 2-ch SPI and PLL for clock generation.

The S3C2440X has been developed using an ARM920T core, 0.13um CMOS standard cells and a memory complier. Its low-power, simple, elegant and fully static design is particularly suitable for cost- and power-sensitive applications. It adopts a new bus architecture known as Advanced Micro controller Bus Architecture (AMBA).

The S3C2440X offers outstanding features with its CPU core, a 16/32-bit ARM920T RISC processor designed by Advanced RISC Machines, Ltd. The ARM920T implements MMU, AMBA BUS, and Harvard cache architecture with separate 16KB instruction and 16KB data caches, each with an 8-word line length.

By providing a complete set of common system peripherals, the S3C2440X minimizes overall system costs and eliminates the need to configure additional components. The integrated on-chip functions that are described in this document include:

- 1.2V internal, 1.8V/2.5V/3.3V memory, 3.3V external I/O microprocessor with 16KB I-Cache/16KB D-Cache/MMU
- External memory controller (SDRAM Control and Chip Select logic)
- LCD controller (up to 4K color STN and 256K color TFT) with 1-ch LCD-dedicated DMA
- 4-ch DMAs with external request pins
- 3-ch UART (IrDA1.0, 64-Byte Tx FIFO, and 64-Byte Rx FIFO) / 2-ch SPI
- 1-ch multi-master IIC-BUS/1-ch IIS-BUS controller
- SD Host interface version 1.0 & Multi-Media Card Protocol version 2.11 compatible
- 2-port USB Host /1- port USB Device (ver 1.1)
- 4-ch PWM timers & 1-ch internal timer
- Watch Dog Timer
- 130-bit general purpose I/O ports / 24-ch external interrupt source
- Power control: Normal, Slow, Idle and Sleep mode
- 8-ch 10-bit ADC and Touch screen interface
- RTC with calendar function
- On-chip clock generator with PLL

# FEATURES

### Architecture

- Integrated system for hand-held devices and general embedded applications.
- 16/32-Bit RISC architecture and powerful instruction set with ARM920T CPU core.
- Enhanced ARM architecture MMU to support WinCE, EPOC 32 and Linux.
- Instruction cache, data cache, write buffer and Physical address TAG RAM to reduce the effect of main memory bandwidth and latency on performance.
- ARM920T CPU core supports the ARM debug architecture.
- Internal Advanced Microcontroller Bus Architecture (AMBA) (AMBA2.0, AHB/APB).

### System Manager

- Little/Big Endian support.
- Address space: 128M bytes for each bank (total 1G bytes).
- Supports programmable 8/16/32-bit data bus width for each bank.
- Fixed bank start address from bank 0 to bank 6.
- Programmable bank start address and bank size for bank 7.
- Eight memory banks:
  - Six memory banks for ROM, SRAM, and others.
  - Two memory banks for ROM/SRAM/ Synchronous DRAM.
- Complete Programmable access cycles for all memory banks.
- Supports external wait signals to expend the bus cycle.
- Supports self-refresh mode in SDRAM for power-down.
- Supports various types of ROM for booting (NOR/NAND Flash, EEPROM, and others).

### NAND Flash Boot Loader

- Supports booting from NAND flash memory.
- 4KB internal buffer for booting.
- Supports storage memory for NAND flash memory after booting.
- Supports Advanced NAND flash

### Cache Memory

- 64-way set-associative cache with I-Cache (16KB) and D-Cache (16KB).
- 8words length per line with one valid bit and two dirty bits per line.
- Pseudo random or round robin replacement algorithm.
- Write-through or write-back cache operation to update the main memory.
- The write buffer can hold 16 words of data and four addresses.

### Clock & Power Manager

- On-chip MPLL and UPLL:
  UPLL generates the clock to operate USB Host/Device.
  MPLL generates the clock to operate MCU at maximum 400Mhz @ 1.2V.
- Clock can be fed selectively to each function block by software.
- **Power mode**: Normal, Slow, Idle, and Sleep mode
  **Normal mode**: Normal operating mode
  **Slow mode**: Low frequency clock without PLL
  **Idle mode**: The clock for only CPU is stopped.
  **Sleep mode**: The Core power including all peripherals is shut down.
- Woken up by EINT[15:0] or RTC alarm interrupt from Sleep mode

## FEATURES (Continued)

### Interrupt Controller

- 59 Interrupt sources
  (One Watch dog timer, 5 timers, 9 UARTs, 24 external interrupts, 4 DMA, 2 RTC, 2 ADC, 1 IIC, 2 SPI, 1 SDI, 2 USB, 1 LCD, 1 Battery Fault, 1 NAND and 2 Camera)
- Level/Edge mode on external interrupt source
- Programmable polarity of edge and level
- Supports Fast Interrupt request (FIQ) for very urgent interrupt request

### Timer with Pulse Width Modulation (PWM)

- 4-ch 16-bit Timer with PWM / 1-ch 16-bit internal timer with DMA-based or interrupt-based operation
- Programmable duty cycle, frequency, and polarity
- Dead-zone generation
- Supports external clock sources

### RTC (Real Time Clock)

- Full clock feature: msec, second, minute, hour, date, day, month, and year
- 32.768 KHz operation
- Alarm interrupt
- Time tick interrupt

### General Purpose Input/Output Ports

- 24 external interrupt ports
- Multiplexed input/output ports

### UART

- 3-channel UART with DMA-based or interrupt-based operation
- Supports 5-bit, 6-bit, 7-bit, or 8-bit serial data transmit/receive (Tx/Rx)
- Supports external clocks for the UART operation (UARTCLK)
- Programmable baud rate
- Supports IrDA 1.0
- Loopback mode for testing
- Each channel has internal 64-byte Tx FIFO and 64-byte Rx FIFO.

### DMA Controller

- 4-ch DMA controller
- Supports memory to memory, IO to memory, memory to IO, and IO to IO transfers
- Burst transfer mode to enhance the transfer rate

### A/D Converter & Touch Screen Interface

- 8-ch multiplexed ADC
- Max. 500KSPS and 10-bit Resolution
- Internal FET for direct Touch screen interface

### LCD Controller STN LCD Displays Feature

- Supports 3 types of STN LCD panels: 4-bit dual scan, 4-bit single scan, 8-bit single scan display type
- Supports monochrome mode, 4 gray levels, 16 gray levels, 256 colors and 4096 colors for STN LCD
- Supports multiple screen size
  - Maximum screen size: 2048x1024
  - Recommended screen size: max 800x600
  - Maximum virtual screen size is 4 Mbytes.
  - Maximum virtual screen size in 256 color mode: 4096x1024, 2048x2048, 1024x4096 and others

### TFT(Thin Film Transistor) Color Displays Feature

- Supports 1, 2, 4 or 8 bpp (bit-per-pixel) palette color displays for color TFT
- Supports 16 bpp non-palette true-color displays for color TFT
- Supports maximum 16M color TFT at 24 bpp mode
- Supports multiple screen size
  - Maximum screen size: 2048x1024
  - Recommended screen size: max 800x600
  - Maximum virtual screen size is 4Mbytes.
  - Maximum virtual screen size in 64K color mode: 2048x1024, and others

## FEATURES (Continued)

**Watchdog Timer**

- 16-bit Watchdog Timer
- Interrupt request or system reset at time-out

**IIC-Bus Interface**

- 1-ch Multi-Master IIC-Bus
- Serial, 8-bit oriented and bi-directional data transfers can be made at up to 100 Kbit/s in Standard mode or up to 400 Kbit/s in Fast mode.

**IIS-Bus Interface**

- 1-ch IIS-bus for audio interface with DMA-based operation
- Serial, 8-/16-bit per channel data transfers
- 128 Bytes (64-Byte + 64-Byte) FIFO for Tx/Rx
- Supports IIS format and MSB-justified data format

**USB Host**

- 2-port USB Host
- Complies with OHCI Rev. 1.0
- Compatible with USB Specification version 1.1

**USB Device**

- 1-port USB Device
- 5 Endpoints for USB Device
- Compatible with USB Specification version 1.1

**SD Host Interface**

- Compatible with SD Memory Card Protocol version 1.0
- Compatible with SDIO Card Protocol version 1.0
- Bytes FIFO for Tx/Rx
- DMA based or Interrupt based operation
- Compatible with Multimedia Card Protocol version 2.11

**SPI Interface**

- Compatible with 2-ch Serial Peripheral Interface Protocol version 2.11
- 2x8 bits Shift register for Tx/Rx
- DMA-based or interrupt-based operation

**Camera Interface**

- ITU601/ITU656-format input support (8-bit)
- YCrCb 4:2:2 to 4:2:0 down-sampling
- Up to 1016 Horizontal resolution support

**Operating Voltage Range**

- Core: 1.2V
- Memory :1.8V/ 2.5V/3.3V
- I/O : 3.3V

**Operating Frequency**

- Up to 400MHz

**Package**

- 289-FBGA

## BLOCK DIAGRAM

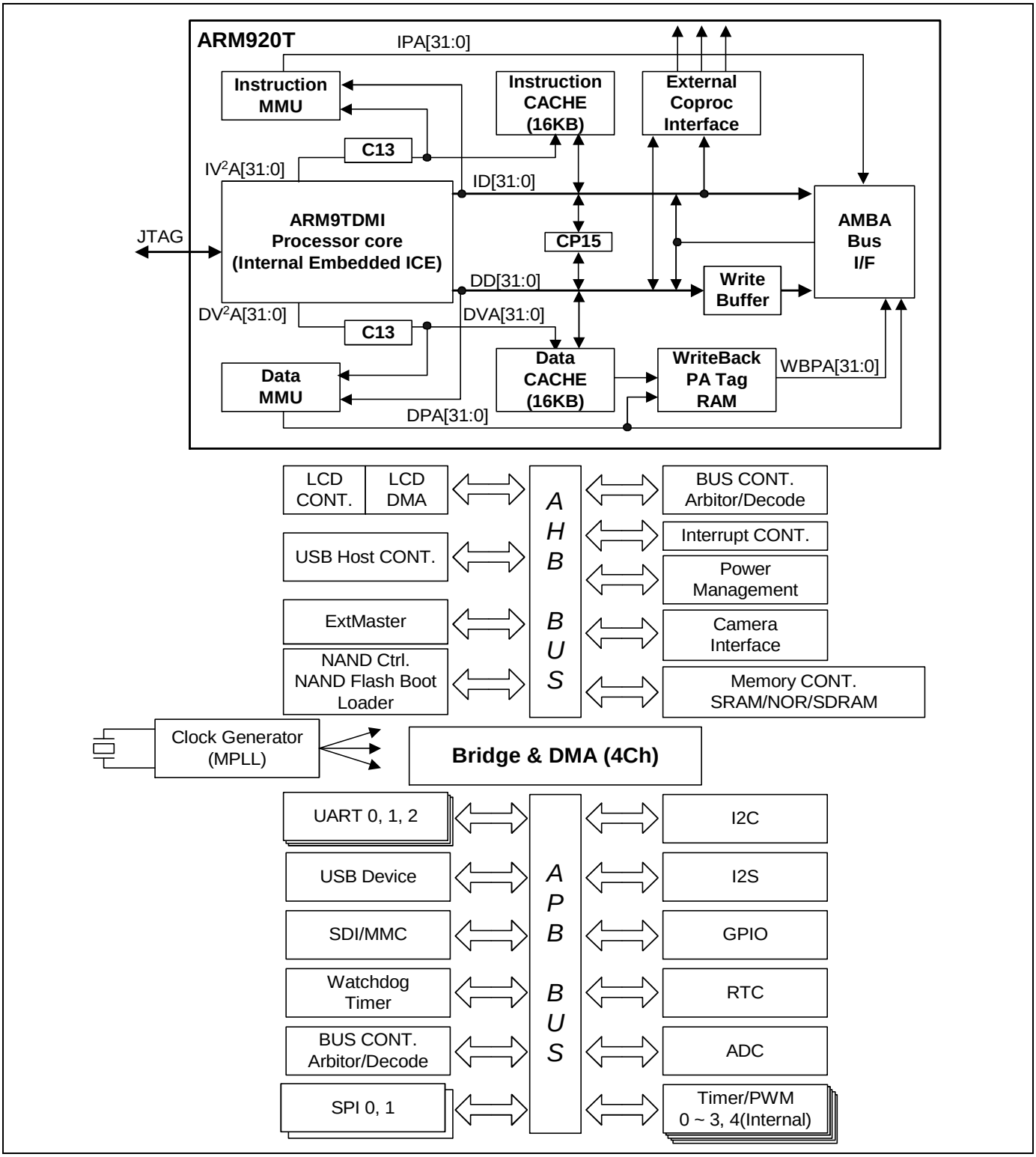



**Figure 1-1. S3C2440X Block Diagram**

## PIN ASSIGNMENTS

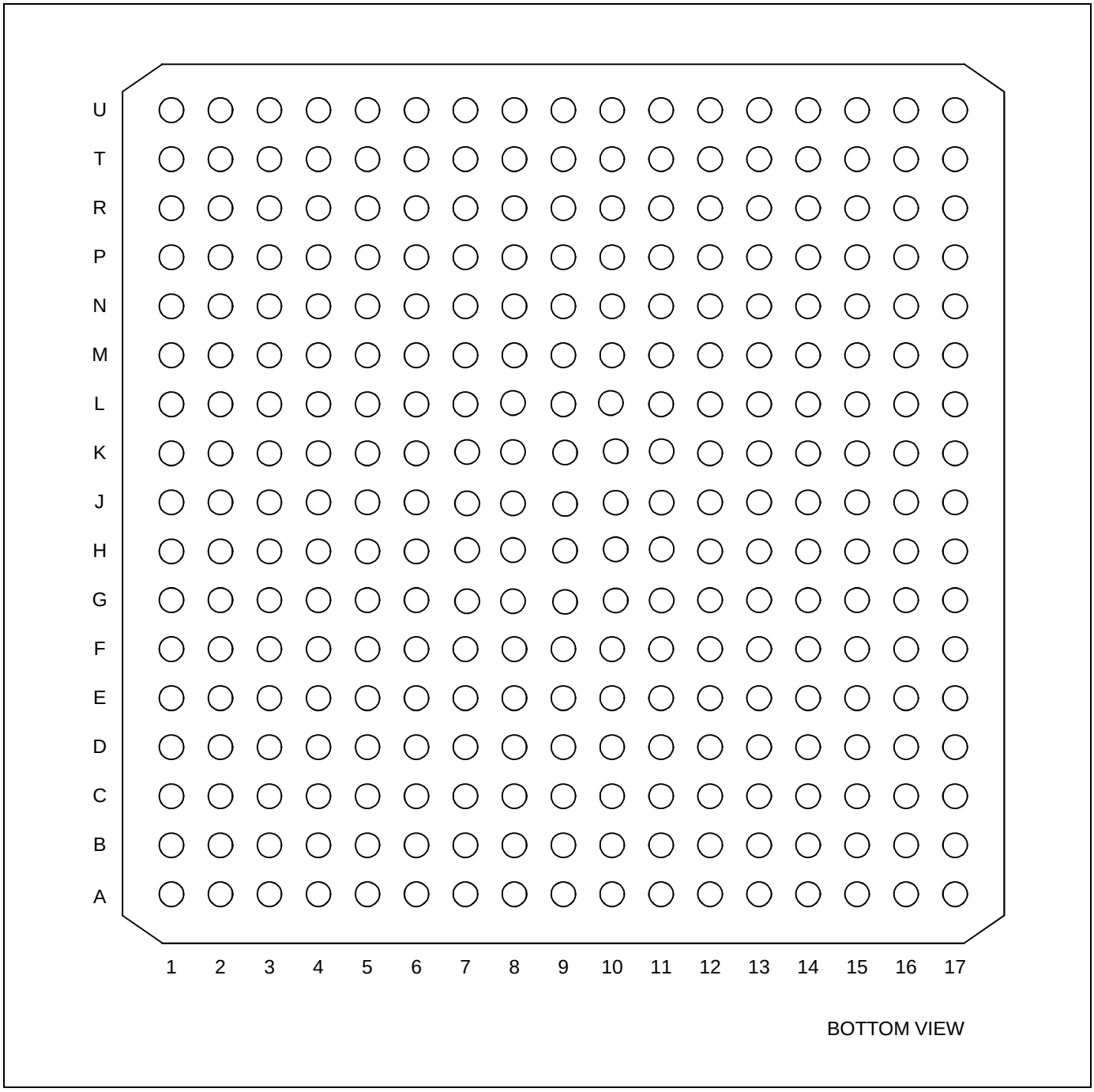



**Figure 1-2. S3C2440X Pin Assignments (289-FBGA)**

**Table 1-1. 289-Pin FBGA Pin Assignments – Pin Number Order (Sheet 1 of 3)**

| Pin Number | Pin Name | Pin Number | Pin Name | Pin Number | Pin Name |
|---|---|---|---|---|---|
| A1 | VDDi | C1 | VDDMOP | E1 | nFRE/GPA20 |
| A2 | SCKE | C2 | nGCS5/GPA16 | E2 | VSSMOP |
| A3 | VSSi | C3 | nGCS2/GPA13 | E3 | nGCS7 |
| A4 | VSSi | C4 | nGCS3/GPA14 | E4 | nWAIT |
| A5 | VSSMOP | C5 | nOE | E5 | nBE3 |
| A6 | VDDi | C6 | nSRAS | E6 | nWE |
| A7 | VSSMOP | C7 | ADDR4 | E7 | ADDR1 |
| A8 | ADDR10 | C8 | ADDR11 | E8 | ADDR6 |
| A9 | VDDMOP | C9 | ADDR15 | E9 | ADDR14 |
| A10 | VDDi | C10 | ADDR21/GPA6 | E10 | ADDR23/GPA8 |
| A11 | VSSMOP | C11 | ADDR24/GPA9 | E11 | DATA2 |
| A12 | VSSi | C12 | DATA1 | E12 | DATA20 |
| A13 | DATA3 | C13 | DATA6 | E13 | DATA19 |
| A14 | DATA7 | C14 | DATA11 | E14 | DATA18 |
| A15 | VSSMOP | C15 | DATA13 | E15 | DATA17 |
| A16 | VDDi | C16 | DATA16 | E16 | DATA21 |
| A17 | DATA10 | C17 | VSSi | E17 | DATA24 |
| B1 | VSSMOP | D1 | ALE/GPA18 | F1 | VDDi |
| B2 | nGCS1/GPA12 | D2 | nGCS6 | F2 | VSSi |
| B3 | SCLK1 | D3 | nGCS4/GPA15 | F3 | nFWE/GPA19 |
| B4 | SCLK0 | D4 | nBE0 | F4 | nFCE/GPA22 |
| B5 | nBE1 | D5 | nBE2 | F5 | CLE/GPA17 |
| B6 | VDDMOP | D6 | nSCAS | F6 | nGCS0 |
| B7 | ADDR2 | D7 | ADDR7 | F7 | ADDR0/GPA0 |
| B8 | ADDR9 | D8 | ADDR5 | F8 | ADDR3 |
| B9 | ADDR12 | D9 | ADDR16/GPA1 | F9 | ADDR18/GPA3 |
| B10 | VSSi | D10 | ADDR20/GPA5 | F10 | DATA4 |
| B11 | VDDi | D11 | ADDR26/GPA11 | F11 | DATA5 |
| B12 | VDDMOP | D12 | DATA0 | F12 | DATA27 |
| B13 | VSSMOP | D13 | DATA8 | F13 | DATA31 |
| B14 | VDDMOP | D14 | DATA14 | F14 | DATA26 |
| B15 | DATA9 | D15 | DATA12 | F15 | DATA22 |
| B16 | VDDMOP | D16 | VSSMOP | F16 | VDDi |
| B17 | DATA15 | D17 | VSSMOP | F17 | VDDMOP |

**Table 1-1. 289-Pin FBGA Pin Assignments – Pin Number Order (Sheet 2 of 3)**

| Pin Number | Pin Name | Pin Number | Pin Name | Pin Number | Pin Name |
|---|---|---|---|---|---|
| G1 | VSSOP | J1 | VDDOP | L1 | LEND/GPC0 |
| G2 | CAMHREF/GPJ10 | J2 | VDDiarm | L2 | VDDiarm |
| G3 | CAMDATA1/GPJ1 | J3 | CAMCLKOUT/GPJ11 | L3 | nXDACK0/GPB9 |
| G4 | VDDalive | J4 | CAMRESET/GPJ12 | L4 | VCLK/GPC1 |
| G5 | CAMPCLK/GPJ8 | J5 | TOUT1/GPB1 | L5 | nXBREQ/GPB6 |
| G6 | FRnB | J6 | TOUT0/GPB0 | L6 | VD1/GPC9 |
| G7 | CAMVSYNC/GPJ9 | J7 | TOUT2/GPB2 | L7 | VFRAME/GPC3 |
| G8 | ADDR8 | J8 | CAMDATA6/GPJ6 | L8 | I2SSDI/nSS0/GPE3 |
| G9 | ADDR17/GPA2 | J9 | SDDAT3/GPE10 | L9 | SPICLK0/GPE13 |
| G10 | ADDR25/GPA10 | J10 | EINT10/nSS0/GPG2 | L10 | EINT15/SPICLK1/GPG7 |
| G11 | DATA28 | J11 | TXD2/nRTS1/GPH6 | L11 | EINT22/GPG14 |
| G12 | DATA25 | J12 | PWREN | L12 | Xtortc |
| G13 | DATA23 | J13 | TCK | L13 | EINT2/GPF2 |
| G14 | XTIpll | J14 | TMS | L14 | EINT5/GPF5 |
| G15 | XTOpll | J15 | RXD2/nCTS1/GPH7 | L15 | EINT6/GPF6 |
| G16 | DATA29 | J16 | TDO | L16 | EINT7/GPF7 |
| G17 | VSSi | J17 | VDDalive | L17 | nRTS0/GPH1 |
| H1 | VSSiarm | K1 | VSSiarm | M1 | VLINE/GPC2 |
| H2 | CAMDATA7/GPJ7 | K2 | nXBACK/GPB5 | M2 | LCD_LPCREV/GPC6 |
| H3 | CAMDATA4/GPJ4 | K3 | TOUT3/GPB3 | M3 | LCD_LPCOE/GPC5 |
| H4 | CAMDATA3/GPJ3 | K4 | TCLK0/GPB4 | M4 | VM/GPC4 |
| H5 | CAMDATA2/GPJ2 | K5 | nXDREQ1/GPB8 | M5 | VD9/GPD1 |
| H6 | CAMDATA0/GPJ0 | K6 | nXDREQ0/GPB10 | M6 | VD6/GPC14 |
| H7 | CAMDATA5/GPJ5 | K7 | nXDACK1/GPB7 | M7 | VD16/SPIMISO1/GPD8 |
| H8 | ADDR13 | K8 | SDCMD/GPE6 | M8 | SDDAT1/GPE8 |
| H9 | ADDR19/GPA4 | K9 | SPIMISO0/GPE11 | M9 | IICSDA/GPE15 |
| H10 | ADDR22/GPA7 | K10 | EINT13/SPIMISO1/GPG5 | M10 | EINT20/GPG12 |
| H11 | VSSOP1 | K11 | nCTS0/GPH0 | M11 | EINT17/nRTS1/GPG9 |
| H12 | EXTCLK | K12 | VDDOP | M12 | VSSA_UPLL |
| H13 | DATA30 | K13 | TXD0/GPH2 | M13 | VDDA_UPLL |
| H14 | nBATT_FLT | K14 | RXD0/GPH3 | M14 | Xtirtc |
| H15 | nTRST | K15 | UARTCLK/GPH8 | M15 | EINT3/GPF3 |
| H16 | nRESET | K16 | TXD1/GPH4 | M16 | EINT1/GPF1 |
| H17 | TDI | K17 | RXD1/GPH5 | M17 | EINT4/GPF4 |

**Table 1-1. 289-Pin FBGA Pin Assignments – Pin Number Order (Sheet 3 of 3)**

| Pin Number | Pin Name | Pin Number | Pin Name | Pin Number | Pin Name |
|---|---|---|---|---|---|
| N1 | VSSOP | R1 | VD3/GPC11 | U1 | VDDiarm |
| N2 | VD0/GPC8 | R2 | VD8/GPD0 | U2 | VDDiarm |
| N3 | VD4/GPC12 | R3 | VD11/GPD3 | U3 | VSSOP |
| N4 | VD2/GPC10 | R4 | VD13/GPD5 | U4 | VSSiarm |
| N5 | VD10/GPD2 | R5 | VD18/SPICLK1/GPD10 | U5 | VD23/nSS0/GPD15 |
| N6 | VD15/GPD7 | R6 | VD21 /GPD13 | U6 | I2SSDO/I2SSDI/GPE4 |
| N7 | VD22/nSS1/GPD14 | R7 | I2SSCLK/GPE1 | U7 | VSSiarm |
| N8 | SDCLK/GPE5 | R8 | SDDAT0/GPE7 | U8 | IICSCL/GPE14 |
| N9 | EINT8/GPG0 | R9 | CLKOUT0/GPH9 | U9 | VSSOP |
| N10 | EINT18/nCTS1/GPG10 | R10 | EINT11/nSS1/GPG3 | U10 | VSSiarm |
| N11 | DP0 | R11 | EINT14/SPIMOSI1/GPG6 | U11 | VDDiarm |
| N12 | DN1/PDN0 | R12 | NCON | U12 | EINT19/TCLK1/GPG11 |
| N13 | nRSTOUT/GPA21 | R13 | OM1 | U13 | EINT23/GPG15 |
| N14 | MPLLCAP | R14 | AIN0 | U14 | DP1/PDP0 |
| N15 | VDD_RTC | R15 | AIN2 | U15 | VSSOP |
| N16 | VDDA_MPLL | R16 | AIN6 | U16 | Vref |
| N17 | EINT0/GPF0 | R17 | VSSA_MPLL | U17 | AIN1 |
| P1 | LCD_LPCREVB/GPC7 | T1 | VSSiarm | | |
| P2 | VD5/GPC13 | T2 | VSSiarm | | |
| P3 | VD7/GPC15 | T3 | VDDOP | | |
| P4 | VD12/GPD4 | T4 | VD17/SPIMOSI1/GPD9 | | |
| P5 | VD14/GPD6 | T5 | VD19/GPD11 | | |
| P6 | VD20/GPD12 | T6 | VDDiarm | | |
| P7 | I2SLRCK/GPE0 | T7 | CDCLK/GPE2 | | |
| P8 | SDDAT2/GPE9 | T8 | VDDiarm | | |
| P9 | SPIMOSI0/GPE12 | T9 | EINT9/GPG1 | | |
| P10 | CLKOUT1/GPH10 | T10 | EINT16/GPG8 | | |
| P11 | EINT12/LCD_PWREN | T11 | EINT21/GPG13 | | |
| P12 | DN0 | T12 | VDDOP | | |
| P13 | OM2 | T13 | OM3 | | |
| P14 | VDDA_ADC | T14 | VSSA_ADC | | |
| P15 | AIN3 | T15 | OM0 | | |
| P16 | AIN7 | T16 | AIN4 | | |
| P17 | UPLLCAP | T17 | AIN5 | | |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 1 of 9)**

| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| F7 | ADDR0/GPA0 | ADDR0 | Hi-z/– | O(L)/– | O(L) | t10s |
| E7 | ADDR1 | ADDR1 | Hi-z | O(L) | O(L) | t10s |
| B7 | ADDR2 | ADDR2 | Hi-z | O(L) | O(L) | t10s |
| F8 | ADDR3 | ADDR3 | Hi-z | O(L) | O(L) | t10s |
| C7 | ADDR4 | ADDR4 | Hi-z | O(L) | O(L) | t10s |
| D8 | ADDR5 | ADDR5 | Hi-z | O(L) | O(L) | t10s |
| E8 | ADDR6 | ADDR6 | Hi-z | O(L) | O(L) | t10s |
| D7 | ADDR7 | ADDR7 | Hi-z | O(L) | O(L) | t10s |
| G8 | ADDR8 | ADDR8 | Hi-z | O(L) | O(L) | t10s |
| B8 | ADDR9 | ADDR9 | Hi-z | O(L) | O(L) | t10s |
| A8 | ADDR10 | ADDR10 | Hi-z | O(L) | O(L) | t10s |
| C8 | ADDR11 | ADDR11 | Hi-z | O(L) | O(L) | t10s |
| B9 | ADDR12 | ADDR12 | Hi-z | O(L) | O(L) | t10s |
| H8 | ADDR13 | ADDR13 | Hi-z | O(L) | O(L) | t10s |
| E9 | ADDR14 | ADDR14 | Hi-z | O(L) | O(L) | t10s |
| C9 | ADDR15 | ADDR15 | Hi-z | O(L) | O(L) | t10s |
| D9 | ADDR16/GPA1 | ADDR16 | Hi-z/– | O(L)/– | O(L) | t10s |
| G9 | ADDR17/GPA2 | ADDR17 | Hi-z/– | O(L)/– | O(L) | t10s |
| F9 | ADDR18/GPA3 | ADDR18 | Hi-z/– | O(L)/– | O(L) | t10s |
| H9 | ADDR19/GPA4 | ADDR19 | Hi-z/– | O(L)/– | O(L) | t10s |
| D10 | ADDR20/GPA5 | ADDR20 | Hi-z/– | O(L)/– | O(L) | t10s |
| C10 | ADDR21/GPA6 | ADDR21 | Hi-z/– | O(L)/– | O(L) | t10s |
| H10 | ADDR22/GPA7 | ADDR22 | Hi-z/– | O(L)/– | O(L) | t10s |
| E10 | ADDR23/GPA8 | ADDR23 | Hi-z/– | O(L)/– | O(L) | t10s |
| C11 | ADDR24/GPA9 | ADDR24 | Hi-z/– | O(L)/– | O(L) | t10s |
| G10 | ADDR25/GPA10 | ADDR25 | Hi-z/– | O(L)/– | O(L) | t10s |
| D11 | ADDR26/GPA11 | ADDR26 | Hi-z/– | O(L)/– | O(L) | t10s |
| R14 | AIN0 | AIN0 | – | – | AI | r10 |
| U17 | AIN1 | AIN1 | – | – | AI | r10 |
| R15 | AIN2 | AIN2 | – | – | AI | r10 |
| P15 | AIN3 | AIN3 | – | – | AI | r10 |
| T16 | YM/AIN4 | AIN4 | –/– | –/– | AI | r10 |
| T17 | YP/AIN5 | YP | –/– | –/– | AI | r10 |
| R16 | XM/AIN6 | AIN6 | –/– | –/– | AI | r10 |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 2 of 9)**

| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| P16 | XP/AIN7 | XP | –/– | –/– | AI | r10 |
| H6 | CAMDATA0/GPJ0 | GPJ0 | –/– | Hi-z/– | I | t8 |
| G3 | CAMDATA1/GPJ1 | GPJ1 | –/– | Hi-z/– | I | t8 |
| H5 | CAMDATA2/GPJ2 | GPJ2 | –/– | Hi-z/– | I | t8 |
| H4 | CAMDATA3/GPJ3 | GPJ3 | –/– | Hi-z/– | I | t8 |
| H3 | CAMDATA4/GPJ4 | GPJ4 | –/– | Hi-z/– | I | t8 |
| H7 | CAMDATA5/GPJ5 | GPJ5 | –/– | Hi-z/– | I | t8 |
| J8 | CAMDATA6/GPJ6 | GPJ6 | –/– | Hi-z/– | I | t8 |
| H2 | CAMDATA7/GPJ7 | GPJ7 | –/– | Hi-z/– | I | t8 |
| G5 | CAMPCLK/GPJ8 | GPJ8 | –/– | Hi-z/– | I | t8 |
| G7 | CAMVSYNC/GPJ9 | GPJ9 | –/– | Hi-z/– | I | t8 |
| G2 | CAMHREF/GPJ10 | GPJ10 | –/– | Hi-z/– | I | t8 |
| J3 | CAMPCLKOUT/GPJ11 | GPJ11 | –/– | O(L)/– | I | t8 |
| J4 | CAMRESET/GPJ12 | GPJ12 | –/– | O(L)/– | I | t8 |
| D12 | DATA0 | DATA0 | Hi-z | Hi-z,O(L) | I | b12s |
| C12 | DATA1 | DATA1 | Hi-z | Hi-z,O(L) | I | b12s |
| E11 | DATA2 | DATA2 | Hi-z | Hi-z,O(L) | I | b12s |
| A13 | DATA3 | DATA3 | Hi-z | Hi-z,O(L) | I | b12s |
| F10 | DATA4 | DATA4 | Hi-z | Hi-z,O(L) | I | b12s |
| F11 | DATA5 | DATA5 | Hi-z | Hi-z,O(L) | I | b12s |
| C13 | DATA6 | DATA6 | Hi-z | Hi-z,O(L) | I | b12s |
| A14 | DATA7 | DATA7 | Hi-z | Hi-z,O(L) | I | b12s |
| D13 | DATA8 | DATA8 | Hi-z | Hi-z,O(L) | I | b12s |
| B15 | DATA9 | DATA9 | Hi-z | Hi-z,O(L) | I | b12s |
| A17 | DATA10 | DATA10 | Hi-z | Hi-z,O(L) | I | b12s |
| C14 | DATA11 | DATA11 | Hi-z | Hi-z,O(L) | I | b12s |
| D15 | DATA12 | DATA12 | Hi-z | Hi-z,O(L) | I | b12s |
| C15 | DATA13 | DATA13 | Hi-z | Hi-z,O(L) | I | b12s |
| D14 | DATA14 | DATA14 | Hi-z | Hi-z,O(L) | I | b12s |
| B17 | DATA15 | DATA15 | Hi-z | Hi-z,O(L) | I | b12s |
| C16 | DATA16 | DATA16 | Hi-z | Hi-z,O(L) | I | b12s |
| E15 | DATA17 | DATA17 | Hi-z | Hi-z,O(L) | I | b12s |
| E14 | DATA18 | DATA18 | Hi-z | Hi-z,O(L) | I | b12s |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 3 of 9)**

| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| E13 | DATA19 | DATA19 | Hi-z | Hi-z,O(L) | I | b12s |
| E12 | DATA20 | DATA20 | Hi-z | Hi-z,O(L) | I | b12s |
| E16 | DATA21 | DATA21 | Hi-z | Hi-z,O(L) | I | b12s |
| F15 | DATA22 | DATA22 | Hi-z | Hi-z,O(L) | I | b12s |
| G13 | DATA23 | DATA23 | Hi-z | Hi-z,O(L) | I | b12s |
| E17 | DATA24 | DATA24 | Hi-z | Hi-z,O(L) | I | b12s |
| G12 | DATA25 | DATA25 | Hi-z | Hi-z,O(L) | I | b12s |
| F14 | DATA26 | DATA26 | Hi-z | Hi-z,O(L) | I | b12s |
| F12 | DATA27 | DATA27 | Hi-z | Hi-z,O(L) | I | b12s |
| G11 | DATA28 | DATA28 | Hi-z | Hi-z,O(L) | I | b12s |
| G16 | DATA29 | DATA29 | Hi-z | Hi-z,O(L) | I | b12s |
| H13 | DATA30 | DATA30 | Hi-z | Hi-z,O(L) | I | b12s |
| F13 | DATA31 | DATA31 | Hi-z | Hi-z,O(L) | I | b12s |
| P12 | DN0 | DN0 | — | — | AI | us |
| N11 | DP0 | DP0 | — | — | AI | us |
| N12 | DN1/PDN0 | DN1 | –/– | — | AI | us |
| U14 | DP1/PDP0 | DP1 | –/– | — | AI | us |
| N17 | EINT0/GPF0 | GPF0 | –/– | Hi-z/– | I | t8 |
| M16 | EINT1/GPF1 | GPF1 | –/– | Hi-z/– | I | t8 |
| L13 | EINT2/GPF2 | GPF2 | –/– | Hi-z/– | I | t8 |
| M15 | EINT3/GPF3 | GPF3 | –/– | Hi-z/– | I | t8 |
| M17 | EINT4/GPF4 | GPF4 | –/– | Hi-z/– | I | t8 |
| L14 | EINT5/GPF5 | GPF5 | –/– | Hi-z/– | I | t8 |
| L15 | EINT6/GPF6 | GPF6 | –/– | Hi-z/– | I | t8 |
| L16 | EINT7/GPF7 | GPF7 | –/– | Hi-z/– | I | t8 |
| N9 | EINT8/GPG0 | GPG0 | –/– | Hi-z/– | I | t8 |
| T9 | EINT9/GPG1 | GPG1 | –/– | Hi-z/– | I | t8 |
| J10 | EINT10/nSS0/GPG2 | GPG2 | –/–/– | Hi-z/Hi-z/– | I | t8 |
| R10 | EINT11/nSS1/GPG3 | GPG3 | –/–/– | Hi-z/Hi-z/– | I | t8 |
| P11 | EINT12/LCD_PWREN/GPG4 | GPG4 | –/–/– | Hi-z/O(L)/– | I | t8 |
| K10 | EINT13/SPIMISO1/GPG5 | GPG5 | –/–/– | Hi-z/Hi-z/– | I | tt8 |
| R11 | EINT14/SPIMOSI1/GPG6 | GPG6 | –/–/– | Hi-z/Hi-z/– | I | tt8 |
| L10 | EINT15/SPICLK1/GPG7 | GPG7 | –/–/– | Hi-z/Hi-z/– | I | tt8 |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 4 of 9)**

| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| T10 | EINT16/GPG8 | GPG8 | –/– | Hi-z/– | I | t8 |
| M11 | EINT17/nRTS1/GPG9 | GPG9 | –/–/– | Hi-z/O(H)/– | I | t8 |
| N10 | EINT18/nCTS1/GPG10 | GPG10 | –/–/– | Hi-z/Hi-z/– | I | t8 |
| U12 | EINT19/TCLK1/GPG11 | GPG11 | –/–/– | Hi-z/Hi-z/– | I | t12 |
| M10 | EINT20/GPG12 | GPG12 | –/– | Hi-z/– | I | t12 |
| T11 | EINT21/GPG13 | GPG13 | –/– | Hi-z/– | I | t12 |
| L11 | EINT22/GPG14 | GPG14 | –/– | Hi-z/– | I | t12 |
| U13 | EINT23/GPG15 | GPG15 | –/– | Hi-z/– | I | t12 |
| H12 | EXTCLK | EXTCLK | – | – | AI | is |
| P17 | UPLLCAP | UPLLCAP | – | – | AI | r50 |
| N14 | MPLLCAP | MPLLCAP | – | – | AI | r50 |
| H14 | nBATT_FLT | nBATT_FLT | – | – | I | is |
| D4 | nBE0 | nBE0 | Hi-z | Hi-z,O(H) | O(H) | t10s |
| B5 | nBE1 | nBE1 | Hi-z | Hi-z,O(H) | O(H) | t10s |
| D5 | nBE2 | nBE2 | Hi-z | Hi-z,O(H) | O(H) | t10s |
| E5 | nBE3 | nBE3 | Hi-z | Hi-z,O(H) | O(H) | t10s |
| R12 | NCON | NCON | – | – | I | is |
| G6 | FRnB | FRnB | – | Hi-z,O(L) | I | d2s |
| F3 | nFWE/GPA19 | GPA19 | O(H)/– | Hi-z,O(H)/– | O(H) | t10s |
| E1 | nFRE/GPA20 | GPA20 | O(H)/– | Hi-z,O(H)/– | O(H) | t10s |
| F4 | nFCE/GPA22 | GPA21 | O(H)/– | Hi-z,O(H)/– | O(H) | t10s |
| F5 | CLE/GPA17 | GPA17 | O(L)/– | Hi-z,O(L)/– | O(L) | t10s |
| D1 | ALE/GPA18 | GPA18 | O(L)/– | Hi-z,O(L)/– | O(L) | t10s |
| N13 | nRSTOUT/GPA21 | GPA21 | –/– | O(L)/– | O(L) | b8 |
| C5 | nOE | nOE | Hi-z | Hi-z,O(H) | O(H) | t10s |
| H16 | nRESET | nRESET | – | – | I | is |
| F6 | nGCS0 | nGCS0 | Hi-z | Hi-z,O(H) | O(H) | t10s |
| B2 | nGCS1/GPA12 | GPA12 | Hi-z/– | Hi-z,O(H)/– | O(H) | t10s |
| C3 | nGCS2/GPA13 | GPA13 | Hi-z/– | Hi-z,O(H)/– | O(H) | t10s |
| C4 | nGCS3/GPA14 | GPA14 | Hi-z/– | Hi-z,O(H)/– | O(H) | t10s |
| D3 | nGCS4/GPA15 | GPA15 | Hi-z/– | Hi-z,O(H)/– | O(H) | t10s |
| C2 | nGCS5/GPA16 | GPA16 | Hi-z/– | Hi-z,O(H)/– | O(H) | t10s |
| D2 | nGCS6 | nGCS6 | Hi-z | Hi-z,O(H) | O(H) | t10s |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 5 of 9)**

| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| E3 | nGCS7 | nGCS7 | Hi-z | Hi-z,O(H) | O(H) | t10s |
| D6 | nSCAS | nSCAS | Hi-z | Hi-z,O(H) | O(H) | t10s |
| C6 | nSRAS | nSRAS | Hi-z | Hi-z,O(H) | O(H) | t10s |
| H15 | nTRST | nTRST | I | – | I | is |
| E4 | nWAIT | nWAIT | – | Hi-z,O(L) | I | d2s |
| E6 | nWE | nWE | Hi-z | Hi-z,O(H) | O(H) | t10s |
| J6 | TOUT0/GPB0 | GPB0 | –/– | O(L)/– | I | t8 |
| J5 | TOUT1/GPB1 | GPB1 | –/– | O(L)/– | I | t8 |
| J7 | TOUT2/GPB2 | GPB2 | –/– | O(L)/– | I | t8 |
| K3 | TOUT3/GPB3 | GPB3 | –/– | O(L)/– | I | t8 |
| K4 | TCLK0/GPB4 | GPB4 | –/– | –/– | I | t8 |
| K2 | nXBACK/GPB5 | GPB5 | –/– | O(H)/– | I | t8 |
| L5 | nXBREQ/GPB6 | GPB6 | –/– | –/– | I | t8 |
| K7 | nXDACK1/GPB7 | GPB7 | –/– | O(H)/– | I | t8 |
| K5 | nXDREQ1/GPB8 | GPB8 | –/– | –/– | I | t8 |
| L3 | nXDACK0/GPB9 | GPB9 | –/– | O(H)/– | I | t8 |
| K6 | nXDREQ0/GPB10 | GPB10 | –/– | –/– | I | t8 |
| T15 | OM0 | OM0 | – | – | I | is |
| R13 | OM1 | OM1 | – | – | I | is |
| P13 | OM2 | OM2 | – | – | I | is |
| T13 | OM3 | OM3 | – | – | I | is |
| J12 | PWREN | PWREN | O(H) | O(L) | O(H) | b8 |
| K11 | nCTS0/GPH0 | GPH0 | –/– | –/– | I | t8 |
| L17 | nRTS0/GPH1 | GPH1 | –/– | O(H)/– | I | t8 |
| K13 | TXD0/GPH2 | GPH2 | –/– | O(H)/– | I | t8 |
| K14 | RXD0/GPH3 | GPH3 | –/– | –/– | I | t8 |
| K16 | TXD1/GPH4 | GPH4 | –/– | O(H)/– | I | t8 |
| K17 | RXD1/GPH5 | GPH5 | –/– | –/– | I | t8 |
| J11 | TXD2/nRTS1/GPH6 | GPH6 | –/–/– | O(H)/O(H)/– | I | t8 |
| J15 | RXD2/nCTS1/GPH7 | GPH7 | –/–/– | Hi-z/Hi-z/– | I | t8 |
| K15 | UARTCLK/GPH8 | GPH8 | –/– | Hi-z/– | I | t8 |
| R9 | CLKOUT0/GPH9 | GPH9 | –/– | O(L)/– | I | t12 |
| P10 | CLKOUT1/GPH10 | GPH10 | –/– | O(L)/– | I | t12 |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 6 of 9)**

| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| A2 | SCKE | SCKE | Hi-z | O(L) | O(H) | t10s |
| B4 | SCLK0 | SCLK0 | Hi-z | O(L) | O(SCLK) | t12s |
| B3 | SCLK1 | SCLK1 | Hi-z | O(L) | O(SCLK) | t12s |
| P7 | I2SLRCK/GPE0 | GPE0 | –/– | Hi-z/– | I | t8 |
| R7 | I2SSCLK/GPE1 | GPE1 | –/– | Hi-z/– | I | t8 |
| T7 | CDCLK/GPE2 | GPE2 | –/– | Hi-z/– | I | t8 |
| L8 | I2SSDI/nSS0/GPE3 | GPE3 | –/–/– | Hi-z/Hi-z/– | I | t8 |
| U6 | I2SSDO/I2SSDI/GPE4 | GPE4 | –/–/– | O(L)/Hi-z/– | I | t8 |
| N8 | SDCLK/GPE5 | GPE5 | –/– | O(L)/– | I | t8 |
| K8 | SDCMD/GPE6 | GPE6 | –/– | Hi-z/– | I | t8 |
| R8 | SDDAT0/GPE7 | GPE7 | –/– | Hi-z/– | I | t8 |
| M8 | SDDAT1/GPE8 | GPE8 | –/– | Hi-z/– | I | t8 |
| P8 | SDDAT2/GPE9 | GPE9 | –/– | Hi-z/– | I | t8 |
| J9 | SDDAT3/GPE10 | GPE10 | –/– | Hi-z/– | I | t8 |
| K9 | SPIMISO0/GPE11 | GPE11 | –/– | Hi-z/– | I | tt8 |
| P9 | SPIMOSI0/GPE12 | GPE12 | –/– | Hi-z/– | I | tt8 |
| L9 | SPICLK0/GPE13 | GPE13 | –/– | Hi-z/– | I | tt8 |
| U8 | IICSCL/GPE14 | GPE14 | –/– | Hi-z/– | I | d8 |
| M9 | IICSDA/GPE15 | GPE15 | –/– | Hi-z/– | I | d8 |
| J13 | TCK | TCK | I | – | I | is |
| H17 | TDI | TDI | I | – | I | is |
| J16 | TDO | TDO | O | O | O | ot |
| J14 | TMS | TMS | I | – | I | is |
| L1 | LEND/GPC0 | GPC0 | –/– | O(L)/– | I | t8 |
| L4 | VCLK/GPC1 | GPC1 | –/– | O(L)/– | I | t8 |
| M1 | VLINE/GPC2 | GPC2 | –/– | O(L)/– | I | t8 |
| L7 | VFRAME/GPC3 | GPC3 | –/– | O(L)/– | I | t8 |
| M4 | VM/GPC4 | GPC4 | –/– | O(L)/– | I | t8 |
| M3 | LCD_LPCOE/GPC5 | GPC5 | –/– | O(L)/– | I | t8 |
| M2 | LCD_LPCREV/GPC6 | GPC6 | –/– | O(L)/– | I | t8 |
| P1 | LCD_LPCREVB/GPC7 | GPC7 | –/– | O(L)/– | I | t8 |
| N2 | VD0/GPC8 | GPC8 | –/– | O(L)/– | I | t8 |
| L6 | VD1/GPC9 | GPC9 | –/– | O(L)/– | I | t8 |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 7 of 9)**

| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| N4 | VD2/GPC10 | GPC10 | –/– | O(L)/– | I | t8 |
| R1 | VD3/GPC11 | GPC11 | –/– | O(L)/– | I | t8 |
| N3 | VD4/GPC12 | GPC12 | –/– | O(L)/– | I | t8 |
| P2 | VD5/GPC13 | GPC13 | –/– | O(L)/– | I | t8 |
| M6 | VD6/GPC14 | GPC14 | –/– | O(L)/– | I | t8 |
| P3 | VD7/GPC15 | GPC15 | –/– | O(L)/– | I | t8 |
| R2 | VD8/GPD0 | GPD0 | –/– | O(L)/– | I | t8 |
| M5 | VD9/GPD1 | GPD1 | –/– | O(L)/– | I | t8 |
| N5 | VD10/GPD2 | GPD2 | –/– | O(L)/– | I | t8 |
| R3 | VD11/GPD3 | GPD3 | –/– | O(L)/– | I | t8 |
| P4 | VD12/GPD4 | GPD4 | –/– | O(L)/– | I | t8 |
| R4 | VD13/USBTXDN1/GPD5 | GPD5 | –/–/– | O(L)/O/– | I | t8 |
| P5 | VD14/USBTXDP1/GPD6 | GPD6 | –/–/– | O(L)/O/– | I | t8 |
| N6 | VD15/USBOEN1/GPD7 | GPD7 | –/–/– | O(L)/O/– | I | t8 |
| M7 | VD16/SPIMISO1/GPD8 | GPD8 | –/–/– | O(L)/Hi-z/– | I | tt8 |
| T4 | VD17/SPIMOSI1/GPD9 | GPD9 | –/–/– | O(L)/Hi-z/– | I | tt8 |
| R5 | VD18/SPICLK1/GPD10 | GPD10 | –/–/– | O(L)/Hi-z/– | I | tt8 |
| T5 | VD19/USBRXDP1/GPD11 | GPD11 | –/–/– | O(L)/Hi-z/– | I | t8 |
| P6 | VD20/USBRXDN1/GPD12 | GPD12 | –/–/– | O(L)/Hi-z/– | I | t8 |
| R6 | VD21/USBRXD1/GPD13 | GPD13 | –/–/– | O(L)/Hi-z/– | I | t8 |
| N7 | VD22/nSS1/GPD14 | GPD14 | –/–/– | O(L)/Hi-z/– | I | t8 |
| U5 | VD23/nSS0/GPD15 | GPD15 | –/–/– | O(L)/Hi-z/– | I | t8 |
| U16 | Vref | Vref | – | – | AI | ia |
| G14 | XTIpll | XTIpll | – | – | AI | m26 |
| M14 | Xtirtc | Xtirtc | – | – | AI | nc |
| G15 | XTOpll | XTOpll | – | – | AO | m26 |
| L12 | Xtortc | Xtortc | – | – | AO | nc |
| N15 | VDD_RTC | VDD_RTC | P | P | P | drtc |
| P14 | VDDA_ADC | VDDA_ADC | P | P | P | d33t |
| N16 | VDDA_MPLL | VDDA_MPLL | P | P | P | d33t |
| M13 | VDDA_UPLL | VDDA_UPLL | P | P | P | d33t |
| G4 | VDDalive | VDDalive | P | P | P | d12i |
| J17 | VDDalive | VDDalive | P | P | P | d12i |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 8 of 9)**

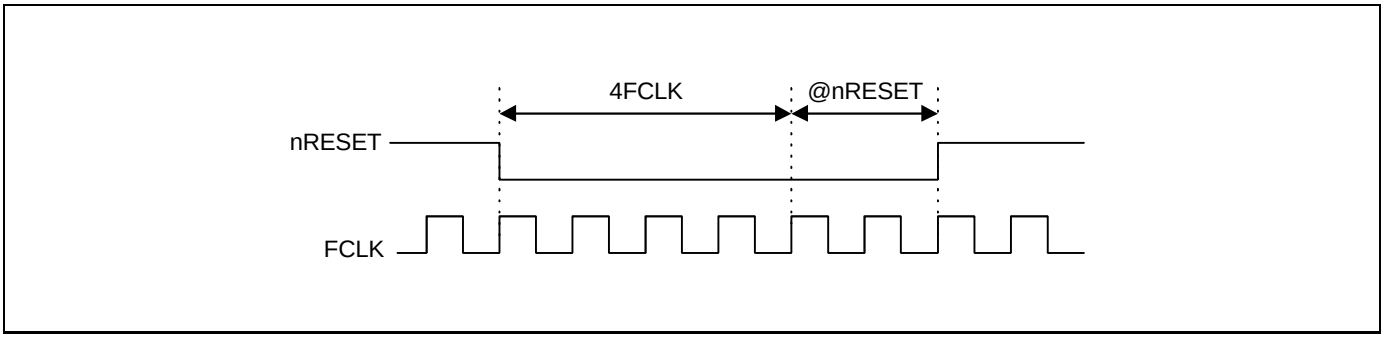
| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| A1 | VDDi | VDDi | P | P | P | d12c |
| A10 | VDDi | VDDi | P | P | P | d12c |
| A16 | VDDi | VDDi | P | P | P | d12c |
| A6 | VDDi | VDDi | P | P | P | d12c |
| B11 | VDDi | VDDi | P | P | P | d12c |
| F1 | VDDi | VDDi | P | P | P | d12c |
| F16 | VDDi | VDDi | P | P | P | d12c |
| J2 | VDDiarm | VDDiarm | P | P | P | d12c |
| L2 | VDDiarm | VDDiarm | P | P | P | d12c |
| T6 | VDDiarm | VDDiarm | P | P | P | d12c |
| T8 | VDDiarm | VDDiarm | P | P | P | d12c |
| U1 | VDDiarm | VDDiarm | P | P | P | d12c |
| U11 | VDDiarm | VDDiarm | P | P | P | d12c |
| U2 | VDDiarm | VDDiarm | P | P | P | d12c |
| A9 | VDDMOP | VDDMOP | P | P | P | d33o |
| B12 | VDDMOP | VDDMOP | P | P | P | d33o |
| B14 | VDDMOP | VDDMOP | P | P | P | d33o |
| B16 | VDDMOP | VDDMOP | P | P | P | d33o |
| B6 | VDDMOP | VDDMOP | P | P | P | d33o |
| C1 | VDDMOP | VDDMOP | P | P | P | d33o |
| F17 | VDDMOP | VDDMOP | P | P | P | d33o |
| J1 | VDDOP | VDDOP | P | P | P | d33o |
| T12 | VDDOP | VDDOP | P | P | P | d33o |
| T3 | VDDOP | VDDOP | P | P | P | d33o |
| K12 | VDDOP | VDDOP | P | P | P | d33o |
| T14 | VSSA_ADC | VSSA_ADC | P | P | P | st |
| R17 | VSSA_MPLL | VSSA_MPLL | P | P | P | st |
| M12 | VSSA_UPLL | VSSA_UPLL | P | P | P | st |
| A12 | VSSi | VSSi | P | P | P | si |
| A3 | VSSi | VSSi | P | P | P | si |
| A4 | VSSi | VSSi | P | P | P | si |
| B10 | VSSi | VSSi | P | P | P | si |
| C17 | VSSi | VSSi | P | P | P | si |

**Table 1-2. S3C2440X 289-Pin FBGA Pin Assignments (Sheet 9 of 9)**

| Pin Number | Pin Name | Default Function | I/O State @BUS REQ | I/O State @Sleep | I/O State @nRESET | I/O Type |
|---|---|---|---|---|---|---|
| F2 | VSSi | VSSi | P | P | P | si |
| G17 | VSSi | VSSi | P | P | P | si |
| H1 | VSSiarm | VSSiarm | P | P | P | si |
| K1 | VSSiarm | VSSiarm | P | P | P | si |
| T1 | VSSiarm | VSSiarm | P | P | P | si |
| T2 | VSSiarm | VSSiarm | P | P | P | si |
| U10 | VSSiarm | VSSiarm | P | P | P | si |
| U4 | VSSiarm | VSSiarm | P | P | P | si |
| U7 | VSSiarm | VSSiarm | P | P | P | si |
| A11 | VSSMOP | VSSMOP | P | P | P | so |
| A15 | VSSMOP | VSSMOP | P | P | P | so |
| A5 | VSSMOP | VSSMOP | P | P | P | so |
| A7 | VSSMOP | VSSMOP | P | P | P | so |
| B1 | VSSMOP | VSSMOP | P | P | P | so |
| B13 | VSSMOP | VSSMOP | P | P | P | so |
| D16 | VSSMOP | VSSMOP | P | P | P | so |
| D17 | VSSMOP | VSSMOP | P | P | P | so |
| E2 | VSSMOP | VSSMOP | P | P | P | so |
| G1 | VSSOP | VSSOP | P | P | P | so |
| N1 | VSSOP | VSSOP | P | P | P | so |
| U15 | VSSOP | VSSOP | P | P | P | so |
| U3 | VSSOP | VSSOP | P | P | P | so |
| U9 | VSSOP | VSSOP | P | P | P | so |
| H11 | VSSOP | VSSOP | P | P | P | so |

**NOTE:**

1. The @BUS REQ. shows the pin state at the external bus, which is used by the other bus master.
2. ' – ' mark indicates the unchanged pin state at Bus Request mode.
3. Hi-z or Pre means Hi-z or early state and it is determined by the setting of MISCCR register.
4. AI/AO means analog input/analog output.
5. P, I, and O mean power, input and output respectively.
6. The I/O state @nRESET shows the pin status in the @nRESET duration below.



7. The table below shows I/O types and the descriptions.

| I/O Type | Descriptions |
|---|---|
| d12i(vdd12ih) | 1.2V Vdd for alive power |
| d12c(vdd12ih_core), si(vssih) | 1.2V Vdd/Vss for internal logic |
| d33o(vdd33oph), so(vssoph) | 3.3V Vdd/Vss for external logic |
| d33t(vdd33th_abb), st(vssbbh_abb) | 3.3V Vdd/Vss for analog circuitry |
| drtc(vdd30th_rtc) | 3.0V Vdd for RTC power |
| t8(phbsu100ct8sm) | Bi-directional pad, LVCMOS schmitt-trigger, 100Kohm pull-up resistor with control, tri-state, Io=8mA |
| is(phis) | Input pad, LVCMOS schmitt-trigger level |
| us(pbusb0) | USB pad |
| t10(phtot10cd) | 5V tolerant Output pad, Tri-state . |
| ot(phot8) | Output pad, tri-state, Io=8mA |
| b8(phob8) | Output pad, Io=8mA |
| t16(phot16sm) | Output pad, tri-state, medium slew rate, Io=16mA |
| r10(phiar10_abb) | Analog input pad with 10-ohm resistor |
| ia(phia_abb) | Analog input pad |
| gp(phgpad_option) | Pad for analog pin |
| m26(phsoscm26_2440) | Oscillator cell with enable and feedback resistor |
| tt8(phtbsu100ct8sm) | 5V Tolerant Bi-directional pad, LVCMOS schmitt-trigger, 100Kohm pull-up resistor with control, tri -state, medium slew rate, Io=8mA |
| t12(phbsu100ct12sm) | Bi-directional pad, LVCMOS schmitt-trigger, 100Kohm pull-up resistor with control, tri-state, Io=12mA |
| d2(phtod2) | 5v tolerant Output pad , Open Drain , Io=2mA |
| d8(phbsd8sm) | Bi-directional pad, LVCMOS schmitt-trigger, Open Drain, Io=8mA |
| t10s(phtot10cd_10_2440x) | 5V Tolerant output pad, LVCMOS , tri -state, output drive strenth control, Io=4,6,8,10mA |
| b12s(phtbsu100ct12cd_12_2440x) | 5V Tolerant Bi-directional pad, LVCMOS schmitt-trigger, 100Kohm pull-up resistor with control, tri -state,output drive strenth control, Io=6,8,10,12mA |
| d2s(phtbsd2_2440x) | 5V Tolerant Bi-directional pad, LVCMOS schmitt-trigger, open-drain, output drive strenth ignore, |
| r50(phoar50_abb) | Analog Output pad, 50Kohm resistor, Separated bulk-bias |
| t12s(phtot12cd_12_2440x) | 5V Tolerant output pad, LVCMOS , tri -state, output drive strenth control, Io=6,8,12,16mA |
| nc(phnc) | No connection pad |

## SIGNAL DESCRIPTIONS

**Table 1-3. S3C2440X Signal Descriptions (Sheet 1 of 6)**

| Signal | I/O | Descriptions | |
|---|---|---|---|
| **Bus Controller** | | | |
| OM[1:0] | I | OM[1:0] sets S3C2440X in the TEST mode, which is used only at fabrication. Also, it determines the bus width of nGCS0. The pull-up/down resistor determines the logic level during RESET cycle.<br>00:Nand-boot 01:16-bit 10:32-bit 11:Test mode | |
| ADDR[26:0] | O | ADDR[26:0] (Address Bus) outputs the memory address of the corresponding bank . | |
| DATA[31:0] | IO | DATA[31:0] (Data Bus) inputs data during memory read and outputs data during memory write. The bus width is programmable among 8/16/32-bit. | |
| nGCS[7:0] | O | nGCS[7:0] (General Chip Select) are activated when the address of a memory is within the address region of each bank. The number of access cycles and the bank size can be programmed. | |
| nWE | O | nWE (Write Enable) indicates that the current bus cycle is a write cycle. | |
| nOE | O | nOE (Output Enable) indicates that the current bus cycle is a read cycle. | |
| nXBREQ | I | nXBREQ (Bus Hold Request) allows another bus master to request control of the local bus. BACK active indicates that bus control has been granted. | |
| nXBACK | O | nXBACK (Bus Hold Acknowledge) indicates that the S3C2440X has surrendered control of the local bus to another bus master. | |
| nWAIT | I | nWAIT requests to prolong a current bus cycle. As long as nWAIT is L, the current bus cycle cannot be completed. | |
| **SDRAM/SRAM** | | | |
| nSRAS | O | SDRAM Row Address Strobe | |
| nSCAS | O | SDRAM Column Address Strobe | |
| nSCS[1:0] | O | SDRAM Chip Select | |
| DQM[3:0] | O | SDRAM Data Mask | |
| SCLK[1:0] | O | SDRAM Clock | |
| SCKE | O | SDRAM Clock Enable | |
| nBE[3:0] | O | Upper Byte/Lower Byte Enable(In case of 16-bit SRAM) | |
| nWBE[3:0] | O | Write Byte Enable | |
| **NAND Flash** | | | |
| CLE | O | Command Latch Enable | |
| ALE | O | Address Latch Enable | |
| nFCE | O | Nand Flash Chip Enable | |
| nFRE | O | Nand Flash Read Enable | |
| nFWE | O | Nand Flash Write Enable | |
| NCON | I | Nand Flash Configuration | * If NAND flash controller isn't used, it has to be pull-up. (3.3V) |
| FRnB | I | Nand Flash Ready/Busy | |

**Table 1-3. S3C2440X Signal Descriptions (Sheet 2 of 6)**

| Signal | I/O | Descriptions |
|---|---|---|
| **LCD Control Unit** | | |
| VD[23:0] | O | **STN/TFT/SEC TFT:** LCD Data Bus |
| LCD_PWREN | O | **STN/TFT/SEC TFT:** LCD panel power enable control signal |
| VCLK | O | **STN/TFT:** LCD clock signal |
| VFRAME | O | **STN:** LCD Frame signal |
| VLINE | O | **STN:** LCD line signal |
| VM | O | **STN:** VM alternates the polarity of the row and column voltage |
| VSYNC | O | **TFT:** Vertical synchronous signal |
| HSYNC | O | **TFT:** Horizontal synchronous signal |
| VDEN | O | **TFT:** Data enable signal |
| LEND | O | **TFT:** Line End signal |
| STV | O | **SEC TFT:** SEC(Samsung Electronics Company) TFT LCD panel control signal |
| CPV | O | **SEC TFT:** SEC(Samsung Electronics Company) TFT LCD panel control signal |
| LCD_HCLK | O | **SEC TFT:** SEC(Samsung Electronics Company) TFT LCD panel control signal |
| TP | O | **SEC TFT:** SEC(Samsung Electronics Company) TFT LCD panel control signal |
| STH | O | **SEC TFT:** SEC(Samsung Electronics Company) TFT LCD panel control signal |
| LCD_LPCOE | O | **SEC TFT:** Timing control signal for specific TFT LCD |
| LCD_LPCREV | O | **SEC TFT:** Timing control signal for specific TFT LCD |
| LCD_LPCREVB | O | **SEC TFT:** Timing control signal for specific TFT LCD |
| **CAMERA Interface** | | |
| CAMRESET | O | Software Reset to the Camera |
| CAMCLKOUT | O | Master Clock to the Camera |
| CAMPCLK | I | Pixel clock from Camera |
| CAMHREF | I | Horizontal sync signal from Camera |
| CAMVSYNC | I | Vertical sync signal from Camera |
| CAMDATA[7:0] | I | Pixel data for YCbCr |
| **Interrupt Control Unit** | | |
| EINT[23:0] | I | External Interrupt request |
| **DMA** | | |
| nXDREQ[1:0] | I | External DMA request |
| nXDACK[1:0] | O | External DMA acknowledge |

**Table 1-3. S3C2440X Signal Descriptions (Sheet 3 of 6)**

| Signal | I/O | Descriptions |
|---|---|---|
| **UART** | | |
| RxD[2:0] | I | UART receives data input |
| TxD[2:0] | O | UART transmits data output |
| nCTS[1:0] | I | UART clear to send input signal |
| nRTS[1:0] | O | UART request to send output signal |
| UARTCLK | I | UART clock signal |
| **ADC** | | |
| AIN[7:0] | AI | ADC input[7:0]. If it isn't used pin, it has to be Low (Ground). |
| Vref | AI | ADC Vref |
| **IIC-Bus** | | |
| IICSDA | IO | IIC-bus data |
| IICSCL | IO | IIC-bus clock |
| **IIS-Bus** | | |
| I2SLRCK | IO | IIS-bus channel select clock |
| I2SSDO | O | IIS-bus serial data output |
| I2SSDI | I | IIS-bus serial data input |
| I2SSCLK | IO | IIS-bus serial clock |
| CDCLK | O | CODEC system clock |
| **Touch Screen** | | |
| nXPON | O | Plus X-axis on-off control signal |
| XMON | O | Minus X-axis on-off control signal |
| nYPON | O | Plus Y-axis on-off control signal |
| YMON | O | Minus Y-axis on-off control signal |
| **USB Host** | | |
| DN[1:0] | IO | DATA(–) from USB host |
| DP[1:0] | IO | DATA(+) from USB host |
| **USB Device** | | |
| PDN0 | IO | DATA(–) for USB peripheral |
| PDP0 | IO | DATA(+) for USB peripheral |
| **SPI** | | |
| SPIMISO[1:0] | IO | SPIMISO is the master data input line, when SPI is configured as a master. When SPI is configured as a slave, these pins reverse its role. |
| SPIMOSI[1:0] | IO | SPIMOSI is the master data output line, when SPI is configured as a master. When SPI is configured as a slave, these pins reverse its role. |
| SPICLK[1:0] | IO | SPI clock |
| nSS[1:0] | I | SPI chip select(only for slave mode) |

**Table 1-3. S3C2440X Signal Descriptions (Sheet 4 of 6)**

| Signal | I/O | Description |
|---|---|---|
| **SD** | | |
| SDDAT[3:0] | IO | SD receive/transmit data |
| SDCMD | IO | SD receive response/ transmit command |
| SDCLK | O | SD clock |
| **General Port** | | |
| GPn[116:0] | IO | General input/output ports (some ports are output only) |
| **TIMMER/PWM** | | |
| TOUT[3:0] | O | Timer output[3:0] |
| TCLK[1:0] | I | External timer clock input |
| **JTAG TEST LOGIC** | | |
| nTRST | I | nTRST(TAP Controller Reset) resets the TAP controller at start.<br>If debugger is used, A 10K pull-up resistor has to be connected.<br>If debugger(black ICE) is not used, nTRST pin must be issued by a low active pulse(Typically connected to nRESET). |
| TMS | I | TMS (TAP Controller Mode Select) controls the sequence of the TAP controller's states. A 10K pull-up resistor has to be connected to TMS pin. |
| TCK | I | TCK (TAP Controller Clock) provides the clock input for the JTAG logic.<br>A 10K pull-up resistor must be connected to TCK pin. |
| TDI | I | TDI (TAP Controller Data Input) is the serial input for test instructions and data.<br>A 10K pull-up resistor must be connected to TDI pin. |
| TDO | O | TDO (TAP Controller Data Output) is the serial output for test instructions and data. |

**Table 1-3. S3C2440X Signal Descriptions (Sheet 5 of 6)**

| Signal | I/O | Description |
|---|---|---|
| **Reset, Clock & Power** | | |
| XTOpll | AO | Crystal Output for internal osc circuit.<br>When OM[3:2] = 00b, XTIpll is used for MPLL CLK source and UPLL CLK source.<br>When OM[3:2] = 01b, XTIpll is used for MPLL CLK source only.<br>When OM[3:2] = 10b, XTIpll is used for UPLL CLK source only.<br>If it isn't used, it has to be a floating pin. |
| MPLLCAP | AI | Loop filter capacitor for main clock. |
| UPLLCAP | AI | Loop filter capacitor for USB clock. |
| XTIrtc | AI | 32 kHz crystal input for RTC. If it isn't used, it has to be High (3.3V). |
| XTOrtc | AO | 32 kHz crystal output for RTC. If it isn't used, it has to be Float. |
| CLKOUT[1:0] | O | Clock output signal. The CLKSEL of MISCCR register configures the clock output mode among the MPLL CLK, UPLL CLK, FCLK, HCLK, PCLK. |
| nRESET | ST | nRESET suspends any operation in progress and places S3C2440X into a known reset state. For a reset, nRESET must be held to L level for at least 4 FCLK after the processor power has been stabilized. |
| nRSTOUT | O | For external device reset control(nRSTOUT = nRESET & nWDTRST & SW_RESET) |
| PWREN | O | 1.2V core power on-off control signal |
| nBATT_FLT | I | Probe for battery state(Does not wake up at Sleep mode in case of low battery state).<br>If it isn't used, it has to be High (3.3V). |
| OM[3:2] | I | OM[3:2] determines how the clock is made.<br>OM[3:2] = 00b, Crystal is used for MPLL CLK source and UPLL CLK source.<br>OM[3:2] = 01b, Crystal is used for MPLL CLK source<br>and EXTCLK is used for UPLL CLK source.<br>OM[3:2] = 10b, EXTCLK is used for MPLL CLK source<br>and Crystal is used for UPLL CLK source.<br>OM[3:2] = 11b, EXTCLK is used for MPLL CLK source and UPLL CLK source. |
| EXTCLK | I | External clock source.<br>When OM[3:2] = 11b, EXTCLK is used for MPLL CLK source and UPLL CLK source.<br>When OM[3:2] = 10b, EXTCLK is used for MPLL CLK source only.<br>When OM[3:2] = 01b, EXTCLK is used for UPLL CLK source only.<br>If it isn't used, it has to be High (3.3V). |
| XTIpll | AI | Crystal Input for internal osc circuit.<br>When OM[3:2] = 00b, XTIpll is used for MPLL CLK source and UPLL CLK source.<br>When OM[3:2] = 01b, XTIpll is used for MPLL CLK source only.<br>When OM[3:2] = 10b, XTIpll is used for UPLL CLK source only.<br>If it isn't used, XTIpll has to be High (3.3V). |

**Table 1-3. S3C2440X Signal Descriptions (Sheet 6 of 6)**

| Signal | I/O | Description |
|---|---|---|
| **Power** | | |
| VDDalive | P | S3C2440X reset block and port status register VDD(1.2V).<br>It should be always supplied whether in normal mode or in Sleep mode. |
| VDDi/VDDiarm | P | S3C2440X core logic VDD(1.2V) for CPU. |
| VSSi/VSSiarm | P | S3C2440X core logic VSS |
| VDDi_MPLL | P | S3C2440X MPLL analog and digital VDD (1.2 V). |
| VSSi_MPLL | P | S3C2440X MPLL analog and digital VSS. |
| VDDOP | P | S3C2440X I/O port VDD(3.3V) |
| VDDMOP | P | S3C2440X Memory I/O VDD<br>3.3V : SCLK up to 100MHz<br>2.5V : SCLK up to 80MHz |
| VSSOP | P | S3C2440X I/O port VSS |
| RTCVDD | P | RTC VDD (3.0V)<br>(This pin must be connected to power properly if RTC isn't used) |
| VDDi_UPLL | P | S3C2440X UPLL analog and digital VDD (1.2V) |
| VSSi_UPLL | P | S3C2440X UPLL analog and digital VSS |
| VDDA_ADC | P | S3C2440X ADC VDD(3.3V) |
| VSSA_ADC | P | S3C2440X ADC VSS |

**NOTE:**

1. I/O means input/output.
2. AI/AO means analog input/analog output.
3. ST means schmitt-trigger.
4. P means power.

## S3C2440X SPECIAL REGISTERS

**Table 1-4. S3C2440X Special Registers (Sheet 1 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **Memory Controller** | | | | | |
| BWSCON | 0x48000000 | ← | W | R/W | Bus Width & Wait Status Control |
| BANKCON0 | 0x48000004 | | | | Boot ROM Control |
| BANKCON1 | 0x48000008 | | | | BANK1 Control |
| BANKCON2 | 0x4800000C | | | | BANK2 Control |
| BANKCON3 | 0x48000010 | | | | BANK3 Control |
| BANKCON4 | 0x48000014 | | | | BANK4 Control |
| BANKCON5 | 0x48000018 | | | | BANK5 Control |
| BANKCON6 | 0x4800001C | | | | BANK6 Control |
| BANKCON7 | 0x48000020 | | | | BANK7 Control |
| REFRESH | 0x48000024 | | | | DRAM/SDRAM Refresh Control |
| BANKSIZE | 0x48000028 | | | | Flexible Bank Size |
| MRSRB6 | 0x4800002C | | | | Mode register set for SDRAM BANK6 |
| MRSRB7 | 0x48000030 | | | | Mode register set for SDRAM BANK7 |

**Table 1-4. S3C2440X Special Registers (Sheet 2 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **USB Host Controller** | | | | | |
| HcRevision | 0x49000000 | ← | W | | Control and Status Group |
| HcControl | 0x49000004 | | | | |
| HcCommonStatus | 0x49000008 | | | | |
| HcInterruptStatus | 0x4900000C | | | | |
| HcInterruptEnable | 0x49000010 | | | | |
| HcInterruptDisable | 0x49000014 | | | | |
| HcHCCA | 0x49000018 | | | | Memory Pointer Group |
| HcPeriodCuttentED | 0x4900001C | | | | |
| HcControlHeadED | 0x49000020 | | | | |
| HcControlCurrentED | 0x49000024 | | | | |
| HcBulkHeadED | 0x49000028 | | | | |
| HcBulkCurrentED | 0x4900002C | | | | |
| HcDoneHead | 0x49000030 | | | | |
| HcRmInterval | 0x49000034 | | | | Frame Counter Group |
| HcFmRemaining | 0x49000038 | | | | |
| HcFmNumber | 0x4900003C | | | | |
| HcPeriodicStart | 0x49000040 | | | | |
| HcLSThreshold | 0x49000044 | | | | |
| HcRhDescriptorA | 0x49000048 | | | | Root Hub Group |
| HcRhDescriptorB | 0x4900004C | | | | |
| HcRhStatus | 0x49000050 | | | | |
| HcRhPortStatus1 | 0x49000054 | | | | |
| HcRhPortStatus2 | 0x49000058 | | | | |
| **Interrupt Controller** | | | | | |
| SRCPND | 0X4A000000 | ← | W | R/W | Interrupt Request Status |
| INTMOD | 0X4A000004 | | | W | Interrupt Mode Control |
| INTMSK | 0X4A000008 | | | R/W | Interrupt Mask Control |
| PRIORITY | 0X4A00000C | | | W | IRQ Priority Control |
| INTPND | 0X4A000010 | | | R/W | Interrupt Request Status |
| INTOFFSET | 0X4A000014 | | | R | Interrupt request source offset |
| SUBSRCPND | 0X4A000018 | | | R/W | Sub source pending |
| INTSUBMSK | 0X4A00001C | | | R/W | Interrupt sub mask |

**Table 1-4. S3C2440X Special Registers (Sheet 3 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **DMA** | | | | | |
| DISRC0 | 0x4B000000 | ← | W | R/W | DMA 0 Initial Source |
| DISRCC0 | 0x4B000004 | | | | DMA 0 Initial Source Control |
| DIDST0 | 0x4B000008 | | | | DMA 0 Initial Destination |
| DIDSTC0 | 0x4B00000C | | | | DMA 0 Initial Destination Control |
| DCON0 | 0x4B000010 | | | | DMA 0 Control |
| DSTAT0 | 0x4B000014 | | | R | DMA 0 Count |
| DCSRC0 | 0x4B000018 | | | | DMA 0 Current Source |
| DCDST0 | 0x4B00001C | | | | DMA 0 Current Destination |
| DMASKTRIG0 | 0x4B000020 | | | R/W | DMA 0 Mask Trigger |
| DISRC1 | 0x4B000040 | | | | DMA 1 Initial Source |
| DISRCC1 | 0x4B000044 | | | | DMA 1 Initial Source Control |
| DIDST1 | 0x4B000048 | | | | DMA 1 Initial Destination |
| DIDSTC1 | 0x4B00004C | | | | DMA 1 Initial Destination Control |
| DCON1 | 0x4B000050 | | | | DMA 1 Control |
| DSTAT1 | 0x4B000054 | | | R | DMA 1 Count |
| DCSRC1 | 0x4B000058 | | | | DMA 1 Current Source |
| DCDST1 | 0x4B00005C | | | | DMA 1 Current Destination |
| DMASKTRIG1 | 0x4B000060 | | | R/W | DMA 1 Mask Trigger |
| DISRC2 | 0x4B000080 | | | | DMA 2 Initial Source |
| DISRCC2 | 0x4B000084 | | | | DMA 2 Initial Source Control |
| DIDST2 | 0x4B000088 | | | | DMA 2 Initial Destination |
| DIDSTC2 | 0x4B00008C | | | | DMA 2 Initial Destination Control |
| DCON2 | 0x4B000090 | | | | DMA 2 Control |
| DSTAT2 | 0x4B000094 | | | R | DMA 2 Count |
| DCSRC2 | 0x4B000098 | | | | DMA 2 Current Source |
| DCDST2 | 0x4B00009C | | | | DMA 2 Current Destination |
| DMASKTRIG2 | 0x4B0000A0 | | | R/W | DMA 2 Mask Trigger |
| DISRC3 | 0x4B0000C0 | ← | W | R/W | DMA 3 Initial Source |
| DISRCC3 | 0x4B0000C4 | | | | DMA 3 Initial Source Control |
| DIDST3 | 0x4B0000C8 | | | | DMA 3 Initial Destination |
| DIDSTC3 | 0x4B0000CC | | | | DMA 3 Initial Destination Control |
| DCON3 | 0x4B0000D0 | | | | DMA 3 Control |
| DSTAT3 | 0x4B0000D4 | | | R | DMA 3 Count |
| DCSRC3 | 0x4B0000D8 | | | | DMA 3 Current Source |
| DCDST3 | 0x4B0000DC | | | | DMA 3 Current Destination |
| DMASKTRIG3 | 0x4B0000E0 | | | R/W | DMA 3 Mask Trigger |

**Table 1-4. S3C2440X Special Registers (Sheet 4 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **Clock & Power Management** | | | | | |
| LOCKTIME | 0x4C000000 | ← | W | R/W | PLL Lock Time Counter |
| MPLLCON | 0x4C000004 | | | | MPLL Control |
| UPLLCON | 0x4C000008 | | | | UPLL Control |
| CLKCON | 0x4C00000C | | | | Clock Generator Control |
| CLKSLOW | 0x4C000010 | | | | Slow Clock Control |
| CLKDIVN | 0x4C000014 | | | | Clock divider Control |
| CAMDIVN | 0x4C000018 | | | | Camera Clock divider Control |
| **LCD Controller** | | | | | |
| LCDCON1 | 0X4D000000 | ← | W | R/W | LCD Control 1 |
| LCDCON2 | 0X4D000004 | | | | LCD Control 2 |
| LCDCON3 | 0X4D000008 | | | | LCD Control 3 |
| LCDCON4 | 0X4D00000C | | | | LCD Control 4 |
| LCDCON5 | 0X4D000010 | | | | LCD Control 5 |
| LCDSADDR1 | 0X4D000014 | | | | STN/TFT: Frame Buffer Start Address1 |
| LCDSADDR2 | 0X4D000018 | | | | STN/TFT: Frame Buffer Start Address2 |
| LCDSADDR3 | 0X4D00001C | | | | STN/TFT: Virtual Screen Address Set |
| REDLUT | 0X4D000020 | | | | STN: Red Lookup Table |
| GREENLUT | 0X4D000024 | | | | STN: Green Lookup Table |
| BLUELUT | 0X4D000028 | | | | STN: Blue Lookup Table |
| DITHMODE | 0X4D00004C | | | | STN: Dithering Mode |
| TPAL | 0X4D000050 | | | | TFT: Temporary Palette |
| LCDINTPND | 0X4D000054 | | | | LCD Interrupt Pending |
| LCDSRCPND | 0X4D000058 | | | | LCD Interrupt Source |
| LCDINTMSK | 0X4D00005C | | | | LCD Interrupt Mask |
| TCONSEL | 0X4D000060 | | | | TCON(LPC3600/LCC3600) Control |

**Table 1-4. S3C2440X Special Registers (Sheet 5 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **NAND Flash** | | | | | |
| NFCONF | 0x4E000000 | ← | W | R/W | NAND Flash Configuration |
| NFCONT | 0x4E000004 | | | | NAND Flash Control |
| NFCMD | 0x4E000008 | | | | NAND Flash Command |
| NFADDR | 0x4E00000C | | | | NAND Flash Address |
| NFDATA | 0x4E000010 | | | | NAND Flash Data |
| NFMECC0 | 0x4E000014 | | | | NAND Flash Main area ECC0/1 |
| NFMECC1 | 0x4E000018 | | | | NAND Flash Main area ECC2/3 |
| NFSECC | 0x4E00001C | | | | NAND Flash Spare area ECC |
| NFSTAT | 0x4E000020 | | | | NAND Flash Operation Status |
| NFESTAT0 | 0x4E000024 | | | | NAND Flash ECC Status for I/O[7:0] |
| NFESTAT1 | 0x4E000028 | | | | NAND Flash ECC Status for I/O[15:8] |
| NFMECC0 | 0x4E00002C | | | R | NAND Flash Main area ECC0 status |
| NFMECC1 | 0x4E000030 | | | | NAND Flash Main Area ECC1 status |
| NFSECC | 0x4E000034 | | | | NAND Flash Spare Area ECC status |
| NFSBLK | 0x4E000038 | | | R/W | NAND Flash start block address |
| NFEBLK | 0x4E00003C | | | | NAND Flash end block address |

**Table 1-4. S3C2440X Special Registers (Sheet 6 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **Camera Interface** | | | | | |
| ASIZE | 0x4F000000 | ← | W | W | A-port Image Size |
| STAY1 | 0x4F000004 | | | R/W | Y start address for 1st ping-pong memory of A-port Image |
| STAY2 | 0x4F000008 | | | | Y start address for 2nd ping-pong memory of A-port Image |
| STAY3 | 0x4F00000C | | | | Y start address for 3rd ping-pong memory of A-port Image |
| STAY4 | 0x4F000010 | | | | Y start address for 4th ping-pong memory of A-port Image |
| AYBURST | 0x4F000014 | | | W | A-port Image Y data burst length |
| ACBBURST | 0x4F000018 | | | | A-port Image Cb data burst length |
| ACRBURST | 0x4F00001C | | | | A-port Image Cr data burst length |
| BSIZE | 0x4F000020 | | | | B-port Image Size |
| STBY1 | 0x4F000024 | | | | Y start address for 1st ping-pong memory of B-port Image |
| STBY2 | 0x4F000028 | | | | Y start address for 2nd ping-pong memory of B-port Image |
| STBY3 | 0x4F00002C | | | | Y start address for 3rd ping-pong memory of B-port Image |
| STBY4 | 0x4F000030 | | | | Y start address for 4th ping-pong memory of B-port Image |
| BYBURST | 0x4F000034 | | | | B-port Image Y data burst length |
| BCBBURST | 0x4F000038 | | | | B-port Image Cb data burst length |
| BCRBURST | 0x4F00003C | | | | B-port Image Cr data burst length |
| ADISTWIDTH | 0x4F000040 | | | | A Last HREF Distance Width |
| BDISTWIDTH | 0x4F000044 | | | | B Last HREF Distance Width |
| YRATIO | 0x4F00004C | | | | Y Scale Ratio |
| CRATIO | 0x4F000050 | | | | C Scale Ratio |
| YORIGINAL | 0x4F000054 | | | | Y Original Size |
| CORIGINAL | 0x4F00005C | | | | C Original Size |
| STACB1 | 0x4F000074 | | | | A Cb 1 Start Address |
| STACB2 | 0x4F000078 | | | | A Cb 2 Start Address |
| STACB3 | 0x4F00007C | | | | A Cb 3 Start Address |
| STACB4 | 0x4F000080 | | | | A Cb 4 Start Address |

**Table 1-4. S3C2440X Special Registers (Sheet 7 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **Camera Interface(Continued)** | | | | | |
| STACR1 | 0x4F000084 | ← | W | W | A Cr 1 Start Address |
| STACR2 | 0x4F000088 | | | | A Cr 2 Start Address |
| STACR3 | 0x4F00008C | | | | A Cr 3 Start Address |
| STACR4 | 0x4F000090 | | | | A Cr 4 Start Address |
| STBCB1 | 0x4F00009C | | | | B Cb 1 Start Address |
| STBCB2 | 0x4F0000A0 | | | | B Cb 2 Start Address |
| STBCB3 | 0x4F0000A4 | | | | B Cb 3 Start Address |
| STBCB4 | 0x4F0000A8 | | | | B Cb 4 Start Address |
| STBCR1 | 0x4F0000AC | | | | B Cr 1 Start Address |
| STBCR2 | 0x4F0000B0 | | | | B Cr 2 Start Address |
| STBCR3 | 0x4F0000B4 | | | | B Cr 3 Start Address |
| STBCR4 | 0x4F0000B8 | | | | B Cr 4 Start Address |
| CTRL | 0x4F0000BC | | | | Control Register |
| RDSTAT | 0x4F000000 | | | R | Status Read Register |
| RDSTAY | 0x4F000014 | | | | A Y Start Address Read |
| RDSTACB | 0x4F000018 | | | | A Cb Start Address Read |
| RDSTACR | 0x4F00001C | | | | A Cr Start Address Read |
| RDSTACB1 | 0x4F000020 | | | | A Cb1 Start Address Read |
| RDSTACR1 | 0x4F000024 | | | | A Cr1 Start Address Read |
| RDSTBY1 | 0x4F000028 | | | | B Y1 Start Address Read |
| RDSTBY2 | 0x4F00002C | | | | B Y2 Start Address Read |
| RDSTBY3 | 0x4F000030 | | | | B Y3 Start Address Read |
| RDSTBY4 | 0x4F000034 | | | | B Y4 Start Address Read |
| RDSTBY | 0x4F000038 | | | | B Y Start Address Read |
| RDSTBCB | 0x4F00003C | | | | B Cb Start Address Read |
| RDSTBCR | 0x4F000040 | | | | B Cr Start Address Read |
| RDSTBCB1 | 0x4F000044 | | | | B Cb1 Start Address Read |
| RDSTBCR1 | 0x4F000048 | | | | B Cr1 Start Address Read |
| RDADISTWIDTH | 0x4F00004C | | | | A Last HREF Distance Width |
| RDBDISTWIDTH | 0x4F000050 | | | | B Last HREF Distance Width |

**Table 1-4. S3C2440X Special Registers (Sheet 8 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **UART** | | | | | |
| ULCON0 | 0x50000000 | ← | W | R/W | UART 0 Line Control |
| UCON0 | 0x50000004 | | | | UART 0 Control |
| UFCON0 | 0x50000008 | | | | UART 0 FIFO Control |
| UMCON0 | 0x5000000C | | | | UART 0 Modem Control |
| UTRSTAT0 | 0x50000010 | | | R | UART 0 Tx/Rx Status |
| UERSTAT0 | 0x50000014 | | | | UART 0 Rx Error Status |
| UFSTAT0 | 0x50000018 | | | | UART 0 FIFO Status |
| UMSTAT0 | 0x5000001C | | | | UART 0 Modem Status |
| UTXH0 | 0x50000023 | 0x50000020 | B | W | UART 0 Transmission Hold |
| URXH0 | 0x50000027 | 0x50000024 | | R | UART 0 Receive Buffer |
| UBRDIV0 | 0x50000028 | ← | W | R/W | UART 0 Baud Rate Divisor |
| ULCON1 | 0x50004000 | | | | UART 1 Line Control |
| UCON1 | 0x50004004 | | | | UART 1 Control |
| UFCON1 | 0x50004008 | | | | UART 1 FIFO Control |
| UMCON1 | 0x5000400C | | | | UART 1 Modem Control |
| UTRSTAT1 | 0x50004010 | | | R | UART 1 Tx/Rx Status |
| UERSTAT1 | 0x50004014 | | | | UART 1 Rx Error Status |
| UFSTAT1 | 0x50004018 | | | | UART 1 FIFO Status |
| UMSTAT1 | 0x5000401C | | | | UART 1 Modem Status |
| UTXH1 | 0x50004023 | 0x50004020 | B | W | UART 1 Transmission Hold |
| URXH1 | 0x50004027 | 0x50004024 | | R | UART 1 Receive Buffer |
| UBRDIV1 | 0x50004028 | ← | W | R/W | UART 1 Baud Rate Divisor |
| ULCON2 | 0x50008000 | | | | UART 2 Line Control |
| UCON2 | 0x50008004 | | | | UART 2 Control |
| UFCON2 | 0x50008008 | | | | UART 2 FIFO Control |
| UTRSTAT2 | 0x50008010 | | | R | UART 2 Tx/Rx Status |
| UERSTAT2 | 0x50008014 | | | | UART 2 Rx Error Status |
| UFSTAT2 | 0x50008018 | | | | UART 2 FIFO Status |
| UTXH2 | 0x50008023 | 0x50008020 | B | W | UART 2 Transmission Hold |
| URXH2 | 0x50008027 | 0x50008024 | | R | UART 2 Receive Buffer |
| UBRDIV2 | 0x50008028 | ← | W | R/W | UART 2 Baud Rate Divisor |

**Table 1-4. S3C2440X Special Registers (Sheet 9 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **PWM Timer** | | | | | |
| TCFG0 | 0x51000000 | ← | W | R/W | Timer Configuration |
| TCFG1 | 0x51000004 | | | | Timer Configuration |
| TCON | 0x51000008 | | | | Timer Control |
| TCNTB0 | 0x5100000C | | | | Timer Count Buffer 0 |
| TCMPB0 | 0x51000010 | | | | Timer Compare Buffer 0 |
| TCNTO0 | 0x51000014 | | | R | Timer Count Observation 0 |
| TCNTB1 | 0x51000018 | | | R/W | Timer Count Buffer 1 |
| TCMPB1 | 0x5100001C | | | | Timer Compare Buffer 1 |
| TCNTO1 | 0x51000020 | | | R | Timer Count Observation 1 |
| TCNTB2 | 0x51000024 | | | R/W | Timer Count Buffer 2 |
| TCMPB2 | 0x51000028 | | | | Timer Compare Buffer 2 |
| TCNTO2 | 0x5100002C | | | R | Timer Count Observation 2 |
| TCNTB3 | 0x51000030 | | | R/W | Timer Count Buffer 3 |
| TCMPB3 | 0x51000034 | | | | Timer Compare Buffer 3 |
| TCNTO3 | 0x51000038 | | | R | Timer Count Observation 3 |
| TCNTB4 | 0x5100003C | | | R/W | Timer Count Buffer 4 |
| TCNTO4 | 0x51000040 | | | R | Timer Count Observation 4 |

**Table 1-4. S3C2440X Special Registers (Sheet 10 of 14))**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/Write | Function |
|---|---|---|---|---|---|
| **USB Device** | | | | | |
| FUNC_ADDR_REG | 0x52000143 | 0x52000140 | B | R/W | Function Address |
| PWR_REG | 0x52000147 | 0x52000144 | | | Power Management |
| EP_INT_REG | 0x5200014B | 0x52000148 | | | EP Interrupt Pending and Clear |
| USB_INT_REG | 0x5200015B | 0x52000158 | | | USB Interrupt Pending and Clear |
| EP_INT_EN_REG | 0x5200015F | 0x5200015C | | | Interrupt Enable |
| USB_INT_EN_REG | 0x5200016F | 0x5200016C | | | Interrupt Enable |
| FRAME_NUM1_REG | 0x52000173 | 0x52000170 | | R | Frame Number Lower Byte |
| FRAME_NUM2_REG | 0x52000177 | 0x52000174 | | | Frame Number Higher Byte |
| INDEX_REG | 0x5200017B | 0x52000178 | | R/W | Register Index |
| EP0_CSR | 0x52000187 | 0x52000184 | | | Endpoint 0 Status |
| IN_CSR1_REG | 0x52000187 | 0x52000184 | | | In Endpoint Control Status |
| IN_CSR2_REG | 0x5200018B | 0x52000188 | | | In Endpoint Control Status |
| MAXP_REG | 0x52000183 | 0x52000180 | | | Endpoint Max Packet |
| OUT_CSR1_REG | 0x52000193 | 0x52000190 | | | Out Endpoint Control Status |
| OUT_CSR2_REG | 0x52000197 | 0x52000194 | | | Out Endpoint Control Status |
| OUT_FIFO_CNT1_REG | 0x5200019B | 0x52000198 | | R | Endpoint Out Write Count |
| OUT_FIFO_CNT2_REG | 0x5200019F | 0x5200019C | | | Endpoint Out Write Count |
| EP0_FIFO | 0x520001C3 | 0x520001C0 | | R/W | Endpoint 0 FIFO |
| EP1_FIFO | 0x520001C7 | 0x520001C4 | | | Endpoint 1 FIFO |
| EP2_FIFO | 0x520001CB | 0x520001C8 | | | Endpoint 2 FIFO |
| EP3_FIFO | 0x520001CF | 0x520001CC | | | Endpoint 3 FIFO |
| EP4_FIFO | 0x520001D3 | 0x520001D0 | | | Endpoint 4 FIFO |
| EP1_DMA_CON | 0x52000203 | 0x52000200 | | | EP1 DMA Interface Control |
| EP1_DMA_UNIT | 0x52000207 | 0x52000204 | | | EP1 DMA Tx Unit Counter |
| EP1_DMA_FIFO | 0x5200020B | 0x52000208 | | | EP1 DMA Tx FIFO Counter |
| EP1_DMA_TTC_L | 0x5200020F | 0x5200020C | | | EP1 DMA Total Tx Counter |
| EP1_DMA_TTC_M | 0x52000213 | 0x52000210 | | | EP1 DMA Total Tx Counter |
| EP1_DMA_TTC_H | 0x52000217 | 0x52000214 | | | EP1 DMA Total Tx Counter |

**Table 1-4. S3C2440X Special Registers (Sheet 11 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/Write | Function |
|---|---|---|---|---|---|
| **USB Device (Continued)** | | | | | |
| EP2_DMA_CON | 0x5200021B | 0x52000218 | B | R/W | EP2 DMA Interface Control |
| EP2_DMA_UNIT | 0x5200021F | 0x5200021C | | | EP2 DMA Tx Unit Counter |
| EP2_DMA_FIFO | 0x52000223 | 0x52000220 | | | EP2 DMA Tx FIFO Counter |
| EP2_DMA_TTC_L | 0x52000227 | 0x52000224 | | | EP2 DMA Total Tx Counter |
| EP2_DMA_TTC_M | 0x5200022B | 0x52000228 | | | EP2 DMA Total Tx Counter |
| EP2_DMA_TTC_H | 0x5200022F | 0x5200022C | | | EP2 DMA Total Tx Counter |
| EP3_DMA_CON | 0x52000243 | 0x52000240 | | | EP3 DMA Interface Control |
| EP3_DMA_UNIT | 0x52000247 | 0x52000244 | | | EP3 DMA Tx Unit Counter |
| EP3_DMA_FIFO | 0x5200024B | 0x52000248 | | | EP3 DMA Tx FIFO Counter |
| EP3_DMA_TTC_L | 0x5200024F | 0x5200024C | | | EP3 DMA Total Tx Counter |
| EP3_DMA_TTC_M | 0x52000253 | 0x52000250 | | | EP3 DMA Total Tx Counter |
| EP3_DMA_TTC_H | 0x52000257 | 0x52000254 | | | EP3 DMA Total Tx Counter |
| EP4_DMA_CON | 0x5200025B | 0x52000258 | | | EP4 DMA Interface Control |
| EP4_DMA_UNIT | 0x5200025F | 0x5200025C | | | EP4 DMA Tx Unit Counter |
| EP4_DMA_FIFO | 0x52000263 | 0x52000260 | | | EP4 DMA Tx FIFO Counter |
| EP4_DMA_TTC_L | 0x52000267 | 0x52000264 | | | EP4 DMA Total Tx Counter |
| EP4_DMA_TTC_M | 0x5200026B | 0x52000268 | | | EP4 DMA Total Tx Counter |
| EP4_DMA_TTC_H | 0x5200026F | 0x5200026C | | | EP4 DMA Total Tx Counter |
| **Watchdog Timer** | | | | | |
| WTCON | 0x53000000 | ← | W | R/W | Watchdog Timer Mode |
| WTDAT | 0x53000004 | | | | Watchdog Timer Data |
| WTCNT | 0x53000008 | | | | Watchdog Timer Count |
| **IIC** | | | | | |
| IICCON | 0x54000000 | ← | W | R/W | IIC Control |
| IICSTAT | 0x54000004 | | | | IIC Status |
| IICADD | 0x54000008 | | | | IIC Address |
| IICDS | 0x5400000C | | | | IIC Data Shift |
| IICLC | 0x54000010 | | | | IIC multi-master line control |
| **IIS** | | | | | |
| IISCON | 0x55000000,02 | 0x55000000 | HW,W | R/W | IIS Control |
| IISMOD | 0x55000004,06 | 0x55000004 | | | IIS Mode |
| IISPSR | 0x55000008,0A | 0x55000008 | | | IIS Prescaler |
| IISFCON | 0x5500000C,0E | 0x5500000C | | | IIS FIFO Control |
| IISFIFO | 0x55000012 | 0x55000010 | HW | | IIS FIFO Entry |

**Table 1-4. S3C2440X Special Registers (Sheet 12 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **I/O port** | | | | | |
| GPACON | 0x56000000 | ← | W | R/W | Port A Control |
| GPADAT | 0x56000004 | | | | Port A Data |
| GPBCON | 0x56000010 | | | | Port B Control |
| GPBDAT | 0x56000014 | | | | Port B Data |
| GPBUP | 0x56000018 | | | | Pull-up Control B |
| GPCCON | 0x56000020 | | | | Port C Control |
| GPCDAT | 0x56000024 | | | | Port C Data |
| GPCUP | 0x56000028 | | | | Pull-up Control C |
| GPDCON | 0x56000030 | | | | Port D Control |
| GPDDA1T | 0x56000034 | | | | Port D Data |
| GPDUP | 0x56000038 | | | | Pull-up Control D |
| GPECON | 0x56000040 | | | | Port E Control |
| GPEDAT | 0x56000044 | | | | Port E Data |
| GPEUP | 0x56000048 | | | | Pull-up Control E |
| GPFCON | 0x56000050 | | | | Port F Control |
| GPFDAT | 0x56000054 | | | | Port F Data |
| GPFUP | 0x56000058 | | | | Pull-up Control F |
| GPGCON | 0x56000060 | | | | Port G Control |
| GPGDAT | 0x56000064 | | | | Port G Data |
| GPGUP | 0x56000068 | | | | Pull-up Control G |
| GPHCON | 0x56000070 | | | | Port H Control |
| GPHDAT | 0x56000074 | | | | Port H Data |
| GPHUP | 0x56000078 | | | | Pull-up Control H |
| GPJCON | 0x560000D0 | | | | Port J Control |
| GPJDAT | 0x560000D4 | | | | Port J Data |
| GPJUP | 0x560000D8 | | | | Pull-up Control J |
| MISCCR | 0x56000080 | | | | Miscellaneous Control |
| DCLKCON | 0x56000084 | | | | DCLK0/1 Control |
| EXTINT0 | 0x56000088 | | | | External Interrupt Control Register 0 |
| EXTINT1 | 0x5600008C | | | | External Interrupt Control Register 1 |
| EXTINT2 | 0x56000090 | | | | External Interrupt Control Register 2 |

**Table 1-4. S3C2440X Special Registers (Sheet 13 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **I/O port (Continued)** | | | | | |
| EINTFLT0 | 0x56000094 | ← | W | R/W | Reserved |
| EINTFLT1 | 0x56000098 | | | | Reserved |
| EINTFLT2 | 0x5600009C | | | | External Interrupt Filter Control Register 2 |
| EINTFLT3 | 0x560000A0 | | | | External Interrupt Filter Control Register 3 |
| EINTMASK | 0x560000A4 | | | | External Interrupt Mask |
| EINTPEND | 0x560000A8 | | | | External Interrupt Pending |
| GSTATUS0 | 0x560000AC | | | *R* | External Pin Status |
| GSTATUS1 | 0x560000B0 | | | R/W | Chip ID |
| GSTATUS2 | 0x560000B4 | | | | Reset Status |
| GSTATUS3 | 0x560000B8 | | | | Inform Register |
| GSTATUS4 | 0x560000BC | | | | Inform Register |
| MSLCON | 0x560000CC | | | | Memory Sleep Control Register |
| **RTC** | | | | | |
| RTCCON | 0x57000043 | 0x57000040 | B | R/W | RTC Control |
| TICNT | 0x57000047 | 0x57000044 | | | Tick time count |
| RTCALM | 0x57000053 | 0x57000050 | | | RTC Alarm Control |
| ALMSEC | 0x57000057 | 0x57000054 | | | Alarm Second |
| ALMMIN | 0x5700005B | 0x57000058 | | | Alarm Minute |
| ALMHOUR | 0x5700005F | 0x5700005C | | | Alarm Hour |
| ALMDATE | 0x57000063 | 0x57000060 | | | Alarm Day |
| ALMMON | 0x57000067 | 0x57000064 | | | Alarm Month |
| ALMYEAR | 0x5700006B | 0x57000068 | | | Alarm Year |
| RTCRST | 0x5700006F | 0x5700006C | | | RTC Round Reset |
| BCDSEC | 0x57000073 | 0x57000070 | | | BCD Second |
| BCDMIN | 0x57000077 | 0x57000074 | | | BCD Minute |
| BCDHOUR | 0x5700007B | 0x57000078 | | | BCD Hour |
| BCDDATE | 0x5700007F | 0x5700007C | | | BCD Day |
| BCDDAY | 0x57000083 | 0x57000080 | | | BCD Date |
| BCDMON | 0x57000087 | 0x57000084 | | | BCD Month |
| BCDYEAR | 0x5700008B | 0x57000088 | | | BCD Year |

**Table 1-4. S3C2440X Special Registers (Sheet 14 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **A/D converter** | | | | | |
| ADCCON | 0x58000000 | ← | W | R/W | ADC Control |
| ADCTSC | 0x58000004 | | | | ADC Touch Screen Control |
| ADCDLY | 0x58000008 | | | | ADC Start or Interval Delay |
| ADCDAT0 | 0x5800000C | | | R | ADC Conversion Data |
| ADCDAT1 | 0x58000010 | | | | ADC Conversion Data |
| ADCUPDN | 0x58000014 | | | R/W | Stylus Up or Down Interrpt status |
| **SPI** | | | | | |
| SPCON0,1 | 0x59000000,20 | ← | W | R/W | SPI Control |
| SPSTA0,1 | 0x59000004,24 | | | R | SPI Status |
| SPPIN0,1 | 0x59000008,28 | | | R/W | SPI Pin Control |
| SPPRE0,1 | 0x5900000C,2C | | | | SPI Baud Rate Prescaler |
| SPTDAT0,1 | 0x59000010,30 | | | | SPI Tx Data |
| SPRDAT0,1 | 0x59000014,34 | | | R | SPI Rx Data |
| **SD interface** | | | | | |
| SDICON | 0x5A000000 | ← | W | R/W | SDI Control |
| SDIPRE | 0x5A000004 | | | | SDI Baud Rate Prescaler |
| SDICARG | 0x5A000008 | | | | SDI Command Argument |
| SDICCON | 0x5A00000C | | | | SDI Command Control |
| SDICSTA | 0x5A000010 | | | R/(C) | SDI Command Status |
| SDIRSP0 | 0x5A000014 | | | R | SDI Response |
| SDIRSP1 | 0x5A000018 | | | | SDI Response |
| SDIRSP2 | 0x5A00001C | | | | SDI Response |
| SDIRSP3 | 0x5A000020 | | | | SDI Response |
| SDIDTIMER | 0x5A000024 | | | R/W | SDI Data / Busy Timer |
| SDIBSIZE | 0x5A000028 | | | | SDI Block Size |
| SDIDCON | 0x5A00002C | | | | SDI Data control |
| SDIDCNT | 0x5A000030 | | | R | SDI Data Remain Counter |
| SDIDSTA | 0x5A000034 | | | R/(C) | SDI Data Status |
| SDIFSTA | 0x5A000038 | | | R | SDI FIFO Status |
| SDIDAT | 0x5A00003F | 0x5A00003C | B | R/W | SDI Data |
| SDIIMSK | 0x5A000040 | ← | W | | SDI Interrupt Mask |

**Table 1-4. S3C2440X Special Registers (Sheet of 14 of 14)**

| Register Name | Address (B. Endian) | Address (L. Endian) | Acc. Unit | Read/ Write | Function |
|---|---|---|---|---|---|
| **A/D converter** | | | | | |
| ADCCON | 0x58000000 | ← | W | R/W | ADC Control |
| ADCTSC | 0x58000004 | | | | ADC Touch Screen Control |
| ADCDLY | 0x58000008 | | | | ADC Start or Interval Delay |
| ADCDAT0 | 0x5800000C | | | R | ADC Conversion Data |
| ADCDAT1 | 0x58000010 | | | | ADC Conversion Data |
| ADCUPDN | 0x58000014 | | | R/W | Stylus Up or Down Interrpt status |
| **SPI** | | | | | |
| SPCON0,1 | 0x59000000,20 | ← | W | R/W | SPI Control |
| SPSTA0,1 | 0x59000004,24 | | | R | SPI Status |
| SPPIN0,1 | 0x59000008,28 | | | R/W | SPI Pin Control |
| SPPRE0,1 | 0x5900000C,2C | | | | SPI Baud Rate Prescaler |
| SPTDAT0,1 | 0x59000010,30 | | | | SPI Tx Data |
| SPRDAT0,1 | 0x59000014,34 | | | R | SPI Rx Data |
| **SD interface** | | | | | |
| SDICON | 0x5A000000 | ← | W | R/W | SDI Control |
| SDIPRE | 0x5A000004 | | | | SDI Baud Rate Prescaler |
| SDICARG | 0x5A000008 | | | | SDI Command Argument |
| SDICCON | 0x5A00000C | | | | SDI Command Control |
| SDICSTA | 0x5A000010 | | | R/(C) | SDI Command Status |
| SDIRSP0 | 0x5A000014 | | | R | SDI Response |
| SDIRSP1 | 0x5A000018 | | | | SDI Response |
| SDIRSP2 | 0x5A00001C | | | | SDI Response |
| SDIRSP3 | 0x5A000020 | | | | SDI Response |
| SDIDTIMER | 0x5A000024 | | | R/W | SDI Data / Busy Timer |
| SDIBSIZE | 0x5A000028 | | | | SDI Block Size |
| SDIDCON | 0x5A00002C | | | | SDI Data control |
| SDIDCNT | 0x5A000030 | | | R | SDI Data Remain Counter |
| SDIDSTA | 0x5A000034 | | | R/(C) | SDI Data Status |
| SDIFSTA | 0x5A000038 | | | R | SDI FIFO Status |
| SDIDAT | 0x5A00003F | 0x5A00003C | B | R/W | SDI Data |
| SDIIMSK | 0x5A000040 | ← | W | | SDI Interrupt Mask |

# 5 MEMORY CONTROLLER

## OVERVIEW

The S3C2440X memory controller provides memory control signals that are required for external memory access.

The S3C2440X has the following features:

— Little/Big endian (selectable by a software)

— Address space: 128Mbytes per bank (total 1GB/8 banks)

— Programmable access size (8/16/32-bit) for all banks except bank0 (16/32-bit)

— Total 8 memory banks
Six memory banks for ROM, SRAM, etc.
Remaining two memory banks for ROM, SRAM, SDRAM, etc .

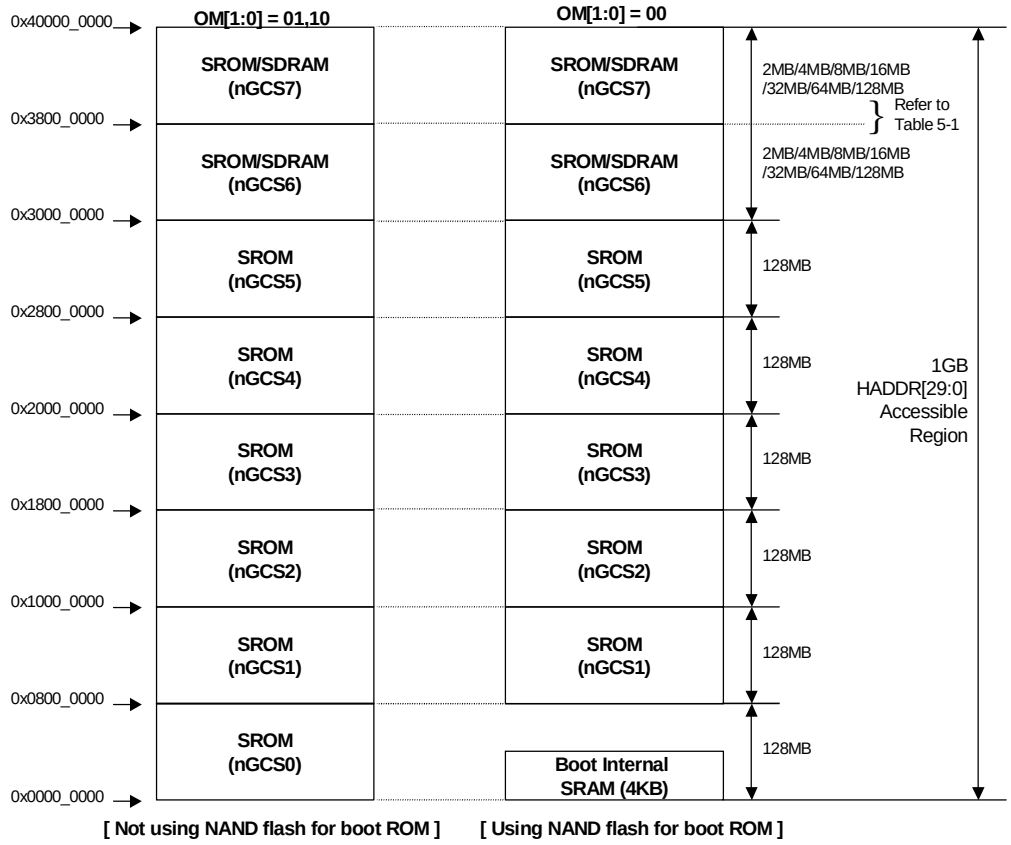
— Seven fixed memory bank start address

— One flexible memory bank start address and programmable bank size

— Programmable access cycles for all memory banks

— External wait to extend the bus cycles

— Supporting self-refresh and power down mode in SDRAM




Note: SROM means ROM or SRAM type memory

**Figure 5-1. S3C2440X Memory Map after Reset**

**Table 5-1. Bank 6/7 Addresses**

| Address | 2MB | 4MB | 8MB | 16MB | 32MB | 64MB | 128MB |
|---|---|---|---|---|---|---|---|
| Bank 6 | | | | | | | |
| Start address | 0x3000_0000 | 0x3000_0000 | 0x3000_0000 | 0x3000_0000 | 0x3000_0000 | 0x3000_0000 | 0x3000_0000 |
| End address | 0x301F_FFFF | 0X303F_FFFF | 0X307F_FFFF | 0X30FF_FFFF | 0X31FF_FFFF | 0X33FF_FFFF | 0X37FF_FFFF |
| Bank 7 | | | | | | | |
| Start address | 0x3020_0000 | 0x3040_0000 | 0x3080_0000 | 0x3100_0000 | 0x3200_0000 | 0x3400_0000 | 0x3800_0000 |
| End address | 0X303F_FFFF | 0X307F_FFFF | 0X30FF_FFFF | 0X31FF_FFFF | 0X33FF_FFFF | 0X37FF_FFFF | 0X3FFF_FFFF |

**Note:** Bank 6 and 7 must have the same memory size.

# FUNCTION DESCRIPTION

### BANK0 BUS WIDTH

The data bus of BANK0 (nGCS0) should be configured in width as one of 16-bit and 32-bit ones. Because the BANK0 works as the booting ROM bank (map to 0x0000_0000), the bus width of BANK0 should be determined before the first ROM access, which will depend on the logic level of OM[1:0] at Reset.

| OM1 (Operating Mode 1) | OM0 (Operating Mode 0) | Booting ROM Data width |
|---|---|---|
| 0 | 0 | Nand Flash Mode |
| 0 | 1 | 16-bit |
| 1 | 0 | 32-bit |
| 1 | 1 | Test Mode |

### MEMORY (SROM/SDRAM) ADDRESS PIN CONNECTIONS

| MEMORY ADDR. PIN | S3C2440X ADDR. @ 8-bit DATA BUS | S3C2440X ADDR. @ 16-bit DATA BUS | S3C2440X ADDR. @ 32-bit DATA BUS |
|---|---|---|---|
| A0 | A0 | A1 | A2 |
| A1 | A1 | A2 | A3 |
| . . . | . . . | . . . | . . . |

## SDRAM BANK ADDRESS PIN CONNECTION EXAMPLE

**Table 5-2. SDRAM Bank Address Configuration**

| Bank Size | Bus Width | Base Component | Memory Configuration | Bank Address |
|---|---|---|---|---|
| 2MByte | x8 | 16Mbit | (1M x 8 x 2Bank) x 1 | A20 |
| | x16 | | (512K x 16 x 2B) x 1 | |
| 4MB | x16 | | (1M x 8 x 2B) x 2 | A21 |
| | x16 | | (1M x 8 x 2B) x 2 | |
| 8MB | x16 | 16Mb | (2M x 4 x 2B) x 4 | A22 |
| | x32 | | (1M x 8x 2B) x 4 | |
| | x8 | 64Mb | (4M x 8 x 2B) x 1 | |
| | x8 | | (2M x 8 x 4B) x 1 | A[22:21] |
| | x16 | | (2M x 16 x 2B) x 1 | A22 |
| | x16 | | (1M x 16 x 4B) x 1 | A[22:21] |
| | x32 | | (512K x 32 x 4B) x 1 | |
| 16MB | x32 | 16Mb | (2M x 4 x 2B) x 8 | A23 |
| | x8 | 64Mb | (8M x 4 x 2B) x 2 | |
| | x8 | | (4M x 4 x 4B) x 2 | A[23:22] |
| | x16 | | (4M x 8 x 2B) x 2 | A23 |
| | x16 | | (2M x 8 x 4B) x 2 | A[23:22] |
| | x32 | | (2M x 16 x 2B) x 2 | A23 |
| | x32 | | (1M x 16 x 4B) x 2 | A[23:22] |
| | x8 | 128Mb | (4M x 8 x 4B) x 1 | |
| | x16 | | (2M x 16 x 4B) x 1 | |
| 32MB | x16 | 64Mb | (8M x 4 x 2B) x 4 | A24 |
| | x16 | | (4M x 4 x 4B) x 4 | A[24:23] |
| | x32 | | (4M x 8 x 2B) x 4 | A24 |
| | x32 | | (2M x 8 x 4B) x 4 | A[24:23] |
| | x16 | 128Mb | (4M x 8 x 4B) x 2 | |
| | x32 | | (2M x 16 x 4B) x 2 | |
| | x8 | 256Mb | (8M x 8 x 4B) x 1 | |
| | x16 | | (4M x 16 x 4B) x 1 | |
| 64MB | x32 | 128Mb | (4M x 8 x 4B) x 4 | A[25:24] |
| | x16 | 256Mb | (8M x 8 x 4B) x 2 | |
| | x32 | | (4M x 16 x 4B) x 2 | |
| | x8 | 512Mb | (16M x 8 x 4B) x 1 | |
| 128MB | x32 | 256Mbit | (8M x 8 x 4Bank) x 4 | A[26:25] |
| | x8 | 512Mb | (32M x 4 x 4B) x 2 | |
| | x16 | | (16M x 8 x 4B) x 2 | |
| | x32 | | (8M x 16 x 4B) x 2 | |

**nWAIT PIN OPERATION**

If the WAIT corresponding to each memory bank is enabled, the nOE duration should be prolonged by the external nWAIT pin while the memory bank is active. nWAIT is checked from tacc-1. nOE will be deasserted at the next clock after sampling nWAIT is high. The nWE signal have the same relation with nOE.

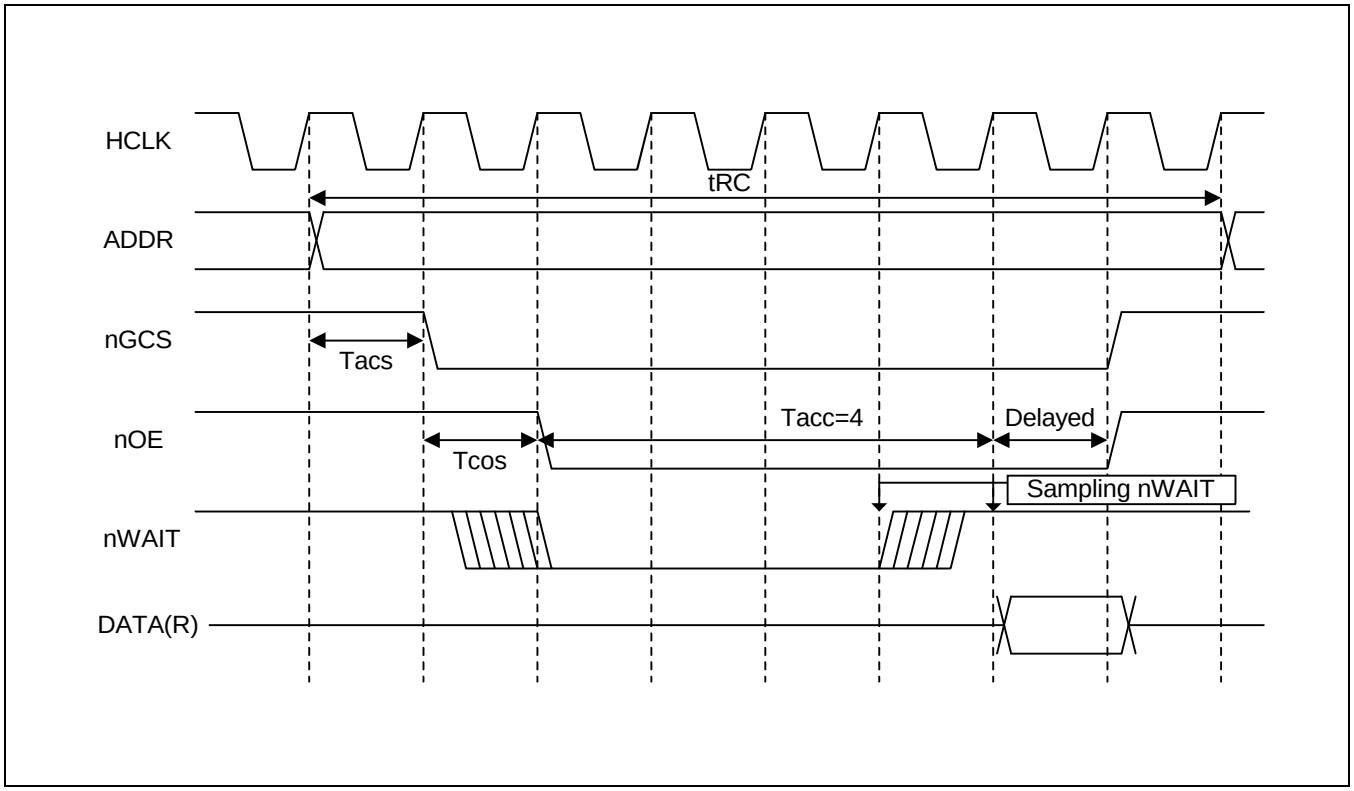



**Figure 5-2. S3C2440X External nWAIT Timing Diagram (Tacc=4)**

### nXBREQ/nXBACK Pin Operation

If nXBREQ is asserted, the S3C2440X will respond by lowering nXBACK. If nXBACK=L, the address/data bus and memory control signals are in Hi-Z state as shown in Table 1-1. When nXBREQ is de-asserted, the nXBACK will also be de-asserted.

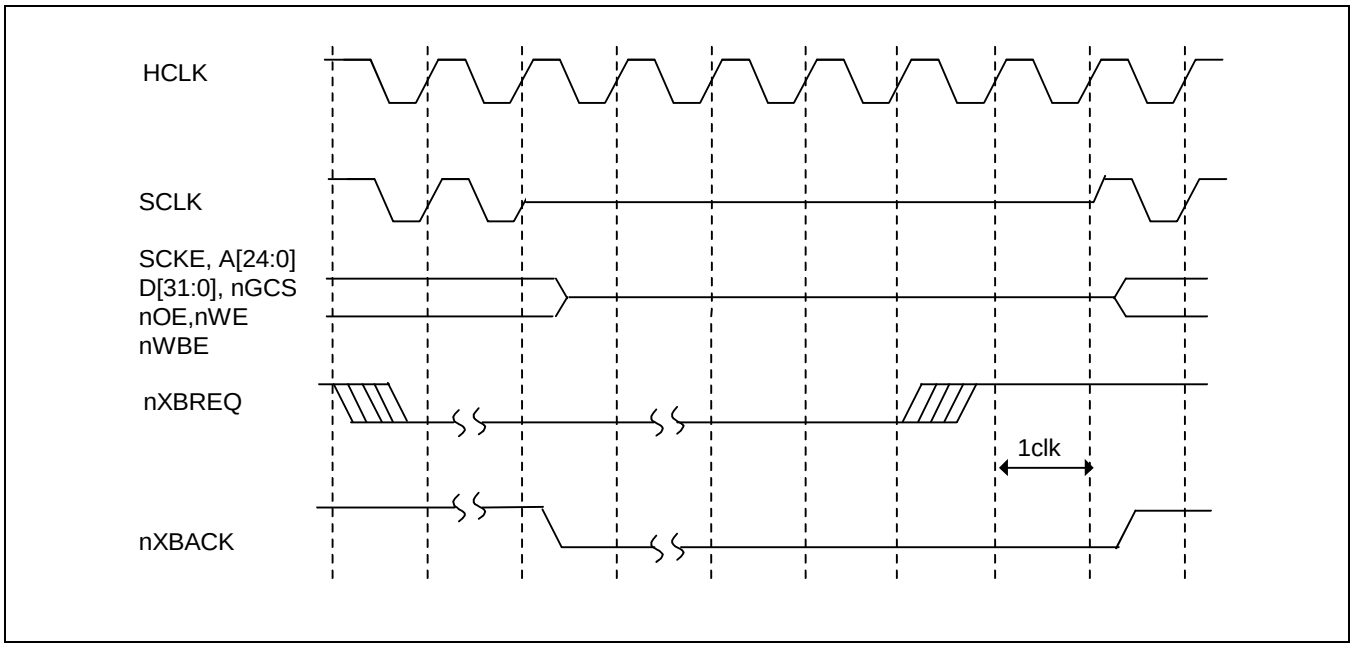



**Figure 5-3. S3C2440X nXBREQ/nXBACK Timing Diagram**

## ROM Memory Interface Examples

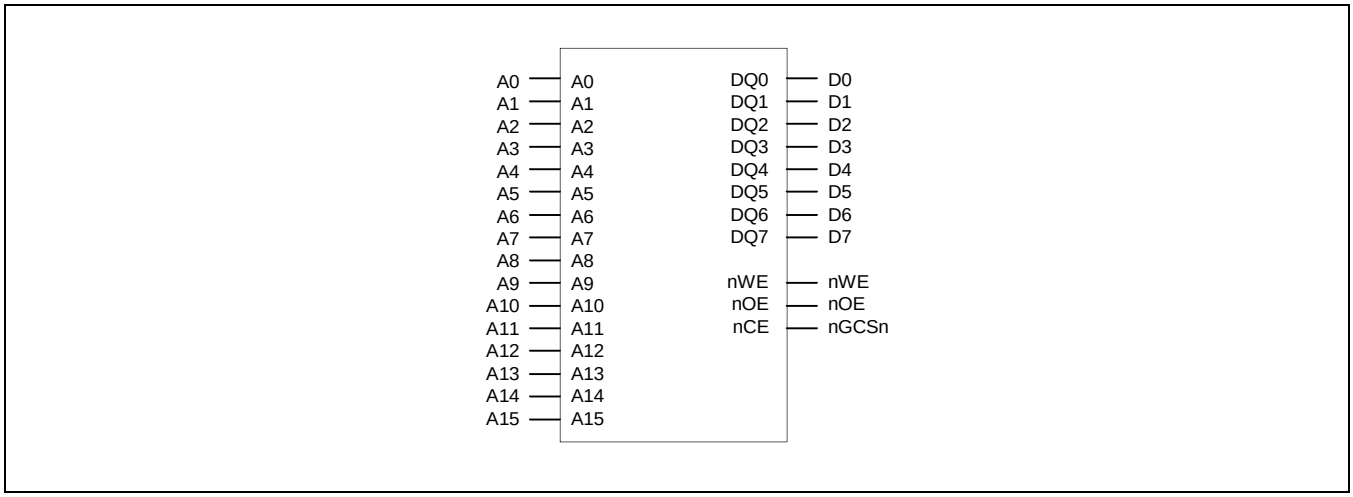



**Figure 5-4. Memory Interface with 8-bit ROM**

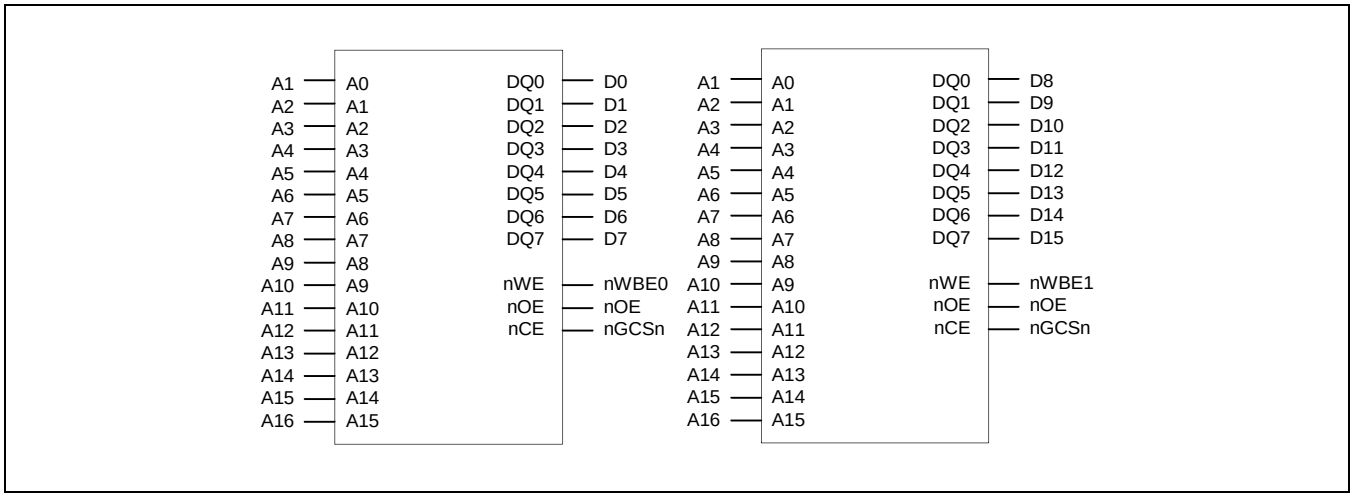



**Figure 5-5. Memory Interface with 8-bit ROM x 2**

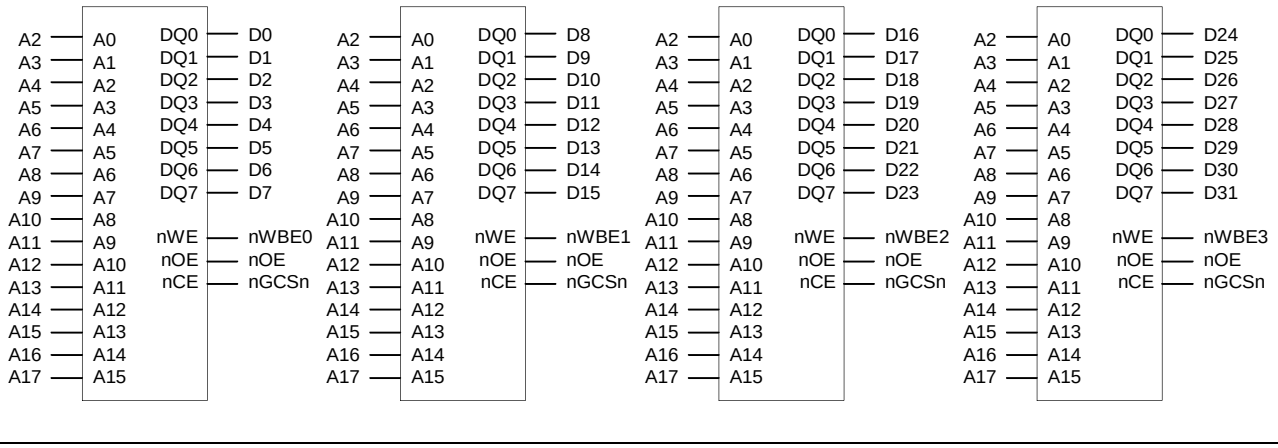



**Figure 5-6. Memory Interface with 8-bit ROM x 4**

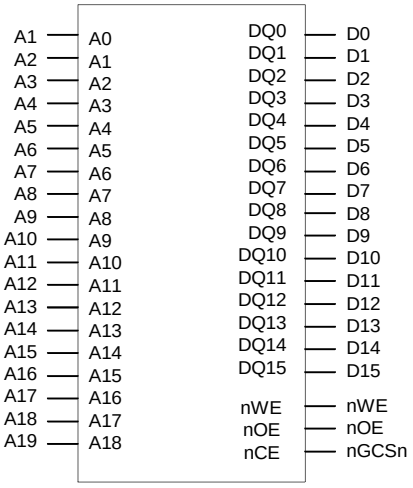



**Figure 5-7. Memory Interface with 16-bit ROM**

## SRAM Memory Interface Examples

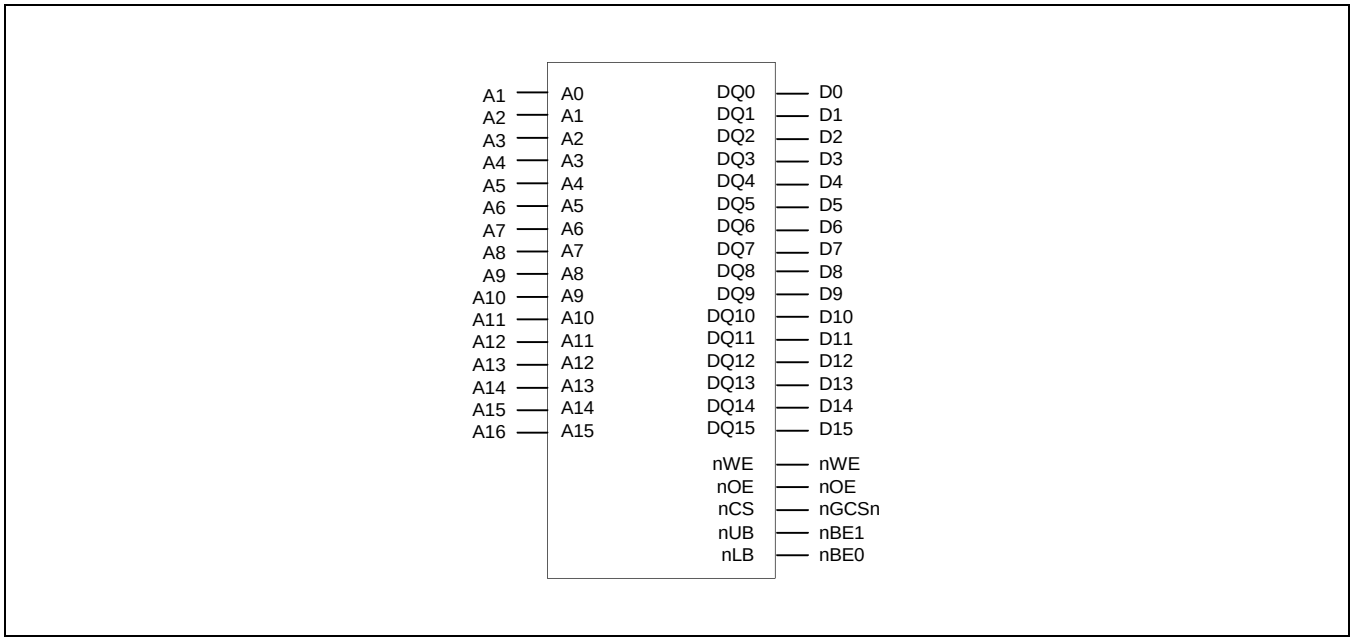



**Figure 5-8. Memory Interface with 16-bit SRAM**

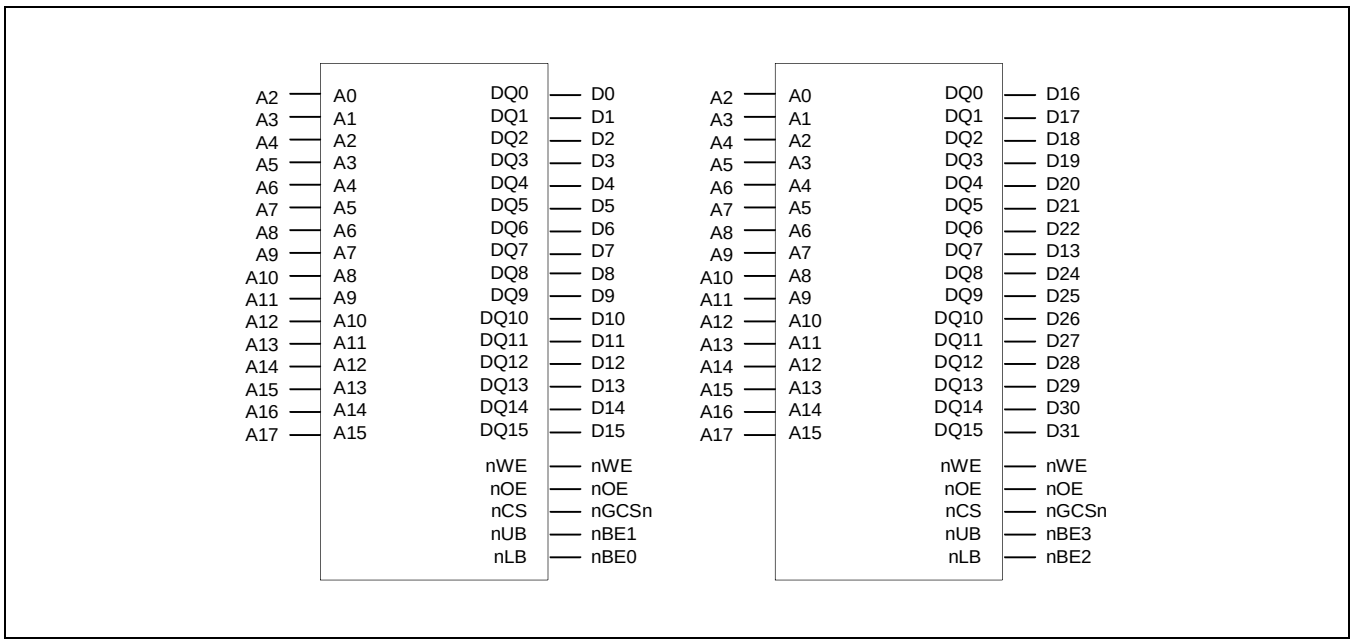



**Figure 5-9. Memory Interface with 16-bit SRAM x 2**

## SDRAM Memory Interface Examples

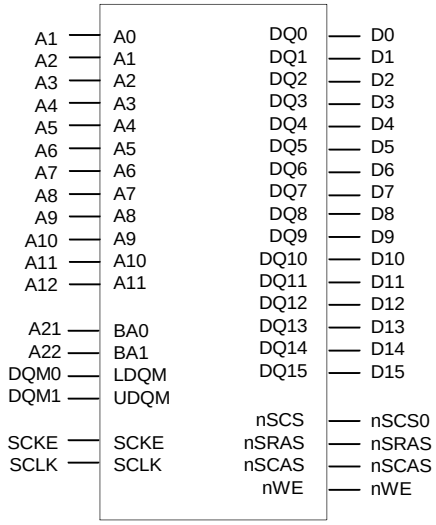



**Figure 5-10. Memory Interface with 16-bit SDRAM (4Mx16, 4banks)**

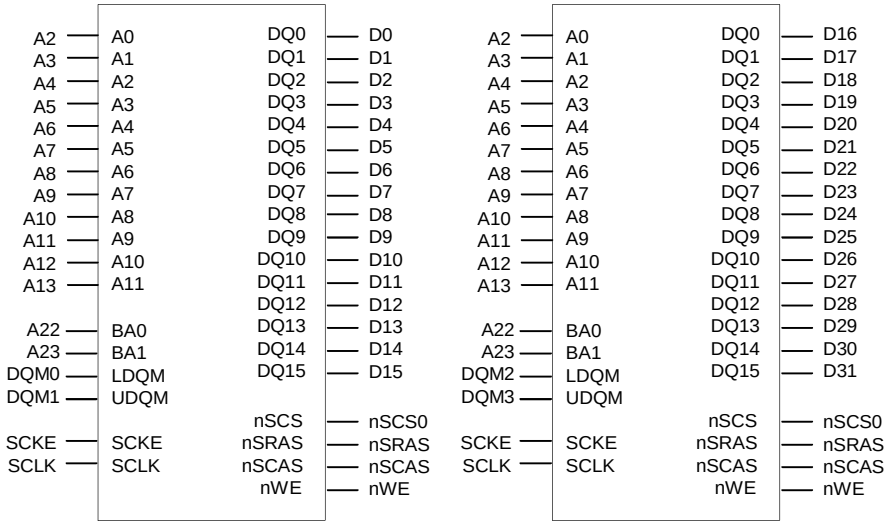



**Figure 5-11. Memory Interface with 16-bit SDRAM (4Mx16 * 2ea, 4banks)**

**Note:** Refer to Table 5-2 for the Bank Address configurations of SDRAM.

**PROGRAMMABLE ACCESS CYCLE**

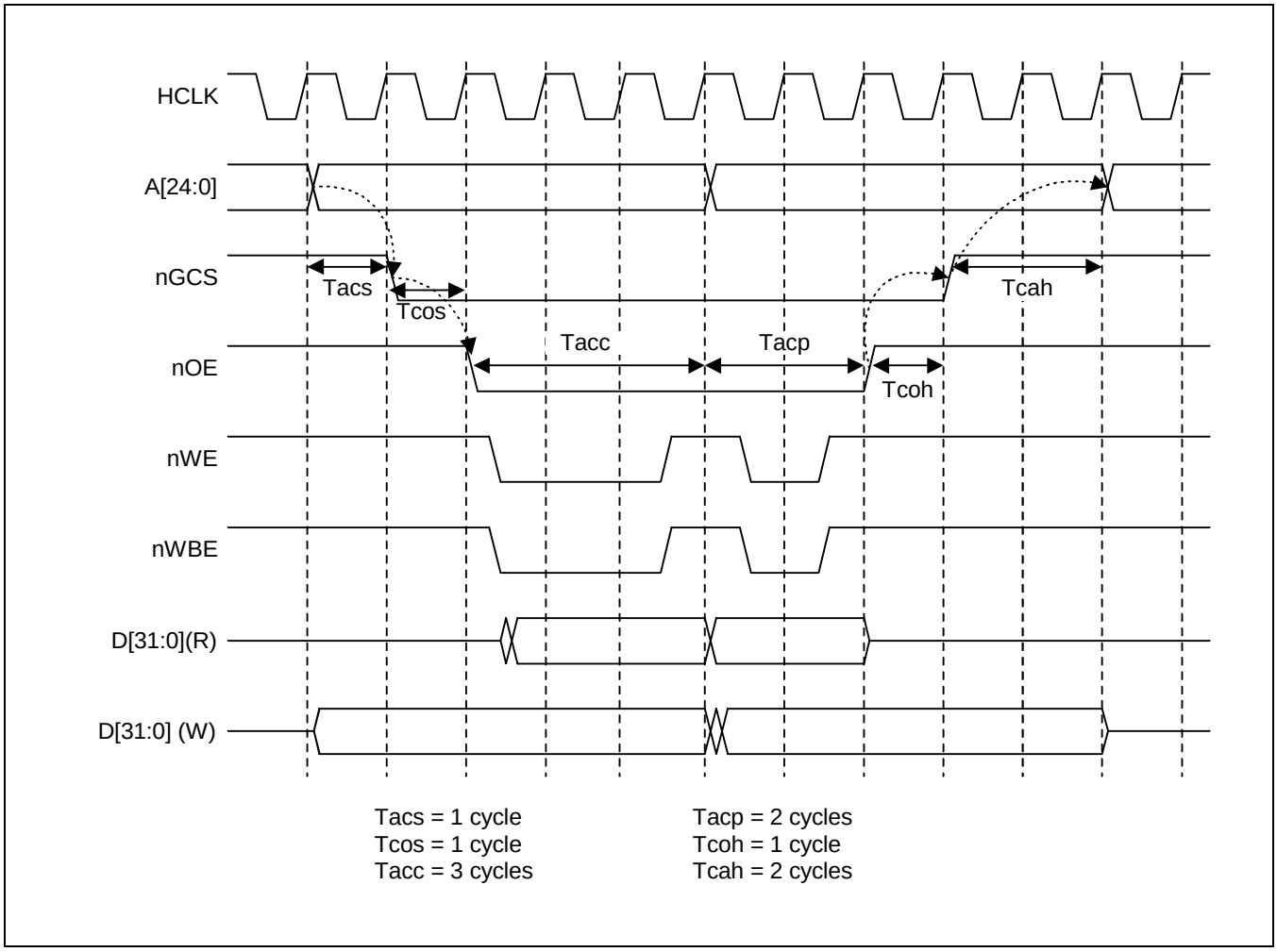



**Figure 5-12. S3C2440X nGCS Timing Diagram**

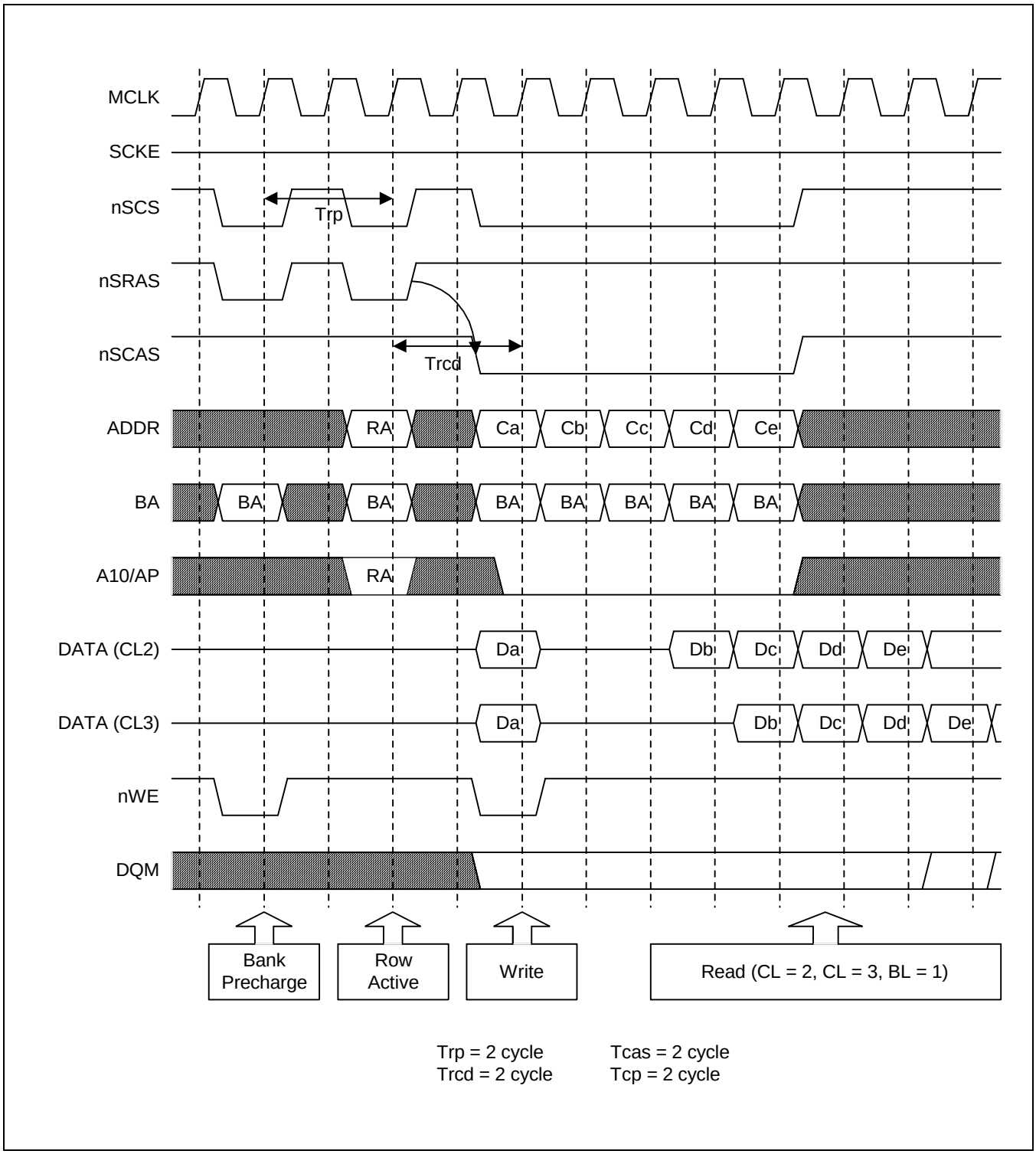



**Figure 5-13. S3C2440X SDRAM Timing Diagram**

## BUS WIDTH & WAIT CONTROL REGISTER (BWSCON)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BWSCON | 0x48000000 | R/W | Bus width & wait status control register | 0x000000 |

| BWSCON | Bit | Description | Initial state |
|---|---|---|---|
| ST7 | [31] | Determine SRAM for using UB/LB for bank 7.<br>0 = Not using UB/LB (The pins are dedicated nWBE[3:0])<br>1 = Using UB/LB (The pins are dedicated nBE[3:0]) | 0 |
| WS7 | [30] | Determine WAIT status for bank 7.<br>0 = WAIT disable 1 = WAIT enable | 0 |
| DW7 | [29:28] | Determine data bus width for bank 7.<br>00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved | 0 |
| ST6 | [27] | Determine SRAM for using UB/LB for bank 6.<br>0 = Not using UB/LB (The pins are dedicated nWBE[3:0 )<br>1 = Using UB/LB (The pins are dedicated nBE[3:0]) | 0 |
| WS6 | [26] | Determine WAIT status for bank 6.<br>0 = WAIT disable, 1 = WAIT enable | 0 |
| DW6 | [25:24] | Determine data bus width for bank 6.<br>00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved | 0 |
| ST5 | [23] | Determine SRAM for using UB/LB for bank 5.<br>0 = Not using UB/LB (The pins are dedicated nWBE[3:0])<br>1 = Using UB/LB (The pins are dedicated nBE[3:0]) | 0 |
| WS5 | [22] | Determine WAIT status for bank 5.<br>0 = WAIT disable, 1 = WAIT enable | 0 |
| DW5 | [21:20] | Determine data bus width for bank 5.<br>00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved | 0 |
| ST4 | [19] | Determine SRAM for using UB/LB for bank 4.<br>0 = Not using UB/LB (The pins are dedicated nWBE[3:0])<br>1 = Using UB/LB (The pins are dedicated nBE[3:0]) | 0 |
| WS4 | [18] | Determine WAIT status for bank 4.<br>0 = WAIT disable 1 = WAIT enable | 0 |
| DW4 | [17:16] | Determine data bus width for bank 4.<br>00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved | 0 |
| ST3 | [15] | Determine SRAM for using UB/LB for bank 3.<br>0 = Not using UB/LB (The pins are dedicated nWBE[3:0])<br>1 = Using UB/LB (The pins are dedicated nBE[3:0]) | 0 |
| WS3 | [14] | Determine WAIT status for bank 3.<br>0 = WAIT disable 1 = WAIT enable | 0 |
| DW3 | [13:12] | Determine data bus width for bank 3.<br>00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved | 0 |
| ST2 | [11] | Determine SRAM for using UB/LB for bank 2.<br>0 = Not using UB/LB (The pins are dedicated nWBE[3:0])<br>1 = Using UB/LB (The pins are dedicated nBE[3:0].) | 0 |

**BUS WIDTH & WAIT CONTROL REGISTER (BWSCON) (Continued)**

| | | | |
|---|---|---|---|
| WS2 | [10] | Determine WAIT status for bank 2.<br>0 = WAIT disable 1 = WAIT enable | 0 |
| DW2 | [9:8] | Determine data bus width for bank 2.<br>00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved | 0 |
| ST1 | [7] | Determine SRAM for using UB/LB for bank 1.<br>0 = Not using UB/LB (The pins are dedicated nWBE[3:0])<br>1 = Using UB/LB (The pins are dedicated nBE[3:0]) | 0 |
| WS1 | [6] | Determine WAIT status for bank 1.<br>0 = WAIT disable, 1 = WAIT enable | 0 |
| DW1 | [5:4] | Determine data bus width for bank 1.<br>00 = 8-bit 01 = 16-bit, 10 = 32-bit 11 = reserved | 0 |
| DW0 | [2:1] | Indicate data bus width for bank 0 (read only).<br>01 = 16-bit, 10 = 32-bit<br>The states are selected by OM[1:0] pins | - |
| Reserved | [0] | Reserve to 0 | 0 |

**Note:**

1. All types of master clock in this memory controller correspond to the bus clock.
   For example, HCLK in SRAM is the same as the bus clock, and SCLK in SDRAM is also the same as the bus clock. In this chapter (Memory Controller), one clock means one bus clock.
2. nBE[3:0] is the 'AND' signal nWBE[3:0] and nOE.

**BANK CONTROL REGISTER (BANKCONn: nGCS0-nGCS5)**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BANKCON0 | 0x48000004 | R/W | Bank 0 control register | 0x0700 |
| BANKCON1 | 0x48000008 | R/W | Bank 1 control register | 0x0700 |
| BANKCON2 | 0x4800000C | R/W | Bank 2 control register | 0x0700 |
| BANKCON3 | 0x48000010 | R/W | Bank 3 control register | 0x0700 |
| BANKCON4 | 0x48000014 | R/W | Bank 4 control register | 0x0700 |
| BANKCON5 | 0x48000018 | R/W | Bank 5 control register | 0x0700 |

| BANKCONn | Bit | Description | Initial State |
|---|---|---|---|
| Tacs | [14:13] | Address set-up time before nGCSn<br>00 = 0 clock 01 = 1 clock<br>10 = 2 clocks 11 = 4 clocks | 00 |
| Tcos | [12:11] | Chip selection set-up time before nOE<br>00 = 0 clock 01 = 1 clock<br>10 = 2 clocks 11 = 4 clocks | 00 |
| Tacc | [10:8] | Access cycle<br>000 = 1 clock 001 = 2 clocks<br>010 = 3 clocks 011 = 4 clocks<br>100 = 6 clocks 101 = 8 clocks<br>110 = 10 clocks 111 = 14 clocks<br>Note: When nWAIT signal is used, Tacc ≥ 4 clocks. | 111 |
| Tcoh | [7:6] | Chip selection hold time after nOE<br>00 = 0 clock 01 = 1 clock<br>10 = 2 clocks 11 = 4 clocks | 000 |
| Tcah | [5:4] | Address hold time after nGCSn<br>00 = 0 clock 01 = 1 clock<br>10 = 2 clocks 11 = 4 clocks | 00 |
| Tacp | [3:2] | Page mode access cycle @ Page mode<br>00 = 2 clocks 01 = 3 clocks<br>10 = 4 clocks 11 = 6 clocks | 00 |
| PMC | [1:0] | Page mode configuration<br>00 = normal (1 data) 01 = 4 data<br>10 = 8 data 11 = 16 data | 00 |

**BANK CONTROL REGISTER (BANKCONn: nGCS6-nGCS7)**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BANKCON6 | 0x4800001C | R/W | Bank 6 control register | 0x18008 |
| BANKCON7 | 0x48000020 | R/W | Bank 7 control register | 0x18008 |

| BANKCONn | Bit | Description | Initial State |
|---|---|---|---|
| MT | [16:15] | Determine the memory type for bank6 and bank7.<br>00 = ROM or SRAM 01 = Reserved (Do not use)<br>10 = Reserved (Do not use) 11 = Sync. DRAM | 11 |
| **Memory Type = ROM or SRAM [MT=00] (15-bit)** | | | |
| Tacs | [14:13] | Address set-up time before nGCS<br>00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks | 00 |
| Tcos | [12:11] | Chip selection set-up time before nOE<br>00 = 0 clock 01 = 1 clock 10 = 2 clocks 11 = 4 clocks | 00 |
| Tacc | [10:8] | Access cycle<br>000 = 1 clock 001 = 2 clocks<br>010 = 3 clocks 011 = 4 clocks<br>100 = 6 clocks 101 = 8 clocks<br>110 = 10 clocks 111 = 14 clocks | 111 |
| Toch | [7:6] | Chip selection hold time after nOE<br>00 = 0 clock 01 = 1 clock<br>10 = 2 clocks 11 = 4 clocks | 00 |
| Tcah | [5:4] | Address hold time after nGCSn<br>00 = 0 clock 01 = 1clock 10 = 2 clocks 11 = 4 clocks | 00 |
| Tacp | [3:2] | Page mode access cycle @ Page mode<br>00 = 2 clocks 01 = 3 clocks<br>10 = 4 clocks 11 = 6 clocks | 00 |
| PMC | [1:0] | Page mode configuration<br>00 = normal (1 data) 01 = 4 consecutive accesses<br>10 = 8 consecutive accesses 11 = 16 consecutive accesses | 00 |
| **Memory Type = SDRAM [MT=11] (4-bit)** | | | |
| Trcd | [3:2] | RAS to CAS delay<br>00 = 2 clocks 01 = 3 clocks 10 = 4 clocks | 10 |
| SCAN | [1:0] | Column address number<br>00 = 8-bit 01 = 9-bit 10= 10-bit | 00 |

## REFRESH CONTROL REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| REFRESH | 0x48000024 | R/W | SDRAM refresh control register | 0xac0000 |

| REFRESH | Bit | Description | Initial State |
|---|---|---|---|
| REFEN | [23] | SDRAM Refresh Enable<br>0 = Disable 1 = Enable (self or CBR/auto refresh) | 1 |
| TREFMD | [22] | SDRAM Refresh Mode<br>0 = CBR/Auto Refresh 1 = Self Refresh<br>In self-refresh time, the SDRAM control signals are driven to the appropriate level. | 0 |
| Trp | [21:20] | SDRAM RAS pre-charge Time<br>00 = 2 clocks 01 = 3 clocks 10 = 4 clocks 11 = Not support | 10 |
| Trc | [19:18] | SDRAM RC minimum Time<br>00 = 4 clocks 01 = 5 clocks 10 = 6 clocks 11 = 7 clocks | 11 |
| Reserved | [17:16] | Not used | 00 |
| Reserved | [15:11] | Not used | 0000 |
| Refresh Counter | [10:0] | SDRAM refresh count value. Refer to chapter 6 SDRAM refresh controller bus priority section.<br>Refresh period = ($2^{11}$-refresh_count+1)/HCLK<br>Ex) If refresh period is 7.8 us and HCLK is 100 MHz,<br>the refresh count is as follows:<br>Refresh count = $2^{11}$ + 1 - 100x7.8 = 1269 | 0 |

**BANKSIZE REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BANKSIZE | 0x48000028 | R/W | Flexible bank size register | 0x0 |

| BANKSIZE | Bit | Description | Initial State |
|---|---|---|---|
| BURST_EN | [7] | ARM core burst operation enable.<br>0 = Disable burst operation.<br>1 = Enable burst operation. | 0 |
| Reserved | [6] | Not used | 0 |
| SCKE_EN | [5] | SDRAM power down mode enable control by SCKE<br>0 = SDRAM power down mode disable<br>1 = SDRAM power down mode enable | 0 |
| SCLK_EN | [4] | SCLK is enabled only during SDRAM access cycle for reducing power consumption. When SDRAM is not accessed, SCLK becomes 'L' level.<br>0 = SCLK is always active.<br>1 = SCLK is active only during the access (recommended). | 0 |
| Reserved | [3] | Not used | 0 |
| BK76MAP | [2:0] | BANK6/7 memory map<br>010 = 128MB/128MB 001 = 64MB/64MB<br>000 = 32M/32M 111 = 16M/16M<br>110 = 8M/8M 101 = 4M/4M<br>100 = 2M/2M | 010 |

### SDRAM MODE REGISTER SET REGISTER (MRSR)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| MRSRB6 | 0x4800002C | R/W | Mode register set register bank6 | xxx |
| MRSRB7 | 0x48000030 | R/W | Mode register set register bank7 | xxx |

| MRSR | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [11:10] | Not used | - |
| WBL | [9] | Write burst length<br>0: Burst (Fixed)<br>1: Reserved | x |
| TM | [8:7] | Test mode<br>00: Mode register set (Fixed)<br>01, 10 and 11: Reserved | xx |
| CL | [6:4] | CAS latency<br>000 = 1 clock, 010 = 2 clocks, 011=3 clocks<br>Others: reserved | xxx |
| BT | [3] | Burst type<br>0: Sequential (Fixed)<br>1: Reserved | x |
| BL | [2:0] | Burst length<br>000: 1 (Fixed)<br>Others: Reserved | xxx |

**Note:** MRSR register must not be reconfigured while the code is running on SDRAM.

**Important Note:** In Sleep mode, SDRAM has to enter SDRAM self-refresh mode.

## NOTES

# 6 NAND FLASH CONTORLLER

## OVERVIEW

In recent times, NOR flash memory gets high in price while an SDRAM and a NAND flash memory get moderate, motivating some users to execute the boot code on a NAND flash and execute the main code on an SDRAM.

S3C2440X boot code can be executed on an external NAND flash memory. In order to support NAND flash boot loader, the S3C2440X is equipped with an internal SRAM buffer called 'Steppingstone'. When booting, the first 4 KBytes of the NAND flash memory will be loaded into Steppingstone and the boot code loaded into Steppingstone will be executed.

Generally, the boot code will copy NAND flash content to SDRAM. Using hardware ECC, the NAND flash data validity will be checked. Upon the completion of the copy, the main program will be executed on the SDRAM.

## FEATURES

1) Auto boot: The boot code is transferred into 4-kbytes Steppingstone during reset. After the transfer, the boot code will be executed on the Steppingstone.
2) NAND Flash memory I/F: Support 256Words, 512Bytes, 1KWords and 2KBytes Page.
3) Software mode: User can directly access NAND flash memory, *for example this feature can be used in read/erase/program NAND flash memory*.
4) Interface: 8 / 16-bit NAND flash memory interface bus.
5) Hardware ECC generation, detection and indication (Software correction).
6) SFR I/F: Support Little Endian Mode, Byte/half word/word access to Data and ECC Data register, and Word access to other registers
7) SteppingStone I/F: Support Little/Big Endian, Byte/half word/word access.
8) The Steppingstone 4-KB internal SRAM buffer can be used for another purpose after NAND flash booting.

## BLOCK DIAGRAM

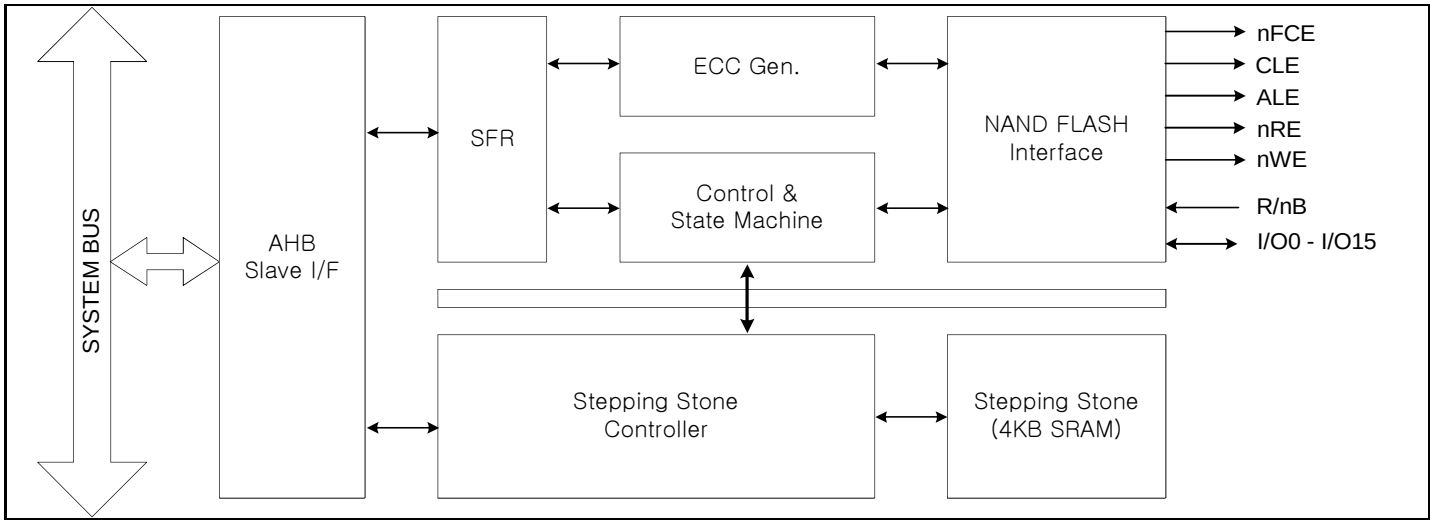



**Figure 6-1 NAND Flash Controller Block Diagram**

## BOOT LOADER FUNCTION

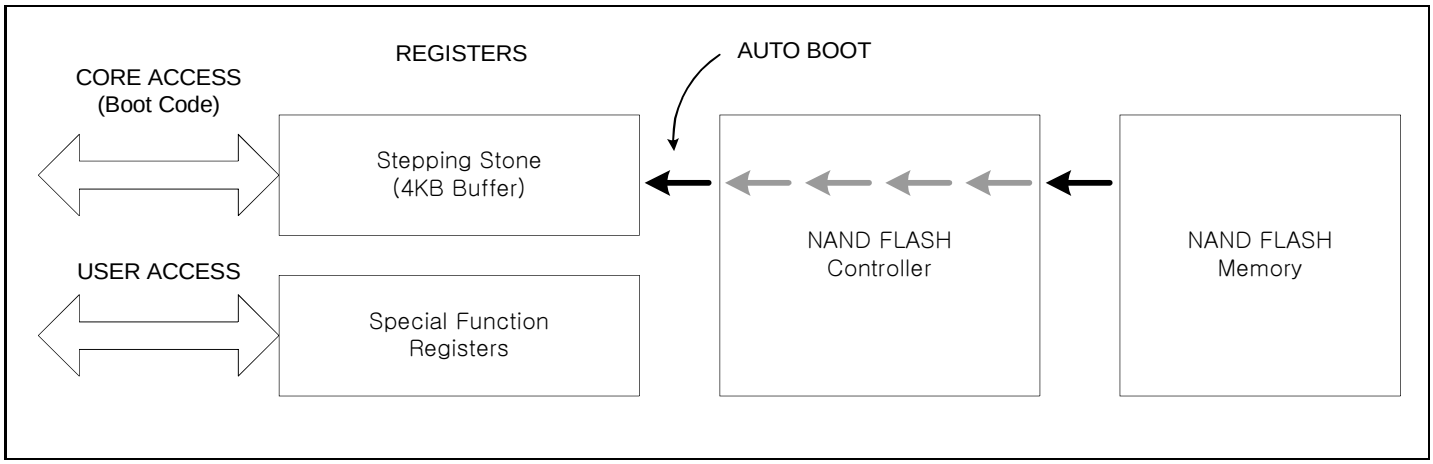



**Figure 6-2 NAND Flash Controller Boot Loader Block Diagram**

During reset, Nand flash controller will get information about connected NAND flash through Pin status(NCON(Adv flash), GPG13(Page size), GPG14(Address cycle), GPG15(Bus width) – refer to **PIN CONFIGURATION**), After power-on or system reset is occurred, the NAND Flash controller load automatically the 4-KBytes boot loader codes. After loading the boot loader codes, the boot loader code in steppingstone is executed.

**NOTE** : During the auto boot, the ECC is not checked. So, the first 4-KB of NAND flash should have no bit error.

## PIN CONFIGURATION

- OM[1:0] = 00: Enable NAND flash memory boot
- NCON : NAND flash memory selection(Normal / Advance)

  0: Normal NAND flash(256Words/512Bytes page size, 3/4 address cycle)

  1: Advance NAND flash(1KWords/2KBytes page size, 4/5 address cycle)
- GPG13 : NAND flash memory page capacitance selection

  0: Page=256Words(NCON = 0) or Page=1KWords(NCON = 1)

  1: Page=512Bytes(NCON = 0) or Page=2KBytes(NCON = 1)
- GPG14: NAND flash memory address cycle selection

  0: 3 address cycle(NCON = 0) or 4 address cycle(NCON = 1)

  1: 4 address cycle(NCON = 0) or 5 address cycle(NCON = 1)
- GPG15 : NAND flash memory bus width selection

  0: 8-bit bus width

  1: 16-bit bus width

## NAND FLASH MEMORY CONFIGURATION TABLE

| NCON0 | GPG13 | GPG14 | GPG15 |
|---|---|---|---|
| 0: Normal NAND | 0: 256Words | 0: 3-Addr | 0: 8-bit bus width |
| | 1: 512Bytes | 1: 4-Addr | |
| 1: Advance NAND | 0: 1Kwords | 0: 4-Addr | 1: 16-bit bus width |
| | 1: 2Kbytes | 1: 5-Addr | |

**Note: With above 4-bit, Possible total combinations are 16, but not all the value can be used.**

**Example)** Nand flash configuration setting example.

| Parts | Page size/Total size | NCON0 | GPG13 | GPG14] | GPG15 |
|---|---|---|---|---|---|
| **K9S1208V0M-xxxx** | 512Byte / 512Mbit | 0 | 1 | 1 | 0 |
| **K9K2G16U0M-xxxx** | 1KW / 2Gbit | 1 | 0 | 1 | 1 |

## NAND FLASH MEMORY TIMING

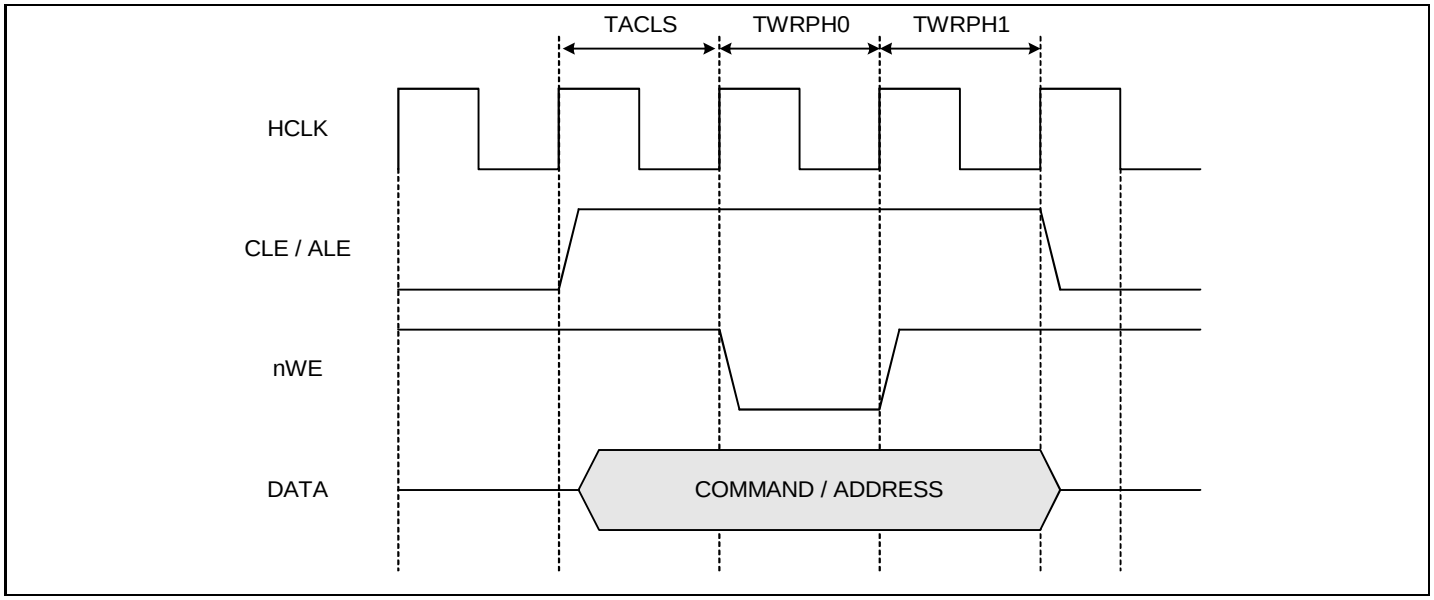



**Figure 6-3. CLE & ALE Timing (TACLS=1, TWRPH0=0, TWRPH1=0)**

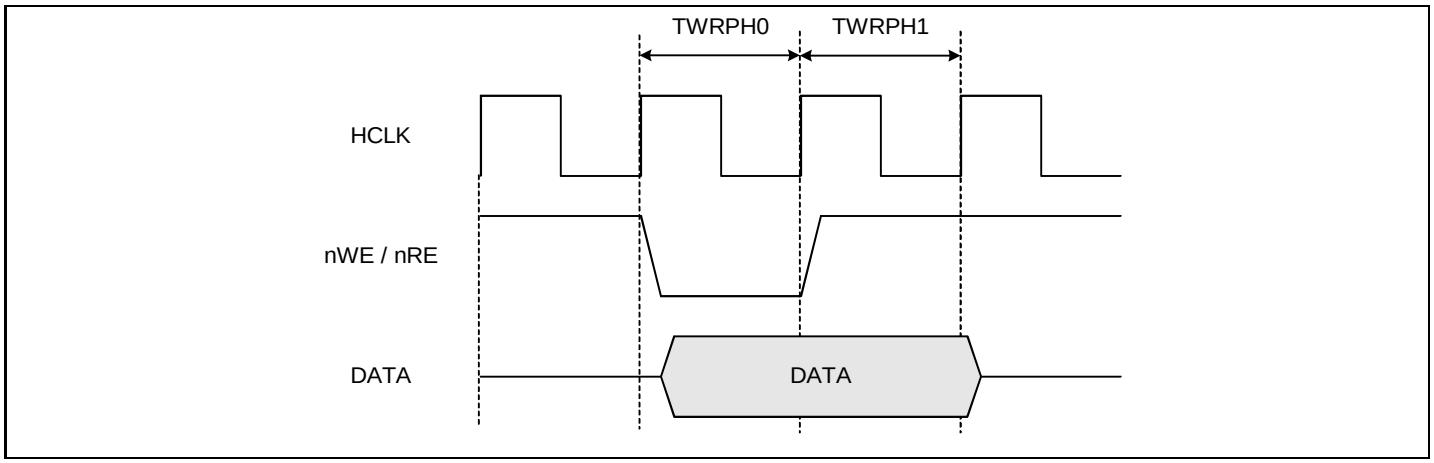



**Figure 6-4 nWE & nRE Timing (TWRPH0=0, TWRPH1=0)**

## SOFTWARE MODE

S3C2440X only supports software mode access. Using this mode, you can completely access the NAND flash memory. The NAND Flash Controller supports direct access interface with the NAND flash memory.

1) Writing to the command register = the NAND Flash Memory command cycle
2) Writing to the address register = the NAND Flash Memory address cycle
3) Writing to the data register = write data to the NAND Flash Memory (write cycle)
4) Reading from the data register = read data from the NAND Flash Memory (read cycle)
5) Reading main ECC registers and Spare ECC registers = read data from the NAND Flash Memory

**NOTE** : In the software mode, you have to check the RnB status input pin by using polling or interrupt.

**Data Register Configuration**

1) 16-bit NAND Flash Memory Interface

A. Word Access

| Register | Endian | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|---|
| NFDATA | Little | 2nd I/O[15:8] | 2nd I/O[ 7:0] | 1st I/O[15:8] | 1st I/O[ 7:0] |
| NFDATA | Big | 1st I/O[15:8] | 1st I/O[ 7:0] | 2nd I/O[15:8] | 2nd I/O[ 7:0] |

B. Half-word Access

| Register | Endian | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|---|
| NFDATA | Little/Big | Invalid value | Invalid value | 1st I/O[15:8] | 1st I/O[ 7:0] |

2) 8-bit NAND Flash Memory Interface

A. Word Access

| Register | Endian | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|---|
| NFDATA | Little | 4th I/O[ 7:0] | 3rd I/O[ 7:0] | 2nd I/O[ 7:0] | 1st I/O[ 7:0] |
| NFDATA | Big | 1st I/O[ 7:0] | 2nd I/O[ 7:0] | 3rd I/O[ 7:0] | 4th I/O[ 7:0] |

B. Half-word Access

| Register | Endian | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|---|
| NFDATA | Little | Invalid value | Invalid value | 2nd I/O[ 7:0] | 1st I/O[ 7:0] |
| NFDATA | Big | Invalid value | Invalid value | 1st I/O[ 7:0] | 2nd I/O[ 7:0] |

C. Byte Access

| Register | Endian | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|---|
| NFDATA | Little/Big | Invalid value | Invalid value | Invalid value | 1st I/O[ 7:0] |

**STEPPINGSTONE (4K-Byte SRAM)**

The NAND Flash controller uses Steppingstone as the buffer on booting and also you can use this area for another purpose.

### ECC(Error Correction Code)

NAND Flash controller has four ECC (Error Correction Code) modules. The two ECC modules (one for data[7:0] and the other for data[15:8]) can be used for (up to) 2048 bytes ECC Parity code generation, and the others(one for data[7:0] and the other for data[15:8]) can be used for (up to) 16 bytes ECC Parity code generation.

28bit ECC Parity Code = 22bit Line parity + 6bit Column Parity

14bit ECC Parity Code = 8bit Line parity + 6bit Column Parity

### 2048 BYTE ECC PARITY CODE ASSIGNMENT TABLE

| | DATA7 | DATA6 | DATA5 | DATA4 | DATA3 | DATA2 | DATA1 | DATA0 |
|---|---|---|---|---|---|---|---|---|
| **MECCn_0** | P64 | P64’ | P32 | P32’ | P16 | P16’ | P8 | P8’ |
| **MECCn_1** | P1024 | P1024’ | P512 | P512’ | P256 | P256’ | P128 | P128’ |
| **MECCn_2** | P4 | P4’ | P2 | P2’ | P1 | P1’ | P2048 | P2048’ |
| **MECCn_3** | P8192 | P8192’ | P4096 | P4096’ | - | - | - | - |

### 16 BYTE ECC PARITY CODE ASSIGNMENT TABLE

| | DATA7 | DATA6 | DATA5 | DATA4 | DATA3 | DATA2 | DATA1 | DATA0 |
|---|---|---|---|---|---|---|---|---|
| **SECCn_0** | P16 | P16’ | P8 | P8’ | P4 | P4’ | P2 | P2’ |
| **SECCn_1** | P1 | P1’ | P64 | P64’ | P32 | P32’ | - | - |

## ECC MODULE FEATURES

ECC generation is controlled by the ECC Lock (MainECCLock, SpareECCLock) bit of the Control register.

ECC Register Configuration (Little / Big Endian)

1) 16-bit NAND Flash Memory Interface

| Register | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|
| NFMECCD0 | 2nd ECC for I/O[15:8] | 2nd ECC for I/O[7:0] | 1st ECC for I/O[15:8] | 1st ECC for I/O[7:0] |
| NFMECCD1 | 4th ECC for I/O[15:8] | 4th ECC for I/O[7:0] | 3rd ECC for I/O[15:8] | 3rd ECC for I/O[7:0] |

| Register | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|
| NFSECCD | 2nd ECC for I/O[15:8] | 2nd ECC for I/O[7:0] | 1st ECC for I/O[15:8] | 1st ECC for I/O[7:0] |

2) 8-bit NAND Flash Memory Interface

| Register | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|
| NFMECCD0 | - | 2nd ECC for I/O[7:0] | - | 1st ECC for I/O[7:0] |
| NFMECCD1 | - | 4th ECC for I/O[7:0] | - | 3rd ECC for I/O[7:0] |

| Register | Bit [31:24] | Bit [23:16] | Bit [15:8] | Bit [7:0] |
|---|---|---|---|---|
| NFSECCD | - | 2nd ECC for I/O[7:0] | - | 1st ECC for I/O[7:0] |

## ECC PROGRAMMING GUIDE

1) In software mode, ECC module generates ECC parity code for all read / write data. So you have to reset ECC value by writing the InitECC(NFCONT[4]) bit as '1' and have to clear theMainECCLock(NFCONT[5]) bit to '0'(Unlock) before read or write data.
MainECCLock(NFCONT[5]) and SpareECCLock(NFCONT[6]) control whether ECC Parity code is generated or not.

2) Whenever data is read or written, the ECC module generates ECC parity code on register NFMECC0/1.

3) After you completely read or write one page (not include spare area data), Set the MainECCLock bit to '1'(Lock). ECC Parity code is locked and the value of the ECC status register will not be changed.

4) To generate spare area ECC parity code, Clear as '0'(Unlock) SpareECCLock(NFCONT[6]) bit.

5) Whenever data is read or written, the spare area ECC module generates ECC parity code on register NFSECC.

6) After you completely read or write spare area, Set the SpareECCLock bit to '1'(Lock). ECC Parity code is locked and the value of the ECC status register will not be changed.

7) From now, you can use these values to record to the spare area or check the bit error.

(Note) NFSECCD is for ECC in the spare area (Usually, the user will write the ECC value of main data area to

Spare area, which value will be the same as NFMECC0/1) and which is generated from the main data area.

### NAND FLASH MEMORY MAPPING

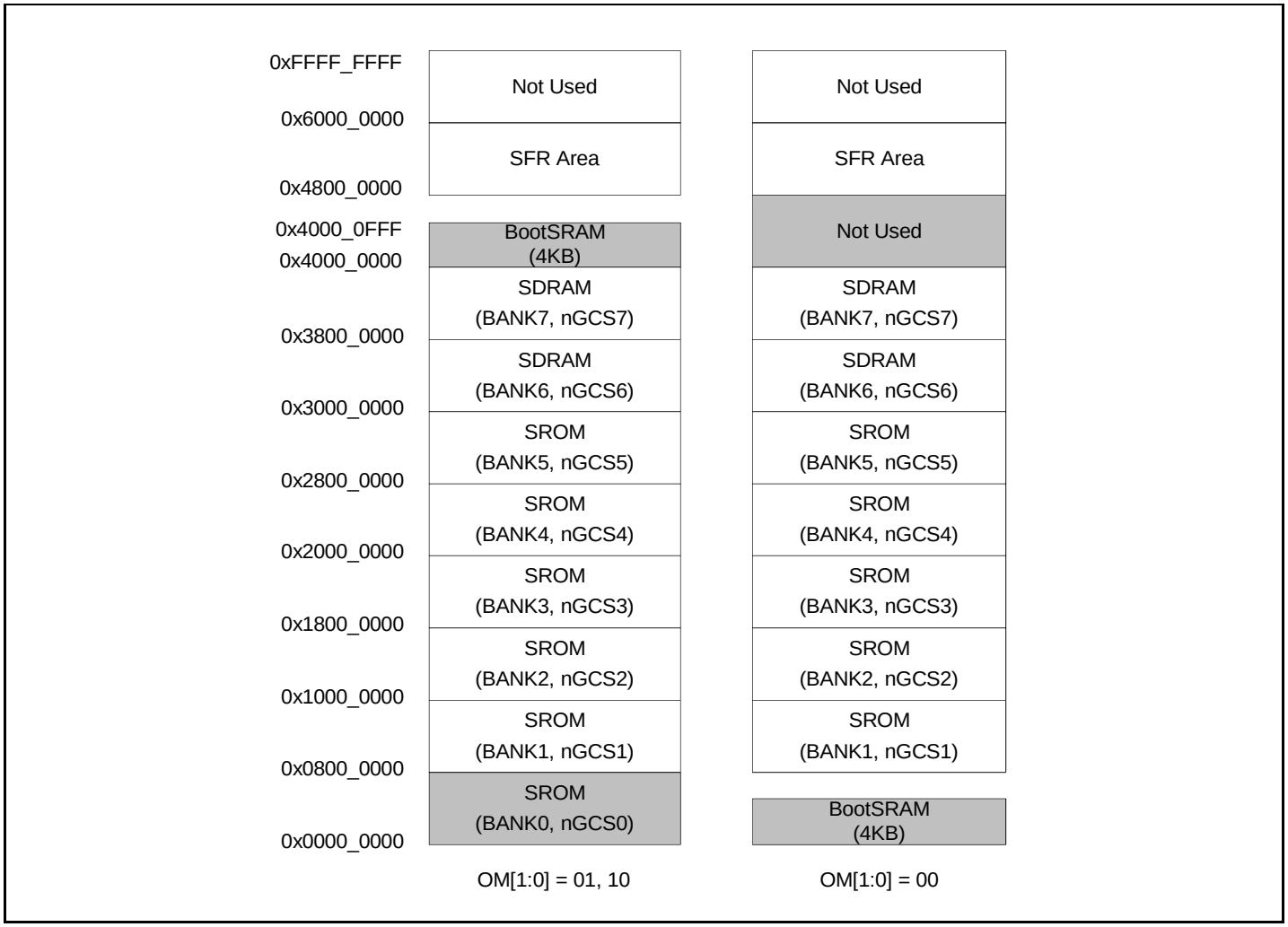



**Figure 6-4. NAND Flash Memory Mapping**

**Note**: SROM means ROM or SRAM type memory

## NAND FLASH MEMORY CONFIGURATION

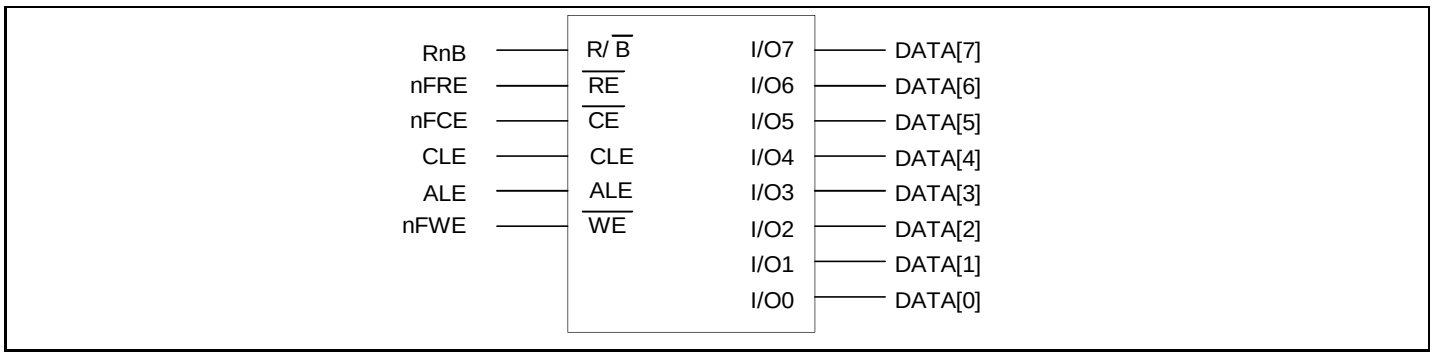



**Figure 6-1 A 8-bit NAND Flash Memory Interface**

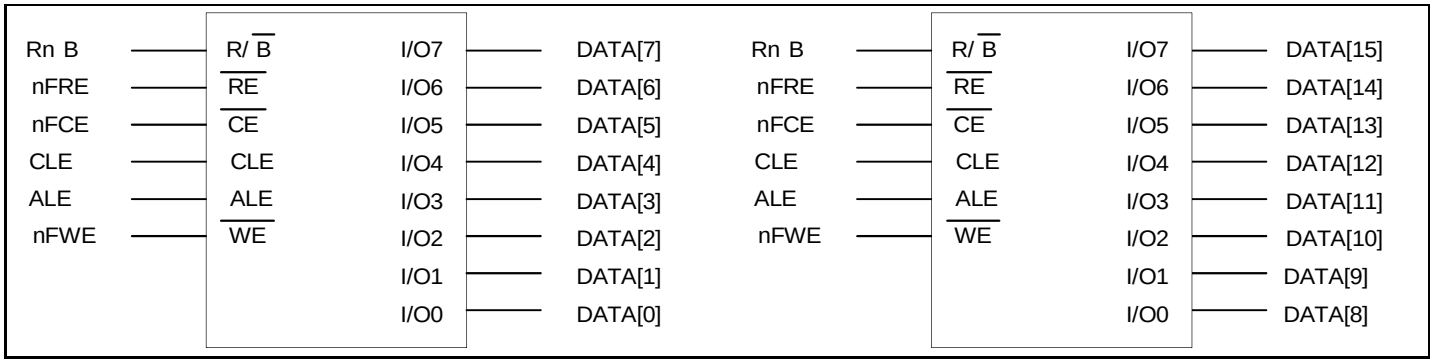



**Figure 6-2 Two 8-bit NAND Flash Memory Interface**

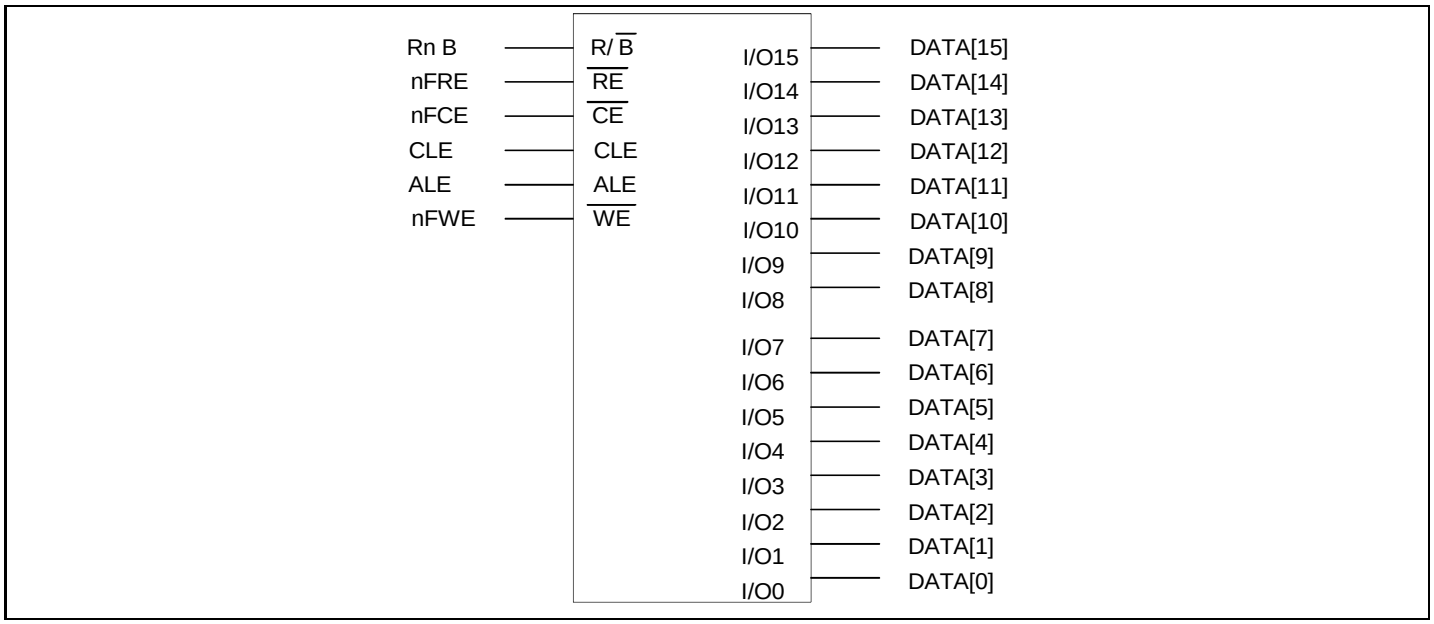



**Figure 6-3 A 16-bit NAND Flash Memory Interface**

## Nand Flash configuration Register

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFCONF | 0x4E000000 | R/W | NAND Flash Configuration register | 0x0000100X |

| NFCONF | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [15] | Reserved | - |
| TACLS | [14:12] | CLE & ALE duration setting value (0~7)<br>Duration = HCLK x TACLS | 001 |
| Reserved | [11] | Reserved | 0 |
| TWRPH0 | [10:8] | TWRPH0 duration setting value (0~7)<br>Duration = HCLK x ( TWRPH0 + 1 ) | 000 |
| Reserved | [7] | Reserved | 0 |
| TWRPH1 | [6:4] | TWRPH1 duration setting value (0~7)<br>Duration = HCLK x ( TWRPH1 + 1 ) | 000 |

| | | | |
|---|---|---|---|
| AdvFlash (Read only) | [3] | Advance NAND flash memory for auto-booting<br>0: Support 256 or 512 byte/page NAND flash memory<br>1: Support 1024 or 2048 byte/page NAND flash memory<br>This bit is determined by NCON0 pin status during reset and wake-up from sleep mode. | H/W Set (NCON0) |
| PageSize (Read only) | [2] | NAND flash memory page size for auto-booting<br>AdvFlash PageSize<br>When AdvFlash is 0,<br>0: 256 Bytes/page, 1: 512 Bytes/page<br>When AdvFlash is 1,<br>0: 1024 Bytes/page, 1: 2048 Bytes/page<br>This bit is determined by GPG13 pin status during reset and wake-up from sleep mode.<br>After reset, the GPG13 can be used as general I/O port or External interrupt. | H/W Set (GPG13) |
| AddrCycle (Read only) | [1] | NAND flash memory Address cycle for auto-booting<br>AdvFlash AddrCycle<br>When AdvFlash is 0,<br>0: 3 address cycle 1: 4 address cycle<br>When AdvFlash is 1,<br>0: 4 address cycle 1: 5 address cycle<br>This bit is determined by GPG14pin status during reset and wake-up from sleep mode.<br>After reset, the GPG14can be used as general I/O port or External interrupt. | H/W Set (GPG14) |
| BusWidth (R/W) | [0] | NAND Flash Memory I/O bus width for auto-booting and general access.<br>0: 8-bit bus 1: 16-bit bus<br>This bit is determined by GPG15 pin status during reset and wake-up from sleep mode.<br>After reset, the GPG15 can be used as general I/O port or External interrupt.<br>This bit can be changed by software. | H/W Set (GPG15) |

**CONTROL REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFCONT | 0x4E000004 | R/W | NAND Flash control register | 0x0384 |

| NFCONT | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [14:15] | Reserved | 0 |
| Lock-tight | [13] | Lock-tight configuration<br>0: Disable lock-tight 1: Enable lock-tight,<br>Once this bit is set to 1, you cannot clear. Only reset or wake up from sleep mode can make this bit disable(can not cleared by software).<br>When it is set to 1, the area setting in NFSBLK(0x4E000038) to NFEBLK(0x4E00003C)-1 is unlocked, and except this area, write or erase command will be invalid and only read command is valid.<br>When you try to write or erase locked area, the illegal access will be occur (NFSTAT[3] bit will be set).<br>If the NFSBLK and NFEBLK are same, entire area will be locked. | 0 |
| Soft Lock | [12] | Soft Lock configuration<br>0: Disable lock 1: Enable lock<br>Soft lock area can be modified at any time by software.<br>When it is set to 1, the area setting in NFSBLK(0x4E000038) to NFEBLK(0x4E00003C)-1 is unlocked, and except this area, write or erase command will be invalid and only read command is valid.<br>When you try to write or erase locked area, the illegal access will be occur (NFSTAT[3] bit will be set).<br>If the NFSBLK and NFEBLK are same, entire area will be locked. | 1 |
| Reserved | [11] | Reserved | 0 |
| EnbIllegalAccINT | [10] | Illegal access interrupt control<br>0: Disable interrupt 1: Enable interrupt<br>Illegal access interrupt is occurs when CPU tries to program or erase locking area (the area setting in NFSBLK(0x4E000038) to NFEBLK(0x4E00003C)-1). | 0 |
| EnbRnBINT | [9] | RnB status input signal transition interrupt control<br>0: Disable RnB interrupt 1: Enable RnB interrupt | 0 |
| RnB_TransMode | [8] | RnB transition detection configuration<br>0: Detect rising edge 1: Detect falling edge | 0 |
| Reserved | [7] | Reserved | 0 |
| SpareECCLock | [6] | Lock Spare area ECC generation. | 1 |

| | | | |
|---|---|---|---|
| | | 0: Unlock Spare ECC 1: Lock Spare ECC<br>Spare area ECC status register is NFSECC(0x4E000034), | |
| MainECCLock | [5] | Lock Main data area ECC generation<br>0: Unlock Main data area ECC generation<br>1: Lock Main data area ECC generation<br>Main area ECC status register is NFMECC0/1(0x4E00002C/30), | 1 |
| InitECC | [4] | Initialize ECC decoder/encoder(Write-only)<br>1: Initialize ECC decoder/encoder | 0 |
| Reserved | [2:3] | Reserved | 00 |
| Reg_nCE | [1] | NAND Flash Memory nFCE signal control<br>0: Force nFCE to low(Enable chip select)<br>1: Force nFCE to High(Disable chip select)<br>**Note**: During boot time, it is controlled automatically. This value is only valid while MODE bit is 1 | 1 |
| MODE | [0] | NAND Flash controller operating mode<br>0: NAND Flash Controller Disable (Don't work)<br>1: NAND Flash Controller Enable | 0 |

**COMMAND REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFCMMD | 0x4E000008 | R/W | NAND Flash command set register | 0x00 |

| NFCMMD | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [15:8] | Reserved | 0x00 |
| NFCMMD | [7:0] | NAND Flash memory command value | 0x00 |

**ADDRESS REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFADDR | 0x4E00000C | R/W | NAND Flash address set register | 0x0000XX00 |

| REG_ADDR | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [15:8] | Reserved | 0x00 |
| NFADDR | [7:0] | NAND Flash memory address value | 0x00 |

**DATA REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFDATA | 0x4E000010 | R/W | NAND Flash data register | 0xXXXX |

| NFDATA | Bit | Description | Initial State |
|---|---|---|---|
| NFDATA | [31:0] | NAND Flash read/program data value for I/O<br>(Note) Refer to **DATA REGISTER CONFIGURATION** in p6-5. | 0xXXXX |

**MAIN DATA AREA REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFMECCD0 | 0x4E000014 | R/W | NAND Flash ECC 1st and 2nd register for main data read<br>(Note) Refer to **ECC MODULE FEATURES** in p6-8. | 0x00000000 |
| NFMECCD1 | 0x4E000018 | R/W | NAND Flash ECC 3rd 4th register for main data read<br>(Note) Refer to **ECC MODULE FEATURES** in p6-8. | 0x00000000 |

| NFMECCD0 | Bit | Description | Initial State |
|---|---|---|---|
| ECCData1_1 | [31:24] | 2nd ECC for I/O[15:8] | 0x00 |
| ECCData1_0 | [23:16] | 2nd ECC for I/O[ 7:0]<br>Note : In Software mode, Read this register when you need to read 2nd ECC value from NAND flash memory | 0x00 |
| ECCData0_1 | [15:8] | 1st ECC for I/O[15:8] | 0x00 |
| ECCData0_0 | [7:0] | 1st ECC for I/O[ 7:0]<br>Note : In Software mode, Read this register when you need to read 1st ECC value from NAND flash memory. This register has same read function of NFDATA. | 0x00 |

(Note) Only word access is valid.

| NFMECCD1 | Bit | Description | Initial State |
|---|---|---|---|
| ECCData3_1 | [31:24] | 4th ECC for I/O[15:8] | 0x00 |
| ECCData3_0 | [23:16] | 4th ECC for I/O[ 7:0]<br>Note : In Software mode, Read this register when you need to read 4th ECC value from NAND flash memory | 0x00 |
| ECCData2_1 | [15:8] | 3rd ECC for I/O[15:8] | 0x00 |
| ECCData2_0 | [7:0] | 3rd ECC for I/O[ 7:0]<br>Note: In Software mode, Read this register when you need to read 3rd ECC value from NAND flash memory. This register has same read function of NFDATA. | 0x00 |

(Note) Only word access is valid.

**Important Note**

*MAIN DATA AREA REGISTER (NFMECCD0/1) does not meet the specification.*

*Next revision chip will meet the specification.*

*Workaround: ECC detection by software*

**SPARE AREA ECC REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFSECCD | 0x4E00001C | R/W | NAND Flash ECC(Error Correction Code) register for spare area data read | 0x00000000 |

| NFSECCD | Bit | Description | Initial State |
|---|---|---|---|
| ECCData1_1 | [31:24] | 2nd ECC for I/O[15:8] | 0x00 |
| ECCData1_0 | [23:16] | 2nd ECC for I/O[ 7:0]<br>Note: In Software mode, Read this register when you need to read 2nd ECC value from NAND flash memory | 0x00 |
| ECCData0_1 | [15:8] | 1st ECC for I/O[15:8] | 0x00 |
| ECCData0_0 | [7:0] | 1st ECC for I/O[ 7:0]<br>Note: In Software mode, Read this register when you need to read 1st ECC value from NAND flash memory. This register has same read function of NFDATA. | 0x00 |

(Note) Only word access is valid.

**NFCON STATUS REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFSTAT | 0x4E000020 | R/W | NAND Flash operation status register | 0xXX00 |

| NFSTAT | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7] | Reserved | X |
| Reserved | [4:6] | Reserved | 0 |
| IllegalAccess | [3] | Once Soft Lock or Lock-tight is enabled, The illegal access (program, erase) to the memory makes this bit set.<br>0: illegal access is not detected<br>1: illegal access is detected | 0 |
| RnB_TransDetect | [2] | When RnB low to high transition is occurred, this value set and issue interrupt if enabled. To clear this value write '1'.<br>0: RnB transition is not detected<br>1: RnB transition is detected<br>Transition configuration is set in RnB_TransMode(NFCONT[8]). | 0 |
| nCE<br>(Read-only) | [1] | The status of nCE output pin | 1 |
| RnB<br>(Read-only) | [0] | The status of RnB input pin.<br>0: NAND Flash memory busy<br>1: NAND Flash memory ready to operate | 1 |

**ECC0/1 STATUS REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFESTAT0 | 0x4E000024 | R/W | NAND Flash ECC Status register for I/O [7:0] | 0x00000000 |
| NFESTAT1 | 0x4E000028 | R/W | NAND Flash ECC Status register for I/O [15:8] | 0x00000000 |

| NFESTAT0 | Bit | Description | Initial State |
|---|---|---|---|
| SErrorDataNo | [24:21] | In spare area, Indicates which number data is error | 00 |
| SErrorBitNo | [20:18] | In spare area, Indicates which bit is error | 000 |
| MErrorDataNo | [17:7] | In main data area, Indicates which number data is error | 0x00 |
| MErrorBitNo | [6:4] | In main data area, Indicates which bit is error | 000 |
| SpareError | [3:2] | Indicates whether spare area bit fail error occurred<br>00: No Error 01: 1-bit error(correctable)<br>10: Multiple error 11: ECC area error | 00 |
| MainError | [1:0] | Indicates whether main data area bit fail error occurred<br>00: No Error 01: 1-bit error(correctable)<br>10: Multiple error 11: ECC area error | 00 |

Note : The above values are only valid when both ECC register and ECC status register have valid value.

| NFESTAT1 | Bit | Description | Initial State |
|---|---|---|---|
| SErrorDataNo | [24:21] | In spare area, Indicates which number data is error | 00 |
| SErrorBitNo | [20:18] | In spare area, Indicates which bit is error | 000 |
| MErrorDataNo | [17:7] | In main data area, Indicates which number data is error | 0x00 |
| MErrorBitNo | [6:4] | In main data area, Indicates which bit is error | 000 |
| SpareError | [3:2] | Indicates whether spare area bit fail error occurred<br>00: No Error 01: 1-bit error(correctable)<br>10: Multiple error 11: ECC area error | 00 |
| MainError | [1:0] | Indicates whether main data area bit fail error occurred<br>00: No Error 01: 1-bit error(correctable)<br>10: Multiple error 11: ECC area error | 00 |

Note : The above values are only valid when both ECC register and ECC status register have valid value.

**MAIN DATA AREA ECC0 STATUS REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFMECC0 | 0x4E00002C | R | NAND Flash ECC register for data[7:0] | 0xXXXXXX |
| NFMECC1 | 0x4E000030 | R | NAND Flash ECC register for data[15:8] | 0xXXXXXX |

| NFMECC0 | Bit | Description | Initial State |
|---|---|---|---|
| MECC0_3 | [31:24] | ECC3 for data[7:0] | 0xXX |
| MECC0_2 | [23:16] | ECC2 for data[7:0] | 0xXX |
| MECC0_1 | [15:8] | ECC1 for data[7:0] | 0xXX |
| MECC0_0 | [7:0] | ECC0 for data[7:0] | 0xXX |

| NFMECC1 | Bit | Description | Initial State |
|---|---|---|---|
| MECC1_3 | [31:24] | ECC3 data[15:8] | 0xXX |
| MECC1_2 | [23:16] | ECC2 data[15:8] | 0xXX |
| MECC1_1 | [15:8] | ECC1 data[15:8] | 0xXX |
| MECC1_0 | [7:0] | ECC0 data[15:8] | 0xXX |

(Note) The NAND flash controller generate NFMECC0/1 when read or write main area data while the MainECCLock(NFCONT[5]) bit is '0'(Unlock).

**SPARE AREA ECC STATUS REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFSECC | 0x4E000034 | R | NAND Flash ECC register for I/O [15:0] | 0xXXXXXX |

| NFSECC | Bit | Description | Initial State |
|---|---|---|---|
| SECC1_1 | [31:24] | Spare area ECC1 Status for I/O[15:8] | 0xXX |
| SECC1_0 | [23:16] | Spare area ECC0 Status for I/O[15:8] | 0xXX |
| SECC0_1 | [15:8] | Spare area ECC1 Status for I/O[7:0] | 0xXX |
| SECC0_0 | [7:0] | Spare area ECC0 Status for I/O[7:0] | 0xXX |

(Note) The NAND flash controller generate NFSECC when read or write spare area data while the SpareECCLock(NFCONT[6]) bit is '0'(Unlock).

**BLOCK ADDRESS REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| NFSBLK | 0x4E000038 | R/W | NAND Flash programmable start block address | 0x000000 |
| NFEBLK | 0x4E00003C | R/W | NAND Flash programmable end block address<br>Nand Flash can be programmed between start and end address.<br>When the Soft lock or Lock-tight is enabled and the Start and End address has same value, Entire area of NAND flash will be locked. | 0x000000 |

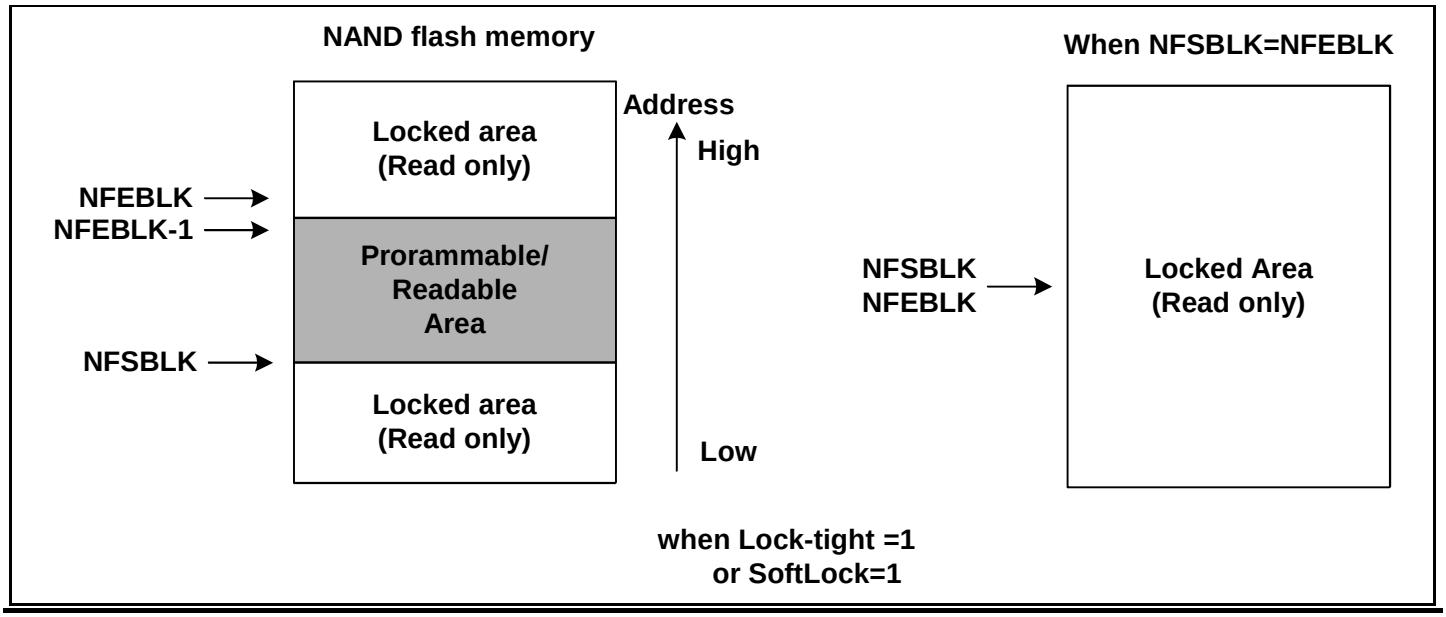
| NFSBLK | Bit | Description | Initial State |
|---|---|---|---|
| SBLK_ADDR2 | [23:16] | The 3rd block address of the block erase operation | 0x00 |
| SBLK_ADDR1 | [15:8] | The 2nd block address of the block erase operation | 0x00 |
| SBLK_ADDR0 | [7:0] | The 1st block address of the block erase operation<br>(Only bit [7:5] are valid) | 0x00 |

Note : Advance Flash's block Address start from 3-address cycle. So block address register only needs 3-bytes.

| NFEBLK | Bit | Description | Initial State |
|---|---|---|---|
| EBLK_ADDR2 | [23:16] | The 3rd block address of the block erase operation | 0x00 |
| EBLK_ADDR1 | [15:8] | The 2nd block address of the block erase operation | 0x00 |
| EBLK_ADDR0 | [7:0] | The 1st block address of the block erase operation<br>(Only bit [7:5] are valid) | 0x00 |

Note : Advance Flash's block Address start from 3-address cycle. So block address register only needs 3-bytes.

The NFSLK and NFEBLK can be changed while Soft lock bit(NFCONT[12]) is enabled. But cannot be changed when Lock-tight bit(NFCONT[13]) is set.



# 7 CLOCK & POWER MANAGEMENT

## OVERVIEW

The clock & power management block consists of three parts: Clock control, USB control, and Power control.

The Clock control logic in S3C2440X can generate the required clock signals including FCLK for CPU, HCLK for the AHB bus peripherals, and PCLK for the APB bus peripherals. The S3C2440X has two Phase Locked Loops (PLLs): one for FCLK, HCLK, and PCLK, and the other dedicated for USB block (48Mhz). The clock control logic can make slow clocks without PLL and connect/disconnect the clock to each peripheral block by software, which will reduce the power consumption.

For the power control logic, the S3C2440X has various power management schemes to keep optimal power consumption for a given task. The power management block in the S3C2440X can activate four modes: NORMAL mode, SLOW mode, IDLE mode, and SLEEP mode.

**NORMAL mode**: The block supplies clocks to CPU as well as all peripherals in the S3C2440X. In this mode, the power consumption will be maximized when all peripherals are turned on. It allows the user to control the operation of peripherals by software. *For example, if a timer is not needed, the user can disconnect the clock to the timer to reduce power consumption.*

**SLOW mode**: Non-PLL mode. Unlike the Normal mode, the Slow mode uses an external clock (XTIpll or EXTCLK) directly as FCLK in the S3C2440X without PLL. In this mode, the power consumption depends on the frequency of the external clock only. The power consumption due to PLL is excluded.

**IDLE mode**: The block disconnects clocks (FCLK) only to the CPU core while it supplies clocks to all other peripherals. The IDLE mode results in reduced power consumption due to CPU core. Any interrupt request to CPU can be woken up from the Idle mode.

**SLEEP mode**: The block disconnects the internal power. So, there occurs no power consumption due to CPU and the internal logic except the wake-up logic in this mode. Activating the SLEEP mode requires two independent power sources. One of the two power sources supplies the power for the wake-up logic. The other one supplies other internal logics including CPU, and should be controlled for power on/off. In the SLEEP mode, the second power supply source for the CPU and internal logics will be turned off. The wakeup from SLEEP mode can be issued by the EINT[15:0] or by RTC alarm interrupt.

# FUNCTIONAL DESCRIPTION

## CLOCK ARCHITECTURE

Figure 7-1 shows a block diagram of the clock architecture. The main clock source comes from an external crystal (XTIpll) or an external clock (EXTCLK). The clock generator includes an oscillator (Oscillation Amplifier), which is connected to an external crystal, and also has two PLLs (Phase-Locked-Loop), which generate the high frequency clock required in the S3C2440X.

## CLOCK SOURCE SELECTION

Table 7-1 shows the relationship between the combination of mode control pins (OM3 and OM2) and the selection of source clock for the S3C2440X. The OM[3:2] status is latched internally by referring the OM3 and OM2 pins at the rising edge of nRESET.

**Table 7-1. Clock Source Selection at Boot-Up**

| Mode OM[3:2] | MPLL State | UPLL State | Main Clock source | USB Clock Source |
|---|---|---|---|---|
| 00 | On | On | Crystal | Crystal |
| 01 | On | On | Crystal | EXTCLK |
| 10 | On | On | EXTCLK | Crystal |
| 11 | On | On | EXTCLK | EXTCLK |

**NOTE:**

1. Although the MPLL starts just after a reset, the MPLL output (Mpll) is not used as the system clock until the software writes valid settings to the MPLLCON register. Before this valid setting, the clock from external crystal or EXTCLK source will be used as the system clock directly. Even if the user does not want to change the default value of MPLLCON register, the user should write the same value into MPLLCON register.
2. OM[3:2] is used to determine a test mode when OM[1:0] is 11.

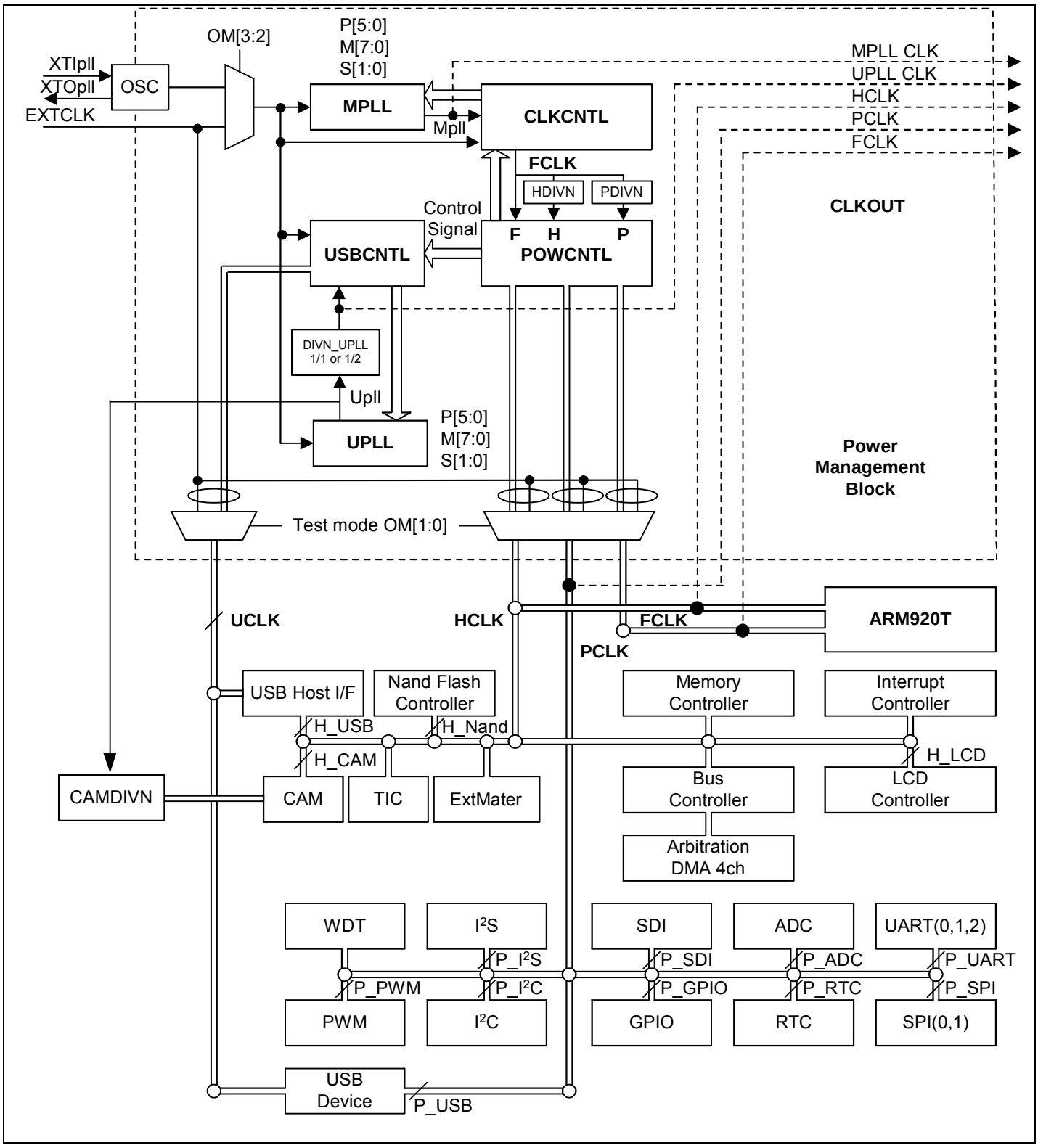



**Figure 7-1. Clock Generator Block Diagram**

## PHASE LOCKED LOOP (PLL)

The MPLL within the clock generator, as a circuit, synchronizes an output signal with a reference input signal in frequency and phase. In this application, it includes the following basic blocks as shown in Figure 7-2: the Voltage Controlled Oscillator (VCO) to generate the output frequency proportional to input DC voltage, the divider P to divide the input frequency (Fin) by p, the divider M to divide the VCO output frequency by m which is input to Phase Frequency Detector (PFD), the divider S to divide the VCO output frequency by s which is Mpll (the output frequency from MPLL block), the phase difference detector, the charge pump, and the loop filter. The output clock frequency Mpll is related to the reference input clock frequency Fin by the following equation:

$Mpll = (m * Fin) / (p * 2^s)$
m = M (the value for divider M)+ 8, p = P (the value for divider P) + 2

The UPLL within the clock generator is the same as the MPLL in every aspect.

The following sections describes the operation of the PLL, including the phase difference detector, the charge pump, the Voltage controlled oscillator (VCO), and the loop filter.

### Phase Difference Detector (PFD)

The PFD monitors the phase difference between Fref and Fvco, and generates a control signal (tracking signal) when the difference is detected. The Fref means the reference frequency as shown in the Figure 7-2.

### Charge Pump (PUMP)

The charge pump converts PFD control signals into a proportional charge in voltage across the external filter that drives the VCO.

### Loop Filter

The control signal, which the PFD generates for the charge pump, may generate large excursions (ripples) each time the Fvco is compared to the Fref. To avoid overloading the VCO, a low pass filter samples and filters the high-frequency components out of the control signal. The filter is typically a single-pole RC filter with a resistor and a capacitor.

### Voltage Controlled Oscillator (VCO)

The output voltage from the loop filter drives the VCO, causing its oscillation frequency to increase or decrease linearly as a function of variations in average voltage. When the Fvco matches Fref in terms of frequency as well as phase, the PFD stops sending control signals to the charge pump, which in turn stabilizes the input voltage to the loop filter. The VCO frequency then remains constant, and the PLL remains fixed onto the system clock.

### Usual Conditions for PLL & Clock Generator

PLL & Clock Generator generally uses the following conditions.

| | | |
|---|---|---|
| **Loop filter capacitance** | $C_{LF}$ | MPLLCAP: 2.8 nF |
| | | UPLLCAP: 700 pF |
| External X-tal frequency | - | 10 – 20 MHz (note) |
| External capacitance used for X-tal | $C_{EXT}$ | 15 – 22 pF |

**NOTES:**
1. The value could be changed.
2. FCLK must be more than three times X-tal or EXTCLK (FCLK ≥ 3X-tal or 3EXTCLK)

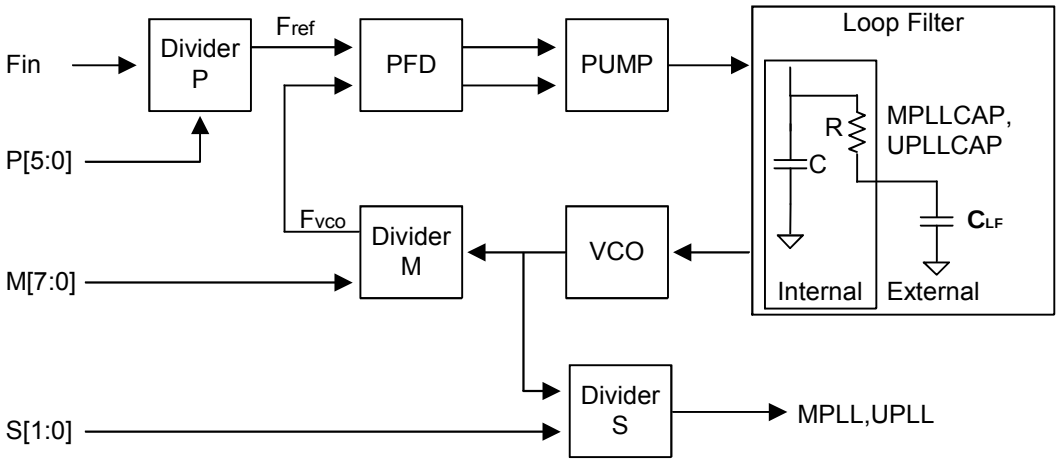
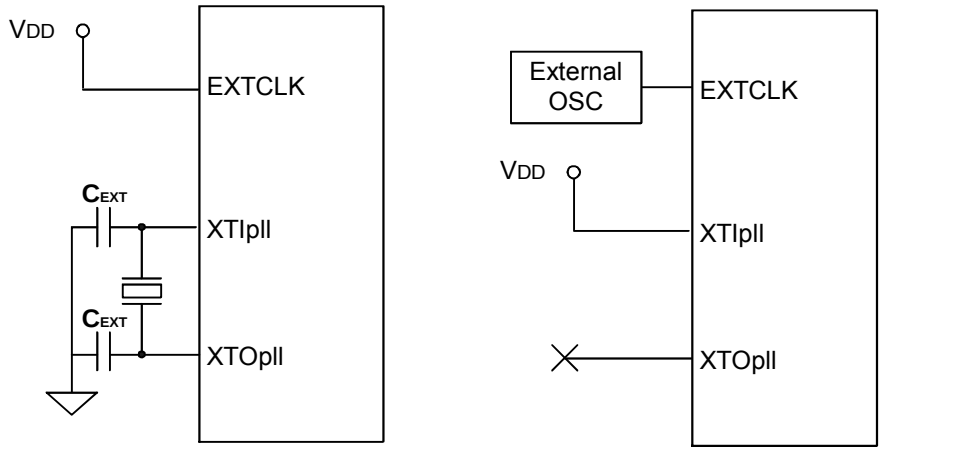



**Figure 7-2. PLL (Phase-Locked Loop) Block Diagram**




**a) X-TAL Oscillation (OM[3:2]=00)** **b) External Clock Source (OM[3:2]=11)**

**Figure 7-3. Main Oscillator Circuit Examples**

## CLOCK CONTROL LOGIC

The clock control logic determines the clock source to be used, i.e., the PLL clock (Mpll) or the direct external clock (XTIpll or EXTCLK). When PLL is configured to a new frequency value, the clock control logic disables the FCLK until the PLL output is stabilized using the PLL locking time. The clock control logic is also activated at power-on reset and wakeup from power-down mode.

### Power-On Reset (XTIpll)

Figure 7-4 shows the clock behavior during the power-on reset sequence. The crystal oscillator begins oscillation within several milliseconds. When nRESET is released after the stabilization of OSC (XTIpll) clock, the PLL starts to operate according to the default PLL configuration. However, PLL is commonly known to be unstable after power-on reset, so Fin is fed directly to FCLK instead of the Mpll (PLL output) before the software newly configures the PLLCON. Even if the user does not want to change the default value of PLLCON register after reset, the user should write the same value into PLLCON register by software.

The PLL restarts the lockup sequence toward the new frequency only after the software configures the PLL with a new frequency. FCLK can be configured as PLL output (Mpll) immediately after lock time.

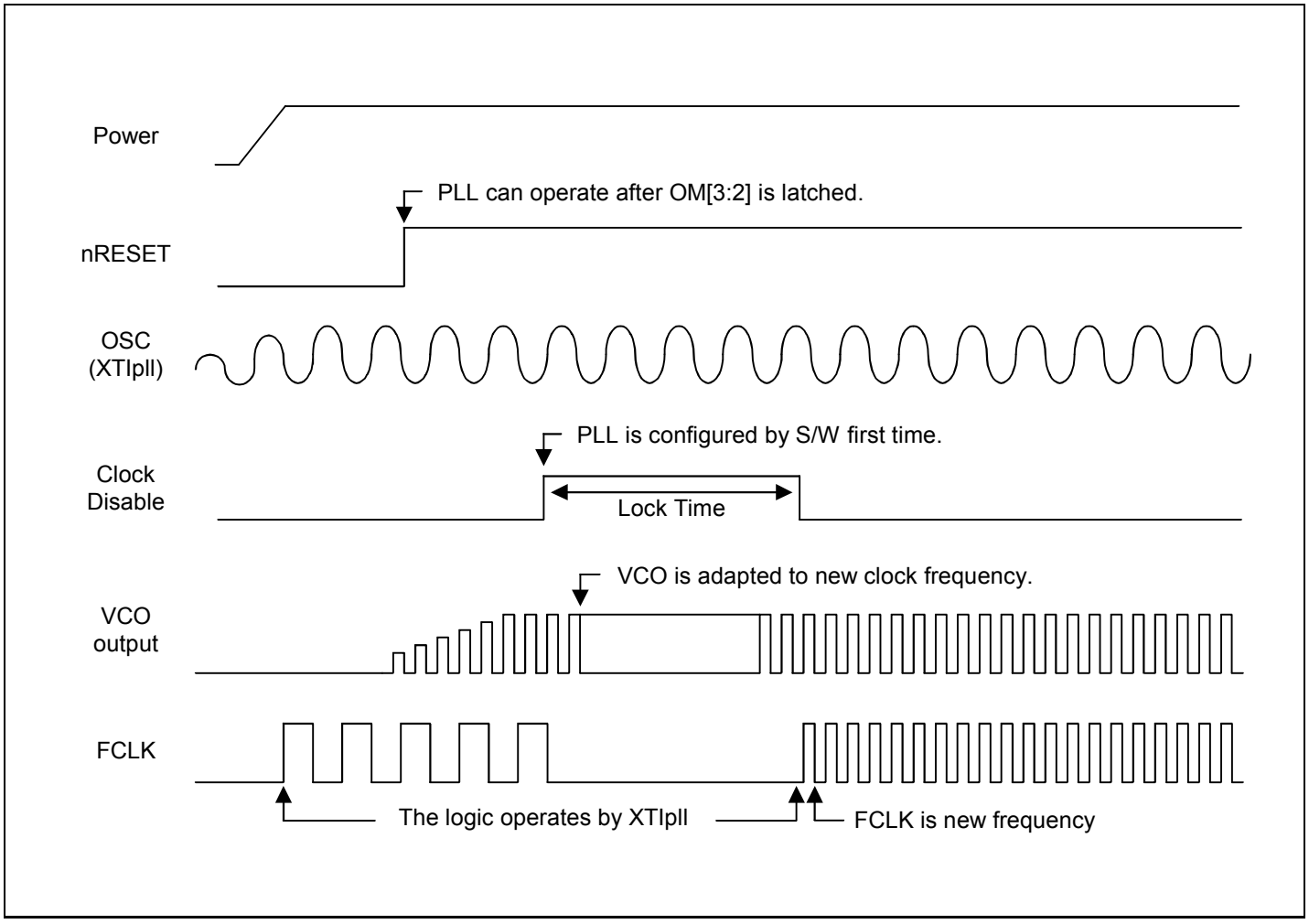



**Figure 7-4. Power-On Reset Sequence (when the external clock source is a crystal oscillator)**

### Change PLL Settings In Normal Operation Mode

During the operation of the S3C2440X in NORMAL mode, the user can change the frequency by writing the PMS value and the PLL lock time will be automatically inserted. During the lock time, the clock is not supplied to the internal blocks in the S3C2440X. Figure 7-5 shows the timing diagram.

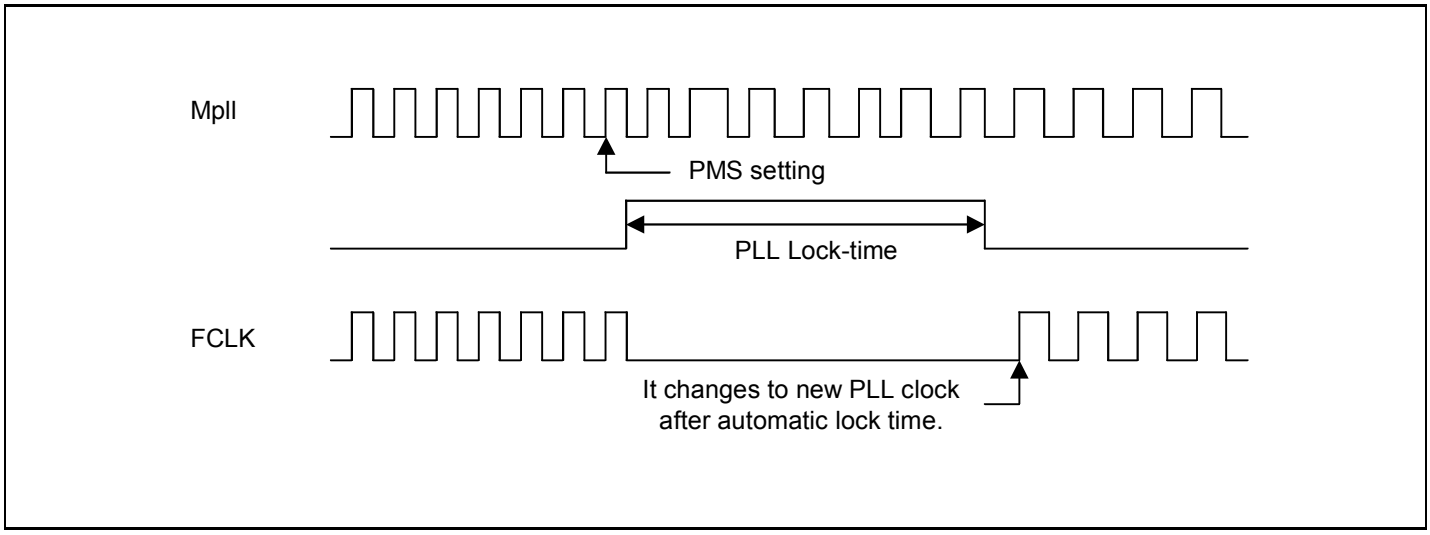



**Figure 7-5. Changing Slow Clock by Setting PMS Value**

### USB Clock Control

USB host interface and USB device interface needs 48Mhz clock. In the S3C2440X, the USB dedicated PLL (UPLL) generates 48Mhz for USB. UCLK does not fed until the PLL (UPLL) is configured.

| Condition | UCLK State | UPLL State |
|---|---|---|
| After reset | XTIpll or EXTCLK | On |
| After configuring UPLL | L : during PLL lock time<br>48MHz: after PLL lock time | On |
| UPLL is turned off by CLKSLOW register | XTIpll or EXTCLK | Off |
| UPLL is turned on by CLKSLOW register | 48MHz | On |

### FCLK, HCLK, and PCLK

FCLK is used by ARM920T.
HCLK is used for AHB bus, which is used by the ARM920T, the memory controller, the interrupt controller, the LCD controller, the DMA and USB host block.
PCLK is used for APB bus, which is used by the peripherals such as WDT, IIS, I2C, PWM timer, MMC interface, ADC, UART, GPIO, RTC and SPI.

The S3C2440X supports selection of Dividing Ratio between FCLK, HLCK and PCLK. This ratio is determined by HDIVN and PDIVN of CLKDIVN control register.

| HDIVN | PDIVN | FCLK | HCLK | PCLK | Divide Ratio |
|---|---|---|---|---|---|
| 0 | 0 | FCLK | FCLK | FCLK | 1 : 1 : 1 (Default) |
| 0 | 1 | FCLK | FCLK | FCLK / 2 | 1 : 1 : 2 |
| 1 | 0 | FCLK | FCLK / 2 | FCLK / 2 | 1 : 2 : 2 |
| 1 | 1 | FCLK | FCLK / 2 | FCLK / 4 | 1 : 2 : 4 |
| 3 | 0 | FCLK | FCLK / 3 | FCLK / 3 | 1 : 3 : 3 |
| 3 | 1 | FCLK | FCLK / 3 | FCLK / 6 | 1 : 3 : 6 |
| 2 | 0 | FCLK | FCLK / 4 | FCLK / 4 | 1 : 4 : 4 |
| 2 | 1 | FCLK | FCLK / 4 | FCLK / 8 | 1 : 4 : 8 |

After setting PMS value, it is required to set CLKDIVN register. The value set for CLKDIVN will be valid after PLL lock time. The value is also available for reset and changing Power Management Mode.
The setting value can also be valid after 1.5 HCLK. Only, 1HCLK can validate the value of CLKDIVN register changed from Default (1:1:1) to other Divide Ratio (1:1:2, 1:2:2 and 1:2:4)

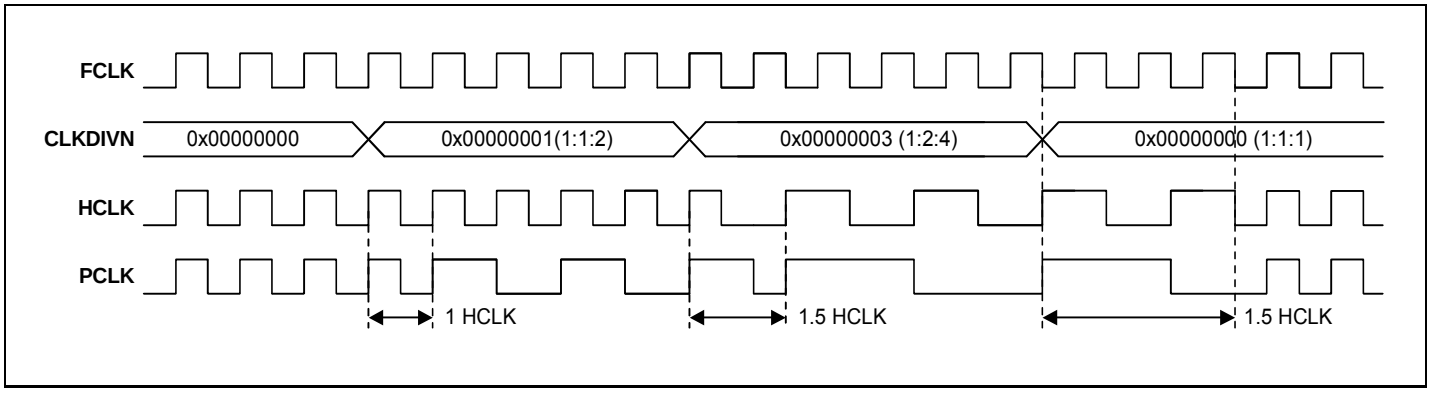



**Figure 7-6. Changing CLKDIVN Register Value**

**NOTE**

1. CLKDIVN should be set carefully not to exceed the limit of HCLK and PCLK.
2. If HDIVN is not 0, the CPU bus mode has to be changed from the fast bus mode to the asynchronous bus mode using following instructions.

   MMU_SetAsyncBusMode

   mrc p15,0,r0,c1,c0,0

   orr r0,r0,#R1_nF:OR:R1_iA

   mcr p15,0,r0,c1,c0,0

   If HDIVN is not 0 and the CPU bus mode is the fast bus mode, the CPU will operate by the HCLK. This feature can be used to change the CPU frequency as a half or more without affecting the HCLK and PCLK.

## POWER MANAGEMENT

The Power Management block controls the system clocks by software for the reduction of power consumption in the S3C2440X. These schemes are related to PLL, clock control logics (FCLK, HCLK, and PCLK) and wakeup signals. Figure 7-7 shows the clock distribution of the S3C2440X.

The S3C2440X has four power modes. The following section describes each power management mode. The transition between the modes is not allowed freely. Please see Figure 7-8 for available transitions among the modes.

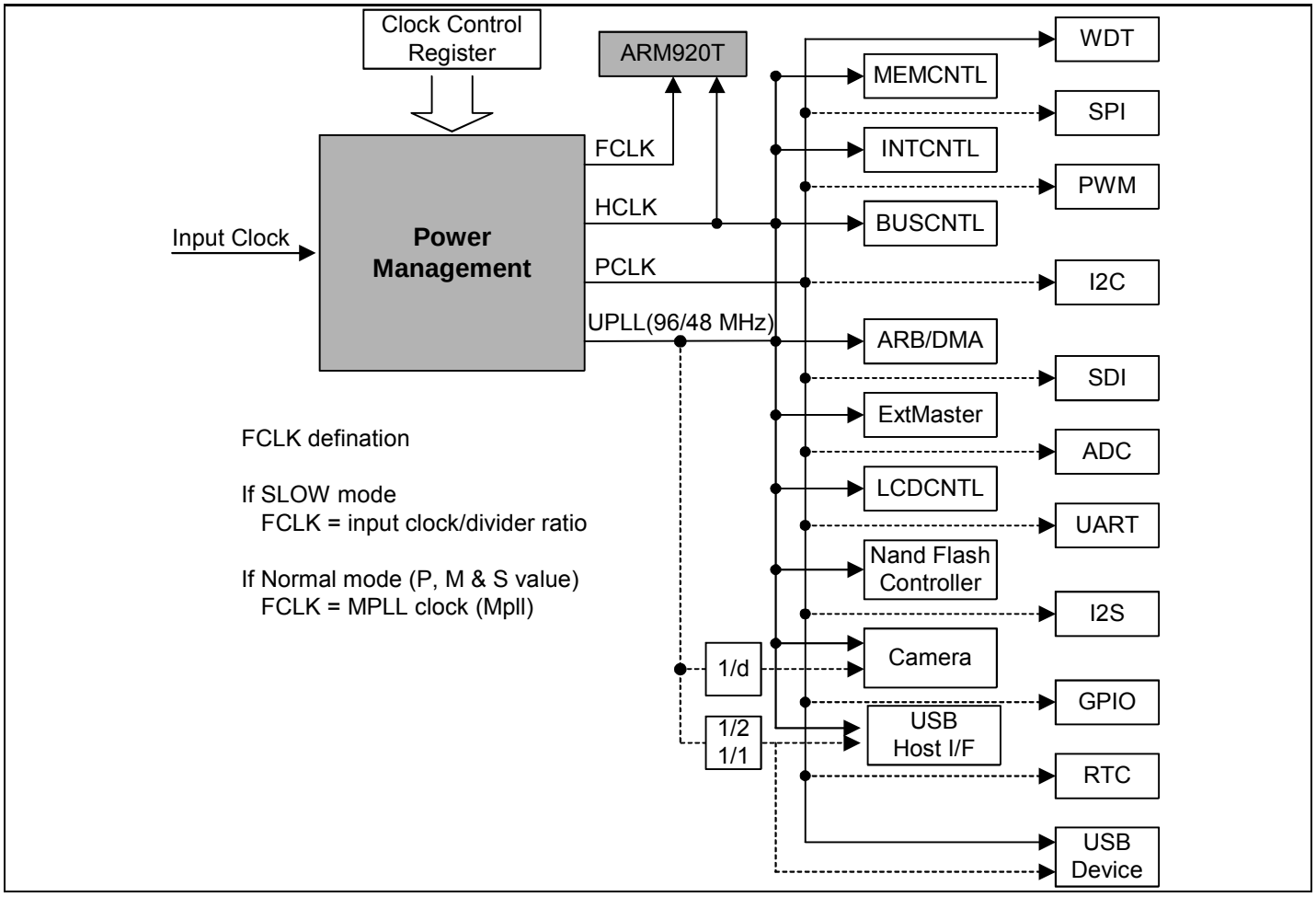



**Figure 7-7. The Clock Distribution Block Diagram**

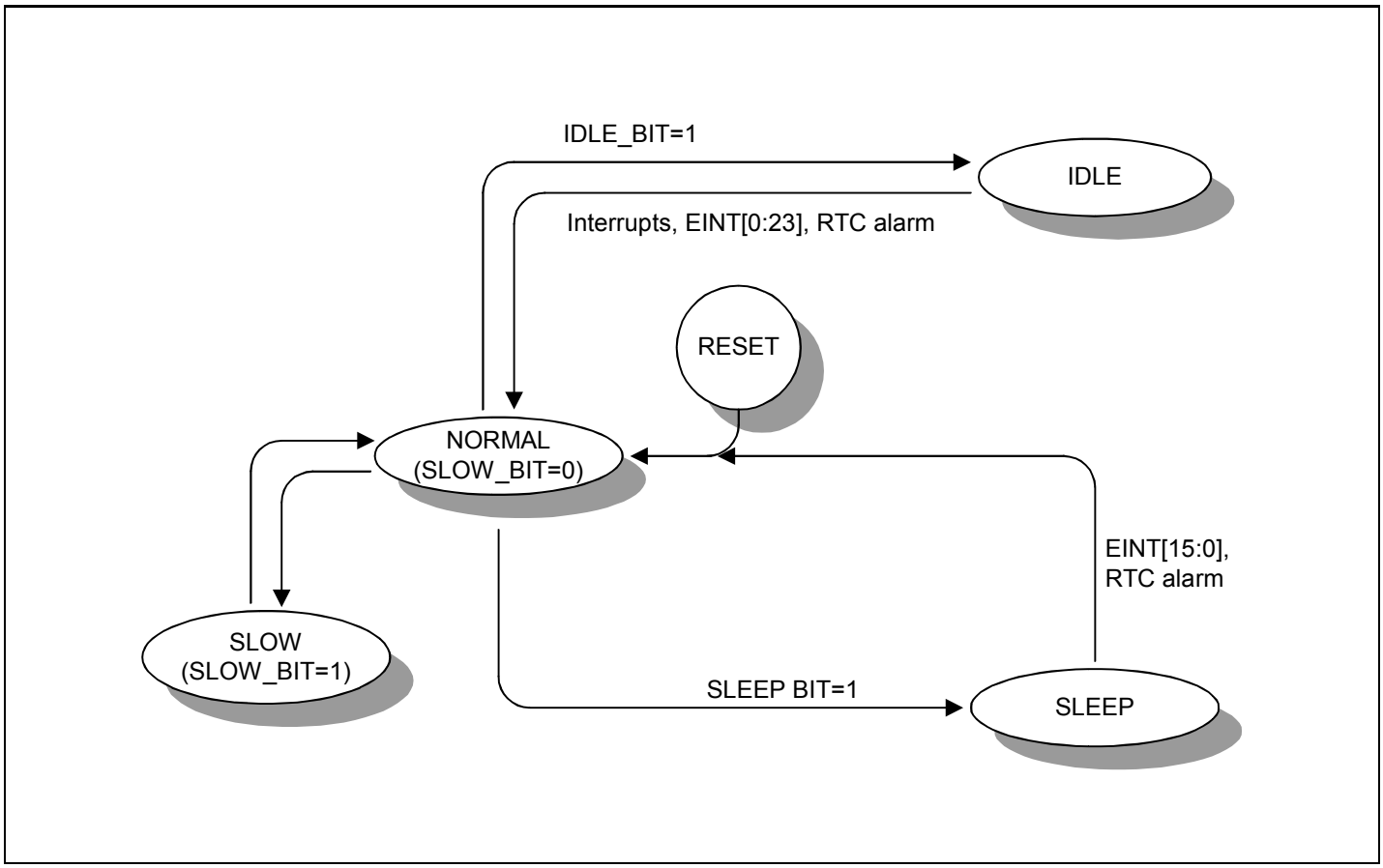



**Figure 7-8. Power Management State Diagram**

**Table 7-2. Clock and Power State in Each Power Mode**

| Mode | ARM920T | AHB Modules (1) /WDT | Power Management | GPIO | 32.768kHz RTC clock | APB Modules (2) & USBH/LCD/NAND |
|---|---|---|---|---|---|---|
| **NORMAL** | O | O | O | SEL | O | SEL |
| **IDLE** | X | O | O | SEL | O | SEL |
| **SLOW** | O | O | O | SEL | O | SEL |
| **SLEEP** | OFF | OFF | Wait for wake-up event | Previous state | O | OFF |

**NOTES:**

1. USB host,LCD, and NAND are excluded.
2. WDT is excluded. RTC interface for CPU access is included.
3. SEL : selectable(O,X), O : enable , X : disable OFF: power is turned off

### NORMAL Mode

In Normal mode, all peripherals and the basic blocks including power management block, the CPU core, the bus controller, the memory controller, the interrupt controller, DMA, and the external master may operate fully. But, the clock to each peripheral, except the basic blocks, can be stopped selectively by software to reduce the power consumption.

### IDLE Mode

In IDLE mode, the clock to the CPU core is stopped except the bus controller, the memory controller, the interrupt controller, and the power management block. To exit the IDLE mode, EINT[23:0], or RTC alarm interrupt, or the other interrupts should be activated. (EINT is not available until GPIO block is turned on).

### SLOW Mode (Non-PLL Mode)

Power consumption can be reduced in the SLOW mode by applying a slow clock and excluding the power consumption from the PLL. The FCLK is the frequency of divide_by_n of the input clock (XTIpll or EXTCLK) without PLL. The divider ratio is determined by SLOW_VAL in the CLKSLOW control register and CLKDIVN control register.

**Table 7-3. CLKSLOW and CLKDIVN Register Settings for SLOW Clock example**

| SLOW_VAL | FCLK | HCLK | | PCLK | | UCLK |
|---|---|---|---|---|---|---|
| | | 1/1 Option (HDIVN=0) | 1/2 Option (HDIVN=1) | 1/1 Option (PDIVN=0) | 1/2 Option (PDIVN=1) | |
| 0 0 0 | EXTCLK or XTIpll / 1 | EXTCLK or XTIpll / 1 | EXTCLK or XTIpll / 2 | HCLK | HCLK / 2 | 48 MHz |
| 0 0 1 | EXTCLK or XTIpll / 2 | EXTCLK or XTIpll / 2 | EXTCLK or XTIpll / 4 | HCLK | HCLK / 2 | 48 MHz |
| 0 1 0 | EXTCLK or XTIpll / 4 | EXTCLK or XTIpll / 4 | EXTCLK or XTIpll / 8 | HCLK | HCLK / 2 | 48 MHz |
| 0 1 1 | EXTCLK or XTIpll / 6 | EXTCLK or XTIpll / 6 | EXTCLK or XTIpll / 12 | HCLK | HCLK / 2 | 48 MHz |
| 1 0 0 | EXTCLK or XTIpll / 8 | EXTCLK or XTIpll / 8 | EXTCLK or XTIpll / 16 | HCLK | HCLK / 2 | 48 MHz |
| 1 0 1 | EXTCLK or XTIpll / 10 | EXTCLK or XTIpll / 10 | EXTCLK or XTIpll / 20 | HCLK | HCLK / 2 | 48 MHz |
| 1 1 0 | EXTCLK or XTIpll / 12 | EXTCLK or XTIpll / 12 | EXTCLK or XTIpll / 24 | HCLK | HCLK / 2 | 48 MHz |
| 1 1 1 | EXTCLK or XTIpll / 14 | EXTCLK or XTIpll / 14 | EXTCLK or XTIpll / 28 | HCLK | HCLK / 2 | 48 MHz |

In SLOW mode, PLL will be turned off to reduce the PLL power consumption. When the PLL is turned off in the SLOW mode and the user changes power mode from SLOW mode to NORMAL mode, the PLL needs clock stabilization time (PLL lock time). This PLL stabilization time is automatically inserted by the internal logic with lock time count register. The PLL stability time will take 300us after the PLL is turned on. During PLL lock time, the FCLK becomes SLOW clock.

Users can change the frequency by enabling SLOW mode bit in CLKSLOW register in PLL on state. The SLOW clock is generated during the SLOW mode. Figure 7-11 shows the timing diagram.

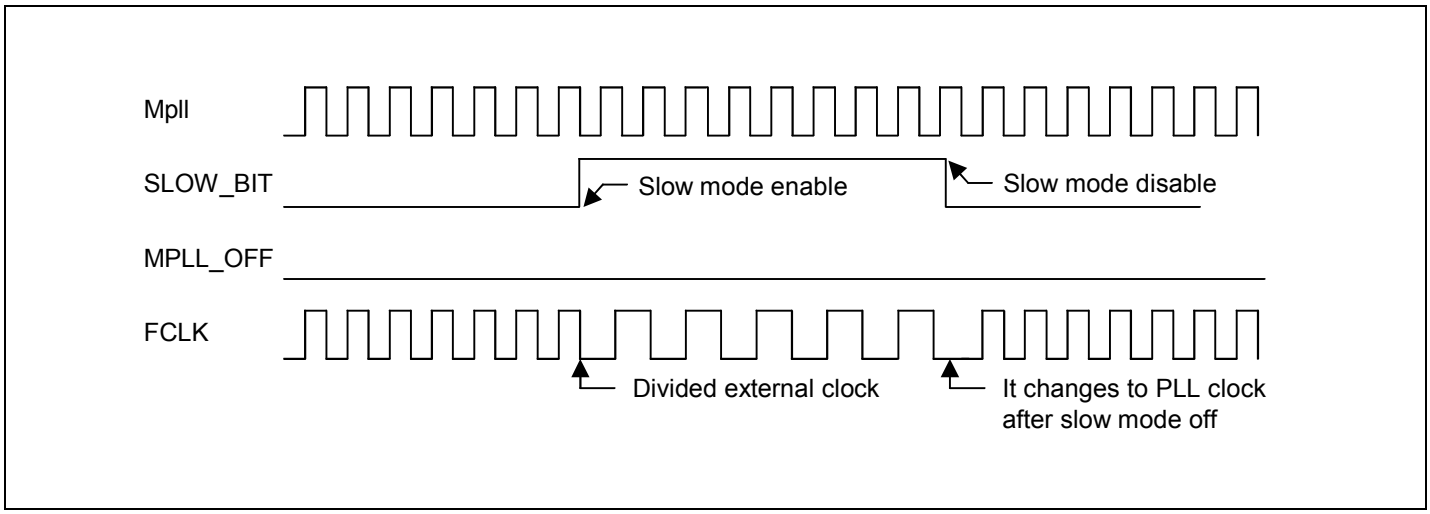



**Figure 7-9. Issuing Exit_from_Slow_mode Command in PLL on State**

If the user switches from SLOW mode to Normal mode by disabling the SLOW_BIT in the CLKSLOW register after PLL lock time, the frequency is changed just after SLOW mode is disabled. Figure 7-12 shows the timing diagram.

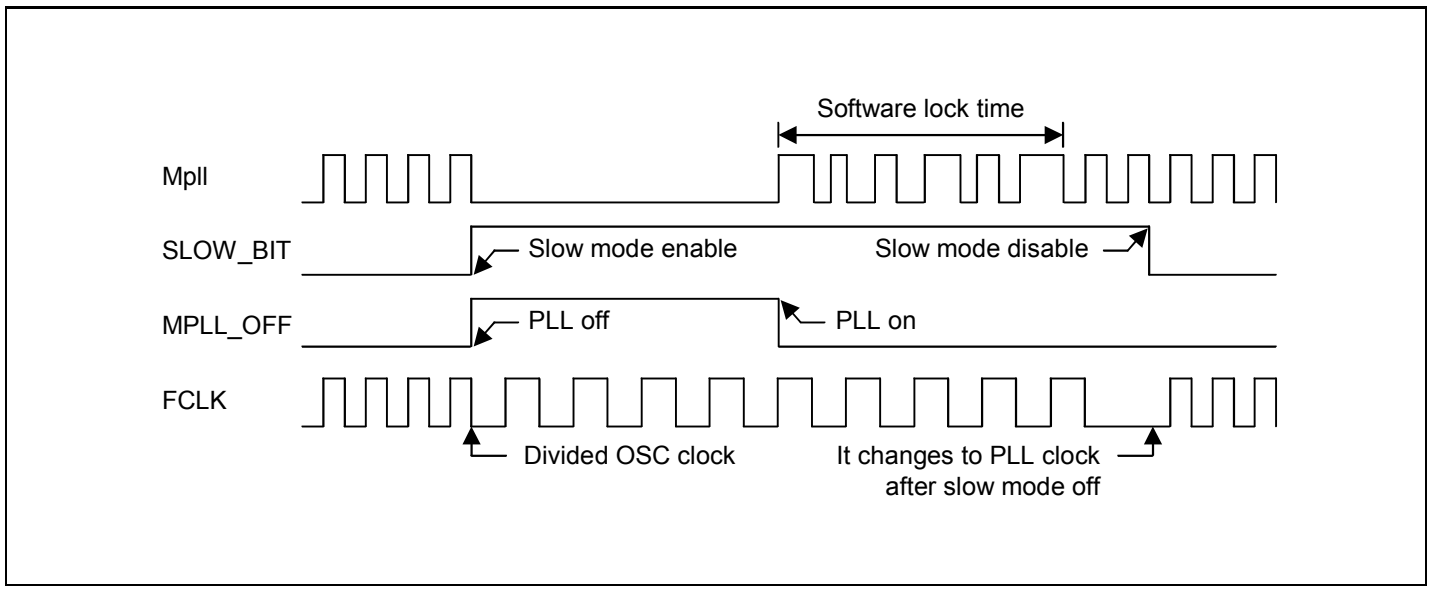



**Figure 7-10. Issuing Exit_from_Slow_mode Command After Lock Time**

If the user switches from SLOW mode to Normal mode by disabling SLOW_BIT and MPLL_OFF bit simultaneously in the CLKSLOW register, the frequency is changed just after the PLL lock time. Figure 7-13 shows the timing diagram.

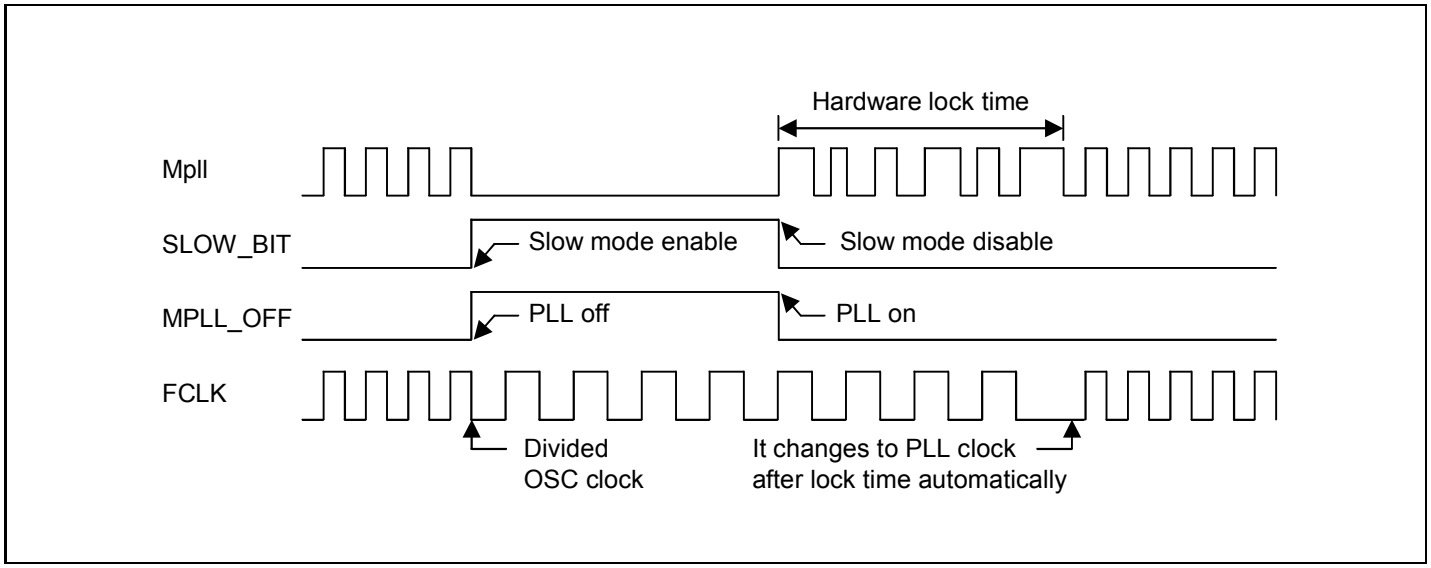



**Figure 7-11. Issuing Exit_from_Slow_mode Command and the Instant PLL_on Command Simultaneously**

### SLEEP Mode

The block disconnects the internal power. So, there occurs no power consumption due to CPU and the internal logic except the wake-up logic in this mode. Activating the SLEEP mode requires two independent power sources. One of the two power sources supplies the power for the wake-up logic. The other one supplies other internal logics including CPU, and should be controlled for power on/off. In the SLEEP mode, the second power supply source for the CPU and internal logics will be turned off. The wakeup from SLEEP mode can be issued by the EINT[15:0] or by RTC alarm interrupt.

**Follow the Procedure to Enter SLEEP mode**

1. Set the GPIO configuration adequate for SLEEP mode.
2. Mask all interrupts in the INTMSK register.
3. Configure the wake-up sources properly including RTC alarm. (The bit of EINTMASK corresponding to the wake-up source has not to be masked in order to let the corresponding bit of SRCPND or EINTPEND set. Although a wake-up source is issued and the corresponding bit of EINTMASK is masked, the wake-up will occur and the corresponding bit of SRCPND or EINTPEND will not be set.)
4. Set USB pads as suspend mode. (MISCCR[13:12]=11b)
5. Save some meaning values into GSTATUS[4:3] register. These register are preserved during SLEEP mode.
6. Configure MISCCR[1:0] for the pull-up resisters on the data bus,D[31:0]. If there is an external BUS holder, such as 74LVCH162245, turn off the pull-up resistors. If not, turn on the pull-up resistors.
   Additionally, The Memory concerning pins are set to two type, one is Hi-z, and the other is Inactive state.
7. Stop LCD by clearing LCDCON1.ENVID bit.
8. Read rREFRESH and rCLKCON registers in order to fill the TLB.
9. Let SDRAM enter the self-refresh mode by setting the REFRESH[22]=1b.
10. Wait until SDRAM self-refresh is effective.
11. Set MISCCR[19:17]=111b to make SDRAM signals(SCLK0,SCLK1 and SCKE) protected during SLEEP mode
12. Set the SLEEP mode bit in the CLKCON register.

**Follow the Procedure to Wake-up from SLEEP mode**

1. The internal reset signal will be asserted if one of the wake-up sources is issued. It's exactly same with the case of the assertion of the external nRESET pin. This reset duration is determined by the internal 16-bit counter logic and the reset assertion time is calculated as tRST = (65535 / XTAL_frequency).
2. Check GSTATUS2[2] in order to know whether or not the power-up is caused by the wake-up from SLEEP mode.
3. Release the SDRAM signal protection by setting MISCCR[19:17]=000b.
4. Configure the SDRAM memory controller.
5. Wait until the SDRAM self-refresh is released. Mostly SDRAM needs the refresh cycle of all SDRAM row.
6. The information in GSTATUS[3:4] can be used for user's own purpose because the value in GSTATUS[3:4] has been preserved during SLEEP mode.
7. – For EINT[3:0], check the SRCPND register.
   – For EINT[15:4], check the EINTPEND instead of SRCPND (SRCPND will not be set although some bits of EINTPEND are set.).
   – For alarm wake-up, check the RTC time because the RTC bit of SRCPND isn't set at the alarm wake-up.
   – If there was the nBATT_FLT assertion during SLEEP mode, the corresponding bit of SRCPND has been set.

**Pin States in SLEEP Mode**

The pin state of the SLEEP mode is as follows;

| Pin Type | Pin Example | Pin States in SLEEP Mode |
|---|---|---|
| GPIO output pin | GPB0:input | Output ( GPIO data register value is used.) |
| GPIO input pin | GPB0:output | Input |
| GPIO bi-directional pin | GPG6:SPIMOSI | Input |
| Function output pin | nGCS0 | Output (the last output level is held.) |
| Function input pin | nWAIT | Input |

### Power Control of VDDi and VDDiarm

In SLEEP mode, only VDDi and VDDiarm will be turned off, which is controlled by PWREN pin.

If PWREN signal is active(H), VDDi and VDDiarm are supplied by an external voltage regulator. If PWREN pin is inactive (L), the VDDi and VDDiarm are turned off.

**NOTE**

Although VDDi and VDDiarm may be turned off, the other power pins have to be supplied.

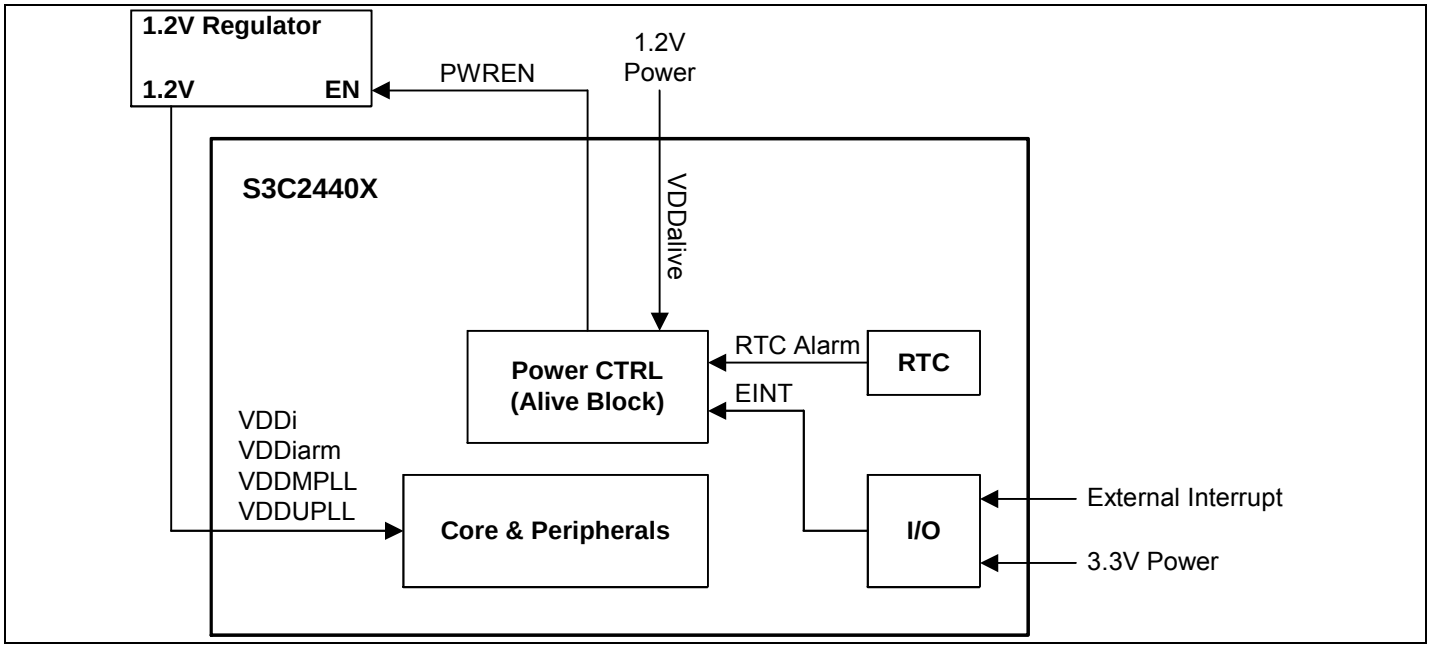



**Figure 7-12. SLEEP Mode**

### Signaling EINT[15:0] for Wakeup

The S3C2440X can be woken up from SLEEP mode only if the following conditions are met.

a) Level signals (H or L) or edge signals (rising or falling or both) are asserted on EINTn input pin.

b) The EINTn pin has to be configured as EINT in the GPIO control register.

c) nBATT_FLT pin has to be H level. It is important to configure the EINTn in the GPIO control register as an external interrupt pins, considering the condition a) above.

Just after the wake-up, the corresponding EINTn pin will not be used for wakeup. This means that the pin can be used as an external interrupt request pin again.

### Entering IDLE Mode

If CLKCON[2] is set to 1 to enter the IDLE mode, the S3C2440X will enter IDLE mode after some delay (until the power control logic receives ACK signal from the CPU wrapper).

### PLL On/Off

The PLL can only be turned off for low power consumption in slow mode. If the PLL is turned off in any other mode, MCU operation is not guaranteed.

When the processor is in SLOW mode and tries to change its state into other state with the PLL turned on, then SLOW_BIT should be clear to move to another state after PLL stabilization

### Pull-up Resistors on the Data Bus and SLEEP Mode

In SLEEP mode, the data bus (D[31:0] or D[15:0] ) is in Hi-z state.

But, because of the characteristics of I/O pad, the data bus pull-up resistors have to be turned on for low power consumption in SLEEP mode. D[31:0] pin pull-up resistors can be controlled by the GPIO control register (MISCCR). However, if there is an external bus holder, such as 74LVCH162245, on the data bus, turning off the data bus pull-up resistors will be reduce power consumption.

### Output Port State and SLEEP Mode

If output is L, the current will be consumed through the internal parasitic resistance; if the output is H, the current will not be consumed. For an output port, the current consumption can be reduced if the output state is H.

It is recommended that the output ports be in H state to reduce current consumption in SLEEP mode.

### Battery Fault Signal(nBATT_FLT)

There are two functions in nBATT_FLT pin as follows;

— When CPU is not in SLEEP mode, nBATT_FLT pin will cause the interrupt request. The interrupt attribute of the nBATT_FLT is L-level triggered.

— While CPU is in SLEEP mode, assertion of the nBATT_FLT will prohibit the wake up from the power-down mode. So, Any wake-up source will be masked if nBATT_FLT is asserted, which is protecting the system malfunction of the low battery capacity

### ADC Power Down

The ADC has an additional power-down bit in ADCCON. If the S3C2440X enters the SLEEP mode, the ADC should enter its own power-down mode.

# CLOCK GENERATOR & POWER MANAGEMENT SPECIAL REGISTER

## LOCK TIME COUNT REGISTER (LOCKTIME)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LOCKTIME | 0x4C000000 | R/W | PLL lock time count register | 0xFFFFFFFF |

| LOCKTIME | Bit | Description | Initial State |
|---|---|---|---|
| U_LTIME | [31:16] | UPLL lock time count value for UCLK.<br>(U_LTIME . 300uS) | 0xFFFF |
| M_LTIME | [15:0] | MPLL lock time count value for FCLK, HCLK, and PCLK<br>(M_LTIME . 300uS) | 0xFFFF |

## PLL Control Register (MPLLCON and UPLLCON)

Mpll = (m * Fin) / (p * $2^s$)
m = (MDIV + 8), p = (PDIV + 2), s = SDIV

## PLL Value Selection Guide (MPLLCON)

1. $F_{out}$ = m * Fin / (p*$2^s$), $F_{vco}$ = m * Fin / p where : m=MDIV+8, p=PDIV+2, s=SDIV
2. 200MHz ≤ $F_{vco}$ ≤ 500MHz
3. 100MHz ≤ $F_{out}$ ≤ 500MHz (The max. $F_{out}$ of the PLL itself is 500MHz)
4. FCLK ≥ 3X-tal or 3EXTCLK

**NOTE:** Although there is the rule for choosing PLL value, we recommend only the values in the PLL value recommendation table. If you have to use another value, please contact us.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| MPLLCON | 0x4C000004 | R/W | MPLL configuration register | 0x00096030 |
| UPLLCON | 0x4C000008 | R/W | UPLL configuration register | 0x0004d030 |

| PLLCON | Bit | Description | Initial State |
|---|---|---|---|
| MDIV | [19:12] | Main divider control | 0x96 / 0x4d |
| PDIV | [9:4] | Pre-divider control | 0x03 / 0x03 |
| SDIV | [1:0] | Post divider control | 0x0 / 0x0 |

**NOTE:** When you set MPLL&UPLL values simultaneously, set MPLL value first and then UPLL value.

## PLL VALUE SELECTION TABLE

It is not easy to find a proper PLL value. So, We recommend referring to the following PLL value recommendation table.

| Input Frequency | Output Frequency | MDIV | PDIV | SDIV |
|---|---|---|---|---|
| 12.00MHz | **48.00 MHz** (Note) | 56(0X38) | 2 | 2 |
| 12.00MHz | **96.00 MHz** (Note) | 56(0x38) | 2 | 1 |
| 12.00MHz | **101.60 MHz** | 246(0xf6) | 13(0xd) | 1 |
| 12.00MHz | **112.909 MHz** | 199(0xc7) | 9 | 1 |
| 12.00MHz | **118.50 MHz** | 150(0x96) | 6 | 1 |
| 12.00MHz | **124.20 MHz** | 199(0xc7) | 8 | 1 |
| 12.00MHz | **135.429 MHz** | 150(0x96) | 5 | 1 |
| 12.00MHz | **146.667 MHz** | 212(0xd4) | 7 | 1 |
| 12.00MHz | **152.400 MHz** | 119(0x77) | 3 | 1 |
| 12.00MHz | **158.00 MHz** | 71(0x47) | 1 | 1 |
| 12.00MHz | **169.333 MHz** | 246(0xf6) | 7 | 1 |
| 12.00MHz | **180.750 MHz** | 233(0xe9) | 6 | 1 |
| 12.00MHz | **186.00 MHz** | 85(0x55) | 1 | 1 |
| 12.00MHz | **192.00 MHz** | 88(0x58) | 1 | 1 |
| 12.00MHz | **203.20 MHz** | 246(0xf6) | 13(0xd) | 0 |
| 12.00MHz | **214.50 MHz** | 135(0x87) | 2 | 1 |
| 12.00MHz | **220.286 MHz** | 249(0xf9) | 12(0xc) | 0 |
| 12.00MHz | **225.818 MHz** | 199(0xc7) | 9 | 0 |
| 12.00MHz | **237.00 MHz** | 150(0x96) | 6 | 0 |
| 12.00MHz | **248.40 MHz** | 199(0xc7) | 8 | 0 |
| 12.00MHz | **254.00 MHz** | 119(0x77) | 4 | 0 |
| 12.00MHz | **259.636 MHz** | 230(0xe6) | 9 | 0 |
| 12.00MHz | **270.857 MHz** | 150(0x96) | 5 | 0 |
| 12.00MHz | **282.00 MHz** | 86(0x56) | 2 | 0 |
| 12.00MHz | **288.00 MHz** | 64(0x40) | 1 | 0 |
| 12.00MHz | **293.333 MHz** | 212(0xd4) | 7 | 0 |
| 12.00MHz | **304.80 MHZ** | 119(0x77) | 3 | 0 |
| 12.00MHz | **316.00 MHz** | 71(0x47) | 1 | 0 |
| 12.00MHz | **321.60 MHz** | 126(0x7e) | 3 | 0 |

## PLL VALUE SELECTION TABLE (Continued)

| Input Frequency | Output Frequency | MDIV | PDIV | SDIV |
|---|---|---|---|---|
| 12.00MHz | **327.429 MHz** | 183(0xb7) | 5 | 0 |
| 12.00MHz | **338.667 MHz** | 246(0xf6) | 7 | 0 |
| 12.00MHz | **350.00 MHz** | 167(0xa7) | 4 | 0 |
| 12.00MHz | **355.50 MHz** | 229(0xe5) | 6 | 0 |
| 12.00MHz | **361.50 MHz** | 233(0xe9) | 6 | 0 |
| 12.00MHz | **372.00 MHz** | 85(0x55) | 1 | 0 |
| 12.00MHz | **384.00 MHz** | 88(0x58) | 1 | 0 |
| 12.00MHz | **389.143 MHz** | 219(0xdb) | 5 | 0 |
| 12.00MHz | **400.00 MHz** | 92(0x5c) | 1 | 0 |

**NOTE:** The 48.00MHz and 96MHz output is used for UPLLCON register.

## CLOCK CONTROL REGISTER (CLKCON)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| CLKCON | 0x4C00000C | R/W | Clock generator control register | 0xFFFF0 |

| CLKCON | Bit | Description | Initial State |
|---|---|---|---|
| Camera | [19] | Control HCLK into Camera block.<br>0 = Disable, 1 = Enable | 1 |
| SPI | [18] | Control PCLK into SPI block.<br>0 = Disable, 1 = Enable | 1 |
| IIS | [17] | Control PCLK into IIS block.<br>0 = Disable, 1 = Enable | 1 |
| IIC | [16] | Control PCLK into IIC block.<br>0 = Disable, 1 = Enable | 1 |
| ADC(&Touch Screen) | [15] | Control PCLK into ADC block.<br>0 = Disable, 1 = Enable | 1 |
| RTC | [14] | Control PCLK into RTC control block.<br>Even if this bit is cleared to 0, RTC timer is alive.<br>0 = Disable, 1 = Enable | 1 |
| GPIO | [13] | Control PCLK into GPIO block.<br>0 = Disable, 1 = Enable | 1 |
| UART2 | [12] | Control PCLK into UART2 block.<br>0 = Disable, 1 = Enable | 1 |
| UART1 | [11] | Control PCLK into UART1 block.<br>0 = Disable, 1 = Enable | 1 |
| UART0 | [10] | Control PCLK into UART0 block.<br>0 = Disable, 1 = Enable | 1 |
| SDI | [9] | Control PCLK into SDI interface block.<br>0 = Disable, 1 = Enable | 1 |
| PWMTIMER | [8] | Control PCLK into PWMTIMER block.<br>0 = Disable, 1 = Enable | 1 |
| USB device | [7] | Control PCLK into USB device block.<br>0 = Disable, 1 = Enable | 1 |
| USB host | [6] | Control HCLK into USB host block.<br>0 = Disable, 1 = Enable | 1 |
| LCDC | [5] | Control HCLK into LCDC block.<br>0 = Disable, 1 = Enable | 1 |
| NAND Flash Controller | [4] | Control HCLK into NAND Flash Controller block.<br>0 = Disable, 1 = Enable | 1 |
| SLEEP | [3] | Control SLEEP mode of S3C2440X.<br>0 = Disable, 1 = Transition to SLEEP mode | 0 |
| IDLE BIT | [2] | Enter IDLE mode. This bit is not cleared automatically.<br>0 = Disable, 1 = Transition to IDLE mode | 0 |
| Reserved | [1] | Reserved | 0 |
| Reserved | [0] | Reserved | 0 |

## CLOCK SLOW CONTROL (CLKSLOW) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| CLKSLOW | 0x4C000010 | R/W | Slow clock control register | 0x00000004 |

| CLKSLOW | Bit | Description | Initial State |
|---|---|---|---|
| UCLK_ON | [7] | 0: UCLK ON (UPLL is also turned on and the UPLL lock time is inserted automatically.)<br>1: UCLK OFF (UPLL is also turned off.) | 0 |
| Reserved | [6] | Reserved | – |
| MPLL_OFF | [5] | 0: Turn on PLL.<br>After PLL stabilization time (minimum 300us), SLOW_BIT can be cleared to 0.<br>1: Turn off PLL.<br>PLL is turned off only when SLOW_BIT is 1. | 0 |
| SLOW_BIT | [4] | 0 : FCLK = Mpll (MPLL output)<br>1: SLOW mode<br>FCLK = input clock/(2xSLOW_VAL), when SLOW_VAL>0<br>FCLK = input clock, when SLOW_VAL=0.<br>Input clock = XTIpll or EXTCLK | 0 |
| Reserved | [3] | – | – |
| SLOW_VAL | [2:0] | The divider value for the slow clock when SLOW_BIT is on. | 0x4 |

### CLOCK DIVIDER CONTROL (CLKDIVN) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| CLKDIVN | 0x4C000014 | R/W | Clock divider control register | 0x00000000 |

| CLKDIVN | Bit | Description | Initial State |
|---|---|---|---|
| DIVN_UPLL | [3] | UCLK select register(UCLK must be 48MHz for USB)<br>0: UCLK = UPLL clock<br>1: UCLK = UPLL clock / 2<br>Set to 0, when UPLL clock is set as 48Mhz<br>Set to 1. when UPLL clock is set as 96Mhz. | 0 |
| HDIVN | [2:1] | 00: HCLK has the clock same as the FCLK/1.<br>01: HCLK has the clock same as the FCLK/2.<br>10: HCLK has the clock same as the **FCLK/4.**<br>11: HCLK has the clock same as the **FCLK/3.** | 0 |
| PDIVN | [0] | 0: PCLK has the clock same as the HCLK/1.<br>1: PCLK has the clock same as the HCLK/2. | 0 |

### CAMERA CLOCK DIVIDER (CAMDIVN) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| CAMDIVN | 0x4C000018 | R/W | Camera clock divider register | 0x00000000 |

| CAMDIVN | Bit | Description | Initial State |
|---|---|---|---|
| CAMCLK_SEL | [4] | 0:Use CAMCLK with UPLL output(CAMCLK=UPLL output).<br>1:CAMCLK is divided by CAMCLK_DIV value. | 0 |
| CAMCLK_DIV | [3:0] | CAMCLK divide factor setting register(0 – 15).<br>Camera clock = UPLL / [(CAMCLK_DIV +1)x2].<br>This bit is valid when CAMCLK_SEL=1. | 0 |

# 8 DMA

## OVERVIEW

The S3C2440X supports four-channel DMA controller that is located between the system bus and the peripheral bus. Each channel of DMA controller can perform data movements between devices in the system bus and/or peripheral bus with no restrictions. In other words, each channel can handle the following four cases: 1) both source and destination are in the system bus, 2) the source is in the system bus while the destination is in the peripheral bus, 3) the source is in the peripheral bus while the destination is in the system bus, and 4) both source and destination are in the peripheral bus.

The main advantage of the DMA is that it can transfer the data without CPU intervention. The operation of DMA can be initiated by software, or requests from internal peripherals or external request pins.

## DMA REQUEST SOURCES

Each channel of the DMA controller can select one of DMA request source among four DMA sources if H/W DMA request mode is selected by DCON register. (Note that if S/W request mode is selected, this DMA request sources have no meaning at all.) Table 8-1 shows four DMA sources for each channel.

| | Source0 | Source1 | Source2 | Source3 | Source4 |
|---|---|---|---|---|---|
| Ch-0 | nXDREQ0 | UART0 | SDI | Timer | USB device EP1 |
| Ch-1 | nXDREQ1 | UART1 | I2SSDI | SPI0 | USB device EP2 |
| Ch-2 | I2SSDO | I2SSDI | SDI | Timer | USB device EP3 |
| Ch-3 | UART2 | SDI | SPI1 | Timer | USB device EP4 |

**Table 8-1. DMA Request Sources for Each Channel**

Here, nXDREQ0 and nXDREQ1 represent two external sources(External Devices), and I2SSDO and I2SSDI represent IIS transmitting and receiving, respectively.

## DMA OPERATION

DMA uses three-state FSM (Finite State Machine) for its operation, which is described in the three following steps:

State-1. As an initial state, the DMA waits for a DMA request. If it comes, it goes to state-2. At this state, DMA ACK and INT REQ are 0.

State-2. In this state, DMA ACK becomes 1 and the counter (CURR_TC) is loaded from DCON[19:0] register. Note that the DMA ACK remains 1 until it is cleared later.

State-3. In this state, sub-FSM handling the atomic operation of DMA is initiated. The sub-FSM reads the data from the source address and then writes it to destination address. In this operation, data size and transfer size (single or burst) are considered. This operation is repeated until the counter (CURR_TC) becomes 0 in Whole service mode, while performed only once in Single service mode. The main FSM (this FSM) counts down the CURR_TC when the sub-FSM finishes each of atomic operation. In addition, this main FSM asserts the INT REQ signal when CURR_TC becomes 0 and the interrupt setting of DCON[29] register is set to 1. In addition, it clears DMA ACK if one of the following conditions is met.

1) CURR_TC becomes 0 in the Whole service mode
2) Atomic operation finishes in the Single service mode.

Note that in the Single service mode, these three states of main FSM are performed and then stops, and waits for another DMA REQ. And if DMA REQ comes in, all three states are repeated. Therefore, DMA ACK is asserted and then deasserted for each atomic transfer. In contrast, in the Whole service mode, main FSM waits at state-3 until CURR_TC becomes 0. Therefore, DMA ACK is asserted during all the transfers and then deasserted when TC reaches 0.

However, INT REQ is asserted only if CURR_TC becomes 0 regardless of the service mode (Single service mode or Whole service mode).

## EXTERNAL DMA DREQ/DACK PROTOCOL

There are three types of external DMA request/acknowledge protocols (Single service Demand, Single service Handshake and Whole service Handshake mode). Each type defines how the signals like DMA request and acknowledge are related to these protocols.

### Basic DMA Timing

The DMA service means performing paired Reads and Writes cycles during DMA operation, which can make one DMA operation. Figure 8-1 shows the basic Timing in the DMA operation of the S3C2440X.

- The setup time and the delay time of XnXDREQ and XnXDACK are the same in all the modes.
- If the completion of XnXDREQ meets its setup time, it is synchronized twice and then XnXDACK is asserted.
- After assertion of XnXDACK, DMA requests the bus and if it gets the bus it performs its operations. XnXDACK is deasserted when DMA operation is completed.

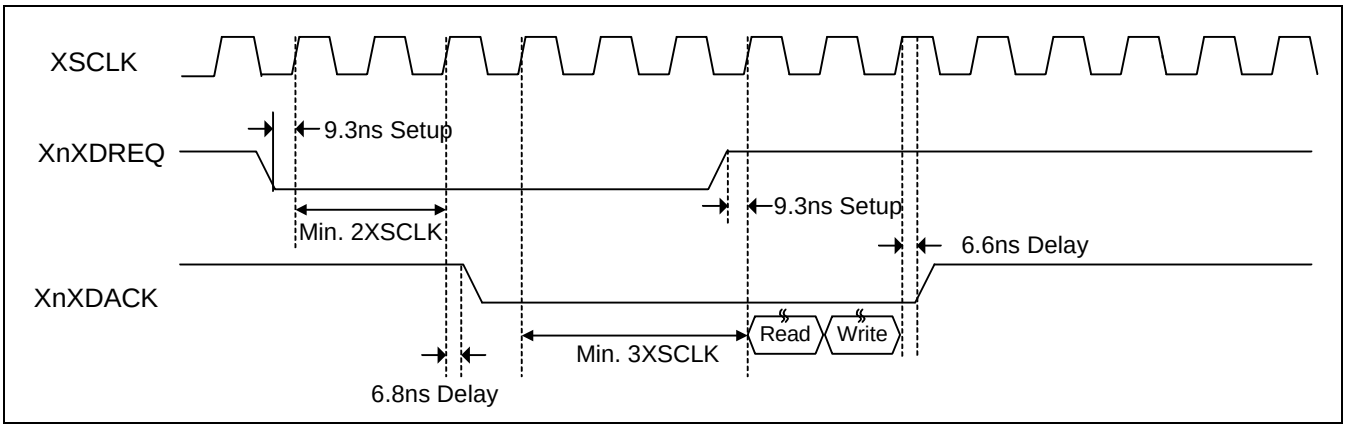



**Figure 8-1. Basic DMA Timing Diagram**

### Demand/Handshake Mode Comparison

Demand and Handshake modes are related to the protocol between XnXDREQ and XnXDACK. Figure 8-2 shows the differences between the two modes.

At the end of one transfer (Single/Burst transfer), DMA checks the state of double-synched XnXDREQ.

**Demand mode**

- If XnXDREQ remains asserted, the next transfer starts immediately. Otherwise it waits for XnXDREQ to be asserted.

**Handshake mode**

- If XnXDREQ is deasserted, DMA deasserts XnXDACK in 2cycles. Otherwise it waits until XnXDREQ is deasserted.

  Caution: XnXDREQ has to be asserted (low) only after the deassertion (high) of XnXDACK.

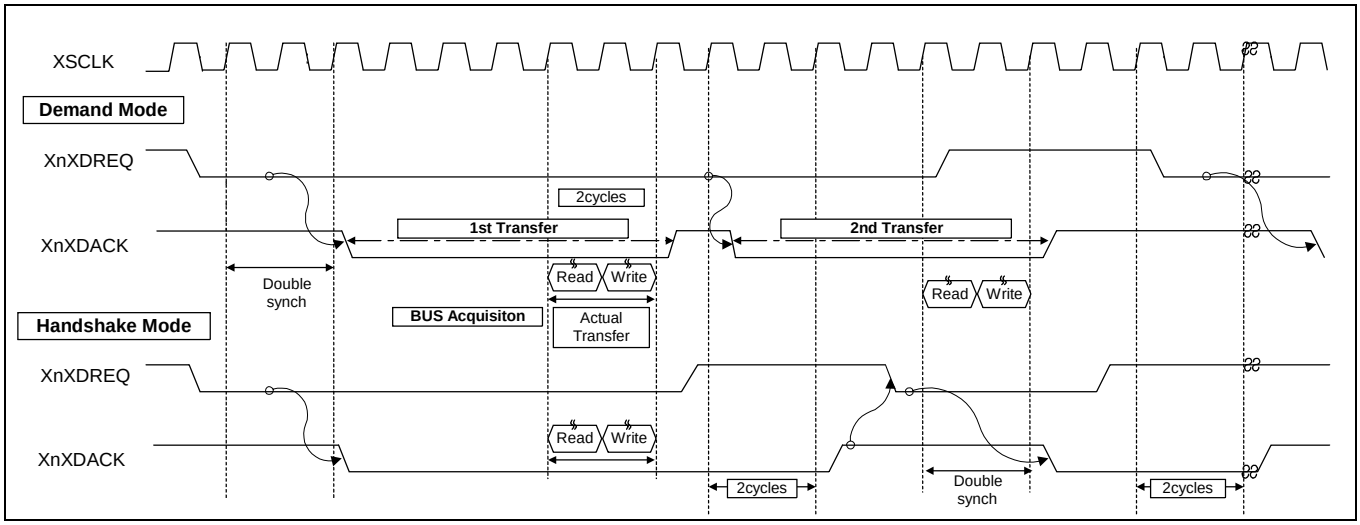



**Figure 8-2. Demand/Handshake Mode Comparison**

## Transfer Size

- There are two different transfer sizes; unit and Burst 4.
- DMA holds the bus firmly during the transfer of the chunk of data. Thus, other bus masters cannot get the bus.

### Burst 4 Transfer Size

There will be four sequential Reads and Writes respectively are performed in the Burst 4 Transfer.

* **Note**: Unit Transfer size: One read and one write are performed.

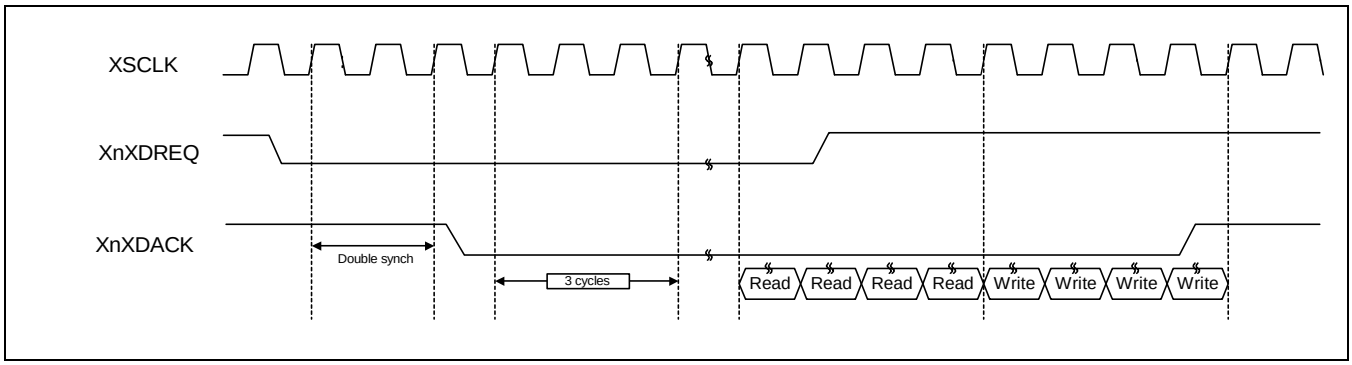



**Figure 8-3. Burst 4 Transfer Size**

## EXAMPLES

### Single service in Demand Mode with Unit Transfer Size

The assertion of XnXDREQ will be a need for every unit transfer (Single service mode). The operation continues while the XnXDREQ is asserted (Demand mode), and one pair of Read and Write (Single transfer size) is performed.

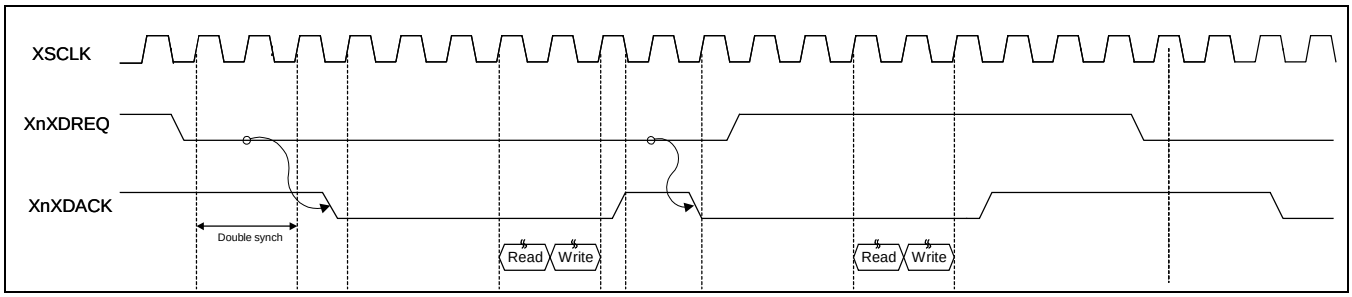



**Figure 8-4. Single service in Demand Mode with Unit Transfer Size**

### Single service in Handshake Mode with Unit Transfer Size

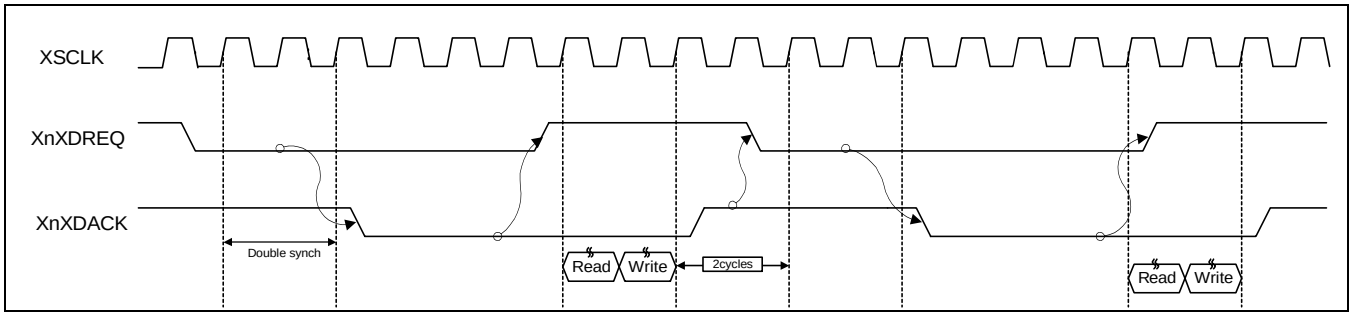



**Figure 8-5. Single service in Handshake Mode with Unit Transfer Size**

### Whole service in Handshake Mode with Unit Transfer Size

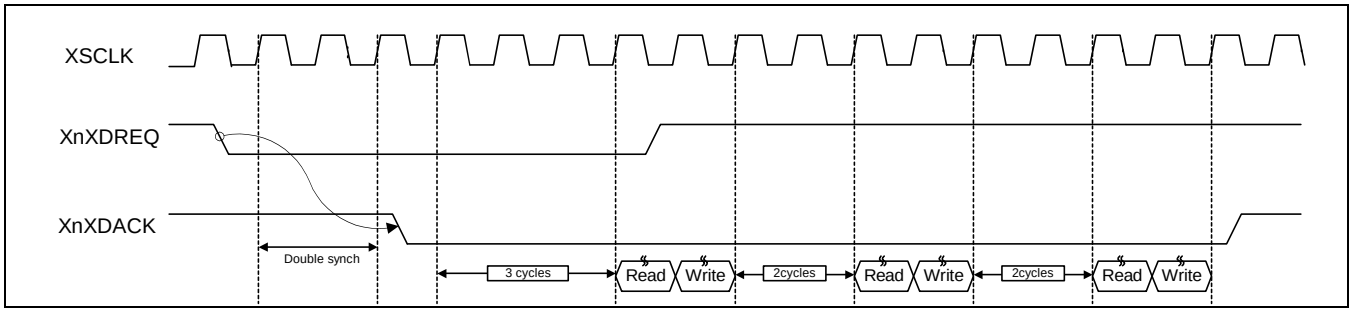



**Figure 8-6. Whole service in Handshake Mode with Unit Transfer Size**

## DMA SPECIAL REGISTERS

Each DMA channel has nine control registers (36 in total since there are four channels for DMA controller). Six of the control registers control the DMA transfer, and other three ones monitor the status of DMA controller. The details of those registers are as follows.

### DMA INITIAL SOURCE (DISRC) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DISRC0 | 0x4B000000 | R/W | DMA 0 initial source register | 0x00000000 |
| DISRC1 | 0x4B000040 | R/W | DMA 1 initial source register | 0x00000000 |
| DISRC2 | 0x4B000080 | R/W | DMA 2 initial source register | 0x00000000 |
| DISRC3 | 0x4B0000C0 | R/W | DMA 3 initial source register | 0x00000000 |

| DISRCn | Bit | Description | Initial State |
|---|---|---|---|
| S_ADDR | [30:0] | Base address (start address) of source data to transfer. This bit value will be loaded into CURR_SRC only if the CURR_SRC is 0 and the DMA ACK is 1. | 0x00000000 |

### DMA INITIAL SOURCE CONTROL (DISRCC) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DISRCC0 | 0x4B000004 | R/W | DMA 0 initial source control register | 0x00000000 |
| DISRCC1 | 0x4B000044 | R/W | DMA 1 initial source control register | 0x00000000 |
| DISRCC2 | 0x4B000084 | R/W | DMA 2 initial source control register | 0x00000000 |
| DISRCC3 | 0x4B0000C4 | R/W | DMA 3 initial source control register | 0x00000000 |

| DISRCCn | Bit | Description | Initial State |
|---|---|---|---|
| LOC | [1] | Bit 1 is used to select the location of source.<br>0: the source is in the system bus (AHB).<br>1: the source is in the peripheral bus (APB). | 0 |
| INC | [0] | Bit 0 is used to select the address increment.<br>0 = Increment 1= Fixed<br>If it is 0, the address is increased by its data size after each transfer in burst and single transfer mode.<br>If it is 1, the address is not changed after the transfer. (In the burst mode, address is increased during the burst transfer, but the address is recovered to its first value after the transfer.) | 0 |

### DMA INITIAL DESTINATION (DIDST) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DIDST0 | 0x4B000008 | R/W | DMA 0 initial destination register | 0x00000000 |
| DIDST1 | 0x4B000048 | R/W | DMA 1 initial destination register | 0x00000000 |
| DIDST2 | 0x4B000088 | R/W | DMA 2 initial destination register | 0x00000000 |
| DIDST3 | 0x4B0000B8 | R/W | DMA 3 initial destination register | 0x00000000 |

| DIDSTn | Bit | Description | Initial State |
|---|---|---|---|
| D_ADDR | [30:0] | Base address (start address) of destination for the transfer. This bit value will be loaded into CURR_SRC only if the CURR_DST is 0 and the DMA ACK is 1. | 0x00000000 |

### DMA INITIAL DESTINATION CONTROL (DIDSTC) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DIDSTC0 | 0x4B00000C | R/W | DMA 0 initial destination control register | 0x00000000 |
| DIDSTC1 | 0x4B00004C | R/W | DMA 1 initial destination control register | 0x00000000 |
| DIDSTC2 | 0x4B00008C | R/W | DMA 2 initial destination control register | 0x00000000 |
| DIDSTC3 | 0x4B0000CC | R/W | DMA 3 initial destination control register | 0x00000000 |

| DIDSTCn | Bit | Description | Initial State |
|---|---|---|---|
| CHK_INT | [2] | Select interrupt occurrence time when auto reload is setting.<br>0 : Interrupt will occur when TC reaches 0.<br>1 : Interrupt will occur after autoreload is performed. | 0 |
| LOC | [1] | Bit 1 is used to select the location of destination.<br>0: the destination is in the system bus (AHB).<br>1: the destination is in the peripheral bus (APB). | 0 |
| INC | [0] | Bit 0 is used to select the address increment.<br>0 = Increment 1= Fixed<br>If it is 0, the address is increased by its data size after each transfer in burst and single transfer mode.<br>If it is 1, the address is not changed after the transfer. (In the burst mode, address is increased during the burst transfer, but the address is recovered to its first value after the transfer.) | 0 |

## DMA CONTROL (DCON) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DCON0 | 0x4B000010 | R/W | DMA 0 control register | 0x00000000 |
| DCON1 | 0x4B000050 | R/W | DMA 1 control register | 0x00000000 |
| DCON2 | 0x4B000090 | R/W | DMA 2 control register | 0x00000000 |
| DCON3 | 0x4B0000D0 | R/W | DMA 3 control register | 0x00000000 |

| DCONn | Bit | Description | Initial State |
|---|---|---|---|
| DMD_HS | [31] | Select one between Demand mode and Handshake mode.<br>0: Demand mode will be selected.<br>1: Handshake mode will be selected.<br>In both modes, DMA controller starts its transfer and asserts DACK for a given asserted DREQ. The difference between the two modes is whether it waits for the deasserted DACK or not. In the Handshake mode, DMA controller waits for the deasserted DREQ before starting a new transfer. If it finds the deasserted DREQ, it deasserts DACK and waits for another asserted DREQ. In contrast, in the Demand mode, DMA controller does not wait until the DREQ is deasserted. It just deasserts DACK and then starts another transfer if DREQ is asserted. We recommend using Handshake mode for external DMA request sources to prevent unintended starts of new transfers. | 0 |
| SYNC | [30] | Select DREQ/DACK synchronization.<br>0: DREQ and DACK are synchronized to PCLK (APB clock).<br>1: DREQ and DACK are synchronized to HCLK (AHB clock).<br>Therefore, for devices attached to AHB system bus, this bit has to be set to 1, while for those attached to APB system, it should be set to 0. For the devices attached to external systems, the user should select this bit depending on which the external system is synchronized with between AHB system and APB system. | 0 |
| INT | [29] | Enable/Disable the interrupt setting for CURR_TC (terminal count)<br>0: CURR_TC interrupt is disabled. The user has to view the transfer count in the status register (i.e. polling).<br>1: interrupt request is generated when all the transfer is done (i.e. CURR_TC becomes 0). | 0 |
| TSZ | [28] | Select the transfer size of an atomic transfer (i.e. transfer performed each time DMA owns the bus before releasing the bus).<br>0: a unit transfer is performed.<br>1: a burst transfer of length four is performed. | 0 |

| DCONn | Bit | Description | Initial State |
|---|---|---|---|
| SERVMODE | [27] | Select the service mode between Single service mode and Whole service mode.<br>0: Single service mode is selected in which after each atomic transfer (single or burst of length four) DMA stops and waits for another DMA request.<br>1: Whole service mode is selected in which one request gets atomic transfers to be repeated until the transfer count reaches to 0. In this mode, additional request are not required.<br>Note that even in the Whole service mode, DMA releases the bus after each atomic transfer and then tries to re-get the bus to prevent starving of other bus masters. | 0 |
| HWSRCSEL | [26:24] | Select DMA request source for each DMA.<br>DCON0: 000:nXDREQ0 001:UART0 010:SDI 011:Timer 100:USB device EP1<br>DCON1: 000:nXDREQ1 001:UART1 010:I2SSDI 011:SPI 100:USB device EP2<br>DCON2: 000:I2SSDO 001:I2SSDI 010:SDI 011:Timer 100:USB device EP3<br>DCON3: 000:UART2 001:SDI 010:SPI 011:Timer 100:USB device EP4<br>These bits control the 4-1 MUX to select the DMA request source of each DMA. These bits have meanings only if H/W request mode is selected by DCONn[23]. | 00 |
| SWHW_SEL | [23] | Select the DMA source between software (S/W request mode) and hardware (H/W request mode).<br>0: S/W request mode is selected and DMA is triggered by setting SW_TRIG bit of DMASKTRIG control register.<br>1: DMA source selected by bit[26:24] triggers the DMA operation. | 0 |
| RELOAD | [22] | Set the reload on/off option.<br>0: auto reload is performed when a current value of transfer count becomes 0 (i.e. all the required transfers are performed).<br>1: DMA channel (DMA REQ) is turned off when a current value of transfer count becomes 0. The channel on/off bit (DMASKTRIGn[1]) is set to 0 (DREQ off) to prevent unintended further start of new DMA operation. | 0 |
| DSZ | [21:20] | Data size to be transferred.<br>00 = Byte 01 = Half word<br>10 = Word 11 = reserved | 00 |
| TC | [19:0] | Initial transfer count (or transfer beat).<br>Note that the actual number of bytes that are transferred is computed by the following equation: DSZ x TSZ x TC. Where, DSZ, TSZ (1 or 4), and TC represent data size (DCONn[21:20]), transfer size (DCONn[28]), and initial transfer count, respectively.<br>This value will be loaded into CURR_TC only if the CURR_TC is 0 and the DMA ACK is 1. | 00000 |

### DMA STATUS (DSTAT) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DSTAT0 | 0x4B000014 | R | DMA 0 count register | 000000h |
| DSTAT1 | 0x4B000054 | R | DMA 1 count register | 000000h |
| DSTAT2 | 0x4B000094 | R | DMA 2 count register | 000000h |
| DSTAT3 | 0x4B0000D4 | R | DMA 3 count register | 000000h |

| DSTATn | Bit | Description | Initial State |
|---|---|---|---|
| STAT | [21:20] | Status of this DMA controller.<br>00: Indicates that DMA controller is ready for another DMA request.<br>01: Indicates that DMA controller is busy for transfers. | 00b |
| CURR_TC | [19:0] | Current value of transfer count.<br>Note that transfer count is initially set to the value of DCONn[19:0] register and decreased by one at the end of every atomic transfer. | 00000h |

### DMA CURRENT SOURCE (DCSRC) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DCSRC0 | 0x4B000018 | R | DMA 0 current Source Register | 0x00000000 |
| DCSRC1 | 0x4B000058 | R | DMA 1 current Source Register | 0x00000000 |
| DCSRC2 | 0x4B000098 | R | DMA 2 current Source Register | 0x00000000 |
| DCSRC3 | 0x4B0000D8 | R | DMA 3 current Source Register | 0x00000000 |

| DCSRCn | Bit | Description | Initial State |
|---|---|---|---|
| CURR_SRC | [30:0] | Current source address for DMAn | 0x00000000 |

### CURRENT DESTINATION (DCDST) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DCDST0 | 0x4B00001C | R | DMA 0 current destination register | 0x00000000 |
| DCDST1 | 0x4B00005C | R | DMA 1 current destination register | 0x00000000 |
| DCDST2 | 0x4B00009C | R | DMA 2 current destination register | 0x00000000 |
| DCDST3 | 0x4B0000DC | R | DMA 3 current destination register | 0x00000000 |

| DCDSTn | Bit | Description | Initial State |
|---|---|---|---|
| CURR_DST | [30:0] | Current destination address for DMAn | 0x00000000 |

### DMA MASK TRIGGER (DMASKTRIG) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DMASKTRIG0 | 0x4B000020 | R/W | DMA 0 mask trigger register | 000 |
| DMASKTRIG1 | 0x4B000060 | R/W | DMA 1 mask trigger register | 000 |
| DMASKTRIG2 | 0x4B0000A0 | R/W | DMA 2 mask trigger register | 000 |
| DMASKTRIG3 | 0x4B0000E0 | R/W | DMA 3 mask trigger register | 000 |

| DMASKTRIGn | Bit | Description | Initial State |
|---|---|---|---|
| STOP | [2] | Stop the DMA operation.<br>1: DMA stops as soon as the current atomic transfer ends. If there is no current running atomic transfer, DMA stops immediately. The CURR_TC, CURR_SRC, and CURR_DST will be 0.<br>Note: Due to possible current atomic transfer, “stop” operation may take several cycles. The finish of the operation (i.e. actual stop time) can be detected as soon as the channel on/off bit (DMASKTRIGn[1]) is set to off. This stop is “actual stop”. | 0 |
| ON_OFF | [1] | DMA channel on/off bit.<br>0: DMA channel is turned off. (DMA request to this channel is ignored.)<br>1: DMA channel is turned on and the DMA request is handled. This bit is automatically set to off if we set the DCONn[22] bit to “no auto reload” and/or STOP bit of DMASKTRIGn to “stop”. Note that when DCON[22] bit is "no auto reload", this bit becomes 0 when CURR_TC reaches 0. If the STOP bit is 1, this bit becomes 0 as soon as the current atomic transfer is completed.<br>Note. This bit should not be changed manually during DMA operations (i.e. this has to be changed only by using DCON[22] or STOP bit). | 0 |
| SW_TRIG | [0] | Trigger the DMA channel in S/W request mode.<br>1: it requests a DMA operation to this controller.<br>Note that this trigger gets effective after S/W request mode has to be selected (DCONn[23]) and channel ON_OFF bit has to be set to 1 (channel on). When DMA operation starts, this bit is cleared automatically. | 0 |

**Note**: You are allowed to change the values of DISRC register, DIDST registers, and TC field of DCON register. Those changes take effect only after the finish of current transfer (i.e. when CURR_TC becomes 0). On the other hand, any change made to other registers and/or fields takes immediate effect. Therefore, be careful in changing those registers and fields.

**NOTES**

# 9 I/O PORTS

## OVERVIEW

S3C2440X has 130 multi-functional input/output port pins. There are eight ports:

- Port A(GPA): 25-output port

- Port B(GPB): 11-input/out port

- Port C(GPC): 16-input/output port

- Port D(GPD): 16-input/output port

- Port E(GPE): 16-input/output port

- Port F(GPF): 8-input/output port

- Port G(GPG): 16-input/output port

- Port H(GPH): 9-input/output port

- Port J(GPJ): 13-input/output port

Each port can be easily configured by software to meet various system configurations and design requirements. You have to define which function of each pin is used before starting the main program. If a pin is not used for multiplexed functions, the pin can be configured as I/O ports.

Initial pin states are configured seamlessly to avoid problems.

**Table 9-1. S3C2440X Port Configuration(Sheet 1 of 5)**

| Port A | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPA22 | Output only | nFCE | – | – |
| GPA21 | Output only | nRSTOUT | – | – |
| GPA20 | Output only | nFRE | – | – |
| GPA19 | Output only | nFWE | – | – |
| GPA18 | Output only | ALE | – | – |
| GPA17 | Output only | CLE | – | – |
| GPA16 | Output only | nGCS5 | – | – |
| GPA15 | Output only | nGCS4 | – | – |
| GPA14 | Output only | nGCS3 | – | – |
| GPA13 | Output only | nGCS2 | – | – |
| GPA12 | Output only | nGCS1 | – | – |
| GPA11 | Output only | ADDR26 | – | – |
| GPA10 | Output only | ADDR25 | – | – |
| GPA9 | Output only | ADDR24 | – | – |
| GPA8 | Output only | ADDR23 | – | – |
| GPA7 | Output only | ADDR22 | – | – |
| GPA6 | Output only | ADDR21 | – | – |
| GPA5 | Output only | ADDR20 | – | – |
| GPA4 | Output only | ADDR19 | – | – |
| GPA3 | Output only | ADDR18 | – | – |
| GPA2 | Output only | ADDR17 | – | – |
| GPA1 | Output only | ADDR16 | – | – |
| GPA0 | Output only | ADDR0 | – | – |

**Table 9-1. S3C2440X Port Configuration(Sheet 2 of 5)**

| Port B | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPB10 | Input/output | nXDREQ0 | – | – |
| GPB9 | Input/output | nXDACK0 | – | – |
| GPB8 | Input/output | nXDREQ1 | – | – |
| GPB7 | Input/output | nXDACK1 | – | – |
| GPB6 | Input/output | nXBREQ | – | – |
| GPB5 | Input/output | nXBACK | – | – |
| GPB4 | Input/output | TCLK0 | – | – |
| GPB3 | Input/output | TOUT3 | – | – |
| GPB2 | Input/output | TOUT2 | – | – |
| GPB1 | Input/output | TOUT1 | – | – |
| GPB0 | Input/output | TOUT0 | – | – |

| Port C | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPC15 | Input/output | VD7 | – | – |
| GPC14 | Input/output | VD6 | – | – |
| GPC13 | Input/output | VD5 | – | – |
| GPC12 | Input/output | VD4 | – | – |
| GPC11 | Input/output | VD3 | – | – |
| GPC10 | Input/output | VD2 | – | – |
| GPC9 | Input/output | VD1 | – | – |
| GPC8 | Input/output | VD0 | – | – |
| GPC7 | Input/output | LCD_LPCREVB | – | – |
| GPC6 | Input/output | LCD_LPCREV | – | – |
| GPC5 | Input/output | LCD_LPCOE | – | – |
| GPC4 | Input/output | VM | – | – |
| GPC3 | Input/output | VFRAME | – | – |
| GPC2 | Input/output | VLINE | – | – |
| GPC1 | Input/output | VCLK | – | – |
| GPC0 | Input/output | LEND | – | – |

**Table 9-1. S3C2440X Port Configuration(Sheet 3 of 5)**

| Port D | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPD15 | Input/output | VD23 | nSS0 | – |
| GPD14 | Input/output | VD22 | nSS1 | – |
| GPD13 | Input/output | VD21 | – | – |
| GPD12 | Input/output | VD20 | – | – |
| GPD11 | Input/output | VD19 | – | – |
| GPD10 | Input/output | VD18 | SPICLK1 | – |
| GPD9 | Input/output | VD17 | SPIMOSI1 | – |
| GPD8 | Input/output | VD16 | SPIMISO1 | – |
| GPD7 | Input/output | VD15 | – | – |
| GPD6 | Input/output | VD14 | – | – |
| GPD5 | Input/output | VD13 | – | – |
| GPD4 | Input/output | VD12 | – | – |
| GPD3 | Input/output | VD11 | – | – |
| GPD2 | Input/output | VD10 | – | – |
| GPD1 | Input/output | VD9 | – | – |
| GPD0 | Input/output | VD8 | – | – |

| Port E | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPE15 | Input/output | IICSDA | – | – |
| GPE14 | Input/output | IICSCL | – | – |
| GPE13 | Input/output | SPICLK0 | – | – |
| GPE12 | Input/output | SPIMOSI0 | – | – |
| GPE11 | Input/output | SPIMISO0 | – | – |
| GPE10 | Input/output | SDDAT3 | – | – |
| GPE9 | Input/output | SDDAT2 | – | – |
| GPE8 | Input/output | SDDAT1 | – | – |
| GPE7 | Input/output | SDDAT0 | – | – |
| GPE6 | Input/output | SDCMD | – | – |
| GPE5 | Input/output | SDCLK | – | – |
| GPE4 | Input/output | I2SSDO | I2SSDI | – |
| GPE3 | Input/output | I2SSDI | nSS0 | – |
| GPE2 | Input/output | CDCLK | – | – |
| GPE1 | Input/output | I2SSCLK | – | – |
| GPE0 | Input/output | I2SLRCK | – | – |

**Table 9-1. S3C2440X Port Configuration(Sheet 4 of 5)**

| Port F | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPF7 | Input/output | EINT7 | – | – |
| GPF6 | Input/output | EINT6 | – | – |
| GPF5 | Input/output | EINT5 | – | – |
| GPF4 | Input/output | EINT4 | – | – |
| GPF3 | Input/output | EINT3 | – | – |
| GPF2 | Input/output | EINT2 | | |
| GPF1 | Input/output | EINT1 | | |
| GPF0 | Input/output | EINT0 | | |

| Port G | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPG15 | Input/output | EINT23 | – | – |
| GPG14 | Input/output | EINT22 | – | – |
| GPG13 | Input/output | EINT21 | – | – |
| GPG12 | Input/output | EINT20 | – | – |
| GPG11 | Input/output | EINT19 | TCLK1 | – |
| GPG10 | Input/output | EINT18 | nCTS1 | – |
| GPG9 | Input/output | EINT17 | nRTS1 | – |
| GPG8 | Input/output | EINT16 | – | – |
| GPG7 | Input/output | EINT15 | SPICLK1 | – |
| GPG6 | Input/output | EINT14 | SPIMOSI1 | – |
| GPG5 | Input/output | EINT13 | SPIMISO1 | – |
| GPG4 | Input/output | EINT12 | LCD_PWREN | – |
| GPG3 | Input/output | EINT11 | nSS1 | – |
| GPG2 | Input/output | EINT10 | nSS0 | – |
| GPG1 | Input/output | EINT9 | – | – |
| GPG0 | Input/output | EINT8 | – | – |

**Table 9-1. S3C2440X Port Configuration(Sheet 5 of 5)**

| Port H | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPH10 | Input/output | CLKOUT1 | – | – |
| GPH9 | Input/output | CLKOUT0 | – | – |
| GPH8 | Input/output | UCLK | – | – |
| GPH7 | Input/output | RXD2 | nCTS1 | – |
| GPH6 | Input/output | TXD2 | nRTS1 | – |
| GPH5 | Input/output | RXD1 | – | – |
| GPH4 | Input/output | TXD1 | – | – |
| GPH3 | Input/output | RXD0 | – | – |
| GPH2 | Input/output | TXD0 | – | – |
| GPH1 | Input/output | nRTS0 | – | – |
| GPH0 | Input/output | nCTS0 | – | – |

| Port J | Selectable Pin Functions | | | |
|---|---|---|---|---|
| GPJ12 | Input/output | CAMRESET | – | – |
| GPJ11 | Input/output | CAMCLKOUT | – | – |
| GPJ10 | Input/output | CAMHREF | – | – |
| GPJ9 | Input/output | CAMVSYNC | – | – |
| GPJ8 | Input/output | CMAPCLKIN | – | – |
| GPJ7 | Input/output | CAMDATA7 | – | – |
| GPJ6 | Input/output | CAMDATA6 | – | – |
| GPJ5 | Input/output | CAMDATA5 | – | – |
| GPJ4 | Input/output | CAMDATA4 | – | – |
| GPJ3 | Input/output | CAMDATA3 | – | – |
| GPJ2 | Input/output | CAMDATA2 | – | – |
| GPJ1 | Input/output | CAMDATA1 | – | – |
| GPJ0 | Input/output | CAMDATA0 | – | – |

## PORT CONTROL DESCRIPTIONS

### PORT CONFIGURATION REGISTER(GPACON-GPJCON)

In S3C2440X, most pins are multiplexed pins. So, It is determined which function is selected for each pins. The PnCON(port control register) determines which function is used for each pin.

If PE0 – PE7 is used for the wakeup signal in power down mode, these ports must be configured in interrupt mode.

### PORT DATA REGISTER(GPADAT-GPJDAT)

If Ports are configured as output ports, data can be written to the corresponding bit of PnDAT. If Ports are configured as input ports, the data can be read from the corresponding bit of PnDAT.

### PORT PULL-UP REGISTER(GPBUP-GPJUP)

The port pull-up register controls the pull-up resister enable/disable of each port group. When the corresponding bit is 0, the pull-up resister of the pin is enabled. When 1, the pull-up resister is disabled.

If the port pull-up register is enabled then the pull-up resisters work without pin's functional setting(input, output, DATAn, EINTn and etc)

### MISCELLANEOUS CONTROL REGISTER

This register controls DATA port pull-up resister in Sleep mode, USB pad, and CLKOUT selection.

### EXTERNAL INTERRUPT CONTROL REGISTER

The 24 external interrupts are requested by various signaling methods. The EXTINT register configures the signaling method among the low level trigger, high level trigger, falling edge trigger, rising edge trigger, and both edge trigger for the external interrupt request

Because each external interrupt pin has a digital filter, the interrupt controller can recognize the request signal that is longer than 3 clocks.

EINT[15:0] are used for wakeup sources.

# I/O PORT CONTROL REGISTER

## PORT A CONTROL REGISTERS(GPACON, GPADAT)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPACON | 0x56000000 | R/W | Configures the pins of port A | 0xffffff |
| GPADAT | 0x56000004 | R/W | The data register for port A | Undef. |
| Reserved | 0x56000008 | - | Reserved | Undef |
| Reserved | 0x5600000c | - | Reserved | Undef |

| GPACON | Bit | Description | |
|---|---|---|---|
| GPA24 | [24] | reserved | |
| GPA23 | [23] | reserved | |
| GPA22 | [22] | 0 = Output | 1 = nFCE |
| GPA21 | [21] | 0 = Output | 1 = nRSTOUT |
| GPA20 | [20] | 0 = Output | 1 = nFRE |
| GPA19 | [19] | 0 = Output | 1 = nFWE |
| GPA18 | [18] | 0 = Output | 1 = ALE |
| GPA17 | [17] | 0 = Output | 1 = CLE |
| GPA16 | [16] | 0 = Output | 1 = nGCS[5] |
| GPA15 | [15] | 0 = Output | 1 = nGCS[4] |
| GPA14 | [14] | 0 = Output | 1 = nGCS[3] |
| GPA13 | [13] | 0 = Output | 1 = nGCS[2] |
| GPA12 | [12] | 0 = Output | 1 = nGCS[1] |
| GPA11 | [11] | 0 = Output | 1 = ADDR26 |
| GPA10 | [10] | 0 = Output | 1 = ADDR25 |
| GPA9 | [9] | 0 = Output | 1 = ADDR24 |
| GPA8 | [8] | 0 = Output | 1 = ADDR23 |
| GPA7 | [7] | 0 = Output | 1 = ADDR22 |
| GPA6 | [6] | 0 = Output | 1 = ADDR21 |
| GPA5 | [5] | 0 = Output | 1 = ADDR20 |
| GPA4 | [4] | 0 = Output | 1 = ADDR19 |
| GPA3 | [3] | 0 = Output | 1 = ADDR18 |
| GPA2 | [2] | 0 = Output | 1 = ADDR17 |
| GPA1 | [1] | 0 = Output | 1 = ADDR16 |
| GPA0 | [0] | 0 = Output | 1 = ADDR0 |

| GPADAT | Bit | Description |
|---|---|---|
| GPA[24:0] | [24:0] | When the port is configured as output port, the pin state is the same as the corresponding bit.<br>When the port is configured as functional pin, the undefined value will be read. |

nRSTOUT = nRESET & nWDTRST & SW_RESET

## PORT B CONTROL REGISTERS(GPBCON, GPBDAT, GPBUP)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPBCON | 0x56000010 | R/W | Configures the pins of port B | 0x0 |
| GPBDAT | 0x56000014 | R/W | The data register for port B | Undef. |
| GPBUP | 0x56000018 | R/W | pull-up disable register for port B | 0x0 |
| Reserved | 0x5600001c | | | |

| PBCON | Bit | Description |
|---|---|---|
| GPB10 | [21:20] | 00 = Input 01 = Output<br>10 = nXDREQ0 11 = reserved |
| GPB9 | [19:18] | 00 = Input 01 = Output<br>10 = nXDACK0 11 = reserved |
| GPB8 | [17:16] | 00 = Input 01 = Output<br>10 = nXDREQ1 11 = Reserved |
| GPB7 | [15:14] | 00 = Input 01 = Output<br>10 = nXDACK1 11 = Reserved |
| GPB6 | [13:12] | 00 = Input 01 = Output<br>10 = nXBREQ 11 = reserved |
| GPB5 | [11:10] | 00 = Input 01 = Output<br>10 = nXBACK 11 = reserved |
| GPB4 | [9:8] | 00 = Input 01 = Output<br>10 = TCLK [0] 11 = reserved |
| GPB3 | [7:6] | 00 = Input 01 = Output<br>10 = TOUT3 11 = reserved |
| GPB2 | [5:4] | 00 = Input 01 = Output<br>10 = TOUT2 11 = reserved] |
| GPB1 | [3:2] | 00 = Input 01 = Output<br>10 = TOUT1 11 = reserved |
| GPB0 | [1:0] | 00 = Input 01 = Output<br>10 = TOUT0 11 = reserved |

| GPBDAT | Bit | Description |
|---|---|---|
| GPB[10:0] | [10:0] | When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. |

| GPBUP | Bit | Description |
|---|---|---|
| GPB[10:0] | [10:0] | 0: the pull up function attached to to the corresponding port pin is enabled.<br>1: the pull up function is disabled. |

## PORT C CONTROL REGISTERS(GPCCON, GPCDAT, GPCUP)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPCCON | 0x56000020 | R/W | Configures the pins of port C | 0x0 |
| GPCDAT | 0x56000024 | R/W | The data register for port C | Undef. |
| GPCUP | 0x56000028 | R/W | pull-up disable register for port C | 0x0 |
| Reserved | 0x5600002c | - | - | - |

| GPCCON | Bit | Description |
|---|---|---|
| GPC15 | [31:30] | 00 = Input 01 = Output<br>10 = VD[7] 11 = Reserved |
| GPC14 | [29:28] | 00 = Input 01 = Output<br>10 = VD[6] 11 = Reserved |
| GPC13 | [27:26] | 00 = Input 01 = Output<br>10 = VD[5] 11 = Reserved |
| GPC12 | [25:24] | 00 = Input 01 = Output<br>10 = VD[4] 11 = Reserved |
| GPC11 | [23:22] | 00 = Input 01 = Output<br>10 = VD[3] 11 = Reserved |
| GPC10 | [21:20] | 00 = Input 01 = Output<br>10 = VD[2] 11 = Reserved |
| GPC9 | [19:18] | 00 = Input 01 = Output<br>10 = VD[1] 11 = Reserved |
| GPC8 | [17:16] | 00 = Input 01 = Output<br>10 = VD[0] 11 = Reserved |
| GPC7 | [15:14] | 00 = Input 01 = Output<br>10 = LCD_LPCREVB 11 = Reserved |
| GPC6 | [13:12] | 00 = Input 01 = Output<br>10 = LCD_LPCREV 11 = Reserved |
| GPC5 | [11:10] | 00 = Input 01 = Output<br>10 = LCD_LPCOE 11 = Reserved |
| GPC4 | [9:8] | 00 = Input 01 = Output<br>10 = VM 11 = I2SSDI |
| GPC3 | [7:6] | 00 = Input 01 = Output<br>10 = VFRAME 11 = Reserved |
| GPC2 | [5:4] | 00 = Input 01 = Output<br>10 = VLINE 11 = Reserved |
| GPC1 | [3:2] | 00 = Input 01 = Output<br>10 = VCLK 11 = Reserved |
| GPC0 | [1:0] | 00 = Input 01 = Output<br>10 = LEND 11 = Reserved |

| GPCDAT | Bit | Description |
|---|---|---|
| GPC[15:0] | [15:0] | When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. |

| GPCUP | Bit | Description |
|---|---|---|
| GPC[15:0] | [15:0] | 0: the pull up function attached to to the corresponding port pin is enabled.<br>1: the pull up function is disabled. |

## PORT D CONTROL REGISTERS(GPDCON, GPDDAT, GPDUP)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPDCON | 0x56000030 | R/W | Configures the pins of port D | 0x0 |
| GPDDAT | 0x56000034 | R/W | The data register for port D | Undef. |
| GPDUP | 0x56000038 | R/W | pull-up disable register for port D | 0xf000 |
| Reserved | 0x5600003c | - | | |

| GPDCON | Bit | Description |
|---|---|---|
| GPD15 | [31:30] | 00 = Input 01 = Output<br>10 = VD[23] 11 = nSS0 |
| GPD14 | [29:28] | 00 = Input 01 = Output<br>10 = VD[22] 11 = nSS1 |
| GPD13 | [27:26] | 00 = Input 01 = Output<br>10 = VD[21] 11 = Reserved |
| GPD12 | [25:24] | 00 = Input 01 = Output<br>10 = VD[20] 11 = Reserved |
| GPD11 | [23:22] | 00 = Input 01 = Output<br>10 = VD[19] 11 = Reserved |
| GPD10 | [21:20] | 00 = Input 01 = Output<br>10 = VD[18] 11 = SPICLK1 |
| GPD9 | [19:18] | 00 = Input 01 = Output<br>10 = VD[17] 11 = SPIMOSI1 |
| GPD8 | [17:16] | 00 = Input 01 = Output<br>10 = VD[16] 11 = SPIMISO1 |
| GPD7 | [15:14] | 00 = Input 01 = Output<br>10 = VD[15] 11 = Reserved |
| GPD6 | [13:12] | 00 = Input 01 = Output<br>10 = VD[14] 11 = Reserved |
| GPD5 | [11:10] | 00 = Input 01 = Output<br>10 = VD[13] 11 = Reserved |
| GPD4 | [9:8] | 00 = Input 01 = Output<br>10 = VD[12] 11 = Reserved |
| GPD3 | [7:6] | 00 = Input 01 = Output<br>10 = VD[11] 11 = Reserved |
| GPD2 | [5:4] | 00 = Input 01 = Output<br>10 = VD[10] 11 = Reserved |
| GPD1 | [3:2] | 00 = Input 01 = Output<br>10 = VD[9] 11 = Reserved |
| GPD0 | [1:0] | 00 = Input 01 = Output<br>10 = VD[8] 11 = Reserved |

| GPDDAT | Bit | Description |
|---|---|---|
| GPD[15:0] | [15:0] | When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read. |

| GPDUP | Bit | Description |
|---|---|---|
| GPD[15:0] | [15:0] | 0: the pull up function attached to to the corresponding port pin is enabled.<br>1: the pull up function is disabled. |

## PORT E CONTROL REGISTERS(GPECON, GPEDAT, GPEUP)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPECON | 0x56000040 | R/W | Configures the pins of port E | 0x0 |
| GPEDAT | 0x56000044 | R/W | The data register for port E | Undef. |
| GPEUP | 0x56000048 | R/W | pull-up disable register for port E | 0x0000 |
| Reserved | 0x5600004c | - | - | - |

| GPECON | Bit | Description |
|---|---|---|
| GPE15 | [31:30] | 00 = Input 01 = Output<br>10 = IICSDA 11 = Reserved<br>This pad is open-drain, There is no Pull-up option. |
| GPE14 | [29:28] | 00 = Input 01 = Output<br>10 = IICSCL 11 = Reserved<br>This pad is open-drain, There is no Pull-up option. |
| GPE13 | [27:26] | 00 = Input 01 = Output<br>10 = SPICLK0 11 = Reserved |
| GPE12 | [25:24] | 00 = Input 01 = Output<br>10 = SPIMOSI0 11 = Reserved |
| GPE11 | [23:22] | 00 = Input 01 = Output<br>10 = SPIMISO0 11 = Reserved |
| GPE10 | [21:20] | 00 = Input 01 = Output<br>10 = SDDAT3 11 = Reserved |
| GPE9 | [19:18] | 00 = Input 01 = Output<br>10 = SDDAT2 11 = Reserved |
| GPE8 | [17:16] | 00 = Input 01 = Output<br>10 = SDDAT1 11 = Reserved |
| GPE7 | [15:14] | 00 = Input 01 = Output<br>10 = SDDAT0 11 = Reserved |
| GPE6 | [13:12] | 00 = Input 01 = Output<br>10 = SDCMD 11 = Reserved |
| GPE5 | [11:10] | 00 = Input 01 = Output<br>10 = SDCLK 11 = Reserved |
| GPE4 | [9:8] | 00 = Input 01 = Output<br>10 = I2SDO 11 = I2SSDI |
| GPE3 | [7:6] | 00 = Input 01 = Output<br>10 = I2SDI 11 = nSS0 |
| GPE2 | [5:4] | 00 = Input 01 = Output<br>10 = CDCLK 11 = Reserved |
| GPE1 | [3:2] | 00 = Input 01 = Output<br>10 = I2SSCLK 11 = Reserved |
| GPE0 | [1:0] | 00 = Input 01 = Output<br>10 = I2SLRCK 11 = Reserved |

.

| GPEDAT | Bit | Description |
|---|---|---|
| GPE[15:0] | [15:0] | When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit.<br>When the port is configured as a functional pin, the undefined value will be read. |

| GPEUP | Bit | Description |
|---|---|---|
| GPE[13:0] | [13:0] | 0: the pull up function attached to to the corresponding port pin is enabled.<br>1: the pull up function is disabled. |

### PORT F CONTROL REGISTERS(GPFCON, GPFDAT)

If GPF0 - GPF7 will be used for wake-up signals at power down mode, the ports will be set in interrupt mode.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPFCON | 0x56000050 | R/W | Configures the pins of port F | 0x0 |
| GPFDAT | 0x56000054 | R/W | The data register for port F | Undef. |
| GPFUP | 0x56000058 | R/W | pull-up disable register for port F | 0x000 |
| Reserved | 0x5600005c | - | - | |

| GPFCON | Bit | Description |
|---|---|---|
| GPF7 | [15:14] | 00 = Input 01 = Output<br>10 = EINT[7] 11 = Reserved |
| GPF6 | [13:12] | 00 = Input 01 = Output<br>10 = EINT[6] 11 = Reserved |
| GPF5 | [11:10] | 00 = Input 01 = Output<br>10 = EINT[5] 11 = Reserved |
| GPF4 | [9:8] | 00 = Input 01 = Output<br>10 = EINT[4] 11 = Reserved |
| GPF3 | [7:6] | 00 = Input 01 = Output<br>10 = EINT[3] 11 = Reserved |
| GPF2 | [5:4] | 00 = Input 01 = Output<br>10 = EINT2] 11 = Reserved |
| GPF1 | [3:2] | 00 = Input 01 = Output<br>10 = EINT[1] 11 = Reserved |
| GPF0 | [1:0] | 00 = Input 01 = Output<br>10 = EINT[0] 11 = Reserved |

| GPFDAT | Bit | Description |
|---|---|---|
| GPF[7:0] | [7:0] | When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit.<br>When the port is configured as functional pin, the undefined value will be read. |

| GPFUP | Bit | Description |
|---|---|---|
| GPF[7:0] | [7:0] | 0: the pull up function attached to to the corresponding port pin is enabled.<br>1: the pull up function is disabled. |

### PORT G CONTROL REGISTERS(GPGCON, GPGDAT)

If GPG0 - GPGF7 will be used for wake-up signals at power down mode, the ports will be set in interrupt mode.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPGCON | 0x56000060 | R/W | Configures the pins of port G | 0x0 |
| GPGDAT | 0x56000064 | R/W | The data register for port G | Undef. |
| GPGUP | 0x56000068 | R/W | pull-up disable register for port G | 0xfc00 |
| Reserved | 0x5600006c | - | - | - |

| GPGCON | Bit | Description | |
|---|---|---|---|
| GPG15* | [31:30] | 00 = Input<br>10 = EINT[23] | 01 = Output<br>11 = Reserved |
| GPG14* | [29:28] | 00 = Input<br>10 = EINT[22] | 01 = Output<br>11 = Reserved |
| GPG13* | [27:26] | 00 = Input<br>10 = EINT[21] | 01 = Output<br>11 = Reserved |
| GPG12 | [25:24] | 00 = Input<br>10 = EINT[20] | 01 = Output<br>11 = Reserved |
| GPG11 | [23:22] | 00 = Input<br>10 = EINT[19] | 01 = Output<br>11 = TCLK[1] |
| GPG10 | [21:20] | 00 = Input<br>10 = EINT[18] | 01 = Output<br>11 = nCTS1 |
| GPG9 | [19:18] | 00 = Input<br>10 = EINT[17] | 01 = Output<br>11 = nRTS1 |
| GPG8 | [17:16] | 00 = Input<br>10 = EINT[16] | 01 = Output<br>11 = Reserved |
| GPG7 | [15:14] | 00 = Input<br>10 = EINT[15] | 01 = Output<br>11 = SPICLK1 |
| GPG6 | [13:12] | 00 = Input<br>10 = EINT[14] | 01 = Output<br>11 = SPIMOSI1 |
| GPG5 | [11:10] | 00 = Input<br>10 = EINT[13] | 01 = Output<br>11 = SPIMISO1 |
| GPG4 | [9:8] | 00 = Input<br>10 = EINT[12] | 01 = Output<br>11 = LCD_PWRDN |
| GPG3 | [7:6] | 00 = Input<br>10 = EINT[11] | 01 = Output<br>11 = nSS1 |
| GPG2 | [5:4] | 00 = Input<br>10 = EINT[10] | 01 = Output<br>11 = nSS0 |
| GPG1 | [3:2] | 00 = Input<br>10 = EINT[9] | 01 = Output<br>11 = Reserved |
| GPG0 | [1:0] | 00 = Input<br>10 = EINT[8] | 01 = Output<br>11 = Reserved |

**NOTE: GPG[15:13] must be selected as Input in Nand-booting mode.**

| GPGDAT | Bit | Description |
|---|---|---|
| GPG[15:0] | [15:0] | When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit.<br>When the port is configured as functional pin, the undefined value will be read. |

| GPGUP | Bit | Description |
|---|---|---|
| GPG[15:0] | [15:0] | 0: the pull up function attached to to the corresponding port pin is enabled.<br>1: the pull up function is disabled. |

### PORT H CONTROL REGISTERS(GPHCON, GPHDAT)

If GPF0 - GPF7 will be used for wake-up signals at power down mode, the ports will be set in interrupt mode.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPHCON | 0x56000070 | R/W | Configures the pins of port H | 0x0 |
| GPHDAT | 0x56000074 | R/W | The data register for port H | Undef. |
| GPHUP | 0x56000078 | R/W | pull-up disable register for port H | 0x000 |
| Reserved | 0x5600007c | - | - | |

| GPHCON | Bit | Description | |
|---|---|---|---|
| GPH10 | [21:20] | 00 = Input<br>10 = CLKOUT1 | 01 = Output<br>11 = Reserved |
| GPH9 | [19:18] | 00 = Input<br>10 = CLKOUT0 | 01 = Output<br>11 = Reserved |
| GPH8 | [17:16] | 00 = Input<br>10 = UARTCLK | 01 = Output<br>11 = Reserved |
| GPH7 | [15:14] | 00 = Input<br>10 = RXD[2] | 01 = Output<br>11 = nCTS1 |
| GPH6 | [13:12] | 00 = Input<br>10 = TXD[2] | 01 = Output<br>11 = nRTS1 |
| GPH5 | [11:10] | 00 = Input<br>10 = RXD[1] | 01 = Output<br>11 = Reserved |
| GPH4 | [9:8] | 00 = Input<br>10 = TXD[1] | 01 = Output<br>11 = Reserved |
| GPH3 | [7:6] | 00 = Input<br>10 = RXD[0] | 01 = Output<br>11 = reserved |
| GPH2 | [5:4] | 00 = Input<br>10 = TXD[0] | 01 = Output<br>11 = Reserved |
| GPH1 | [3:2] | 00 = Input<br>10 = nRTS0 | 01 = Output<br>11 = Reserved |
| GPH0 | [1:0] | 00 = Input<br>10 = nCTS0 | 01 = Output<br>11 = Reserved |

| GPHDAT | Bit | Description |
|---|---|---|
| GPH[10:0] | [10:0] | When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit.<br>When the port is configured as functional pin, the undefined value will be read. |

| GPHUP | Bit | Description |
|---|---|---|
| GPH[10:0] | [10:0] | 0: the pull up function attached to to the corresponding port pin is enabled.<br>1: the pull up function is disabled. |

## PORT J CONTROL REGISTERS(GPJCON, GPJDAT)

If GPG0 - GPGF7 will be used for wake-up signals at power down mode, the ports will be set in interrupt mode.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GPJCON | 0x560000d0 | R/W | Configures the pins of port J | 0x0 |
| GPJDAT | 0x560000d4 | R/W | The data register for port J | Undef. |
| GPJUP | 0x560000d8 | R/W | pull-up disable register for port J | 0x0000 |
| Reserved | 0x560000dc | - | - | - |

| GPJCON | Bit | Description | |
|---|---|---|---|
| GPJ12 | [25:24] | 00 = Input<br>10 = CAMRESET (O) | 01 = Output<br>11 = Reserved |
| GPJ11 | [23:22] | 00 = Input<br>10 = CAMCLKOUT(O) | 01 = Output<br>11 = Reserved |
| GPJ10 | [21:20] | 00 = Input<br>10 =CAMHREF (I) | 01 = Output<br>11 = Reserved |
| GPJ9 | [19:18] | 00 = Input<br>10 = CAMVSYNC] (I) | 01 = Output<br>11 = Reserved |
| GPJ8 | [17:16] | 00 = Input<br>10 = CAMPCLKIN] (I) | 01 = Output<br>11 = Reserved |
| GPJ7 | [15:14] | 00 = Input<br>10 = CAMDATA[7] (I) | 01 = Output<br>11 = Reserved |
| GPJ6 | [13:12] | 00 = Input<br>10 = CAMDATA[6] (I) | 01 = Output<br>11 = Reserved |
| GPJ5 | [11:10] | 00 = Input<br>10 = CAMDATA[5] (I) | 01 = Output<br>11 = Reserved |
| GPJ4 | [9:8] | 00 = Input<br>10 = CAMDATA[4] (I) | 01 = Output<br>11 = Reserved |
| GPJ3 | [7:6] | 00 = Input<br>10 = CAMDATA[3] (I) | 01 = Output<br>11 = Reserved |
| GPJ2 | [5:4] | 00 = Input<br>10 = CAMDATA[2] (I) | 01 = Output<br>11 = Reserved |
| GPJ1 | [3:2] | 00 = Input<br>10 = CAMDATA[1] (I) | 01 = Output<br>11 = Reserved |
| GPJ0 | [1:0] | 00 = Input<br>10 = CAMDATA[0] (I) | 01 = Output<br>11 = Reserved |

| GPJDAT | Bit | Description |
|---|---|---|
| GPJ15:0] | [12:0] | When the port is configured as an input port, the corresponding bit is the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit.<br>When the port is configured as functional pin, the undefined value will be read. |

| GPJUP | Bit | Description |
|---|---|---|
| GPJ[12:0] | [12:0] | 0: the pull up function attached to to the corresponding port pin is enabled.<br>1: the pull up function is disabled. |

### MISCELLANEOUS control register(MISCCR)

In Sleep mode, the data bus(D[31:0] or D[15:0] can be set as Hi-Z and Output '0' state. But, because of the characteristics of IO pad, the data bus pull-up resisters have to be turned on or off to reduce the power consumption. D[31:0] pin pull-up resisters can be controlled by MISCCR register.
Pads related USB are controlled by this register for USB host, or for USB device.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| MISCCR | 0x56000080 | R/W | Miscellaneous control register | 0x10330 |

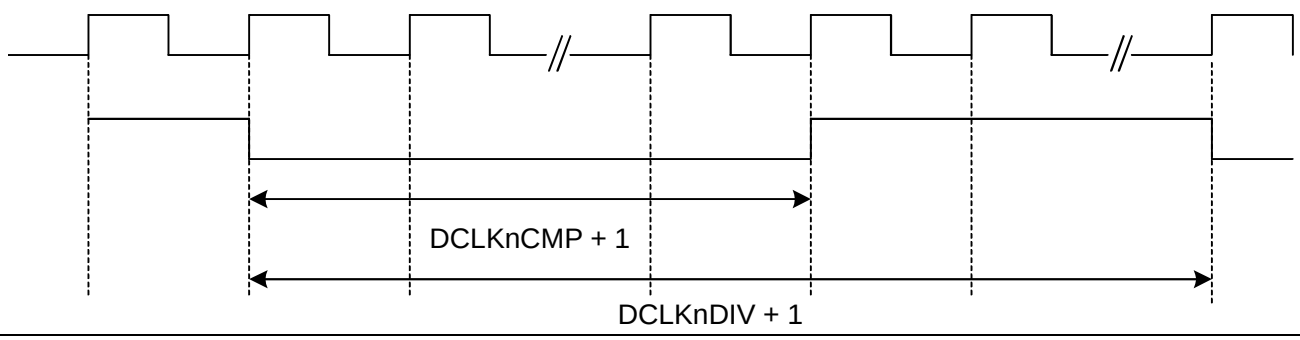
| MISCCR | Bit | Description | Reset Value |
|---|---|---|---|
| Reserved | [24] | Reserve to 0. | 0 |
| Reserved | [23] | Reserve to 0. | 0 |
| Reserved | [22] | This bit must be 1. | 0 |
| BATTFLT_INTR | [21] | BATT_FLT Interrupt On/Off.<br>0: Enable 1: Disable<br>When 1, Battery fault interrupt will be masked by hardware. | 0 |
| BATTFLT_FUNC | [20] | BATT_FLT function On/Off.<br>0: Enable 1: Disable<br>When 0, Battary fault function will be turned on. | 0 |
| OFFREFRESH | [19] | 0: Self refresh retain disable 1: Self refresh retain enable<br>When 1, After wake-up from sleep, The self-refresh will be retained. | 0 |
| nEN_SCLK1 | [18] | SCLK0 output enable<br>0: SCLK1 = SCLK , 1: SCLK1 = 0 | 0 |
| nEN_SCLK0 | [17] | SCLK0 output enable<br>0: SCLK0 = SCLK , 1: SCLK 0 = 0 | 0 |
| nRSTCON | [16] | nRSTOUT의 S/W reset<br>0: nRSTOUT = 0, 1: nRSTOUT = 1 | 1 |
| Reserved | [15:14] | - | 00 |
| SEL_SUSPND1 | [13] | USB Port 1 Suspend mode<br>0 = normal mode 1= suspend mode | 0 |
| SEL_SUSPND0 | [12] | USB Port 0 Suspend mode<br>0 = normal mode 1= suspend mode | 0 |
| CLKSEL1 | [10:8] | 001 = Select UPLL output with CLKOUT0 pad<br>011 = Select HCLK with CLKOUT1 pad<br>100 = Select PCLK with CLKOUT1 pad<br>101 = Select DCLK1 with CLKOUT1 pad<br>11x = reserved | 011 |

| MISCCR | Bit | Description | Reset Value |
|---|---|---|---|
| Reserved | [7] | - | 0 |
| CLKSEL0 | [6:4] | 001 = Select UPLL output with CLKOUT0 pad<br>011 = Select HCLK with CLKOUT0 pad<br>100 = Select PCLK with CLKOUT0 pad<br>101 = Select DCLK0 with CLKOUT0 pad<br>11x = reserved | 011 |
| SEL_USBPAD | [3] | USB1 Host/Device select register.<br>0 = use USB1 as Device<br>1 = use USB1 as Host | 0 |
| Reserved | [2] | Reserved | 0 |
| SPUCR1 | [1] | 0 = DATA[31:16] port pull-up resister is enabled<br>1 = DATA[31:16] port pull-up resister is disabled | 0 |
| SPUCR0 | [0] | 0 = DATA[15:0] port pull-up resister is enabled<br>1 = DATA[15:0] port pull-up resister is disabled | 0 |

## DCLK CONTROL REGISTERS(DCLKCON)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DCLKCON | 0x56000084 | R/W | DCLK0/1 Control Register | 0x0 |

| DCLKCON | Bit | Description |
|---|---|---|
| DCLK1CMP | [27:24] | DCLK1 Compare value clock toggle value.( < DCLK1DIV)<br>If the DCLK1DIV is n, Low level duration is( n + 1),<br>High level duration is((DCLK1DIV + 1) –( n +1)) |
| DCLK1DIV | [23:20] | DCLK1 Divde value<br>DCLK1 frequency = source clock /( DCLK1DIV + 1) |
| DCLK1SelCK | [17] | Select DCLK1 source clock<br>0 = PCLK 1 = UCLK( USB) |
| DCLK1EN | [16] | DCLK1 Enable<br>0 = DCLK1 disable 1 = DCLK1 enable |
| DCLK0CMP | [11:8] | DCLK0 Compare value clock toggle value.( < DCLK0DIV)<br>If the DCLK0DIV is n, Low level duration is( n + 1),<br>High level duration is((DCLK0DIV + 1) –( n +1)) |
| DCLK0DIV | [7:4] | DCLK0 Divde value.<br>DCLK0 frequency = source clock /( DCLK0DIV + 1) |
| DCLK0SelCK | [1] | Select DCLK0 source clock<br>0 = PCLK 1 = UCLK( USB) |
| DCLK0EN | [0] | DCLK0 Enable<br>0 = DCLK0 disable 1 = DCLK0 enable |



**EXTINTn(External Interrupt Control Register n)**

The 8 external interrupts can be requested by various signaling methods. The EXTINT register configures the signaling method between the level trigger and edge trigger for the external interrupt request, and also configures the signal polarity.

To recognize the level interrupt, the valid logic level on EXTINTn pin must be retained for 40ns at least because of the noise filter.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EXTINT0 | 0x56000088 | R/W | External Interrupt control Register 0 | 0x000000 |
| EXTINT1 | 0x5600008c | R/W | External Interrupt control Register 1 | 0x000000 |
| EXTINT2 | 0x56000090 | R/W | External Interrupt control Register 2 | 0x000000 |

| EXTINT0 | Bit | Description |
|---|---|---|
| EINT7 | [30:28] | Setting the signaling method of the EINT7.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| EINT6 | [26:24] | Setting the signaling method of the EINT6.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| EINT5 | [22:20] | Setting the signaling method of the EINT5.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| EINT4 | [18:16] | Setting the signaling method of the EINT4.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| EINT3 | [14:12] | Setting the signaling method of the EINT3.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| EINT2 | [10:8] | Setting the signaling method of the EINT2.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| EINT1 | [6:4] | Setting the signaling method of the EINT1.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| EINT0 | [2:0] | Setting the signaling method of the EINT0.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |

| EXTINT1 | Bit | Description |
|---|---|---|
| FLTEN15 | [31] | Filter Enable for EINT15<br>0 = Filter Disable 1= Filter Enable |
| EINT15 | [30:28] | Setting the signaling method of the EINT15.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| FLTEN14 | [27] | Filter Enable for EINT14<br>0 = Filter Disable 1= Filter Enable |
| EINT14 | [26:24] | Setting the signaling method of the EINT14.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| FLTEN13 | [23] | Filter Enable for EINT13<br>0 = Filter Disable 1= Filter Enable |
| EINT13 | [22:20] | Setting the signaling method of the EINT13.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| FLTEN12 | [19] | Filter Enable for EINT12<br>0 = Filter Disable 1= Filter Enable |
| EINT12 | [18:16] | Setting the signaling method of the EINT12.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| FLTEN11 | [15] | Filter Enable for EINT11<br>0 = Filter Disable 1= Filter Enable |
| EINT11 | [14:12] | Setting the signaling method of the EINT11.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| FLTEN10 | [11] | Filter Enable for EINT10<br>0 = Filter Disable 1= Filter Enable |
| EINT10 | [10:8] | Setting the signaling method of the EINT10.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| FLTEN9 | [7] | Filter Enable for EINT9<br>0 = Filter Disable 1= Filter Enable |
| EINT9 | [6:4] | Setting the signaling method of the EINT9.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |
| FLTEN8 | [3] | Filter Enable for EINT8<br>0 = Filter Disable 1= Filter Enable |
| EINT8 | [2:0] | Setting the signaling method of the EINT8.<br>000 = Low level 001 = High level 01x = Falling edge triggered<br>10x = Rising edge triggered 11x = Both edge triggered |

| EXTINT2 | Bit | Description | Reset Value |
|---|---|---|---|
| FLTEN23 | [31] | Filter Enable for EINT23<br>0 = Filter Disable 1= Filter Enable | 0 |
| EINT23 | [30:28] | Setting the signaling method of the EINT23.<br>000 = Low level 001 = High level<br>01x = Falling edge triggered 10x = Rising edge triggered<br>11x = Both edge triggered | 000 |
| FLTEN22 | [27] | Filter Enable for EINT22<br>0 = Filter Disable 1= Filter Enable | 0 |
| EINT22 | [26:24] | Setting the signaling method of the EINT22.<br>000 = Low level 001 = High level<br>01x = Falling edge triggered 10x = Rising edge triggered<br>11x = Both edge triggered | 000 |
| FLTEN21 | [23] | Filter Enable for EINT21<br>0 = Filter Disable 1= Filter Enable | 0 |
| EINT21 | [22:20] | Setting the signaling method of the EINT21.<br>000 = Low level 001 = High level<br>01x = Falling edge triggered 10x = Rising edge triggered<br>11x = Both edge triggered | 000 |
| FLTEN20 | [19] | Filter Enable for EINT20<br>0 = Filter Disable 1= Filter Enable | 0 |
| EINT20 | [18:16] | Setting the signaling method of the EINT20.<br>000 = Low level 001 = High level<br>01x = Falling edge triggered 10x = Rising edge triggered<br>11x = Both edge triggered | 000 |
| FLTEN19 | [15] | Filter Enable for EINT19<br>0 = Filter Disable 1= Filter Enable | 0 |
| EINT19 | [14:12] | Setting the signaling method of the EINT19.<br>000 = Low level 001 = High level<br>01x = Falling edge triggered 10x = Rising edge triggered<br>11x = Both edge triggered | 000 |
| FLTEN18 | [11] | Filter Enable for EINT18<br>0 = Filter Disable 1= Filter Enable | 0 |
| EINT18 | [10:8] | Setting the signaling method of the EINT18.<br>000 = Low level 001 = High level<br>01x = Falling edge triggered 10x = Rising edge triggered<br>11x = Both edge triggered | 000 |
| FLTEN17 | [7] | Filter Enable for EINT17<br>0 = Filter Disable 1= Filter Enable | 0 |

| | | | |
|---|---|---|---|
| EINT17 | [6:4] | Setting the signaling method of the EINT17.<br>000 = Low level 001 = High level<br>01x = Falling edge triggered 10x = Rising edge triggered<br>11x = Both edge triggered | 000 |
| FLTEN16 | [3] | Filter Enable for EINT16<br>0 = Filter Disable 1= Filter Enable | 0 |
| EINT16 | [2:0] | Setting the signaling method of the EINT16.<br>000 = Low level 001 = High level<br>01x = Falling edge triggered 10x = Rising edge triggered<br>11x = Both edge triggered | 000 |

### EINTFLTn(External Interrupt Filter Register n)

To recognize the level interrupt, the valid logic level on EXTINTn pin must be retained for 40ns at least because of the noise filter.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EINTFLT0 | 0x56000094 | R/W | reserved | 0x000000 |
| EINTFLT1 | 0x56000098 | R/W | reserved | 0x000000 |
| EINTFLT2 | 0x5600009c | R/W | External Interrupt control Register 2 | 0x000000 |
| EINTFLT3 | 0x4c6000a0 | R/W | External Interrupt control Register 3 | 0x000000 |

| EINTFLT2 | Bit | Description |
|---|---|---|
| EINTFLT19 | [30:24] | Filtering width of EINT19 |
| FLTCLK18 | [23] | Filter clock of EINT18(Configured by OM)<br>0 = PCLK 1= EXTCLK/OSC_CLK |
| EINTFLT18 | [22:16] | Filtering width of EINT18 |
| FLTCLK17 | [15] | Filter clock of EINT17(Configured by OM)<br>0 = PCLK 1= EXTCLK/OSC_CLK |
| EINTFLT17 | [14:8] | Filtering width of EINT17 |
| FLTCLK16 | [7] | Filter clock of EINT16(Configured by OM)<br>0 = PCLK 1= EXTCLK/OSC_CLK |
| EINTFLT16 | [6:0] | Filtering width of EINT16 |

| EINTFLT3 | Bit | Description |
|---|---|---|
| FLTCLK23 | [31] | Filter clock of EINT23(Configured by OM)<br>0 = PCLK 1= EXTCLK/OSC_CLK |
| EINTFLT23 | [30:24] | Filtering width of EINT23 |
| FLTCLK22 | [23] | Filter clock of EINT22(Configured by OM)<br>0 = PCLK 1= EXTCLK/OSC_CLK |
| EINTFLT22 | [22:16] | Filtering width of EINT22 |
| FLTCLK21 | [15] | Filter clock of EINT21(Configured by OM)<br>0 = PCLK 1= EXTCLK/OSC_CLK |
| EINTFLT21 | [14:8] | Filtering width of EINT21 |
| FLTCLK20 | [7] | Filter clock of EINT20(Configured by OM)<br>0 = PCLK 1= EXTCLK/OSC_CLK |
| EINTFLT20 | [6:0] | Filtering width of EINT20 |

**EINTMASK(External Interrupt Mask Register)**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EINTMASK | 0x560000a4 | R/W | External interupt mask Register | 0x000fffff |

| EINTMASK | Bit | Description |
|---|---|---|
| EINT23 | [23] | 0 = enable interrupt 1= masked |
| EINT22 | [22] | 0 = enable interrupt 1= masked |
| EINT21 | [21] | 0 = enable interrupt 1= masked |
| EINT20 | [20] | 0 = enable interrupt 1= masked |
| EINT19 | [19] | 0 = enable interrupt 1= masked |
| EINT18 | [18] | 0 = enable interrupt 1= masked |
| EINT17 | [17] | 0 = enable interrupt 1= masked |
| EINT16 | [16] | 0 = enable interrupt 1= masked |
| EINT15 | [15] | 0 = enable interrupt 1= masked |
| EINT14 | [14] | 0 = enable interrupt 1= masked |
| EINT13 | [13] | 0 = enable interrupt 1= masked |
| EINT12 | [12] | 0 = enable interrupt 1= masked |
| EINT11 | [11] | 0 = enable interrupt 1= masked |
| EINT10 | [10] | 0 = enable interrupt 1= masked |
| EINT9 | [9] | 0 = enable interrupt 1= masked |
| EINT8 | [8] | 0 = enable interrupt 1= masked |
| EINT7 | [7] | 0 = enable interrupt 1= masked |
| EINT6 | [6] | 0 = enable interrupt 1= masked |
| EINT5 | [5] | 0 = enable interrupt 1= masked |
| EINT4 | [4] | 0 = enable interrupt 1= masked |
| Reserved | [3:0] | Reserved |

**EINTPEND(External Interrupt Pending Register)**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EINTPEND | 0x560000a8 | R/W | External interupt pending Register | 0x00 |

| EINTPEND | Bit | Description | Reset Value |
|---|---|---|---|
| EINT23 | [23] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT22 | [22] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT21 | [21] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT20 | [20] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT19 | [19] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT18 | [18] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT17 | [17] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT16 | [16] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT15 | [15] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT14 | [14] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT13 | [13] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT12 | [12] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT11 | [11] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT10 | [10] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT9 | [9] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT8 | [8] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT7 | [7] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT6 | [6] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |

| | | | |
|---|---|---|---|
| EINT5 | [5] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| EINT4 | [4] | It is cleard by writing “1”<br>0 = not occur 1= occur interrupt | 0 |
| Reserved | [3:0] | Reserved | 0000 |

### GSTATUSn (General Status Registers)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GSTATUS0 | 0x560000ac | R | External pin status | Not define |
| GSTATUS1 | 0x560000b0 | R/W | Chip ID | 0x324100xx |
| GSTATUS2 | 0x560000b4 | R/W | Reset Status | 0x1 |
| GSTATUS3 | 0x560000b8 | R/W | Inform register | 0x0 |
| GSTATUS4 | 0x560000bc | R/W | Inform register | 0x0 |

| GSTATUS0 | Bit | Description |
|---|---|---|
| nWAIT | [3] | Status of nWAIT pin |
| NCON | [2] | Status of NCON pin |
| RnB | [1] | Status of RnB pin |
| BATT_FLT | [0] | Status of BATT_FLT pin |

| GSTATUS1 | Bit | Description |
|---|---|---|
| CHIP ID | [0] | ID register = 0x324400-00 |

| GSTATUS2 | Bit | Description |
|---|---|---|
| Reserved | [3] | Reserved |
| WDTRST | [2] | Boot is caused by Watch Dog Reset<br>cleared by writing “1” |
| SLEEPRST | [1] | Boot is caused by wakeup reset in sleep mode<br>cleared by writing “1”. |
| PWRST | [0] | Boot is caused by Power On Reset<br>cleared by writing “1” |

| GSTATUS3 | Bit | Description |
|---|---|---|
| inform | [31:0] | Inform register. This register is cleard by power on reset. Otherwise, preserve data value. |

| GSTATUS4 | Bit | Description |
|---|---|---|
| inform | [31:0] | Inform register. This register is cleard by power on reset. Otherwise, preserve data value. |

### DSCn (Drive Strength Control)

Control the Memory I/O drive strength

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DSC0 | 0x560000c4 | R/W | strength control register 0 | 0x0 |
| DSC1 | 0x560000c8 | R/W | strength control register 1 | 0x0 |

| DSC0 | Bit | Description | Reset Value |
|---|---|---|---|
| nEN_DSC | [31] | enable Drive Strength Control<br>0: enable 1: Disable | 0 |
| Reserved | [30:10] | - | 0 |
| DSC_ADR | [9:8] | Address Bus Drive strength.<br>00: 12mA 01: 10mA 10: 8mA 11: 6mA | 00 |
| DSC_DATA3 | [7:6] | DATA[31:24] I/O Drive strength.<br>00: 12mA 01: 10mA 10: 8mA 11: 6mA | 00 |
| DSC_DATA2 | [5:4] | DATA[23:16] I/O Drive strength.<br>00: 12mA 01: 10mA 10: 8mA 11: 6mA | 00 |
| DSC_DATA1 | [3:2] | DATA[15:8] I/O Drive strength.<br>00: 12mA 01: 10mA 10: 8mA 11: 6mA | 00 |
| DSC_DATA0 | [1:0] | DATA[7:0] I/O Drive strength.<br>00: 12mA 01: 10mA 10: 8mA 11: 6mA | 00 |

| DSC1 | Bit | Description | Reset Value |
|---|---|---|---|
| DSC_SCK1 | [29:28] | SCLK1 Drive strength.<br>00: 12mA 01: 10mA 10: 8mA 11: 6mA | 00 |
| DSC_SCK0 | [27:26] | SCLK0 Drive strength.<br>00: 16mA 01: 12mA 10: 8mA 11: 6mA | 00 |
| DSC_SCKE | [25:24] | SCKE Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_SDR | [23:22] | nSRAS/nSCAS Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_NFC | [21:20] | Nand Flash Control Drive strength( nFCE, nFRE, nFWE, CLE, ALE).<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_BE | [19:18] | nBE[3:0] Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_WOE | [17:16] | nWE/nOE Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_CS7 | [15:14] | nGCS7 Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_CS6 | [13:12] | nGCS6 Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_CS5 | [11:10] | nGCS5 Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_CS4 | [9:8] | nGCS4 Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_CS3 | [7:6] | nGCS3 Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_CS2 | [5:4] | nGCS2 Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_CS1 | [3:2] | nGCS1 Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |
| DSC_CS0 | [1:0] | nGCS0 Drive strength.<br>00: 10mA 01: 8mA 10: 6mA 11: 4mA | 00 |

**MSLCON (Memory Sleep Control Register)**

Select memory interface status when in SLEEP mode.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| MSLCON | 0x560000cc | R/W | Memory Sleep Control Register | 0x0 |

| DSC0 | Bit | Description | Reset Value |
|---|---|---|---|
| PSC_DATA | [11] | DATA[31:0] pin status in Sleep mode.<br>0: Hi-Z 1: Output “0”. | 0 |
| PSC_WAIT | [10] | nWAIT pin status in Sleep mode.<br>0: Output “0” 1: Input | 0 |
| PSC_RnB | [9] | RnB pin status in Sleep mode.<br>0: Output “0” 1: Input | 0 |
| PSC_NF | [8] | NAND Flash I/F pin status in Sleep mode( nFCE,nFRE,nFWE,ALE,CLE).<br>0: inactive(nFCE,nFRE,nFWE,ALE,CLE = 11100)<br>1: Hi-Z | 0 |
| PSC_SDR | [7] | nSRAS, nSCAS pin status in Sleep mode.<br>0: inactive(“1”) 1: Hi-Z | 0 |
| PSC_DQM | [6] | DQM[3:0]/nWE[3:0] pin status in Sleep mode.<br>0: inactive 1: Hi-Z | 0 |
| PSC_OE | [5] | nOE pin status in Sleep mode.<br>0: inactive(“1”) 1: Hi-Z | 0 |
| PSC_WE | [4] | nWE pin status in Sleep mode.<br>0: inactive(“1”) 1: Hi-Z | 0 |
| PSC_GCS0 | [3] | nGCS[0] pin status in Sleep mode.<br>0: inactive(“1”) 1: Hi-Z | 0 |
| PSC_GCS51 | [2] | nGCS[5:1] pin status in Sleep mode.<br>0: inactive(“1”) 1: Hi-Z | 0 |
| PSC_GCS6 | [1] | nGCS[6] pin status in Sleep mode.<br>0: inactive(“1”) 1: Hi-Z | 0 |
| PSC_GCS7 | [0] | nGCS[7] pin status in Sleep mode.<br>0: inactive(“1”) 1: Hi-Z | 0 |

# 10 PWM TIMER

## OVERVIEW

The S3C2440X has five 16-bit timers. Timer 0, 1, 2, and 3 have Pulse Width Modulation (PWM) function. Timer 4 has an internal timer only with no output pins. The timer 0 has a dead-zone generator, which is used with a large current device.

The timer 0 and 1 share an 8-bit prescaler, while the timer 2, 3 and 4 share other 8-bit prescaler. Each timer has a clock divider, which generates 5 different divided signals (1/2, 1/4, 1/8, 1/16, and TCLK). Each timer block receives its own clock signals from the clock divider, which receives the clock from the corresponding 8-bit prescaler. The 8-bit prescaler is programmable and divides the PCLK according to the loading value, which is stored in TCFG0 and TCFG1 registers.

The timer count buffer register (TCNTBn) has an initial value which is loaded into the down-counter when the timer is enabled. The timer compare buffer register (TCMPBn) has an initial value which is loaded into the compare register to be compared with the down-counter value. This double buffering feature of TCNTBn and TCMPBn makes the timer generate a stable output when the frequency and duty ratio are changed.

Each timer has its own 16-bit down counter, which is driven by the timer clock. When the down counter reaches zero, the timer interrupt request is generated to inform the CPU that the timer operation has been completed. When the timer counter reaches zero, the value of corresponding TCNTBn is automatically loaded into the down counter to continue the next operation. However, if the timer stops, for example, by clearing the timer enable bit of TCONn during the timer running mode, the value of TCNTBn will not be reloaded into the counter.

The value of TCMPBn is used for pulse width modulation (PWM). The timer control logic changes the output level when the down-counter value matches the value of the compare register in the timer control logic. Therefore, the compare register determines the turn-on time (or turn-off time) of a PWM output.

### FEATURE

— Five 16-bit timers

— Two 8-bit prescalers & Two 4-bit divider

— Programmable duty control of output waveform (PWM)

— Auto reload mode or one-shot pulse mode

— Dead-zone generator

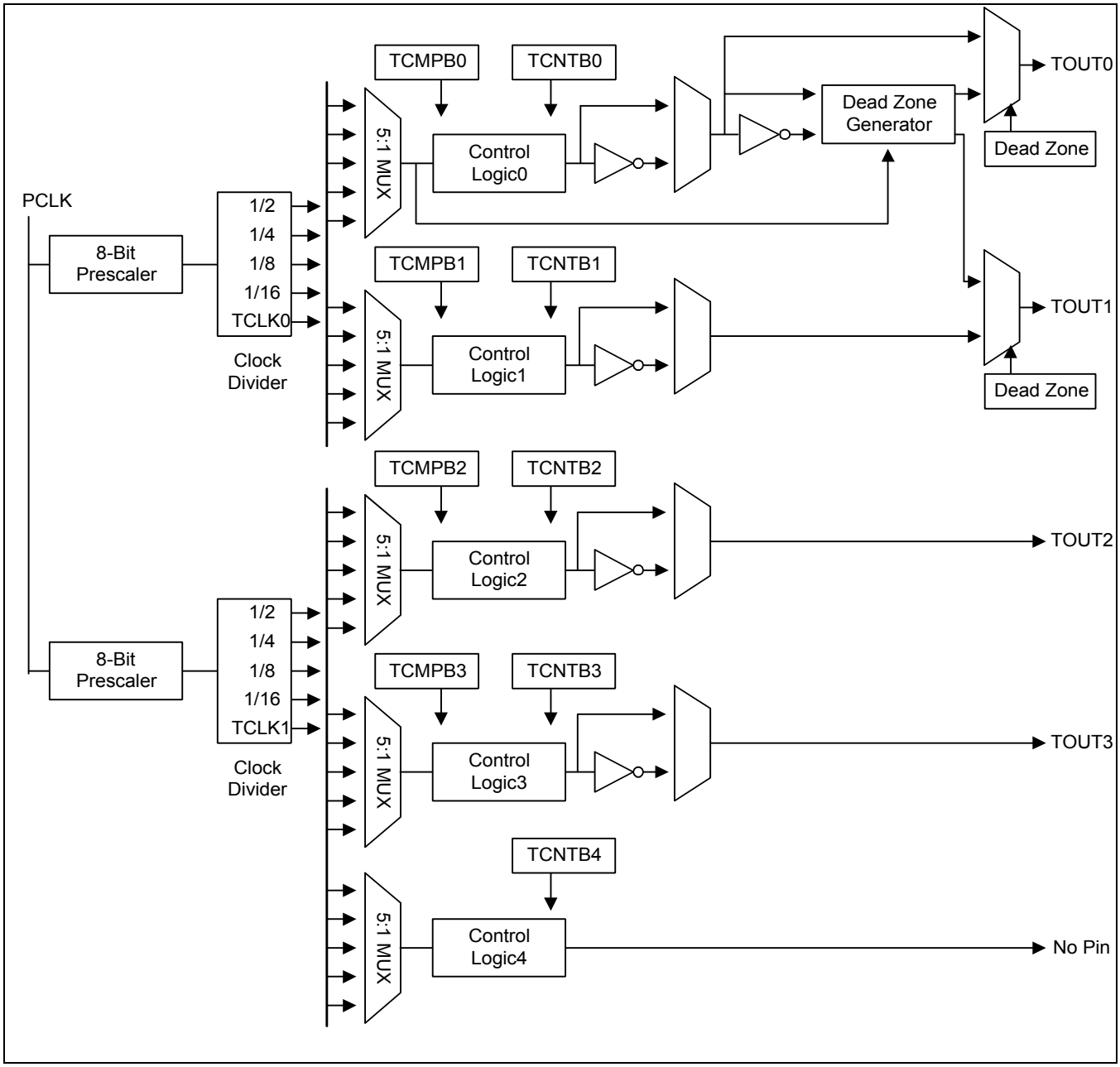



**Figure 10-1. 16-bit PWM Timer Block Diagram**

## PWM TIMER OPERATION

### PRESCALER & DIVIDER

An 8-bit prescaler and a 4-bit divider make the following output frequencies:

| 4-bit divider settings | Minimum resolution (prescaler = 0) | Maximum resolution (prescaler = 255) | Maximum interval (TCNTBn = 65535) |
|---|---|---|---|
| 1/2 (PCLK = 50 MHz) | 0.0400 us (25.0000 MHz) | 10.2400 us (97.6562 KHz) | 0.6710 sec |
| 1/4 (PCLK = 50 MHz) | 0.0800 us (12.5000 MHz) | 20.4800 us (48.8281 KHz) | 1.3421 sec |
| 1/8 (PCLK = 50 MHz) | 0.1600 us ( 6.2500 MHz) | 40.9601 us (24.4140 KHz) | 2.6843 sec |
| 1/16 (PCLK = 50 MHz) | 0.3200 us ( 3.1250 MHz) | 81.9188 us (12.2070 KHz) | 5.3686 sec |

### BASIC TIMER OPERATION

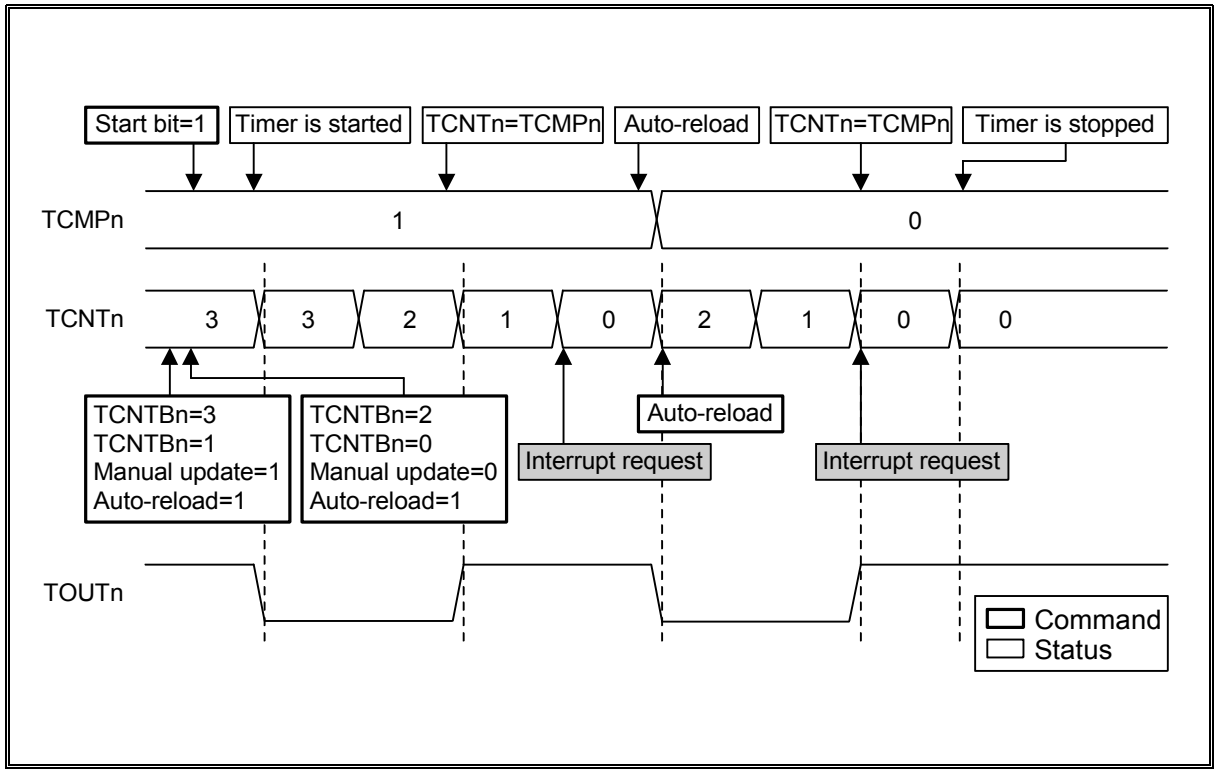



**Figure 10-2. Timer Operations**

A timer (except the timer ch-5) has TCNTBn, TCNTn, TCMPBn and TCMPn. (TCNTn and TCMPn are the names of the internal registers. The TCNTn register can be read from the TCNTOn register) The TCNTBn and the TCMPBn are loaded into the TCNTn and the TCMPn when the timer reaches 0. When the TCNTn reaches 0, an interrupt request will occur if the interrupt is enabled.

### AUTO RELOAD & DOUBLE BUFFERING

S3C2440X PWM Timers have a double buffering function, enabling the reload value changed for the next timer operation without stopping the current timer operation. So, although the new timer value is set, a current timer operation is completed successfully.

The timer value can be written into Timer Count Buffer register (TCNTBn) and the current counter value of the timer can be read from Timer Count Observation register (TCNTOn). If the TCNTBn is read, the read value does not indicate the current state of the counter but the reload value for the next timer duration.

The auto-reload operation copies the TCNTBn into TCNTn when the TCNTn reaches 0. The value, written into the TCNTBn, is loaded to the TCNTn only when the TCNTn reaches 0 and auto reload is enabled. If the TCNTn becomes 0 and the auto reload bit is 0, the TCNTn does not operate any further.

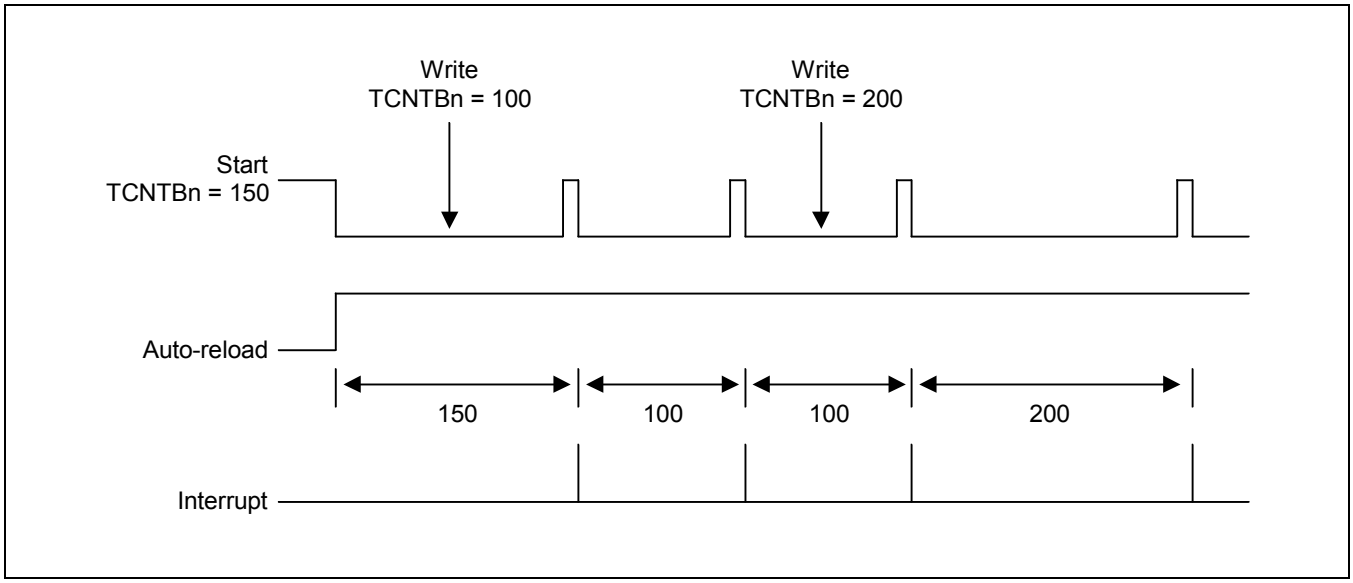



**Figure 10-3. Example of Double Buffering Function**

### TIMER INITIALIZATION USING MANUAL UPDATE BIT AND INVERTER BIT

An auto reload operation of the timer occurs when the down counter reaches 0. So, a starting value of the TCNTn has to be defined by the user in advance. In this case, the starting value has to be loaded by the manual update bit. The following steps describe how to start a timer:

1) Write the initial value into TCNTBn and TCMPBn.

2) Set the manual update bit of the corresponding timer. It is recommended that you configure the inverter on/off bit.
(whether use inverter or not).

3) Set start bit of the corresponding timer to start the timer (and clear the manual update bit).

If the timer is stopped by force, the TCNTn retains the counter value and is not reloaded from TCNTBn. If a new value has to be set, perform manual update.

**NOTE**

Whenever TOUT inverter on/off bit is changed, the TOUTn logic value will also be changed whether the timer runs. Therefore, it is desirable that the inverter on/off bit is configured with the manual update bit.

## TIMER OPERATION

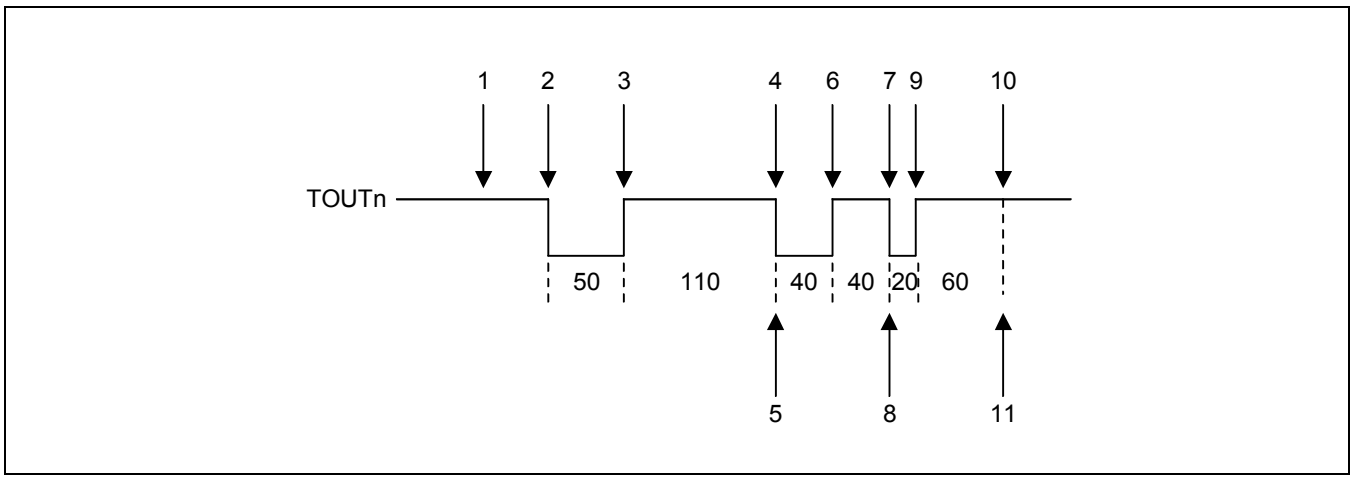



**Figure 10-4. Example of a Timer Operation**

Figure 10-4 shows the result of the following procedure:

1. Enable the auto reload function. Set the TCNTBn to 160 (50+110) and the TCMPBn to 110. Set the manual update bit and configure the inverter bit (on/off). The manual update bit sets TCNTn and TCMPn to the values of TCNTBn and TCMPBn, respectively.

   And then, set the TCNTBn and the TCMPBn to 80 (40+40) and 40, respectively, to determine the next reload value.
2. Set the start bit, provided that manual_update is 0 and the inverter is off and auto reload is on. The timer starts counting down after latency time within the timer resolution.
3. When the TCNTn has the same value as that of the TCMPn, the logic level of the TOUTn is changed from low to high.
4. When the TCNTn reaches 0, the interrupt request is generated and TCNTBn value is loaded into a temporary register. At the next timer tick, the TCNTn is reloaded with the temporary register value (TCNTBn).
5. In Interrupt Service Routine (ISR), the TCNTBn and the TCMPBn are set to 80 (20+60) and 60, respectively, for the next duration.
6. When the TCNTn has the same value as the TCMPn, the logic level of TOUTn is changed from low to high.
7. When the TCNTn reaches 0, the TCNTn is reloaded automatically with the TCNTBn, triggering an interrupt request.
8. In Interrupt Service Routine (ISR), auto reload and interrupt request are disabled to stop the timer.
9. When the value of the TCNTn is same as the TCMPn, the logic level of the TOUTn is changed from low to high.
10. Even when the TCNTn reaches 0, the TCNTn is not any more reloaded and the timer is stopped because auto reload has been disabled.
11. No more interrupt requests are generated.

## PULSE WIDTH MODULATION (PWM)

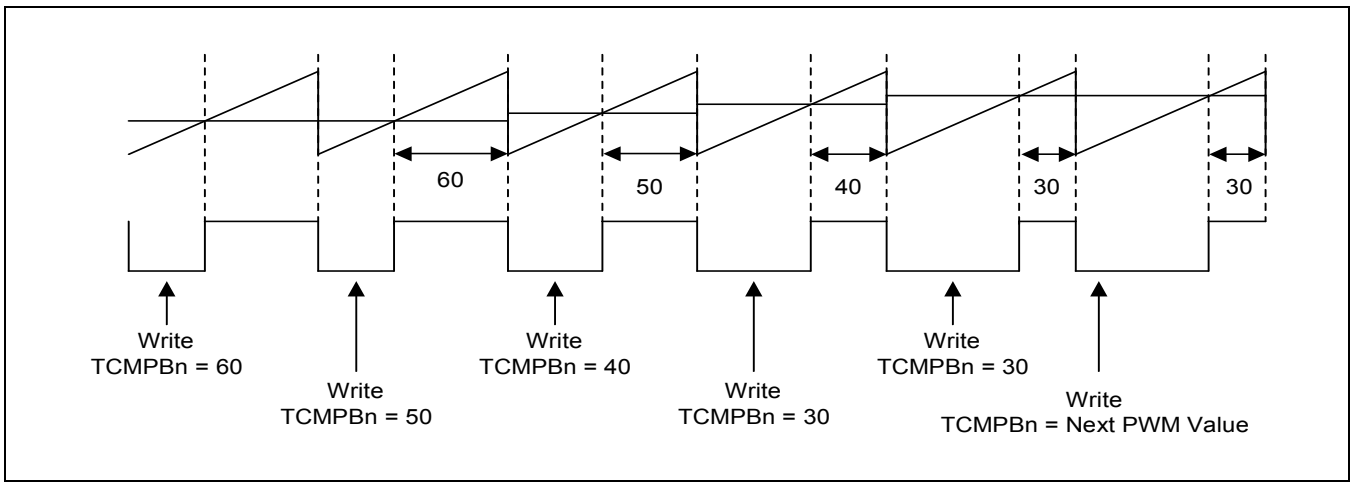



**Figure 10-5. Example of PWM**

PWM function can be implemented by using the TCMPBn. PWM frequency is determined by TCNTBn. Figure 10-5 shows a PWM value determined by TCMPBn.

For a higher PWM value, decrease the TCMPBn value. For a lower PWM value, increase the TCMPBn value. If an output inverter is enabled, the increment/decrement may be reversed.

The double buffering function allows the TCMPBn, for the next PWM cycle, written at any point in the current PWM cycle by ISR or other routine.

## OUTPUT LEVEL CONTROL

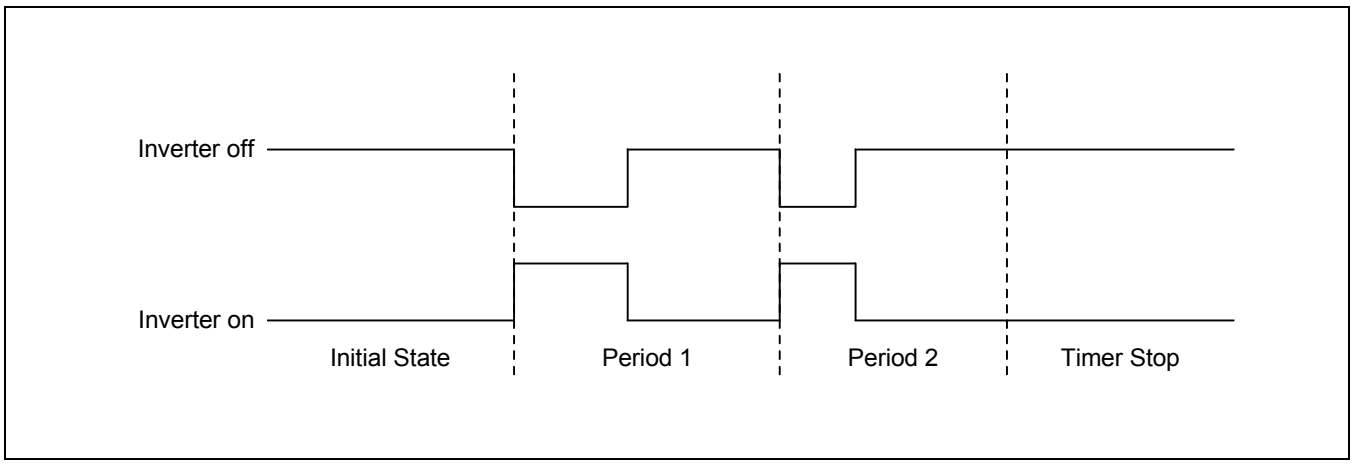



**Figure 10-6. Inverter On/Off**

The following procedure describes how to maintain TOUT as high or low (assume the inverter is off):

1. Turn off the auto reload bit. And then, TOUTn goes to high level and the timer is stopped after the TCNTn reaches 0 (recommended).
2. Stop the timer by clearing the timer start/stop bit to 0. If TCNTn ≤ TCMPn, the output level is **high**. If TCNTn >TCMPn, the output level is **low**.
3. The TOUTn can be inverted by the inverter on/off bit in TCON. The inverter removes the additional circuit to adjust the output level.

## DEAD ZONE GENERATOR

The dead zone is for the PWM control in a power device. This function enables the insertion of the time gap between a turn-off of a switching device and a turn on of another switching device. This time gap prohibits the two switching devices from being turned on simultaneously, even for a very short time.

TOUT0 is the PWM output. nTOUT0 is the inversion of the TOUT0. If the dead zone is enabled, the output wave form of TOUT0 and nTOUT0 will be TOUT0_DZ and nTOUT0_DZ, respectively. nTOUT0_DZ is routed to the TOUT1 pin.

In the dead zone interval, TOUT0_DZ and nTOUT0_DZ can never be turned on simultaneously.

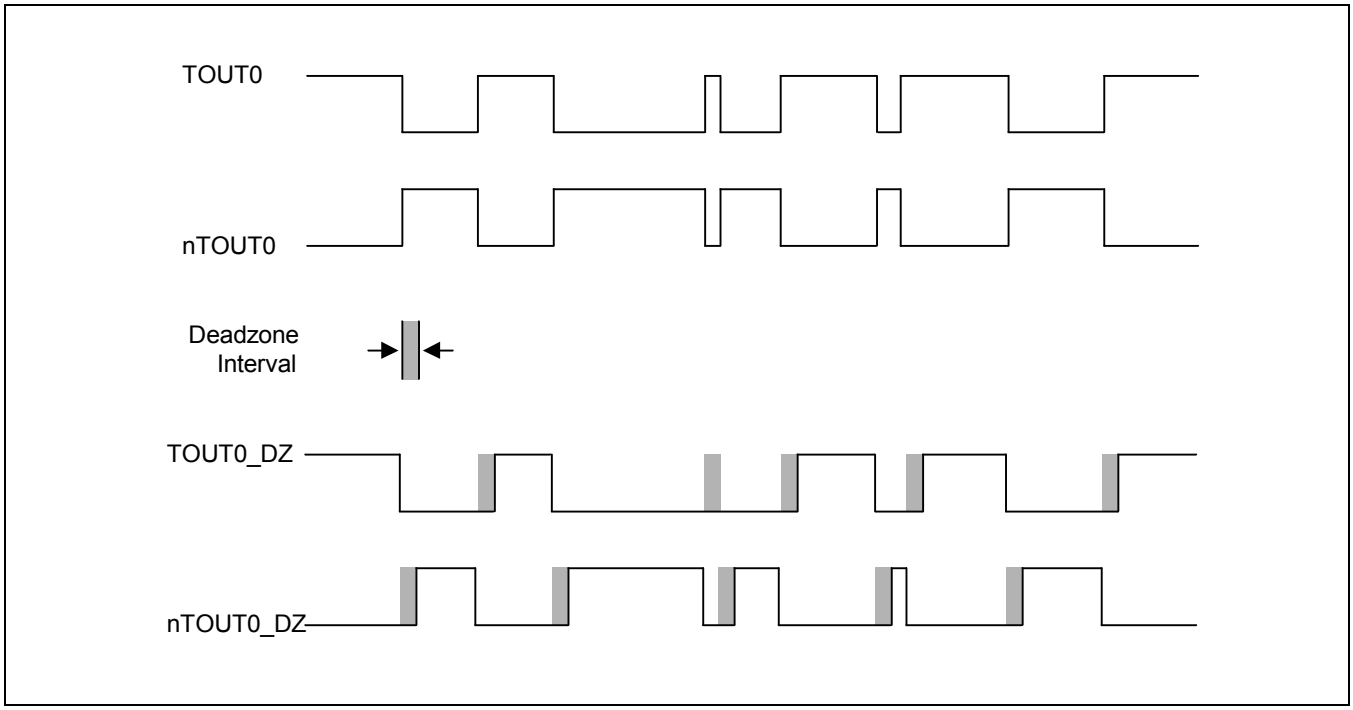



**Figure 10-7. The Wave Form When a Dead Zone Feature is Enabled**

## DMA REQUEST MODE

The PWM timer can generate a DMA request at every specific time. The timer keeps DMA request signals (nDMA_REQ) low until the timer receives an ACK signal. When the timer receives the ACK signal, it makes the request signal inactive. The timer, which generates the DMA request, is determined by setting DMA mode bits (in TCFG1 register). If one of timers is configured as DMA request mode, that timer does not generate an interrupt request. The others can generate interrupt normally.

DMA mode configuration and DMA / interrupt operation

| DMA Mode | DMA Request | Timer0 INT | Timer1 INT | Timer2 INT | Timer3 INT | Timer4 INT |
|---|---|---|---|---|---|---|
| 0000 | No select | ON | ON | ON | ON | ON |
| 0001 | Timer0 | OFF | ON | ON | ON | ON |
| 0010 | Timer1 | ON | OFF | ON | ON | ON |
| 0011 | Timer2 | ON | ON | OFF | ON | ON |
| 0100 | Timer3 | ON | ON | ON | OFF | ON |
| 0101 | Timer4 | ON | ON | ON | ON | OFF |
| 0110 | No select | ON | ON | ON | ON | ON |

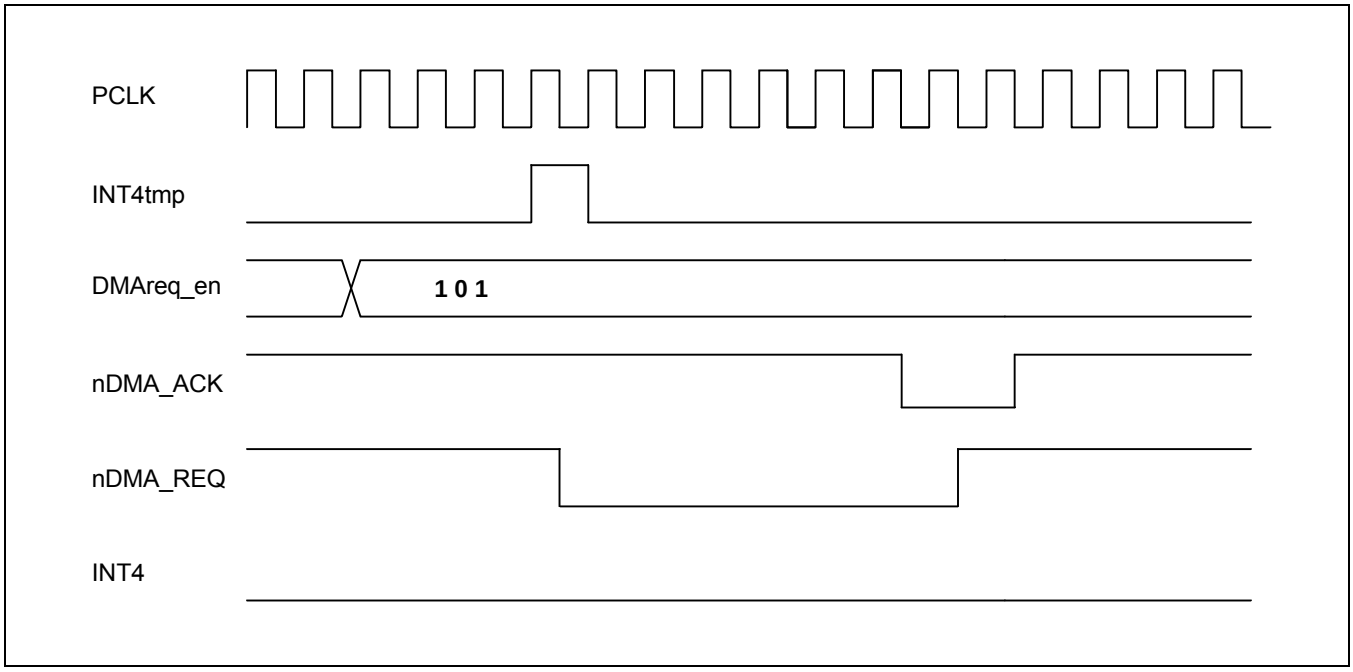



**Figure 10-8. Timer4 DMA Mode Operation**

## PWM TIMER CONTROL REGISTERS

### TIMER CONFIGURATION REGISTER0 (TCFG0)

Timer input clock Frequency = PCLK / {prescaler value+1} / {divider value}
{prescaler value} = 0~255
{divider value} = 2, 4, 8, 16

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCFG0 | 0x51000000 | R/W | Configures the two 8-bit prescalers | 0x00000000 |

| TCFG0 | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [31:24] | | 0x00 |
| Dead zone length | [23:16] | These 8 bits determine the dead zone length. The 1 unit time of the dead zone length is equal to that of timer 0. | 0x00 |
| Prescaler 1 | [15:8] | These 8 bits determine prescaler value for Timer 2, 3 and 4. | 0x00 |
| Prescaler 0 | [7:0] | These 8 bits determine prescaler value for Timer 0 and 1. | 0x00 |

### TIMER CONFIGURATION REGISTER1 (TCFG1)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCFG1 | 0x51000004 | R/W | 5-MUX & DMA mode selecton register | 0x00000000 |

| TCFG1 | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [31:24] | | 00000000 |
| DMA mode | [23:20] | Select DMA request channel<br>0000 = No select (all interrupt) 0001 = Timer0<br>0010 = Timer1 0011 = Timer2<br>0100 = Timer3 0101 = Timer4<br>0110 = Reserved | 0000 |
| MUX 4 | [19:16] | Select MUX input for PWM Timer4.<br>0000 = 1/2 0001 = 1/4 0010 = 1/8<br>0011 = 1/16 01xx = External TCLK1 | 0000 |
| MUX 3 | [15:12] | Select MUX input for PWM Timer3.<br>0000 = 1/2 0001 = 1/4 0010 = 1/8<br>0011 = 1/16 01xx = External TCLK1 | 0000 |
| MUX 2 | [11:8] | Select MUX input for PWM Timer2.<br>0000 = 1/2 0001 = 1/4 0010 = 1/8<br>0011 = 1/16 01xx = External TCLK1 | 0000 |
| MUX 1 | [7:4] | Select MUX input for PWM Timer1.<br>0000 = 1/2 0001 = 1/4 0010 = 1/8<br>0011 = 1/16 01xx = External TCLK0 | 0000 |
| MUX 0 | [3:0] | Select MUX input for PWM Timer0.<br>0000 = 1/2 0001 = 1/4 0010 = 1/8<br>0011 = 1/16 01xx = External TCLK0 | 0000 |

### TIMER CONTROL (TCON) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCON | 0x51000008 | R/W | Timer control register | 0x00000000 |

| TCON | Bit | Description | Initial state |
|---|---|---|---|
| Timer 4 auto reload on/off | [22] | Determine auto reload on/off for Timer 4.<br>0 = One-shot 1 = Interval mode (auto reload) | 0 |
| Timer 4 manual update (note) | [21] | Determine the manual update for Timer 4.<br>0 = No operation 1 = Update TCNTB4 | 0 |
| Timer 4 start/stop | [20] | Determine start/stop for Timer 4.<br>0 = Stop 1 = Start for Timer 4 | 0 |
| Timer 3 auto reload on/off | [19] | Determine auto reload on/off for Timer 3.<br>0 = One-shot 1 = Interval mode (auto reload) | 0 |
| Timer 3 output inverter on/off | [18] | Determine output inverter on/off for Timer 3.<br>0 = Inverter off 1 = Inverter on for TOUT3 | 0 |
| Timer 3 manual update (note) | [17] | Determine manual update for Timer 3.<br>0 = No operation 1 = Update TCNTB3 & TCMPB3 | 0 |
| Timer 3 start/stop | [16] | Determine start/stop for Timer 3.<br>0 = Stop 1 = Start for Timer 3 | 0 |
| Timer 2 auto reload on/off | [15] | Determine auto reload on/off for Timer 2.<br>0 = One-shot 1 = Interval mode (auto reload) | 0 |
| Timer 2 output inverter on/off | [14] | Determine output inverter on/off for Timer 2.<br>0 = Inverter off 1 = Inverter on for TOUT2 | 0 |
| Timer 2 manual update (note) | [13] | Determine the manual update for Timer 2.<br>0 = No operation 1 = Update TCNTB2 & TCMPB2 | 0 |
| Timer 2 start/stop | [12] | Determine start/stop for Timer 2.<br>0 = Stop 1 = Start for Timer 2 | 0 |
| Timer 1 auto reload on/off | [11] | Determine the auto reload on/off for Timer1.<br>0 = One-shot 1 = Interval mode (auto reload) | 0 |
| Timer 1 output inverter on/off | [10] | Determine the output inverter on/off for Timer1.<br>0 = Inverter off 1 = Inverter on for TOUT1 | 0 |
| Timer 1 manual update (note) | [9] | Determine the manual update for Timer 1.<br>0 = No operation 1 = Update TCNTB1 & TCMPB1 | 0 |
| Timer 1 start/stop | [8] | Determine start/stop for Timer 1.<br>0 = Stop 1 = Start for Timer 1 | 0 |

**Note:** The bits have to be cleared at next writing.

**TCON (Continued)**

| TCON | Bit | Description | Initial state |
|---|---|---|---|
| Reserved | [7:5] | Reserved | |
| Dead zone enable | [4] | Determine the dead zone operation.<br>0 = Disable 1 = Enable | 0 |
| Timer 0 auto reload on/off | [3] | Determine auto reload on/off for Timer 0.<br>0 = One-shot 1 = Interval mode(auto reload) | 0 |
| Timer 0 output inverter on/off | [2] | Determine the output inverter on/off for Timer 0.<br>0 = Inverter off 1 = Inverter on for TOUT0 | 0 |
| Timer 0 manual update (note) | [1] | Determine the manual update for Timer 0.<br>0 = No operation 1 = Update TCNTB0 & TCMPB0 | 0 |
| Timer 0 start/stop | [0] | Determine start/stop for Timer 0.<br>0 = Stop 1 = Start for Timer 0 | 0 |

**NOTE:** The bit have to be cleared at next writing.

### TIMER 0 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB0/TCMPB0)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTB0 | 0x5100000C | R/W | Timer 0 count buffer register | 0x00000000 |
| TCMPB0 | 0x51000010 | R/W | Timer 0 compare buffer register | 0x00000000 |

| TCMPB0 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 0 compare buffer register | [15:0] | Set compare buffer value for Timer 0 | 0x00000000 |

| TCNTB0 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 0 count buffer register | [15:0] | Set count buffer value for Timer 0 | 0x00000000 |

### TIMER 0 COUNT OBSERVATION REGISTER (TCNTO0)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTO0 | 0x51000014 | R | Timer 0 count observation register | 0x00000000 |

| TCNTO0 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 0 observation register | [15:0] | Set count observation value for Timer 0 | 0x00000000 |

### TIMER 1 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB1/TCMPB1)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTB1 | 0x51000018 | R/W | Timer 1 count buffer register | 0x00000000 |
| TCMPB1 | 0x5100001C | R/W | Timer 1 campare buffer register | 0x00000000 |

| TCMPB1 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 1 compare buffer register | [15:0] | Set compare buffer value for Timer 1 | 0x00000000 |

| TCNTB1 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 1 count buffer register | [15:0] | Set count buffer value for Timer 1 | 0x00000000 |

### TIMER 1 COUNT OBSERVATION REGISTER (TCNTO1)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTO1 | 0x51000020 | R | Timer 1 count observation register | 0x00000000 |

| TCNTO1 | Bit | Description | initial state |
|---|---|---|---|
| Timer 1 observation register | [15:0] | Set count observation value for Timer 1 | 0x00000000 |

### TIMER 2 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB2/TCMPB2)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTB2 | 0x51000024 | R/W | Timer 2 count buffer register | 0x00000000 |
| TCMPB2 | 0x51000028 | R/W | Timer 2 campare buffer register | 0x00000000 |

| TCMPB2 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 2 compare buffer register | [15:0] | Set compare buffer value for Timer 2 | 0x00000000 |

| TCNTB2 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 2 count buffer register | [15:0] | Set count buffer value for Timer 2 | 0x00000000 |

### TIMER 2 COUNT OBSERVATION REGISTER (TCNTO2)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTO2 | 0x5100002C | R | Timer 2 count observation register | 0x00000000 |

| TCNTO2 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 2 observation register | [15:0] | Set count observation value for Timer 2 | 0x00000000 |

### TIMER 3 COUNT BUFFER REGISTER & COMPARE BUFFER REGISTER (TCNTB3/TCMPB3)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTB3 | 0x51000030 | R/W | Timer 3 count buffer register | 0x00000000 |
| TCMPB3 | 0x51000034 | R/W | Timer 3 campare buffer register | 0x00000000 |

| TCMPB3 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 3 compare buffer register | [15:0] | Set compare buffer value for Timer 3 | 0x00000000 |

| TCNTB3 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 3 count buffer register | [15:0] | Set count buffer value for Timer 3 | 0x00000000 |

### TIMER 3 COUNT OBSERVATION REGISTER (TCNTO3)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTO3 | 0x51000038 | R | Timer 3 count observation register | 0x00000000 |

| TCNTO3 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 3 observation register | [15:0] | Set count observation value for Timer 3 | 0x00000000 |

### TIMER 4 COUNT BUFFER REGISTER (TCNTB4)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTB4 | 0x5100003C | R/W | Timer 4 count buffer register | 0x00000000 |

| TCNTB4 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 4 count buffer register | [15:0] | Set count buffer value for Timer 4 | 0x00000000 |

### TIMER 4 COUNT OBSERVATION REGISTER (TCNTO4)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCNTO4 | 0x51000040 | R | Timer 4 count observation register | 0x00000000 |

| TCNTO4 | Bit | Description | Initial State |
|---|---|---|---|
| Timer 4 observation register | [15:0] | Set count observation value for Timer 4 | 0x00000000 |

## NOTES

# 11 UART

## OVERVIEW

The S3C2440X Universal Asynchronous Receiver and Transmitter (UART) provide three independent asynchronous serial I/O (SIO) ports, each of which can operate in Interrupt-based or DMA-based mode. In other words,the UART can generate an interrupt or a DMA request to transfer data between CPU and the UART. The UART can support bit rates of up to 115.2K bps using system clock. If an external device provides the UART with UARTCLK, then the UART can operate at higher speed. Each UART channel contains two 64-byte FIFOs for receiver and transmitter.

The S3C2440X UART includes programmable baud rates, infrared (IR) transmit/receive, one or two stop bit insertion, 5-bit, 6-bit, 7-bit or 8-bit data width and parity checking.

Each UART contains a baud-rate generator,a transmitter,a receiver and a control unit, as shown in Figure 11-1. The baud-rate generator can be clocked by PCLK or UARTCLK. The transmitter and the receiver contain 64-byte FIFOs and data shifters. Data is written to FIFO and then copied to the transmit shifter before being transmitted. The data is then shifted out by the transmit data pin (TxDn). Meanwhile, received data is shifted from the receive data pin (RxDn), and then copied to FIFO from the shifter.

## FEATURES

— RxD0, TxD0, RxD1, TxD1, RxD2,and TxD2 with DMA-based or interrupt-based operation

— UART Ch 0, 1,and 2 with IrDA 1.0 & 64-byte FIFO

— UART Ch 0 and 1 with nRTS0, nCTS0, nRTS1,and nCTS1

— Supports handshake transmit/receive

## BLOCK DIAGRAM

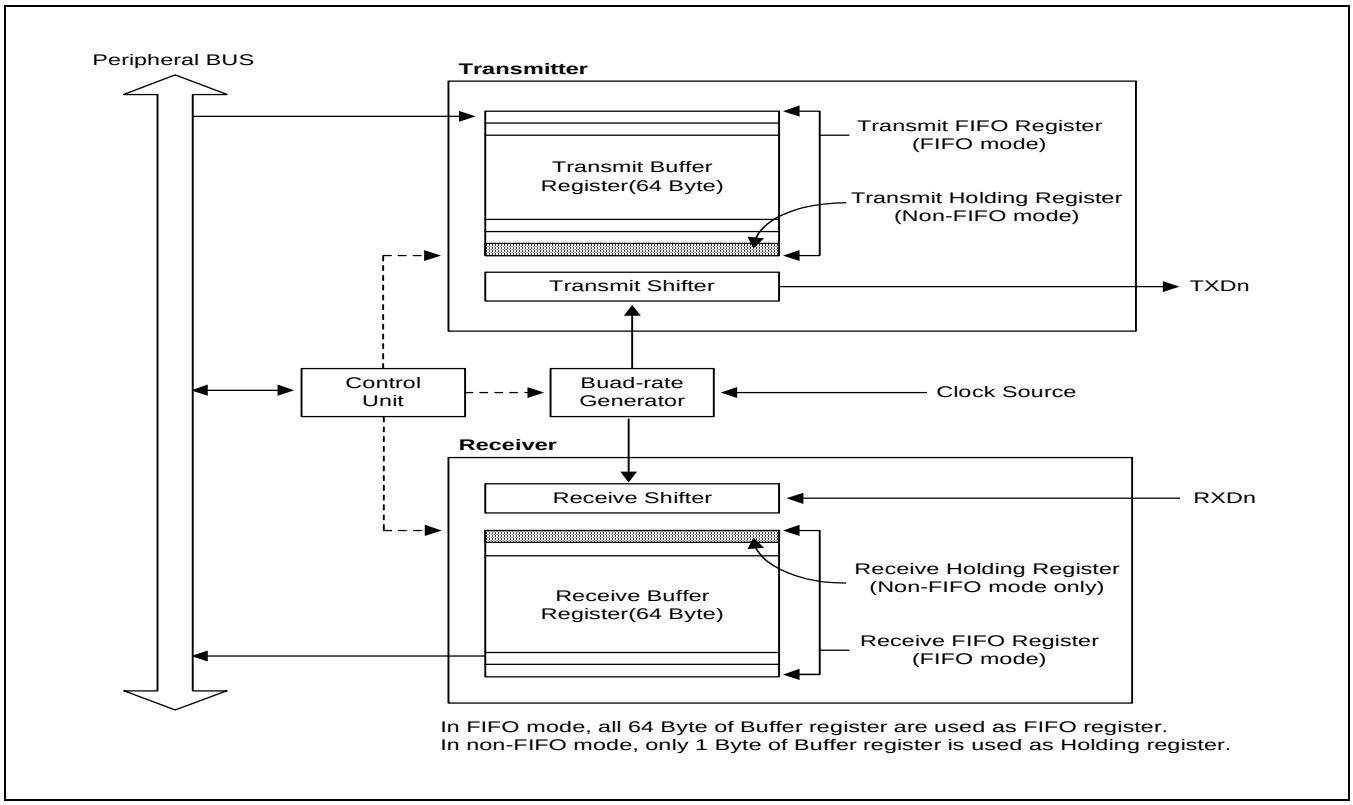



**Figure 11-1 UART Block Diagram (with FIFO)**

## UART OPERATION

The following sections describe the UART operations that include data transmission, data reception, interrupt generation, baud-rate generation, Loopback mode, Infrared mode, and auto flow control.

### Data Transmission

The data frame for transmission is programmable. It consists of a start bit, 5 to 8 data bits, an optional parity bit and 1 to 2 stop bits, which can be specified by the line control register (ULCONn). The transmitter can also produce the break condition, which forces the serial output to logic 0 state for one frame transmission time. This block transmits break signals after the present transmission word is transmitted completely. After the break signal transmission, it continuously transmits data into the Tx FIFO (Tx holding register in the case of Non-FIFO mode).

### Data Reception

Like the transmission, the data frame for reception is also programmable. It consists of a start bit, 5 to 8 data bits, an optional parity bit and 1 to 2 stop bits in the line control register (ULCONn). The receiver can detect overrun error. The overrun error indicates that new data has overwritten the old data before the old data has been read. Receive time-out condition occurs when it does not receive any data during the 3 word time (this interval follows the setting of Word Length bit) and the Rx FIFO is not empty in the FIFO mode.

### Auto Flow Control (AFC)

The S3C2440X's UART 0 and UART 1 support auto flow control with nRTS and nCTS signals. In case, it can be connected to external UARTs. If users want to connect a UART to a Modem, disable auto flow control bit in UMCONn register and control the signal of nRTS by software.

In AFC, nRTS depends on the condition of the receiver and nCTS signals control the operation of the transmitter. The UART's transmitter transfers the data in FIFO only when nCTS signals are activated (in AFC, nCTS means that other UART's FIFO is ready to receive data). Before the UART receives data, nRTS has to be activated when its receive FIFO has a spare more than 32-byte and has to be inactivated when its receive FIFO has a spare under 32-byte (in AFC, nRTS means that its own receive FIFO is ready to receive data).

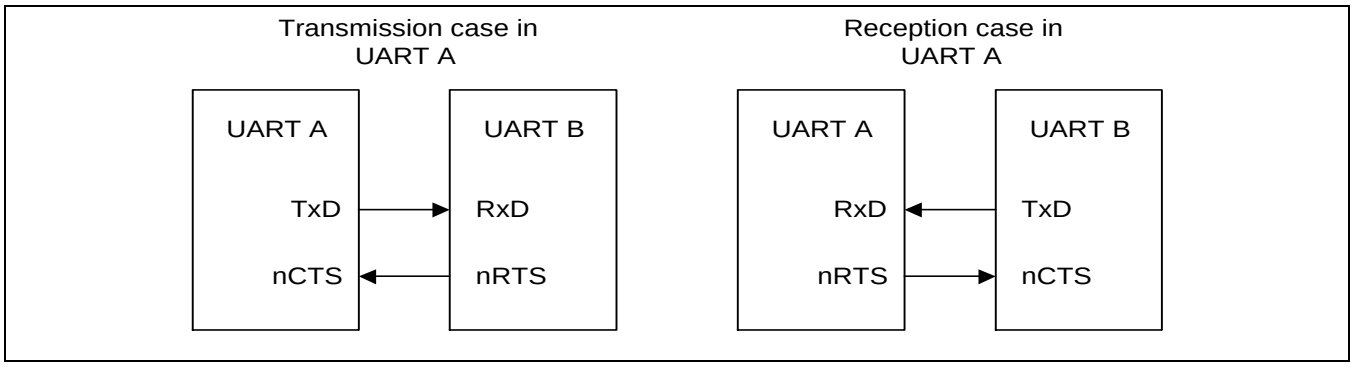



**Figure 11-2 UART AFC interface**

**Note:** UART 2 does not support AFC function, because the S3C2440X has no nRTS2 and nCTS2.

**Example of Non Auto-Flow control (controlling nRTS and nCTS by software)**

**Rx operation with FIFO**

1. Select receive mode (Interrupt or DMA mode).
2. Check the value of Rx FIFO count in UFSTATn register. If the value is less than 32, users have to set the value of UMCONn[0] to '1' (activating nRTS), and if it is equal or larger than 32 users have to set the value to '0' (inactivating nRTS).
3. Repeat the Step 2.

**Tx operation with FIFO**

1. Select transmit mode (Interrupt or DMA mode).
2. Check the value of UMSTATn[0]. If the value is '1' (activating nCTS), users write the data to Tx FIFO register.

### RS-232C interface

If users want to connect the UART to modem interface (instead of null modem), nRTS, nCTS, nDSR, nDTR, DCD and nRI signals are needed. In this case, the users can control these signals with general I/O ports by software because the AFC does not support the RS-232C interface.

### Interrupt/DMA Request Generation

Each UART of the S3C2440X has seven status (Tx/Rx/Error) signals: Overrun error, Receive buffer data ready, Transmit buffer empty, and Transmit shifter empty, all of which are indicated by the corresponding UART status register (UTRSTATn/UERSTATn).

The overrun error can cause the receive error status interrupt request, if the receive-error-status-interrupt-enable bit is set to one in the control register, UCONn. When a receive-error-status-interrupt-request is detected, the signal causing the request can be identified by reading the value of UERSTSTn.

When the receiver transfers the data of the receive shifter to the receive FIFO register in FIFO mode and the number of received data reaches Rx FIFO Trigger Level, Rx interrupt is generated, if Receive mode in control register (UCONn) is selected as 1 (Interrupt request or polling mode).
In the Non-FIFO mode, transferring the data of the receive shifter to the receive holding register will cause Rx interrupt under the Interrupt request and polling mode.

When the transmitter transfers data from its transmit FIFO register to its transmit shifter and the number of data left in transmit FIFO reaches Tx FIFO Trigger Level, Tx interrupt is generated, if Transmit mode in control register is selected as Interrupt request or polling mode.
In the Non-FIFO mode, transferring data from the transmit holding register to the transmit shifter will cause Tx interrupt under the Interrupt request and polling mode.

If the Receive mode and Transmit mode in control register are selected as the DMAn request mode then DMAn request occurs instead of Rx or Tx interrupt in the situation mentioned above.

**Table 11-1 Interrupts in Connection with FIFO**

| Type | FIFO Mode | Non-FIFO Mode |
|---|---|---|
| Rx interrupt | Generated whenever receive data reaches the trigger level of receive FIFO.<br>Generated when the number of data in FIFO does not reaches Rx FIFO trigger Level and does not receive any data during 3 word time (receive time out). This interval follows the setting of Word Length bit. | Generated by the receive holding register whenever receive buffer becomes full. |
| Tx interrupt | Generated whenever transmit data reaches the trigger level of transmit FIFO (Tx FIFO trigger Level). | Generated by the transmit holding register whenever transmit buffer becomes empty. |
| Error interrupt | Overrun error will be generated, when it gets to the top of the receive FIFO without reading out data in it. | Overrun error generates an error interrupt immediately. |

**Baud-Rate Generation**

Each UART's baud-rate generator provides the serial clock for the transmitter and the receiver. The source clock for the baud-rate generator can be selected with the S3C2440X's internal system clock or UARTCLK. In other words, dividend is selectable by setting Clock Selection of UCONn. The baud-rate clock is generated by dividing the source clock (PCLK or UARTCLK) by 16 and a 16-bit divisor specified in the UART baud-rate divisor register (UBRDIVn). The UBRDIVn can be determined by the following expression:

*UBRDIVn* = (int)(PCLK/(bps *x 16) ) -1*

Where, UBRDIVn should be from 1 to ($2^{16}$-1).

For accurate UART operation, the S3C2440X also supports UARTCLK as a dividend.
If the S3C2440X uses UARTCLK, which is supplied by an external UART device or system, then the serial clock of UART is exactly synchronized with UARTCLK. So, the user can get the more precise UART operation. The UBRDIVn can be determined:

UBRDIVn = (int)(UARTCLK / (bps x 16) ) –1

Where, UBRDIVn should be from 1 to ($2^{16}$-1) and UARTCLK should be smaller than PCLK.

For example, if the baud-rate is 115200 bps and PCLK or UARTCLK is 40 MHz, UBRDIVn is determined:

*UBRDIVn = (int)(40000000/(115200 x 16)) -1*
= (int)(21.7) -1
= 21 -1 = 20

**Loopback Mode**

The S3C2440X UART provides a test mode referred to as the Loopback mode, to aid in isolating faults in the communication link.This mode structurally enables the connection of RXD and TXD in the UART. In this mode, therefore, transmitted data is received to the receiver, via RXD. This feature allows the processor to verify the internal transmit and to receive the data path of each SIO channel. This mode can be selected by setting the loopback bit in the UART control register (UCONn).

**Break Condition**

The break is defined as a continuous low level signal for one frame transmission time on the transmit data output.

### Infrared (IR) Mode

The S3C2440X UART block supports infrared (IR) transmission and reception, which can be selected by setting the Infrared-mode bit in the UART line control register (ULCONn). Figure 11-4 illustrates how to implement the IR mode.

In IR transmit mode, the transmit pulse comes out at a rate of 3/16, the normal serial transmit rate (when the transmit data bit is zero); In IR receive mode, the receiver must detect the 3/16 pulsed period to recognize a zero value (see the frame timing diagrams shown in Figure 11-6 and 11-7).

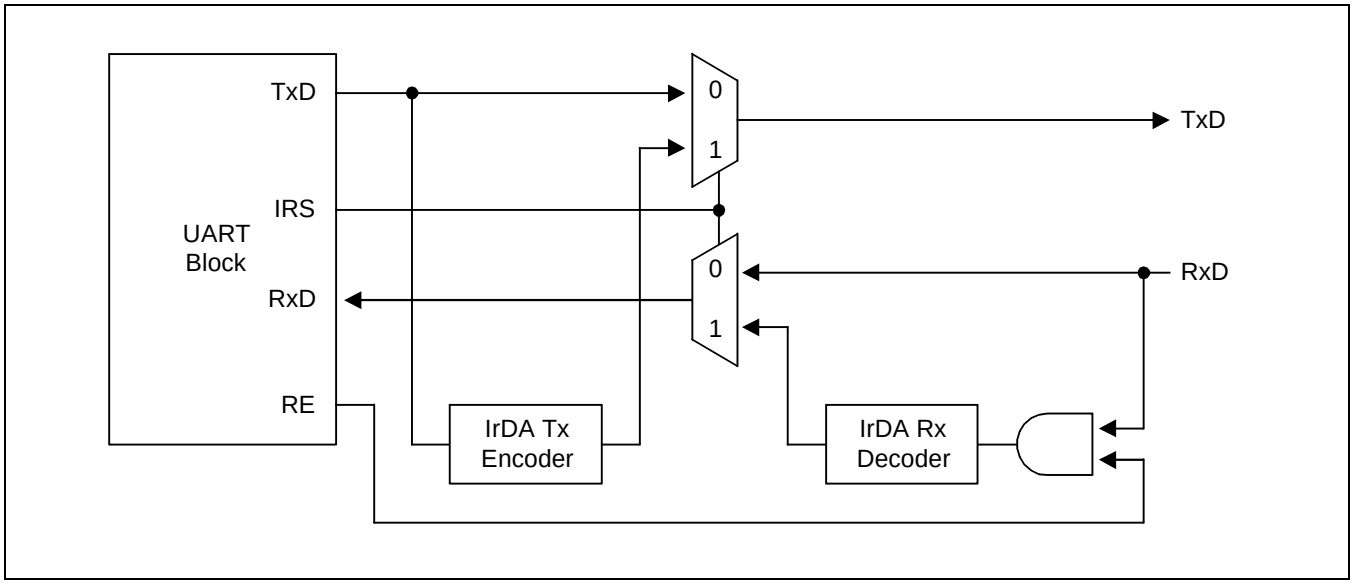



**Figure 11-3. IrDA Function Block Diagram**

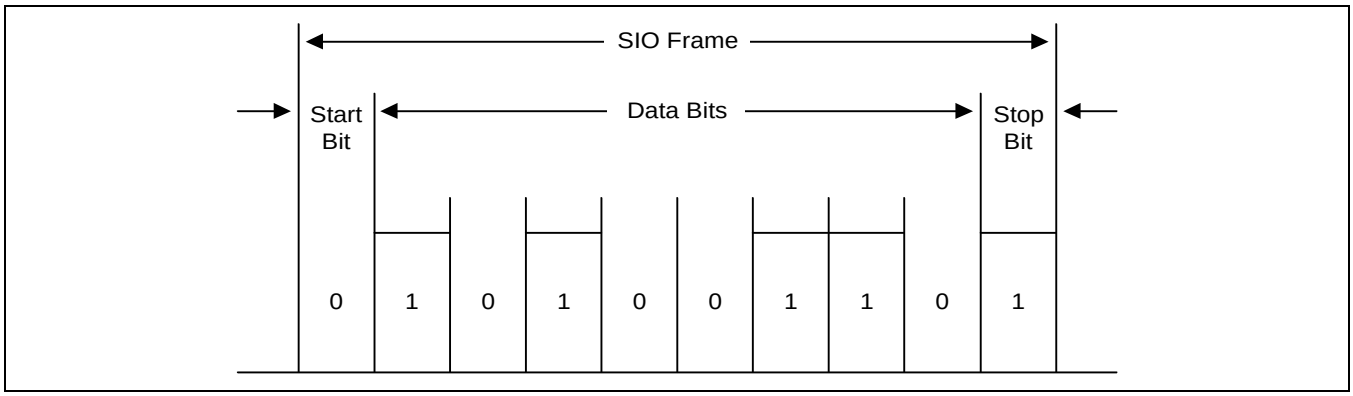



**Figure 11-4. Serial I/O Frame Timing Diagram (Normal UART)**

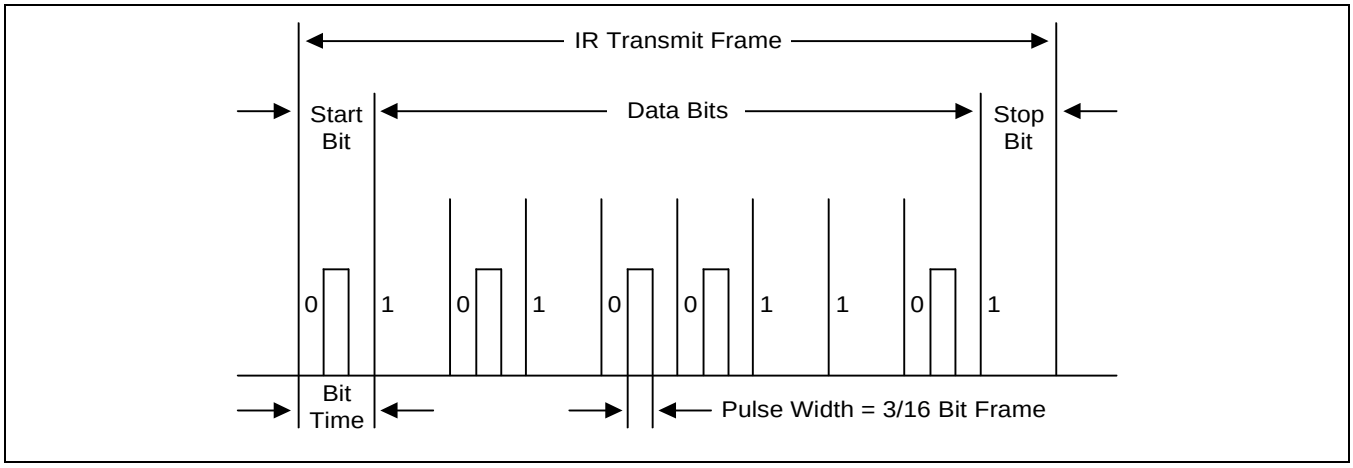



**Figure 11-5. Infrared Transmit Mode Frame Timing Diagram**

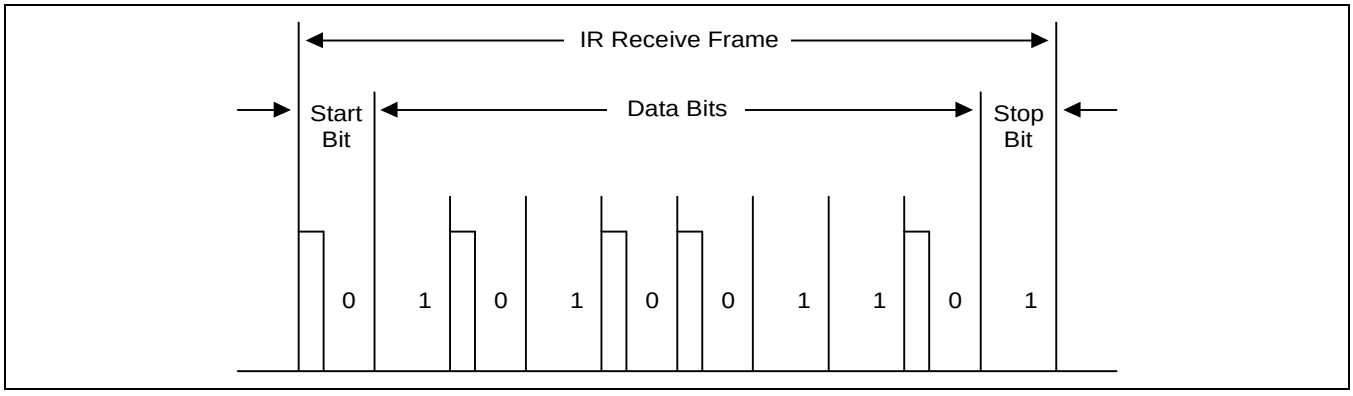



**Figure 11-6. Infrared Receive Mode Frame Timing Diagram**

## UART SPECIAL REGISTERS

### UART LINE CONTROL REGISTER

There are three UART line control registers including ULCON0, ULCON1, and ULCON2 in the UART block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ULCON0 | 0x50000000 | R/W | UART channel 0 line control register | 0x00 |
| ULCON1 | 0x50004000 | R/W | UART channel 1 line control register | 0x00 |
| ULCON2 | 0x50008000 | R/W | UART channel 2 line control register | 0x00 |

| ULCONn | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7] | | 0 |
| Infrared Mode | [6] | Determine whether or not to use the Infrared mode.<br>0 = Normal mode operation<br>1 = Infrared Tx/Rx mode | 0 |
| Parity Mode | [5:3] | Specify the type of parity generation and checking during UART transmit and receive operation.<br>0xx = No parity<br>100 = Odd parity<br>101 = Even parity<br>110 = Parity forced/checked as 1<br>111 = Parity forced/checked as 0 | 000 |
| Number of Stop Bit | [2] | Specify how many stop bits are to be used for end-of-frame signal.<br>0 = One stop bit per frame<br>1 = Two stop bit per frame | 0 |
| Word Length | [1:0] | Indicate the number of data bits to be transmitted or received per frame.<br>00 = 5-bits 01 = 6-bits<br>10 = 7-bits 11 = 8-bits | 00 |

### UART CONTROL REGISTER

There are three UART control registers including UCON0, UCON1 and UCON2 in the UART block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UCON0 | 0x50000004 | R/W | UART channel 0 control register | 0x00 |
| UCON1 | 0x50004004 | R/W | UART channel 1 control register | 0x00 |
| UCON2 | 0x50008004 | R/W | UART channel 2 control register | 0x00 |

| UCONn | Bit | Description | Initial State |
|---|---|---|---|
| Clock Selection | [10] | Select PCLK or UARTCLK for the UART baud rate.<br>0=PCLK : UBRDIVn = (int)(PCLK / (bps x 16) ) -1<br>1=UARTCLK : UBRDIVn = (int)(UARTCLK / (bps x 16) ) -1 | 0 |
| Tx Interrupt Type | [9] | Interrupt request type.<br>0 = Pulse (Interrupt is requested as soon as the Tx buffer becomes empty in Non-FIFO mode or reaches Tx FIFO Trigger Level in FIFO mode.)<br>1 = Level (Interrupt is requested while Tx buffer is empty in Non-FIFO mode or reaches Tx FIFO Trigger Level in FIFO mode.) | 0 |
| Rx Interrupt Type | [8] | Interrupt request type.<br>0 = Pulse (Interrupt is requested the instant Rx buffer receives the data in Non-FIFO mode or reaches Rx FIFO Trigger Level in FIFO mode.)<br>1 = Level (Interrupt is requested while Rx buffer is receiving data in Non-FIFO mode or reaches Rx FIFO Trigger Level in FIFO mode.) | 0 |
| Rx Time Out Enable | [7] | Enable/Disable Rx time out interrupt when UART FIFO is enabled. The interrupt is a receive interrupt.<br>0 = Disable 1 = Enable | 0 |
| Rx Error Status Interrupt Enable | [6] | This bit enables the UART to generate an interrupt if overrun error occurs during a receive operation<br>0 = Do not generate receive error status interrupt.<br>1 = Generate receive error status interrupt. | 0 |
| Loopback Mode | [5] | Setting loopback bit to 1 causes the UART to enter the loopback mode. This mode is provided for test purposes only.<br>0 = Normal operation 1 = Loopback mode | 0 |
| Send Break Signal | [4] | Setting this bit causes the UART to send a break during 1 frame time. This bit is automatically cleared after sending the break signal.<br>0 = Normal transmit 1 = Send break signal | 0 |

## UART CONTROL REGISTER (CONTINUED)

| | | | |
|---|---|---|---|
| Transmit Mode | [3:2] | Determine which function is currently able to write Tx data to the UART transmit buffer register. (UART Tx Enable/Disable)<br>00 = Disable<br>01 = Interrupt request or polling mode<br>10 = DMA0 request (Only for UART0),<br>DMA3 request (Only for UART2)<br>11 = DMA1 request (Only for UART1) | 00 |
| Receive Mode | [1:0] | Determine which function is currently able to read data from UART receive buffer register. (UART Rx Enable/Disable)<br>00 = Disable<br>01 = Interrupt request or polling mode<br>10 = DMA0 request (Only for UART0),<br>DMA3 request (Only for UART2)<br>11 = DMA1 request (Only for UART1) | 00 |

**Note:** When the UART does not reach the FIFO trigger level and does not receive data during 3 word time in DMA receive mode with FIFO, the Rx interrupt will be generated (receive time out), and the users should check the FIFO status and read out the rest.

### UART FIFO CONTROL REGISTER

There are three UART FIFO control registers including UFCON0, UFCON1 and UFCON2 in the UART block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UFCON0 | 0x50000008 | R/W | UART channel 0 FIFO control register | 0x0 |
| UFCON1 | 0x50004008 | R/W | UART channel 1 FIFO control register | 0x0 |
| UFCON2 | 0x50008008 | R/W | UART channel 2 FIFO control register | 0x0 |

| UFCONn | Bit | Description | Initial State |
|---|---|---|---|
| Tx FIFO Trigger Level | [7:6] | Determine the trigger level of transmit FIFO.<br>00 = Empty 01 = 16-byte<br>10 = 32-byte 11 = 48-byte | 00 |
| Rx FIFO Trigger Level | [5:4] | Determine the trigger level of receive FIFO.<br>00 = 1-byte 01 = 8-byte<br>10 = 16-byte 11 = 32-byte | 00 |
| Reserved | [3] | | 0 |
| Tx FIFO Reset | [2] | Auto-cleared after resetting FIFO<br>0 = Normal 1= Tx FIFO reset | 0 |
| Rx FIFO Reset | [1] | Auto-cleared after resetting FIFO<br>0 = Normal 1= Rx FIFO reset | 0 |
| FIFO Enable | [0] | 0 = Disable 1 = Enable | 0 |

**Note:** When the UART does not reach the FIFO trigger level and does not receive data during 3 word time in DMA receive mode with FIFO, the Rx interrupt will be generated (receive time out), and the users should check the FIFO status and read out the rest.

### UART MODEM CONTROL REGISTER

There are two UART MODEM control registers including UMCON0 and UMCON1 in the UART block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UMCON0 | 0x5000000C | R/W | UART channel 0 Modem control register | 0x0 |
| UMCON1 | 0x5000400C | R/W | UART channel 1 Modem control register | 0x0 |
| Reserved | 0x5000800C | - | Reserved | Undef |

| UMCONn | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:5] | These bits must be 0's | 00 |
| Auto Flow Control (AFC) | [4] | 0 = Disable 1 = Enable | 0 |
| Reserved | [3:1] | These bits must be 0's | 00 |
| Request to Send | [0] | If AFC bit is enabled, this value will be ignored. In this case the S3C2440X will control nRTS automatically.<br>If AFC bit is disabled, nRTS must be controlled by software.<br>0 = 'H' level (Inactivate nRTS) 1 = 'L' level (Activate nRTS) | 0 |

**Note:** UART 2 does not support AFC function, because the S3C2440X has no nRTS2 and nCTS2.

### UART TX/RX STATUS REGISTER

There are three UART Tx/Rx status registers including UTRSTAT0, UTRSTAT1 and UTRSTAT2 in the UART block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UTRSTAT0 | 0x50000010 | R | UART channel 0 Tx/Rx status register | 0x6 |
| UTRSTAT1 | 0x50004010 | R | UART channel 1 Tx/Rx status register | 0x6 |
| UTRSTAT2 | 0x50008010 | R | UART channel 2 Tx/Rx status register | 0x6 |

| UTRSTATn | Bit | Description | Initial State |
|---|---|---|---|
| Transmitter empty | [2] | Set to 1 automatically when the transmit buffer register has no valid data to transmit and the transmit shift register is empty.<br>0 = Not empty<br>1 = Transmitter (transmit buffer & shifter register) empty | 1 |
| Transmit buffer empty | [1] | Set to 1 automatically when transmit buffer register is empty.<br>0 =The buffer register is not empty<br>1 = Empty<br>(In Non-FIFO mode, Interrupt or DMA is requested. In FIFO mode, Interrupt or DMA is requested, when Tx FIFO Trigger Level is set to 00 (Empty))<br>If the UART uses the FIFO, users should check Tx FIFO Count bits and Tx FIFO Full bit in the UFSTAT register instead of this bit. | 1 |
| Receive buffer data ready | [0] | Set to 1 automatically whenever receive buffer register contains valid data, received over the RXDn port.<br>0 = Empty<br>1 = The buffer register has a received data<br>(In Non-FIFO mode, Interrupt or DMA is requested)<br>If the UART uses the FIFO, users should check Rx FIFO Count bits and Rx FIFO Full bit in the UFSTAT register instead of this bit. | 0 |

### UART ERROR STATUS REGISTER

There are three UART Rx error status registers including UERSTAT0, UERSTAT1 and UERSTAT2 in the UART block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UERSTAT0 | 0x50000014 | R | UART channel 0 Rx error status register | 0x0 |
| UERSTAT1 | 0x50004014 | R | UART channel 1 Rx error status register | 0x0 |
| UERSTAT2 | 0x50008014 | R | UART channel 2 Rx error status register | 0x0 |

| UERSTATn | Bit | Description | Initial State |
|---|---|---|---|
| Overrun Error | [0] | Set to 1 automatically whenever an overrun error occurs during receive operation.<br>0 = No overrun error during receive<br>1 = Overrun error (Interrupt is requested.) | 0 |

**Note:** This bit is automatically cleared to 0 when the UART error status register is read.

### UART FIFO STATUS REGISTER

There are three UART FIFO status registers including UFSTAT0, UFSTAT1 and UFSTAT2 in the UART block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UFSTAT0 | 0x50000018 | R | UART channel 0 FIFO status register | 0x00 |
| UFSTAT1 | 0x50004018 | R | UART channel 1 FIFO status register | 0x00 |
| UFSTAT2 | 0x50008018 | R | UART channel 2 FIFO status register | 0x00 |

| UFSTATn | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [15] | | 0 |
| Tx FIFO Full | [14] | Set to 1 automatically whenever transmit FIFO is full during transmit operation<br>0 = 0-byte ≤ Tx FIFO data ≤ 63-byte<br>1 = Full | 0 |
| Tx FIFO Count | [13:8] | Number of data in Tx FIFO | 0 |
| Reserved | [7] | | 0 |
| Rx FIFO Full | [6] | Set to 1 automatically whenever receive FIFO is full during receive operation<br>0 = 0-byte ≤ Rx FIFO data ≤ 63-byte<br>1 = Full | 0 |
| Rx FIFO Count | [5:0] | Number of data in Rx FIFO | 0 |

### UART MODEM STATUS REGISTER

There are two UART modem status registers including UMSTAT0, UMSTAT1 in the UART block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UMSTAT0 | 0x5000001C | R | UART channel 0 Modem status register | 0x0 |
| UMSTAT1 | 0x5000401C | R | UART channel 1 Modem status register | 0x0 |
| Reserved | 0x5000801C | - | Reserved | Undef |

| UMSTAT0 | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [3] | | 0 |
| Delta CTS | [2] | Indicate that the nCTS input to the S3C2440X has changed state since the last time it was read by CPU.<br>(Refer to Figure 11-8.)<br>0 = Has not changed<br>1 = Has changed | 0 |
| Reserved | [1] | | 0 |
| Clear to Send | [0] | 0 = CTS signal is not activated (nCTS pin is high)<br>1 = CTS signal is activated (nCTS pin is low) | 0 |

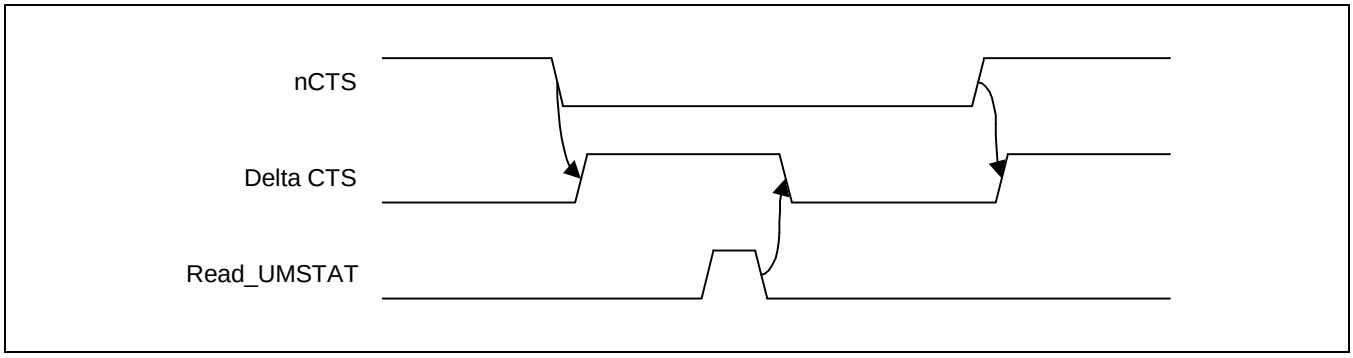



**Figure 11-7. nCTS and Delta CTS Timing Diagram**

### UART TRANSMIT BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are three UART transmit buffer registers including UTXH0, UTXH1 and UTXH2 in the UART block. UTXHn has an 8-bit data for transmission data.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UTXH0 | 0x50000020(L)<br>0x50000023(B) | W (by byte) | UART channel 0 transmit buffer register | - |
| UTXH1 | 0x50004020(L)<br>0x50004023(B) | W (by byte) | UART channel 1 transmit buffer register | - |
| UTXH2 | 0x50008020(L)<br>0x50008023(B) | W (by byte) | UART channel 2 transmit buffer register | - |

| UTXHn | Bit | Description | Initial State |
|---|---|---|---|
| TXDATAn | [7:0] | Transmit data for UARTn | - |

**Note:**

(L): The endian mode is Little endian.

(B): The endian mode is Big endian.

### UART RECEIVE BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are three UART receive buffer registers including URXH0, URXH1 and URXH2 in the UART block. URXHn has an 8-bit data for received data.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| URXH0 | 0x50000024(L)<br>0x50000027(B) | R (by byte) | UART channel 0 receive buffer register | - |
| URXH1 | 0x50004024(L)<br>0x50004027(B) | R (by byte) | UART channel 1 receive buffer register | - |
| URXH2 | 0x50008024(L)<br>0x50008027(B) | R (by byte) | UART channel 2 receive buffer register | - |

| URXHn | Bit | Description | Initial State |
|---|---|---|---|
| RXDATAn | [7:0] | Receive data for UARTn | - |

**NOTE:**

When an overrun error occurs, the URXHn must be read. If not, the next received data will also make an overrun error, even though the overrun bit of UERSTATn had been cleared.

### UART BAUD RATE DIVISOR REGISTER

There are three UART baud rate divisor registers including UBRDIV0, UBRDIV1 and UBRDIV2 in the UART block. The value stored in the baud rate divisor register (UBRDIVn), is used to determine the serial Tx/Rx clock rate (baud rate) as follows:

UBRDIVn = (int)(PCLK / (bps x 16) ) –1

or

UBRDIVn = (int)(UARTCLK / (bps x 16) ) –1

Where, the divisor should be from 1 to ($2^{16}$-1) and UARTCLK should be smaller than PCLK.

For example, if the baud-rate is 115200 bps and PCLK or UARTCLK is 40 MHz, UBRDIVn is:

UBRDIVn = (int)(40000000 / (115200 x 16) ) -1
= (int)(21.7) -1
= 21 -1 = 20

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| UBRDIV0 | 0x50000028 | R/W | Baud rate divisior register 0 | - |
| UBRDIV1 | 0x50004028 | R/W | Baud rate divisior register 1 | - |
| UBRDIV2 | 0x50008028 | R/W | Baud rate divisior register 2 | - |

| UBRDIV n | Bit | Description | Initial State |
|---|---|---|---|
| UBRDIV | [15:0] | Baud rate division value<br>UBRDIVn >0 | - |

# 12 USB HOST CONTROLLER

## OVERVIEW

S3C2440X supports 2-port USB host interface as follows:

- OHCI Rev 1.0 compatible
- USB Rev1.1 compatible
- Two down stream ports
- Support for both LowSpeed and HighSpeed USB devices

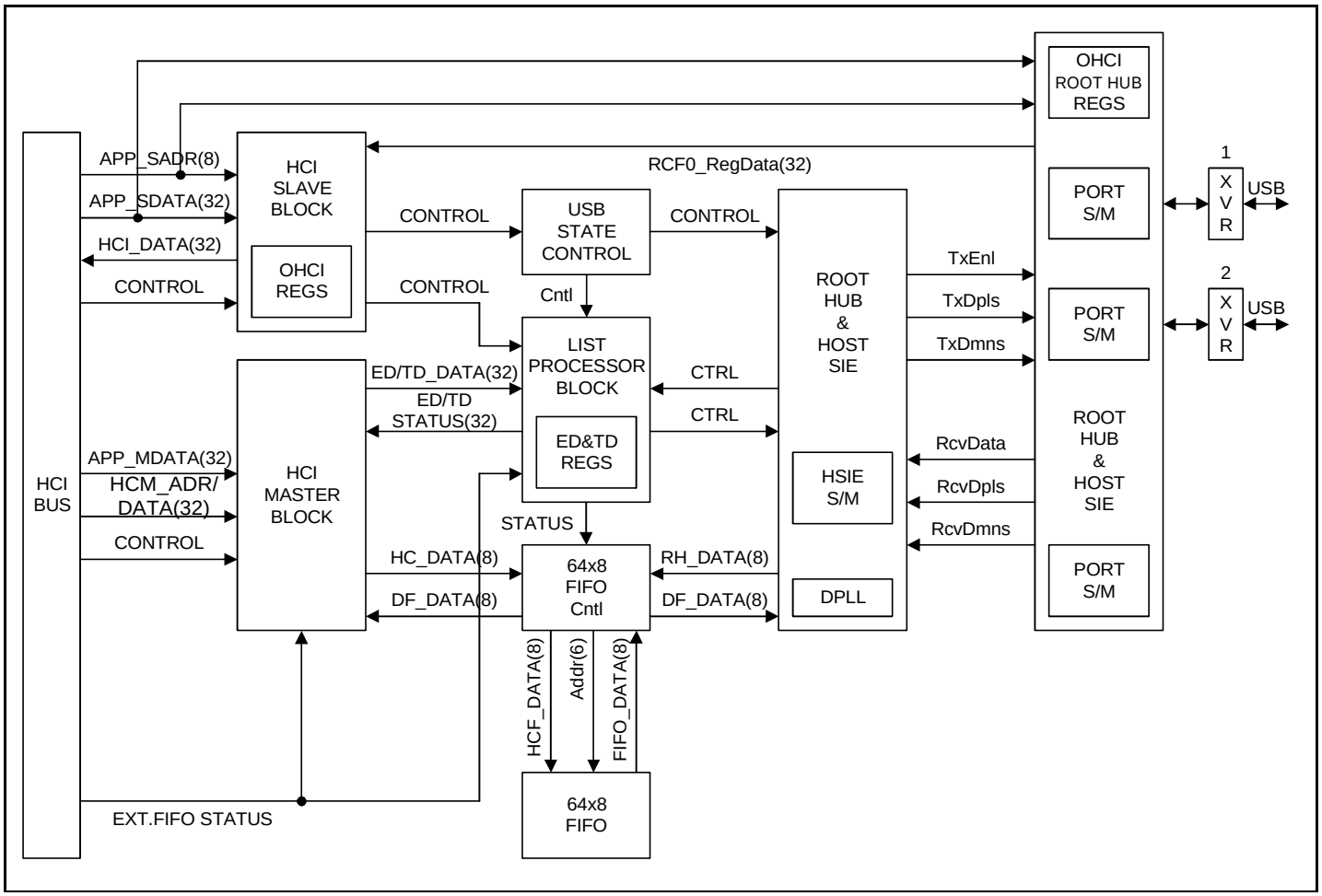



**Figure 12-1. USB Host Controller Block Diagram**

## USB HOST CONTROLLER SPECIAL REGISTERS

The S3C2440X USB cost controller complies with OHCI Rev 1.0. Refer to Open Host Controller Interface Rev 1.0 specification for detailed information.

### OHCI REGISTERS FOR USB HOST CONTROLLER

| Register | Base Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| HcRevision | 0x49000000 | – | Control and status group | – |
| HcControl | 0x49000004 | – | | – |
| HcCommonStatus | 0x49000008 | – | | – |
| HcInterruptStatus | 0x4900000C | – | | – |
| HcInterruptEnable | 0x49000010 | – | | – |
| HcInterruptDisable | 0x49000014 | – | | – |
| HcHCCA | 0x49000018 | – | Memory pointer group | – |
| HcPeriodCuttentED | 0x4900001C | – | | – |
| HcControlHeadED | 0x49000020 | – | | – |
| HcControlCurrentED | 0x49000024 | – | | – |
| HcBulkHeadED | 0x49000028 | – | | – |
| HcBulkCurrentED | 0x4900002C | – | | – |
| HcDoneHead | 0x49000030 | – | | – |
| HcRmInterval | 0x49000034 | – | Frame counter group | – |
| HcFmRemaining | 0x49000038 | – | | – |
| HcFmNumber | 0x4900003C | – | | – |
| HcPeriodicStart | 0x49000040 | – | | – |
| HcLSThreshold | 0x49000044 | – | | – |
| HcRhDescriptorA | 0x49000048 | – | Root hub group | – |
| HcRhDescriptorB | 0x4900004C | – | | – |
| HcRhStatus | 0x49000050 | – | | – |
| HcRhPortStatus1 | 0x49000054 | – | | – |
| HcRhPortStatus2 | 0x49000058 | – | | – |

# 13 USB DEVICE CONTROLLER

## OVERVIEW

Universal Serial Bus (USB) device controller is designed to provide a high performance full speed function controller solution with DMA interface. USB device controller allows bulk transfer with DMA, interrupt transfer and control transfer.

USB device controller supports:

- Full speed USB device controller compatible with the USB specification version 1.1
- DMA interface for bulk transfer
- Five endpoints with FIFO
  EP0: 16byte (Register)
  EP1: 128byte IN/OUT FIFO (dual port asynchronous RAM): interrupt or DMA
  EP2: 128byte IN/OUT FIFO (dual port asynchronous RAM): interrupt or DMA
  EP3: 128byte IN/OUT FIFO (dual port asynchronous RAM): interrupt or DMA
  EP4: 128byte IN/OUT FIFO (dual port asynchronous RAM): interrupt or DMA
- Integrated USB Transceiver

### FEATURE

— Fully compliant with USB Specification Version 1.1

— Full speed (12Mbps) device

— Integrated USB Transceiver

— Supports control, interrupt and bulk transfer

— Five endpoints with FIFO:
  One bi-directional control endpoint with 16-byte FIFO (EP0)
  Four bi-directional bulk endpoints with 128-byte FIFO (EP1, EP2, EP3, and EP4)

— Supports DMA interface for receive and transmit bulk endpoints. (EP1, EP2, EP3, and EP4)

— Independent 128-byte receive and transmit FIFO to maximize throughput

— Supports suspend and remote wakeup function

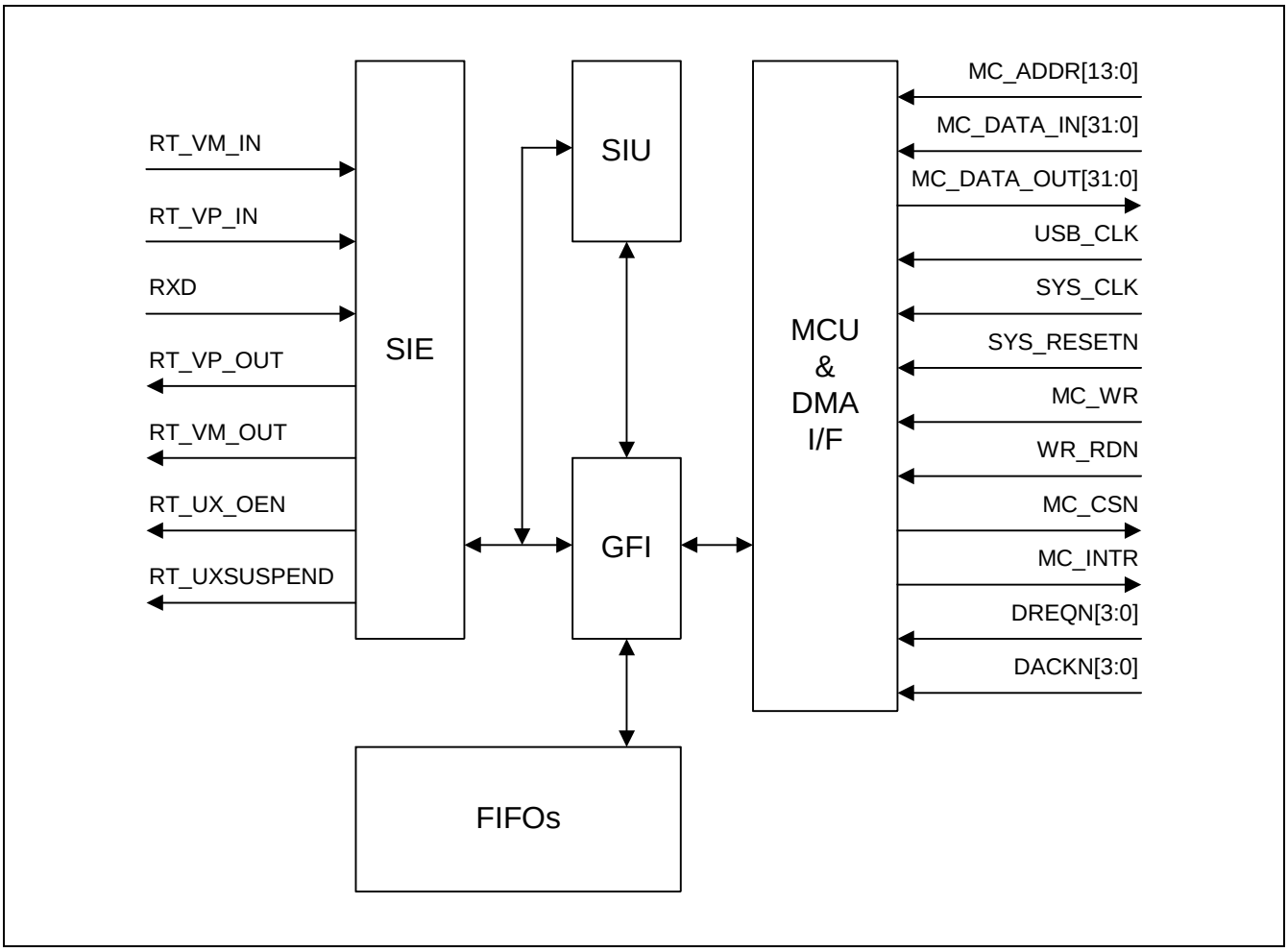



**Figure 13-1. USB Device Controller Block Diagram**

## USB DEVICE CONTROLLER SPECIAL REGISTERS

This section describes detailed functionalities about register sets of USB device controller.
All special function register is byte-accessible or word-accessible. If you access byte mode offset-address is different in little endian and big endian. All reserved bit is zero.
Common indexed registers depend on INDEX register (INDEX_REG) (offset address: 0X178) value. For example if you want to write EP0 CSR register, you must write '0x00' on the INDEX_REG before writing IN_CSR1 register.

**NOTE.** All register must be resettled after performing Host Reset Signaling.

| Register Name | Description | Offset Address |
|---|---|---|
| **NON INDEXED REGISTERS** | | |
| FUNC_ADDR_REG | Function address register | 0x140(L) / 0x143(B) |
| PWR_REG | Power management register | 0x144(L) / 0x147(B) |
| EP_INT_REG (EP0–EP4) | Endpoint interrupt register | 0x148(L) / 0x14B(B) |
| USB_INT_REG | USB interrupt register | 0x158(L) / 0x15B(B) |
| EP_INT_EN_REG (EP0–EP4) | Endpoint interrupt enable register | 0x15C(L) / 0x15F(B) |
| USB_INT_EN_REG | USB Interrupt enable register | 0x16C(L) / 0x16F(B) |
| FRAME_NUM1_REG | Frame number 1 register | 0x170(L) / 0x173(B) |
| FRAME_NUM2_REG | Frame number 2 register | 0x174(L) / 0x177(B) |
| INDEX_REG | Index register | 0x178(L) / 0x17B(B) |
| EP0_FIFO_REG | Endpoint0 FIFO register | 0x1C0(L) / 0x1C3(B) |
| EP1_FIFO_REG | Endpoint1 FIFO register | 0x1C4(L) / 0x1C7(B) |
| EP2_FIFO_REG | Endpoint2 FIFO register | 0x1C8(L) / 0x1CB(B) |
| EP3_FIFO_REG | Endpoint3 FIFO register | 0x1CC(L) / 0x1CF(B) |
| EP4_FIFO_REG | Endpoint4 FIFO register | 0x1D0(L) / 0x1D3(B) |
| EP1_DMA_CON | Endpoint1 DMA control register | 0x200(L) / 0x203(B) |
| EP1_DMA_UNIT | Endpoint1 DMA unit counter register | 0x204(L) / 0x207(B) |
| EP1_DMA_FIFO | Endpoint1 DMA FIFO counter register | 0x208(L) / 0x20B(B) |
| EP1_DMA_TTC_L | Endpoint1 DMA transfer counter low-byte register | 0x20C(L) / 0x20F(B) |
| EP1_DMA_TTC_M | Endpoint1 DMA transfer counter middle-byte register | 0x210(L) / 0x213(B) |
| EP1_DMA_TTC_H | Endpoint1 DMA transfer counter high-byte register | 0x214(L) / 0x217(B) |

| | | |
|---|---|---|
| EP2_DMA_CON | Endpoint2 DMA control register | 0x218(L) / 0x21B(B) |
| EP2_DMA_UNIT | Endpoint2 DMA unit counter register | 0x21C(L) / 0x21F(B) |
| EP2_DMA_FIFO | Endpoint2 DMA FIFO counter register | 0x220(L) / 0x223(B) |
| EP2_DMA_TTC_L | Endpoint2 DMA transfer counter low-byte register | 0x224(L) / 0x227(B) |
| EP2_DMA_TTC_M | Endpoint2 DMA transfer counter middle-byte register | 0x228(L) / 0x22B(B) |
| EP2_DMA_TTC_H | Endpoint2 DMA transfer counter high-byte register | 0x22C(L) / 0x22F(B) |
| EP3_DMA_CON | Endpoint3 DMA control register | 0x240(L) / 0x243(B) |
| EP3_DMA_UNIT | Endpoint3 DMA unit counter register | 0x244(L) / 0x247(B) |
| EP3_DMA_FIFO | Endpoint3 DMA FIFO counter register | 0x248(L) / 0x24B(B) |
| EP3_DMA_TTC_L | Endpoint3 DMA transfer counter low-byte register | 0x24C(L) / 0x24F(B) |
| EP3_DMA_TTC_M | Endpoint3 DMA transfer counter middle-byte register | 0x250(L) / 0x253(B) |
| EP3_DMA_TTC_H | Endpoint3 DMA transfer counter high-byte register | 0x254(L) / 0x247(B) |
| EP4_DMA_CON | Endpoint4 DMA control register | 0x258(L) / 0x25B(B) |
| EP4_DMA_UNIT | Endpoint4 DMA unit counter register | 0x25C(L) / 0x25F(B) |
| EP4_DMA_FIFO | Endpoint4 DMA FIFO counter register | 0x260(L) / 0x263(B) |
| EP4_DMA_TTC_L | Endpoint4 DMA transfer counter low-byte register | 0x264(L) / 0x267(B) |
| EP4_DMA_TTC_M | Endpoint4 DMA transfer counter middle-byte register | 0x268(L) / 0x26B(B) |
| EP4_DMA_TTC_H | Endpoint4 DMA transfer counter high-byte register | 0x26C(L) / 0x26F(B) |
| **COMMON INDEXED REGISTERS** | | |
| MAXP_REG | Endpoint MAX packet register | 0x180(L) / 0x183(B) |
| **IN INDEXED REGISTERS** | | |
| IN_CSR1_REG/EP0_CSR | EP In control status register 1/EP0 control status register | 0x184(L) / 0x187(B) |
| IN_CSR2_REG | EP In control status register 2 | 0x188(L) / 0x18B(B) |
| **OUT INDEXED REGISTERS** | | |
| OUT_CSR1_REG | EP out control status register 1 | 0x190(L) / 0x193(B) |
| OUT_CSR2_REG | EP out control status register 2 | 0x194(L) / 0x197(B) |
| OUT_FIFO_CNT1_REG | EP out write count register 1 | 0x198(L) / 0x19B(B) |
| OUT_FIFO_CNT2_REG | EP out write count register 2 | 0x19C(L) / 0x19F(B) |

### FUNCTION ADDRESS REGISTER (FUNC_ADDR_REG)

This register maintains the USB device controller address assigned by the host. The Micro Controller Unit (MCU) writes the value received through a SET_ADDRESS descriptor to this register. This address is used for the next token.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| FUNC_ADDR_REG | 0x52000140(L)<br>0x52000143(B) | R/W<br>(byte) | Function address register | 0x00 |

| FUNC_ADDR_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| ADDR_UPDATE | [7] | R<br>/SET | R<br>/CLEAR | Set by the MCU whenever it updates the FUNCTION_ADDR field in this register. This bit will be cleared by USB when DATA_END bit in EP0_CSR register. | 0 |
| FUNCTION_ADDR | [6:0] | R/W | R | The MCU write the unique address, assigned by host, to this field. | 00 |

### POWER MANAGEMENT REGISTER (PWR_REG)

This register acts as a power control register in the USB block.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| PWR_REG | 0x52000144(L)<br>0x52000147(B) | R/W<br>(byte) | Power management register | 0x00 |

| PWR_ADDR | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| Reserved | [7:4] | - | - | - | - |
| USB_RESET | [3] | R | SET | Set by the USB if reset signaling is received from the host. This bit remains set as long as reset signaling persists on the bus | 0 |
| MCU_RESUME | [2] | R/W | R<br>/CLEAR | Set by the MCU for MCU Resume.<br>The USB generates the resume signaling during 10ms, if this bit is set in suspend mode. | |
| SUSPEND_MODE | [1] | R | SET<br>/CLEAR | Set by USB automatically when the device enter into suspend mode.<br>It is cleared under the following conditions:<br>1) The MCU clears the MCU_RESUME bit by writing ‘0’, in order to end remote resume signaling.<br>2) The resume signal from host is received. | 0 |
| SUSPEND_EN | [0] | R/W | R | Suspend mode enable control bit<br>0 = Disable (default). The device will not enter suspend mode.<br>1 = Enable suspend mode | 0 |

### INTERRUPT REGISTER (EP_INT_REG/USB_INT_REG)

The USB core has two interrupt registers.
These registers act as status registers for the MCU when it is interrupted. The bits are cleared by writing a '1' (not '0') to each bit that was set.
Once the MCU is interrupted, MCU should read the contents of interrupt-related registers and write back to clear the contents if it is necessary.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP_INT_REG | 0x52000148(L)<br>0x5200014B(B) | R/W<br>(byte) | EP interrupt pending/clear register | 0x00 |

| EP_INT_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EP1~EP4 Interrupt | [4:1] | R /CLEAR | SET | **For BULK/INTERRUPT IN endpoints:**<br>Set by the USB under the following conditions:<br>1. IN_PKT_RDY bit is cleared.<br>2. FIFO is flushed<br>3. SENT_STALL set.<br><br>**For BULK/INTERRUPT OUT endpoints:**<br>Set by the USB under the following conditions:<br>1. Sets OUT_PKT_RDY bit<br>2. Sets SENT_STALL bit | 0 |
| EP0 Interrupt | [0] | R /CLEAR | SET | Correspond to endpoint 0 interrupt.<br>Set by the USB under the following conditions:<br>1. OUT_PKT_RDY bit is set.<br>2. IN_PKT_RDY bit is cleared.<br>3. SENT_STALL bit is set<br>4. SETUP_END bit is set<br>5. DATA_END bit is cleared (it indicates the end of control transfer). | 0 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| USB_INT_REG | 0x52000158(L)<br>0x5200015B(B) | R/W<br>(byte) | USB interrupt pending/clear register | 0x00 |

| USB_INT_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| RESET Interrupt | [2] | R /CLEAR | SET | Set by the USB when it receives reset signaling. | 0 |
| RESUME Interrupt | [1] | R /CLEAR | SET | Set by the USB when it receives resume signaling, while in Suspend mode.<br><br>If the resume occurs due to a USB reset, then the MCU is first interrupted with a RESUME interrupt. Once the clocks resume and the SE0 condition persists for 3ms, USB RESET interrupt will be asserted. | 0 |
| SUSPEND Interrupt | [0] | R /CLEAR | SET | Set by the USB when it receives suspend signalizing.<br><br>This bit is set whenever there is no activity for 3ms on the bus. Thus, if the MCU does not stop the clock after the first suspend interrupt, it will continue to be interrupted every 3ms as long as there is no activity on the USB bus.<br><br>By default, this interrupt is disabled. | 0 |

### INTERRUPT ENABLE REGISTER (EP_INT_EN_REG/USB_INT_EN_REG)

Corresponding to each interrupt register, The USB device controller also has two interrupt enable registers (except resume interrupt enable). By default, usb reset interrupt is enabled.

If bit = 0, the interrupt is disabled.
If bit = 1, the interrupt is enabled.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP_INT_EN_REG | 0x5200015C(L)<br>0x5200015F(B) | R/W<br>(byte) | Determine which interrupt is enabled | 0xFF |

| EP_INT_EN_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EP4_INT_EN | [4] | R/W | R | EP4 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| EP3_INT_EN | [3] | R/W | R | EP3 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| EP2_INT_EN | [2] | R/W | R | EP2 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| EP1_INT_EN | [1] | R/W | R | EP1 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| EP0_INT_EN | [0] | R/W | R | EP0 Interrupt Enable bit<br>0 = Interrupt disable 1 = Enable | 1 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| USB_INT_EN_REG | 0x520016C(L)<br>0x5200016F(B) | R/W<br>(byte) | Determine which interrupt is enabled | 0x04 |

| INT_MASK_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| RESET_INT_EN | [2] | R/W | R | Reset interrupt enable bit<br>0 = Interrupt disable 1 = Enable | 1 |
| Reserved | [1] | - | - | - | 0 |
| SUSPEND_INT_EN | [0] | R/W | R | Suspend interrupt enable bit<br>0 = Interrupt disable 1 = Enable | 0 |

### FRAME NUMBER REGISTER (FPAME_NUM1_REG/FRAME_NUM2_REG)

When the host transfers USB packets, each Start Of Frame (SOF) packit includes a frame number. The USB device controller catches this frame number and loads it into this register automatically.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| FRAME_NUM1_REG | 0x52000170(L)<br>0x52000173(B) | R<br>(byte) | Frame number lower byte register | 0x00 |

| FRAME_NUM_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| FRAME_NUM1 | [7:0] | R | W | Frame number lower byte value | 00 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| FRAME_NUM2_REG | 0x52000174(L)<br>0x52000177(B) | R<br>(byte) | Frame number higher byte register | 0x00 |

| FRAME_NUM_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| FRAME_NUM2 | [7:0] | R | W | Frame number higher byte value | 00 |

### INDEX REGISTER (INDEX_REG)

The INDEX register is used to indicate certain endpoint registers effectively. The MCU can access the endpoint registers (MAXP_REG, IN_CSR1_REG, IN_CSR2_REG, OUT_CSR1_REG, OUT_CSR2_REG, OUT_FIFO_CNT1_REG, and OUT_FIFO_CNT2_REG) for an endpoint inside the core using the INDEX register.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INDEX_REG | 0x52000178(L)<br>0x5200017B(B) | R/W<br>(byte) | Register index register | 0x00 |

| INDEX_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| INDEX | [7:0] | R/W | R | Indicate a certain endpoint | 00 |

### MAX PACKET REGISTER (MAXP_REG)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| MAXP_REG | 0x52000180(L)<br>0x52000183(B) | R/W<br>(byte) | End Point MAX packet register | 0x01 |

| MAXP_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| MAXP | [3:0] | R/W | R | 0000: Reserved<br>0001: MAXP = 8 Byte 0010: MAXP = 16 Byte<br>0100: MAXP = 32 Byte 1000: MAXP = 64 Byte<br>For EP0, MAXP=8 is recommended.<br>For EP1~4, MAXP=64 is recommended. And, if MAXP=64, the dual packet mode will be enabled automatically. | 0001 |

### END POINT0 CONTROL STATUS REGISTER (EP0_CSR)

This register has the control and status bits for Endpoint 0. Since a control transaction is involved with both IN and OUT tokens, there is only one CSR register, mapped to the IN CSR1 register. (share IN1_CSR and can access by writing index register “0” and read/write IN1_CSR)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP0_CSR | 0x52000184(L)<br>0x52000187(B) | R/W<br>(byte) | Endpoint 0 status register | 0x00 |

| EP0_CSR | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| SERVICED_SETUP_END | [7] | W | CLEAR | The MCU should write a "1" to this bit to clear SETUP_END. | 0 |
| SERVICED_OUT_PKT_RDY | [6] | W | CLEAR | The MCU should write a "1" to this bit to clear OUT_PKT_RDY. | 0 |
| SEND_STALL | [5] | R/W | CLEAR | MCU should write a "1" to this bit at the same time it clears OUT_PKT_RDY, if it decodes an invalid token.<br>0 = Finish the STALL condition<br>1 = The USB issues a STALL and shake to the current control transfer. | 0 |
| SETUP_END | [4] | R | SET | Set by the USB when a control transfer ends before DATA_END is set.<br>When the USB sets this bit, an interrupt is generated to the MCU.<br>When such a condition occurs, the USB flushes the FIFO and invalidates MCU access to the FIFO. | 0 |
| DATA_END | [3] | SET | CLEAR | Set by the MCU on the conditions below:<br>1. After loading the last packet of data into the FIFO, at the same time IN_PKT_RDY is set.<br>2. While it clears OUT_PKT_RDY after unloading the last packet of data.<br>3. For a zero length data phase. | 0 |
| SENT_STALL | [2] | CLEAR | SET | Set by the USB if a control transaction is stopped due to a protocol violation. An interrupt is generated when this bit is set. The MCU should write "0" to clear this bit. | 0 |
| IN_PKT_RDY | [1] | SET | CLEAR | Set by the MCU after writing a packet of data into EP0 FIFO. The USB clears this bit once the packet has been successfully sent to the host. An interrupt is generated when the USB clears this bit, so as the MCU to load the next packet. For a zero length data phase, the MCU sets DATA_END at the same time. | 0 |
| OUT_PKT_RDY | [0] | R | SET | Set by the USB once a valid token is written to the FIFO. An interrupt is generated when the USB sets this bit. The MCU clears this bit by writing a "1" to the SERVICED_OUT_PKT_RDY bit. | 0 |

**END POINT IN CONTROL STATUS REGISTER (IN_CSR1_REG/IN_CSR2_REG)**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IN_CSR1_REG | 0x52000184(L)<br>0x52000187(B) | R/W (byte) | IN END POINT control status register1 | 0x00 |

| IN_CSR1_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| Reserved | [7] | - | - | - | 0 |
| CLR_DATA_ TOGGLE | [6] | R/W | R/ CLEAR | Used in Set-up procedure.<br>0: There are alternation of DATA0 and DATA1<br>1: The data toggle bit is cleared and PID in packet will maintain DATA0 | 0 |
| SENT_STALL | [5] | R/ CLEAR | SET | Set by the USB when an IN token issues a STALL handshake, after the MCU sets SEND_STALL bit to start STALL handshaking. When the USB issues a STALL handshake, IN_PKT_RDY is cleared | 0 |
| SEND_STALL | [4] | W/R | R | 0: The MCU clears this bit to finish the STALL condition.<br>1: The MCU issues a STALL handshake to the USB. | 0 |
| FIFO_FLUSH | [3] | R/W | CLEAR | Set by the MCU if it intends to flush the packet in Input-related FIFO. This bit is cleared by the USB when the FIFO is flushed. The MCU is interrupted when this happens. If a token is in process, the USB waits until the transmission is complete before FIFO flushing. If two packets are loaded into the FIFO, only first packet (The packet is intended to be sent to the host) is flushed, and the corresponding IN_PKT_RDY bit is cleared | 0 |
| Reserved | [2:1] | - | - | - | 0 |
| IN_PKT_RDY | [0 | R/SET | CLEAR | Set by the MCU after writing a packet of data into the FIFO.<br>The USB clears this bit once the packet has been successfully sent to the host.<br>An interrupt is generated when the USB clears this bit, so the MCU can load the next packet. While this bit is set, the MCU will not be able to write to the FIFO.<br>If the MCU sets SEND STALL bit, this bit cannot be set. | 0 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IN_CSR2_REG | 0x52000188(L)<br>0x5200018B(B) | R/W<br>(byte) | IN END POINT control status register2 | 0x20 |

| IN_CSR2_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| AUTO_SET | [7] | R/W | R | If set, whenever the MCU writes MAXP data, IN_PKT_RDY will automatically be set by the core without any intervention from MCU.<br>If the MCU writes less than MAXP data, IN_PKT_RDY bit has to be set by the MCU. | 0 |
| ISO | [6] | R/W | R | Used only for endpoints whose transfer type is programmable.<br>1: Reserved<br>0: Configures endpoint to Bulk mode | 0 |
| MODE_IN | [5] | R/W | R | Used only for endpoints whose direction is programmable.<br>1: Configures Endpoint Direction as IN<br>0: Configures Endpoint Direction as OUT | 1 |
| IN_DMA_INT_EN | [4] | R/W | R | Determine whether the interrupt should be issued or not, when the EP1 IN_PKT_RDY condition happens. This is only useful for DMA mode.<br>0 = Interrupt enable, 1 = Interrupt Disable | 0 |
| Reserved | [3:0] | - | - | - | - |

**END POINT OUT CONTROL STATUS REGISTER (OUT_CSR1_REG/OUT_CSR2_REG)**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| OUT_CSR1_REG | 0x52000190(L)<br>0x52000193(B) | R/W (byte) | End Point out control status register1 | 0x00 |

| OUT_CSR1_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| CLR_DATA_TOGGLE | [7] | R/W | CLEAR | When the MCU writes a 1 to this bit, the data toggle sequence bit is reset to DATA0. | 0 |
| SENT_STALL | [6] | CLEAR /R | SET | Set by the USB when an OUT token is ended with a STALL handshake. The USB issues a stall handshake to the host if it sends more than MAXP data for the OUT TOKEN. | 0 |
| SEND_STALL | [5] | R/W | R | 0: The MCU clears this bit to end the STALL condition handshake, IN PKT RDY is cleared.<br>1: The MCU issues a STALL handshake to the USB. The MCU clears this bit to end the STALL condition handshake, IN PKT RDY is cleared. | 0 |
| FIFO_FLUSH | [4] | R/W | CLEAR | The MCU writes a 1 to flush the FIFO.<br>This bit can be set only when OUT_PKT_RDY (D0) is set. The packet due to be unloaded by the MCU will be flushed. | 0 |
| Reserved | [3:1] | - | - | - | 0 |
| OUT_PKT_RDY | [0] | R/ CLEAR | SET | Set by the USB after it has loaded a packet of data into the FIFO. Once the MCU reads the packet from FIFO, this bit should be cleared by MCU (write a "0"). | 0 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| OUT_CSR2_REG | 0x52000194(L)<br>0x52000197(B) | R/W (byte) | End Point out control status register2 | 0x00 |

| OUT_CSR2_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| AUTO_CLR | [7] | R/W | R | If the MCU is set, whenever the MCU reads data from the OUT FIFO, OUT_PKT_RDY will automatically be cleared by the logic without any intervention from the MCU. | 0 |
| ISO | [6] | R/W | R | Determine endpoint transfer type.<br>0: Configures endpoint to Bulk mode.<br>1: Reserved. | 0 |
| OUT_DMA_INT_MASK | [5] | R/W | R | Determine whether the interrupt should be issued or not.<br>OUT_PKT_RDY condition happens. This is only useful for DMA mode<br>0 = Interrupt Enable 1 = Interrupt Disable | 0 |

**END POINT OUT WRITE COUNT REGISTER (OUT_FIFO_CNT1_REG/OUT_FIFO_CNT2_REG)**

These registers maintain the number of bytes in the packet as the number is unloaded by the MCU.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| OUT_FIFO_CNT1_REG | 0x52000198(L)<br>0x5200019B(B) | R<br>(byte) | End Point out write count register1 | 0x00 |

| OUT_FIFO_CNT1_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| OUT_CNT_LOW | [7:0] | R | W | Lower byte of write count | 0x00 |

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| OUT_FIFO_CNT2_REG | 0x5200019C(L)<br>0x5200019F(B) | R<br>(byte) | End Point out write count register2 | 0x00 |

| OUT_FIFO_CNT2_REG | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| OUT_CNT_HIGH | [7:0] | R | W | Higher byte of write count. The OUT_CNT_HIGH may be always 0 normally. | 0x00 |

**END POINT FIFO REGISTER (EPN_FIFO_REG)**

The EPn_FIFO_REG enables the MCU to access to the EPn FIFO.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP0_FIFO | 0x520001C0(L)<br>0x520001C3 (B) | R/W<br>(byte) | End Point0 FIFO register | 0xXX |
| EP1_FIFO | 0x520001C4(L)<br>0x520001C7(B) | R/W<br>(byte) | End Point1 FIFO register | 0xXX |
| EP2_FIFO | 0x520001C8(L)<br>0x520001CB(B) | R/W<br>(byte) | End Point2 FIFO register | 0xXX |
| EP3_FIFO | 0x520001CC(L)<br>0x520001CF(B) | R/W<br>(byte) | End Point3 FIFO register | 0xXX |
| EP4_FIFO | 0x520001D0(L)<br>0x520001D3(B) | R/W<br>(byte) | End Point4 FIFO register | 0xXX |

| EPn_FIFO | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| FIFO_DATA | [7:0] | R/W | R/W | FIFO data value | 0xXX |

**DMA INTERFACE CONTROL REGISTER (EPN_DMA_CON)**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP1_DMA_CON | 0x52000200(L)<br>0x52000203(B) | R/W<br>(byte) | EP1 DMA interface control register | 0x00 |
| EP2_DMA_CON | 0x52000218(L)<br>0x5200021B(B) | R/W<br>(byte) | EP2 DMA interface control register | 0x00 |
| EP3_DMA_CON | 0x52000240(L)<br>0x52000243(B) | R/W<br>(byte) | EP3 DMA interface control register | 0x00 |
| EP4_DMA_CON | 0x52000258(L)<br>0x5200025B(B) | R/W<br>(byte) | EP4 DMA interface control register | 0x00 |

| EPn_DMA_CON | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| RUN_OB | [7] | R/W | W | Read) DMA Run Observation<br>0: DMA is stopped 1:DMA is running<br>Write) Ignore EPn_DMA_TTC_n register<br>0: DMA requests will be stopped if EPn_DMA_TTC_n reaches 0.<br>1: DMA requests will be continued although EPn_DMA_TTC_n reaches 0. | 0 |
| STATE | [6:4] | R | W | DMA State Monitoring | 0 |
| DEMAND_MODE | [3] | R/W | R | DMA Demand mode enable bit<br>0: Demand mode disable<br>1: Demand mode enable | 0 |
| OUT_RUN_OB / OUT_DMA_RUN | [2] | R/W | R/W | Functionally separated into write and read operation.<br>Write operation: '0' = Stop '1' = Run<br>Read operation: OUT DMA Run Observation | 0 |
| IN_DMA_RUN | [1] | R/W | R | Start DMA operation.<br>0 = Stop 1 = Run | 0 |
| DMA_MODE_EN | [0] | R/W | R/Clear | Set DMA mode.If the RUN_OB has been wrtten as 0 and EPn_DMA_TTC_n reaches 0, DMA_MODE_EN bit will be cleared by USB.<br>0 = Interrupt Mode<br>1 = DMA Mode | 0 |

### DMA UNIT COUNTER REGISTER (EPN_DMA_UNIT)

This register is valid in Demand mode. In other modes, this register value must be set to '0x01'

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP1_DMA_UNIT | 0x52000204(L)<br>0x52000207(B) | R/W<br>(byte) | EP1 DMA transfer unit counter base register | 0x00 |
| EP2_DMA_UNIT | 0x5200021C(L)<br>0x5200021F(B) | R/W<br>(byte) | EP2 DMA transfer unit counter base register | 0x00 |
| EP3_DMA_UNIT | 0x52000244(L)<br>0x52000247(B) | R/W<br>(byte) | EP3 DMA transfer unit counter base register | 0x00 |
| EP4_DMA_UNIT | 0x5200025C(L)<br>0x5200025F(B) | R/W<br>(byte) | EP4 DMA transfer unit counter base register | 0x00 |

| DMA_UNIT | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EPn_UNIT_CNT | [7:0] | R/W | R | EP DMA transfer unit counter value | 0x00 |

### DMA FIFO COUNTER REGISTER (EPN_DMA_FIFO)

This register has values in byte size in FIFO to be transferred by DMA. In case of OUT_DMA_RUN enabled, the value in OUT FIFO Write Count Register1 will be loaded in this register automatically. In case of IN DMA mode, the MCU should set proper value by software.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP1_DMA_FIFO | 0x52000208(L)<br>0x5200020B(B) | R/W (byte) | EP1 DMA transfer FIFO counter base register | 0x00 |
| EP2_DMA_FIFO | 0x52000220(L)<br>0x52000223(B) | R/W (byte) | EP2 DMA transfer FIFO counter base register | 0x00 |
| EP3_DMA_FIFO | 0x52000248(L)<br>0x5200024B(B) | R/W (byte) | EP3 DMA transfer FIFO counter base register | 0x00 |
| EP4_DMA_FIFO | 0x52000260(L)<br>0x52000263(B) | R/W (byte) | EP4 DMA transfer FIFO counter base register | 0x00 |

| DMA_FIFO | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EPn_FIFO_CNT | [7:0] | R/W | R | EP DMA transfer FIFO counter value | 0x00 |

### DMA TOTAL TRANSFER COUNTER REGISTER (EPn_DMA_TTC_L,M,H)

This register should have total number of bytes to be transferred using DMA (total 20-bit counter).

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| EP1_DMA_TTC_L | 0x5200020C(L)<br>0x5200020F(B) | R/W (byte) | EP1 DMA total transfer counter(lower byte) | 0x00 |
| EP1_DMA_TTC_M | 0x52000210(L)<br>0x52000213(B) | R/W (byte) | EP1 DMA total transfer counter(middle byte) | 0x00 |
| EP1_DMA_TTC_H | 0x52000214(L)<br>0x52000217(B) | R/W (byte) | EP1 DMA total transfer counter(higher byte) | 0x00 |
| EP2_DMA_TTC_L | 0x52000224(L)<br>0x52000227(B) | R/W (byte) | EP2 DMA total transfer counter(lower byte) | 0x00 |
| EP2_DMA_TTC_M | 0x52000228(L)<br>0x5200022B(B) | R/W (byte) | EP2 DMA total transfer counter(middle byte) | 0x00 |
| EP2_DMA_TTC_H | 0x5200022C(L)<br>0x5200022F(B) | R/W (byte) | EP2 DMA total transfer counter(higher byte) | 0x00 |
| EP3_DMA_TTC_L | 0x5200024C(L)<br>0x5200024F(B) | R/W (byte) | EP3 DMA total transfer counter(lower byte) | 0x00 |
| EP3_DMA_TTC_M | 0x52000250(L)<br>0x52000253(B) | R/W (byte) | EP3 DMA total transfer counter(middle byte) | 0x00 |
| EP3_DMA_TTC_H | 0x52000254(L)<br>0x52000257(B) | R/W (byte) | EP3 DMA total transfer counter(higher byte) | 0x00 |
| EP4_DMA_TTC_L | 0x52000264(L)<br>0x52000267(B) | R/W (byte) | EP4 DMA total transfer counter(lower byte) | 0x00 |
| EP4_DMA_TTC_M | 0x52000268(L)<br>0x5200026B(B) | R/W (byte) | EP4 DMA total transfer counter(middle byte) | 0x00 |
| EP4_DMA_TTC_H | 0x5200026C(L)<br>0x5200026F(B) | R/W (byte) | EP4 DMA total transfer counter(higher byte) | 0x00 |

| DMA_TX | Bit | MCU | USB | Description | Initial State |
|---|---|---|---|---|---|
| EPn_TTC_L | [7:0] | R/W | R | DMA total transfer count value (lower byte) | 0x00 |
| EPn_TTC_M | [7:0] | R/W | R | DMA total transfer count value (middle byte) | 0x00 |
| EPn_TTC_H | [3:0] | R/W | R | DMA total transfer count value (higher byte) | 0x00 |

# 14 INTERRUPT CONTROLLER

## OVERVIEW

The interrupt controller in the S3C2440X receives the request from 59 interrupt sources. These interrupt sources are provided by internal peripherals such as the DMA controller, the UART, IIC, and others. In these interrupt sources, the UARTn and EINTn interrupts are 'OR'ed to the interrupt controller.

When receiving multiple interrupt requests from internal peripherals and external interrupt request pins, the interrupt controller requests FIQ or IRQ interrupt of the ARM920T core after the arbitration procedure.

The arbitration procedure depends on the hardware priority logic and the result is written to the interrupt pending register, which helps users notify which interrupt is generated out of various interrupt sources.

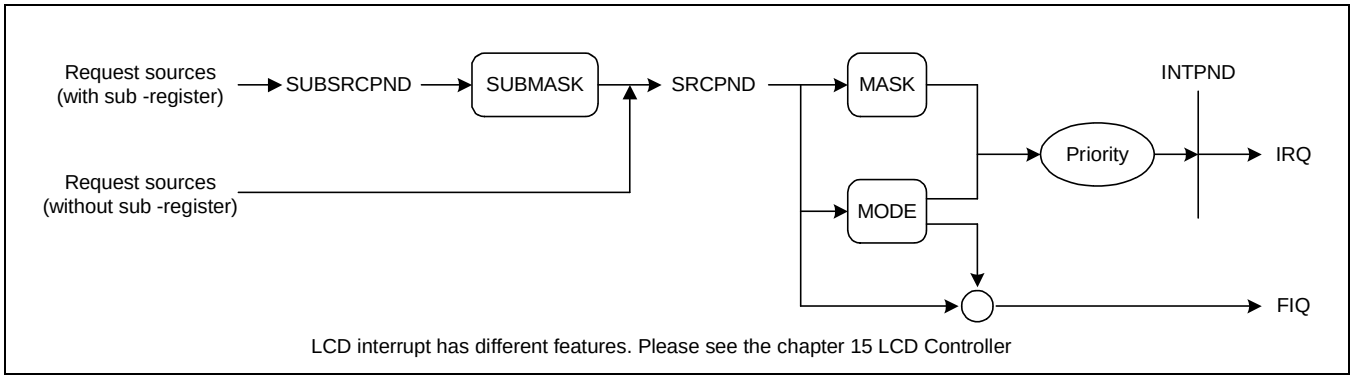



**Figure 14-1. Interrupt Process Diagram**

## INTERRUPT CONTROLLER OPERATION

### F-bit and I-bit of Program Status Register (PSR)

If the F-bit of PSR in ARM920T CPU is set to 1, the CPU does not accept the Fast Interrupt Request (FIQ) from the interrupt controller. Likewise, If I-bit of the PSR is set to 1, the CPU does not accept the Interrupt Request (IRQ) from the interrupt controller. So, the interrupt controller can receive interrupts by clearing F-bit or I-bit of the PSR to 0 and setting the corresponding bit of INTMSK to 0.

### Interrupt Mode

The ARM920T has two types of Interrupt mode: FIQ or IRQ. All the interrupt sources determine which mode is used at interrupt request.

### Interrupt Pending Register

The S3C2440X has two interrupt pending resisters: source pending register (SRCPND) and interrupt pending register (INTPND). These pending registers indicate whether or not an interrupt request is pending. When the interrupt sources request interrupt service, the corresponding bits of SRCPND register are set to 1, and at the same time, only one bit of the INTPND register is set to 1 automatically after arbitration procedure. If interrupts are masked, the corresponding bits of the SRCPND register are set to 1. This does not cause the bit of INTPND register changed. When a pending bit of the INTPND register is set, the interrupt service routine starts whenever the I-flag or F-flag is cleared to 0. The SRCPND and INTPND registers can be read and written, so the service routine must clear the pending condition by writing a 1 to the corresponding bit in the SRCPND register first and then clear the pending condition in the INTPND registers by using the same method.

### Interrupt Mask Register

This register indicates that an interrupt has been disabled if the corresponding mask bit is set to 1. If an interrupt mask bit of INTMSK is 0, the interrupt will be serviced normally. If the corresponding mask bit is 1 and the interrupt is generated, the source pending bit will be set.

## INTERRUPT SOURCES

The interrupt controller supports 59 interrupt sources as shown in the table below.

| Sources | Descriptions | Arbiter Group |
|---|---|---|
| INT_ADC | ADC EOC and Touch interrupt (INT_ADC/INT_TC) | ARB5 |
| INT_RTC | RTC alarm interrupt | ARB5 |
| INT_SPI1 | SPI1 interrupt | ARB5 |
| INT_UART0 | UART0 Interrupt (ERR, RXD, and TXD) | ARB5 |
| INT_IIC | IIC interrupt | ARB4 |
| INT_USBH | USB Host interrupt | ARB4 |
| INT_USBD | USB Device interrupt | ARB4 |
| INT_NFCON | Nand Flash Control Interrupt | ARB4 |
| INT_UART1 | UART1 Interrupt (ERR, RXD, and TXD) | ARB4 |
| INT_SPI0 | SPI0 interrupt | ARB4 |
| INT_SDI | SDI interrupt | ARB 3 |
| INT_DMA3 | DMA channel 3 interrupt | ARB3 |
| INT_DMA2 | DMA channel 2 interrupt | ARB3 |
| INT_DMA1 | DMA channel 1 interrupt | ARB3 |
| INT_DMA0 | DMA channel 0 interrupt | ARB3 |
| INT_LCD | LCD interrupt (INT_FrSyn and INT_FiCnt) | ARB3 |
| INT_UART2 | UART2 Interrupt (ERR, RXD, and TXD) | ARB2 |
| INT_TIMER4 | Timer4 interrupt | ARB2 |
| INT_TIMER3 | Timer3 interrupt | ARB2 |
| INT_TIMER2 | Timer2 interrupt | ARB2 |
| INT_TIMER1 | Timer1 interrupt | ARB 2 |
| INT_TIMER0 | Timer0 interrupt | ARB2 |
| INT_WDT | Watch-Dog timer interrupt | ARB1 |
| INT_TICK | RTC Time tick interrupt | ARB1 |
| nBATT_FLT | Battery Fault interrupt | ARB1 |
| INT_CAM | Camera Interface (INT_CAM_S, INT_CAM_C) | ARB1 |
| EINT8_23 | External interrupt 8 – 23 | ARB1 |
| EINT4_7 | External interrupt 4 – 7 | ARB1 |
| EINT3 | External interrupt 3 | ARB0 |
| EINT2 | External interrupt 2 | ARB0 |
| EINT1 | External interrupt 1 | ARB0 |
| EINT0 | External interrupt 0 | ARB0 |

## INTERRUPT PRIORITY GENERATING BLOCK

The priority logic for 32 interrupt requests is composed of seven rotation based arbiters: six first-level arbiters and one second-level arbiter as shown in Figure 14-1 below.

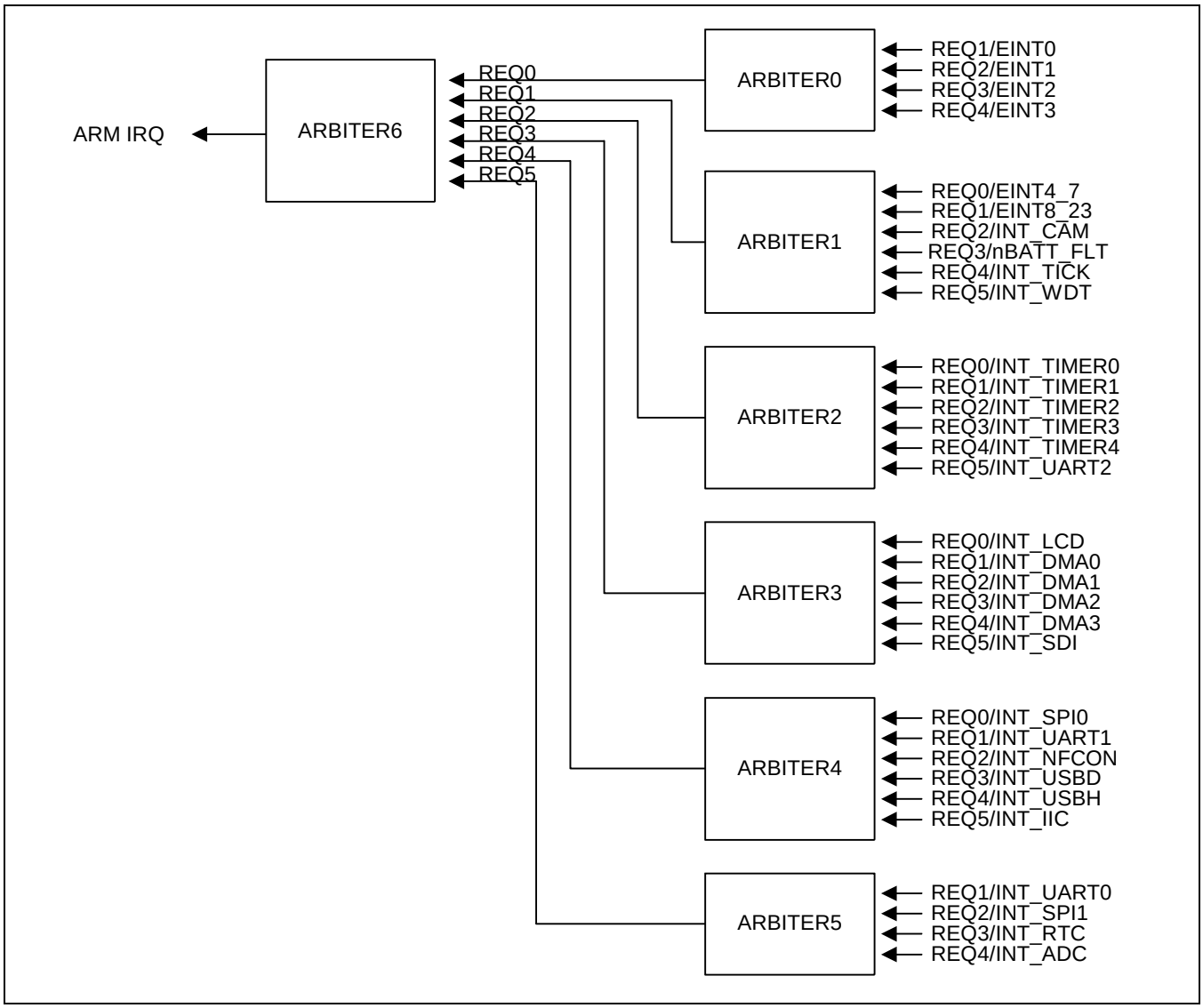



**Figure 14-2. Priority Generating Block**

## INTERRUPT PRIORITY

Each arbiter can handle six interrupt requests based on the one bit arbiter mode control (ARB_MODE) and two bits of selection control signals (ARB_SEL) as follows:

If ARB_SEL bits are 00b, the priority order is REQ0, REQ1, REQ2, REQ3, REQ4, and REQ5.

If ARB_SEL bits are 01b, the priority order is REQ0, REQ2, REQ3, REQ4, REQ1, and REQ5.

If ARB_SEL bits are 10b, the priority order is REQ0, REQ3, REQ4, REQ1, REQ2, and REQ5.

If ARB_SEL bits are 11b, the priority order is REQ0, REQ4, REQ1, REQ2, REQ3, and REQ5.

Note that REQ0 of an arbiter always has the highest priority, and REQ5 has the lowest one. In addition, by changing the ARB_SEL bits, we can rotate the priority of REQ1 to REQ4.

Here, if ARB_MODE bit is set to 0, ARB_SEL bits are not automatically changed, making the arbiter to operate in the fixed priority mode (note that even in this mode, we can reconfigure the priority by manually changing the ARB_SEL bits). On the other hand, if ARB_MODE bit is 1, ARB_SEL bits are changed in rotation fashion, e.g., if REQ1 is serviced, ARB_SEL bits are changed to 01b automatically so as to put REQ1 into the lowest priority. The detailed rules of ARB_SEL change are as follows:

If REQ0 or REQ5 is serviced, ARB_SEL bits are not changed at all.

If REQ1 is serviced, ARB_SEL bits are changed to 01b.

If REQ2 is serviced, ARB_SEL bits are changed to 10b.

If REQ3 is serviced, ARB_SEL bits are changed to 11b.

If REQ4 is serviced, ARB_SEL bits are changed to 00b.

## INTERRUPT CONTROLLER SPECIAL REGISTERS

There are five control registers in the interrupt controller: source pending register, interrupt mode register, mask register, priority register, and interrupt pending register.

All the interrupt requests from the interrupt sources are first registered in the source pending register. They are divided into two groups including Fast Interrupt Request (FIQ) and Interrupt Request (IRQ), based on the interrupt mode register. The arbitration procedure for multiple IRQs is based on the priority register.

### SOURCE PENDING (SRCPND) REGISTER

The SRCPND register is composed of 32 bits each of which is related to an interrupt source. Each bit is set to 1 if the corresponding interrupt source generates the interrupt request and waits for the interrupt to be serviced. Accordingly, this register indicates which interrupt source is waiting for the request to be serviced. Note that each bit of the SRCPND register is automatically set by the interrupt sources regardless of the masking bits in the INTMASK register. In addition, the SRCPND register is not affected by the priority logic of interrupt controller.

In the interrupt service routine for a specific interrupt source, the corresponding bit of the SRCPND register has to be cleared to get the interrupt request from the same source correctly. If you return from the ISR without clearing the bit, the interrupt controller operates as if another interrupt request came in from the same source. In other words, if a specific bit of the SRCPND register is set to 1, it is always considered as a valid interrupt request waiting to be serviced.

The time to clear the corresponding bit depends on the user's requirement. If you want to receive another valid request from the same source, you should clear the corresponding bit first, and then enable the interrupt.

You can clear a specific bit of the SRCPND register by writing a data to this register. It clears only the bit positions of the SRCPND corresponding to those set to one in the data. The bit positions corresponding to those that are set to 0 in the data remains as they are.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SRCPND | 0X4A000000 | R/W | Indicate the interrupt request status.<br>0 = The interrupt has not been requested.<br>1 = The interrupt source has asserted the interrupt request. | 0x00000000 |

| SRCPND | Bit | Description | Initial State |
|---|---|---|---|
| INT_ADC | [31] | 0 = Not requested, 1 = Requested | 0 |
| INT_RTC | [30] | 0 = Not requested, 1 = Requested | 0 |
| INT_SPI1 | [29] | 0 = Not requested, 1 = Requested | 0 |
| INT_UART0 | [28] | 0 = Not requested, 1 = Requested | 0 |
| INT_IIC | [27] | 0 = Not requested, 1 = Requested | 0 |
| INT_USBH | [26] | 0 = Not requested, 1 = Requested | 0 |
| INT_USBD | [25] | 0 = Not requested, 1 = Requested | 0 |
| INT_NFCON | [24] | 0 = Not requested, 1 = Requested | 0 |
| INT_UART1 | [23] | 0 = Not requested, 1 = Requested | 0 |
| INT_SPI0 | [22] | 0 = Not requested, 1 = Requested | 0 |
| INT_SDI | [21] | 0 = Not requested, 1 = Requested | 0 |
| INT_DMA3 | [20] | 0 = Not requested, 1 = Requested | 0 |
| INT_DMA2 | [19] | 0 = Not requested, 1 = Requested | 0 |
| INT_DMA1 | [18] | 0 = Not requested, 1 = Requested | 0 |
| INT_DMA0 | [17] | 0 = Not requested, 1 = Requested | 0 |
| INT_LCD | [16] | 0 = Not requested, 1 = Requested | 0 |
| INT_UART2 | [15] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER4 | [14] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER3 | [13] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER2 | [12] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER1 | [11] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER0 | [10] | 0 = Not requested, 1 = Requested | 0 |
| INT_WDT | [9] | 0 = Not requested, 1 = Requested | 0 |
| INT_TICK | [8] | 0 = Not requested, 1 = Requested | 0 |
| nBATT_FLT | [7] | 0 = Not requested, 1 = Requested | 0 |
| INT_CAM | [6] | 0 = Not requested, 1 = Requested | 0 |
| EINT8_23 | [5] | 0 = Not requested, 1 = Requested | 0 |
| EINT4_7 | [4] | 0 = Not requested, 1 = Requested | 0 |
| EINT3 | [3] | 0 = Not requested, 1 = Requested | 0 |
| EINT2 | [2] | 0 = Not requested, 1 = Requested | 0 |
| EINT1 | [1] | 0 = Not requested, 1 = Requested | 0 |
| EINT0 | [0] | 0 = Not requested, 1 = Requested | 0 |

.

### INTERRUPT MODE (INTMOD) REGISTER

This register is composed of 32 bits each of which is related to an interrupt source. If a specific bit is set to 1, the corresponding interrupt is processed in the FIQ (fast interrupt) mode. Otherwise, it is processed in the IRQ mode (normal interrupt).

Note that only one interrupt source can be serviced in the FIQ mode in the interrupt controller (you should use the FIQ mode only for the urgent interrupt). Thus, only one bit of INTMOD can be set to 1.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTMOD | 0X4A000004 | R/W | Interrupt mode regiseter.<br>0 = IRQ mode 1 = FIQ mode | 0x00000000 |

**Note**: If an interrupt mode is set to FIQ mode in the INTMOD register, FIQ interrupt will not affect both INTPND and INTOFFSET registers. In this case, the two registers are valid only for IRQ mode interrupt source.

| INTMOD | Bit | Description | Initial State |
|---|---|---|---|
| INT_ADC | [31] | 0 = IRQ, 1 = FIQ | 0 |
| INT_RTC | [30] | 0 = IRQ, 1 = FIQ | 0 |
| INT_SPI1 | [29] | 0 = IRQ, 1 = FIQ | 0 |
| INT_UART0 | [28] | 0 = IRQ, 1 = FIQ | 0 |
| INT_IIC | [27] | 0 = IRQ, 1 = FIQ | 0 |
| INT_USBH | [26] | 0 = IRQ, 1 = FIQ | 0 |
| INT_USBD | [25] | 0 = IRQ, 1 = FIQ | 0 |
| INT_NFCON | [24] | 0 = IRQ, 1 = FIQ | 0 |
| INT_URRT1 | [23] | 0 = IRQ, 1 = FIQ | 0 |
| INT_SPI0 | [22] | 0 = IRQ, 1 = FIQ | 0 |
| INT_SDI | [21] | 0 = IRQ, 1 = FIQ | 0 |
| INT_DMA3 | [20] | 0 = IRQ, 1 = FIQ | 0 |
| INT_DMA2 | [19] | 0 = IRQ, 1 = FIQ | 0 |
| INT_DMA1 | [18] | 0 = IRQ, 1 = FIQ | 0 |
| INT_DMA0 | [17] | 0 = IRQ, 1 = FIQ | 0 |
| INT_LCD | [16] | 0 = IRQ, 1 = FIQ | 0 |
| INT_UART2 | [15] | 0 = IRQ, 1 = FIQ | 0 |
| INT_TIMER4 | [14] | 0 = IRQ, 1 = FIQ | 0 |
| INT_TIMER3 | [13] | 0 = IRQ, 1 = FIQ | 0 |
| INT_TIMER2 | [12] | 0 = IRQ, 1 = FIQ | 0 |
| INT_TIMER1 | [11] | 0 = IRQ, 1 = FIQ | 0 |
| INT_TIMER0 | [10] | 0 = IRQ, 1 = FIQ | 0 |
| INT_WDT | [9] | 0 = IRQ, 1 = FIQ | 0 |
| INT_TICK | [8] | 0 = IRQ, 1 = FIQ | 0 |
| nBATT_FLT | [7] | 0 = IRQ, 1 = FIQ | 0 |
| INT_CAM | [6] | 0 = IRQ, 1 = FIQ | 0 |
| EINT8_23 | [5] | 0 = IRQ, 1 = FIQ | 0 |
| EINT4_7 | [4] | 0 = IRQ, 1 = FIQ | 0 |
| EINT3 | [3] | 0 = IRQ, 1 = FIQ | 0 |
| EINT2 | [2] | 0 = IRQ, 1 = FIQ | 0 |
| EINT1 | [1] | 0 = IRQ, 1 = FIQ | 0 |
| EINT0 | [0] | 0 = IRQ, 1 = FIQ | 0 |

### INTERRUPT MASK (INTMSK) REGISTER

This register also has 32 bits each of which is related to an interrupt source. If a specific bit is set to 1, the CPU does not service the interrupt request from the corresponding interrupt source (note that even in such a case, the corresponding bit of SRCPND register is set to 1). If the mask bit is 0, the interrupt request can be serviced.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTMSK | 0X4A000008 | R/W | Determine which interrupt source is masked. The masked interrupt source will not be serviced.<br>0 = Interrupt service is available.<br>1 = Interrupt service is masked. | 0xFFFFFFFF |

| INTMSK | Bit | Description | Initial State |
|---|---|---|---|
| INT_ADC | [31] | 0 = Service available, 1 = Masked | 1 |
| INT_RTC | [30] | 0 = Service available, 1 = Masked | 1 |
| INT_SPI1 | [29] | 0 = Service available, 1 = Masked | 1 |
| INT_UART0 | [28] | 0 = Service available, 1 = Masked | 1 |
| INT_IIC | [27] | 0 = Service available, 1 = Masked | 1 |
| INT_USBH | [26] | 0 = Service available, 1 = Masked | 1 |
| INT_USBD | [25] | 0 = Service available, 1 = Masked | 1 |
| INT_NFCON | [24] | 0 = Service available, 1 = Masked | 1 |
| INT_UART1 | [23] | 0 = Service available, 1 = Masked | 1 |
| INT_SPI0 | [22] | 0 = Service available, 1 = Masked | 1 |
| INT_SDI | [21] | 0 = Service available, 1 = Masked | 1 |
| INT_DMA3 | [20] | 0 = Service available, 1 = Masked | 1 |
| INT_DMA2 | [19] | 0 = Service available, 1 = Masked | 1 |
| INT_DMA1 | [18] | 0 = Service available, 1 = Masked | 1 |
| INT_DMA0 | [17] | 0 = Service available, 1 = Masked | 1 |
| INT_LCD | [16] | 0 = Service available, 1 = Masked | 1 |
| INT_UART2 | [15] | 0 = Service available, 1 = Masked | 1 |
| INT_TIMER4 | [14] | 0 = Service available, 1 = Masked | 1 |
| INT_TIMER3 | [13] | 0 = Service available, 1 = Masked | 1 |
| INT_TIMER2 | [12] | 0 = Service available, 1 = Masked | 1 |
| INT_TIMER1 | [11] | 0 = Service available, 1 = Masked | 1 |
| INT_TIMER0 | [10] | 0 = Service available, 1 = Masked | 1 |
| INT_WDT | [9] | 0 = Service available, 1 = Masked | 1 |
| INT_TICK | [8] | 0 = Service available, 1 = Masked | 1 |
| nBATT_FLT | [7] | 0 = Service available, 1 = Masked | 1 |
| INT_CAM | [6] | 0 = Service available, 1 = Masked | 1 |
| EINT8_23 | [5] | 0 = Service available, 1 = Masked | 1 |
| EINT4_7 | [4] | 0 = Service available, 1 = Masked | 1 |
| EINT3 | [3] | 0 = Service available, 1 = Masked | 1 |
| EINT2 | [2] | 0 = Service available, 1 = Masked | 1 |
| EINT1 | [1] | 0 = Service available, 1 = Masked | 1 |
| EINT0 | [0] | 0 = Service available, 1 = Masked | 1 |

**PRIORITY REGISTER (PRIORITY)**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| PRIORITY | 0x4A00000C | R/W | IRQ priority control register | 0x7F |

| PRIORITY | Bit | Description | Initial State |
|---|---|---|---|
| ARB_SEL6 | [20:19] | Arbiter 6 group priority order set<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 0 |
| ARB_SEL5 | [18:17] | Arbiter 5 group priority order set<br>00 = REQ 1-2-3-4 01 = REQ 2-3-4-1<br>10 = REQ 3-4-1-2 11 = REQ 4-1-2-3 | 0 |
| ARB_SEL4 | [16:15] | Arbiter 4 group priority order set<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 0 |
| ARB_SEL3 | [14:13] | Arbiter 3 group priority order set<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 0 |
| ARB_SEL2 | [12:11] | Arbiter 2 group priority order set<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 0 |
| ARB_SEL1 | [10:9] | Arbiter 1 group priority order set<br>00 = REQ 0-1-2-3-4-5 01 = REQ 0-2-3-4-1-5<br>10 = REQ 0-3-4-1-2-5 11 = REQ 0-4-1-2-3-5 | 0 |
| ARB_SEL0 | [8:7] | Arbiter 0 group priority order set<br>00 = REQ 1-2-3-4 01 = REQ 2-3-4-1<br>10 = REQ 3-4-1-2 11 = REQ 4-1-2-3 | 0 |
| ARB_MODE6 | [6] | Arbiter 6 group priority rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE5 | [5] | Arbiter 5 group priority rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE4 | [4] | Arbiter 4 group priority rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE3 | [3] | Arbiter 3 group priority rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE2 | [2] | Arbiter 2 group priority rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE1 | [1] | Arbiter 1 group priority rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |
| ARB_MODE0 | [0] | Arbiter 0 group priority rotate enable<br>0 = Priority does not rotate, 1 = Priority rotate enable | 1 |

### INTERRUPT PENDING (INTPND) REGISTER

Each of the 32 bits in the interrupt pending register shows whether the corresponding interrupt request, which is unmasked and waits for the interrupt to be serviced, has the highest priority . Since the INTPND register is located after the priority logic, only one bit can be set to 1, and that interrupt request generates IRQ to CPU. In interrupt service routine for IRQ, you can read this register to determine which interrupt source is serviced among the 32 sources.

Like the SRCPND register, this register has to be cleared in the interrupt service routine after clearing the SRCPND register. We can clear a specific bit of the INTPND register by writing a data to this register. It clears only the bit positions of the INTPND register corresponding to those set to one in the data. The bit positions corresponding to those that are set to 0 in the data remains as they are.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTPND | 0X4A000010 | R/W | Indicate the interrupt request status.<br>0 = The interrupt has not been requested.<br>1 = The interrupt source has asserted the interrupt request. | 0x00000000 |

**Note**: If the FIQ mode interrupt occurs, the corresponding bit of INTPND will not be turned on as the INTPND register is available only for IRQ mode interrupt.

| INTPND | Bit | Description | Initial State |
|---|---|---|---|
| INT_ADC | [31] | 0 = Not requested, 1 = Requested | 0 |
| INT_RTC | [30] | 0 = Not requested, 1 = Requested | 0 |
| INT_SPI1 | [29] | 0 = Not requested, 1 = Requested | 0 |
| INT_UART0 | [28] | 0 = Not requested, 1 = Requested | 0 |
| INT_IIC | [27] | 0 = Not requested, 1 = Requested | 0 |
| INT_USBH | [26] | 0 = Not requested, 1 = Requested | 0 |
| INT_USBD | [25] | 0 = Not requested, 1 = Requested | 0 |
| INT_NFCON | [24] | 0 = Not requested, 1 = Requested | 0 |
| INT_UART1 | [23] | 0 = Not requested, 1 = Requested | 0 |
| INT_SPI0 | [22] | 0 = Not requested, 1 = Requested | 0 |
| INT_SDI | [21] | 0 = Not requested, 1 = Requested | 0 |
| INT_DMA3 | [20] | 0 = Not requested, 1 = Requested | 0 |
| INT_DMA2 | [19] | 0 = Not requested, 1 = Requested | 0 |
| INT_DMA1 | [18] | 0 = Not requested, 1 = Requested | 0 |
| INT_DMA0 | [17] | 0 = Not requested, 1 = Requested | 0 |
| INT_LCD | [16] | 0 = Not requested, 1 = Requested | 0 |
| INT_UART2 | [15] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER4 | [14] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER3 | [13] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER2 | [12] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER1 | [11] | 0 = Not requested, 1 = Requested | 0 |
| INT_TIMER0 | [10] | 0 = Not requested, 1 = Requested | 0 |
| INT_WDT | [9] | 0 = Not requested, 1 = Requested | 0 |
| INT_TICK | [8] | 0 = Not requested, 1 = Requested | 0 |
| nBATT_FLT | [7] | 0 = Not requested, 1 = Requested | 0 |
| INT_CAM | [6] | 0 = Not requested, 1 = Requested | 0 |
| EINT8_23 | [5] | 0 = Not requested, 1 = Requested | 0 |
| EINT4_7 | [4] | 0 = Not requested, 1 = Requested | 0 |
| EINT3 | [3] | 0 = Not requested, 1 = Requested | 0 |
| EINT2 | [2] | 0 = Not requested, 1 = Requested | 0 |
| EINT1 | [1] | 0 = Not requested, 1 = Requested | 0 |
| EINT0 | [0] | 0 = Not requested, 1 = Requested | 0 |

### INTERRUPT OFFSET (INTOFFSET) REGISTER

The value in the interrupt offset register shows which interrupt request of IRQ mode is in the INTPND register. This bit can be cleared automatically by clearing SRCPND and INTPND.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTOFFSET | 0X4A000014 | R | Indicate the IRQ interrupt request source | 0x00000000 |

| INT Source | The OFFSET value | INT Source | The OFFSET value |
|---|---|---|---|
| INT_ADC | 31 | INT_UART2 | 15 |
| INT_RTC | 30 | INT_TIMER4 | 14 |
| INT_SPI1 | 29 | INT_TIMER3 | 13 |
| INT_UART0 | 28 | INT_TIMER2 | 12 |
| INT_IIC | 27 | INT_TIMER1 | 11 |
| INT_USBH | 26 | INT_TIMER0 | 10 |
| INT_USBD | 25 | INT_WDT | 9 |
| INT_NFCON | 24 | INT_TICK | 8 |
| INT_UART1 | 23 | nBATT_FLT | 7 |
| INT_SPI0 | 22 | INT_CAM | 6 |
| INT_SDI | 21 | EINT8_23 | 5 |
| INT_DMA3 | 20 | EINT4_7 | 4 |
| INT_DMA2 | 19 | EINT3 | 3 |
| INT_DMA1 | 18 | EINT2 | 2 |
| INT_DMA0 | 17 | EINT1 | 1 |
| INT_LCD | 16 | EINT0 | 0 |

**Note**: FIQ mode interrupt does not affect the INTOFFSET register as the register is available only for IRQ mode interrupt.

### SUB SOURCE PENDING (SUBSRCPND) REGISTER

You can clear a specific bit of the SUBSRCPND register by writing a data to this register. It clears only the bit positions of the SUBSRCPND register corresponding to those set to one in the data. The bit positions corresponding to those that are set to 0 in the data remains as they are.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SUBSRCPND | 0X4A000018 | R/W | Indicate the interrupt request status.<br>0 = The interrupt has not been requested.<br>1 = The interrupt source has asserted the interrupt request. | 0x00000000 |

| SUBSRCPND | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [31:13] | Not used | 0 |
| INT_CAM_C | [12] | 0 = Not requested, 1 = Requested | 0 |
| INT_CAM_S | [11] | 0 = Not requested, 1 = Requested | 0 |
| INT_ADC | [10] | 0 = Not requested, 1 = Requested | 0 |
| INT_TC | [9] | 0 = Not requested, 1 = Requested | 0 |
| INT_ERR2 | [8] | 0 = Not requested, 1 = Requested | 0 |
| INT_TXD2 | [7] | 0 = Not requested, 1 = Requested | 0 |
| INT_RXD2 | [6] | 0 = Not requested, 1 = Requested | 0 |
| INT_ERR1 | [5] | 0 = Not requested, 1 = Requested | 0 |
| INT_TXD1 | [4] | 0 = Not requested, 1 = Requested | 0 |
| INT_RXD1 | [3] | 0 = Not requested, 1 = Requested | 0 |
| INT_ERR0 | [2] | 0 = Not requested, 1 = Requested | 0 |
| INT_TXD0 | [1] | 0 = Not requested, 1 = Requested | 0 |
| INT_RXD0 | [0] | 0 = Not requested, 1 = Requested | 0 |

Map To SRCPND

| SRCPND | SUBSRCPND | Remark |
|---|---|---|
| INT_UART0 | INT_RXD0,INT_TXD0,INT_ERR0 | |
| INT_UART1 | INT_RXD1,INT_TXD1,INT_ERR1 | |
| INT_UART2 | INT_RXD2,INT_TXD2,INT_ERR2 | |
| INT_ADC | INT_ADC, INT_TC | |
| INT_CAM | INT_CAM_S, INT_CAM_C | |

### INTERRUPT SUB MASK (INTSUBMSK) REGISTER

This register has 11 bits each of which is related to an interrupt source. If a specific bit is set to 1, the interrupt request from the corresponding interrupt source is not serviced by the CPU (note that even in such a case, the corresponding bit of the SUBSRCPND register is set to 1). If the mask bit is 0, the interrupt request can be serviced.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| INTSUBMSK | 0X4A00001C | R/W | Determine which interrupt source is masked. The masked interrupt source will not be serviced.<br>0 = Interrupt service is available.<br>1 = Interrupt service is masked. | 0xFFFF |

| INTSUBMSK | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [31:16] | Not used | 0 |
| Reserved | [15:13] | Not used | 0x7 |
| INT_CAM_C | [12] | 0 = Service available, 1 = Masked | 1 |
| INT_CAM_S | [11] | 0 = Service available, 1 = Masked | 1 |
| INT_ADC | [10] | 0 = Service available, 1 = Masked | 1 |
| INT_TC | [9] | 0 = Service available, 1 = Masked | 1 |
| INT_ERR2 | [8] | 0 = Service available, 1 = Masked | 1 |
| INT_TXD2 | [7] | 0 = Service available, 1 = Masked | 1 |
| INT_RXD2 | [6] | 0 = Service available, 1 = Masked | 1 |
| INT_ERR1 | [5] | 0 = Service available, 1 = Masked | 1 |
| INT_TXD1 | [4] | 0 = Service available, 1 = Masked | 1 |
| INT_RXD1 | [3] | 0 = Service available, 1 = Masked | 1 |
| INT_ERR0 | [2] | 0 = Service available, 1 = Masked | 1 |
| INT_TXD0 | [1] | 0 = Service available, 1 = Masked | 1 |
| INT_RXD0 | [0] | 0 = Service available, 1 = Masked | 1 |

# NOTES

# 15 LCD CONTROLLER

## OVERVIEW

The LCD controller in the S3C2440X consists of the logic for transferring LCD image data from a video buffer located in system memory to an external LCD driver.

The LCD controller supports monochrome, 2-bit per pixel (4-level gray scale) or 4-bit per pixel (16-level gray scale) mode on a monochrome LCD, using a time-based dithering algorithm and Frame Rate Control (FRC) method and it can be interfaced with a color LCD panel at 8-bit per pixel (256-level color) and 12-bit per pixel (4096-level color) for interfacing with STN LCD.
It can support 1-bit per pixel, 2-bit per pixel, 4-bit per pixel, and 8-bit per pixel for interfacing with the palettized TFT color LCD panel, and 16-bit per pixel and 24-bit per pixel for non-palettized true-color display.

The LCD controller can be programmed to support different requirements on the screen related to the number of horizontal and vertical pixels, data line width for the data interface, interface timing, and refresh rate.

### FEATURES

**STN LCD displays:**

— Supports 3 types of LCD panels: 4-bit dual scan, 4-bit single scan, and 8-bit single scan display type

— Supports the monochrome, 4 gray levels, and 16 gray levels

— Supports 256 colors and 4096 colors for color STN LCD panel

— Supports multiple screen size
Typical actual screen size: 640 x 480, 320 x 240, 160 x 160, and others
Maximum virtual screen size is 4Mbytes.
Maximum virtual screen size in 256 color mode: 4096 x 1024, 2048 x 2048, 1024 x 4096, and others

**TFT LCD displays:**

— Supports 1, 2, 4 or 8-bpp (bit per pixel) palettized color displays for TFT

— Supports 16-bpp non-palettized true-color displays for color TFT

— Supports 24-bpp non-palettized true-color displays for color TFT

— Supports maximum 16M color TFT at 24bit per pixel mode

— Supports multiple screen size
Typical actual screen size: 640 x 480, 320 x 240, 160 x 160, and others
Maximum virtual screen size is 4Mbytes.
Maximum virtual screen size in 64K color mode: 2048 x 1024 and others

## COMMON FEATURES

The LCD controller has a dedicated DMA that supports to fetch the image data from video buffer located in system memory. Its features also include:

— Dedicated interrupt functions (INT_FrSyn and INT_FiCnt)

— The system memory is used as the display memory.

— Supports Multiple Virtual Display Screen (Supports Hardware Horizontal/Vertical Scrolling)

— Programmable timing control for different display panels

— Supports little and big-endian byte ordering, as well as WinCE data formats

— Supports 2-type SEC TFT LCD panel
(SAMSUNG 3.5" Portrait / 256K Color /Reflective and Transflective a-Si TFT LCD)
LTS350Q1-PD1: TFT LCD panel with touch panel and front light unit (Reflective type)
LTS350Q1-PD2: TFT LCD panel only
LTS350Q1-PE1: TFT LCD panel with touch panel and front light unit (Transflective type)
LTS350Q1-PE2: TFT LCD panel only

**NOTE**

WinCE doesn't support the 12-bit packed data format.

Please check if WinCE can support the 12-bit color-mode.

## EXTERNAL INTERFACE SIGNAL

| STN | TFT | SEC TFT (LTS350Q1-PD1/2) | SEC TFT (LTS350Q1-PE1/2) |
|---|---|---|---|
| **VFRAME** (Frame sync. Signal) | **VSYNC** (Vertical sync. Signal) | **STV** | **STV** |
| **VLINE** (Line sync pulse signal) | **HSYNC** (Horizontal sync. Signal) | **CPV** | **CPV** |
| **VCLK** (Pixel clock signal) | **VCLK** (Pixel clock signal) | **LCD_HCLK** | **LCD_HCLK** |
| **VD[23:0]** (LCD pixel data output ports) | **VD[23:0]** (LCD pixel data output ports) | **VD[23:0]** | **VD[23:0]** |
| **VM** (AC bias signal for LCD driver) | **VDEN** (Data enable signal) | **TP** | **TP** |
| - | **LEND** (Line end signal) | **STH** | **STH** |
| **LCD_PWREN** | **LCD_PWREN** | **LCD_PWREN** | **LCD_PWREN** |
| - | - | **LPC_OE** | **LCC_INV** |
| - | - | **LPC_REV** | **LCC_REV** |
| - | - | **LPC_REVB** | **LCC_REVB** |

## BLOCK DIAGRAM

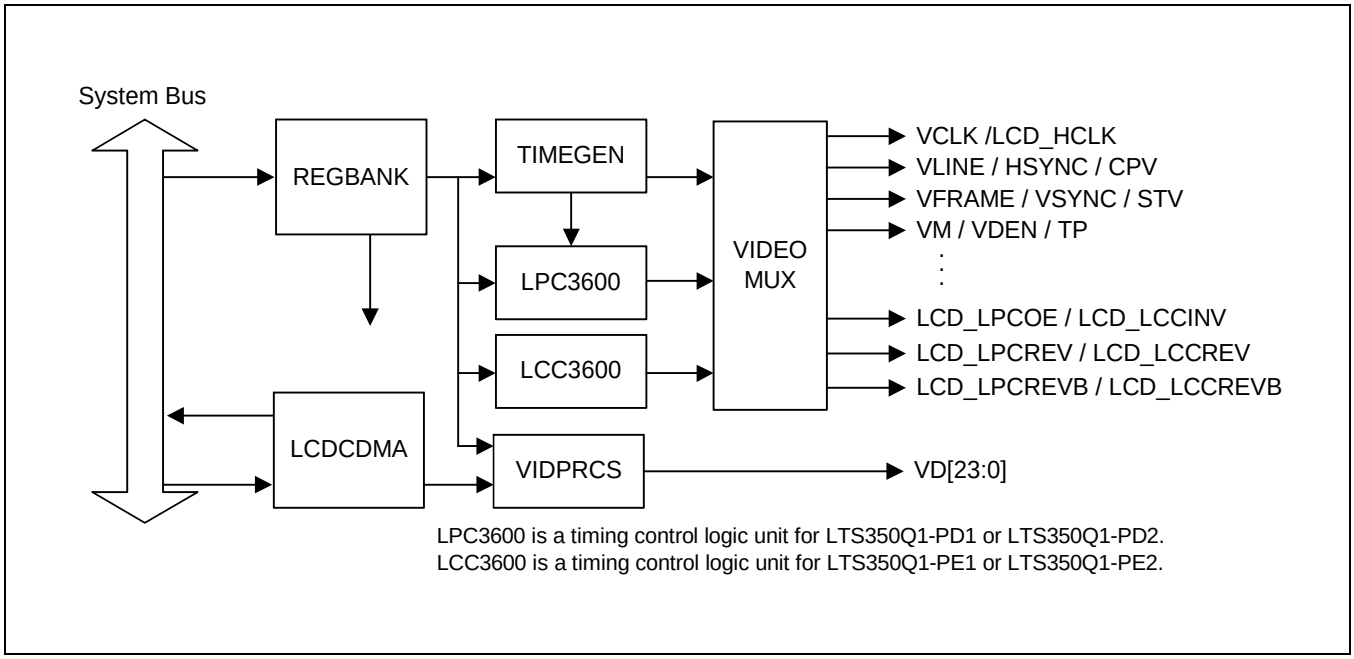



**Figure 15-1. LCD Controller Block Diagram**

The S3C2440X LCD controller is used to transfer the video data and to generate the necessary control signals, such as VFRAME, VLINE, VCLK, VM, and so on. In addition to the control signals, the S3C2440X has the data ports for video data, which are VD[23:0] as shown in Figure 15-1. The LCD controller consists of a REGBANK, LCDCDMA, VIDPRCS, TIMEGEN, and LPC3600 (See the Figure 15-1 LCD Controller Block Diagram). The REGBANK has 17 programmable register sets and 256x16 palette memory which are used to configure the LCD controller. The LCDCDMA is a dedicated DMA, which can transfer the video data in frame memory to LCD driver automatically. By using this special DMA, the video data can be displayed on the screen without CPU intervention. The VIDPRCS receives the video data from the LCDCDMA and sends the video data through the VD[23:0] data ports to the LCD driver after changing them into a suitable data format, for example 4/8-bit single scan or 4-bit dual scan display mode. The TIMEGEN consists of programmable logic to support the variable requirements of interface timing and rates commonly found in different LCD drivers. The TIMEGEN block generates VFRAME, VLINE, VCLK, VM, and so on.

The description of data flow is as follows:

FIFO memory is present in the LCDCDMA. When FIFO is empty or partially empty, the LCDCDMA requests data fetching from the frame memory based on the burst memory transfer mode (consecutive memory fetching of 4 words (16 bytes) per one burst request without allowing the bus mastership to another bus master during the bus transfer). When the transfer request is accepted by bus arbitrator in the memory controller, there will be four successive word data transfers from system memory to internal FIFO. The total size of FIFO is 28 words, which consists of 12 words FIFOL and 16 words FIFOH, respectively. The S3C2440X has two FIFOs to support the dual scan display mode. In case of single scan mode, one of the FIFOs (FIFOH) can only be used.

## STN LCD CONTROLLER OPERATION

### TIMING GENERATOR (TIMEGEN)

The TIMEGEN generates the control signals for the LCD driver, such as VFRAME, VLINE, VCLK, and VM. These control signals are closely related to the configuration on the LCDCON1/2/3/4/5 registers in the REGBANK. Based on these programmable configurations on the LCD control registers in the REGBANK, the TIMEGEN can generate the programmable control signals suitable to support many different types of LCD drivers.

The VFRAME pulse is asserted for the duration of the entire first line at a frequency of once per frame. The VFRAME signal is asserted to bring the LCD's line pointer to the top of the display to start over.

The VM signal helps the LCD driver alternate the polarity of the row and column voltages, which are used to turn the pixel on and off. The toggling rate of VM signals depends on the MMODE bit of the LCDCON1 register and MVAL field of the LCDCON4 register. If the MMODE bit is 0, the VM signal is configured to toggle on every frame. If the MMODE bit is 1, the VM signal is configured to toggle on the every event of the elapse of the specified number of VLINE by the MVAL[7:0] value. Figure 15-4 shows an example for MMODE=0 and for MMODE=1 with the value of MVAL[7:0]=0x2. When MMODE=1, the VM rate is related to MVAL[7:0], as shown below:

VM Rate = VLINE Rate / ( 2 x MVAL)

The VFRAME and VLINE pulse generation relies on the configurations of the HOZVAL field and the LINEVAL field in the LCDCON2/3 register. Each field is related to the LCD size and display mode. In other words, the HOZVAL and LINEVAL can be determined by the size of the LCD panel and the display mode according to the following equation:

HOZVAL = (Horizontal display size / Number of the valid VD data line)-1

In color mode: Horizontal display size = 3 x Number of Horizontal Pixel

In the 4-bit single scan display mode, the Number of valid VD data line should be 4. In case of 4-bit dual scan display, the Number of valid VD data lineshould also be 4 while in case of 8-bit single scan display mode, the Number of valid VD data line should be 8.

LINEVAL = (Vertical display size) -1: In case of single scan display type

LINEVAL = (Vertical display size / 2) -1: In case of dual scan display type

The rate of VCLK signal depends on the configuration of the CLKVAL field in the LCDCON1 register. Table 15-1 defines the relationship of VCLK and CLKVAL. The minimum value of CLKVAL is 2.

VCLK(Hz)=HCLK/(CLKVAL x 2)

The frame rate is the VFRAM signal frequency. The frame rate is closely related to the field of WLH[1:0](VLINE pulse width) WDLY[1:0] (the delay width of VCLK after VLINE pulse), HOZVAL, LINEBLANK, and LINEVAL in the LCDCON1/2/3/4 registers as well as VCLK and HCLK. Most LCD drivers need their own adequate frame rate. The frame rate is calculated as follows:

frame_rate(Hz) = 1 / [ { (1/VCLK) x (HOZVAL+1)+(1/HCLK) x (**A**+**B**+(LINEBLANK x 8) ) } x ( LINEVAL+1) ]

$$\boldsymbol{A} = 2^{(4+WLH)},\ \boldsymbol{B} = 2^{(4+WDLY)}$$

**Table 15-1. Relation between VCLK and CLKVAL (STN, HCLK=60MHz)**

| CLKVAL | 60MHz/X | VCLK |
|---|---|---|
| 2 | 60 MHz/4 | 15.0 MHz |
| 3 | 60 MHz/6 | 10.0 MHz |
| : | : | : |
| 1023 | 60 MHz/2046 | 29.3 kHz |

## VIDEO OPERATION

The S3C2440X LCD controller supports 8-bit color mode (256 color mode), 12-bit color mode (4096 color mode), 4 level gray scale mode, 16 level gray scale mode as well as the monochrome mode. For the gray or color mode, it is required to implement the shades of gray level or color according to time-based dithering algorithm and Frame Rate Control (FRC) method. The selection can be made following a programmable lockup table, which will be explained later. The monochrome mode bypasses these modules (FRC and lookup table) and basically serializes the data in FIFOH (and FIFOL if a dual scan display type is used) into 4-bit (or 8-bit if a 4-bit dual scan or 8-bit single scan display type is used) streams by shifting the video data to the LCD driver.

The following sections describe the operation on the gray and color mode in terms of the lookup table and FRC.

### Lookup Table

The S3C2440X can support the lookup table for various selection of color or gray level mapping, ensuring flexible operation for users. The lookup table is the palette which allows the selection on the level of color or gray (Selection on 4-gray levels among 16 gray levels in case of 4 gray mode, selection on 8 red levels among 16 levels, 8 green levels among 16 levels and 4 blue levels among 16 levels in case of 256 color mode). In other words, users can select 4 gray levels among 16 gray levels by using the lookup table in the 4 gray level mode. The gray levels cannot be selected in the 16 gray level mode; all 16 gray levels must be chosen among the possible 16 gray levels. In case of 256 color mode, 3 bits are allocated for red, 3 bits for green and 2 bits for blue. The 256 colors mean that the colors are formed from the combination of 8 red, 8 green and 4 blue levels (8x8x4 = 256). In the color mode, the lookup table can be used for suitable selections. Eight red levels can be selected among 16 possible red levels, 8 green levels among 16 green levels, and 4 blue levels among 16 blue levels. In case of 4096 color mode, there is no selection as in the 256 color mode.

### Gray Mode Operation

The S3C2440X LCD controller supports two gray modes: 2-bit per pixel gray (4 level gray scale) and 4-bit per pixel gray (16 level gray scale). The 2-bit per pixel gray mode uses a lookup table (BLUELUT), which allows selection on 4 gray levels among 16 possible gray levels. The 2-bit per pixel gray lookup table uses the BULEVAL[15:0] in Blue Lookup Table (BLUELUT) register as same as blue lookup table in color mode. The gray level 0 will be denoted by BLUEVAL[3:0] value. If BLUEVAL[3:0] is 9, level 0 will be represented by gray level 9 among 16 gray levels. If BLUEVAL[3:0] is 15, level 0 will be represented by gray level 15 among 16 gray levels, and so on. Following the same method as above, level 1 will also be denoted by BLUEVAL[7:4], the level 2 by BLUEVAL[11:8], and the level 3 by BLUEVAL[15:12]. These four groups among BLUEVAL[15:0] will represent level 0, level 1, level 2, and level 3. In 16 gray levels, there is no selection as in the 16 gray levels.

### 256 Level Color Mode Operation

The S3C2440X LCD controller can support an 8-bit per pixel 256 color display mode. The color display mode can generate 256 levels of color using the dithering algorithm and FRC. The 8-bit per pixel are encoded into 3-bits for red, 3-bits for green, and 2-bits for blue. The color display mode uses separate lookup tables for red, green, and blue. Each lookup table uses the REDVAL[31:0] of REDLUT register, GREENVAL[31:0] of GREENLUT register, and BLUEVAL[15:0] of BLUELUT register as the programmable lookup table entries.

Similar to the gray level display, 8 group or field of 4 bits in the REDLUR register, i.e., REDVAL[31:28], REDLUT[27:24], REDLUT[23:20], REDLUT[19:16], REDLUT[15:12], REDLUT[11:8], REDLUT[7:4], and REDLUT[3:0], are assigned to each red level. The possible combination of 4 bits (each field) is 16, and each red level should be assigned to one level among possible 16 cases. In other words, the user can select the suitable red level by using this type of lookup table. For green color, the GREENVAL[31:0] of the GREENLUT register is assigned as the lookup table, as was done in the case of red color. Similarly, the BLUEVAL[15:0] of the BLUELUT register is also assigned as a lookup table. For blue color, 2 bits are allocated for 4 blue levels, different from the 8 red or green levels.

### 4096 Level Color Mode Operation

The S3C2440X LCD controller can support a 12-bit per pixel 4096 color display mode. The color display mode can generate 4096 levels of color using the dithering algorithm and FRC. The 12-bit per pixel are encoded into 4-bits for red, 4-bits for green, and 4-bits for blue. The 4096 color display mode does not use lookup tables.

## DITHERING AND FRAME RATE CONTROL

For STN LCD displays (except monochrome), video data must be processed by a dithering algorithm. The DITHFRC block has two functions, such as Time-based Dithering Algorithm for reducing flicker and Frame Rate Control (FRC) for displaying gray and color level on the STN panel. The main principle of gray and color level display on the STN panel based on FRC is described. For example, to display the third gray (3/16) level from a total of 16 levels, the 3 times pixel should be on and 13 times pixel off. In other words, 3 frames should be selected among the 16 frames, of which 3 frames should have a pixel-on on a specific pixel while the remaining 13 frames should have a pixel-off on a specific pixel. These 16 frames should be displayed periodically. This is basic principle on how to display the gray level on the screen, so-called gray level display by FRC. The actual example is shown in Table 15-2. To represent the $14^{th}$ gray level in the table, we should have a 6/7 duty cycle, which mean that there are 6 times pixel-on and one time pixel-off. The other cases for all gray levels are also shown in Table 15-2.

In the STN LCD display, we should be reminded of one item, i.e., Flicker Noise due to the simultaneous pixel-on and -off on adjacent frames. For example, if all pixels on first frame are turned on and all pixels on next frame are turned off, the Flicker Noise will be maximized. To reduce the Flicker Noise on the screen, the average probability of pixel-on and -off between frames should be the same. In order to realize this, the Time-based Dithering Algorithm, which varies the pattern of adjacent pixels on every frame, should be used. This is explained in detail. For the 16 gray level, FRC should have the following relationship between gray level and FRC. The $15^{th}$ gray level should always have pixel-on, and the $14^{th}$ gray level should have 6 times pixel-on and one times pixel-off, and the $13^{th}$ gray level should have 4 times pixel-on and one times pixel-off, ,,,,,,,, , and the $0^{th}$ gray level should always have pixel-off as shown in Table 15-2.

**Table 15-2. Dither Duty Cycle Examples**

| Pre-dithered Data (gray level number) | Duty Cycle | Pre-dithered Data (gray level number) | Duty Cycle |
|---|---|---|---|
| 15 | 1 | 7 | 1/2 |
| 14 | 6/7 | 6 | 3/7 |
| 13 | 4/5 | 5 | 2/5 |
| 12 | 3/4 | 4 | 1/3 |
| 11 | 5/7 | 3 | 1/4 |
| 10 | 2/3 | 2 | 1/5 |
| 9 | 3/5 | 1 | 1/7 |
| 8 | 4/7 | 0 | 0 |

### Display Types

The LCD controller supports 3 types of LCD drivers: 4-bit dual scan, 4-bit single scan, and 8-bit single scan display mode. Figure 15-2 shows these 3 different display types for monochrome displays, and Figure 15-3 show these 3 different display types for color displays.

#### 4-bit Dual Scan Display Type

A 4-bit dual scan display uses 8 parallel data lines to shift data to both the upper and lower halves of the display at the same time. The 4 bits of data in the 8 parallel data lines are shifted to the upper half and 4 bits of data is shifted to the lower half, as shown in Figure 15-2. The end of frame is reached when each half of the display has been shifted and transferred. The 8 pins (VD[7:0]) for the LCD output from the LCD controller can be directly connected to the LCD driver.

#### 4-bit Single Scan Display Type

A 4-bit single scan display uses 4 parallel data lines to shift data to successive single horizontal lines of the display at a time, until the entire frame has been shifted and transferred. The 4 pins (VD[3:0]) for the LCD output from the LCD controller can be directly connected to the LCD driver, and the 4 pins (VD[7:4]) for the LCD output are not used.

#### 8-bit Single Scan Display Type

An 8-bit single scan display uses 8 parallel data lines to shift data to successive single horizontal lines of the display at a time, until the entire frame has been shifted and transferred. The 8 pins (VD[7:0]) for the LCD output from the LCD controller can be directly connected to the LCD driver.

#### 256 Color Displays

Color displays require 3 bits (Red, Green, and Blue) of image data per pixel, and so the number of horizontal shift registers for each horizontal line corresponds to three times the number of pixels of one horizontal line. resulting in a horizontal shift register of length 3 times the number of pixels per horizontal line This RGB is shifted to the LCD driver as consecutive bits via the parallel data lines. Figure 15-3 shows the RGB and order of the pixels in the parallel data lines for the 3 types of color displays.

#### 4096 Color Displays

Color displays require 3 bits (Red, Green, and Blue) of image data per pixel, and so the number of horizontal shift registers for each horizontal line corresponds to three times the number of pixels of one horizontal line. This RGB is shifted to the LCD driver as consecutive bits via the parallel data lines. This RGB order is determined by the sequence of video data in video buffers.

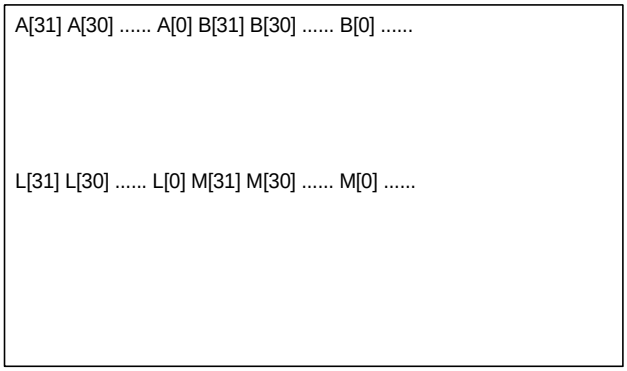
## MEMORY DATA FORMAT (STN, BSWP=0)

### Mono 4-bit Dual Scan Display:

Video Buffer Memory:

| Address | Data |
|---|---|
| 0000H | A[31:0] |
| 0004H | B[31:0] |
| • | |
| • | |
| • | |
| 1000H | L[31:0] |
| 1004H | M[31:0] |
| • | |
| • | |
| • | |

LCD Panel




### Mono 4-bit Single Scan Display & 8-bit Single Scan Display:

Video Buffer Memory:

| Address | Data |
|---|---|
| 0000H | A[31:0] |
| 0004H | B[31:0] |
| 0008H | C[31:0] |
| • | |
| • | |
| • | |

LCD Panel

**MEMORY DATA FORMAT ( STN, BSWP=0 ) (CONTINUED)**

In 4-level gray mode, 2 bits of video data correspond to 1 pixel.

In 16-level gray mode, 4 bits of video data correspond to 1 pixel.

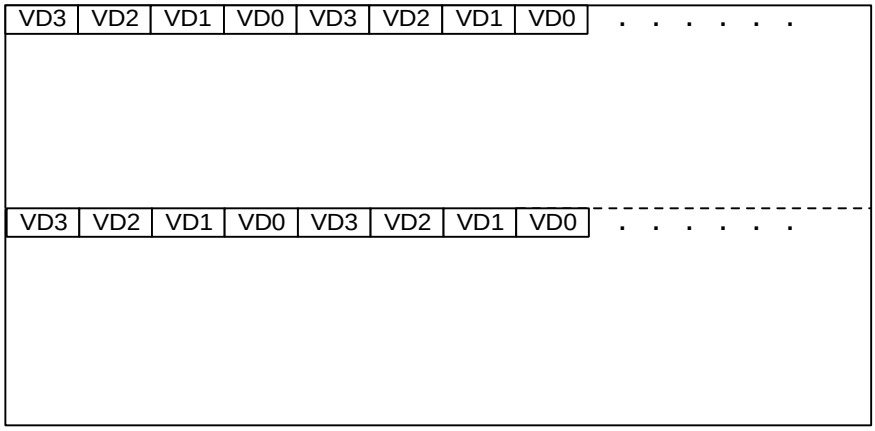
In 256 level color mode, 8 bits (3 bits of red, 3 bits of green, and 2 bits of blue) of video data correspond to 1 pixel. The color data format in a byte is as follows:

| Bit [ 7:5 ] | Bit [ 4:2 ] | Bit[1:0] |
|---|---|---|
| Red | Green | Blue |

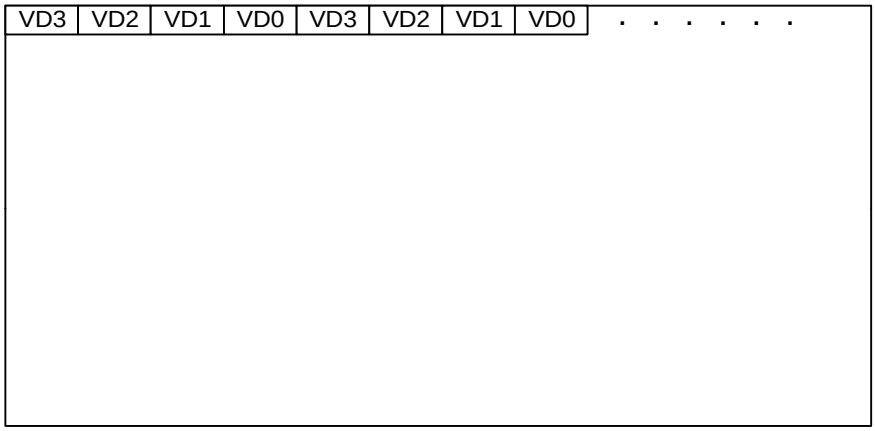
In 4096 level color mode, 12 bits (4 bits of red, 4 bits of green, 4 bits of blue) of video data correspond to 1 pixel. The following table shows color data format in words: (Video data must reside at 3 word boundaries (8 pixel), as follows)

**RGB order**

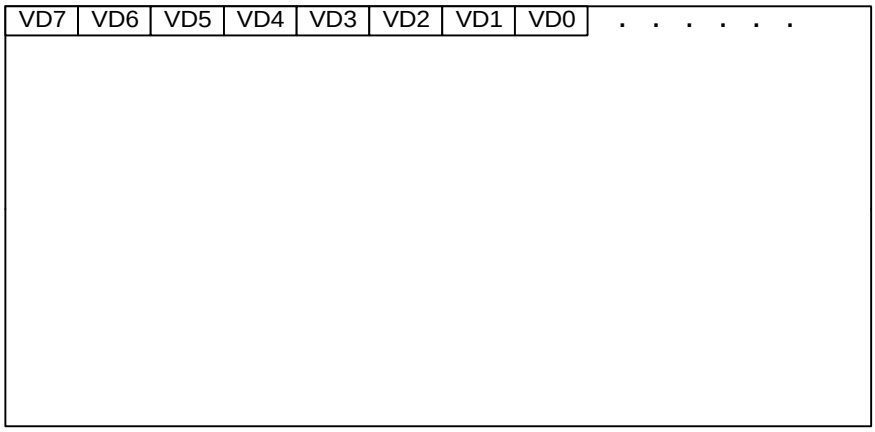
| DATA | [31:28] | [27:24] | [23:20] | [19:16] | [15:12] | [11:8] | [7:4] | [3:0] |
|---|---|---|---|---|---|---|---|---|
| Word #1 | Red( 1) | Green(1) | Blue( 1) | Red( 2) | Green( 2) | Blue( 2) | Red(3) | Green(3) |
| Word #2 | Blue(3) | Red(4) | Green(4) | Blue(4) | Red(5) | Green(5) | Blue(5) | Red(6) |
| Word #3 | Green(6) | Blue(6) | Red(7) | Green(7) | Blue(7) | Red(8) | Green(8) | Blue(8) |




4-bit Dual Scan Display

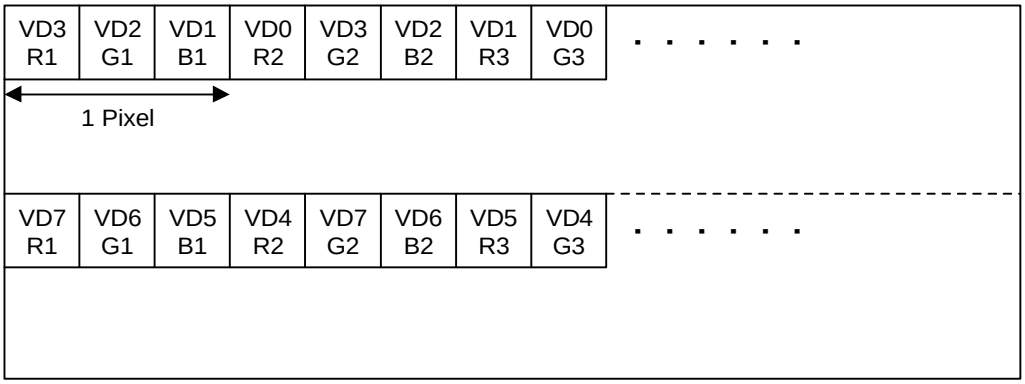



4-bit Single Scan Display

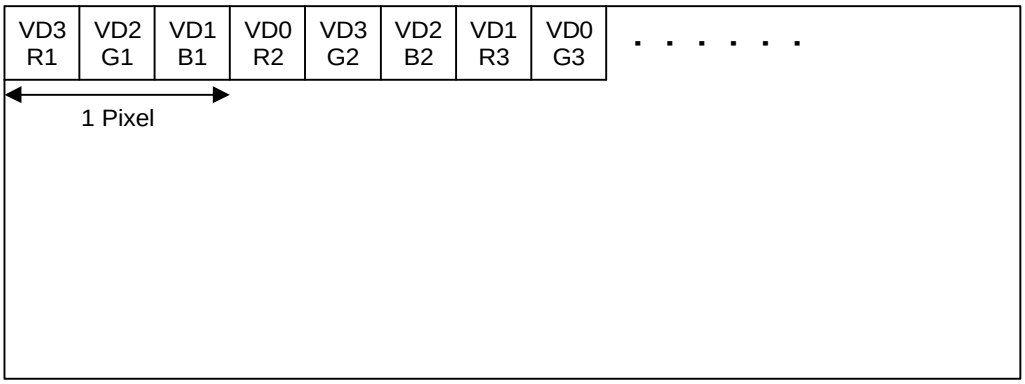



8-bit Single Scan Display

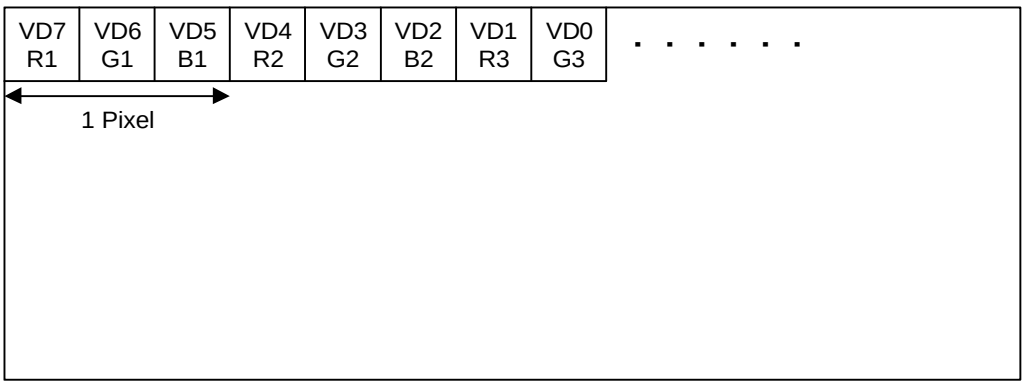
**Figure 15-2. Monochrome Display Types (STN)**




4-bit Dual Scan Display




4-bit Single Scan Display




8-bit Single Scan Display

**Figure 15-3. Color Display Types (STN)**

### Timing Requirements

Image data should be transferred from the memory to the LCD driver using the VD[7:0] signal. VCLK signal is used to clock the data into the LCD driver's shift register. After each horizontal line of data has been shifted into the LCD driver's shift register, the VLINE signal is asserted to display the line on the panel.

The VM signal provides an AC signal for the display. The LCD uses the signal to alternate the polarity of the row and column voltages, which are used to turn the pixels on and off, because the LCD plasma tends to deteriorate whenever subjected to a DC voltage. It can be configured to toggle on every frame or to toggle every programmable number of VLINE signals.

Figure 15-4 shows the timing requirements for the LCD driver interface.

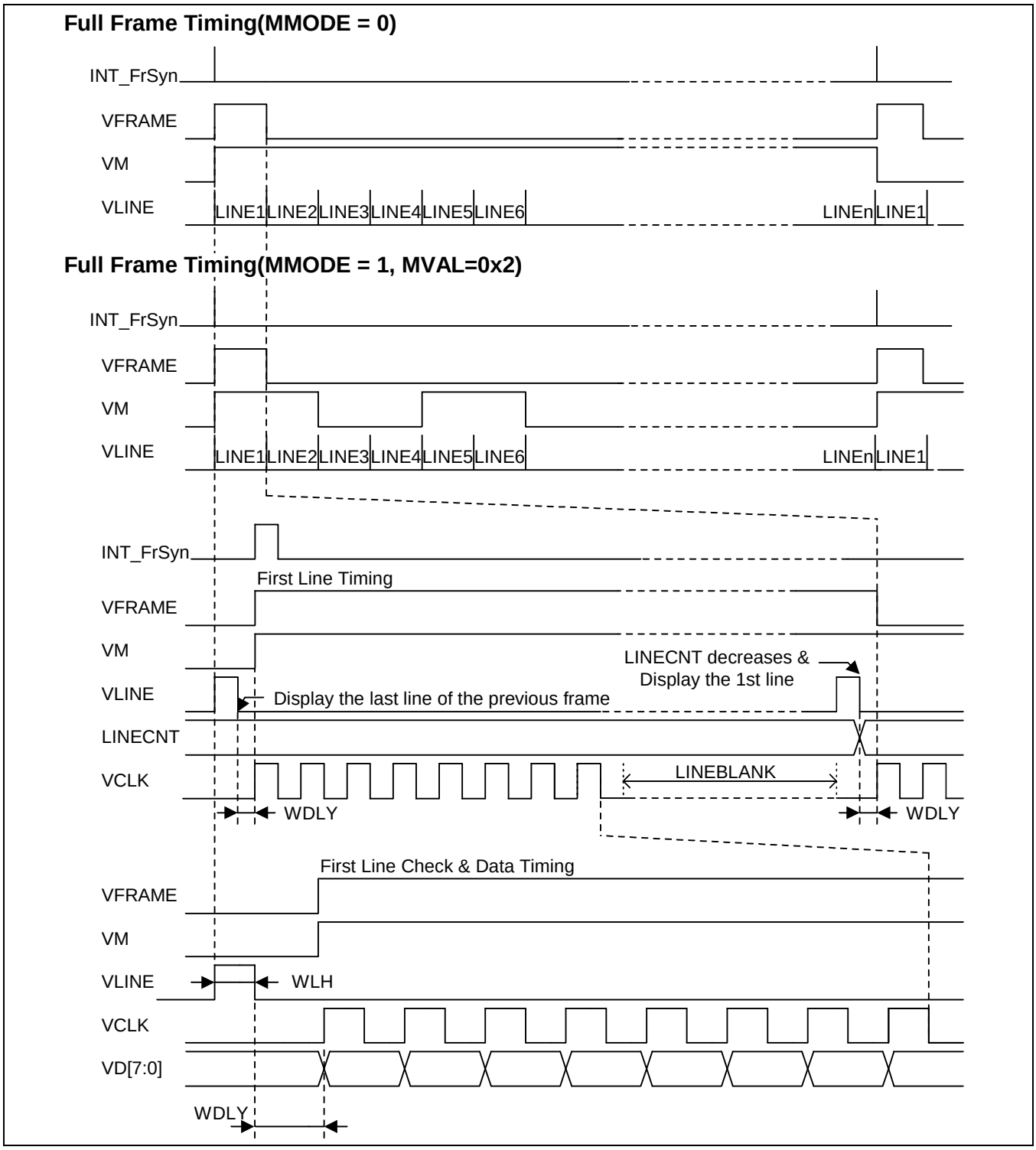



**Figure 15-4. 8-bit Single Scan Display Type STN LCD Timing**

## TFT LCD CONTROLLER OPERATION

The TIMEGEN generates the control signals for LCD driver, such as VSYNC, HSYNC, VCLK, VDEN, and LEND signal. These control signals are highly related with the configurations on the LCDCON1/2/3/4/5 registers in the REGBANK. Base on these programmable configurations on the LCD control registers in the REGBANK, the TIMEGEN can generate the programmable control signals suitable for the support of many different types of LCD drivers.

The VSYNC signal is asserted to cause the LCD's line pointer to start over at the top of the display.

The VSYNC and HSYNC pulse generation depends on the configurations of both the HOZVAL field and the LINEVAL field in the LCDCON2/3 registers. The HOZVAL and LINEVAL can be determined by the size of the LCD panel according to the following equations:

HOZVAL = (Horizontal display size) -1

LINEVAL = (Vertical display size) -1

The rate of VCLK signal depends on the CLKVAL field in the LCDCON1 register. Table 15-3 defines the relationship of VCLK and CLKVAL. The minimum value of CLKVAL is 0.

VCLK(Hz)=HCLK/[(CLKVAL+1)x2]

The frame rate is VSYNC signal frequency. The frame rate is related with the field of VSYNC, VBPD, VFPD, LINEVAL, HSYNC, HBPD, HFPD, HOZVAL, and CLKVAL in LCDCON1 and LCDCON2/3/4 registers. Most LCD drivers need their own adequate frame rate. The frame rate is calculated as follows:

Frame Rate = 1/ [ { (VSPW+1) + (VBPD+1) + (LIINEVAL + 1) + (VFPD+1) } x {(HSPW+1) + (HBPD +1) + (HFPD+1) + (HOZVAL + 1) } x { 2 x ( CLKVAL+1 ) / ( HCLK ) } ]

**Table 15-3. Relation between VCLK and CLKVAL (TFT, HCLK=60MHz)**

| CLKVAL | 60MHz/X | VCLK |
|---|---|---|
| 1 | 60 MHz/4 | 15.0 MHz |
| 2 | 60 MHz/6 | 10.0 MHz |
| : | : | : |
| 1023 | 60 MHz/2048 | 30.0 kHz |

### VIDEO OPERATION

The TFT LCD controller within the S3C2440X supports 1, 2, 4 or 8 bpp (bit per pixel) palettized color displays and 16 or 24 bpp non-palettized true-color displays.

#### 256 Color Palette

The S3C2440X can support the 256 color palette for various selection of color mapping, providing flexible operation for users.

### MEMORY DATA FORMAT (TFT)

This section includes some examples of each display mode.

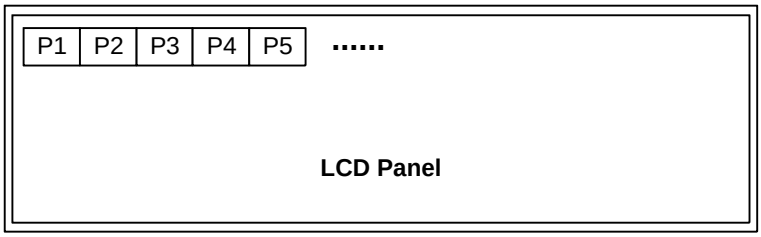
**24BPP Display**

(BSWP = 0, HWSWP = 0, BPP24BL = 0)

| | D[31:24] | D[23:0] |
|---|---|---|
| 000H | Dummy Bit | P1 |
| 004H | Dummy Bit | P2 |
| 008H | Dummy Bit | P3 |
| ... | | |

(BSWP = 0, HWSWP = 0, BPP24BL = 1)

| | D[31:8] | D[7:0] |
|---|---|---|
| 000H | P1 | Dummy Bit |
| 004H | P2 | Dummy Bit |
| 008H | P3 | Dummy Bit |
| ... | | |




**VD Pin Descriptions at 24BPP**

| VD | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RED | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | | | | | | |
| GREEN | | | | | | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | |
| BLUE | | | | | | | | | | | | | | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

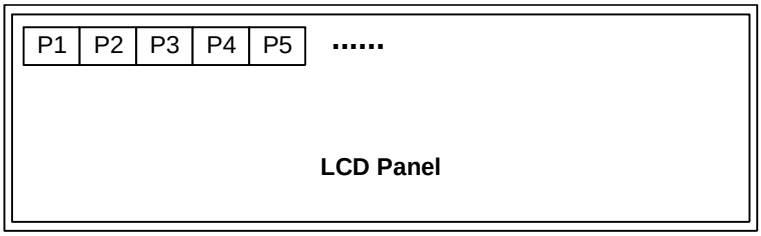
**16BPP Display**

(BSWP = 0, HWSWP = 0)

| | D[31:16] | D[15:0] |
|---|---|---|
| 000H | P1 | P2 |
| 004H | P3 | P4 |
| 008H | P5 | P6 |
| ... | | |

(BSWP = 0, HWSWP = 1)

| | D[31:16] | D[15:0] |
|---|---|---|
| 000H | P2 | P1 |
| 004H | P4 | P3 |
| 008H | P6 | P5 |
| ... | | |




**VD Pin Descriptions at 16BPP**

(5:6:5)

| VD | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RED | 4 | 3 | 2 | 1 | 0 | NC | | | | | | | | | NC | | | | | | | NC | | |
| GREEN | | | | | | | | | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | |
| BLUE | | | | | | | | | | | | | | | | | 4 | 3 | 2 | 1 | 0 | | | |

(5:5:5:I)

| VD | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RED | 4 | 3 | 2 | 1 | 0 | I | NC | | | | | | | | NC | | | | | | | | NC | |
| GREEN | | | | | | | | | 4 | 3 | 2 | 1 | 0 | I | | | | | | | | | | |
| BLUE | | | | | | | | | | | | | | | | | 4 | 3 | 2 | 1 | 0 | I | | |

NOTE : The unused VD pins can be used as GPIO

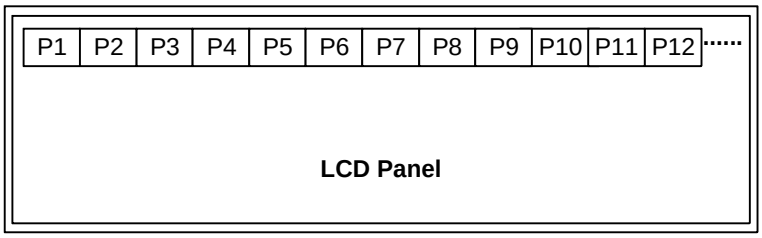
**8BPP Display**

(BSWP = 0, HWSWP = 0)

| | D[31:24] | D[23:16] | D[15:8] | D[7:0] |
|---|---|---|---|---|
| 000H | P1 | P2 | P3 | P4 |
| 004H | P5 | P6 | P7 | P8 |
| 008H | P9 | P10 | P11 | P12 |
| ... | | | | |

(BSWP = 1, HWSWP = 0)

| | D[31:24] | D[23:16] | D[15:8] | D[7:0] |
|---|---|---|---|---|
| 000H | P4 | P3 | P2 | P1 |
| 004H | P8 | P7 | P6 | P5 |
| 008H | P12 | P11 | P10 | P9 |
| ... | | | | |



**4BPP Display**

(BSWP = 0, HWSWP = 0)

| | D[31:28] | D[27:24] | D[23:20] | D[19:16] | D[15:12] | D[11:8] | D[7:4] | D[3:0] |
|---|---|---|---|---|---|---|---|---|
| 000H | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 |
| 004H | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 |
| 008H | P17 | P18 | P19 | P20 | P21 | P22 | P23 | P24 |
| ... | | | | | | | | |

(BSWP = 1, HWSWP = 0)

| | D[31:28] | D[27:24] | D[23:20] | D[19:16] | D[15:12] | D[11:8] | D[7:4] | D[3:0] |
|---|---|---|---|---|---|---|---|---|
| 000H | P7 | P8 | P5 | P6 | P3 | P4 | P1 | P2 |
| 004H | P15 | P16 | P13 | P14 | P11 | P12 | P9 | P10 |
| 008H | P23 | P24 | P21 | P22 | P19 | P20 | P17 | P18 |
| ... | | | | | | | | |

**2BPP Display**

(BSWP = 0, HWSWP = 0)

| D | [31:30] | [29:28] | [27:26] | [25:24] | [23:22] | [21:20] | [19:18] | [17:16] |
|---|---|---|---|---|---|---|---|---|
| 000H | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 |
| 004H | P17 | P18 | P19 | P20 | P21 | P22 | P23 | P24 |
| 008H | P33 | P34 | P35 | P36 | P37 | P38 | P39 | P40 |
| ... | | | | | | | | |

| D | [15:14] | [13:12] | [11:10] | [9:8] | [7:6] | [5:4] | [3:2] | [1:0] |
|---|---|---|---|---|---|---|---|---|
| 000H | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 |
| 004H | P25 | P26 | P27 | P28 | P29 | P30 | P31 | P32 |
| 008H | P41 | P42 | P43 | P44 | P45 | P46 | P47 | P48 |
| ... | | | | | | | | |

## 256 PALETTE USAGE (TFT)

### Palette Configuration and Format Control

The S3C2440X provides 256 color palette for TFT LCD Control.
The user can select 256 colors from the 64K colors in these two formats.
The 256 color palette consists of the 256 (depth) x 16-bit SPSRAM. The palette supports 5:6:5 (R:G:B) format and 5:5:5:1(R:G:B:I) format.
When the user uses 5:5:5:1 format, the intensity data(I) is used as a common LSB bit of each RGB data. So, 5:5:5:1 format is the same as R(5+I):G(5+I):B(5+I) format.
In 5:5:5:1 format, for example, the user can write the palette as in Table 15-5 and then connect VD pin to TFT LCD panel(R(5+I)=VD[23:19]+VD[18], VD[10] or VD[2], G(5+I)=VD[15:11]+ VD[18], VD[10] or VD[2], B(5+I)=VD[7:3]+ VD[18], VD[10] or VD[2].), and set FRM565 of LCDCON5 register to 0.

**Table 15-4. 5:6:5 Format**

| INDEX\Bit Pos. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00H | R4 | R3 | R2 | R1 | R0 | G5 | G4 | G3 | G2 | G1 | G0 | B4 | B3 | B2 | B1 | B0 | [1)]0X4D000400 |
| 01H | R4 | R3 | R2 | R1 | R0 | G5 | G4 | G3 | G2 | G1 | G0 | B4 | B3 | B2 | B1 | B0 | 0X4D000404 |
| ..... | | | | | | | | | | | | | | | | | ..... |
| FFH | R4 | R3 | R2 | R1 | R0 | G5 | G4 | G3 | G2 | G1 | G0 | B4 | B3 | B2 | B1 | B0 | 0X4D0007FC |
| **Number of VD** | 23 | 22 | 21 | 20 | 19 | 15 | 14 | 13 | 12 | 11 | 10 | 7 | 6 | 5 | 4 | 3 | |

**Table 15-5. 5:5:5:1 Format**

| INDEX\Bit Pos. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00H | R4 | R3 | R2 | R1 | R0 | G4 | G3 | G2 | G1 | G0 | B4 | B3 | B2 | B1 | B0 | I | 0X4D000400 |
| 01H | R4 | R3 | R2 | R1 | R0 | G4 | G3 | G2 | G1 | G0 | B4 | B3 | B2 | B1 | B0 | I | 0X4D000404 |
| ..... | | | | | | | | | | | | | | | | | ..... |
| FFH | R4 | R3 | R2 | R1 | R0 | G4 | G3 | G2 | G1 | G0 | B4 | B3 | B2 | B1 | B0 | I | 0X4D0007FC |
| **Number of VD** | 23 | 22 | 21 | 20 | 19 | 15 | 14 | 13 | 12 | 11 | 7 | 6 | 5 | 4 | 3 | [2)] | |

**Notes:**
1. 0x4D000400 is Palette start address.
2. VD18, VD10 and VD2 have the same output value, I.
3. DATA[31:16] is invalid.

### Palette Read/Write

When the user performs Read/Write operation on the palette, HSTATUS and VSTATUS of LCDCON5 register must be checked, for Read/Write operation is prohibited during the ACTIVE status of HSTATUS and VSTATUS.

### Temporary Palette Configuration

The S3C2440X allows the user to fill a frame with one color without complex modification to fill the one color to the frame buffer or palette. The one colored frame can be displayed by the writing a value of the color which is displayed on LCD panel to TPALVAL of TPAL register and enable TPALEN.

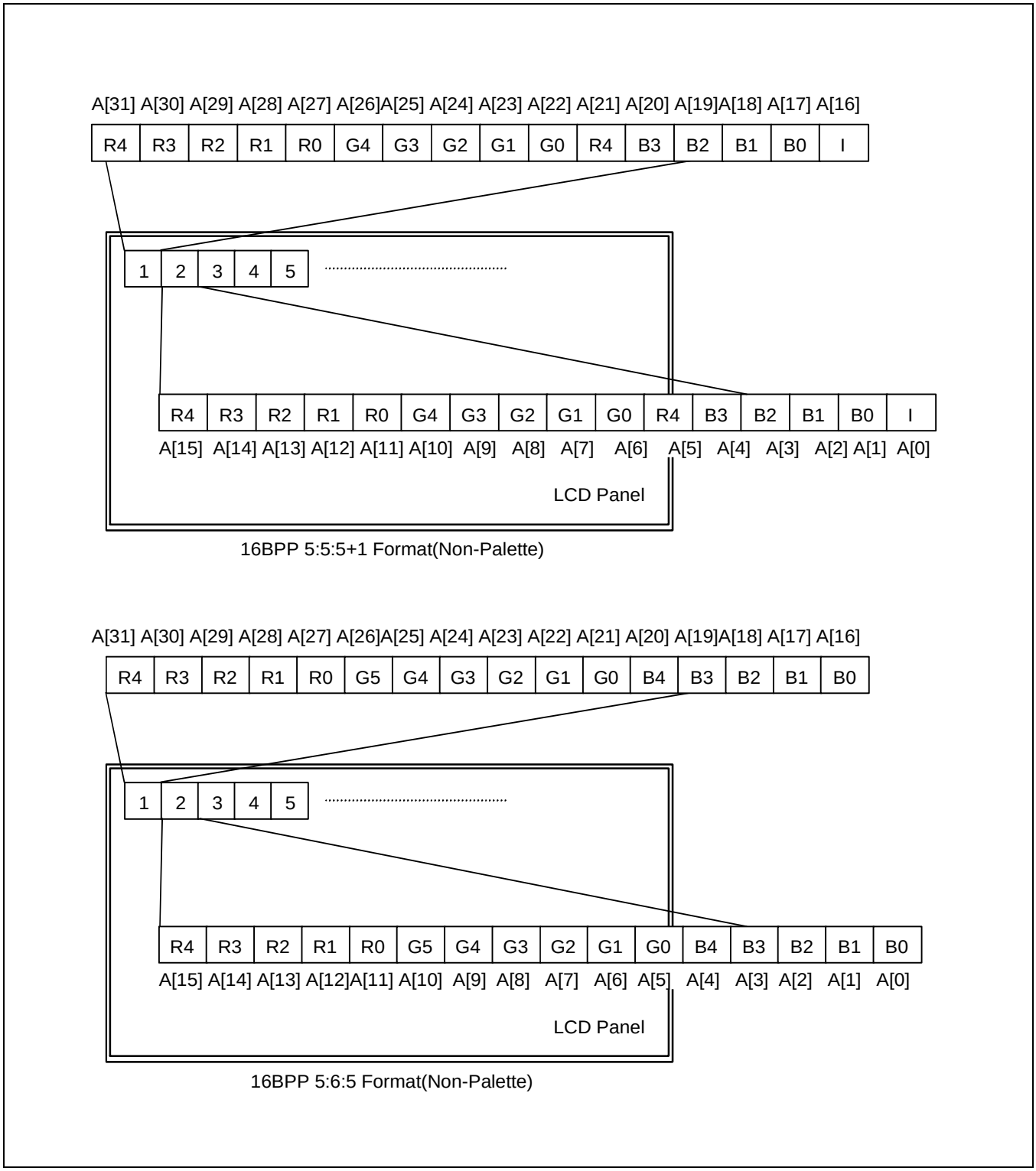



**Figure 15-5. 16BPP Display Types (TFT)**

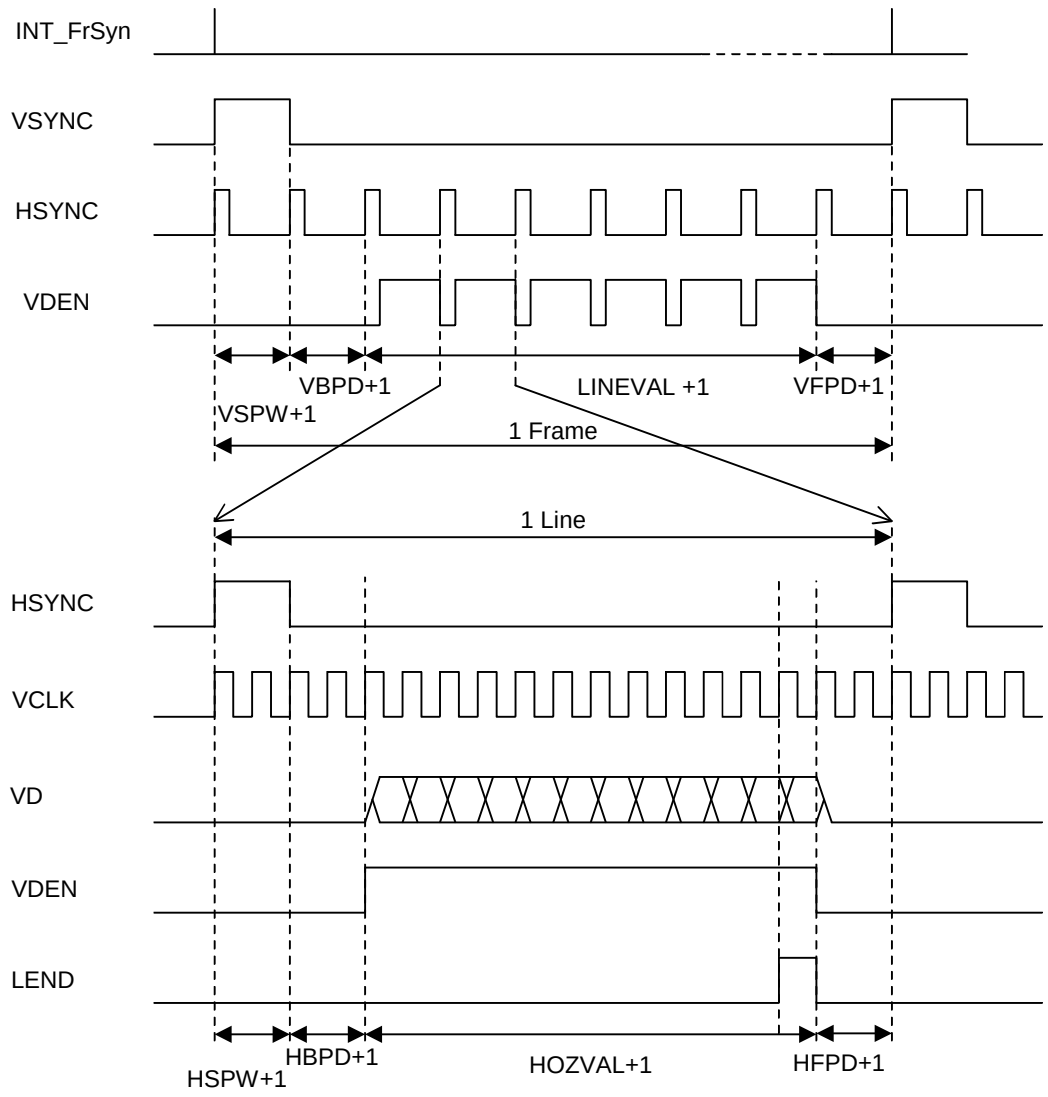



**Figure 15-6. TFT LCD Timing Example**

**SAMSUNG TFT LCD PANEL (3.5" PORTRAIT / 256K COLOR / REFLECTIVE A-SI/TRANSFLECTIVE A-SI TFT LCD)**

The S3C2440X supports following SEC TFT LCD panels.

1. SAMSUNG 3.5" Portrait / 256K Color /Reflective a-Si TFT LCD.
   LTS350Q1-PD1: TFT LCD panel with touch panel and front light unit
   LTS350Q1-PD2: TFT LCD panel only

2. SAMSUNG 3.5" Portrait / 256K Color /Transflective a-Si TFT LCD.
   LTS350Q1-PE1: TFT LCD panel with touch panel and front light unit
   LTS350Q1-PE2: TFT LCD panel only

The S3C2440X provides timing signals as follows to use LTS350Q1-PD1 / PD2 and LTS350Q1-PE1 / PE2

| LTS350Q1-PD1 / PD2 | LTS350Q1-PE1 / PE2 |
|---|---|
| STH: Horizontal Start Pulse<br>TP: Source Driver Data Load Pulse<br>INV: Digital Data Inversion<br>LCD_HCLK: Horizontal Sampling Clock<br>CPV: Vertical Shift Clock<br>STV: Vertical Start Pulse<br>OE: Gate On Enable<br>REV: Inversion Signal<br>REVB: Inversion Signal | STH: Horizontal Start Pulse<br>TP: Source Driver Data Load Pulse<br>INV: Digital Data Inversion<br>LCD_HCLK: Horizontal Sampling Clock<br>CPV: Vertical Shift Clock<br>STV: Vertical Start Pulse<br>LCCINV: Source drive IC sampling inversion signal<br>REV: VCOM modulation Signal<br>REVB: Inversion Signal |

So, LTS350Q1-PD1/2 and PE1/2 can be connected with the S3C2440X without using the additional timing control logic.
But the user should additionally apply Vcom generator circuit, various voltages, INV signal and Gray scale voltage generator circuit, which is recommended by PRODUCT INFORMATION (SPEC) of LTS350Q1-PD1/2 and PE1/2.
Detailed timing diagram is also described in PRODUCT INFORMATION (SPEC) of LTS350Q1-PD1/2 and PE1/2.

Refer to the documentation (PRODUCT INFORMATION of LTS350Q1-PD1/2 and PE1/2), which is prepared by AMLCD Technical Customer Center of Samsung Electronics Co., LTD.

**CAUTION:**
The S3C2440X has HCLK, working as the clock of AHB bus.
Accidentally, SEC TFT LCD panel (LTS350Q1-PD1/2 and PE1/2) has Horizontal Sampling Clock (HCLK).
These two HCLKs may cause a confusion. So, note that HCLK of the S3C2440X is HCLK and other HCLK of the LTS350 is LCD_HCLK.

**Check that the HCLK of SEC TFT LCD panel (LTS350Q1-PD1/2 and PE1/2) is changed to LCD_HCLK.**

**VIRTUAL DISPLAY (TFT/STN)**

The S3C2440X supports hardware horizontal or vertical scrolling. If the screen is scrolled, the fields of LCDBASEU and LCDBASEL in LCDSADDR1/2 registers need to be changed (see Figure 15-8), except the values of PAGEWIDTH and OFFSIZE.
The video buffer in which the image is stored should be larger than the LCD panel screen in size.

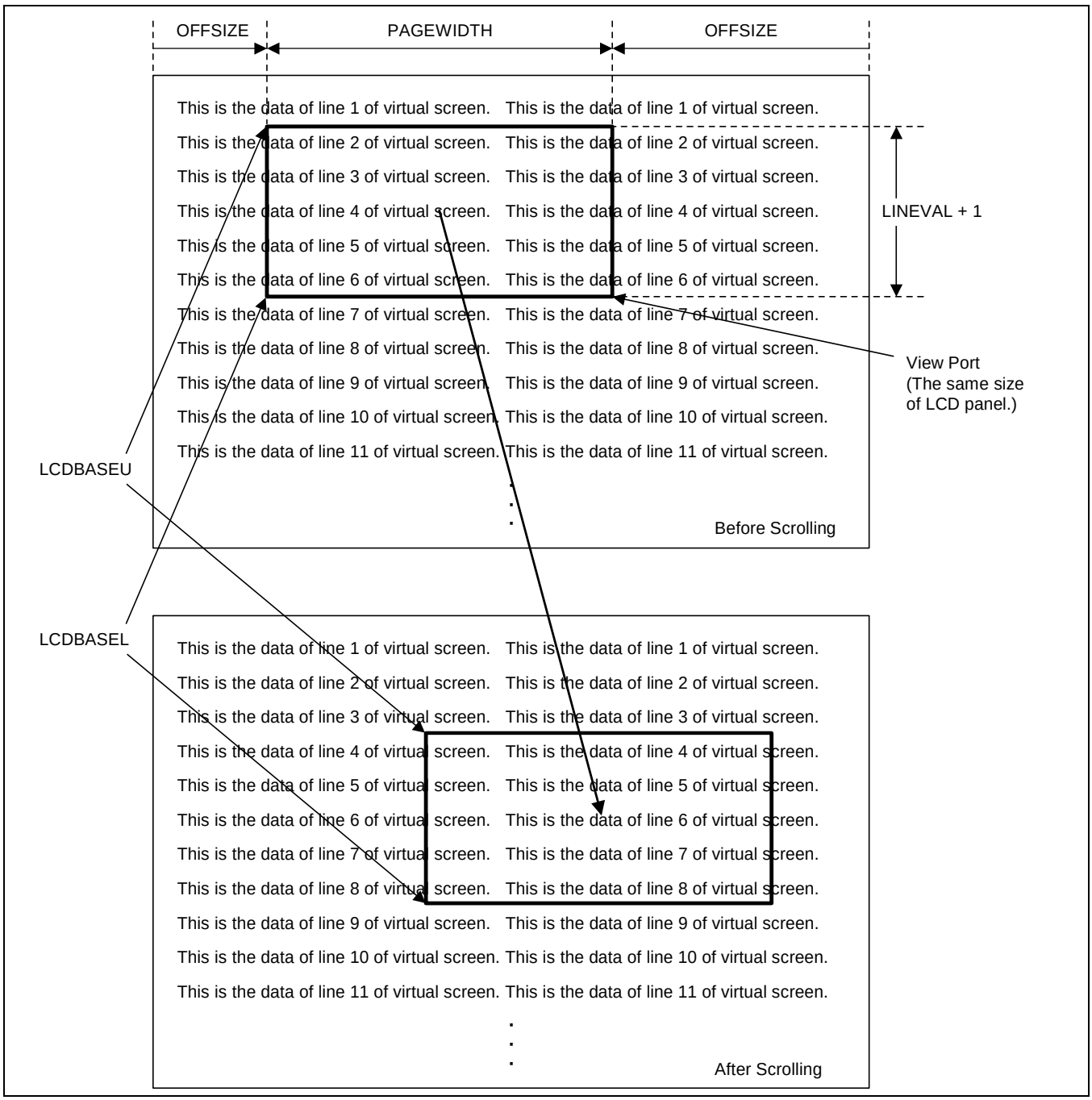



**Figure 15-7. Example of Scrolling in Virtual Display (Single Scan)**

### LCD POWER ENABLE (STN/TFT)

The S3C2440X provides Power enable (PWREN) function. When PWREN is set to make PWREN signal enabled, the output value of LCD_PWREN pin is controlled by ENVID. In other words, If LCD_PWREN pin is connected to the power on/off control pin of the LCD panel, the power of LCD panel is controlled by the setting of ENVID automatically.
The S3C2440X also supports INVPWREN bit to invert polarity of the PWREN signal.

This function is available only when LCD panel has its own power on/off control port and when port is connected to LCD_PWREN pin.

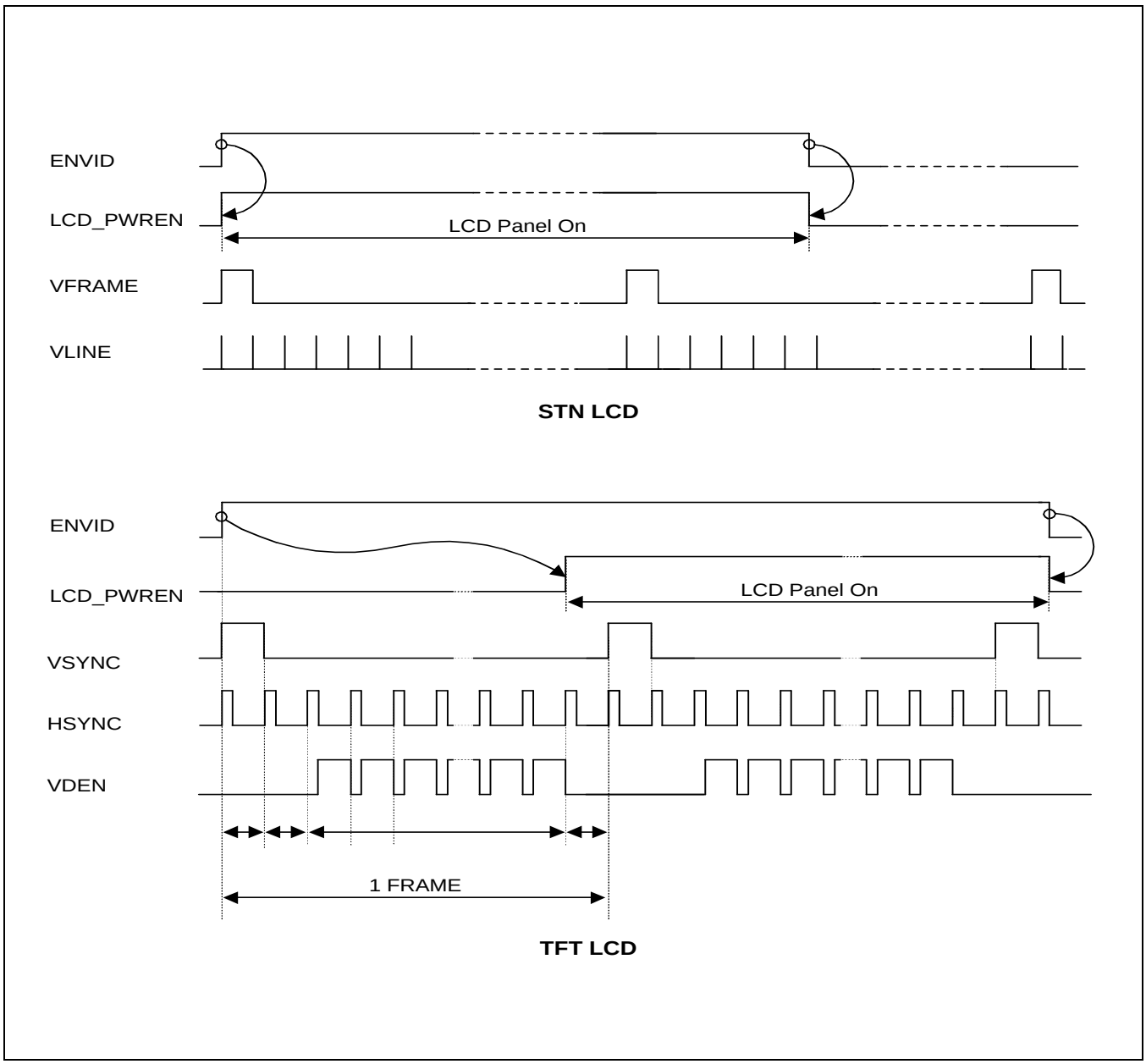



**Figure 15-8. Example of PWREN function (PWREN=1, INVPWREN=0)**

## LCD CONTROLLER SPECIAL REGISTERS

### LCD Control 1 Register

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDCON1 | 0X4D000000 | R/W | LCD control 1 register | 0x00000000 |

| LCDCON1 | Bit | Description | Initial State |
|---|---|---|---|
| LINECNT (read only) | [27:18] | Provide the status of the line counter.<br>Down count from LINEVAL to 0 | 0000000000 |
| CLKVAL | [17:8] | Determine the rates of VCLK and CLKVAL[9:0].<br>**STN**: VCLK = HCLK / (CLKVAL x 2) ( CLKVAL ≥2 )<br>**TFT**: VCLK = HCLK / [(CLKVAL+1) x 2] ( CLKVAL ≥ 0 ) | 0000000000 |
| MMODE | [7] | Determine the toggle rate of the VM.<br>0 = Each Frame 1 = The rate defined by the MVAL | 0 |
| PNRMODE | [6:5] | Select the display mode.<br>00 = 4-bit dual scan display mode (STN)<br>01 = 4-bit single scan display mode (STN)<br>10 = 8-bit single scan display mode (STN)<br>11 = TFT LCD panel | 00 |
| BPPMODE | [4:1] | Select the BPP (Bits Per Pixel) mode.<br>0000 = 1 bpp for STN, Monochrome mode<br>0001 = 2 bpp for STN, 4-level gray mode<br>0010 = 4 bpp for STN, 16-level gray mode<br>0011 = 8 bpp for STN, color mode<br>0100 = 12 bpp for STN, color mode<br>1000 = 1 bpp for TFT<br>1001 = 2 bpp for TFT<br>1010 = 4 bpp for TFT<br>1011 = 8 bpp for TFT<br>1100 = 16 bpp for TFT<br>1101 = 24 bpp for TFT | 0000 |
| ENVID | [0] | LCD video output and the logic enable/disable.<br>0 = Disable the video output and the LCD control signal.<br>1 = Enable the video output and the LCD control signal. | 0 |

**LCD Control 2 Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDCON2 | 0X4D000004 | R/W | LCD control 2 register | 0x00000000 |

| LCDCON2 | Bit | Description | Initial State |
|---|---|---|---|
| VBPD | [31:24] | **TFT**: Vertical back porch is the number of inactive lines at the start of a frame, after vertical synchronization period.<br>**STN**: These bits should be set to zero on STN LCD. | 0x00 |
| LINEVAL | [23:14] | **TFT/STN**: These bits determine the vertical size of LCD panel. | 0000000000 |
| VFPD | [13:6] | **TFT**: Vertical front porch is the number of inactive lines at the end of a frame, before vertical synchronization period.<br>**STN**: These bits should be set to zero on STN LCD. | 00000000 |
| VSPW | [5:0] | **TFT**: Vertical sync pulse width determines the VSYNC pulse's high level width by counting the number of inactive lines.<br>**STN**: These bits should be set to zero on STN LCD. | 000000 |

### LCD Control 3 Register

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDCON3 | 0X4D000008 | R/W | LCD control 3 register | 0x00000000 |

| LCDCON3 | Bit | Description | Initial state |
|---|---|---|---|
| HBPD (TFT) | [25:19] | **TFT**: Horizontal back porch is the number of VCLK periods between the falling edge of HSYNC and the start of active data. | 0000000 |
| WDLY (STN) | | **STN**: WDLY[1:0] bits determine the delay between VLINE and VCLK by counting the number of the HCLK. WDLY[7:2] are reserved.<br>00 = 16 HCLK, 01 = 32 HCLK, 10 = 48 HCLK, 11 = 64 HCLK | |
| HOZVAL | [18:8] | **TFT/STN**: These bits determine the horizontal size of LCD panel.<br>HOZVAL has to be determined to meet the condition that total bytes of 1 line are 4n bytes. If the x size of LCD is 120 dot in mono mode, x=120 cannot be supported because 1 line consists of 15 bytes. Instead, x=128 in mono mode can be supported because 1 line is composed of 16 bytes (2n). LCD panel driver will discard the additional 8 dot. | 00000000000 |
| HFPD (TFT) | [7:0] | **TFT**: Horizontal front porch is the number of VCLK periods between the end of active data and the rising edge of HSYNC. | 0X00 |
| LINEBLANK (STN) | | **STN**: These bits indicate the blank time in one horizontal line duration time. These bits adjust the rate of the VLINE finely.<br>The unit of LINEBLANK is HCLK x 8.<br>Ex) If the value of LINEBLANK is 10, the blank time is inserted to VCLK during 80 HCLK. | |

**LCD Control 4 Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDCON4 | 0X4D00000C | R/W | LCD control 4 register | 0x00000000 |

| LCDCON4 | Bit | Description | Initial state |
|---|---|---|---|
| MVAL | [15:8] | **STN**: These bit define the rate at which the VM signal will toggle if the MMODE bit is set to logic '1'. | 0X00 |
| HSPW(TFT) | [7:0] | **TFT**: Horizontal sync pulse width determines the HSYNC pulse's high level width by counting the number of the VCLK. | 0X00 |
| WLH(STN) | | **STN**: WLH[1:0] bits determine the VLINE pulse's high level width by counting the number of the HCLK.<br>WLH[7:2] are reserved.<br>00 = 16 HCLK, 01 = 32 HCLK, 10 = 48 HCLK, 11 = 64 HCLK | |

**LCD Control 5 Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDCON5 | 0X4D000010 | R/W | LCD control 5 register | 0x00000000 |

| LCDCON5 | Bit | Description | Initial state |
|---|---|---|---|
| Reserved | [31:17] | This bit is reserved and the value should be '0'. | 0 |
| VSTATUS | [16:15] | **TFT**: Vertical Status (read only).<br>00 = VSYNC 01 = BACK Porch<br>10 = ACTIVE 11 = FRONT Porch | 00 |
| HSTATUS | [14:13] | **TFT**: Horizontal Status (read only).<br>00 = HSYNC 01 = BACK Porch<br>10 = ACTIVE 11 = FRONT Porch | 00 |
| BPP24BL | [12] | **TFT**: This bit determines the order of 24 bpp video memory.<br>0 = LSB valid 1 = MSB Valid | 0 |
| FRM565 | [11] | **TFT**: This bit selects the format of 16 bpp output video data.<br>0 = 5:5:5:1 Format 1 = 5:6:5 Format | 0 |
| INVVCLK | [10] | **STN/TFT**: This bit controls the polarity of the VCLK active edge.<br>0 = The video data is fetched at VCLK falling edge<br>1 = The video data is fetched at VCLK rising edge | 0 |
| INVVLINE | [9] | **STN/TFT**: This bit indicates the VLINE/HSYNC pulse polarity.<br>0 = Normal 1 = Inverted | 0 |
| INVVFRAME | [8] | **STN/TFT**: This bit indicates the VFRAME/VSYNC pulse polarity.<br>0 = Normal 1 = Inverted | 0 |
| INVVD | [7] | **STN/TFT**: This bit indicates the VD (video data) pulse polarity.<br>0 = Normal 1 = VD is inverted. | 0 |

**LCD Control 5 Register (Continued)**

| LCDCON5 | Bit | Description | Initial state |
|---|---|---|---|
| INVVDEN | [6] | **TFT**: This bit indicates the VDEN signal polarity.<br>0 = normal 1 = inverted | 0 |
| INVPWREN | [5] | **STN/TFT**: This bit indicates the PWREN signal polarity.<br>0 = normal 1 = inverted | 0 |
| INVLEND | [4] | **TFT**: This bit indicates the LEND signal polarity.<br>0 = normal 1 = inverted | 0 |
| PWREN | [3] | **STN/TFT**: LCD_PWREN output signal enable/disable.<br>0 = Disable PWREN signal 1 = Enable PWREN signal | 0 |
| ENLEND | [2] | **TFT**: LEND output signal enable/disable.<br>0 = Disable LEND signal 1 = Enable LEND signal | 0 |
| BSWP | [1] | **STN/TFT**: Byte swap control bit.<br>0 = Swap Disable 1 = Swap Enable | 0 |
| HWSWP | [0] | **STN/TFT**: Half-Word swap control bit.<br>0 = Swap Disable 1 = Swap Enable | 0 |

**FRAME BUFFER START ADDRESS 1 REGISTER**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDSADDR1 | 0X4D000014 | R/W | **STN/TFT**: Frame buffer start address 1 register | 0x00000000 |

| LCDSADDR1 | Bit | Description | Initial State |
|---|---|---|---|
| LCDBANK | [29:21] | These bits indicate A[30:22] of the bank location for the video buffer in the system memory. LCDBANK value cannot be changed even when moving the view port. LCD frame buffer should be within aligned 4MB region, which ensures that LCDBANK value will not be changed when moving the view port. So, care should be taken to use the malloc() function. | 0x00 |
| LCDBASEU | [20:0] | For dual-scan LCD : These bits indicate A[21:1] of the start address of the upper address counter, which is for the upper frame memory of dual scan LCD or the frame memory of single scan LCD.<br>For single-scan LCD : These bits indicate A[21:1] of the start address of the LCD frame buffer. | 0x000000 |

**FRAME Buffer Start Address 2 Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDSADDR2 | 0X4D000018 | R/W | **STN/TFT**: Frame buffer start address 2 register | 0x00000000 |

| LCDSADDR2 | Bit | Description | Initial State |
|---|---|---|---|
| LCDBASEL | [20:0] | For dual-scan LCD: These bits indicate A[21:1] of the start address of the lower address counter, which is used for the lower frame memory of dual scan LCD.<br>For single scan LCD: These bits indicate A[21:1] of the end address of the LCD frame buffer.<br>LCDBASEL = ((the frame end address) >>1) + 1<br>= LCDBASEU +<br>(PAGEWIDTH+OFFSIZE) x (LINEVAL+1) | 0x0000 |

**Note:** Users can change the LCDBASEU and LCDBASEL values for scrolling while the LCD controller is turned on. But, users must not change the value of the LCDBASEU and LCDBASEL registers at the end of FRAME by referring to the LINECNT field in LCDCON1 register, for the LCD FIFO fetches the next frame data prior to the change in the frame.
So, if you change the frame, the pre-fetched FIFO data will be obsolete and LCD controller will display an incorrect screen. To check the LINECNT, interrupts should be masked. If any interrupt is executed just after reading LINECNT, the read LINECNT value may be obsolete because of the execution time of Interrupt Service Routine (ISR).

**FRAME Buffer Start Address 3 Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDSADDR3 | 0X4D00001C | R/W | **STN/TFT**: Virtual screen address set | 0x00000000 |

| LCDSADDR3 | Bit | Description | Initial State |
|---|---|---|---|
| OFFSIZE | [21:11] | Virtual screen offset size (the number of half words).<br>This value defines the difference between the address of the last half word displayed on the previous LCD line and the address of the first half word to be displayed in the new LCD line. | 00000000000 |
| PAGEWIDTH | [10:0] | Virtual screen page width (the number of half words).<br>This value defines the width of the view port in the frame. | 000000000 |

**Note:** The values of PAGEWIDTH and OFFSIZE must be changed when ENVID bit is 0.

Example 1. LCD panel = 320 x 240, 16gray, single scan
Frame start address = 0x0c500000
Offset dot number = 2048 dots ( 512 half words )

LINEVAL = 240-1 = 0xef
PAGEWIDTH = 320 x 4 / 16 = 0x50
OFFSIZE = 512 = 0x200
LCDBANK = 0x0c500000 >> 22 = 0x31
LCDBASEU = 0x100000 >> 1 = 0x80000
LCDBASEL = 0x80000 + ( 0x50 + 0x200 ) x ( 0xef + 1 ) = 0xa2b00

Example 2. LCD panel = 320 x 240, 16gray, dual scan
Frame start address = 0x0c500000
Offset dot number = 2048 dots ( 512 half words )

LINEVAL = 120-1 = 0x77
PAGEWIDTH = 320 x 4 / 16 = 0x50
OFFSIZE = 512 = 0x200
LCDBANK = 0x0c500000 >> 22 = 0x31
LCDBASEU = 0x100000 >> 1 = 0x80000
LCDBASEL = 0x80000 + ( 0x50 + 0x200 ) x ( 0x77 + 1 ) = 0x91580

Example 3. LCD panel = 320*240, color, single scan
Frame start address = 0x0c500000
Offset dot number = 1024 dots ( 512 half words )

LINEVAL = 240-1 = 0xef
PAGEWIDTH = 320 x 8 / 16 = 0xa0
OFFSIZE = 512 = 0x200
LCDBANK = 0x0c500000 >> 22 = 0x31
LCDBASEU = 0x100000 >> 1 = 0x80000
LCDBASEL = 0x80000 + ( 0xa0 + 0x200 ) x ( 0xef + 1 ) = 0xa7600

**RED Lookup Table Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| REDLUT | 0X4D000020 | R/W | **STN**: Red lookup table register | 0x00000000 |

| REDLUT | Bit | Description | Initial State |
|---|---|---|---|
| REDVAL | [31:0] | These bits define which of the 16 shades will be chosen by each of the 8 possible red combinations.<br>000 = REDVAL[3:0], 001 = REDVAL[7:4]<br>010 = REDVAL[11:8], 011 = REDVAL[15:12]<br>100 = REDVAL[19:16], 101 = REDVAL[23:20]<br>110 = REDVAL[27:24], 111 = REDVAL[31:28] | 0x00000000 |

**GREEN Lookup Table Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| GREENLUT | 0X4D000024 | R/W | **STN**: Green lookup table register | 0x00000000 |

| GREENLUT | Bit | Description | Initial State |
|---|---|---|---|
| GREENVAL | [31:0] | These bits define which of the 16 shades will be chosen by each of the 8 possible green combinations.<br>000 = GREENVAL[3:0], 001 = GREENVAL[7:4]<br>010 = GREENVAL[11:8], 011 = GREENVAL[15:12]<br>100 = GREENVAL[19:16], 101 = GREENVAL[23:20]<br>110 = GREENVAL[27:24], 111 = GREENVAL[31:28] | 0x00000000 |

**BLUE Lookup Table Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BLUELUT | 0X4D000028 | R/W | **STN**: Blue lookup table register | 0x0000 |

| BULELUT | Bit | Description | Initial State |
|---|---|---|---|
| BLUEVAL | [15:0] | These bits define which of the 16 shades will be chosen by each of the 4 possible blue combinations.<br>00 = BLUEVAL[3:0], 01 = BLUEVAL[7:4]<br>10 = BLUEVAL[11:8], 11 = BLUEVAL[15:12] | 0x0000 |

**Note:** Address from **0x14A0002C** to **0x14A00048** should not be used. This area is reserved for Test mode.

**Dithering Mode Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| DITHMODE | 0X4D00004C | R/W | **STN**: Dithering mode register.<br>This register reset value is 0x00000 But, user can change this value to 0x12210.<br>(Refer to a sample program source for the latest value of this register.) | 0x00000 |

| DITHMODE | Bit | Description | Initial state |
|---|---|---|---|
| DITHMODE | [18:0] | Use one of following value for your LCD :<br>0x00000 or 0x12210 | 0x00000 |

**Temp Palette Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TPAL | 0X4D000050 | R/W | **TFT**: Temporary palette register.<br>This register value will be video data at next frame. | 0x00000000 |

| TPAL | Bit | Description | Initial state |
|---|---|---|---|
| TPALEN | [24] | Temporary palette register enable bit.<br>0 = Disable 1 = Enable | 0 |
| TPALVAL | [23:0] | Temporary palette value register.<br>TPALVAL[23:16] : RED<br>TPALVAL[15:8] : GREEN<br>TPALVAL[7:0] : BLUE | 0x000000 |

**LCD Interrupt Pending Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDINTPND | 0X4D000054 | R/W | Indicate the LCD interrupt pending register | 0x0 |

| LCDINTPND | Bit | Description | Initial state |
|---|---|---|---|
| INT_FrSyn | [1] | LCD frame synchronized interrupt pending bit.<br>0 = The interrupt has not been requested.<br>1 = The frame has asserted the interrupt request. | 0 |
| INT_FiCnt | [0] | LCD FIFO interrupt pending bit.<br>0 = The interrupt has not been requested.<br>1 = LCD FIFO interrupt is requested when LCD FIFO reaches trigger level. | 0 |

**LCD Source Pending Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDSRCPND | 0X4D000058 | R/W | Indicate the LCD interrupt source pending register | 0x0 |

| LCDSRCPND | Bit | Description | Initial state |
|---|---|---|---|
| INT_FrSyn | [1] | LCD frame synchronized interrupt source pending bit.<br>0 = The interrupt has not been requested.<br>1 = The frame has asserted the interrupt request. | 0 |
| INT_FiCnt | [0] | LCD FIFO interrupt source pending bit.<br>0 = The interrupt has not been requested.<br>1 = LCD FIFO interrupt is requested when LCD FIFO reaches trigger level. | 0 |

**LCD Interrupt Mask Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| LCDINTMSK | 0X4D00005C | R/W | Determine which interrupt source is masked.<br>The masked interrupt source will not be serviced. | 0x3 |

| LCDINTMSK | Bit | Description | Initial state |
|---|---|---|---|
| FIWSEL | [2] | Determine the trigger level of LCD FIFO.<br>0 = 4 words 1 = 8 words | |
| INT_FrSyn | [1] | Mask LCD frame synchronized interrupt.<br>0 = The interrupt service is available.<br>1 = The interrupt service is masked. | 1 |
| INT_FiCnt | [0] | Mask LCD FIFO interrupt.<br>0 = The interrupt service is available.<br>1 = The interrupt service is masked. | 1 |

**TCON Control Register**

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TCONSEL | 0X4D000060 | R/W | This register controls the LPC3600/LCC3600 modes. | 0xF84 |

| TCONSEL | Bit | Description | Initial state |
|---|---|---|---|
| LCC_TEST2 | [11] | LCC3600 Test Mode 2 ( Read Only ) | 1 |
| LCC_TEST1 | [10] | LCC3600 Test Mode 1 ( Read Only ) | 1 |
| LCC_SEL5 | [9] | Select STV polarity | 1 |
| LCC_SEL4 | [8] | Select CPV signal pin 0 | 1 |
| LCC_SEL3 | [7] | Select CPV signal pin 1 | 1 |
| LCC_SEL2 | [6] | Select Line/Dot inversion | 0 |
| LCC_SEL1 | [5] | Select DG/Normal mode | 0 |
| LCC_EN | [4] | Determine LCC3600 Enable/Disable<br>0 = LCC3600 Disable<br>1 = LCC3600 Enable | 0 |
| CPV_SEL | [3] | Select CPV Pulse low width | 0 |
| MODE_SEL | [2] | Select DE/Sync mode<br>0 = Sync mode<br>1 = DE mode | 1 |
| RES_SEL | [1] | Select output resolution type<br>0 = 320 x 240<br>1 = 240 x 320 | 0 |
| LPC_EN | [0] | Determine LPC3600 Enable/Disable<br>0 = LPC3600 Disable<br>1 = LPC3600 Enable | 0 |

**Note :** Both LPC_EN and LCC_EN enable is not permitted. Only one TCON can be enabled at the same time.

**Register Setting Guide (STN)**

The LCD controller supports multiple screen sizes by special register setting.

The CLKVAL value determines the frequency of VCLK. This value has to be determined such that the VCLK value is greater than data transmission rate. The data transmission rate for the VD port of the LCD controller is used to determine the value of CLKVAL register.

The data transmission rate is given by the following equation:

Data transmission rate = HS x VS x FR x MV
HS: Horizontal LCD size
VS: Vertical LCD size
FR: Frame rate
MV: Mode dependent value

**Table 15-6. MV Value for Each Display Mode**

| Mode | MV Value |
|---|---|
| Mono, 4-bit single scan display | 1/4 |
| Mono, 8-bit single scan display or 4-bit dual scan display | 1/8 |
| 4 level gray, 4-bit single scan display | 1/4 |
| 4 level gray, 8-bit single scan display or 4-bit dual scan display | 1/8 |
| 16 level gray, 4-bit single scan display | 1/4 |
| 16 level gray, 8-bit single scan display or 4-bit dual scan display | 1/8 |
| Color, 4-bit single scan display | 3/4 |
| Color, 8-bit single scan display or 4-bit dual scan display | 3/8 |

The LCDBASEU register value is the first address value of the frame buffer. The lowest 4 bits must be eliminated for burst 4 word access. The LCDBASEL register value depends on LCD size and LCDBASEU. The LCDBASEL value is given by the following equation:

LCDBASEL = LCDBASEU + LCDBASEL offset

**Example 1:**

160 x 160, 4-level gray, 80 frame/sec, 4-bit single scan display, HCLK frequency is 60 MHz WLH = 1, WDLY = 1.

Data transmission rate = 160 x 160 x 80 x 1/4 = 512 kHz
CLKVAL = 58, VCLK = 517KHz
HOZVAL = 39, LINEVAL = 159

LINEBLANK =10

LCDBASEL = LCDBASEU + 3200

**Note:** The higher the system load is, the lower the cpu performance is.

**Example 2 (Virtual screen register):**

4 -level gray, Virtual screen size = 1024 x 1024, LCD size = 320 x 240, LCDBASEU = 0x64, 4-bit dual scan.

1 halfword = 8 pixels (4-level gray),
Virtual screen 1 line = 128 halfword = 1024 pixels,
LCD 1 line = 320 pixels = 40 halfword,
OFFSIZE = 128 - 40 = 88 = 0x58,
PAGEWIDTH = 40 = 0x28

LCDBASEL = LCDBASEU + (PAGEWIDTH + OFFSIZE) x (LINEVAL +1) = 100 + (40 +88) x 120 = 0x3C64

**Gray Level Selection Guide**

The S3C2440X LCD controller can generate 16 gray level using Frame Rate Control (FRC). The FRC characteristics may cause unexpected patterns in gray level. These unwanted erroneous patterns may be shown in fast response LCD or at lower frame rates.

Because the quality of LCD gray levels depends on LCD's own characteristics, the user has to select an appropriate gray level after viewing all gray levels on user's own LCD.

Select the gray level quality through the following procedures:

1. Get the latest dithering pattern register value from SAMSUNG.
2. Display 16 gray bar in LCD.
3. Change the frame rate into an optimal value.
4. Change the VM alternating period to get the best quality.
5. As viewing 16 gray bars, select a good gray level, which is displayed well on your LCD.
6. Use only the good gray levels for quality.

**LCD Refresh Bus Bandwidth Calculation Guide**

The S3C2440X LCD controller can support various LCD display sizes. To select a suitable size (for the flicker free LCD system application), the user have to consider the LCD refresh bus bandwidth determined by the LCD display size, bit per pixel (bpp), frame rate, memory bus width, memory type, and so on.

LCD Data Rate (Byte/s) = bpp x (Horizontal display size) x (Vertical display size) x (Frame rate) /8

LCD DMA Burst Count (Times/s) = LCD Data Rate(Byte/s) /16(Byte) ; LCD DMA using 4words(16Byte) burst

Pdma means LCD DMA access period. In other words, the value of Pdma indicates the period of four-beat burst (4-words burst) for video data fetch. So, Pdma depends on memory type and memory setting.

Eventually, LCD System Load is determined by LCD DMA Burst Count and Pdma.

LCD System Load = LCD DMA Burst Count x Pdma

**Example 3:**

640 x 480, 8bpp, 60 frame/sec, 16-bit data bus width, SDRAM (Trp=2HCLK / Trcd=2HCLK / CL=2HCLK) and HCLK frequency is 60 MHz

LCD Data Rate = 8 x 640 x 480 x 60 / 8 = 18.432Mbyte/s
LCD DMA Burst Count = 18.432 / 16 = 1.152M/s
Pdma = (Trp+Trcd+CL+(2 x 4)+1) x (1/60MHz) = 0.250ms
LCD System Load = 1.152 x 250 = 0.288

System Bus Occupation Rate = (0.288/1) x 100 = 28.8%

**Register Setting Guide (TFT LCD)**

The CLKVAL register value determines the frequency of VCLK and frame rate.

Frame Rate = 1/ [ { (VSPW+1) + (VBPD+1) + (LIINEVAL + 1) + (VFPD+1) } x {(HSPW+1) + (HBPD +1) + (HFPD+1) + (HOZVAL + 1) } x { 2 x ( CLKVAL+1 ) / ( HCLK ) } ]

For applications, the system timing must be considered to avoid under-run condition of the fifo of the lcd controller caused by memory bandwidth contention.

**Example 4:**

TFT Resolution: 240 x 240,

VSPW =2, VBPD =14, LINEVAL = 239, VFPD =4

HSPW =25, HBPD =15, HOZVAL = 239, HFPD =1

CLKVAL = 5

HCLK = 60 M (hz)

The parameters below must be referenced by LCD size and driver specifications:

VSPW, VBPD, LINEVAL, VFPD, HSPW, HBPD, HOZVAL, and HFPD

If target frame rate is 60–70Hz, then CLKVAL should be 5.

So, Frame Rate = 67Hz

## NOTES

# 16 ADC & TOUCH SCREEN INTERFACE

## OVERVIEW

The 10-bit CMOS ADC(Analog to Digital Converter) is a recycling type device with 8-channel analog inputs. It converts the analog input signal into 10-bit binary digital codes at a maximum conversion rate of 500KSPS with 2.5MHz A/D converter clock. A/D converter operates with on-chip sample-and-hold function and power down mode is supported.

Touch Screen Interface is controlling and selecting pads(XP, XM, YP, YM) of the Touch Screen for X, Y position conversion. The Touch Screen Interface contains Touch Screen Pads control logic and ADC interface logic with an interrupt generation logic.

### FEATURES

- Resolution: 10-bit
- Differential Linearity Error: ± 1.0 LSB
- Integral Linearity Error: ± 2.0 LSB
- Maximum Conversion Rate: 500 KSPS
- Low Power Consumption
- Power Supply Voltage: 3.3V
- Analog Input Range: 0 ~ 3.3V
- On-chip sample-and-hold function
- Normal Conversion Mode
- Separate X/Y position conversion Mode
- Auto(Sequential) X/Y Position Conversion Mode
- Waiting for Interrupt Mode

## ADC & TOUCH SCREEN INTERFACE OPERATION

### BLOCK DIAGRAM

Figure 16-1 shows the functional block diagram of A/D converter and Touch Screen Interface. Note that the A/D converter device is a recycling type.

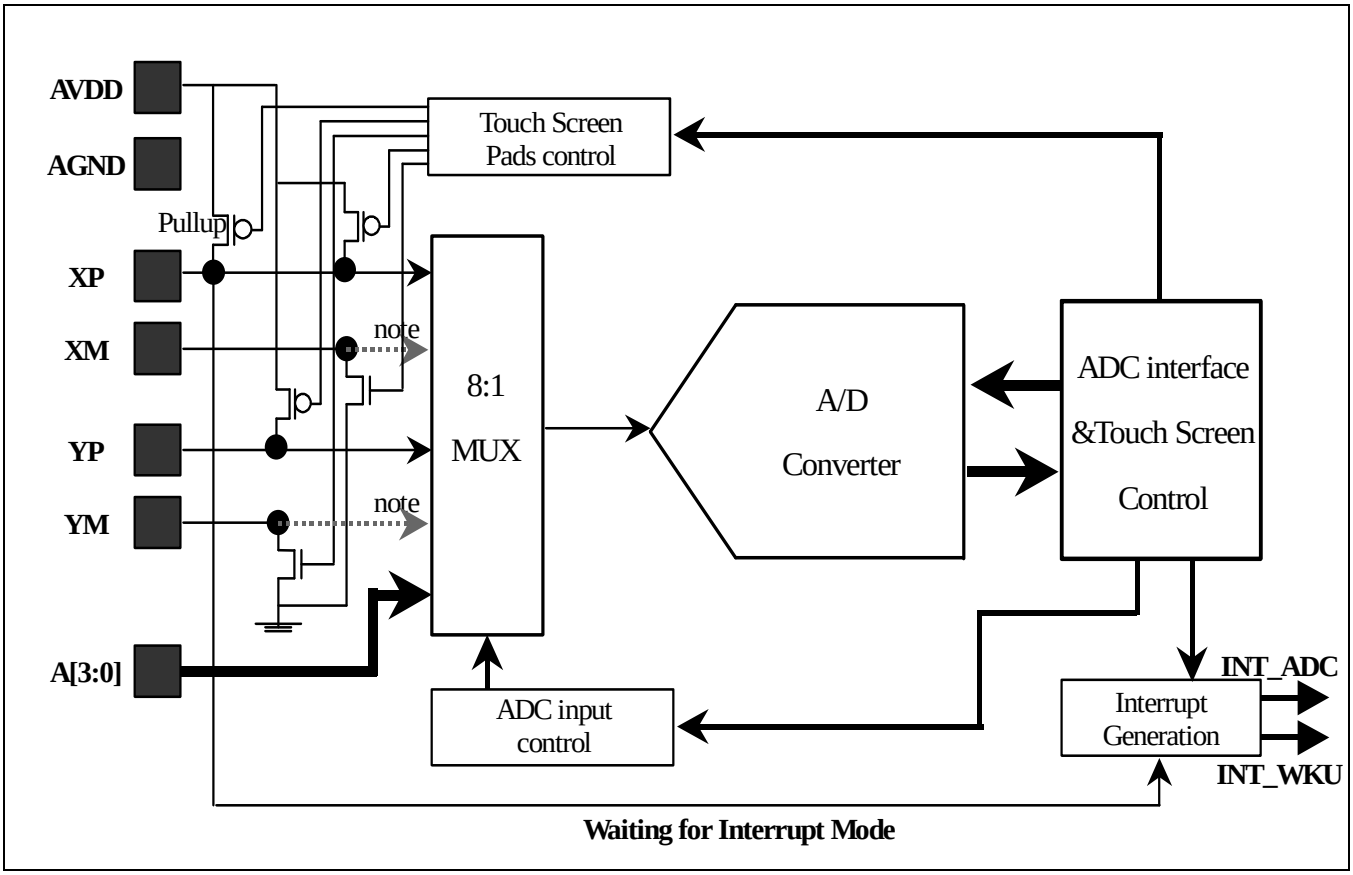



**Figure 16-1. ADC and Touch Screen Interface Functional Block Diagram**

*__note__ (symbol ····➤ )
When Touch Screen device is used, XM or PM is only connected ground for Touch Screen I/F.
When Touch Screen device is not used, XM or PM is connecting Analog Input Signal for Normal ADC conversion.

## FUNCTION DESCRIPTIONS

### A/D Conversion Time

When the GCLK frequency is 50MHz and the prescaler value is 49, total 10-bit conversion time is as follows.

A/D converter freq. = 50MHz/(49+1) = 1MHz

Conversion time = 1/(1MHz / 5cycles) = 1/200KHz = 5 us

**NOTE:**
This A/D converter was designed to operate at maximum 2.5MHz clock, so the conversion rate can go up to 500 KSPS.

### Touch Screen Interface Mode

1. Normal Conversion Mode

   Single Conversion Mode is the most likely used for General Purpose ADC Conversion. This mode can be initialized by setting the ADCCON (ADC Control Register) and completed with a read and a write to the ADCDAT0 (ADC Data Register 0).

2. Separate X/Y position conversion Mode

   Touch Screen Controller can be operated by one of two Conversion Modes. Separate X/Y Position Conversion Mode is operated as the following way. X-Position Mode writes X-Position Conversion Data to ADCDAT0, so Touch Screen Interface generates the Interrupt source to Interrupt Controller. Y-Position Mode writes Y-Position Conversion Data to ADCDAT1, so Touch Screen Interface generates the Interrupt source to Interrupt Controller.

3. Auto(Sequential) X/Y Position Conversion Mode

   Auto(Sequential) X/Y Position Conversion Mode is operated as the following. Touch Screen Controller sequentially converts X-Position and Y-Position that is touched. After Touch controller writes X-measurement data to ADCDAT0 and writes Y-measurement data to ADCDAT1, Touch Screen Interface is generating Interrupt source to Interrupt Controller in Auto Position Conversion Mode.

4. Waiting for Interrupt Mode

   Touch Screen Controller is generating wake-up (WKU) signal when the system is STOP mode (Power Down). The Waiting for Interrupt Mode of Touch Screen Controller must be set state of Pads(XP, XM, YP, YM) in Touch Screen Interface.

   After Touch Screen Controller is generating Wake-Up signal (INT_WKU), Waiting for interrupt Mode must be cleared. (XY_PST sets to the No operation Mode)

### Standby Mode

Standby mode is activated when ADCCON [2] is set to '1'. In this mode, A/D conversion operation is halted and ADCDAT0, ADCDAT1 register contains the previous converted data.

**Programming Notes**

1. The A/D converted data can be accessed by means of interrupt or polling method. With interrupt method the overall conversion time - from A/D converter start to converted data read - may be delayed because of the return time of interrupt service routine and data access time. With polling method, by checking the ADCCON[15] - end of conversion flag-bit, the read time from ADCDAT register can be determined.

2. Another way for starting A/D conversion is provided. After ADCCON[1] - A/D conversion start-by-read mode-is set to 1, A/D conversion starts simultaneously whenever converted data is read.

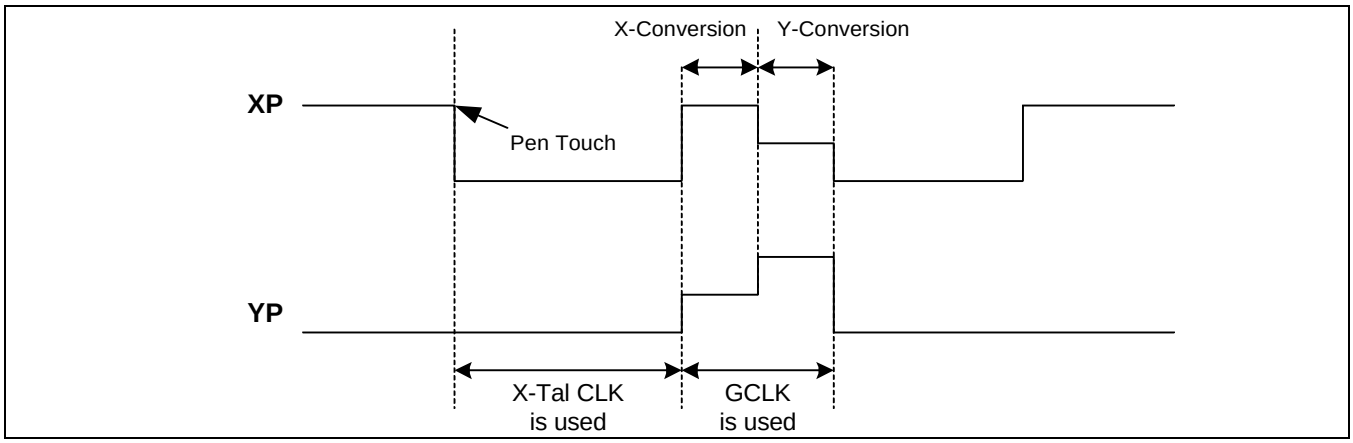



**Figure 16-2 ADC and Touch Screen Operation signal**

# ADC AND TOUCH SCREEN INTERFACE SPECIAL REGISTERS

### ADC CONTROL REGISTER (ADCCON)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ADCCON | 0x5800000 | R/W | ADC Control Register | 0x3FC4 |

| ADCCON | Bit | Description | Initial State |
|---|---|---|---|
| ECFLG | [15] | End of conversion flag(Read only)<br>0 = A/D conversion in process<br>1 = End of A/D conversion | 0 |
| PRSCEN | [14] | A/D converter prescaler enable<br>0 = Disable<br>1 = Enable | 0 |
| PRSCVL | [13:6] | A/D converter prescaler value<br>Data value: 0 ~ 255<br>Note that division factor is (N+1) when prescaler value is N. | 0xFF |
| SEL_MUX | [5:3] | Analog input channel select<br>000 = AIN 0<br>001 = AIN 1<br>010 = AIN 2<br>011 = AIN 3<br>100 = YM<br>101 = YP<br>110 = XM<br>111 = XP | 0 |
| STDBM | [2] | Standby mode select<br>0 = Normal operation mode<br>1 = Standby mode | 1 |
| READ_ START | [1] | A/D conversion start by read<br>0 = Disable start by read operation<br>1 = Enable start by read operation | 0 |
| ENABLE_START | [0] | A/D conversion starts by enable.<br>If READ_START is enabled, this value is not valid.<br>0 = No operation<br>1 = A/D conversion starts and this bit is cleared after the start-up. | 0 |

**NOTE** : When Touch Screen Pads(YM, YP, XM, XP) were disabled, these ports can be used as Analog input ports(AIN4, AIN5, AIN6, AIN7) for ADC.

### ADC TOUCH SCREEN CONTROL REGISTER (ADCTSC)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ADCTSC | 0x5800004 | R/W | ADC Touch Screen Control Register | 0x58 |

| ADCTSC | Bit | Description | Initial State |
|---|---|---|---|
| UD_SEN | [8] | Detect Stylus Up or Down status.<br>0 = Detect Stylus Down Interrupt Signal.<br>1 = Detect Stylus Up Interrupt Signal. | 0 |
| YM_SEN | [7] | YM Switch Enable<br>0 = YM Output Driver Disable.<br>1 = YM Output Driver Enable. | 0 |
| YP_SEN | [6] | YP Switch Enable<br>0 = YP Output Driver Enable.<br>1 = YP Output Driver Disable. | 1 |
| XM_SEN | [5] | XM Switch Enable<br>0 = XM Output Driver Disable.<br>1 = XM Output Driver Enable. | 0 |
| XP_SEN | [4] | XP Switch Enable<br>0 = XP Output Driver Enable.<br>1 = XP Output Driver Disable. | 1 |
| PULL_UP | [3] | Pull-up Switch Enable<br>0 = XP Pull-up Enable.<br>1 = XP Pull-up Disable. | 1 |
| AUTO_PST | [2] | Automatically sequencing conversion of X-Position and Y-Position<br>0 = Normal ADC conversion.<br>1 = Auto Sequential measurement of X-position, Y-position. | 0 |
| XY_PST | [1:0] | Manually measurement of X-Position or Y-Position.<br>00 = No operation mode<br>01 = X-position measurement<br>10 = Y-position measurement<br>11 = Waiting for Interrupt Mode | 0 |

**NOTE:** 1) While waiting for Touch screen Interrupt, XP_SEN bit should be set to '1', namely 'XP Output disable' and PULL_UP bit should be set to '0', namely 'XP Pull-up enable'.
2) AUTO_PST bit should be set '1' only in Automatic & Sequential X/Y Position conversion.

**Touch screen pin conditions in X/Y position conversion.**

| | XP | XM | YP | YM | ADC ch. select |
|---|---|---|---|---|---|
| **X** Position | Vref | GND | Hi-Z | Hi-Z | YP |
| **Y** Position | Hi-Z | Hi-Z | Vref | GND | XP |

### ADC START DELAY REGISTER (ADCDLY)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ADCDLY | 0x5800008 | R/W | ADC Start or Interval Delay Register | 0x00ff |

| ADCDLY | Bit | Description | Initial State |
|---|---|---|---|
| DELAY | [15:0] | 1) Normal Conversion Mode, XY Position Mode, Auto Position Mode.<br>→ ADC conversion start delay value.<br>2) Waiting for Interrupt Mode.<br>When Stylus Down occurs at SLEEP MODE, generates Wake-Up signal, having interval(several ms), for Exiting SLEEP MODE.<br>**Note) Don't use Zero value(0x0000)** | 00ff |

**NOTE:** Before ADC conversion, Touch screen uses X-tal clock(3.68MHz).
During ADC conversion GCLK( Max. 50MHz) is used.

### ADC CONVERSION DATA REGISTER (ADCDAT0)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ADCDAT0 | 0x580000C | R | ADC Conversion Data Register | - |

| ADCDAT0 | Bit | Description | Initial State |
|---|---|---|---|
| UPDOWN | [15] | Up or Down state of Stylus at Waiting for Interrupt Mode.<br>0 = Stylus down state.<br>1 = Stylus up state. | - |
| AUTO_PST | [14] | Automatic sequencing conversion of X-Position and Y-Position<br>0 = Normal ADC conversion.<br>1 = Sequencing measurement of X-position, Y-position. | - |
| XY_PST | [13:12] | Manually measurement of X-Position or Y-Position.<br>00 = No operation mode<br>01 = X-position measurement<br>10 = Y-position measurement<br>11 = Waiting for Interrupt Mode | - |
| Reserved | [11:10] | Reserved | |
| XPDATA<br>(Normal ADC) | [9:0] | X-Position Conversion data value<br>(include Normal ADC Conversion data value)<br>Data value : 0 ~ 3FF | - |

### ADC CONVERSION DATA REGISTER (ADCDAT1)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ADCDAT1 | 0x5800010 | R | ADC Conversion Data Register | - |

| ADCDAT1 | Bit | Description | Initial State |
|---|---|---|---|
| UPDOWN | [15] | Up or Down state of Stylus at Waiting for Interrupt Mode.<br>0 = Stylus down state.<br>1 = No stylus down state. | - |
| AUTO_PST | [14] | Automatically sequencing conversion of X-Position and Y-Position<br>0 = Normal ADC conversion.<br>1 = Sequencing measurement of X-position, Y-position. | - |
| XY_PST | [13:12] | Manually measurement of X-Position or Y-Position.<br>00 = No operation mode<br>01 = X-position measurement<br>10 = Y-position measurement<br>11 = Waiting for Interrupt Mode | - |
| Reserved | [11:10] | Reserved | |
| YPDATA | [9:0] | Y-Position Conversion data value<br>Data value : 0 ~ 3FF | - |

### ADC TOUCH SCREEN UP-DOWN REGISTER (ADCUPDN)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ADCUPDN | 0x5800014 | R/W | Stylus Up or Down Interrpt status register | 0x0 |

| ADCUPDN | Bit | Description | Initial State |
|---|---|---|---|
| TSC_UP | [1] | Stylus Up Interrupt.<br>0 = No stylus up status.<br>1 = Stylus up status. | 0 |
| TSC_DN | [0] | Stylus Down Interrupt.<br>0 = No stylus down status.<br>1 = Stylus down status. | 0 |

# 17 REAL TIME CLOCK

## OVERVIEW

The Real Time Clock (RTC) unit can be operated by the backup battery while the system power is off. The RTC can transmit 8-bit data to CPU as Binary Coded Decimal (BCD) values using the STRB/LDRB ARM operation. The data include the time by second, minute, hour, date, day, month, and year. The RTC unit works with an external 32.768 KHz crystal and also can perform the alarm function.

### FEATURES

— BCD number: second, minute, hour, date, day, month, and year

— Leap year generator

— Alarm function: alarm interrupt or wake-up from power-off mode

— Year 2000 problem is removed.

— Independent power pin (RTCVDD)

— Supports millisecond tick time interrupt for RTOS kernel time tick.

— Round reset function

## REAL TIME CLOCK OPERATION

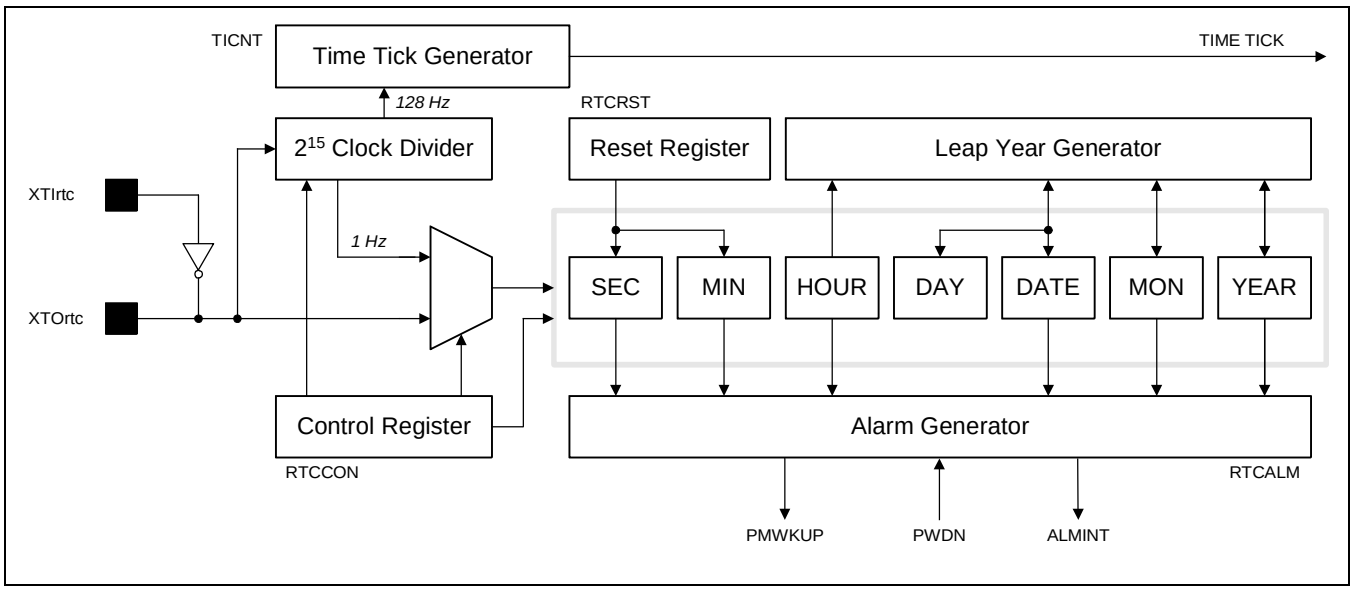



**Figure 17-1. Real Time Clock Block Diagram**

### LEAP YEAR GENERATOR

The leap year generator can determine the last date of each month out of 28, 29, 30, or 31, based on data from BCDDATE, BCDMON, and BCDYEAR. This block considers leap year in deciding on the last date. An 8-bit counter can only represent 2 BCD digits, so it cannot decide whether "00" year (the year with its last two digits zeros) is a leap year or not. For example, it cannot discriminate between 1900 and 2000. To solve this problem, the RTC block in S3C2440X has hard-wired logic to support the leap year in 2000. Note 1900 is not leap year while 2000 is leap year. Therefore, two digits of 00 in S3C2440X denote 2000, not 1900.

### READ/WRITE REGISTERS

Bit 0 of the RTCCON register must be set high in order to write the BCD register in RTC block. To display the second, minute, hour, date, month, and year, the CPU should read the data in BCDSEC, BCDMIN, BCDHOUR, BCDDAY, BCDDATE, BCDMON, and BCDYEAR registers, respectively, in the RTC block. However, a one second deviation may exist because multiple registers are read. For example, when the user reads the registers from BCDYEAR to BCDMIN, the result is assumed to be 2059 (Year), 12 (Month), 31 (Date), 23 (Hour) and 59 (Minute). When the user read the BCDSEC register and the value ranges from 1 to 59 (Second), there is no problem, but, if the value is 0 sec., the year, month, date, hour, and minute may be changed to 2060 (Year), 1 (Month), 1 (Date), 0 (Hour) and 0 (Minute) because of the one second deviation that was mentioned. In this case, the user should re-read from BCDYEAR to BCDSEC if BCDSEC is zero.

### BACKUP BATTERY OPERATION

The RTC logic can be driven by the backup battery, which supplies the power through the RTCVDD pin into the RTC block, even if the system power is off. When the system is off, the interfaces of the CPU and RTC logic should be blocked, and the backup battery only drives the oscillation circuit and the BCD counters to minimize power dissipation.

### ALARM FUNCTION

The RTC generates an alarm signal at a specified time in the power-off mode or normal operation mode. In normal operation mode, the alarm interrupt (ALMINT) is activated. In the power-off mode, the power management wakeup (PMWKUP) signal is activated as well as the ALMINT. The RTC alarm register (RTCALM) determines the alarm enable/disable status and the condition of the alarm time setting.

### TICK TIME INTERRUPT

The RTC tick time is used for interrupt request. The TICNT register has an interrupt enable bit and the count value for the interrupt. The count value reaches '0' when the tick time interrupt occurs. Then the period of interrupt is as follows:

Period = ( n+1 ) / 128 second
n: Tick time count value (1~127)

This RTC time tick may be used for real time operating system (RTOS) kernel time tick. If time tick is generated by the RTC time tick, the time related function of RTOS will always synchronized in real time.

### ROUND RESET FUNCTION

The round reset function can be performed by the RTC round reset register (RTCRST). The round boundary (30, 40, or 50 sec.) of the second carry generation can be selected, and the second value is rounded to zero in the round reset. For example, when the current time is 23:37:47 and the round boundary is selected to 40 sec, the round reset changes the current time to 23:38:00.

**NOTE**

All RTC registers have to be accessed for each byte unit using the STRB and LDRB instructions or char type pointer.

### 32.768KHZ X-TAL CONNECTION EXAMPLE

The Figure 17-2 shows a circuit of the RTC unit oscillation at 32.768Khz.

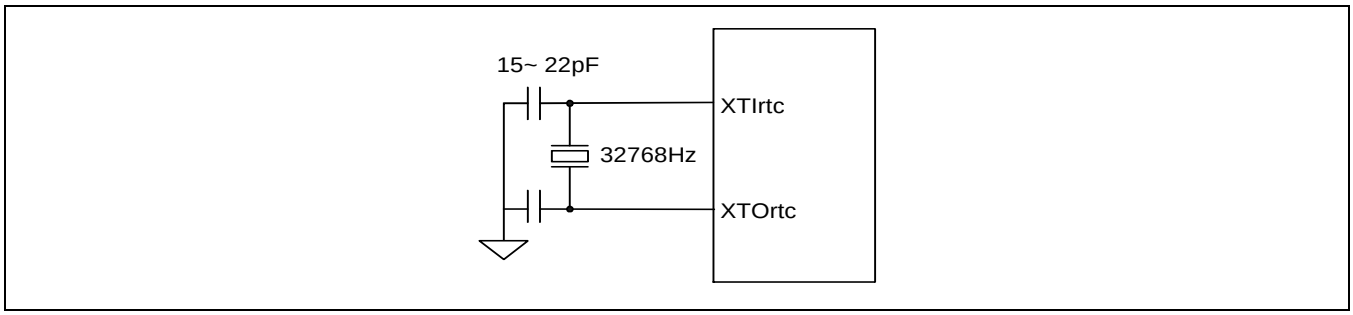



**Figure 17-2. Main Oscillator Circuit Example**

# REAL TIME CLOCK SPECIAL REGISTERS

## REAL TIME CLOCK CONTROL (RTCCON) REGISTER

The RTCCON register consists of 4 bits such as the RTCEN, which controls the read/write enable of the BCD registers, CLKSEL, CNTSEL, and CLKRST for testing.

RTCEN bit can control all interfaces between the CPU and the RTC, so it should be set to 1 in an RTC control routine to enable data read/write after a system reset. Also before power off, the RTCEN bit should be cleared to 0 to prevent inadvertent writing into RTC registers.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RTCCON | 0x57000040(L)<br>0x57000043(B) | R/W<br>(by byte) | RTC control register | 0x0 |

| RTCCON | Bit | Description | Initial State |
|---|---|---|---|
| CLKRST | [3] | RTC clock count reset.<br>0 = No reset, 1 = Reset | 0 |
| CNTSEL | [2] | BCD count select.<br>0 = Merge BCD counters<br>1 = Reserved (Separate BCD counters) | 0 |
| CLKSEL | [1] | BCD clock select.<br>0 = XTAL $1/2^{15}$ divided clock<br>1 = Reserved (XTAL clock only for test) | 0 |
| RTCEN | [0] | RTC control enable.<br>0 = Disable 1 = Enable<br>Note: Only BCD time count and read operation can be performed. | 0 |

**Notes:**

1. All RTC registers have to be accessed for each byte unit using STRB and LDRB instructions or char type pointer.
2. (L): Little endian.
   (B): Big endian.

## TICK TIME COUNT (TICNT) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| TICNT | 0x57000044(L)<br>0x57000047(B) | R/W<br>(by byte) | Tick time count register | 0x0 |

| TICNT | Bit | Description | Initial State |
|---|---|---|---|
| TICK INT ENABLE | [7] | Tick time interrupt enable.<br>0 = Disable 1 = Enable | 0 |
| TICK TIME COUNT | [6:0] | Tick time count value (1~127).<br>This counter value decreases internally, and users cannot read this counter value in working. | 000000 |

### RTC ALARM CONTROL (RTCALM) REGISTER

The RTCALM register determines the alarm enable and the alarm time. Note that the RTCALM register generates the alarm signal through both ALMINT and PMWKUP in power down mode, but only through ALMINT in the normal operation mode.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RTCALM | 0x57000050(L)<br>0x57000053(B) | R/W<br>(by byte) | RTC alarm control register | 0x0 |

| RTCALM | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7] | | 0 |
| ALMEN | [6] | Alarm global enable.<br>0 = Disable, 1 = Enable | 0 |
| YEAREN | [5] | Year alarm enable.<br>0 = Disable, 1 = Enable | 0 |
| MONREN | [4] | Month alarm enable.<br>0 = Disable, 1 = Enable | 0 |
| DATEEN | [3] | Date alarm enable.<br>0 = Disable, 1 = Enable | 0 |
| HOUREN | [2] | Hour alarm enable.<br>0 = Disable, 1 = Enable | 0 |
| MINEN | [1] | Minute alarm enable.<br>0 = Disable, 1 = Enable | 0 |
| SECEN | [0] | Second alarm enable.<br>0 = Disable, 1 = Enable | 0 |

### ALARM SECOND DATA (ALMSEC) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ALMSEC | 0x57000054(L)<br>0x57000057(B) | R/W<br>(by byte) | Alarm second data register | 0x0 |

| ALMSEC | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7] | | 0 |
| SECDATA | [6:4] | BCD value for alarm second.<br>0 ~ 5 | 000 |
| | [3:0] | 0 ~ 9 | 0000 |

### ALARM MIN DATA (ALMMIN) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ALMMIN | 0x57000058(L)<br>0x5700005B(B) | R/W<br>(by byte) | Alarm minute data register | 0x00 |

| ALMMIN | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7] | | 0 |
| MINDATA | [6:4] | BCD value for alarm minute.<br>0 ~ 5 | 000 |
| | [3:0] | 0 ~ 9 | 0000 |

### ALARM HOUR DATA (ALMHOUR) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ALMHOUR | 0x5700005C(L)<br>0x5700005F(B) | R/W<br>(by byte) | Alarm hour data register | 0x0 |

| ALMHOUR | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:6] | | 00 |
| HOURDATA | [5:4] | BCD value for alarm hour.<br>0 ~ 2 | 00 |
| | [3:0] | 0 ~ 9 | 0000 |

### ALARM DATE DATA (ALMDATE) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ALMDATE | 0x57000060(L)<br>0x57000063(B) | R/W<br>(by byte) | Alarm date data register | 0x01 |

| ALMDATE | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:6] | | 00 |
| DATEDATA | [5:4] | BCD value for alarm date, from 0 to 28, 29, 30, 31.<br>0 ~ 3 | 00 |
| | [3:0] | 0 ~ 9 | 0001 |

### ALARM MON DATA (ALMMON) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ALMMON | 0x57000064(L)<br>0x57000067(B) | R/W<br>(by byte) | Alarm month data register | 0x01 |

| ALMMON | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:5] | | 00 |
| MONDATA | [4] | BCD value for alarm month.<br>0 ~ 1 | 0 |
| | [3:0] | 0 ~ 9 | 0001 |

### ALARM YEAR DATA (ALMYEAR) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ALMYEAR | 0x57000068(L)<br>0x5700006B(B) | R/W<br>(by byte) | Alarm year data register | 0x0 |

| ALMYEAR | Bit | Description | Initial State |
|---|---|---|---|
| YEARDATA | [7:0] | BCD value for year.<br>00 ~ 99 | 0x0 |

### RTC ROUND RESET (RTCRST) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RTCRST | 0x5700006C(L)<br>0x5700006F(B) | R/W<br>(by byte) | RTC round reset register | 0x0 |

| RTCRST | Bit | Description | Initial State |
|---|---|---|---|
| SRSTEN | [3] | Round second reset enable.<br>0 = Disable, 1 = Enable | 0 |
| SECCR | [2:0] | Round boundary for second carry generation.<br>011 = over than 30 sec<br>100 = over than 40 sec<br>101 = over than 50 sec<br>**Note:** If other values (0, 1, 2, 6, or 7) are set, no second carry is generated. But second value can be reset. | 000 |

### BCD SECOND (BCDSEC) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCDSEC | 0x57000070(L)<br>0x57000073(B) | R/W<br>(by byte) | BCD second register | Undefined |

| BCDSEC | Bit | Description | Initial State |
|---|---|---|---|
| SECDATA | [6:4] | BCD value for second.<br>0 ~ 5 | - |
| | [3:0] | 0 ~ 9 | - |

### BCD MINUTE (BCDMIN) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCDMIN | 0x57000074(L)<br>0x57000077(B) | R/W<br>(by byte) | BCD minute register | Undefined |

| BCDMIN | Bit | Description | Initial State |
|---|---|---|---|
| MINDATA | [6:4] | BCD value for minute.<br>0 ~ 5 | - |
| | [3:0] | 0 ~ 9 | - |

### BCD HOUR (BCDHOUR) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCDHOUR | 0x57000078(L)<br>0x5700007B(B) | R/W<br>(by byte) | BCD hour register | Undefined |

| BCDHOUR | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:6] | | - |
| HOURDATA | [5:4] | BCD value for hour.<br>0 ~ 2 | - |
| | [3:0] | 0 ~ 9 | - |

### BCD DATE (BCDDATE) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCDDATE | 0x5700007C(L)<br>0x5700007F(B) | R/W<br>(by byte) | BCD date register | Undefined |

| BCDDATE | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:6] | | - |
| DATEDATA | [5:4] | BCD value for date.<br>0 ~ 3 | - |
| | [3:0] | 0 ~ 9 | - |

### BCD DAY (BCDDAY) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCDDAY | 0x57000080(L)<br>0x57000083(B) | R/W<br>(by byte) | BCD a day of the week register | Undefined |

| BCDDAY | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:3] | | - |
| DAYDATA | [2:0] | BCD value for a day of the week.<br>1 ~ 7 | - |

### BCD MONTH (BCDMON) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCDMON | 0x57000084(L)<br>0x57000087(B) | R/W<br>(by byte) | BCD month register | Undefined |

| BCDMON | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:5] | | - |
| MONDATA | [4] | BCD value for month.<br>0 ~ 1 | - |
| | [3:0] | 0 ~ 9 | - |

### BCD YEAR (BCDYEAR) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCDYEAR | 0x57000088(L)<br>0x5700008B(B) | R/W<br>(by byte) | BCD year register | Undefined |

| BCDYEAR | Bit | Description | Initial State |
|---|---|---|---|
| YEARDATA | [7:0] | BCD value for year.<br>00 ~ 99 | - |

# 18 WATCHDOG TIMER

## OVERVIEW

The S3C2440X watchdog timer is used to resume the controller operation whenever it is disturbed by malfunctions such as noise and system errors. It can be used as a normal 16-bit interval timer to request interrupt service. The watchdog timer generates the reset signal for 128 PCLK cycles.

### FEATURES

— Normal interval timer mode with interrupt request

— Internal reset signal is activated for 128 PCLK cycles when the timer count value reaches 0 (time-out).

## WATCHDOG TIMER OPERATION

Figure 18-1 shows the functional block diagram of the watchdog timer. The watchdog timer uses only PCLK as its source clock. The PCLK frequency is prescaled to generate the corresponding watchdog timer clock, and the resulting frequency is divided again.

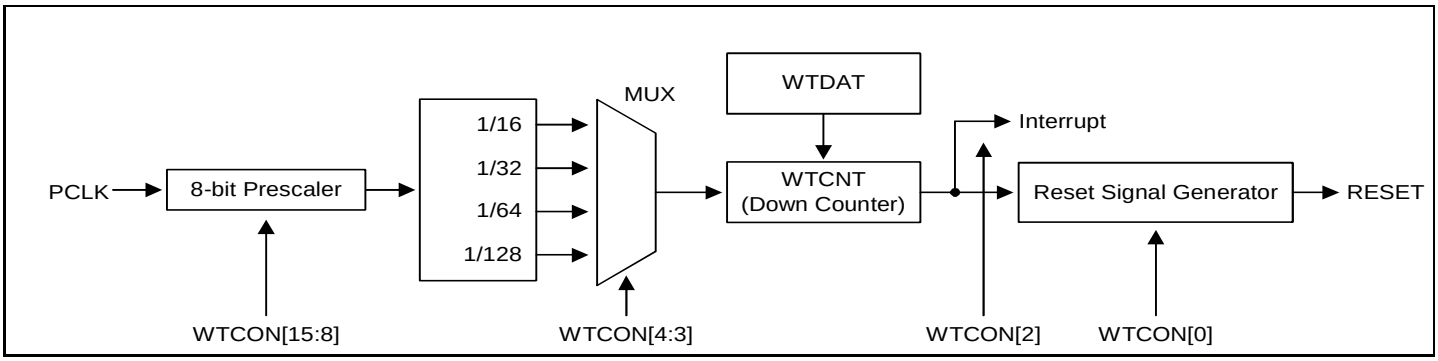



**Figure 18-1. Watchdog Timer Block Diagram**

The prescaler value and the frequency division factor are specified in the watchdog timer control (WTCON) register. Valid prescaler values range from 0 to $2^8$-1. The frequency division factor can be selected as 16, 32, 64, or 128.

Use the following equation to calculate the watchdog timer clock frequency and the duration of each timer clock cycle:

t_watchdog = 1/[ PCLK / (Prescaler value + 1) / Division_factor ]

### WTDAT & WTCNT

Once the watchdog timer is enabled, the value of watchdog timer data (WTDAT) register cannot be automatically reloaded into the timer counter (WTCNT). In this reason, an initial value must be written to the watchdog timer count (WTCNT) register, before the watchdog timer starts.

## CONSIDERATION OF DEBUGGING ENVIRONMENT

When the S3C2440X is in debug mode using Embedded ICE, the watchdog timer must not operate.

The watchdog timer can determine whether or not it is currently in the debug mode from the CPU core signal (DBGACK signal). Once the DBGACK signal is asserted, the reset output of the watchdog timer is not activated as the watchdog timer is expired.

# WATCHDOG TIMER SPECIAL REGISTERS

## WATCHDOG TIMER CONTROL (WTCON) REGISTER

The WTCON register allows the user to enable/disable the watchdog timer, select the clock signal from 4 different sources, enable/disable interrupts, and enable/disable the watchdog timer output.
The Watchdog timer is used to resume the S3C2440X restart on mal-function after its power on; if controller restart is not desired, the Watchdog timer should be disabled.

If the user wants to use the normal timer provided by the Watchdog timer, enable the interrupt and disable the Watchdog timer.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| WTCON | 0x53000000 | R/W | Watchdog timer control register | 0x8021 |

| WTCON | Bit | Description | Initial State |
|---|---|---|---|
| Prescaler value | [15:8] | Prescaler value.<br>The valid range is from 0 to 255($2^8$-1). | 0x80 |
| Reserved | [7:6] | Reserved.<br>These two bits must be 00 in normal operation. | 00 |
| Watchdog timer | [5] | Enable or disable bit of Watchdog timer.<br>0 = Disable<br>1 = Enable | 1 |
| Clock select | [4:3] | Determine the clock division factor.<br>00: 16 01 : 32<br>10: 64 11 : 128 | 00 |
| Interrupt generation | [2] | Enable or disable bit of the interrupt.<br>0 = Disable<br>1 = Enable | 0 |
| Reserved | [1] | Reserved.<br>This bit must be 0 in normal operation. | 0 |
| Reset enable/disable | [0] | Enable or disable bit of Watchdog timer output for reset signal.<br>1: Assert reset signal of the S3C2440X at watchdog time-out<br>0: Disable the reset function of the watchdog timer. | 1 |

### WATCHDOG TIMER DATA (WTDAT) REGISTER

The WTDAT register is used to specify the time-out duration. The content of WTDAT cannot be automatically loaded into the timer counter at initial watchdog timer operation. However, using 0x8000 (initial value) will drive the first time-out. In this case, the value of WTDAT will be automatically reloaded into WTCNT.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| WTDAT | 0x53000004 | R/W | Watchdog timer data register | 0x8000 |

| WTDAT | Bit | Description | Initial State |
|---|---|---|---|
| Count reload value | [15:0] | Watchdog timer count value for reload. | 0x8000 |

### WATCHDOG TIMER COUNT (WTCNT) REGISTER

The WTCNT register contains the current count values for the watchdog timer during normal operation. Note that the content of the WTDAT register cannot be automatically loaded into the timer count register when the watchdog timer is enabled initially, so the WTCNT register must be set to an initial value before enabling it.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| WTCNT | 0x53000008 | R/W | Watchdog timer count register | 0x8000 |

| WTCNT | Bit | Description | Initial State |
|---|---|---|---|
| Count value | [15:0] | The current count value of the watchdog timer | 0x8000 |

# 19 SDI

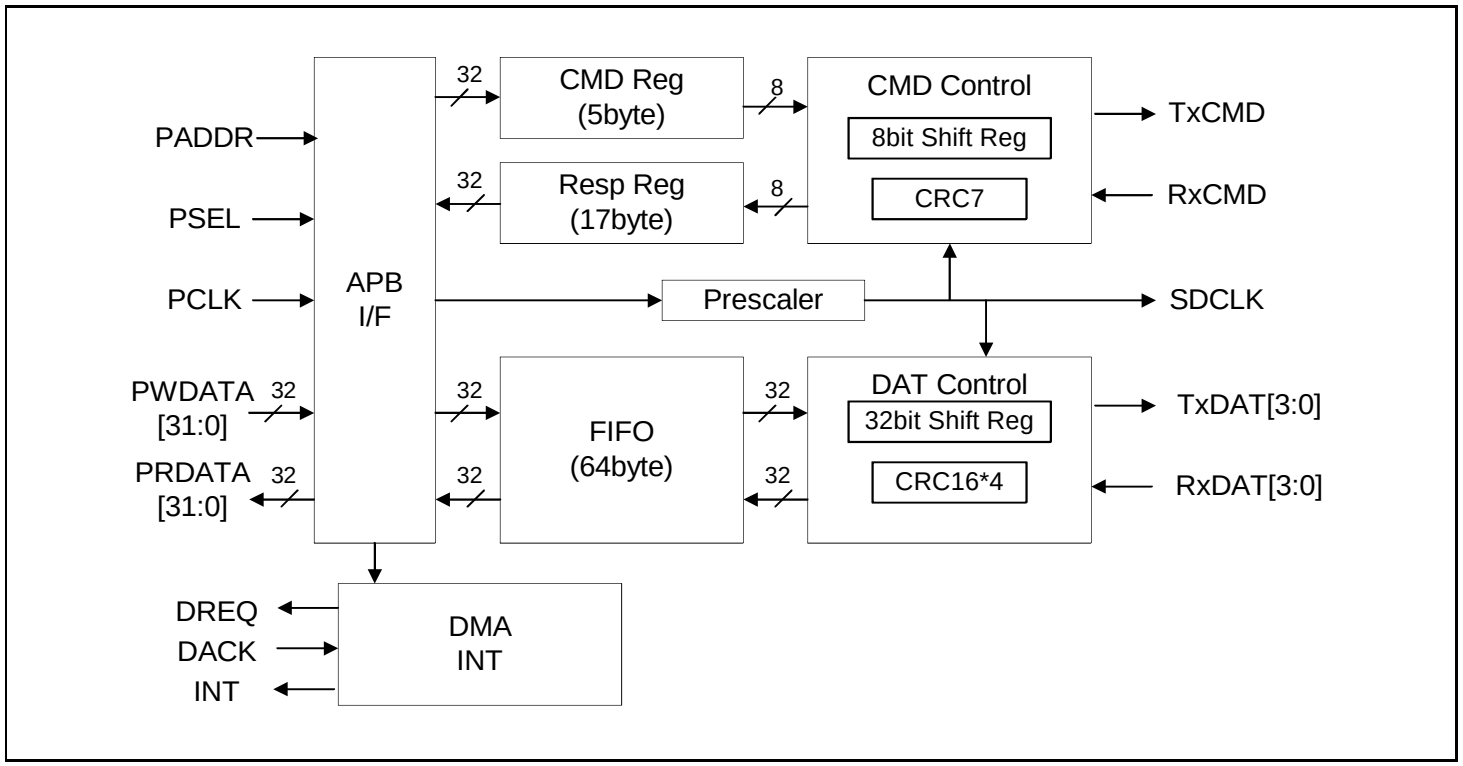
## OVERVIEW

The S3C2440X SDI(Secure Digital Interface) can interface for SD memory card, SDIO device and MMC(Multi-Media Card).

### FEATURES

— SD Memory Card Spec(ver 1.0) / MMC Spec(2.11) compatible
— SDIO Card Spec(Ver 1.0) compatible
— 16 words(64 bytes) FIFO(depth 16) for data Tx/Rx
— 40-bit Command Register
— 136-bit Response Register
— 8-bit Prescaler logic(Freq = System Clock / (P + 1))
— CRC7 & CRC16 generator
— Normal, and DMA data transfer mode(byte or word transfer)
— 1bit / 4bit(wide bus) mode & block / stream mode switch support

## BLOCK DIAGRAM



## SD OPERATION

A serial clock line synchronizes shifting and sampling of the information on the five data lines. The transmission frequency is controlled by making the appropriate bit settings to the SDIPRE register. You can modify its frequency to adjust the baud rate data register value.

Programming Procedure (common)

To program the SDI modules, follow these basic steps :

1. Set SDICON to configure properly with clock & interrupt enable
2. Set SDIPRE to configure with a proper value.
3. Wait 74 SDCLK clock cycle in order to initialize the card.

CMD Path Programming

1. Write command argument 32bit to SDICARG.
2. Determine command types and start command transmit with setting SDICCON.
3. Confirm the end of SDI CMD path operation when the specific flag of SDICSTA is set
4. The flag is CmdSent if command type is no response.
5. The flag is RspFin if command type is with response.
6. Clear the corresponding flag of SDICSTA through writing one with this bit

DAT Path Programming

1. Write data timeout period to SDIDTimer.
2. Write block size(block length) to SDIBSIZE(normally 0x80 word).
3. Determine the mode of block, wide bus, dma, etc and start data transfer with setting SDIDCON.
4. Tx data → Write data to Data Register(SDIDAT) while Tx FIFO is available(TFDET is set), or half(TFHalf is set), or empty(TFEmpty is set).
5. Rx data → Read data from Data Register(SDIDAT) while Rx FIFO is available(RFDET is set), or full(RFFull is set), or half(RFHalf is set), or ready for last data(RFLast is set).
6. Confirm the end of SDI DAT path operation when DatFin flag of SDIDSTA is set
7. Clear the corresponding flag of SDIDSTA through writing one with this bit

## SDIO OPERATION

There are two functions of SDIO operation: SDIO Interrupt receiving and Read Wait Request generation. These two functions can operate when RcvIOInt bit and RwaitEn bit of SDICON register is activated respectively. And two functions have the steps and conditions below:-

SDIO Interrupt

In SD 1bit mode, Interrupt is received through all range from RxDAT[1] pin.

In SD 4bit mode, RxDAT[1] pin is shared between data receiving and interrupt receiving.
When interrupt detection range(Interrupt Period) is :

1. Single Block : the time between A and B
   - A : 2clocks after the completion of a data packet
   - B : The completion of sending the end bit of the next with data command
2. Multi Block, PrdType = 0 : the time between A and B, restart at C
   - A : 2clocks after the completion of a data packet
   - B : 2clocks after A
   - C : 2clocks after the end bit of the abort command response
3. Multi Block, PrdType = 1 : the time between A and B, restart at A
   - A : 2clocks after the completion of a data packet
   - B : 2clocks after A
   - In case of last block, interrupt period begins at A, but not ends at B(CMD53 case)

Read Wait Request

Regardless of 1bit or 4bit mode, Read Wait Request signal transmits to TxDAT[2] pin in condition of below.

- In read multiple operation, request signal transmission begins at 2clocks after the end of the data block
- Transmission ends when user sets to one RwaitReq bit of SDIDSTA register

## SDI SPECIAL REGISTERS

### SDI Control Register(SDICON)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDICON | 0x5A000000 | R/W | SDI Control Register | 0x0 |

| SDICON | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:6] | | |
| Clock Type (CTYP) | [5] | Determines which clock type is used as SDCLK.<br>0 = SD type, 1 = MMC type | 0 |
| Byte Order Type(ByteOrder) | [4] | Determines byte order type when you read(write) data from(to) sd host FIFO with word boundary.<br>0 = Type A, 1 = Type B | 0 |
| Receive SDIO Interrupt from card (RcvIOInt) | [3] | Determines whether sd host receives SDIO Interrupt from the card or not(for SDIO).<br>0 = ignore, 1 = receive SDIO Interrupt | 0 |
| Read Wait Enable(RWaitEn) | [2] | Determines read wait request signal generate when sd host waits the next block in multiple block read mode. This bit needs to delay the next block to be transmitted from the card(for SDIO).<br>0 = disable(no generate), 1 = Read wait enable(use SDIO) | 0 |
| FIFO Reset(FRST) | [1] | Reset FIFO value. This bit is automatically clear.<br>0 = normal mode, 1 = FIFO reset | 0 |
| Clock Out Enable (ENCLK) | [0] | Determines whether SDCLK Out enable or not<br>0 = disable(prescaler off), 1 = clock enable | 0 |

* Byte Order Type

- Type A : D[7:0] → D[15:8] → D[23:16] → D[31:24]
- Type B : D[31:24] → D[23:16] → D[15:8] → D[7:0]

### SDI Baud Rate Prescaler Register(SDIPRE)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIPRE | 0x5A000004 | R/W | SDI Buad Rate Prescaler Register | 0x01 |

| SDIPRE | Bit | Description | Initial Value |
|---|---|---|---|
| Prescaler Value | [7:0] | Determines SDI clock (SDCLK) rate as above equation.<br>Baud rate = PCLK / (Prescaler value + 1) | 0x01 |

* Prescaler Value should be greater than zero.

### SDI Command Argument Register (SDICARG)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDICARG | 0x5A000008 | R/W | SDI Command Argument Register | 0x0 |

| SDICARG | Bit | Description | Initial Value |
|---|---|---|---|
| CmdArg | [31:0] | Command Argument | 0x00000000 |

### SDI Command Control Register (SDICCON)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDICCON | 0x5A00000C | R/W | SDI Command Control Register | 0x0 |

| SDICCON | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:13] | | |
| Abort Command (AbortCmd) | [12] | Determines whether command type is for abort(for SDIO).<br>0 = normal command, 1 = abort command(CMD12, CMD52) | 0 |
| Command with Data (WithData) | [11] | Determines whether command type is with data(for SDIO).<br>0 = without data, 1 = with data | 0 |
| LongRsp | [10] | Determines whether host receives a 136-bit long response or not<br>0 = short response, 1 = long response | 0 |
| WaitRsp | [9] | Determines whether host waits for a response or not<br>0 = no response, 1 = wait response | 0 |
| Command Start(CMST) | [8] | Determines whether command operation starts or not<br>0 = command ready, 1 = command start | 0 |
| CmdIndex | [7:0] | Command index with start 2bit(8bit) | 0x00 |

### SDI Command Status Register (SDICSTA)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDICSTA | 0x5A000010 | R/(C) | SDI Command Status Register | 0x0 |

| SDICSTA | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:13] | | |
| Response CRC Fail(RspCrc) | [12] R/C | CRC check failed when command response received. This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = crc fail | 0 |
| Command Sent (CmdSent) | [11] R/C | Command sent(not concerned with response). This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = command end | 0 |
| Command Time Out (CmdTout) | [10] R/C | Command response timeout(64clk). This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = timeout | 0 |
| Response Receive End (RspFin) | [9] R/C | Command response received. This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = response end | 0 |
| CMD line progress On (CmdOn) | [8] | Command transfer in progress<br>0 = not detect, 1 = in progress | 0 |
| RspIndex | [7:0] | Response index 6bit with start 2bit(8bit) | 0x00 |

### SDI Response Register 0(SDIRSP0)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIRSP0 | 0x5A000014 | R | SDI Response Register 0 | 0x0 |

| SDIRSP0 | Bit | Description | Initial Value |
|---|---|---|---|
| Response0 | [31:0] | Card status[31:0](short), card status[127:96](long) | 0x00000000 |

### SDI Response Register 1(SDIRSP1)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIRSP1 | 0x5A000018 | R | SDI Response Register 1 | 0x0 |

| SDIRSP1 | Bit | Description | Initial Value |
|---|---|---|---|
| RCRC7 | [31:24] | CRC7(with end bit, short), card status[95:88](long) | 0x00 |
| Response1 | [23:0] | unused(short), card status[87:64](long) | 0x000000 |

### SDI Response Register 2(SDIRSP2)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIRSP2 | 0x5A00001c | R | SDI Response Register 2 | 0x0 |

| SDIRSP2 | Bit | Description | Initial Value |
|---|---|---|---|
| Response2 | [31:0] | unused(short), card status[63:32](long) | 0x00000000 |

### SDI Response Register 3(SDIRSP3)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIRSP3 | 0x5A000020 | R | SDI Response Register 3 | 0x0 |

| SDIRSP3 | Bit | Description | Initial Value |
|---|---|---|---|
| Response3 | [31:0] | unused(short), card status[31:0](long) | 0x00000000 |

### SDI Data / Busy Timer Register(SDIDTimer)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIDTimer | 0x5A000024 | R/W | SDI Data / Busy Timer Register | 0x0 |

| SDIDTimer | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:23] | | |
| DataTimer | [22:0] | Data / Busy timeout period | 0x10000 |

### SDI Block Size Register(SDIBSIZE)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIBSIZE | 0x5A000028 | R/W | SDI Block Size Register | 0x0 |

| SDIBSIZE | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:12] | | |
| BlkSize | [11:0] | Block Size value(0~4095 byte) , don't care when stream mode | 0x000 |

* In Case of multi block, BlkSize must be aligned to word(4byte) size.(BlkSize[1:0] = 00)

### SDI Data Control Register(SDIDCON)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIDCON | 0x5A00002c | R/W | SDI Data control Register | 0x0 |

| SDIDCON | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:22] | | |
| SDIO Interrupt Period Type (PrdType) | [21] | Determines whether SDIO Interrupt period is 2 cycle or extend more cycle when last data block is transferred(for SDIO).<br>0 = exactly 2 cycle, 1 = more cycle(likely single block) | 0 |
| Transmit After Response (TARSP) | [20] | Determines when data transmit start after response receive or not<br>0 = directly after DatMode set,<br>1 = after response receive(assume DatMode sets to 2'b11) | 0 |
| Receive After Command (RACMD) | [19] | Determines when data receive start after command sent or not<br>0 = directly after DatMode set,<br>1 = after command sent (assume DatMode sets to 2'b10) | 0 |
| Busy After Command (BACMD) | [18] | Determines when busy receive start after command sent or not<br>0 = directly after DatMode set,<br>1 = after command sent (assume DatMode sets to 2'b01) | 0 |
| Block mode (BlkMode) | [17] | Data transfer mode<br>0 = stream data transfer, 1 = block data transfer | 0 |
| Wide bus enable (WideBus) | [16] | Determines enable wide bus mode<br>0 = standard bus mode(only SDIDAT[0] used),<br>1 = wide bus mode(SDIDAT[3:0] used) | 0 |
| DMA Enable (EnDMA) | [15] | Enable DMA<br>0 = disable(polling), 1 = dma enable | 0 |
| Stop by force (STOP) | [14] | Determines whether data transfer stop by force or not<br>0 = normal, 1 = stop by force | 0 |
| Data Transfer Mode (DatMode) | [13:12] | Determines which direction of data transfer<br>00 = ready, 01 = only busy check start<br>10 = data receive start, 11 = data transmit start | 00 |
| BlkNum | [11:0] | Block Number(0~4095), don't care when stream mode | 0x000 |

**NOTES** :

1. If you want one of TARSP, RACMD, BACMD bits(SDIDCON[20:18]) to "1", you need to write on SDIDCON register ahead of on SDICCON register.(always need for SDIO)

2. When DMA operation is completed, DMA Enable[15] bit of SDIDCON register should be disabled.

### SDI Data Remain Counter Register(SDIDCNT)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIDCNT | 0x5A000030 | R | SDI Data Remain Counter Register | 0x0 |

| SDIDCNT | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:24] | | |
| BlkNumCnt | [23:12] | Remaining Block number | 0x000 |
| BlkCnt | [11:0] | Remaining data byte of 1 block | 0x000 |

### SDI Data Status Register(SDIDSTA)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIDSTA | 0x5A000034 | R/(C) | SDI Data Status Register | 0x0 |

| SDIDSTA | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:11] | | |
| Read Wait Request Occur (RWaitReq) | [10] R/C | Read wait request signal transmits to sd card. The request signal is stopped and this flag is cleared by setting to one this bit.<br>0 = not occur, 1 = Read wait request occur | 0 |
| SDIO Interrupt Detect(IOIntDet) | [9] R/C | SDIO interrupt detect. This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = SDIO interrupt detect | 0 |
| FIFO Fail error (FFfail) | [8] R/C | FIFO fail error when FIFO occurs overrun / underrun / misaligned data saving. This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = FIFO fail | 0 |
| CRC Status Fail(CrcSta) | [7] R/C | CRC Status error when data block sent(CRC check failed). This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = crc status fail | 0 |
| Data Receive CRC Fail(DatCrc) | [6] R/C | Data block received error(CRC check failed). This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = receive crc fail | 0 |
| Data Time Out(DatTout) | [5] R/C | Data / Busy receive timeout. This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = timeout | 0 |
| Data Transfer Finish(DatFin) | [4] R/C | Data transfer completes(data counter is zero). This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = data finish detect | 0 |
| Busy Finish (BusyFin) | [3] R/C | Only busy check finish. This flag is cleared by setting to one this bit<br>0 = not detect, 1 = busy finish detect | 0 |
| Start Bit Error(SbitErr) | [2] R/C | Start bit is not detected on all data signals in wide bus mode. This flag is cleared by setting to one this bit.<br>0 = not detect, 1 = command end | 0 |
| Tx Data progress On(TxDatOn) | [1] | Data transmit in progress<br>0 = not active, 1 = data Tx in progress | 0 |
| Rx Data Progress On(RxDatOn) | [0] | Data receive in progress<br>0 = not active, 1 = data Rx in progress | 0 |

### SDI FIFO Status Register(SDIFSTA)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIFSTA | 0x5A000038 | R | SDI FIFO Status Register | 0x0 |

| SDIFSTA | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [31:14] | | |
| FIFO available Detect for Tx (TFDET) | [13] | This bit indicates that FIFO data is available for transmit when DatMode is data transmit mode. If DMA mode is enable, sd host requests DMA operation.<br>0 = not detect(FIFO full), 1 = detect(0 ≤ FIFO ≤ 63) | 0 |
| FIFO available Detect for Rx (RFDET) | [12] | This bit indicates that FIFO data is available for receive when DatMode is data receive mode. If DMA mode is enable, sd host requests DMA operation.<br>0 = not detect(FIFO empty), 1 = detect(1 ≤ FIFO ≤ 64) | 0 |
| Tx FIFO Half Full (TFHalf) | [11] | This bit sets to 1 whenever Tx FIFO is less than 33byte.<br>0 = 33 ≤ Tx FIFO ≤ 64, 1 = 0 ≤ Tx FIFO ≤ 32 | 0 |
| Tx FIFO Empty (TFEmpty) | [10] | This bit sets to 1 whenever Tx FIFO is empty.<br>0 = 1 ≤ Tx FIFO ≤ 64, 1 = Empty(0byte) | 0 |
| Rx FIFO Last Data Ready (RFLast) | [9] | This bit sets to 1 whenever Rx FIFO has last data of all block.<br>0 = not received yet, 1 = Last data ready | 0 |
| Rx FIFO Full (RFFull) | [8] | This bit sets to 1 whenever Rx FIFO is full.<br>0 = 0 ≤ Rx FIFO ≤ 63, 1 = Full(64byte) | 0 |
| Rx FIFO Half Full (RFHalf) | [7] | This bit sets to 1 whenever Rx FIFO is more than 31byte.<br>0 = 0 ≤ Rx FIFO ≤ 31, 1 = 32 ≤ Rx FIFO ≤ 64 | 0 |
| FIFO Count (FFCNT) | [6:0] | Number of data(byte) in FIFO | 0000000 |

### SDI Data Register(SDIDAT)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIDAT | 0x5A00003c(Li/W, Li/B, Bi/W)<br>0x5A00003f(Bi/B) | R/W | SDI Data Register | 0x0 |

| SDIDAT | Bit | Description | Initial State |
|---|---|---|---|
| Data Register | [31:0] | This field contains the data to be transmitted or received over the SDI channel | 0x00000000 |

* (Li/W, Li/B) : Access by Word/Byte unit when endian mode is Little
* (Bi/W) : Access by Word unit when endian mode is Big
* (Bi/B) : Access by Byte unit when endian mode is Big

### SDI Interrupt Mask Register(SDIIMSK)

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SDIIMSK | 0x5A000040 | R/W | SDI Interrupt Mask Register | 0x0 |

| SDICON | Bit | Description | Initial Value |
|---|---|---|---|
| Reserved | [31:18] | | |
| RspCrc Interrupt Enable (RspCrcInt) | [17] | Determines SDI generate an interrupt if response CRC check fails<br>0 = disable, 1 = interrupt enable | 0 |
| CmdSent Interrupt Enable (CmdSentInt) | [16] | Determines SDI generate an interrupt if command sent(no response required)<br>0 = disable, 1 = interrupt enable | 0 |
| CmdTout Interrupt Enable (CmdToutInt) | [15] | Determines SDI generate an interrupt if command response timeout occurs<br>0 = disable, 1 = interrupt enable | 0 |
| RspEnd Interrupt Enable (RspEndInt) | [14] | Determines SDI generate an interrupt if command response received<br>0 = disable, 1 = interrupt enable | 0 |
| RWaitReq Interrupt Enable (RWReqInt) | [13] | Determines SDI generate an interrupt if read wait request occur.<br>0 = disable, 1 = interrupt enable | 0 |
| IOIntDet Interrupt Enable (IntDetInt) | [12] | Determines SDI generate an interrupt if sd host receives SDIO Interrupt from the card(for SDIO).<br>0 = disable, 1 = interrupt enable | 0 |
| FFfail Interrupt Enable (FFfailInt) | [11] | Determines SDI generate an interrupt if FIFO fail error occurs<br>0 = disable, 1 = interrupt enable | 0 |
| CrcSta Interrupt Enable (CrcStaInt) | [10] | Determines SDI generate an interrupt if CRC status error occurs<br>0 = disable, 1 = interrupt enable | 0 |
| DatCrc Interrupt Enable (DatCrcInt) | [9] | Determines SDI generate an interrupt if data receive CRC failed<br>0 = disable, 1 = interrupt enable | 0 |
| DatTout Interrupt Enable (DatToutInt) | [8] | Determines SDI generate an interrupt if data receive timeout occurs<br>0 = disable, 1 = interrupt enable | 0 |
| DatFin Interrupt Enable (DatFinInt) | [7] | Determines SDI generate an interrupt if data counter is zero<br>0 = disable, 1 = interrupt enable | 0 |
| BusyFin Interrupt Enable(BusyFinInt) | [6] | Determines SDI generate an interrupt if only busy check completes<br>0 = disable, 1 = interrupt enable | 0 |
| SBitErr Interrupt Enable (SBitErrInt) | [5] | Determines SDI generate an interrupt if start bit error detect<br>0 = disable, 1 = interrupt enable | 0 |
| TFHalf Interrupt Enable (TFHalfInt) | [4] | Determines SDI generate an interrupt if Tx FIFO fills half<br>0 = disable, 1 = interrupt enable | 0 |
| TFEmpty Interrupt Enable(TFEmptInt) | [3] | Determines SDI generate an interrupt if Tx FIFO is empty<br>0 = disable, 1 = interrupt enable | 0 |
| RFLast Interrupt Enable (RFLastInt) | [2] | Determines SDI generate an interrupt if Rx FIFO has last data<br>0 = disable, 1 = interrupt enable | 0 |
| RFFull Interrupt Enable (RFFullInt) | [1] | Determines SDI generate an interrupt if Rx FIFO fills full<br>0 = disable, 1 = interrupt enable | 0 |
| RFHalf Interrupt Enable (RFHalfInt) | [0] | Determines SDI generate an interrupt if Rx FIFO fills half<br>0 = disable, 1 = interrupt enable | 0 |

**NOTE**

# 20 IIC-BUS INTERFACE

## OVERVIEW

The S3C2440X RISC microprocessor can support a multi-master IIC-bus serial interface. A dedicated serial data line (SDA) and a serial clock line (SCL) carry information between bus masters and peripheral devices which are connected to the IIC-bus. The SDA and SCL lines are bi-directional.

In multi-master IIC-bus mode, multiple S3C2440X RISC microprocessors can receive or transmit serial data to or from slave devices. The master S3C2440X can initiate and terminate a data transfer over the IIC-bus. The IIC-bus in the S3C2440X uses Standard bus arbitration procedure.

To control multi-master IIC-bus operations, values must be written to the following registers:

— Multi-master IIC-bus control register, IICCON

— Multi-master IIC-bus control/status register, IICSTAT

— Multi-master IIC-bus Tx/Rx data shift register, IICDS

— Multi-master IIC-bus address register, IICADD

When the IIC-bus is free, the SDA and SCL lines should be both at High level. A High-to-Low transition of SDA can initiate a Start condition. A Low-to-High transition of SDA can initiate a Stop condition while SCL remains steady at High Level.

The Start and Stop conditions can always be generated by the master devices. A 7-bit address value in the first data byte, which is put onto the bus after the Start condition has been initiated, can determine the slave device which the bus master device has selected. The $8^{th}$ bit determines the direction of the transfer (read or write).

Every data byte put onto the SDA line should be eight bits in total. The bytes can be unlimitedly sent or received during the bus transfer operation. Data is always sent from most-significant bit (MSB) first, and every byte should be immediately followed by acknowledge (ACK) bit.

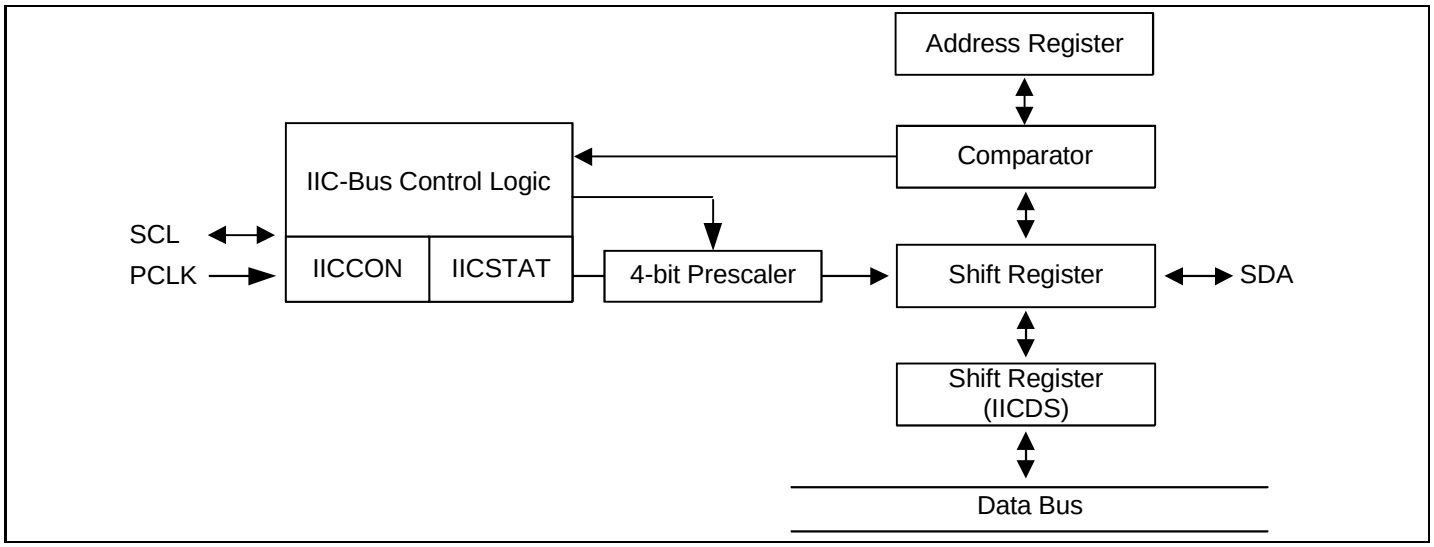



**Figure 20-1. IIC-Bus Block Diagram**

## IIC-BUS INTERFACE

The S3C2440X IIC-bus interface has four operation modes:

— Master transmitter mode
— Master receive mode
— Slave transmitter mode
— Slave receive mode

Functional relationships among these operating modes are described below.

### START AND STOP CONDITIONS

When the IIC-bus interface is inactive, it is usually in Slave mode. In other words, the interface should be in Slave mode before detecting a Start condition on the SDA line (a Start condition can be initiated with a High-to-Low transition of the SDA line while the clock signal of SCL is High). When the interface state is changed to Master mode, a data transfer on the SDA line can be initiated and SCL signal generated.

A Start condition can transfer a one-byte serial data over the SDA line, and a Stop condition can terminate the data transfer. A Stop condition is a Low-to-High transition of the SDA line while SCL is High. Start and Stop conditions are always generated by the master. The IIC-bus gets busy when a Start condition is generated. A Stop condition will make the IIC-bus free.

When a master initiates a Start condition, it should send a slave address to notify the slave device. One byte of address field consists of a 7-bit address and a 1-bit transfer direction indicator (showing write or read).
If bit 8 is 0, it indicates a write operation (transmit operation); if bit 8 is 1, it indicates a request for data read (receive operation).

The master will finish the transfer operation by transmitting a Stop condition. If the master wants to continue the data transmission to the bus, it should generate another Start condition as well as a slave address. In this way, the read-write operation can be performed in various formats.

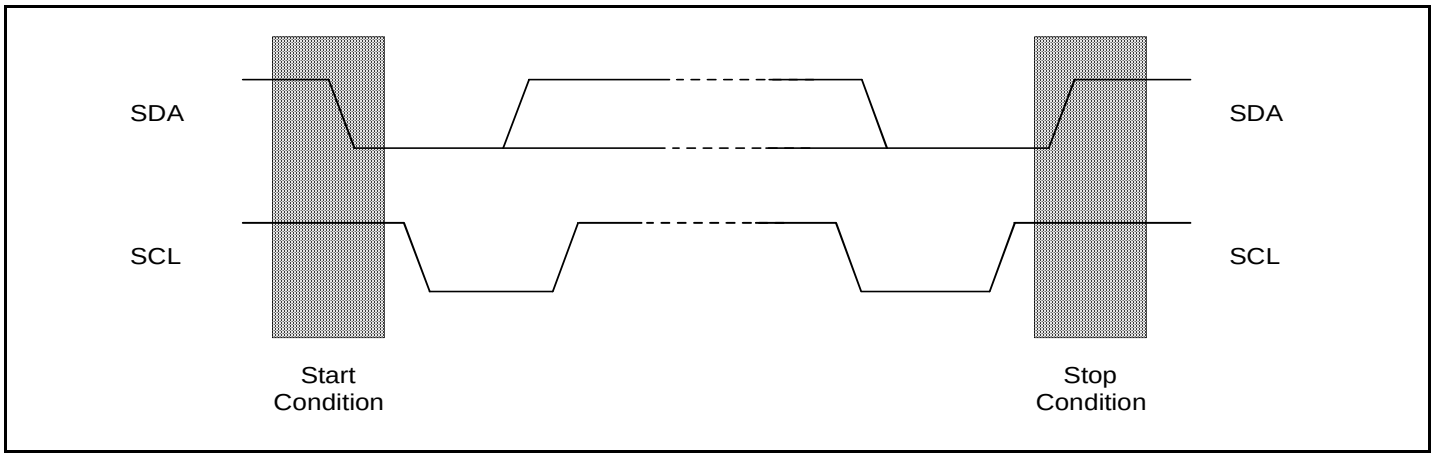



**Figure 20-2. Start and Stop Condition**

## DATA TRANSFER FORMAT

Every byte placed on the SDA line should be eight bits in length. The bytes can be unlimitedly transmitted per transfer. The first byte following a Start condition should have the address field. The address field can be transmitted by the master when the IIC-bus is operating in Master mode. Each byte should be followed by an acknowledgement (ACK) bit. The MSB bit of the serial data and addresses are always sent first.

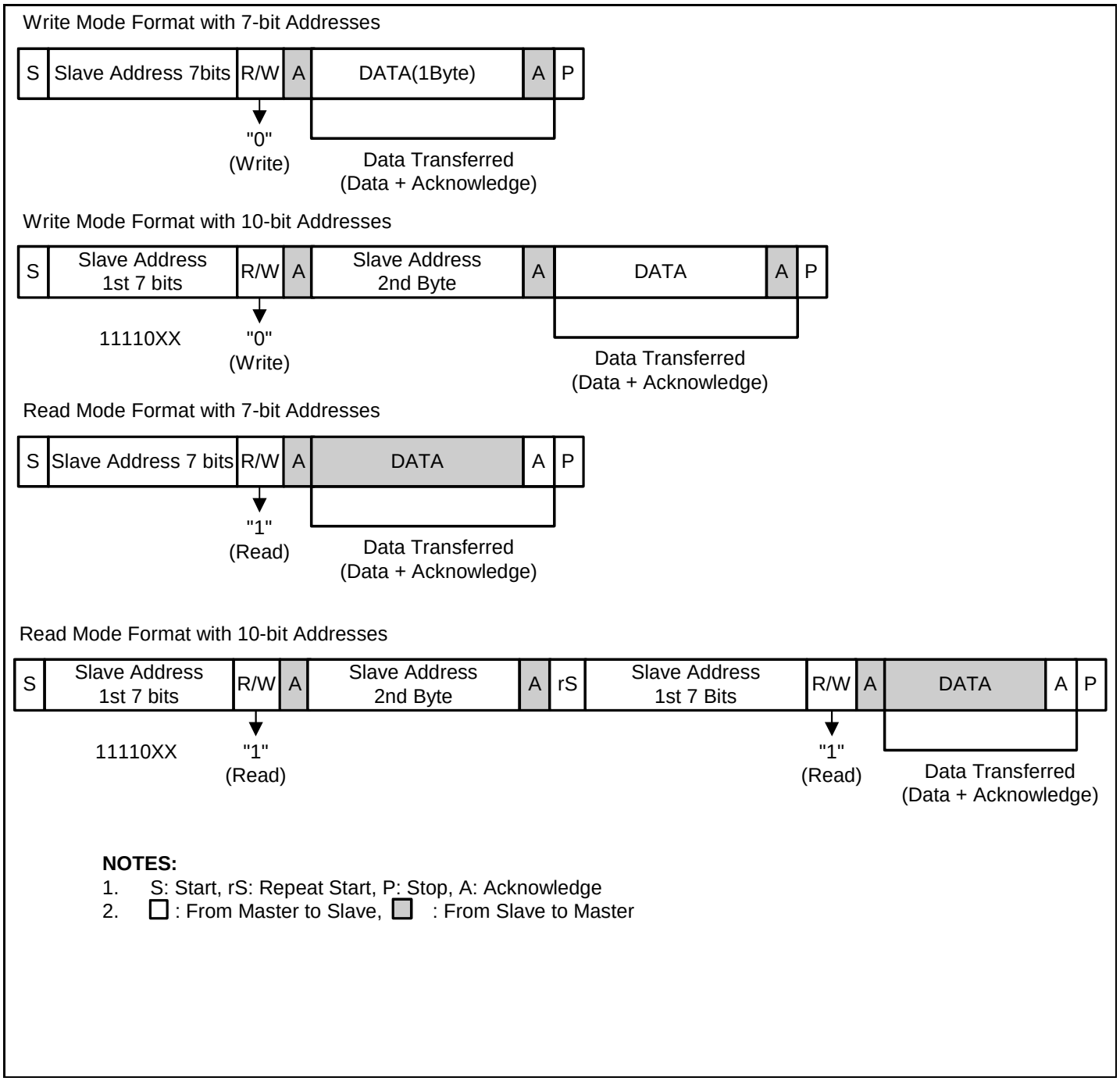



**Figure 20-3. IIC-Bus Interface Data Format**

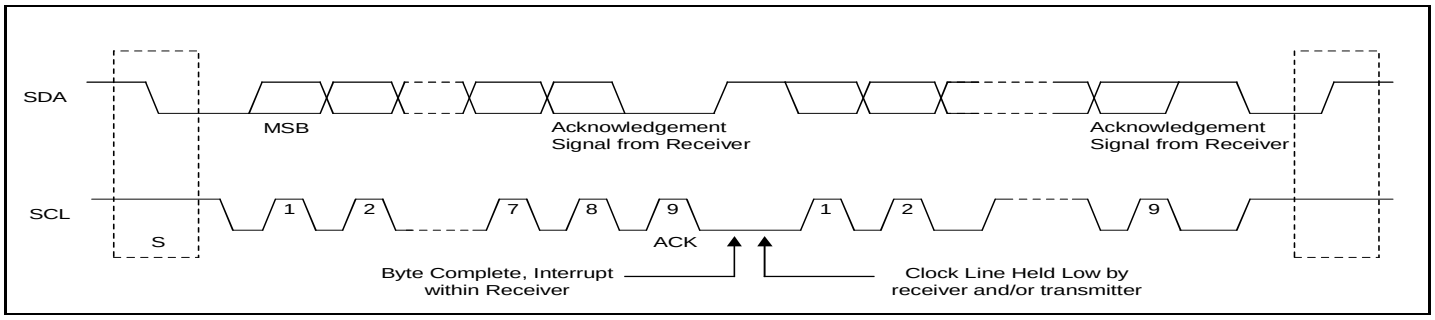



**Figure 20-4. Data Transfer on the IIC-Bus**

### ACK SIGNAL TRANSMISSION

To complete a one-byte transfer operation, the receiver should send an ACK bit to the transmitter. The ACK pulse should occur at the ninth clock of the SCL line. Eight clocks are required for the one-byte data transfer. The master should generate the clock pulse required to transmit the ACK bit.

The transmitter should release the SDA line by making the SDA line High when the ACK clock pulse is received. The receiver should also drive the SDA line Low during the ACK clock pulse so that the SDA keeps Low during the High period of the ninth SCL pulse.

The ACK bit transmit function can be enabled or disabled by software (IICSTAT). However, the ACK pulse on the ninth clock of SCL is required to complete the one-byte data transfer operation.

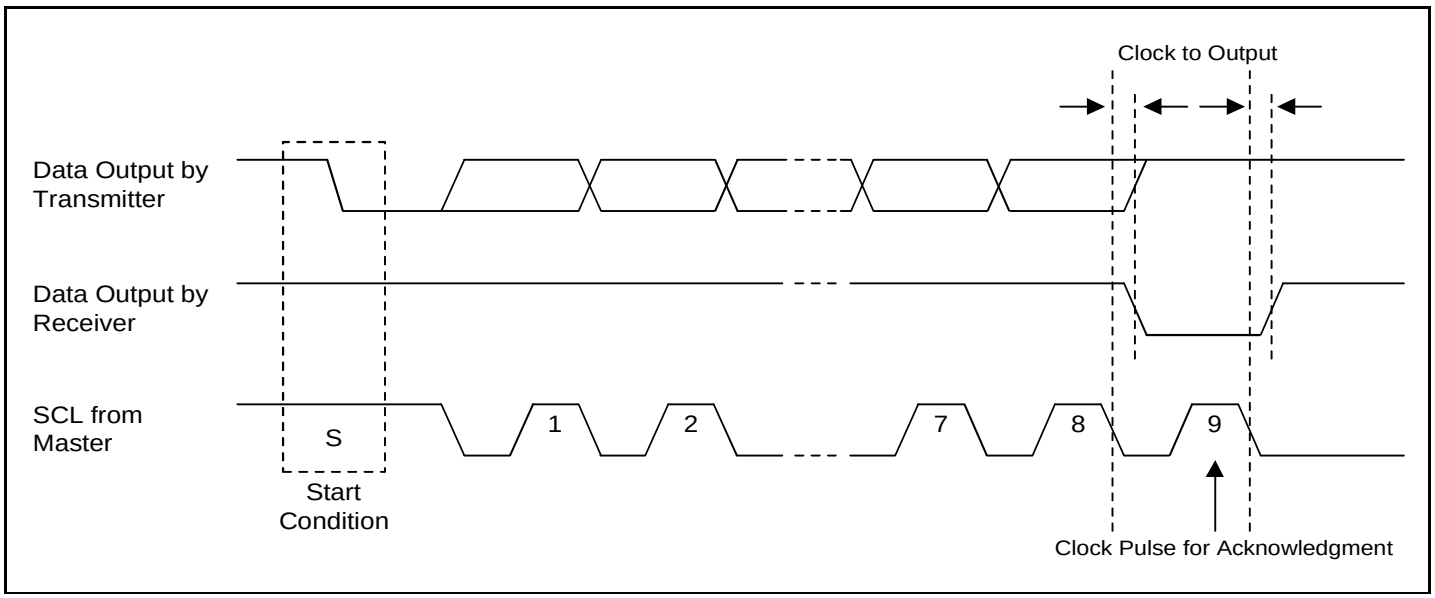



**Figure 20-5. Acknowledge on the IIC-Bus**

## READ-WRITE OPERATION

In Transmitter mode, when the data is transferred, the IIC-bus interface will wait until IIC-bus Data Shift (IICDS) register receives a new data. Before the new data is written into the register, the SCL line will be held low, and then released after it is written. The S3C2440X should hold the interrupt to identify the completion of current data transfer. After the CPU receives the interrupt request, it should write a new data into the IICDS register, again.

In Receive mode, when data is received, the IIC-bus interface will wait until IICDS register is read. Before the new data is read out, the SCL line will be held low and then released after it is read. The S3C2440X should hold the interrupt to identify the completion of the new data reception. After the CPU receives the interrupt request, it should read the data from the IICDS register.

## BUS ARBITRATION PROCEDURES

Arbitration takes place on the SDA line to prevent the contention on the bus between two masters. If a master with a SDA High level detects the other master with a SDA active Low level, it will not initiate a data transfer because the current level on the bus does not correspond to its own. The arbitration procedure will be extended until the SDA line turns High.

However, when the masters simultaneously lower the SDA line, each master should evaluate whether or not the mastership is allocated to itself. For the purpose of evaluation, each master should detect the address bits. While each master generates the slaver address, it should also detect the address bit on the SDA line because the SDA line is likely to get Low rather than to keep High. Assume that one master generates a Low as first address bit, while the other master is maintaining High. In this case, both masters will detect Low on the bus because the Low status is superior to the High status in power. When this happens, Low (as the first bit of address) generating master will get the mastership while High (as the first bit of address) generating master should withdraw the mastership. If both masters generate Low as the first bit of address, there should be arbitration for the second address bit, again. This arbitration will continue to the end of last address bit.

## ABORT CONDITIONS

If a slave receiver cannot acknowledge the confirmation of the slave address, it should hold the level of the SDA line High. In this case, the master should generate a Stop condition and to abort the transfer.

If a master receiver is involved in the aborted transfer, it should signal the end of the slave transmit operation by canceling the generation of an ACK after the last data byte received from the slave. The slave transmitter should then release the SDA to allow a master to generate a Stop condition.

## CONFIGURING IIC-BUS

To control the frequency of the serial clock (SCL), the 4-bit prescaler value can be programmed in the IICCON register. The IIC-bus interface address is stored in the IIC-bus address (IICADD) register. (By default, the IIC-bus interface address has an unknown value.)

## FLOWCHARTS OF OPERATIONS IN EACH MODE

The following steps must be executed before any IIC Tx/Rx operations.

1) Write own slave address on IICADD register, if needed.
2) Set IICCON register.
    a) Enable interrupt
    b) Define SCL period
3) Set IICSTAT to enable Serial Output

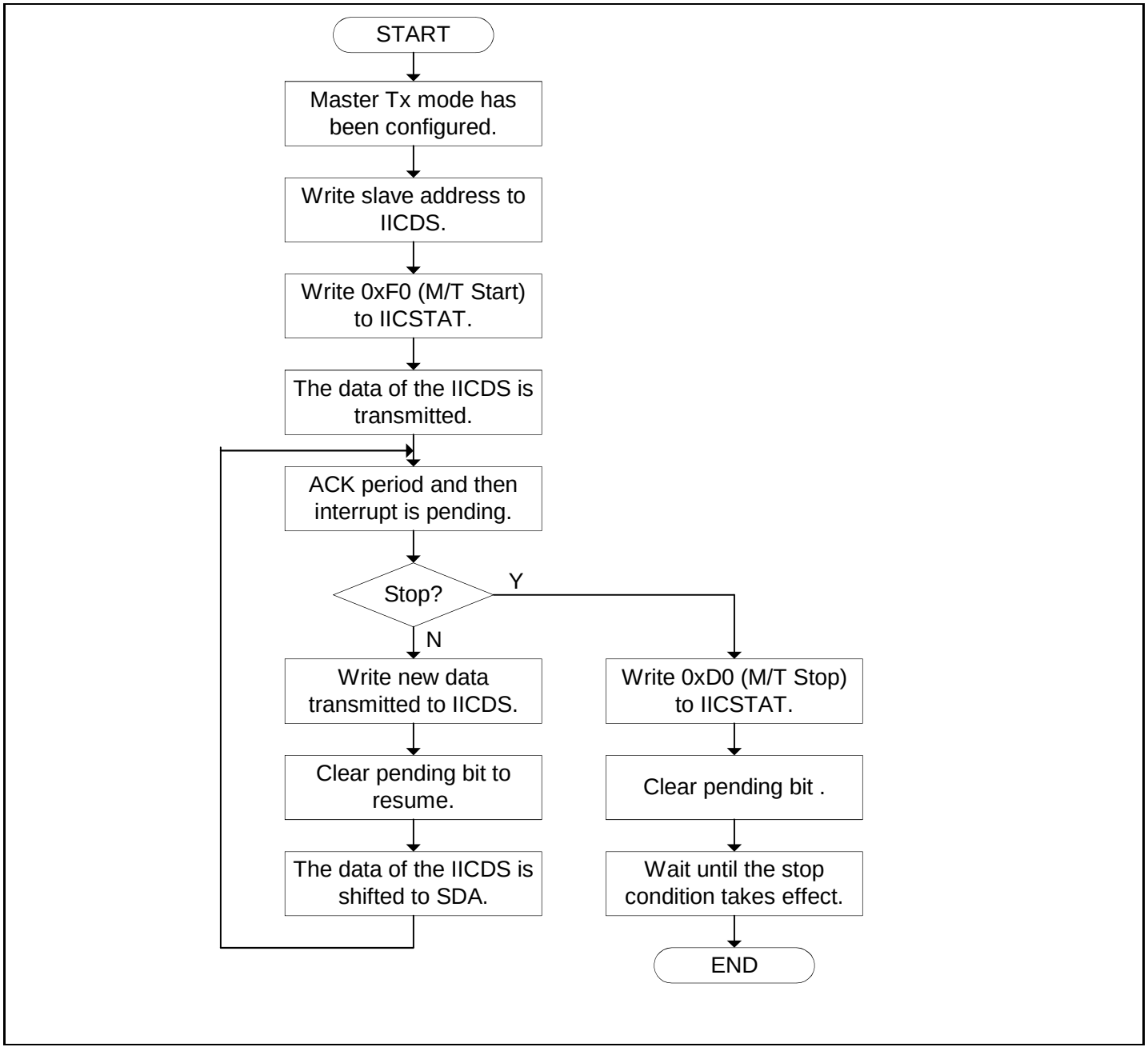



**Figure 20-6 Operations for Master/Transmitter Mode**

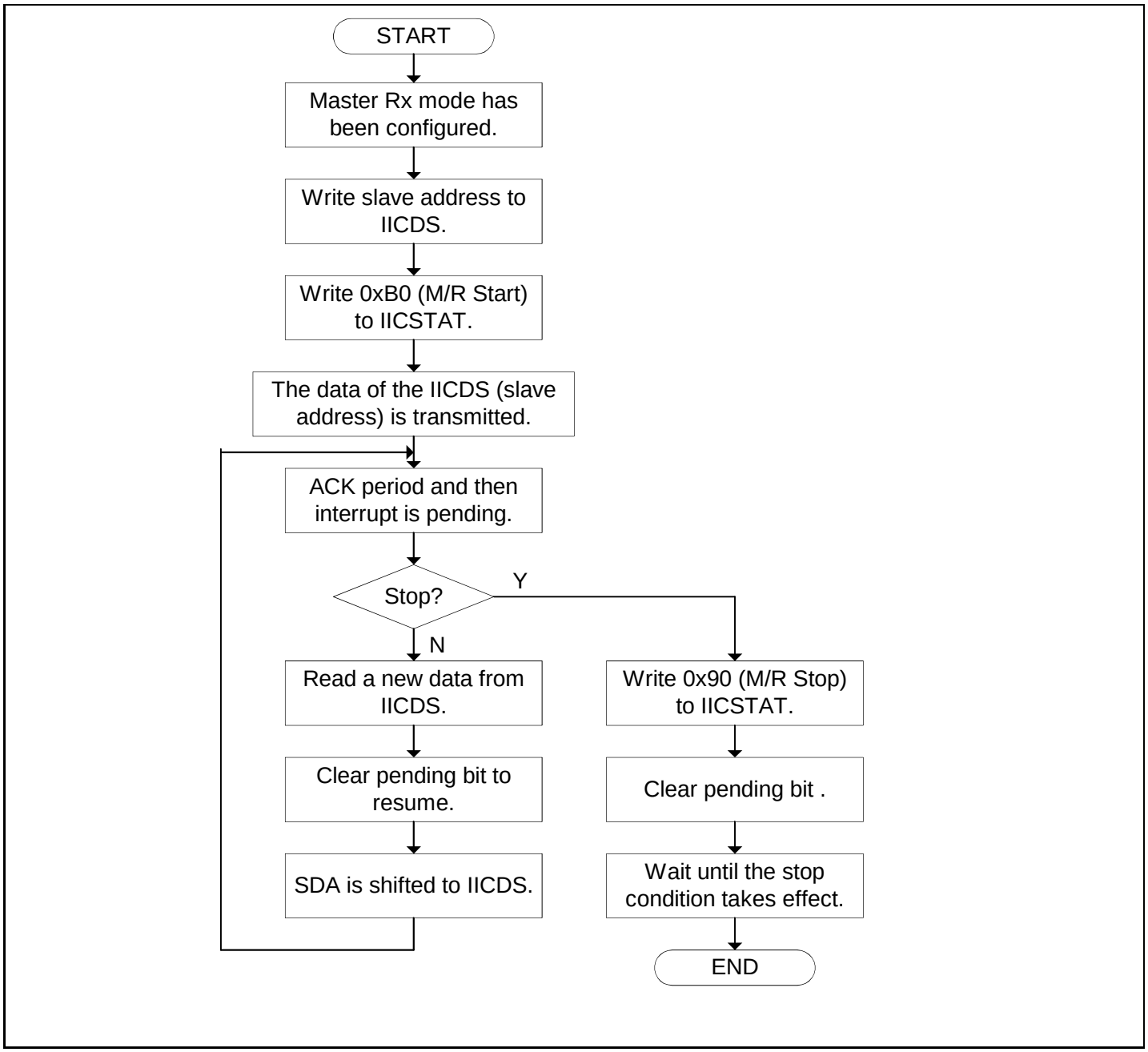



**Figure 20-7 Operations for Master/Receiver Mode**

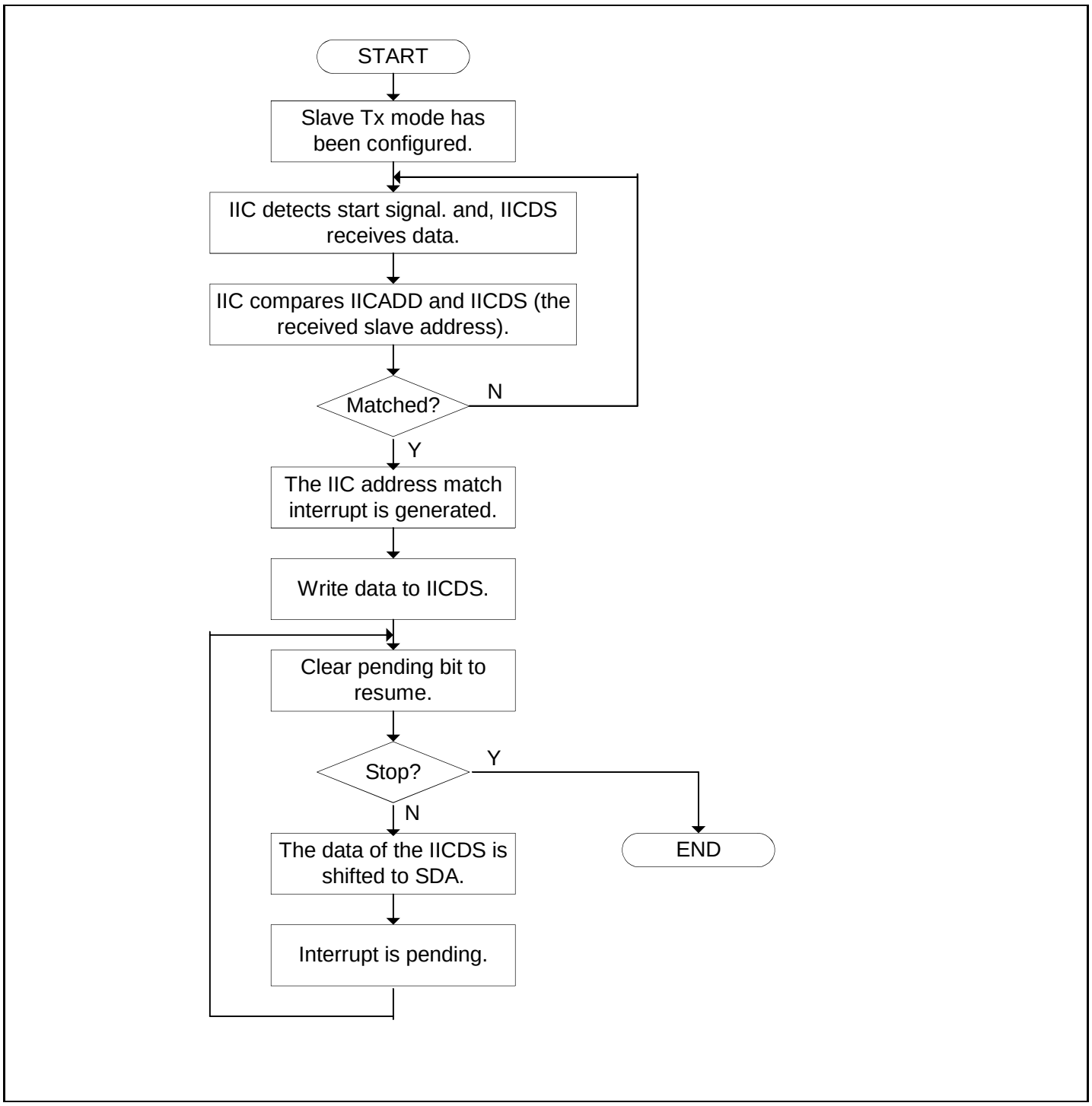



**Figure 20-8 Operations for Slave/Transmitter Mode**

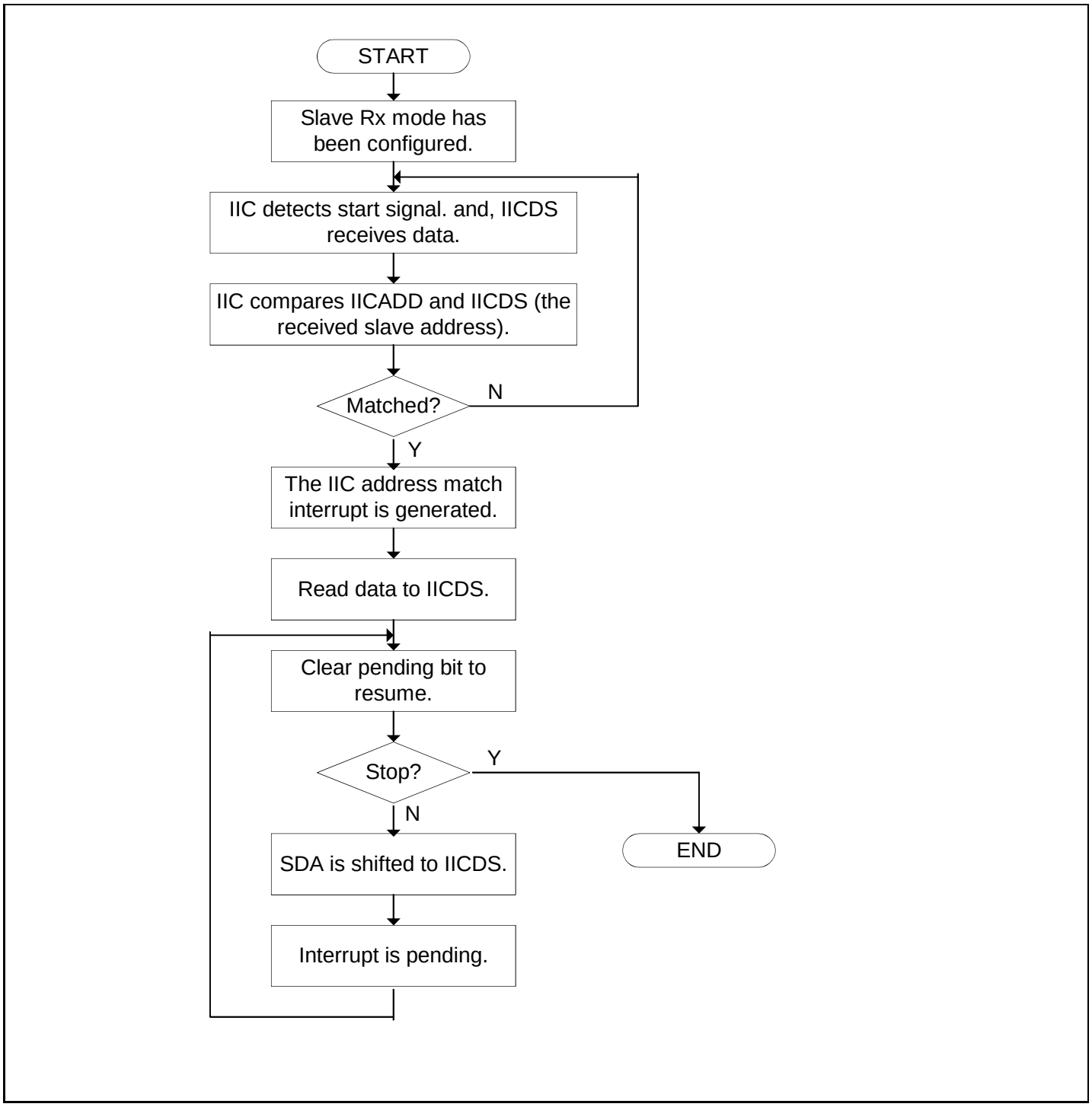



**Figure 20-9 Operations for Slave/Receiver Mode**

# IIC-BUS INTERFACE SPECIAL REGISTERS

## MULTI-MASTER IIC-BUS CONTROL (IICCON) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IICCON | 0x54000000 | R/W | IIC-Bus control register | 0x0X |

| IICCON | Bit | Description | Initial State |
|---|---|---|---|
| Acknowledge generation (1) | [7] | IIC-bus acknowledge enable bit.<br>0 : Disable<br>1 : Enable<br>In Tx mode, the IICSDA is free in the ack time.<br>In Rx mode, the IICSDA is L in the ack time. | 0 |
| Tx clock source selection | [6] | Source clock of IIC-bus transmit clock prescaler selection bit.<br>0 : IICCLK = $f_{PCLK}$ /16<br>1 : IICCLK = $f_{PCLK}$ /512 | 0 |
| Tx/Rx Interrupt (5) | [5] | IIC-Bus Tx/Rx interrupt enable/disable bit.<br>0 : Disable, 1 : Enable | 0 |
| Interrupt pending flag (2) (3) | [4] | IIC-bus Tx/Rx interrupt pending flag. This bit cannot be written to 1. When this bit is read as 1, the IICSCL is tied to L and the IIC is stopped. To resume the operation, clear this bit as 0.<br>0 : 1) No interrupt pending (when read).<br>2) Clear pending condition & Resume the operation (when write).<br>1 : 1) Interrupt is pending (when read)<br>2) N/A (when write) | 0 |
| Transmit clock value (4) | [3:0] | IIC-Bus transmit clock prescaler.<br>IIC-Bus transmit clock frequency is determined by this 4-bit prescaler value, according to the following formula:<br>Tx clock = IICCLK/(IICCON[3:0]+1). | Undefined |

**Notes:**

1. Interfacing with EEPROM, the ack generation may be disabled before reading the last data in order to generate the STOP condition in Rx mode.
2. An IIC-bus interrupt occurs 1) when a 1-byte transmit or receive operation is completed, 2) when a general call or a slave address match occurs, or 3) if bus arbitration fails.
3. To adjust the setup time of IICSDA before IISSCL rising edge, IICDS has to be written before clearing the IIC interrupt pending bit.
4. IICCLK is determined by IICCON[6].
   Tx clock can vary by SCL transition time.
   When IICCON[6]=0, IICCON[3:0]=0x0 or 0x1 is not available.
5. If the IICON[5]=0, IICON[4] does not operate correctly.
   So, It is recommended that you should set IICCON[5]=1, although you does not use the IIC interrupt.

## MULTI-MASTER IIC-BUS CONTROL/STATUS (IICSTAT) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IICSTAT | 0x54000004 | R/W | IIC-Bus control/status register | 0x0 |

| IICSTAT | Bit | Description | Initial State |
|---|---|---|---|
| Mode selection | [7:6] | IIC-bus master/slave Tx/Rx mode select bits.<br>00 : Slave receive mode<br>01 : Slave transmit mode<br>10 : Master receive mode<br>11 : Master transmit mode | 00 |
| Busy signal status / START STOP condition | [5] | IIC-Bus busy signal status bit.<br>0 : read) Not busy (when read)<br>write) STOP signal generation<br>1 : read) Busy (when read)<br>write) START signal generation.<br>The data in IICDS will be transferred automatically just after the start signal. | 0 |
| Serial output | [4] | IIC-bus data output enable/disable bit.<br>0 : Disable Rx/Tx, 1 : Enable Rx/Tx | 0 |
| Arbitration status flag | [3] | IIC-bus arbitration procedure status flag bit.<br>0 : Bus arbitration successful<br>1 : Bus arbitration failed during serial I/O | 0 |
| Address-as-slave status flag | [2] | IIC-bus address-as-slave status flag bit.<br>0 : Cleared when START/STOP condition was detected<br>1 : Received slave address matches the address value in the IICADD | 0 |
| Address zero status flag | [1] | IIC-bus address zero status flag bit.<br>0 : Cleared when START/STOP condition was detected<br>1 : Received slave address is 00000000b. | 0 |
| Last-received bit status flag | [0] | IIC-bus last-received bit status flag bit.<br>0 : Last-received bit is 0 (ACK was received).<br>1 : Last-received bit is 1 (ACK was not received). | 0 |

### MULTI-MASTER IIC-BUS ADDRESS (IICADD) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IICADD | 0x54000008 | R/W | IIC-Bus address register | 0xXX |

| IICADD | Bit | Description | Initial State |
|---|---|---|---|
| Slave address | [7:0] | 7-bit slave address, latched from the IIC-bus.<br>When serial output enable = 0 in the IICSTAT, IICADD is write-enabled. The IICADD value can be read any time, regardless of the current serial output enable bit (IICSTAT) setting.<br>Slave address : [7:1]<br>Not mapped : [0] | XXXXXXXX |

### MULTI-MASTER IIC-BUS TRANSMIT/RECEIVE DATA SHIFT (IICDS) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IICDS | 0x5400000C | R/W | IIC-Bus transmit/receive data shift register | 0xXX |

| IICDS | Bit | Description | Initial State |
|---|---|---|---|
| Data shift | [7:0] | 8-bit data shift register for IIC-bus Tx/Rx operation.<br>When serial output enable = 1 in the IICSTAT, IICDS is write-enabled. The IICDS value can be read any time, regardless of the current serial output enable bit (IICSTAT) setting. | XXXXXXXX |

## MULTI-MASTER IIC-BUS LINE CONTROL(IICLC) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IICLC | 0x54000010 | R/W | IIC-Bus multi-master line control register | 0x00 |

| IICLC | Bit | Description | Initial State |
|---|---|---|---|
| Filter Enable | [2] | IIC-bus filter enable bit.<br>When SDA port is operating as input, the filter enable bit should be High. This filter can prevent from occurred error by a glitch during double of PCLK time.<br>0 : Filter disable<br>1 : Filter enable | 0 |
| SDA output delay | [1:0] | IIC-Bus SDA line delay length selection bits.<br>It is delayed for following clock time(PCLK) after High-to-Low transition of SCL line is occurred. Then a High-to-Low transition of the SDA line will be occurred.<br>00 : 0 clocks 01 : 5 clocks<br>10 : 10 clocks 11 : 15 clocks | 00 |

# 21 IIS-BUS INTERFACE

## OVERVIEW

Currently, many digital audio systems are attracting the consumers on the market, in the form of compact discs, digital audio tapes, digital sound processors, and digital TV sound. The S3C2440X Inter-IC Sound (IIS) bus interface can be used to implement a CODEC interface to an external 8/16-bit stereo audio CODEC IC for mini-disc and portable applications. The IIS bus interface supports both IIS bus data format and MSB-justified data format. The interface provides DMA transfer mode for FIFO access instead of an interrupt. It can transmit and receive data simultaneously as well as transmit or receive data alternatively at a time.

## BLOCK DIAGRAM

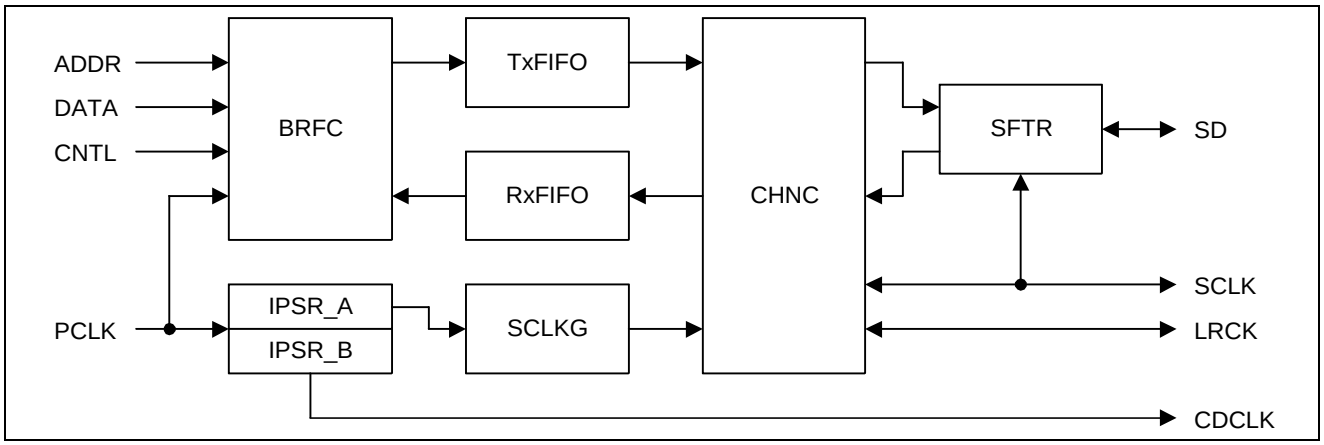



**Figure 21-1. IIS-Bus Block Diagram**

## FUNCTIONAL DESCRIPTIONS

Bus interface, register bank, and state machine (BRFC): Bus interface logic and FIFO access are controlled by the state machine.

5-bit dual prescaler (IPSR): One prescaler is used as the master clock generator of the IIS bus interface and the other is used as the external CODEC clock generator.

64-byte FIFOs (TxFIFO and RxFIFO): In transmit data transfer, data are written to TxFIFO, and, in the receive data transfer, data are read from RxFIFO.

Master IISCLK generator (SCLKG): In master mode, serial bit clock is generated from the master clock.

Channel generator and state machine (CHNC): IISCLK and IISLRCK are generated and controlled by the channel state machine.

16-bit shift register (SFTR): Parallel data is shifted to serial data output in the transmit mode, and serial data input is shifted to parallel data in the receive mode.

### TRANSMIT OR RECEIVE ONLY MODE

Normal transfer

IIS control register has FIFO ready flag bits for transmit and receive FIFOs. When FIFO is ready to transmit data, the FIFO ready flag is set to '1' if transmit FIFO is not empty.
If transmit FIFO is empty, FIFO ready flag is set to '0'. When receive FIFO is not full, the FIFO ready flag for receive FIFO is set to '1' ; it indicates that FIFO is ready to receive data. If receive FIFO is full, FIFO ready flag is set to '0'. These flags can determine the time that CPU is to write or read FIFOs. Serial data can be transmitted or received while the CPU is accessing transmit and receive FIFOs in this way.

### DMA TRANSFER

In this mode, transmit or receive FIFO is accessible by the DMA controller. DMA service request in transmit or receive mode is made by the FIFO ready flag automatically.

### TRANSMIT AND RECEIVE MODE

In this mode, IIS bus interface can transmit and receive data simultaneously.

## AUDIO SERIAL INTERFACE FORMAT

### IIS-BUS FORMAT

The IIS bus has four lines including serial data input (IISDI), serial data output (IISDO), left/right channel select (IISLRCK), and serial bit clock (IISCLK); the device generating IISLRCK and IISCLK is the master.

Serial data is transmitted in 2's complement with the MSB first. The MSB is transmitted first because the transmitter and receiver may have different word lengths. The transmitter does not have to know how many bits the receiver can handle, nor does the receiver need to know how many bits are being transmitted.

When the system word length is greater than the transmitter word length, the word is truncated (least significant data bits are set to '0') for data transmission. If the receiver gets more bits than its word length, the bits after the LSB are ignored. On the other hand, if the receiver gets fewer bits than its word length, the missing bits are set to zero internally. And therefore, the MSB has a fixed position, whereas the position of the LSB depends on the word length. The transmitter sends the MSB of the next word at one clock period whenever the IISLRCK is changed.

Serial data sent by the transmitter may be synchronized with either the trailing (HIGH to LOW) or the leading (LOW to HIGH) edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and so there are some restrictions when transmitting data that is synchronized with the leading edge.

The LR channel select line indicates the channel being transmitted. IISLRCK may be changed either on a trailing or leading edge of the serial clock, but it does not need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The IISLRCK line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clear the input for the next word.

### MSB (LEFT) JUSTIFIED

MSB / left justified bus format is the same as IIS bus format architecturally. Only, different from the IIS bus format, the MSB justified format realizes that the transmitter always sends the MSB of the next word whenever the IISLRCK is changed.

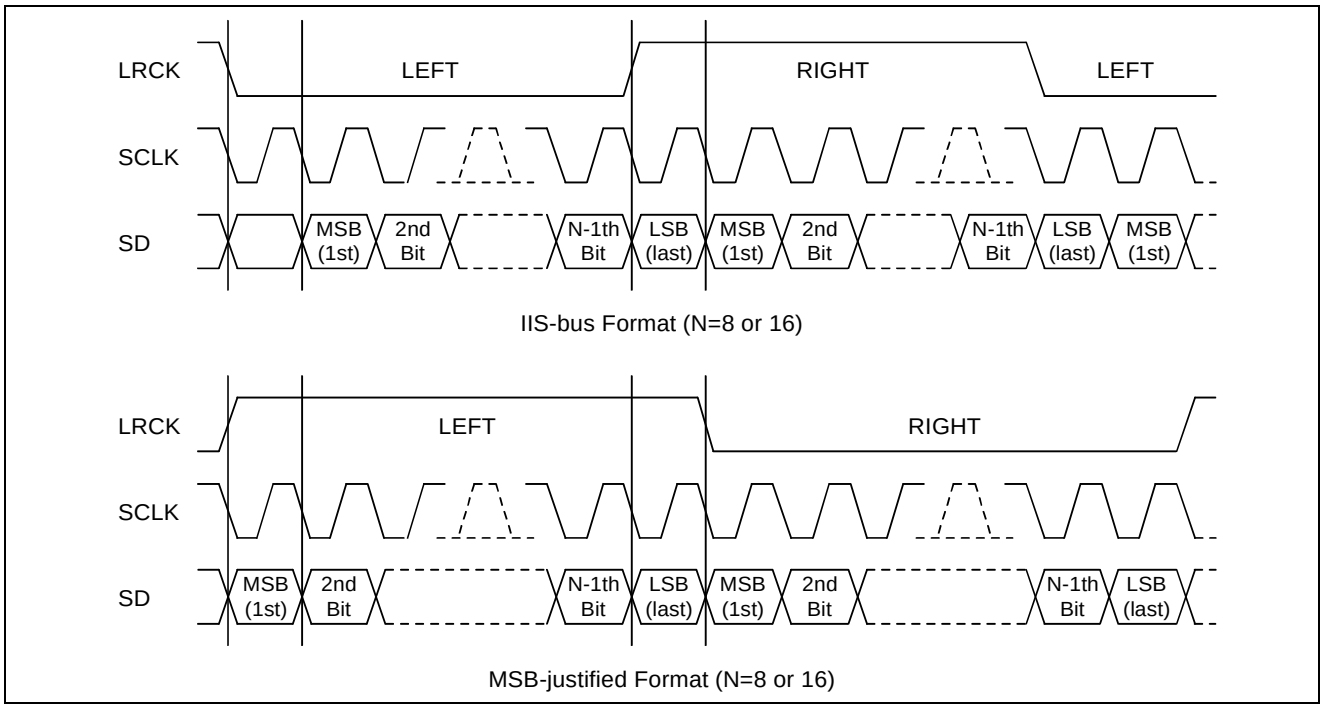



**Figure 21-2. IIS-Bus and MSB (Left)-justified Data Interface Formats**

### SAMPLING FREQUENCY AND MASTER CLOCK

Master clock frequency (PCLK) can be selected by sampling frequency as shown in Table 21-1. Because PCLK is made by IIS prescaler, the prescaler value and PCLK type (256 or 384fs) should be determined properly. Serial bit clock frequency type (16/32/48fs) can be selected by the serial bit per channel and PCLK as shown in Table 21-2.

**Table 21-1 CODEC clock (CODECLK = 256 or 384fs)**

| IISLRCK (fs) | 8.000 KHz | 11.025 KHz | 16.000 KHz | 22.050 KHz | 32.000 KHz | 44.100 KHz | 48.000 KHz | 64.000 KHz | 88.200 KHz | 96.000 KHz |
|---|---|---|---|---|---|---|---|---|---|---|
| CODECLK (MHz) | 256fs | | | | | | | | | |
| | 2.0480 | 2.8224 | 4.0960 | 5.6448 | 8.1920 | 11.2896 | 12.2880 | 16.3840 | 22.5792 | 24.5760 |
| | 384fs | | | | | | | | | |
| | 3.0720 | 4.2336 | 6.1440 | 8.4672 | 12.2880 | 16.9344 | 18.4320 | 24.5760 | 33.8688 | 36.8640 |

**Table 21-2 Usable serial bit clock frequency (IISCLK = 16 or 32 or 48fs)**

| Serial bit per channel | 8-bit | 16-bit |
|---|---|---|
| Serial clock frequency (IISCLK) | | |
| @CODECLK = 256fs | 16fs, 32fs | 32fs |
| @CODECLK = 384fs | 16fs, 32fs, 48fs | 32fs, 48fs |

## IIS-BUS INTERFACE SPECIAL REGISTERS

### IIS CONTROL (IISCON) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IISCON | 0x55000000 (Li/HW, Li/W, Bi/W)<br>0x55000002 (Bi/HW) | R/W | IIS control register | 0x100 |

| IISCON | Bit | Description | Initial State |
|---|---|---|---|
| Left/Right channel index<br>(Read only) | [8] | 0 = Left<br>1 = Right | 1 |
| Transmit FIFO ready flag<br>(Read only) | [7] | 0 = Not ready (empty)<br>1 = Ready (not empty) | 0 |
| Receive FIFO ready flag<br>(Read only) | [6] | 0 = Not ready (full)<br>1 = Ready (not full) | 0 |
| Transmit DMA service request | [5] | 0 = Disable<br>1 = Enable | 0 |
| Receive DMA service request | [4] | 0 = Disable<br>1 = Enable | 0 |
| Transmit channel idle command | [3] | In Idle state the IISLRCK is inactive (Pause Tx).<br>0 = Not idle<br>1 = Idle | 0 |
| Receive channel idle command | [2] | In Idle state the IISLRCK is inactive (Pause Rx).<br>0 = Not idle<br>1 = Idle | 0 |
| IIS prescaler | [1] | 0 = Disable<br>1 = Enable | 0 |
| IIS interface | [0] | 0 = Disable (stop)<br>1 = Enable (start) | 0 |

**Notes:**

1. The IISCON register is accessible for each byte, halfword and word unit using STRB/STRH/STR and LDRB/LDRH/LDR instructions or char/short int/int type pointer in Little/Big endian mode.
2. (Li/HW/W) : Little/HalfWord/Word
   (Bi/HW/W) : Big/HalfWord/Word

### IIS MODE REGISTER (IISMOD) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IISMOD | 0x55000004 (Li/W, Li/HW, Bi/W)<br>0x55000006 (Bi/HW) | R/W | IIS mode register | 0x0 |

| IISMOD | Bit | Description | Initial State |
|---|---|---|---|
| Master/slave mode select | [8] | 0 = Master mode (IISLRCK and IISCLK are output mode).<br>1 = Slave mode (IISLRCK and IISCLK are input mode). | 0 |
| Transmit/receive mode select | [7:6] | 00 = No transfer 01 = Receive mode<br>10 = Transmit mode 11 = Transmit and receive mode | 00 |
| Active level of left/right channel | [5] | 0 = Low for left channel (High for right channel)<br>1 = High for left channel (Low for right channel) | 0 |
| Serial interface format | [4] | 0 = IIS compatible format<br>1 = MSB (Left)-justified format | 0 |
| Serial data bit per channel | [3] | 0 = 8-bit 1 = 16-bit | 0 |
| Master clock frequency select | [2] | 0 = 256fs 1 = 384fs<br>(fs : sampling frequency) | 0 |
| Serial bit clock frequency select | [1:0] | 00 = 16fs 01 = 32fs<br>10 = 48fs 11 = N/A | 00 |

**Notes:**

1. The IISMOD register is accessible for each halfword and wordunit using STRH/STR and LDRH/LDR instructions or short int/int type pointer in Little/Big endian mode.
2. (Li/HW/W) : Little/HalfWord/Word.
   (Bi/HW/W) : Big/HalfWord/Word.

### IIS PRESCALER (IISPSR) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IISPSR | 0x55000008 (Li/HW, Li/W, Bi/W)<br>0x5500000A (Bi/HW) | R/W | IIS prescaler register | 0x0 |

| IISPSR | Bit | Description | Initial State |
|---|---|---|---|
| Prescaler control A | [9:5] | Data value: 0 ~ 31<br>Note: Prescaler A makes the master clock that is used the internal block and division factor is N+1. | 00000 |
| Prescaler control B | [4:0] | Data value: 0 ~ 31<br>Note: Prescaler B makes the master clock that is used the external block and division factor is N+1. | 00000 |

**Notes:**

1. The IISPSR register is accessible for each byte, halfword and word unit using STRB/STRH/STR and LDRB/LDRH/LDR instructions or char/short int/int type pointer in Little/Big endian mode.
2. (Li/HW/W) : Little/HalfWord/Word.
   (Bi/HW/W) : Big/HalfWord/Word.

### IIS FIFO CONTROL (IISFCON) REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IISFCON | 0x5500000C (Li/HW, Li/W, Bi/W)<br>0x5500000E (Bi/HW) | R/W | IIS FIFO interface register | 0x0 |

| IISFCON | Bit | Description | Initial State |
|---|---|---|---|
| Transmit FIFO access mode select | [15] | 0 = Normal<br>1 = DMA | 0 |
| Receive FIFO access mode select | [14] | 0 = Normal<br>1 = DMA | 0 |
| Transmit FIFO | [13] | 0 = Disable 1 = Enable | 0 |
| Receive FIFO | [12] | 0 = Disable 1 = Enable | 0 |
| Transmit FIFO data count<br>(Read only) | [11:6] | Data count value = 0 ~ 32 | 000000 |
| Receive FIFO data count<br>(Read only) | [5:0] | Data count value = 0 ~ 32 | 000000 |

**NOTES:**

1. The IISFCON register is accessible for each halfword and word unit using STRH/STR and LDRH/LDR instructions or short int/int type pointer in Little/Big endian mode.
2. (Li/HW/W): Little/HalfWord/Word.
   (Bi/HW/W): Big/HalfWord/Word.

### IIS FIFO (IISFIFO) REGISTER

IIS bus interface contains two 16-byte FIFO for the transmit and receive mode. Each FIFO has 16-width and 24-depth form, which allows the FIFO to handles data for each halfword unit regardless of valid data size. Transmit and receive FIFO access is performed through FIFO entry; the address of FENTRY is 0x55000010.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| IISFIFO | 0x55000010(Li/HW)<br>0x55000012(Bi/HW) | R/W | IIS FIFO register | 0x0 |

| IISFIF | Bit | Description | Initial State |
|---|---|---|---|
| FENTRY | [15:0] | Transmit/Receive data for IIS | 0x0 |

**NOTES:**

1. The IISFIFO register is accessible for each halfword and word unit using STRH and LDRH instructions or short int type pointer in Little/Big endian mode.
2. (Li/HW): Little/HalfWord.
   (Bi/HW): Big/HalfWord.

# 22 SPI

## OVERVIEW

The S3C2440X Serial Peripheral Interface (SPI) can interface the serial data transfer. The S3C2440X includes two SPI, each of which has two 8-bit shift registers for transmission and receiving, respectively. During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). 8-bit serial data at a frequency is determined by its corresponding control register settings. If you only want to transmit, received data can be dummy. Otherwise, if you only want to receive, you should transmit dummy '1' data.

There are 4 I/O pin signals associated with SPI transfers: the SCK (SPICLK0,1), the MISO (SPIMISO0,1) data line, the MOSI (SPIMOSI0,1) data line, and the active low /SS (nSS0,1) pin (input).

### FEATURES

— Support 2-ch SPI
— SPI Protocol (ver. 2.11) compatible
— 8-bit Shift Register for transmit
— 8-bit Shift Register for receive
— 8-bit Prescaler logic
— Polling, Interrupt, and DMA transfer mode
— 5V tolerant input (SPI channel 1)

## BLOCK DIAGRAM

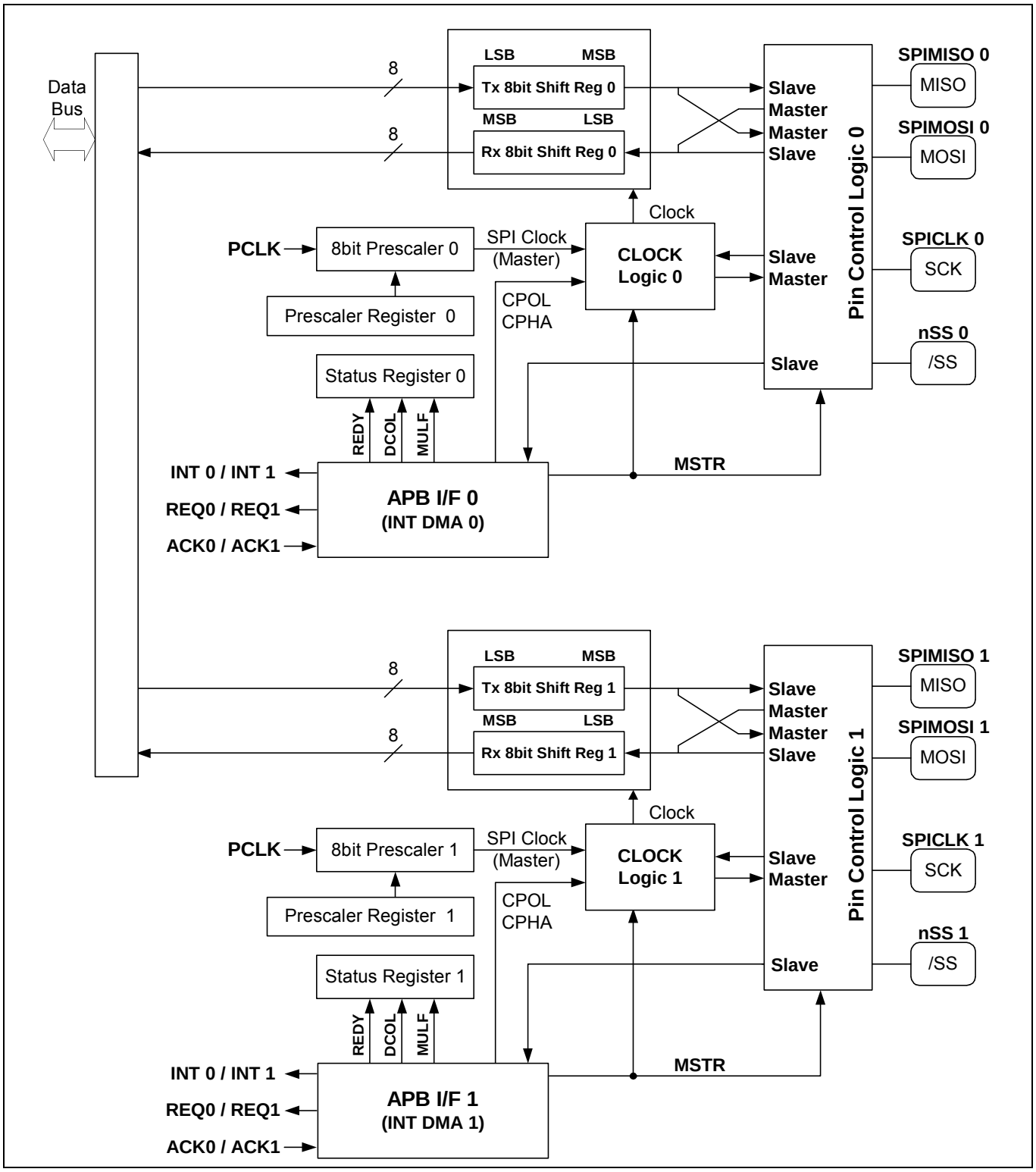



**Figure 22-1. SPI Block Diagram**

## SPI OPERATION

Using the SPI interface, the S3C2440X can send/receive 8 –bit data simultaneously with an external device. A serial clock line is synchronized with the two data lines for shifting and sampling of the information. When the SPI is the master, transmission frequency can be controlled by setting the appropriate bit to SPPREn register. You can modify its frequency to adjust the baud rate data register value. When the SPI is a slave, other master supplies the clock. When the programmer writes byte data to SPTDATn register, SPI transmit/receive operation will start simultaneously. In some cases, nSS should be activated before writing byte data to SPTDATn.

**Programming Procedure**

When a byte data is written into the SPTDATn register, SPI starts to transmit if ENSCK and MSTR of SPCONn register are set. You can use a typical programming procedure to operate an SPI card.

To program the SPI modules, follow these basic steps:

1. Set Baud Rate Prescaler Register (SPPREn).
2. Set SPCONn to configure properly the SPI module.
3. Write data 0xFF to SPTDATn 10 times in order to initialize MMC or SD card.
4. Set a GPIO pin, which acts as nSS, to low to activate the MMC or SD card.
5. Tx data → Check the status of Transfer Ready flag (REDY=1), and then write data to SPTDATn.
6. Rx data(1): SPCONn's TAGD bit disable = normal mode

→ write 0xFF to SPTDATn, then confirm REDY to set, and then read data from Read Buffer.

7. Rx data(2): SPCONn's TAGD bit enable = Tx Auto Garbage Data mode

→ confirm REDY to set, and then read data from Read Buffer(then automatically start to transfer).

8. Set a GPIO pin, which acts as nSS, to high, to deactivate MMC or SD card.

### SPI Transfer Format

The S3C2440X supports 4 different format to transfer the data. Figure 22-2 shows four waveforms for SPICLK..

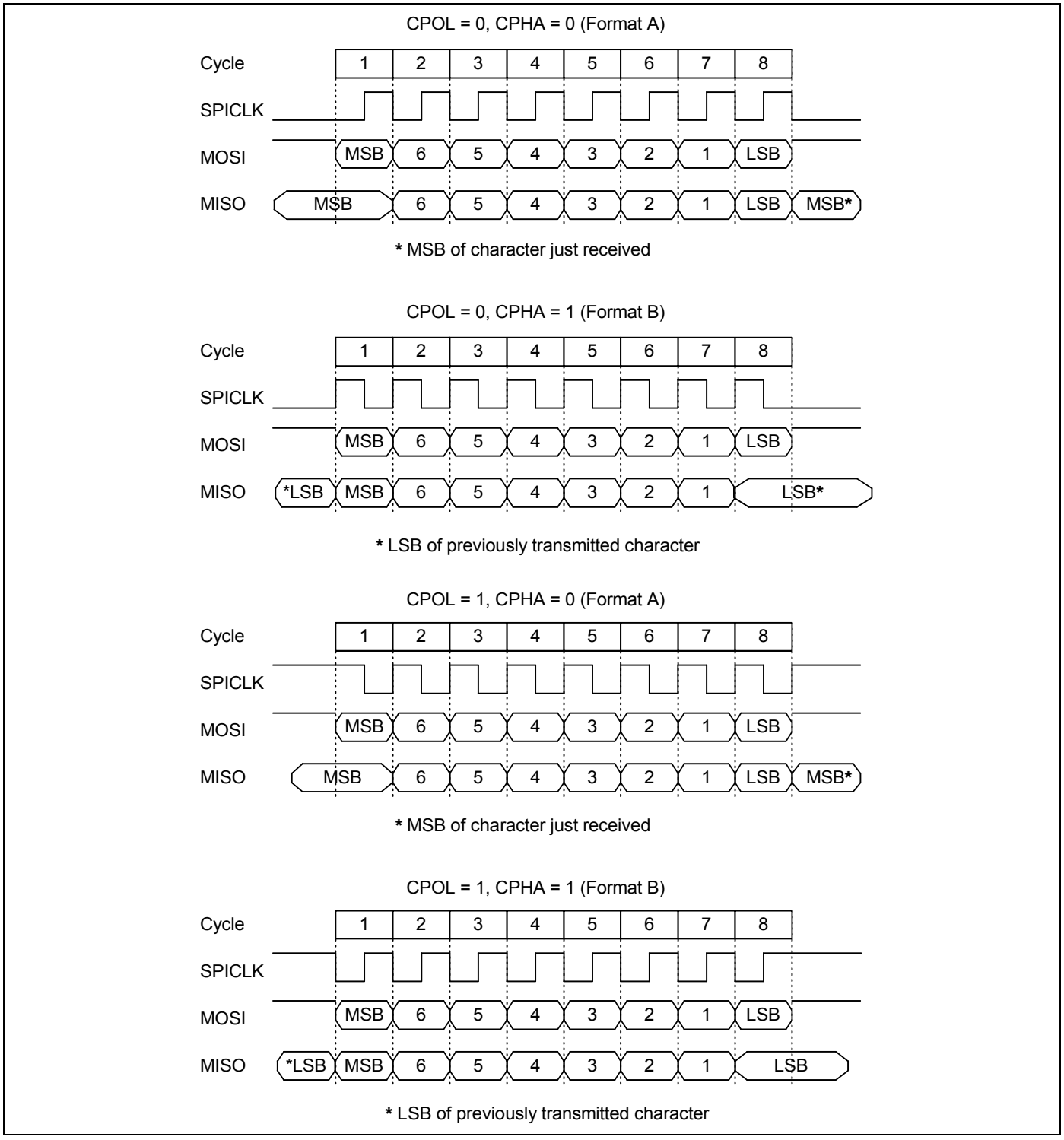



**Figure 22-2. SPI Transfer Format**

**Transmitting procedure by DMA**

1. The SPI is configured as DMA mode.
2. DMA is configured properly.
3. The SPI requests DMA service.
4. DMA transmits 1byte data to the SPI.
5. The SPI transmits the data to card.
6. Return to Step 3 until DMA count becomes 0.
7. The SPI is configured as interrupt or polling mode with SMOD bits.

**Receiving procedure by DMA**

1. The SPI is configured as DMA start with SMOD bits and setting TAGD bit.
2. DMA is configured properly.
3. The SPI receives 1byte data from card.
4. The SPI requests DMA service.
5. DMA receives the data from the SPI.
6. Write data 0xFF automatically to SPTDATn.
7. Return to Step 4 until DMA count becomes 0.
8. The SPI is configured as polling mode with SMOD bits and clearing TAGD bit.
9. If SPSTAn's REDY flag is set, then read the last byte data.

**Note:** Total received data = DMA TC values + the last data in polling mode (Step 9).
The first DMA received data is dummy and so the user can neglect it.

## SPI SPECIAL REGISTERS

### SPI CONTROL REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SPCON0 | 0x59000000 | R/W | SPI channel 0 control register | 0x00 |
| SPCON1 | 0x59000020 | R/W | SPI channel 1 control register | 0x00 |

| SPCONn | Bit | Description | Initial State |
|---|---|---|---|
| SPI Mode Select (SMOD) | [6:5] | Determine how and by what SPTDAT is read/written.<br>00 = polling mode 01 = interrupt mode<br>10 = DMA mode 11 = reserved | 00 |
| SCK Enable (ENSCK) | [4] | Determine whether you want SCK enable or not (for only master).<br>0 = disable 1 = enable | 0 |
| Master/Slave Select (MSTR) | [3] | Determine the desired mode (master or slave).<br>0 = slave 1 = master<br>**Note:** In slave mode, there should be set up time for master to initiate Tx/Rx. | 0 |
| Clock Polarity Select (CPOL) | [2] | Determine an active high or active low clock.<br>0 = active high 1 = active low | 0 |
| Clock Phase Select (CPHA) | [1] | Select one of two fundamentally different transfer formats.<br>0 = format A 1 = format B | 0 |
| Tx Auto Garbage Data mode enable (TAGD) | [0] | Decide whether the receiving data only needs or not.<br>0 = normal mode 1 = Tx auto garbage data mode<br>**Note:** In normal mode, if you only want to receive data, you should transmit dummy 0xFF data. | 0 |

### SPI STATUS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SPSTA0 | 0x59000004 | R | SPI channel 0 status register | 0x01 |
| SPSTA1 | 0x59000024 | R | SPI channel 1 status register | 0x01 |

| SPSTAn | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:3] | | |
| Data Collision Error Flag (DCOL) | [2] | This flag is set if the SPTDATn is written or the SPRDATn is read while a transfer is in progress and cleared by reading the SPSTAn.<br>0 = not detect 1 = collision error detect | 0 |
| Multi Master Error Flag (MULF) | [1] | This flag is set if the nSS signal goes to active low while the SPI is configured as a master, and SPPINn's ENMUL bit is multi master errors detect mode. MULF is cleared by reading SPSTAn.<br>0 = not detect 1 = multi master error detect | 0 |
| Transfer Ready Flag (REDY) | [0] | This bit indicates that SPTDATn or SPRDATn is ready to transmit or receive. This flag is automatically cleared by writing data to SPTDATn.<br>0 = not ready 1 = data Tx/Rx ready | 1 |

### SPI PIN CONTROL REGISTER

When the SPI system is enabled, the direction of pins, except nSS pin, is controlled by MSTR bit of SPCONn register. The direction of nSS pin is always input.
When the SPI is a master, nSS pin is used to check multi-master error, provided the SPPIN's ENMUL bit is active, and another GPIO should be used to select a slave.
If the SPI is configured as a slave, the nSS pin is used to select SPI as a slave by one master.

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SPPIN0 | 0x59000008 | R/W | SPI channel 0 pin control register | 0x02 |
| SPPIN1 | 0x59000028 | R/W | SPI channel 1 pin control register | 0x02 |

| SPPINn | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [7:3] | | |
| Multi Master error detect Enable (ENMUL) | [2] | The /SS pin is used as an input to detect multi master error when the SPI system is a master.<br>0 = disable (general purpose) 1 = multi master error detect enable | 0 |
| Reserved | [1] | This bit should be '1'. | 1 |
| Master Out Keep (KEEP) | [0] | Determine MOSI drive or release when 1byte transmit is completed (only master).<br>0 = release 1 = drive the previous level | 0 |

The SPIMISO (MISO) and SPIMOSI (MOSI) data pins are used for transmitting and receiving serial data. When the SPI is configured as a master, SPIMISO (MISO) is the master data input line, SPIMOSI (MOSI) is the master data output line, and SPICLK (SCK) is the clock output line. When the SPI becomes a slave, these pins perform reversed roles. In a multiple-master system, SPICLK (SCK) pins, SPIMOSI (MOSI) pins, and SPIMISO (MISO) pins are tied to configure a group respectively.

A master SPI can experience a multi master error, when other SPI device working as a master selects the S3C2410 SPI as a slave. When this error is detected, the following actions are taken immediately. But you must previously set SPPINn's ENMUL bit if you want to detect this error.

1. The SPCONn's MSTR bit is forced to 0 to operate slave mode.
2. The SPSTAn's MULF flag is set, and an SPI interrupt is generated.

### SPI Baud Rate Prescaler Register

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SPPRE0 | 0x5900000C | R/W | SPI cannel 0 baud rate prescaler register | 0x00 |
| SPPRE1 | 0x5900002C | R/W | SPI cannel 1 baud rate prescaler register | 0x00 |

| SPPREn | Bit | Description | Initial State |
|---|---|---|---|
| Prescaler Value | [7:0] | Determine SPI clock rate as above equation.<br>Baud rate = PCLK / 2 / (Prescaler value + 1) | 0x00 |

**Note**: Baud rate should be less than 25MHz.

### SPI Tx Data Register

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SPTDAT0 | 0x59000010 | R/W | SPI channel 0 Tx data register | 0x00 |
| SPTDAT1 | 0x59000030 | R/W | SPI channel 1 Tx data register | 0x00 |

| SPTDATn | Bit | Description | Initial State |
|---|---|---|---|
| Tx Data Register | [7:0] | This field contains the data to be transmitted over the SPI channel. | 0x00 |

### SPI Rx Data Register

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| SPRDAT0 | 0x59000014 | R | SPI channel 0 Rx data register | 0x00 |
| SPRDAT1 | 0x59000034 | R | SPI channel 1 Rx data register | 0x00 |

| SPRDATn | Bit | Description | Initial State |
|---|---|---|---|
| Rx Data Register | [7:0] | This field contains the data to be received over the SPI channel. | 0x00 |

# 23 CAMERA INTERFACE

## OVERVIEW

This specification defines the interface of camera. The camera interface within S3C2440X consists of three parts. The one is the logic of catching camera input signals. The second one is the logic of format conversion and down scaling. And, the last one is the dedicated DMA part.

The camera interface supports ITU BT.601/656 8-bit mode. The scaler of camera interface can scale down from below XGA(up to horizontal 1016 pixels) input image into SVGA, VGA, QVGA, CIF, QCIF and any other smaller sizes. Two master ports can be used variable applications like DSC, JPEG, MPEG self image and so on.

Camera interface can generate self test patterns as color bar, square box. It could be used in the calibration of image sync signals. Also, video sync signals and pixel clock polarity can be inverted in camera interface by register setting.

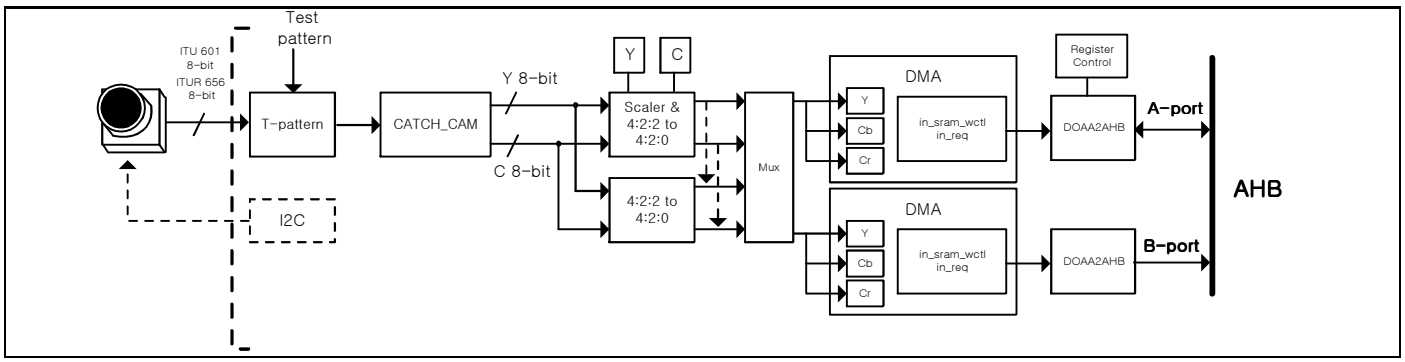



**Figure 23-1. Camera interface overview**

### FEATURES

- Supports ITU-R BT.601/656 (4:2:2 YcbCr 8-bit mode).
- Image down scaling capability for variable applications.
- Two master port for dedicated DMA operation.
- Programmable the polarity of video sync signals.
- Wide horizontal line buffer (Maximum 1016 pixels).
- Format conversion from YCbCr 4:2:2 to YCbCr 4:2:0
- Programmable burst length for DMA operation.

## EXTERNAL INTERFACE

Camera interface in S3C2440X can support next ITU video standard.

— ITU BT.601 YCbCr 8-bit mode

— ITU BT.656 YCbCr 8-bit mode

### SIGNAL DESCRIPTION

| Name | I/O[1)] | Active | Description |
|---|---|---|---|
| PCLK | I | - | Pixel Clock, driven by the Camera processor |
| VSYNC | I | H | Vertical Sync, driven by the Camera processor |
| HREF | I | H | Horizontal Sync, driven by the Camera processor |
| DATA[7:0] | I | - | Pixel Data for YCbCr, driven by the Camera processor |
| CAMCLK | O | - | Master Clock to the Camera processor |
| CAMRST | O | H | Software Reset to the Camera processor |

Note 1) I/O direction is on the AP side. I : input, O : output, B : bi-direction

**Table 23-1. Camera interface signal description**

### TIMING DIAGRAM

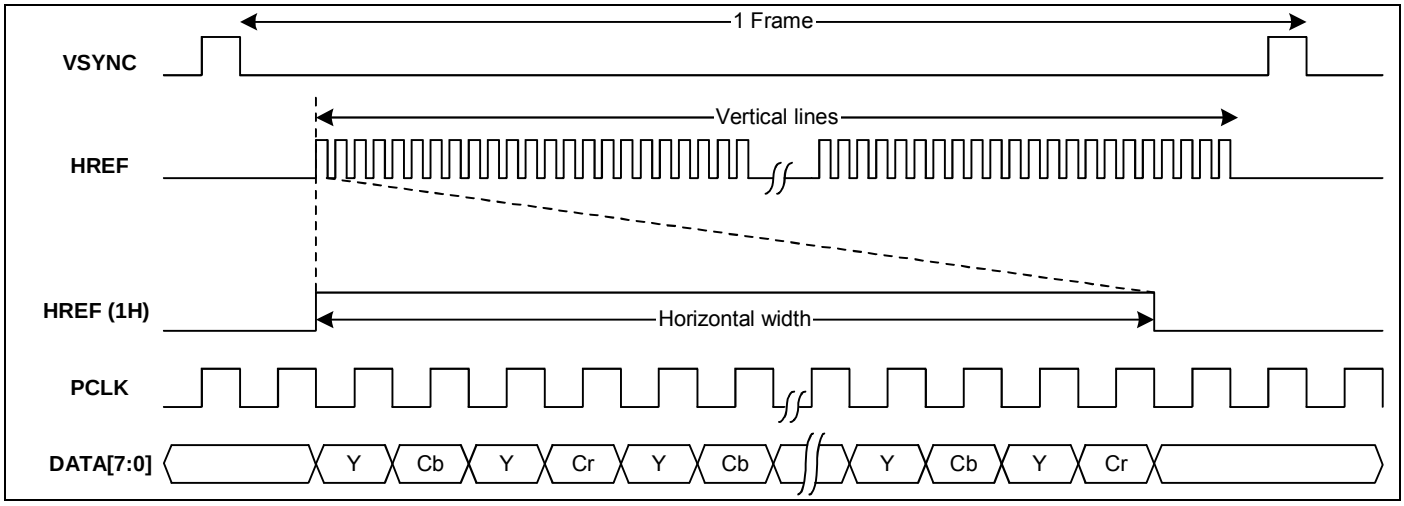



**Figure 23-2. ITU-R BT.601 Input timing diagram**

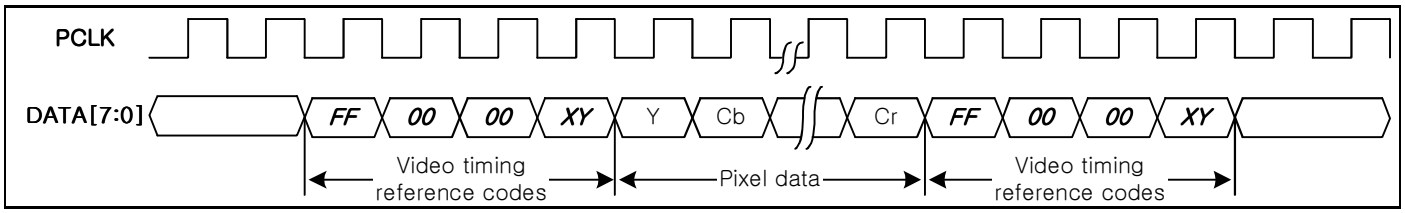



**Figure 23-3. ITU-R BT.656 Input timing diagram**

There are two timing reference signals in ITU-R BT.656 format, one at the beginning of each vedio data block (start of active video, SAV) and one at the end of each video data block(end of active video, EAV) as shown in Figure 23-3 and Table 23-2.

| Data bit number | First word (FF) | Second word (00) | Third word (00) | Forth word (XY) |
|---|---|---|---|---|
| 9 (MSB) | 1 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | F |
| 7 | 1 | 0 | 0 | V |
| 6 | 1 | 0 | 0 | H |
| 5 | 1 | 0 | 0 | P3 |
| 4 | 1 | 0 | 0 | P2 |
| 3 | 1 | 0 | 0 | P1 |
| 2 | 1 | 0 | 0 | P0 |
| 1 (Note 1) | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |

Note 1) For compatibility with existing 8-bit interfaces, the values of bits D1 and D0 are not defined.

F = 0 (during field 1), 1 (during field 2)

V = 0 (elsewhere), 1 (during field blanking)

H = 0 (in SAV : Start of Active Video), 1 (in EAV : End of Active Video)

P0, P1, P2, P3 = protection bit

**Table 23-2. Video timing reference codes of ITU-R BT.656 format**

Camera interface logic can catch the video sync bits like H(SAV,EAV) and V(Frame Sync) after reserved data as “FF-00-00”.

## CAMERA INTERFACE OPERATION

### TWO DMA PORTS

CAMIF has two DMA port. A-port and B-port are separated each other. At view of system bus, two ports are independent. The A-port stores the scale-downed and format-converted image to A-port ping-pong memories. The B-port stores the only format-converted image to B-port ping-pong memories. These two master ports enable variable application like DSC (Digital Steel Camera), MPEG self image, etc. For example, A-port image can be used as preview image, and B-port image can be used as JPEG image in DSC application. Also, A-port or B-port can be separately disabled.

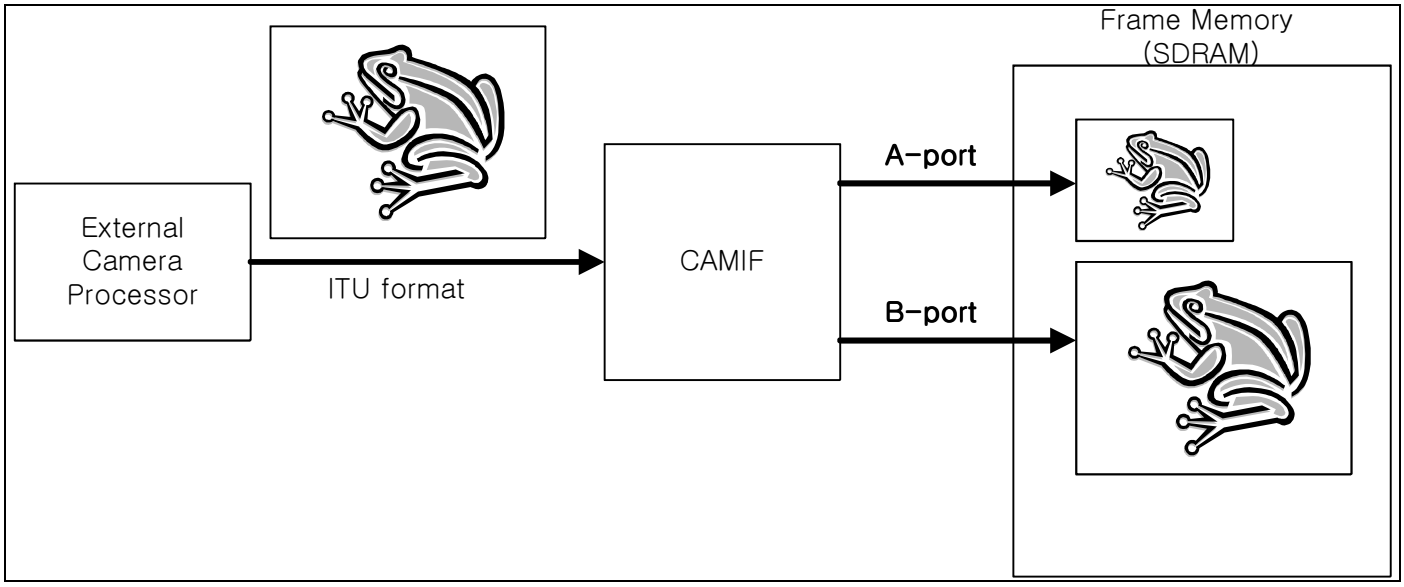



**Figure 23-4. Two DMA ports**

## CLOCK DOMAIN

CAMIF has three clock domains. The one is the system bus clock, which is HCLK. Another is the pixel clock, which is PCLK. And, the other is the camera interface operation clock, which is OP_CLOCK. The system clock must be faster than any other clock. And, the pixel clock must be double speed compared to the operation clock of camera interface. As shown in figure 1-6, CAMCLK must be divided from the fixed frequency like USB PLL clock. OP_CLOCK has the half frequency of CAMCLK.

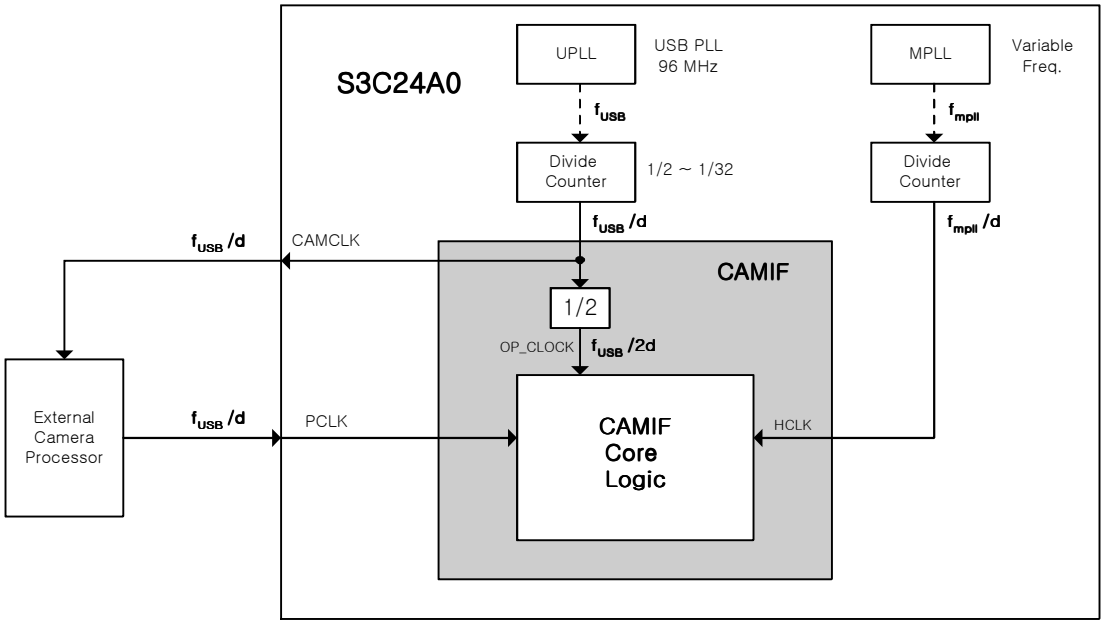



**Figure 23-5. CAMIF clock generation**

## FRAME MEMORY HIRERARCHY

Frame memories consist of 4 ping-pong memories for each A- and B-ports. Each ping-pong memory has three elements memory that is luminance, chrominance Cb, and chrominance Cr. After getting the input of ITU format, CAMIF transfers output data to AHB-bus for memory access. It is recommended that the arbitration priority of CAMIF must be higher than other master except display controller. If AHB-bus is traffic enough that DMA operation is not ending during one horizontal sync signal, it will enter into mal-function. So, the priority of CAMIF must be separated to other round-robin priorities.

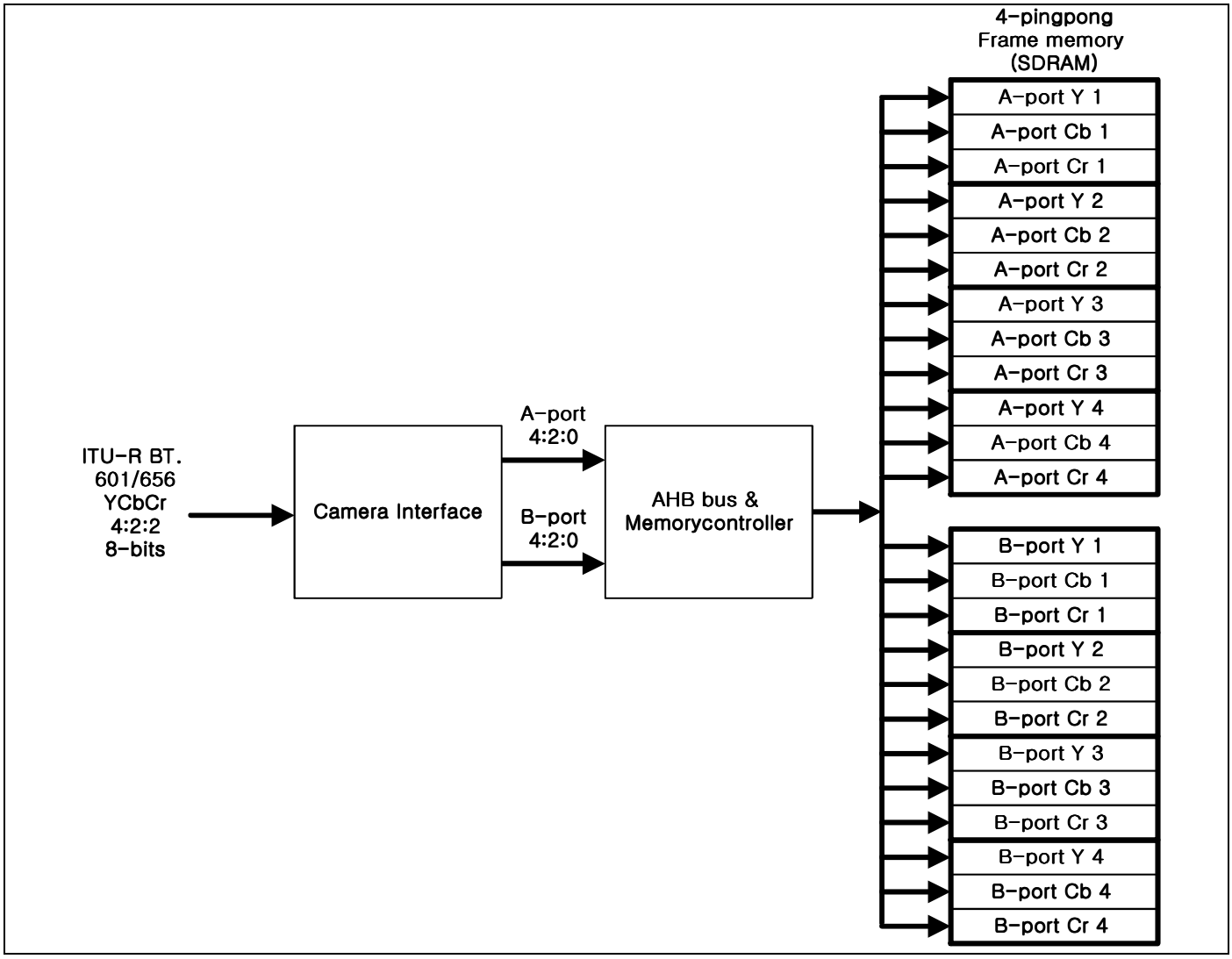



**Figure 23-6. Ping-pong memory hierarchy**

## MEMORY STORING METHOD

The storing method to the frame memories is the little-endian method. The first entering pixels stored into LSB sides, and the last entering pixels stored into MSB sides. The carried data by AHB bus is 32-bit word. So, CAMIF make the each Y-Cb-Cr words by little endian style. Rightly, the carried data is the format-converted or scaled and format-converted data.

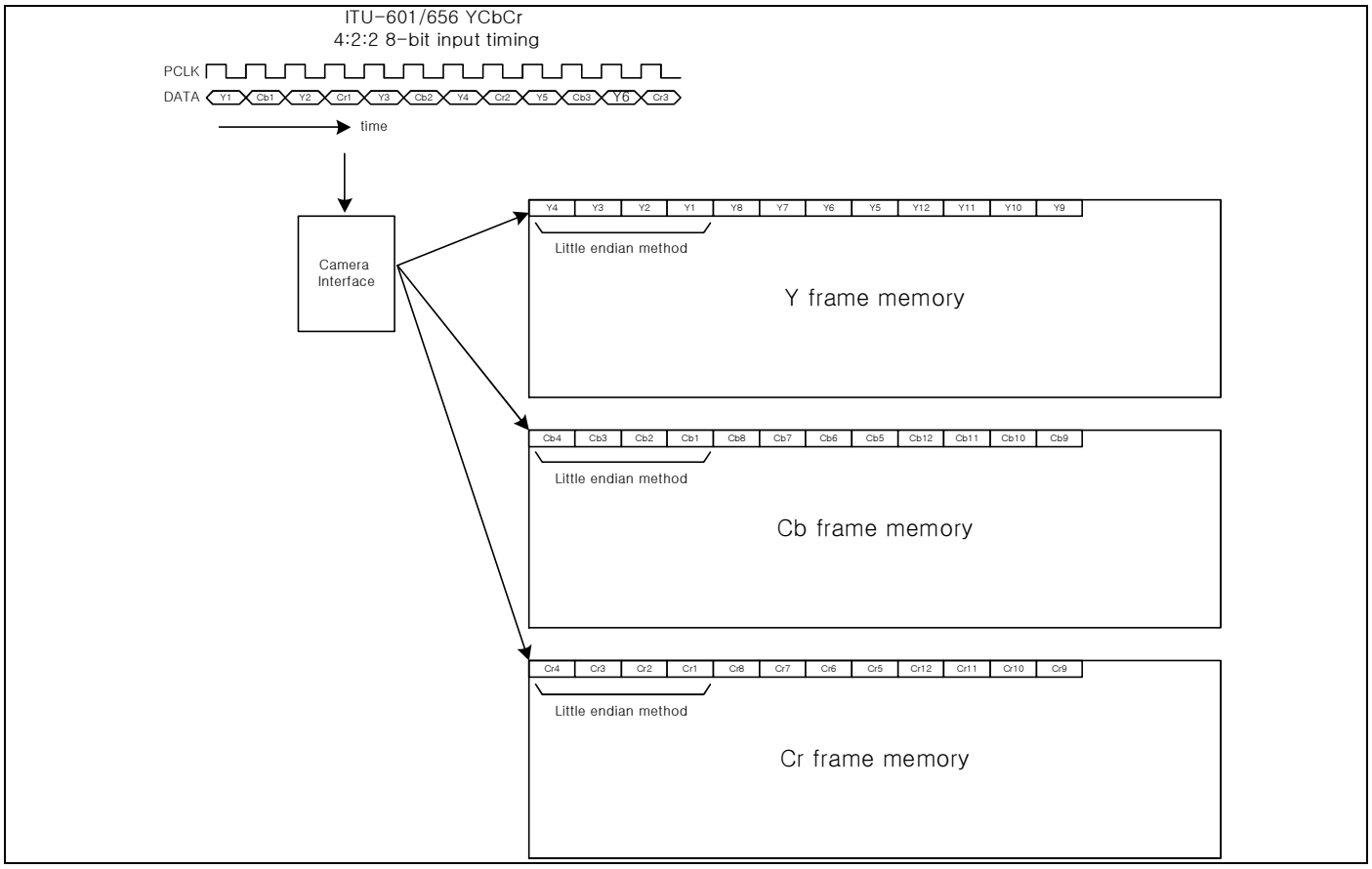



**Figure 23-7. Memory storing style**

## TIMING DIAGRAM FOR REGISTER SETTING

The first register setting for frame capture command can be occurred anytime in operation. For capturing fully occupied frame, the next frame must be captured. And, for another capture, you can program the second register setting in interrupt service routine. Be sure that interrupt is flagged on the start of the captured frame period. One pulse interrupt signal can be programmed by IRQFREE of CTRL register as “1”.

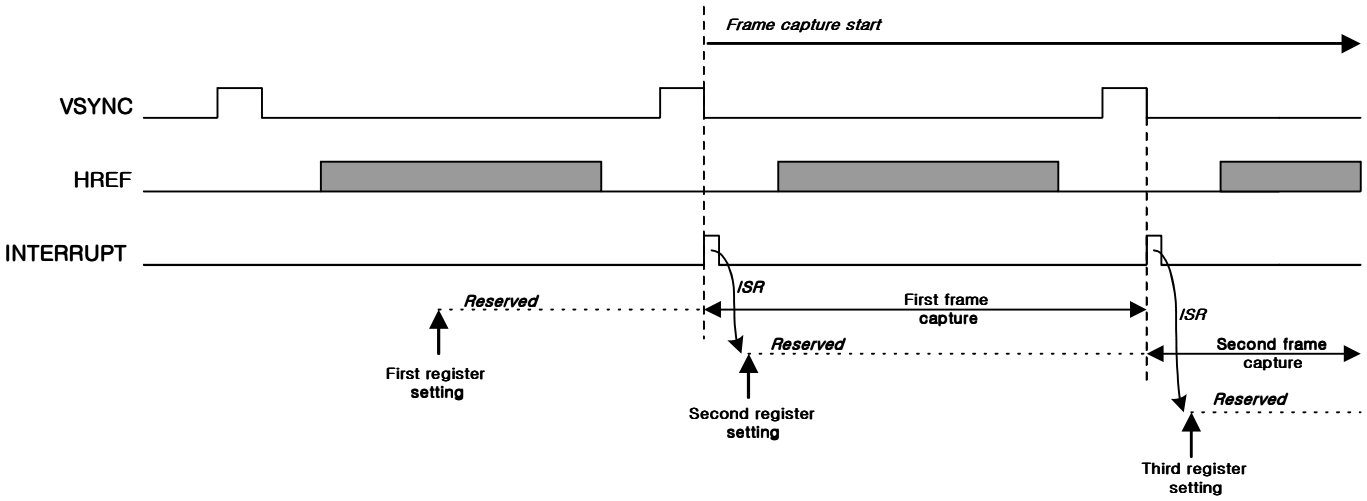



**Figure 23-8. Timing diagram for register setting**

# SOFTWARE INTERFACE

This Camera interface provides a generic data-exchange method. But, you should keep in mind that write registers and read registers are separated. A-port sends scaled output to the AHB bus as in Figure 23-1. B-port sends only format conversion or scaled output to the AHB bus. There is 4 Ping-Pong memories. Consequently, 4 sets of Y-Cb-Cr addresses exist.

## CAMERA INTERFACE SPECIAL REGISTERS

## 1. WRITE REGISTER

### A IMAGE SIZE REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ASIZE | 0x4F000000 | W | A-port Image Size | 0xc8258 |

**NOTE) The Base address is TBD**

| ASIZE | Bit | Description | Initial State |
|---|---|---|---|
| AHSIZE | [19:10] | These bits indicate the horizontal pixel number of the target image for A-port. If target image size is QCIF, these bits should be 10'd176, that is 10'hb0. | 0x320 |
| AVSIZE | [9:0] | These bits indicate the vertical pixel number of the target image for A-port. If target image size is QCIF, these bits should be 10'd144, that is 10'h90. | 0x258 |

### A Y1 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STAY1 | 0x4F000004 | R/W | A-port Image 1st ping-pong memory Y start address | 0xc073f000 |

| STAY1 | Bit | Description | Initial State |
|---|---|---|---|
| STAY1 | [31:0] | This register value will be the start address of luminance for 1st Ping-Pong memory of A-port | 0xc073f000 |

### A Y2 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STAY2 | 0x4F000008 | R/W | A-port Image 2nd ping-pong memory Y start address | 0xc07eec80 |

| STAY2 | Bit | Description | Initial State |
|---|---|---|---|
| STAY2 | [31:0] | This register value will be the start address of luminance for 2nd Ping-Pong memory of A-port | 0xc07eec80 |

### A Y3 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STAY3 | 0x4F00000C | R/W | A-port Image 3rd ping-pong memory Y start address | 0xc089e900 |

| STAY3 | Bit | Description | Initial State |
|---|---|---|---|
| STAY3 | [31:0] | This register value will be the start address of luminance for 3rd Ping-Pong memory of A-port | 0xc089e900 |

### A Y4 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STAY4 | 0x4F000010 | R/W | A-port Image 4th ping-pong memory Y start address | 0xc094e580 |

| STAY4 | Bit | Description | Initial State |
|---|---|---|---|
| STAY4 | [31:0] | This register value will be the start address of luminance for 4th Ping-Pong memory of A-port | 0xc094e580 |

### A Y BURST REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| AYBURST | 0x4F000014 | W | A-port Image Y data burst length | 0x00000000 |

| AYBURST | Bit | Description | Initial State |
|---|---|---|---|
| AYBURST1 | [31:16] | These bits indicate main burst length of A-port during the memory read/write access for luminance data. | 0x0000 |
| AYBURST2 | [15:0] | These bits indicate remained burst length of A-port during the memory read/write access for luminance data. | 0x0000 |

* In CIF case, AYBURST1=8, AYBURST2=8 are recommended.

Example 1. Target image size : QCIF (horizontal Y width = 176 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

176 / 4 = 44 word.

44 % 8 = 4 → main burst = 8, remained burst = 4

Example 2. Target image size : VGA (horizontal Y width = 640 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

640 / 4 = 160 word.

160 % 16 = 0 → main burst = 16, remained burst = 16

### A CB BURST REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ACBBURST | 0x4F000018 | W | A-port Image Cb data burst length | 0x00000000 |

| ACBBURST | Bit | Description | Initial State |
|---|---|---|---|
| ACBBURST1 | [31:16] | These bits indicate main burst length of A-port during the memory read/write access for Chrominance Cb data. | 0x0000 |
| ACBBURST2 | [15:0] | These bits indicate remained burst length of A-port during the memory read/write access for Chrominance Cb data. | 0x0000 |

* In CIF case, ACBBURST1=4, ACBBURST2=4 are recommended.

Example 1. Target image size : QCIF (horizontal C width = 88 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

88 / 4 = 22 word.

22 % 4 = 2 → main burst = 4, remained burst = 2 (internally two single operations)

Example 2. Target image size : VGA (horizontal C width = 320 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

320 / 4 = 80 word.

80 % 16 = 0 → main burst = 16, remained burst = 16

### A CR BURST REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ACRBURST | 0x4F00001C | W | A-port Image Cr data burst length | 0x00000000 |

| ACRBURST | Bit | Description | Initial State |
|---|---|---|---|
| ACRBURST1 | [31:16] | These bits indicate main burst length of A-port during the memory read/write access for Chrominance Cr data. | 0x0000 |
| ACRBURST2 | [15:0] | These bits indicate remained burst length of A-port during the memory read/write access for Chrominance Cr data. | 0x0000 |

* In CIF case, ACRBURST1=4, ACRBURST2=4 are recommended.

Example 1. Target image size : QCIF (horizontal C width = 88 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

88 / 4 = 22 word.

22 % 4 = 2 → main burst = 4, remained burst = 2 (internally two single operations)

Example 2. Target image size : VGA (horizontal C width = 320 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

320 / 4 = 80 word.

80 % 16 = 0 → main burst = 16, remained burst = 16

### B IMAGE SIZE REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BSIZE | 0x4F000020 | W | B-port Image Size | 0xc8258 |

| ASIZE | Bit | Description | Initial State |
|---|---|---|---|
| BHSIZE | [19:10] | These bits indicate the horizontal pixel number of the target image for B-port. If target image size is QCIF, these bits should be 10'd176, that is 10'hb0. | 0x320 |
| BVSIZE | [9:0] | These bits indicate the vertical pixel number of the target image for B-port. If target image size is QCIF, these bits should be 10'd144, that is 10'h90. | 0x258 |

### B Y1 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBY1 | 0x4F000024 | W | B-port Image 1st ping-pong memory Y start address | 0xc073f000 |

| STBY1 | Bit | Description | Initial State |
|---|---|---|---|
| STBY1 | [31:0] | This register value will be the start address of luminance for 1st Ping-Pong memory of B-port | 0xc073f000 |

### B Y2 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBY2 | 0x4F000028 | W | B-port Image 2nd ping-pong memory Y start address | 0xc07eec80 |

| STBY2 | Bit | Description | Initial State |
|---|---|---|---|
| STBY2 | [31:0] | This register value will be the start address of luminance for 2nd Ping-Pong memory of B-port | 0xc07eec80 |

### B Y3 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBY3 | 0x4F00002C | W | B-port Image 3rd ping-pong memory Y start address | 0xc089e900 |

| STBY3 | Bit | Description | Initial State |
|---|---|---|---|
| STBY3 | [31:0] | This register value will be the start address of luminance for 3rd Ping-Pong memory of B-port | 0xc089e900 |

### B Y4 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBY4 | 0x4F000030 | W | B-port Image 4th ping-pong memory Y start address | 0xc094e580 |

| STBY4 | Bit | Description | Initial State |
|---|---|---|---|
| STBY4 | [31:0] | This register value will be the start address of luminance for 4th Ping-Pong memory of B-port | 0xc094e580 |

### B Y BURST REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BYBURST | 0x4F000034 | W | B-port Image Y data burst length | 0x00000000 |

| BYBURST | Bit | Description | Initial State |
|---|---|---|---|
| BYBURST1 | [31:16] | These bits indicate main burst length of B-port during the memory read/write access for luminance data. | 0x0000 |
| BYBURST2 | [15:0] | These bits indicate remained burst length of B-port during the memory read/write access for luminance data. | 0x0000 |

* In CIF case, BYBURST1=8, BYBURST2=8 are recommended.

Example 1. Target image size : QCIF (horizontal Y width = 176 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

176 / 4 = 44 word.

44 % 8 = 4 → main burst = 8, remained burst = 4

Example 2. Target image size : VGA (horizontal Y width = 640 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

640 / 4 = 160 word.

160 % 16 = 0 → main burst = 16, remained burst = 16

### B CB BURST REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCBBURST | 0x4F000038 | W | B-port Image Cb data burst length | 0x00000000 |

| BCBBURST | Bit | Description | Initial State |
|---|---|---|---|
| BCBBURST1 | [31:16] | These bits indicate main burst length of B-port during the memory read/write access for Chrominance Cb data. | 0x0000 |
| BCBBURST2 | [15:0] | These bits indicate remained burst length of B-port during the memory read/write access for Chrominance Cb data. | 0x0000 |

* In CIF case, BCBBURST1=4, BCBBURST2=4 are recommended.

Example 1. Target image size : QCIF (horizontal C width = 88 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

88 / 4 = 22 word.

22 % 4 = 2 → main burst = 4, remained burst = 2 (internally two single operations)

Example 2. Target image size : VGA (horizontal C width = 320 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

320 / 4 = 80 word.

80 % 16 = 0 → main burst = 16, remained burst = 16

### B CR BURST REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BCRBURST | 0x4F00003C | W | B-port Image Cr data burst length | 0x00000000 |

| BCRBURST | Bit | Description | Initial State |
|---|---|---|---|
| BCRBURST1 | [31:16] | These bits indicate main burst length of B-port during the memory read/write access for Chrominance Cr data. | 0x0000 |
| BCRBURST2 | [15:0] | These bits indicate remained burst length of B-port during the memory read/write access for Chrominance Cr data. | 0x0000 |

* In CIF case, BCRBURST1=4, BCRBURST2=4 are recommended.

Example 1. Target image size : QCIF (horizontal C width = 88 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

88 / 4 = 22 word.

22 % 4 = 2 → main burst = 4, remained burst = 2 (internally two single operations)

Example 2. Target image size : VGA (horizontal C width = 320 pixels. 1 pixel = 1 Byte. 1 word = 4 pixel)

320 / 4 = 80 word.

80 % 16 = 0 → main burst = 16, remained burst = 16

### A LAST HREF DISTANCE WIDTH REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| ADISTWIDTH | 0x4F000040 | W | A-port distance and width for last hsync generation | 0x1b581040 |

| ADISTWIDTH | Bit | Description | Initial State |
|---|---|---|---|
| ADIST | [31:16] | These bits indicate the distance from the start of external last HREF to the start of imitative HREF for last DMA operation for A-port.<br>ADIST = (Tpclk/Tsys) x 1.5 x H x 2<br>(Tpclk : the period of incoming pixel clock,<br>Tsys : the period of system clock,<br>H : the horizontal pixel number of source image)<br>If you do not view the last horizontal line in your display, you should increase the number of ADIST by sufficient values. If it is not effective, contact the developer in samsung, please. | 0x1b58 |
| AWIDTH | [15:0] | These bits indicate the width of imitative HREF for last DMA operation for A-port.<br>AWIDTH = (Tpclk/Tsys) x 1.5 x H x 2<br>(Tpclk : the period of incoming pixel clock,<br>Tsys : the period of system clock,<br>H : the horizontal pixel number of source image)<br>If you do not view the last horizontal line in your display, you should increase the number of ADIST by sufficient values. If it is not effective, contact the developer in samsung, please. | 0x1040 |

Example 1. Source image size : VGA (640 x 480), Tpclk : 74ns (13.5MHz), Tsys : 10ns (100MHz)

→ ADIST = (74/10) x 1.5 x 640 x 2 = 14208

→ AWIDTH = (74/10) x 1.5 x 640 x 2 = 14208

### B LAST HREF DISTANCE WIDTH REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| BDISTWIDTH | 0x4F000044 | W | B-port distance and width for last hsync generation | 0x1b581040 |

| ADISTWIDTH | Bit | Description | Initial State |
|---|---|---|---|
| BDIST | [31:16] | These bits indicate the distance from the start of external last HREF to the start of imitative HREF for last DMA operation for B-port.<br>BDIST = (Tpclk/Tsys) x 1.5 x H x 2<br>(Tpclk : the period of incoming pixel clock,<br>Tsys : the period of system clock,<br>H : the horizontal pixel number of source image)<br>If you do not view the last horizontal line in your display, you should increase the number of BDIST by sufficient values. If it is not effective, contact the developer in samsung, please. | 0x1b58 |
| BWIDTH | [15:0] | These bits indicate the width of imitative HREF for last DMA operation for B-port.<br>BWIDTH = (Tpclk/Tsys) x 1.5 x H x 2<br>(Tpclk : the period of incoming pixel clock,<br>Tsys : the period of system clock,<br>H : the horizontal pixel number of source image)<br>If you do not view the last horizontal line in your display, you should increase the number of BDIST by sufficient values. If it is not effective, contact the developer in samsung, please. | 0x1040 |

Example 1. Source image size : VGA (640 x 480), Tpclk : 74ns (13.5MHz), Tsys : 10ns (100MHz)

→ BDIST = (74/10) x 1.5 x 640 x 2 = 14208

→ BWIDTH = (74/10) x 1.5 x 640 x 2 = 14208

### Y SCALE RATIO REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| YRATIO | 0x4F00004C | W | A-port scaling ratio for luminace | 0x14001400 |

| YRATIO | Bit | Description | Initial State |
|---|---|---|---|
| YHRATIO | [31:16] | (Source Y size / Target Y size) x 4096 | 0x1400 |
| YVRATIO | [15:0] | (Source Y size / Target Y size) x 4096 | 0x1400 |

Example 1. Source image Y size : VGA (640 x 480), Target image Y size for A-port : QCIF (176 x 144)

→ YHRATIO = (640 / 176) x 4096 = 14894.545 ⇒ 14894 (in round numbers) : 0x3a2e

→ YVRATIO = (480 / 144) x 4096 = 13653.333 ⇒ 13653 (in round numbers) : 0x3555

### C SCALE RATIO REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| CRATIO | 0x4F000050 | W | A-port scaling ratio for chrominace | 0x14002800 |

| CRATIO | Bit | Description | Initial State |
|---|---|---|---|
| CHRATIO | [31:16] | (Source C size / Target C size) x 4096 | 0x1400 |
| CVRATIO | [15:0] | (Source C size / Target C size) x 4096 | 0x2800 |

Example 1. Source image C size : VGA (320 x 480), Target image C size for A-port : QCIF (88 x 72)

→ CHRATIO = (320 / 88) x 4096 = 14894.545 ⇒ 14894 (in round numbers) : 0x3a2e

→ CVRATIO = (480 / 72) x 4096 = 27306.666 ⇒ 27306 (in round numbers) : 0x6aaa

### Y ORIGINAL SIZE REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| YORIGINAL | 0x4F000054 | W | Y original size | 0xc8258 |

| YORIGINAL | Bit | Description | Initial State |
|---|---|---|---|
| YORIGINALX | [25:16] | These bits indicate the Y horizontal size of source image.<br>If input size is VGA (640 x 480), YORIGINALX should be 640. | 0x320 |
| YORIGINALY | [9:0] | These bits indicate the Y vertical size of source image.<br>If input size is VGA (640 x 480), YORIGINALY should be 480. | 0x258 |

### C ORIGINAL SIZE REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| CORIGINAL | 0x4F00005C | W | C original size | 0x64258 |

| YORIGINAL | Bit | Description | Initial State |
|---|---|---|---|
| CORIGINALX | [25:16] | These bits indicate the C horizontal size of source image.<br>If input size is VGA (640 x 480), CORIGINALX should be 320. | 0x190 |
| CORIGINALY | [9:0] | These bits indicate the C vertical size of source image.<br>If input size is VGA (640 x 480), CORIGINALY should be 480. | 0x258 |

### A CB1 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STACB1 | 0x4F000074 | W | A-port Image 1st ping-pong memory Cb start address | 0xc073f000 |

| STACB1 | Bit | Description | Initial State |
|---|---|---|---|
| STACB1 | [31:0] | This register value will be the start address of Chrominace Cb for 1st Ping-Pong memory of A-port | 0xc073f000 |

### A CB2 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STACB2 | 0x4F000078 | W | A-port Image 2nd ping-pong memory Cb start address | 0xc07eec80 |

| STACB2 | Bit | Description | Initial State |
|---|---|---|---|
| STACB2 | [31:0] | This register value will be the start address of Chrominace Cb for 2nd Ping-Pong memory of A-port | 0xc07eec80 |

### A CB3 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STACB3 | 0x4F00007C | W | A-port Image 3rd ping-pong memory Cb start address | 0xc089e900 |

| STACB3 | Bit | Description | Initial State |
|---|---|---|---|
| STACB3 | [31:0] | This register value will be the start address of Chrominace Cb for 3rd Ping-Pong memory of A-port | 0xc089e900 |

### A CB4 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STACB4 | 0x4F000080 | W | A-port Image 4th ping-pong memory Cb start address | 0xc094e580 |

| STACB4 | Bit | Description | Initial State |
|---|---|---|---|
| STACB4 | [31:0] | This register value will be the start address of Chrominace Cb for 4th Ping-Pong memory of A-port | 0xc094e580 |

### A CR1 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STACR1 | 0x4F000084 | W | A-port Image 1st ping-pong memory Cr start address | 0xc073f000 |

| STACR1 | Bit | Description | Initial State |
|---|---|---|---|
| STACR1 | [31:0] | This register value will be the start address of Chrominace Cr for 1st Ping-Pong memory of A-port | 0xc073f000 |

### A CR2 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STACR2 | 0x4F000088 | W | A-port Image 2nd ping-pong memory Cr start address | 0xc07eec80 |

| STACR2 | Bit | Description | Initial State |
|---|---|---|---|
| STACR2 | [31:0] | This register value will be the start address of Chrominace Cr for 2nd Ping-Pong memory of A-port | 0xc07eec80 |

### A CR3 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STACR3 | 0x4F00008C | W | A-port Image 3rd ping-pong memory Cr start address | 0xc089e900 |

| STACR3 | Bit | Description | Initial State |
|---|---|---|---|
| STACR3 | [31:0] | This register value will be the start address of Chrominace Cr for 3rd Ping-Pong memory of A-port | 0xc089e900 |

### A CR4 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STACR4 | 0x4F000090 | W | A-port Image 4th ping-pong memory Cr start address | 0xc094e580 |

| STACR4 | Bit | Description | Initial State |
|---|---|---|---|
| STACR4 | [31:0] | This register value will be the start address of Chrominace Cr for 4th Ping-Pong memory of A-port | 0xc094e580 |

### B CB1 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBCB1 | 0x4F00009C | W | B-port Image 1st ping-pong memory Cb start address | 0xc073f000 |

| STBCB1 | Bit | Description | Initial State |
|---|---|---|---|
| STBCB1 | [31:0] | This register value will be the start address of Chrominace Cb for 1st Ping-Pong memory of B-port | 0xc073f000 |

### B CB2 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBCB2 | 0x4F0000A0 | W | B-port Image 2nd ping-pong memory Cb start address | 0xc07eec80 |

| STBCB2 | Bit | Description | Initial State |
|---|---|---|---|
| STBCB2 | [31:0] | This register value will be the start address of Chrominace Cb for 2nd Ping-Pong memory of B-port | 0xc07eec80 |

### B CB3 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBCB3 | 0x4F0000A4 | W | B-port Image 3rd ping-pong memory Cb start address | 0xc089e900 |

| STBCB3 | Bit | Description | Initial State |
|---|---|---|---|
| STBCB3 | [31:0] | This register value will be the start address of Chrominace Cb for 3rd Ping-Pong memory of B-port | 0xc089e900 |

### B CB4 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBCB4 | 0x4F0000A8 | W | B-port Image 4th ping-pong memory Cb start address | 0xc094e580 |

| STBCB4 | Bit | Description | Initial State |
|---|---|---|---|
| STBCB4 | [31:0] | This register value will be the start address of Chrominace Cb for 4th Ping-Pong memory of B-port | 0xc094e580 |

### B CR1 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBCR1 | 0x4F0000AC | W | B-port Image 1st ping-pong memory Cr start address | 0xc073f000 |

| STBCR1 | Bit | Description | Initial State |
|---|---|---|---|
| STBCR1 | [31:0] | This register value will be the start address of Chrominace Cr for 1st Ping-Pong memory of B-port | 0xc073f000 |

### B CR2 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBCR2 | 0x4F0000B0 | W | B-port Image 2nd ping-pong memory Cr start address | 0xc07eec80 |

| STBCR2 | Bit | Description | Initial State |
|---|---|---|---|
| STBCR2 | [31:0] | This register value will be the start address of Chrominace Cr for 2nd Ping-Pong memory of B-port | 0xc07eec80 |

### B CR3 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBCR3 | 0x4F0000B4 | W | B-port Image 3rd ping-pong memory Cr start address | 0xc089e900 |

| STBCR3 | Bit | Description | Initial State |
|---|---|---|---|
| STBCR3 | [31:0] | This register value will be the start address of Chrominace Cr for 3rd Ping-Pong memory of B-port | 0xc089e900 |

### B CR4 START ADDRESS REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| STBCR4 | 0x4F0000B8 | W | B-port Image 4th ping-pong memory Cr start address | 0xc094e580 |

| STBCR4 | Bit | Description | Initial State |
|---|---|---|---|
| STBCR4 | [31:0] | This register value will be the start address of Chrominace Cr for 4th Ping-Pong memory of B-port | 0xc094e580 |

## CONTROL REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| CTRL | 0x4F0000BC | W | Camera interface control | 0x00000000 |

| CTRL | Bit | Description | Initial State |
|---|---|---|---|
| SOFTRST | [30] | This bit indicates software reset of camera interface.<br>1 = software reset<br>0 = normal | 0 |
| ABSAME | [29] | This bit indicates which is same both A-port image and B-port image. If this bit is 1, A/B-port image will be the scaled images. | 0 |
| BENAS | [20] | This bit indicates the controllability of burst length for DMA operations.<br>1 = software gives burst length<br>0 = self burst length generation | 0 |
| CAMRST | [19] | This bit indicates the software reset of external camera processor.<br>1 = software reset for camera processor<br>0 = normal | 0 |
| IMGCAPA | [18] | This bit indicates the image capture enable for A-port. If this bit set to 1, one clock pulse will be internally generated. | 0 |
| IRQFREE | [17] | This bit can clear the interrupt of camera interface.<br>1 = Interrupt clear<br>0 = normal | 0 |
| TESTPT | [16:14] | These bits indicate the test pattern for verifying the incoming ITU-R BT.601/656 sync signals and fundamental operations.<br>000 = bypass (external input)<br>001 = color-bar pattern<br>010 = square-box pattern<br>011 = solid-line pattern<br>100 = check pattern<br>101 = horizontal increasing pattern<br>110 = DC pattern<br>111 = reserved. | 000 |
| SWAPYUV | [13] | This bit controls the swap between Y and UV sequence.<br>1 = UYVY<br>0 = YUYV (recommended) | 0 |
| YUVORD | [12] | This bit indicates Y/UV order. It is recommended to fix 0. | 0 |
| UVOFFSET | [11] | This bit indicates the data offset of UV signals.<br>1 = offset binary<br>0 = normal (recommended) | 0 |

| | | | |
|---|---|---|---|
| HSYNCPOL | [9] | This bit indicates the polarity of incoming horizontal sync signal.<br>1 = inversion (low active)<br>0 = normal (high active) | 0 |
| VSYNCPOL | [8] | This bit indicates the polarity of incoming vertical sync signal.<br>1 = inversion (low active)<br>0 = normal (high active) | 0 |
| PCLKPOL | [7] | This bit indicates the polarity of incoming pixel clock signal.<br>1 = inversion (for not sufficient setup/hold timing)<br>0 = normal (recommended) | 0 |
| IMGFMT | [6] | This bit indicates the format of input interface.<br>1 = ITU-R BT.601 YCbCr 4:2:2 8-bit mode<br>0 = ITU-R BT.656 YCbCr 4:2:2 8-bit mode | 0 |
| IMGCAPB | [2] | This bit indicates the image capture enable for B-port. If this bit set to 1, one clock pulse will be internally generated. | 0 |

## 2. READ REGISTER

### STATUS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTAT | 0x4F000000 | R | Camera interface status for CPU read | 0x00000000 |

| RDSTAT | Bit | Description | Initial State |
|---|---|---|---|
| Reserved | [31:30] | | 00 |
| RDFCNTA | [29:28] | These bits indicate the count of 4 Ping-Pong frame memories for A-port.<br>00 = A-port 1st frame<br>01 = A-port 2nd frame<br>10 = A-port 3rd frame<br>11 = A-port 4th frame | 00 |
| RDFCNTB | [27:26] | These bits indicate the count of 4 Ping-Pong frame memories for B-port.<br>00 = B-port 1st frame<br>01 = B-port 2nd frame<br>10 = B-port 3rd frame<br>11 = B-port 4th frame | 00 |
| RDCAMRST | [19] | This bit indicates the software reset of external camera processor. | 0 |
| RDIMGCAPA | [18] | This bit indicates the enable state of A-port image capture sync to VSYNC signal. If capture is enabled, this bit will remain high to VYNC falling edge. | 0 |
| RDIRQFREE | [17] | This bit indicates the state of interrupt free.<br>1 = Interrupt clear<br>0 = normal | 0 |
| RDTESTPT | [16:14] | These bits indicate the test pattern for verifying the incoming ITU-R BT.601/656 sync signals and fundamental operations.<br>000 = bypass (external input)<br>001 = color-bar pattern<br>010 = square-box pattern<br>011 = solid-line pattern<br>100 = check pattern<br>101 = horizontal increasing pattern<br>110 = DC pattern<br>111 = reserved. | 000 |
| RDSWAPYUV | [13] | This bit indicates the swap between Y and UV sequence.<br>1 = UYVY<br>0 = YUYV (recommended) | 0 |

| | | | |
|---|---|---|---|
| RDYUVORD | [12] | This bit indicates Y/UV order. It is recommended to fix 0. | 0 |
| RDUVOFFSET | [11] | This bit indicates the data offset of UV signals.<br>1 = offset binary<br>0 = normal (recommended) | 0 |
| RDHSYNCPOL | [9] | This bit indicates the polarity of incoming horizontal sync signal.<br>1 = inversion (low active)<br>0 = normal (high active) | 0 |
| RDVSYNCPOL | [8] | This bit indicates the polarity of incoming vertical sync signal.<br>1 = inversion (low active)<br>0 = normal (high active) | 0 |
| RDPCLKPOL | [7] | This bit indicates the polarity of incoming pixel clock signal.<br>1 = inversion (for not sufficient setup/hold timing)<br>0 = normal (recommended) | 0 |
| RDIMGFMT | [6] | This bit indicates the format of input interface.<br>1 = ITU-R BT.601 YCbCr 4:2:2 8-bit mode<br>0 = ITU-R BT.656 YCbCr 4:2:2 8-bit mode | 0 |

### A Y START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTAY | 0x4F000014 | R | A-port Image current Y start address | 0x00000000 |

| RDSTAY | Bit | Description | Initial State |
|---|---|---|---|
| RDSTAY | [31:0] | This register value indicates the current start address of luminance for A-port | 0x00000000 |

### A CB START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTACB | 0x4F000018 | R | A-port Image current Cb start address | 0x00000000 |

| RDSTACB | Bit | Description | Initial State |
|---|---|---|---|
| RDSTACB | [31:0] | This register value indicates the current start address of Chrominance Cb for A-port | 0x00000000 |

### A CR START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTACR | 0x4F00001C | R | A-port Image current Cr start address | 0x00000000 |

| RDSTACR | Bit | Description | Initial State |
|---|---|---|---|
| RDSTACR | [31:0] | This register value indicates the current start address of Chrominance Cr for A-port | 0x00000000 |

### A CB1 START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTACB1 | 0x4F000020 | R | A-port Image 1st ping-pong memory Cb start address | 0xc073f000 |

| RDSTACB1 | Bit | Description | Initial State |
|---|---|---|---|
| RDSTACB1 | [31:0] | This register value indicates the start address of Chrominance Cb for 1st Ping-Pong memory of A-port | 0xc073f000 |

### A CR1 START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTACR1 | 0x4F000024 | R | A-port Image 1st ping-pong memory Cr start address | 0xc073f000 |

| RDSTACR1 | Bit | Description | Initial State |
|---|---|---|---|
| RDSTACR1 | [31:0] | This register value indicates the start address of Chrominance Cr for 1st Ping-Pong memory of A-port | 0xc073f000 |

### B Y1 START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBY1 | 0x4F000028 | R | B-port Image 1st ping-pong memory Y start address | 0xc073f000 |

| RDSTBY1 | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBY1 | [31:0] | This register value indicates the start address of luminance for 1st Ping-Pong memory of B-port | 0xc073f000 |

### B Y2 START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBY2 | 0x4F00002C | R | B-port Image 2nd ping-pong memory Y start address | 0xc07eec80 |

| RDSTBY2 | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBY2 | [31:0] | his register value indicates the start address of luminance for 2nd Ping-Pong memory of B-port | 0xc07eec80 |

### B Y3 START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBY3 | 0x4F000030 | R | B-port Image 3rd ping-pong memory Y start address | 0xc089e900 |

| RDSTBY3 | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBY3 | [31:0] | his register value indicates the start address of luminance for 3rd Ping-Pong memory of B-port | 0xc089e900 |

### B Y4 START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBY4 | 0x4F000034 | R | B-port Image 4th ping-pong memory Y start address | 0xc094e580 |

| RDSTBY4 | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBY4 | [31:0] | his register value indicates the start address of luminance for 4th Ping-Pong memory of B-port | 0xc094e580 |

### B Y START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBY | 0x4F000038 | R | B-port Image current Y start address | 0x00000000 |

| RDSTBY | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBY | [31:0] | This register value indicates the current start address of luminance for B-port | 0x00000000 |

### B CB START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBCB | 0x4F00003C | R | B-port Image current Cb start address | 0x00000000 |

| RDSTBCB | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBCB | [31:0] | This register value indicates the current start address of chrominance Cb for B-port | 0x00000000 |

### B CR START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBCR | 0x4F000040 | R | B-port Image current Cr start address | 0x00000000 |

| RDSTBCR | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBCR | [31:0] | This register value indicates the current start address of chrominance Cr for B-port | 0x00000000 |

### B CB1 START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBCB1 | 0x4F000044 | R | B-port Image 1st ping-pong memory Cb start address | 0xc073f000 |

| RDSTBCB1 | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBCB1 | [31:0] | This register value indicates the start address of Chrominance Cb for 1st Ping-Pong memory of B-port | 0xc073f000 |

### B CR1 START ADDRESS READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDSTBCR1 | 0x4F000048 | R | B-port Image 1st ping-pong memory Cr start address | 0xc073f000 |

| RDSTBCR1 | Bit | Description | Initial State |
|---|---|---|---|
| RDSTBCR1 | [31:0] | This register value indicates the start address of Chrominance Cr for 1st Ping-Pong memory of B-port | 0xc073f000 |

### A LAST HREF DISTANCE WIDTH READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDADISTWIDTH | 0x4F00004C | R | A-port distance and width for last hsync generation | 0x1b581040 |

| RDADISTWIDTH | Bit | Description | Initial State |
|---|---|---|---|
| RDADIST | [31:16] | These bits indicate the distance from the start of external last HREF to the start of imitative HREF for last DMA operation for A-port. | 0x1b58 |
| RDAWIDTH | [15:0] | These bits indicate the width of imitative HREF for last DMA operation for A-port. | 0x1040 |

### B LAST HREF DISTANCE WIDTH READ REGISTER

| Register | Address | R/W | Description | Reset Value |
|---|---|---|---|---|
| RDBDISTWIDTH | 0x4F000050 | R | B-port distance and width for last hsync generation | 0x1b581040 |

| RDBDISTWIDTH | Bit | Description | Initial State |
|---|---|---|---|
| RDBDIST | [31:16] | These bits indicate the distance from the start of external last HREF to the start of imitative HREF for last DMA operation for B-port. | 0x1b58 |
| RDBWIDTH | [15:0] | These bits indicate the width of imitative HREF for last DMA operation for B-port. | 0x1040 |

# 24 BUS PRIORITIES

## OVERVIEW

The bus arbitration logic determines the priorities of bus masters. It supports a combination of rotation priority mode and fixed priority mode.

### BUS PRIORITY MAP

The S3C2440X01 holds eleven bus masters including DRAM refresh controller, LCD_DMA, DMA0, DMA1, DMA2, DMA3, USB_HOST_DMA, EXT_BUS_MASTER, Test interface controller (TIC), and ARM920T. The following list shows the priorities among these bus masters after a reset:

1. DRAM refresh controller
2. LCD_DMA
3. DMA0
4. DMA1
5. DMA2
6. DMA3
7. USB host DMA
8. External bus master
9. TIC
10. ARM920T
11. Reserved

Among those bus masters, four DMAs operate under the rotation priority, while others run under the fixed priority.

# NOTES

# 25 ELECTRICAL DATA

## ABSOLUTE MAXIMUM RATINGS

**Table 25-1. Absolute Maximum Rating**

| Parameter | Symbol | Rating | | Unit |
|---|---|---|---|---|
| DC Supply Voltage | $V_{DDi}$ | 1.2V $V_{DD}$ | 1.8 | V |
| | $V_{DDIO}$ | 3.3V $V_{DD}$ | 4.8 | |
| | $V_{DDRTC}$ | 3.0V $V_{DD}$ | 4.5 | |
| DC Input Voltage | $V_{IN}$ | 3.3V Input buffer | 4.8 | |
| | | 3.3V Interface / 5V Tolerant input buffer | 6.5 | |
| DC Output Voltage | $V_{OUT}$ | 3.3V Output buffer | 4.8 | |
| DC Input (Latch-up) Current | $I_{IN}$ | ± 200 | | mA |
| Storage Temperature | $T_{STG}$ | – 65 to 150 | | °C |

## RECOMMENDED OPERATING CONDITIONS

**Table 25-2. Recommended Operating Conditions**

| Parameter | Symbol | Rating | | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| DC Supply Voltage for Internal | $V_{DDi}$ | 1.2V $V_{DD}$ | 1.1 | 1.3 | V |
| DC Supply Voltage for I/O Block | $V_{DDIO}$ | 3.3V $V_{DD}$ | 3.0 | 3.6 | |
| DC Supply Voltage for Analog Core | $V_{DD}$ | 3.3V $V_{DD}$ | 3.0 | 3.6 | |
| DC Supply Voltage for RTC | $V_{DDRTC}$ | 3.0V $V_{DD}$ | 2.7 | 3.6 | |
| DC Input Voltage | $V_{IN}$ | 3.3V Input buffer | – 0.3 | VDDIO+0.3 | |
| | | 3.3V Interface / 5V Tolerant input buffer | – 0.3 | 5.25 | |
| DC Output Voltage | $V_{OUT}$ | 3.3V Output buffer | – 0.3 | VDDIO+0.3 | |
| Operating Temperature | $T_{OPR}$ | Commercial | 0 to 70 | | °C |

## D.C. ELECTRICAL CHARACTERISTICS

Table 25-3 and 25-4 define the DC electrical characteristics for the standard LVCMOS I/O buffers.

**Table 25-3. Normal I/O PAD DC Electrical Characteristics**

($V_{DD}$ = 3.3V ± 0.3V, $T_A$ = 0 to 70 °C)

| Symbol | Parameters | Condition | Min | Type | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{IH}$ | High level input voltage | | | | | |
| | LVCMOS interface | | 2.0 | | | V |
| $V_{IL}$ | Low level input voltage | | | | | |
| | LVCMOS interface | | | | 0.8 | V |
| VT | Switching threshold | | | $0.5V_{DD}$ | | V |
| VT+ | Schmitt trigger, positive-going threshold | CMOS | | | 2.0 | V |
| VT- | Schmitt trigger, negative-going threshold | CMOS | 0.8 | | | V |
| $I_{IH}$ | High level input current | | | | | |
| | Input buffer | $V_{IN} = V_{DD}$ | -10 | | 10 | μA |
| $I_{IL}$ | Low level input current | | | | | |
| | Input buffer | $V_{IN} = V_{SS}$ | -10 | | 10 | μA |
| | Input buffer with pull-up | | -60 | -33 | -10 | |
| $V_{OH}$ | High level output voltage | | | | | |
| | Type B4 | $I_{OH}$= - 4 mA | 2.4 | | | V |
| | Type B6 | $I_{OH}$= - 6 mA | | | | |
| | Type B8 | $I_{OH}$= - 8 mA | | | | |
| | Type B10 | $I_{OH}$= -10 mA | | | | |
| | Type B12 | $I_{OH}$ = -12 mA | | | | |
| $V_{OL}$ | Low level output voltage | | | | | |
| | Type B4 | $I_{OL}$ = 4 mA | | | 0.4 | V |
| | Type B6 | $I_{OL}$ = 6 mA | | | | |
| | Type B8 | $I_{OL}$ = 8 mA | | | | |
| | Type B10 | $I_{OL}$ = 10 mA | | | | |
| | Type B12 | $I_{OL}$ = 12 mA | | | | |

**NOTES:**
1. Type B6 means 6mA output driver cell.
2. Type B8 means 8mA output driver cell.
3. Type B12 means 12mA output driver cells.

**Table 25-4. USB DC Electrical Characteristics**

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{IH}$ | High level input voltage | | 2.5 | | V |
| $V_{IL}$ | Low level input voltage | | | 0.8 | V |
| $I_{IH}$ | High level input current | Vin = 3.3V | -10 | 10 | μA |
| $I_{IL}$ | Low level input current | Vin = 0.0V | -10 | 10 | μA |
| $V_{OH}$ | Static Output High | 15K to GND | 2.8 | 3.6 | V |
| $V_{OL}$ | Static Output Low | 1.5K to 3.6V | | 0.3 | V |

**Table 25-5. S3C2440X Power Supply Voltage and Current**

| Parameter | Value | Unit | Condition |
|---|---|---|---|
| Typical $V_{DDi}$ / $V_{DDIO}$ | 1.2 / 3.3 | V | |
| Max. Operating frequency (FCLK) | 400 | MHz | |
| Max. Operating frequency (HCLK) | 133 | MHz | |
| Max. Operating frequency (PCLK) | 67 | MHz | |
| Typical normal mode power NOTE(3) (Total $V_{DDi}$ + $V_{DDIO}$) | 369 | mW | NOTE(1) |
| Typical normal mode power NOTE(3) (Total $V_{DDi}$ + $V_{DDIO}$) | 266 | mW | NOTE(2) |
| Typical idle mode power NOTE(3) (Total $V_{DDi}$ + $V_{DDIO}$) | 172 | mW | FCLK = 400MHz (F:H:P = 1:3:6) |
| Typical slow mode power NOTE(3) (Total $V_{DDi}$ + $V_{DDIO}$) | 97 | mW | FCLK = 12MHz (F:H:P = 1:1:1) |
| Maximum Sleep mode power<br>Typical Sleep mode power NOTE(3) | 100<br>10 | uA<br>uA | @1.2/3.3V, Room temperature<br>All other I/O static. |
| Maximum RTC power<br>Typical RTC power NOTE(3) | 63<br>4 | uA<br>uA | @3.0V, Room temperature<br>X-tal = 32.768KHz for RTC |

**NOTES:**

1. I/D cache: ON, MMU: ON, Code on SRAM, FCLK:HCLK:PCLK = 400MHz:133MHz:66.7MHz
   :LCD ON (320x240x16bppx60Hz, color TFT):13KHz Timer internal mode(5 Channel run)
   :Audio(IIS&DMA,CDCLK=16.9MHz,LRCK=44.1KHz):Integer data quick sort(65536 EA)
2. Pocket PC 2003 MPEG play.
3. Room temperature specification.

**Table 25-6. Typical Current Decrease by CLKCON Register (FCLK@400MHz)**

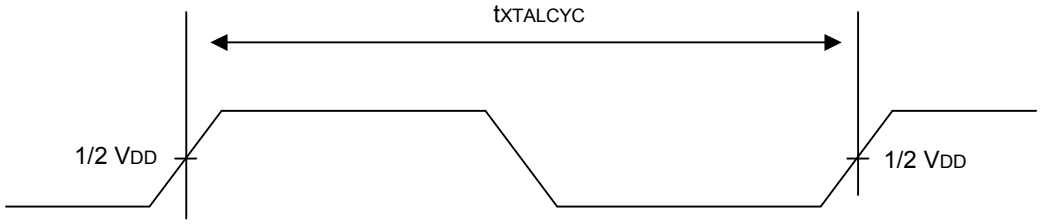
(Unit: mA)

| Peripherals | NFC | LCD | USBH | USBD | Timer | SDI | UART | RTC | ADC | IIC | IIS | SPI | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Current | | | | | | | | | | | | | |

**NOTE:** This table includes each power consumption of each peripherals. For example, If you do not use IIS and you turned off IIS block by CLKCON register, you can save the 0.5mA .

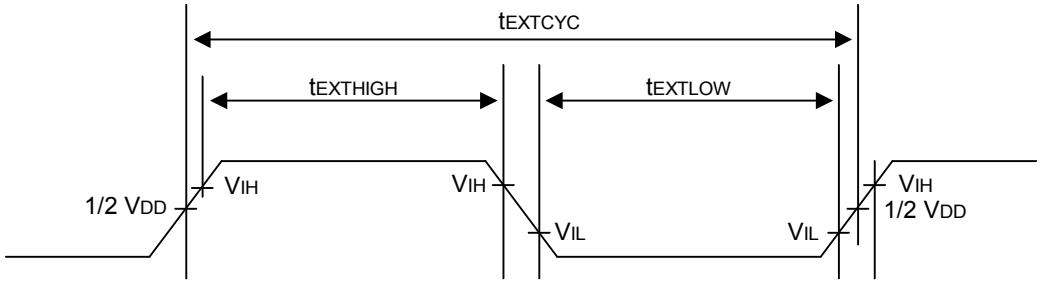
**Figure 25-1. Typical Operating Voltage/Frequency Range (VDDIO=3.3V, @Room temperature & SMDK2440 board)**

## A.C. ELECTRICAL CHARACTERISTICS




**NOTE:** The clock input from the XTIpll pin.

**Figure 25-2. XTIpll Clock Timing**




**NOTE:** The clock input from the EXTCLK pin.

**Figure 25-3. EXTCLK Clock Input Timing**

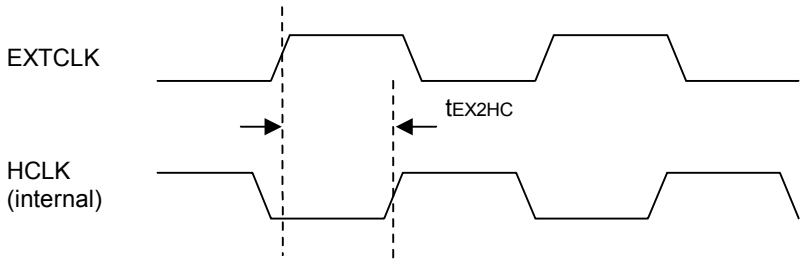



**Figure 25-4. EXTCLK/HCLK in case that EXTCLK is used without the PLL**

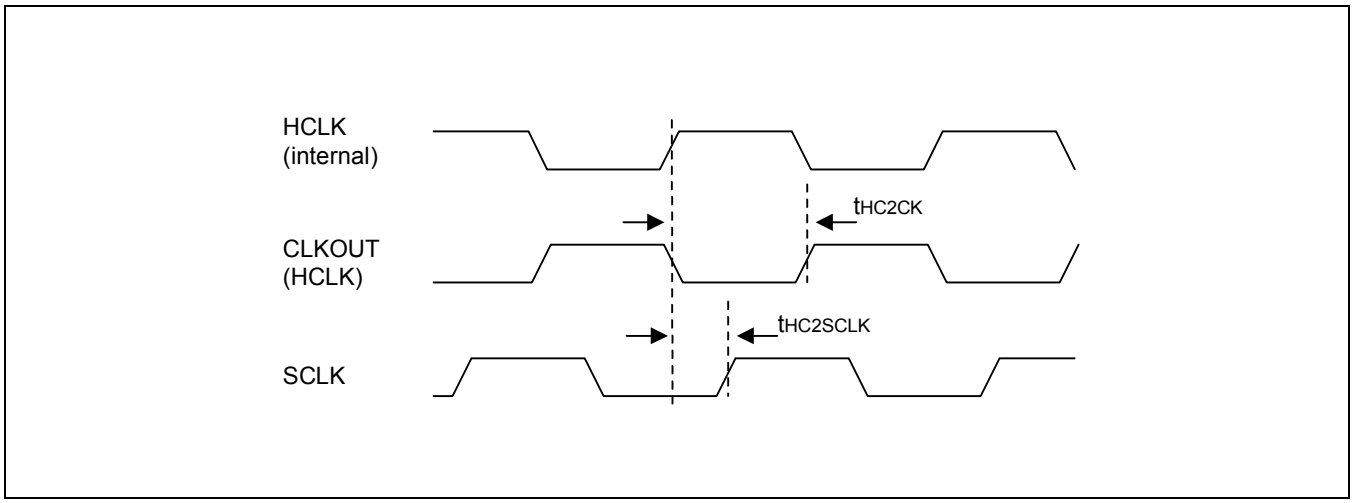



**Figure 25-5. HCLK/CLKOUT/SCLK in case that EXTCLK is used**

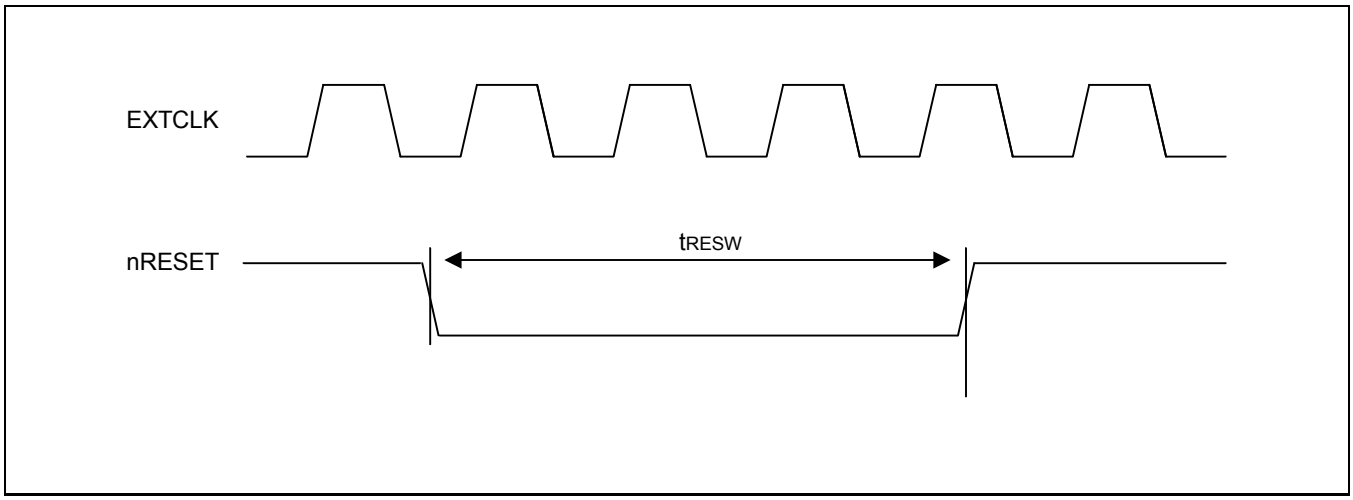



**Figure 25-6. Manual Reset Input Timing**

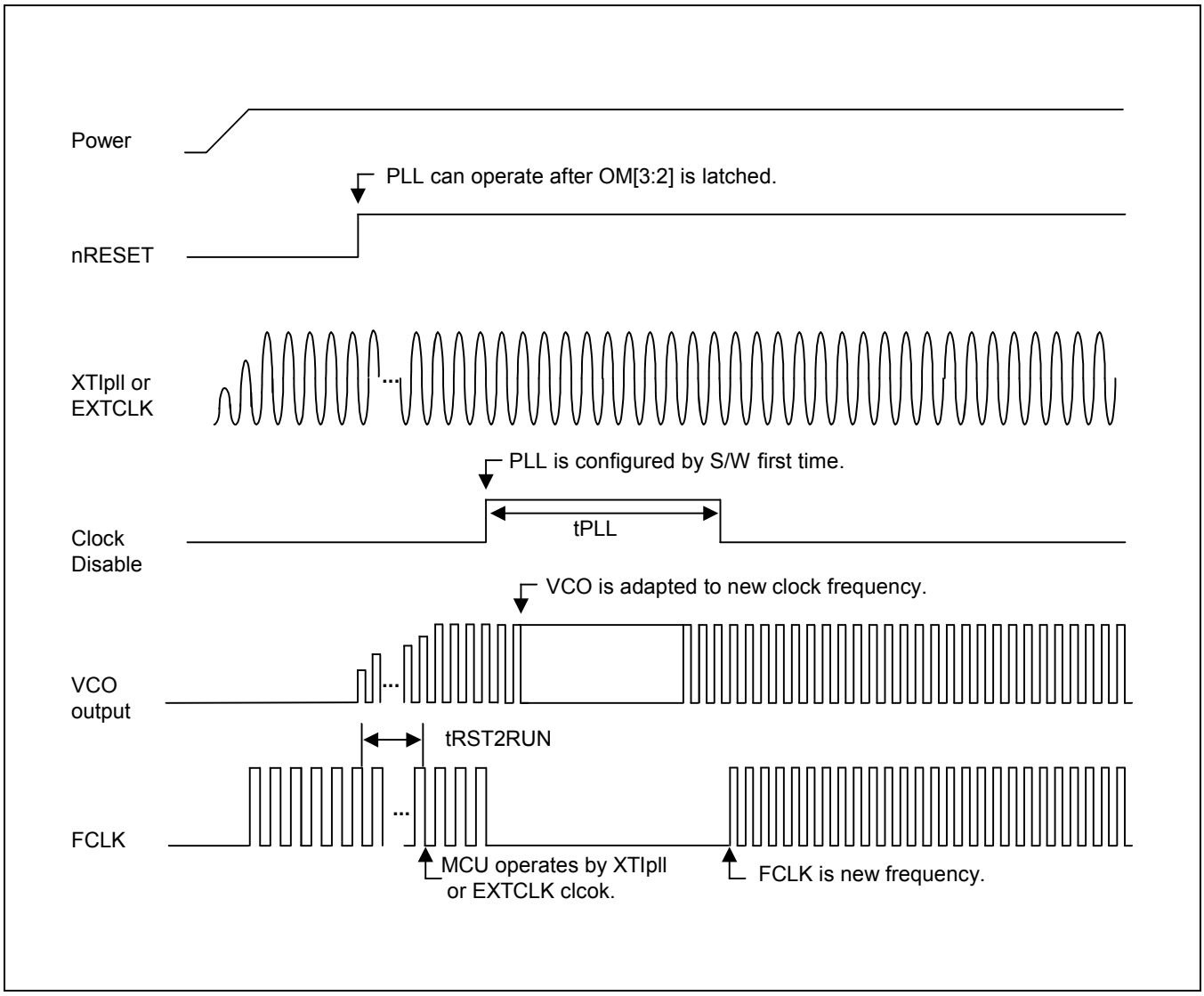



**Figure 25-7. Power-On Oscillation Setting Timing**

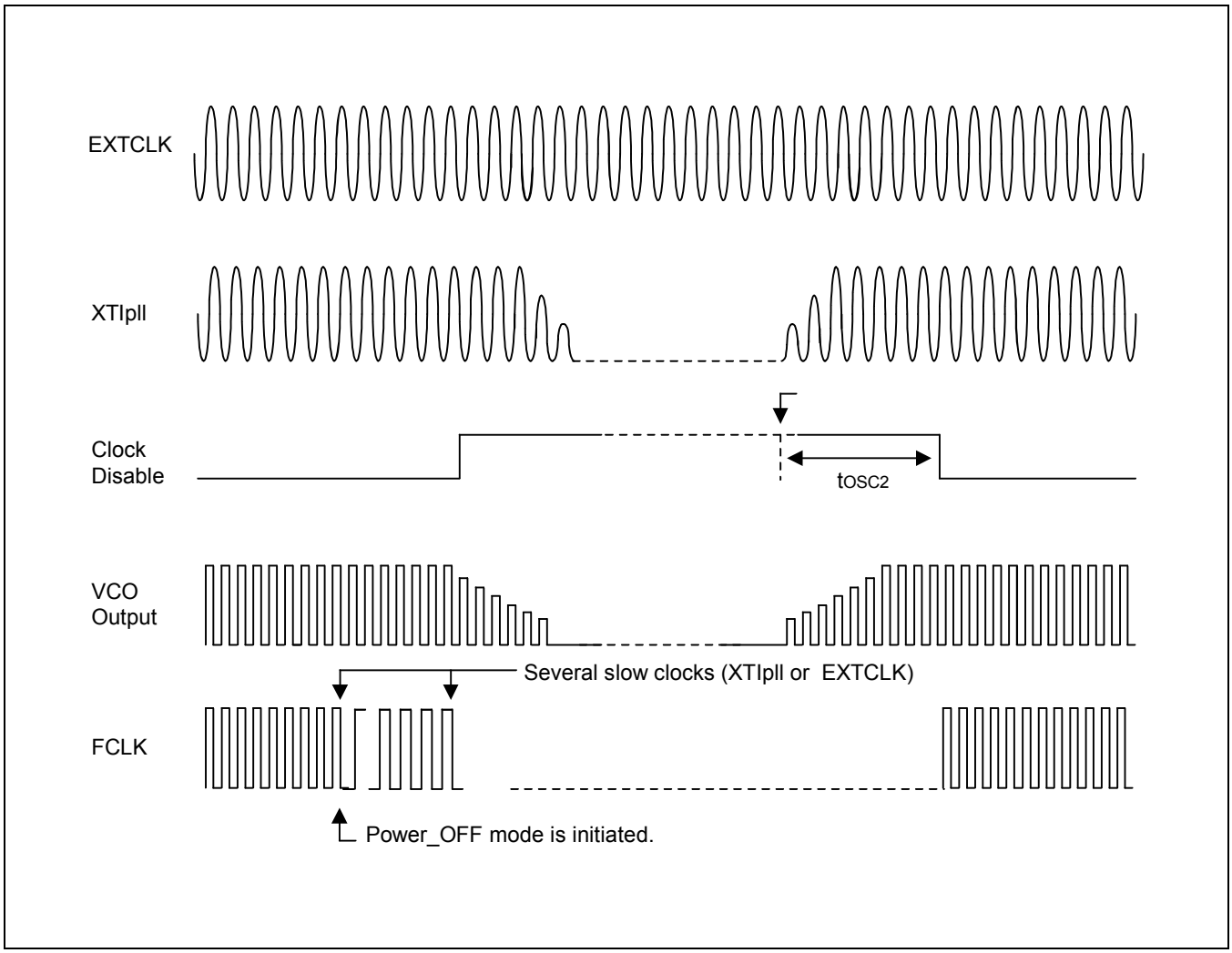



**Figure 25-8. Sleep Mode Return Oscillation Setting Timing**

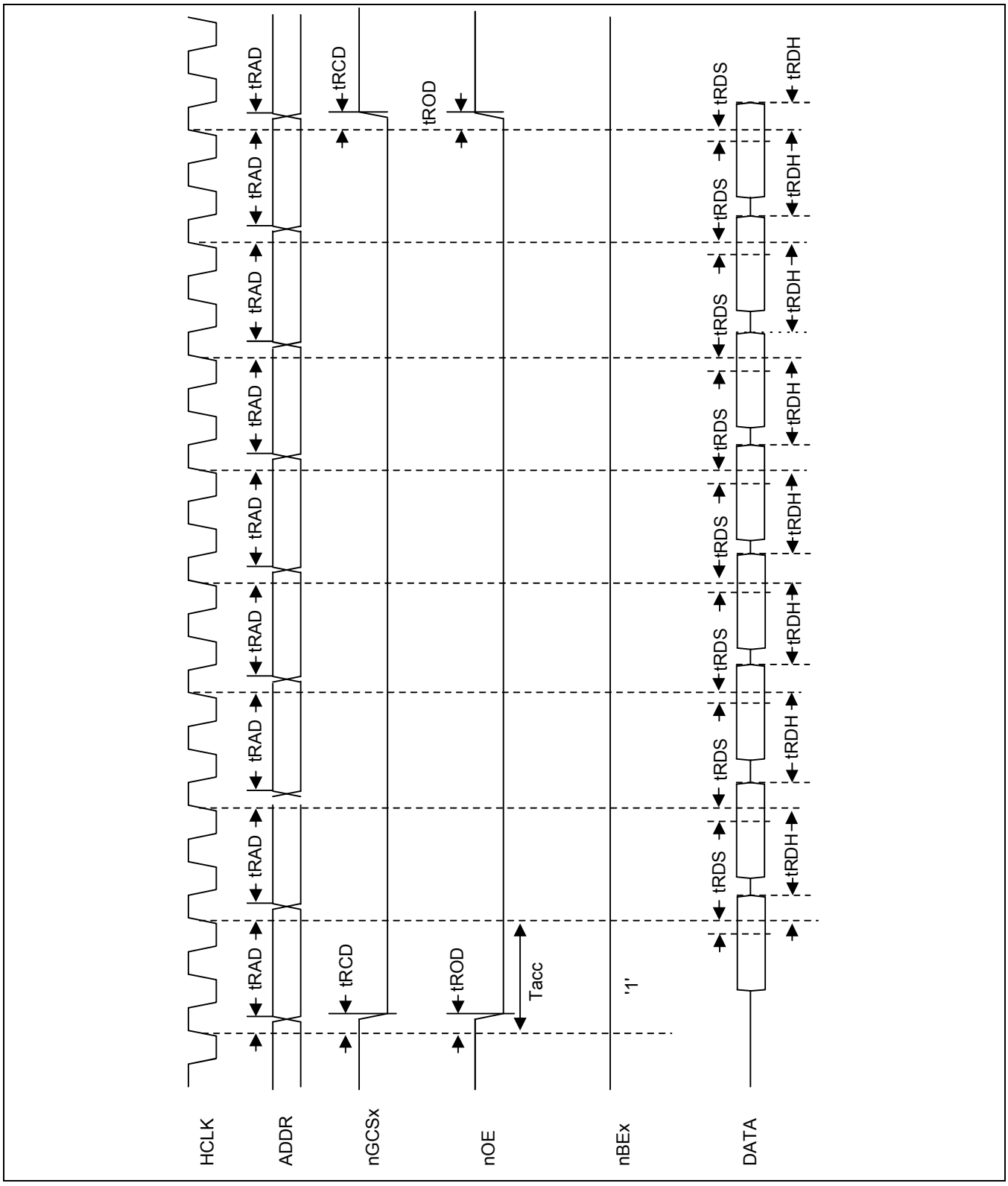



**Figure 25-9. ROM/SRAM Burst READ Timing(I)**
**(Tacs=0, Tcos=0, Tacc=2, Toch=0, Tcah=0, PMC=0, ST=0, DW=16bit)**

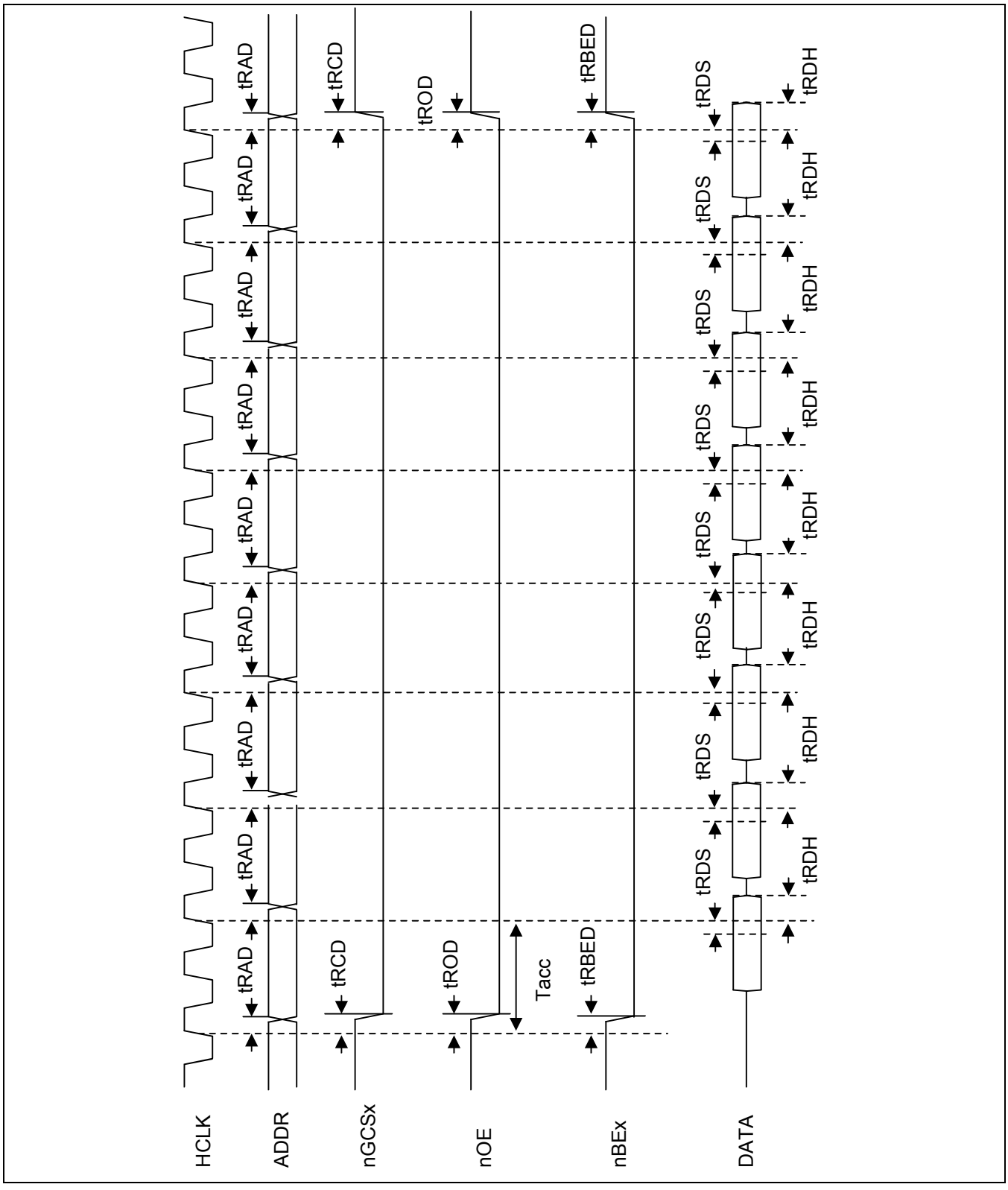



**Figure 25-10. ROM/SRAM Burst READ Timing(II)**
**(Tacs=0, Tcos=0, Tacc=2, Toch=0, Tcah=0, PMC=0, ST=1, DW=16bit)**

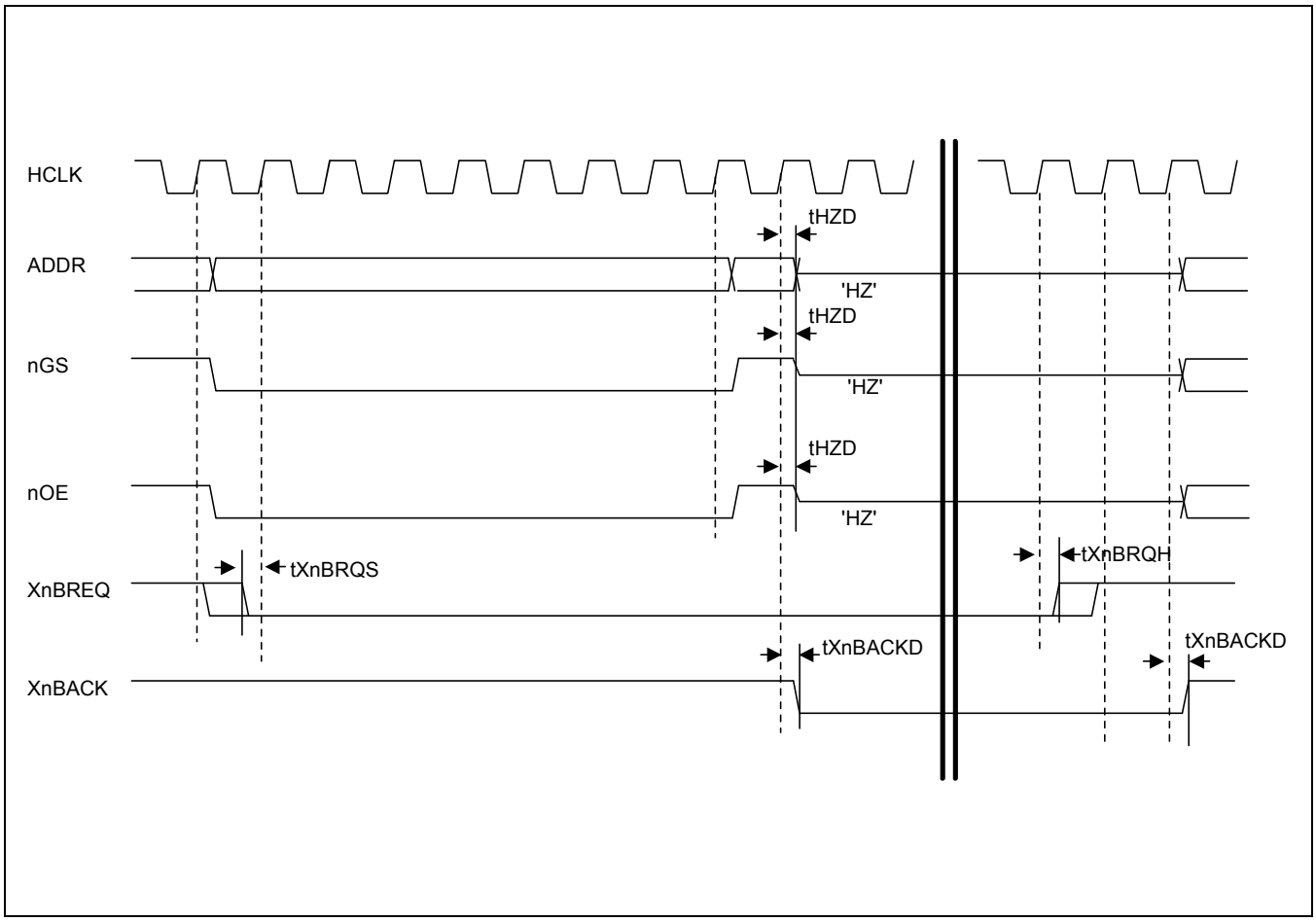



**Figure 25-11. External Bus Request in ROM/SRAM Cycle (Tacs=0, Tcos=0, Tacc=8, Toch=0, Tcah=0, PMC=0, ST=0)**

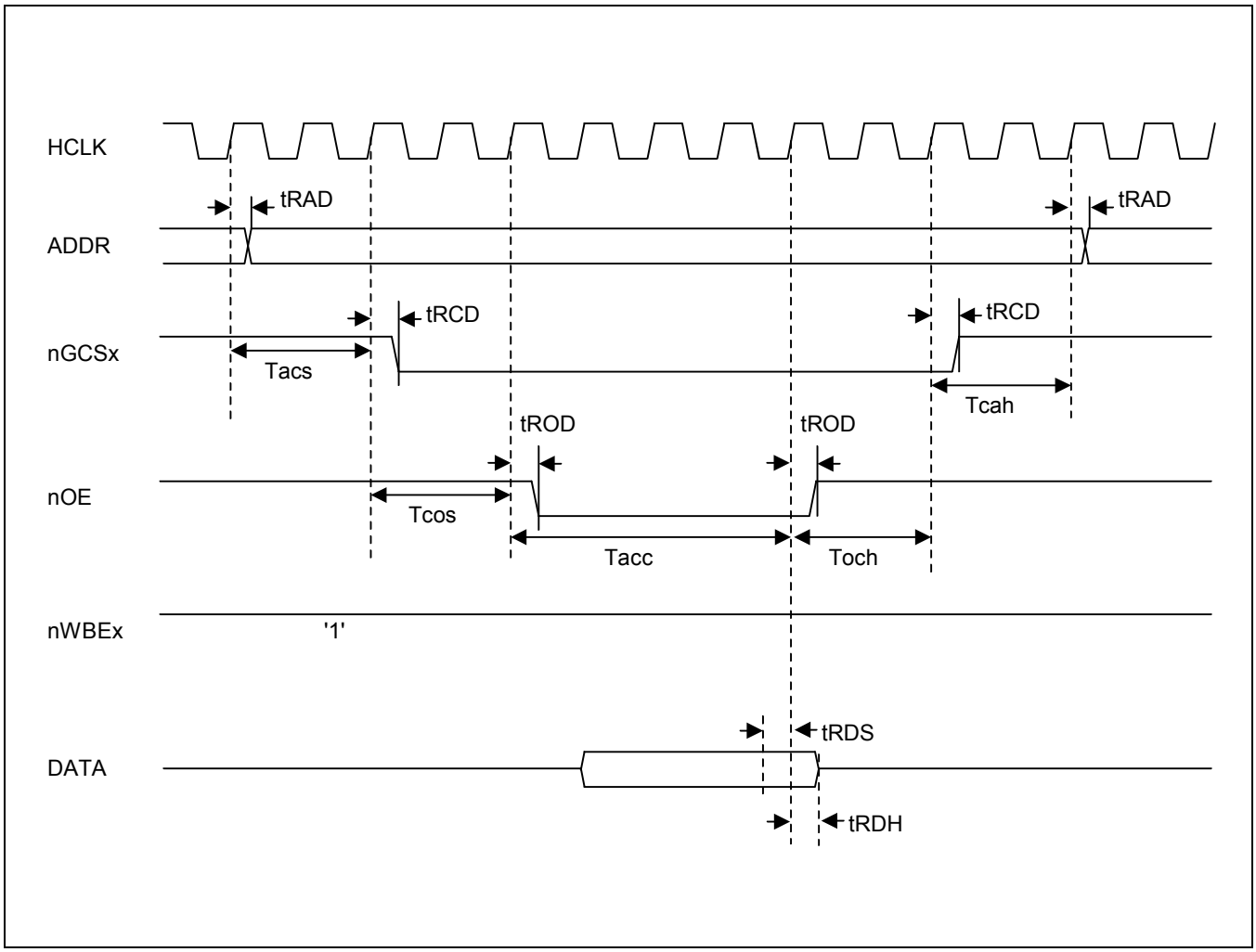



**Figure 25-12. ROM/SRAM READ Timing (I)**
**(Tacs=2,Tcos=2, Tacc=4, Toch=2, Tcah=2, PMC=0, ST=0)**

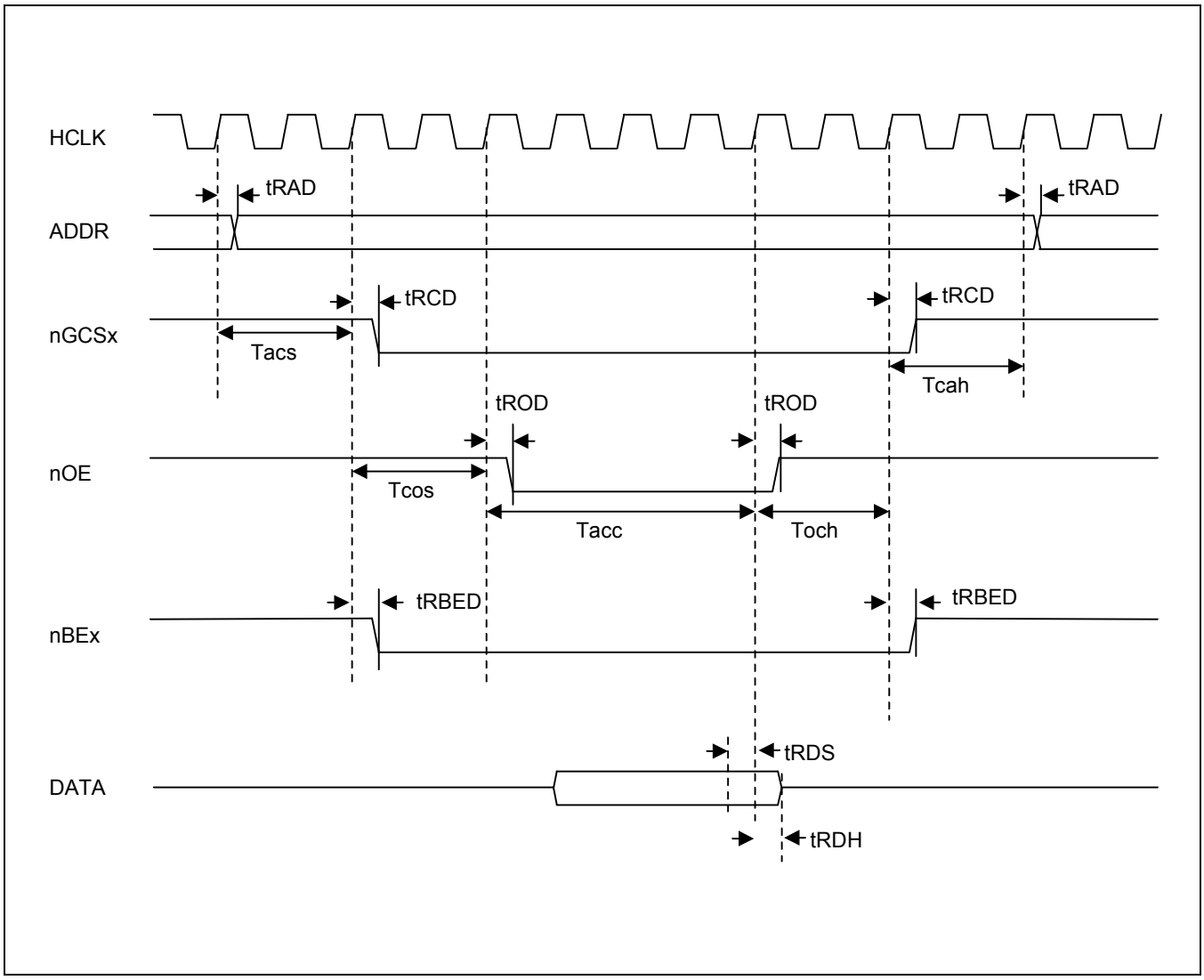



**Figure 25-13. ROM/SRAM READ Timing (II)**
**(Tacs=2, Tcos=2, Tacc=4, Toch=2, Tcah=2cycle, PMC=0, ST=1)**

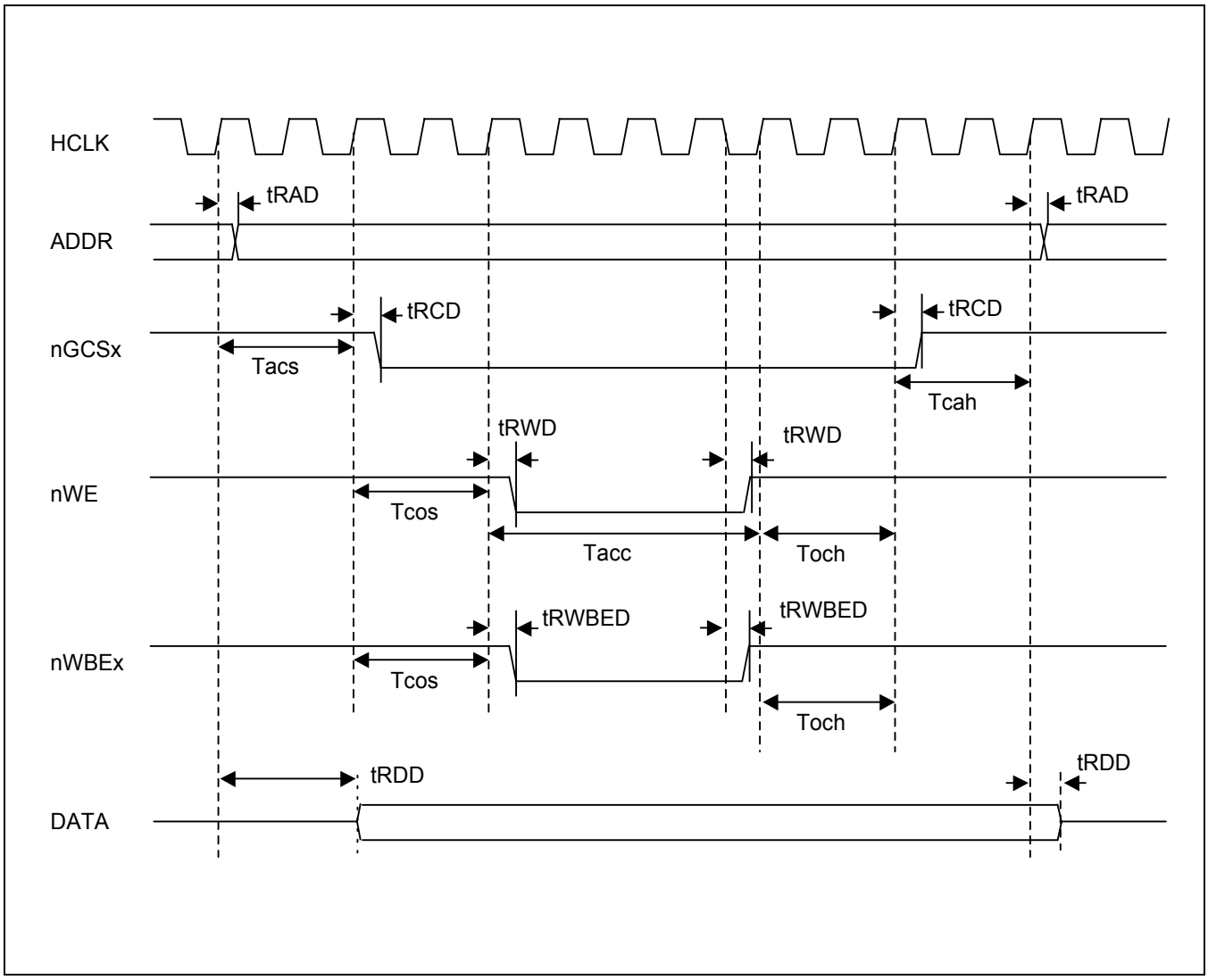



**Figure 25-14. ROM/SRAM WRITE Timing (I)**
**(Tacs=2,Tcos=2,Tacc=4,Toch=2, Tcah=2, PMC=0, ST=0**

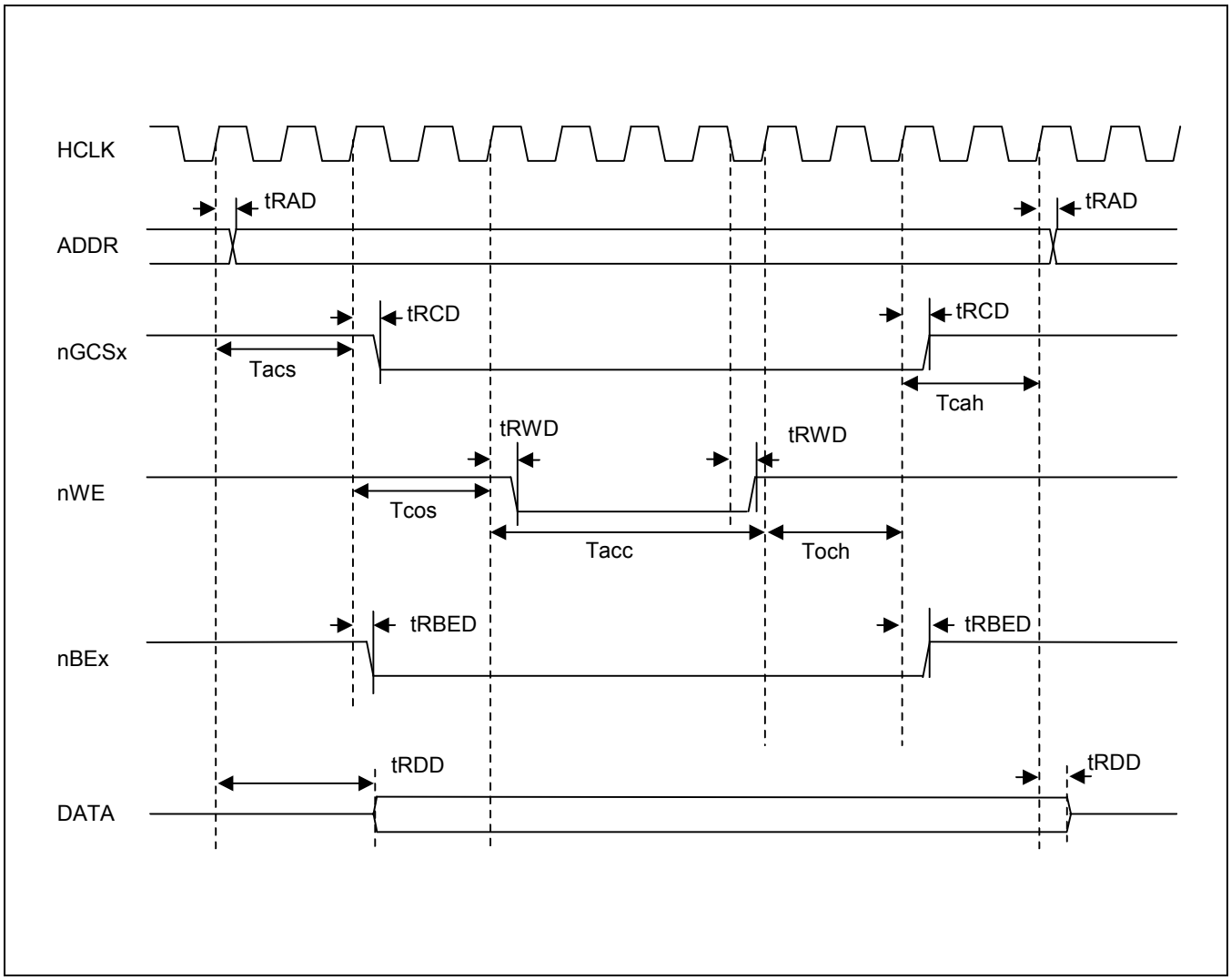
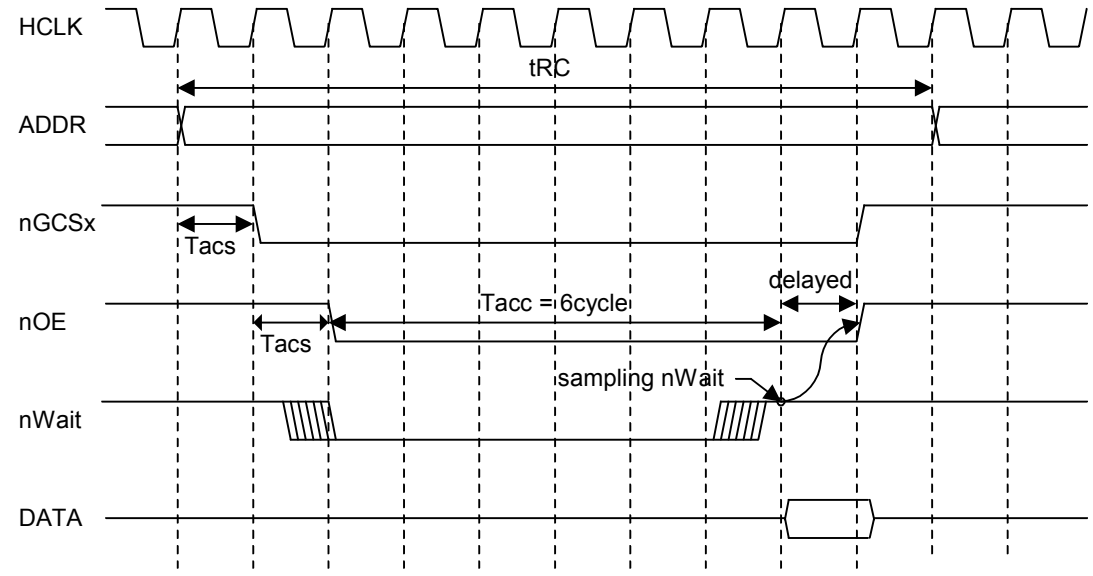



**Figure 25-15. ROM/SRAM WRITE Timing (II)**
**(Tacs=2, Tcos=2, Tacc=4, Toch=2, Tcah=2, PMC=0, ST=1)**




**NOTE:** The status of nWait is checked at (Tacc-1) cycle.

**Figure 25-16. External nWAIT READ Timing (Tacs=0, Tcos=0, Tacc=6, Toch=0, Tcah=0, PMC=0, ST=0)**

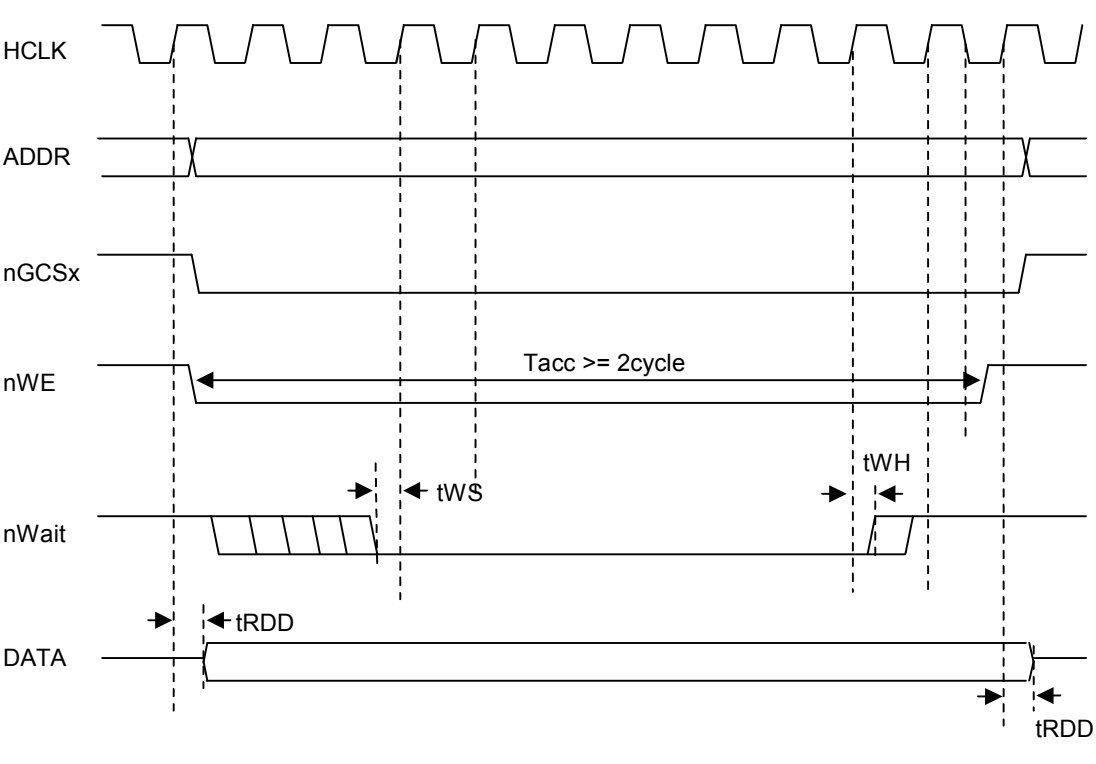



**Figure 25-17. External nWAIT WRITE Timing (Tacs=0, Tcos=0, Tacc=4, Toch=0, Tcah=0, PMC=0, ST=0)**

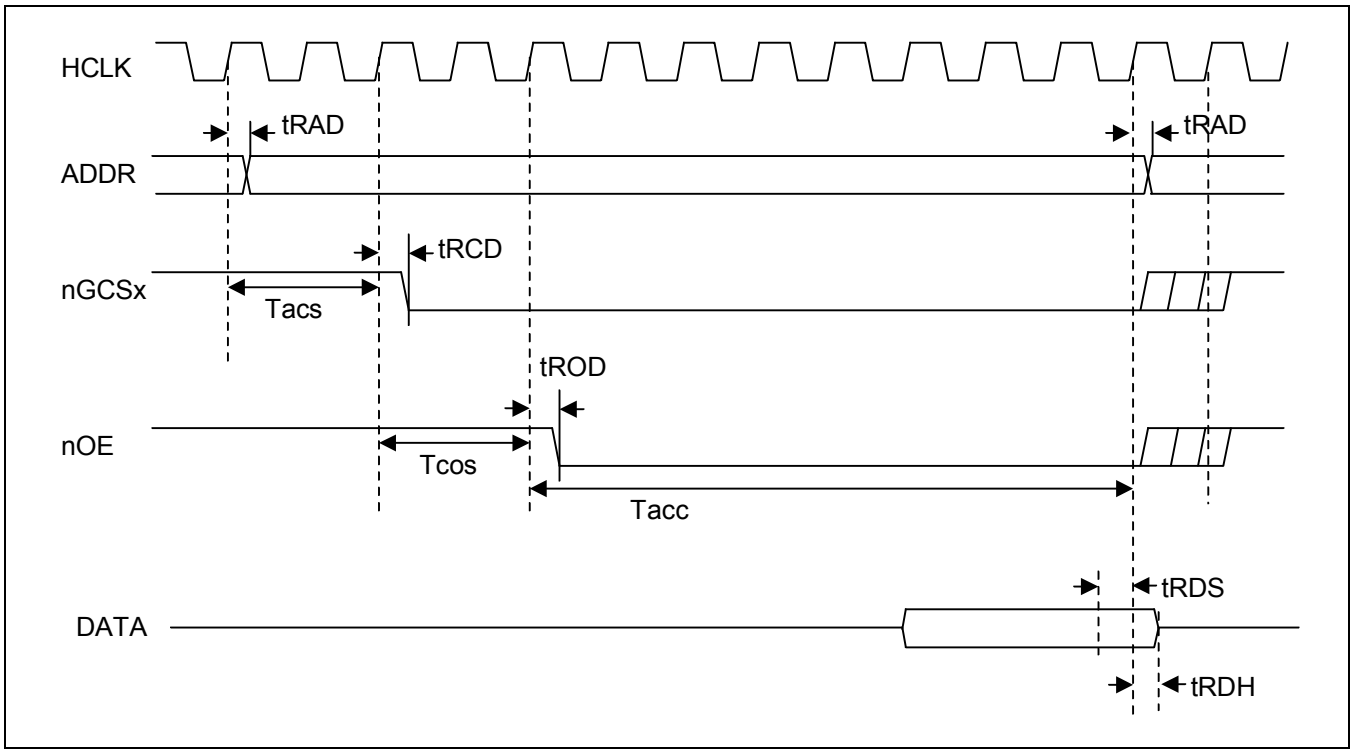



**Figure 25-18. Masked-ROM Single READ Timing (Tacs=2, Tcos=2, Tacc=8, PMC=01/10/11)**

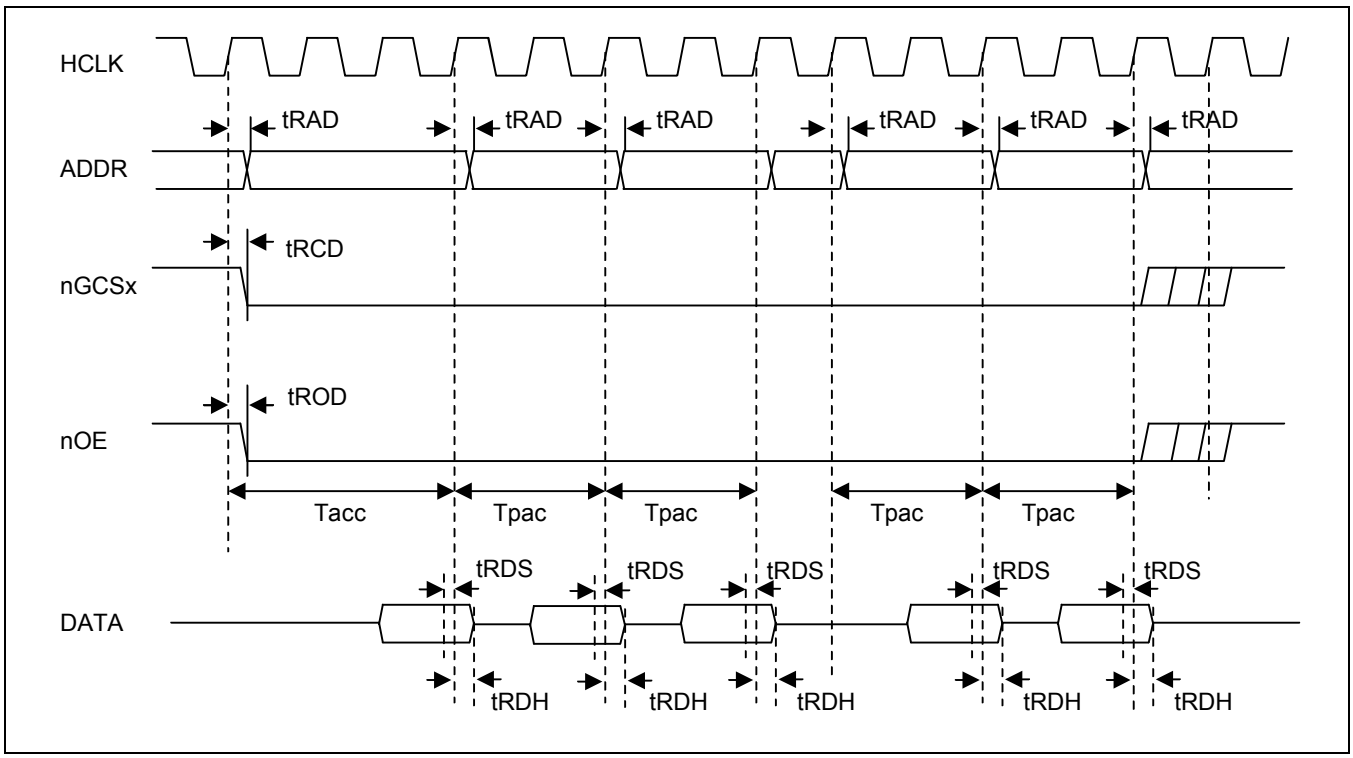



**Figure 25-19. Masked-ROM Consecutive READ Timing (Tacs=0, Tcos=0, Tacc=3, Tpac=2, PMC=01/10/11)**

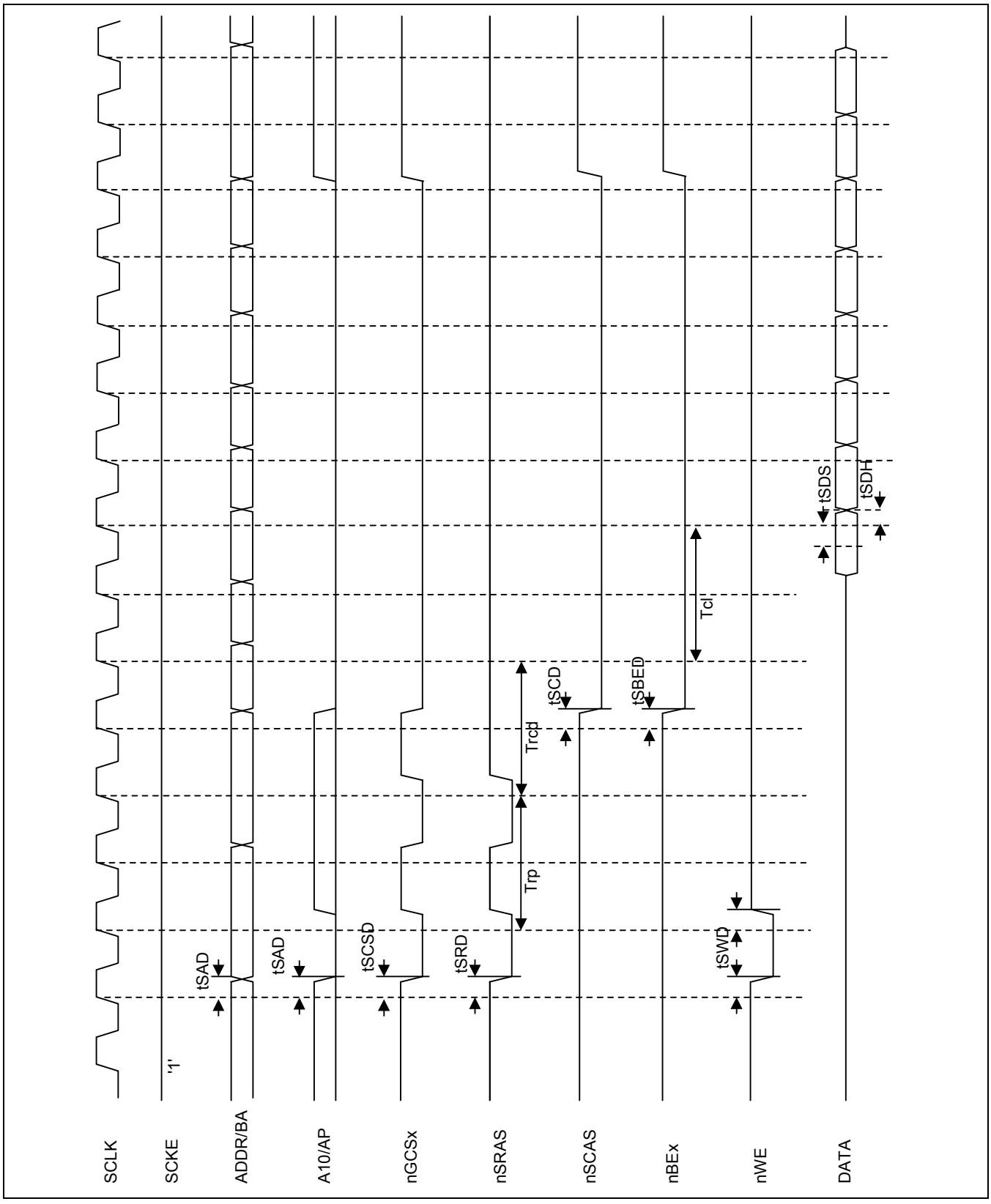



**Figure 25-20. SDRAM Single Burst READ Timing (Trp=2, Trcd=2, Tcl=2, DW=16bit)**

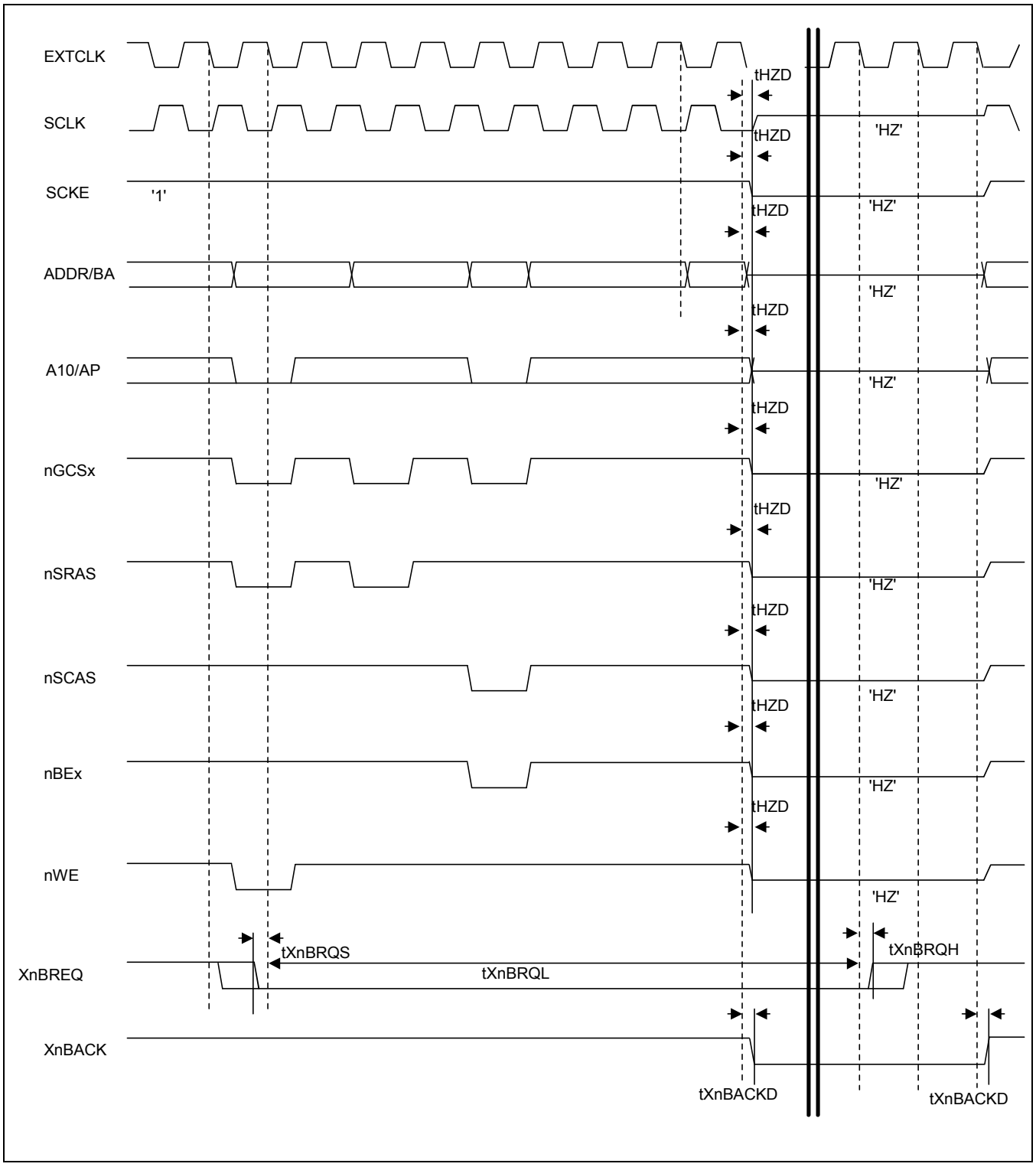



**Figure 25-21. External Bus Request in SDRAM Timing (Trp=2, Trcd=2, Tcl=2)**

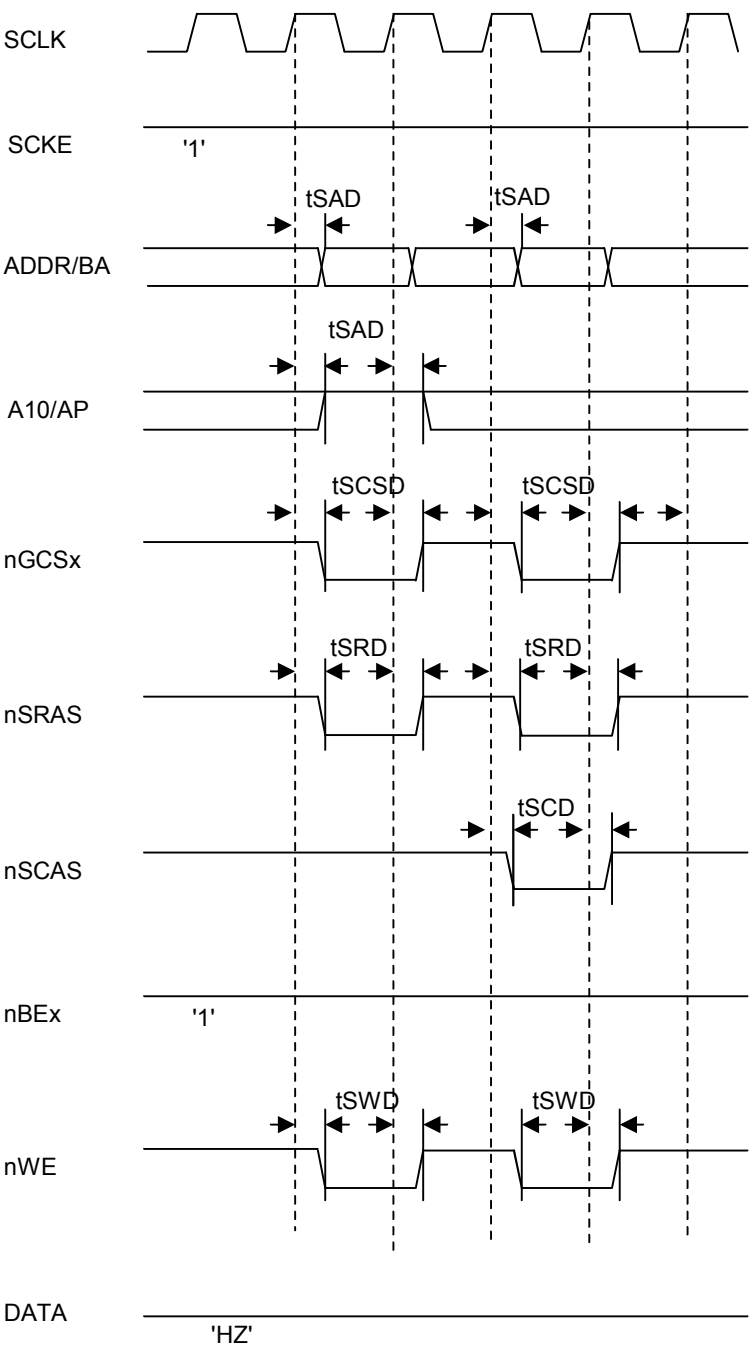



**Figure 25-22. SDRAM MRS Timing**

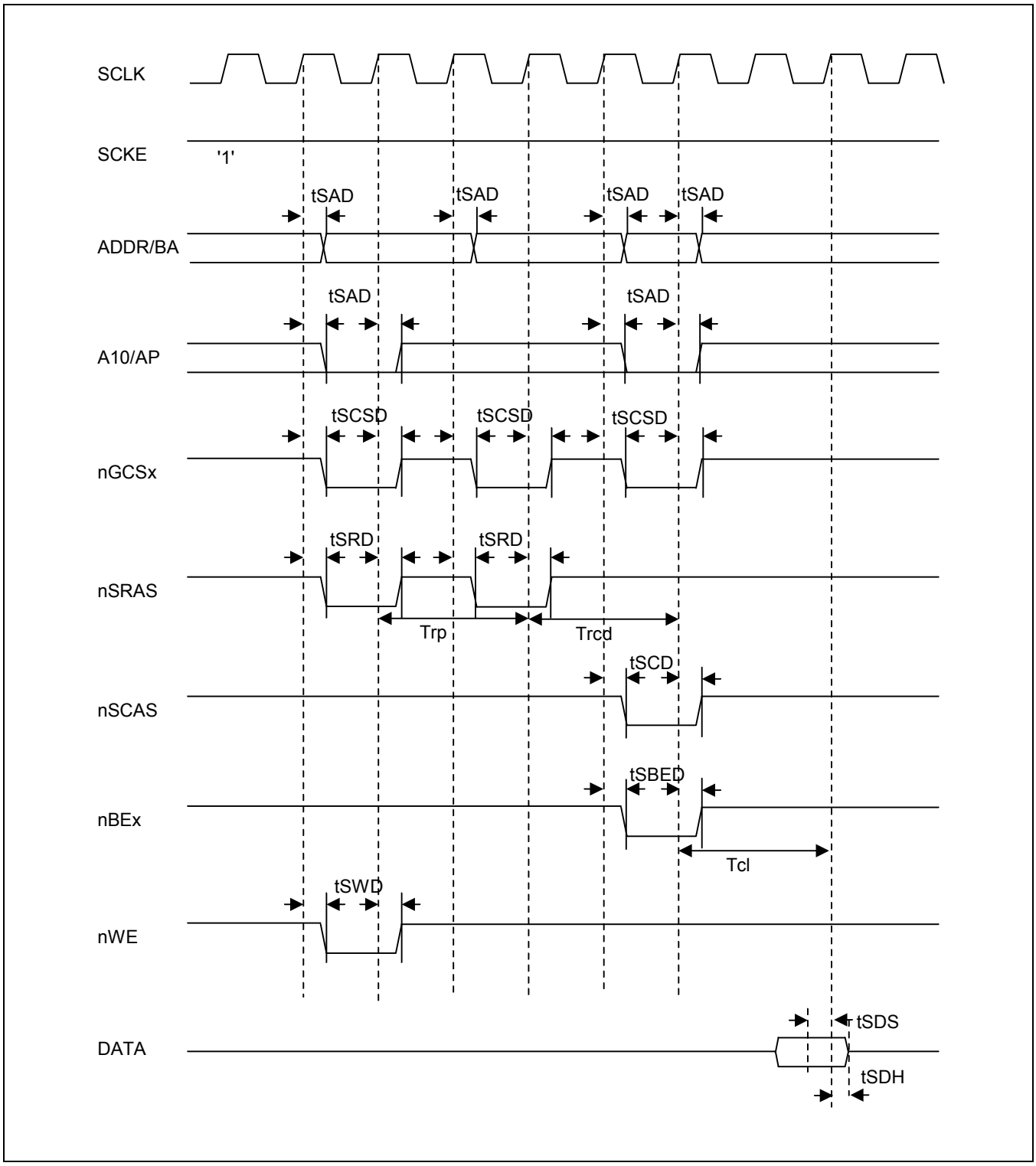



**Figure 25-23. SDRAM Single READ Timing(I) (Trp=2, Trcd=2, Tcl=2)**

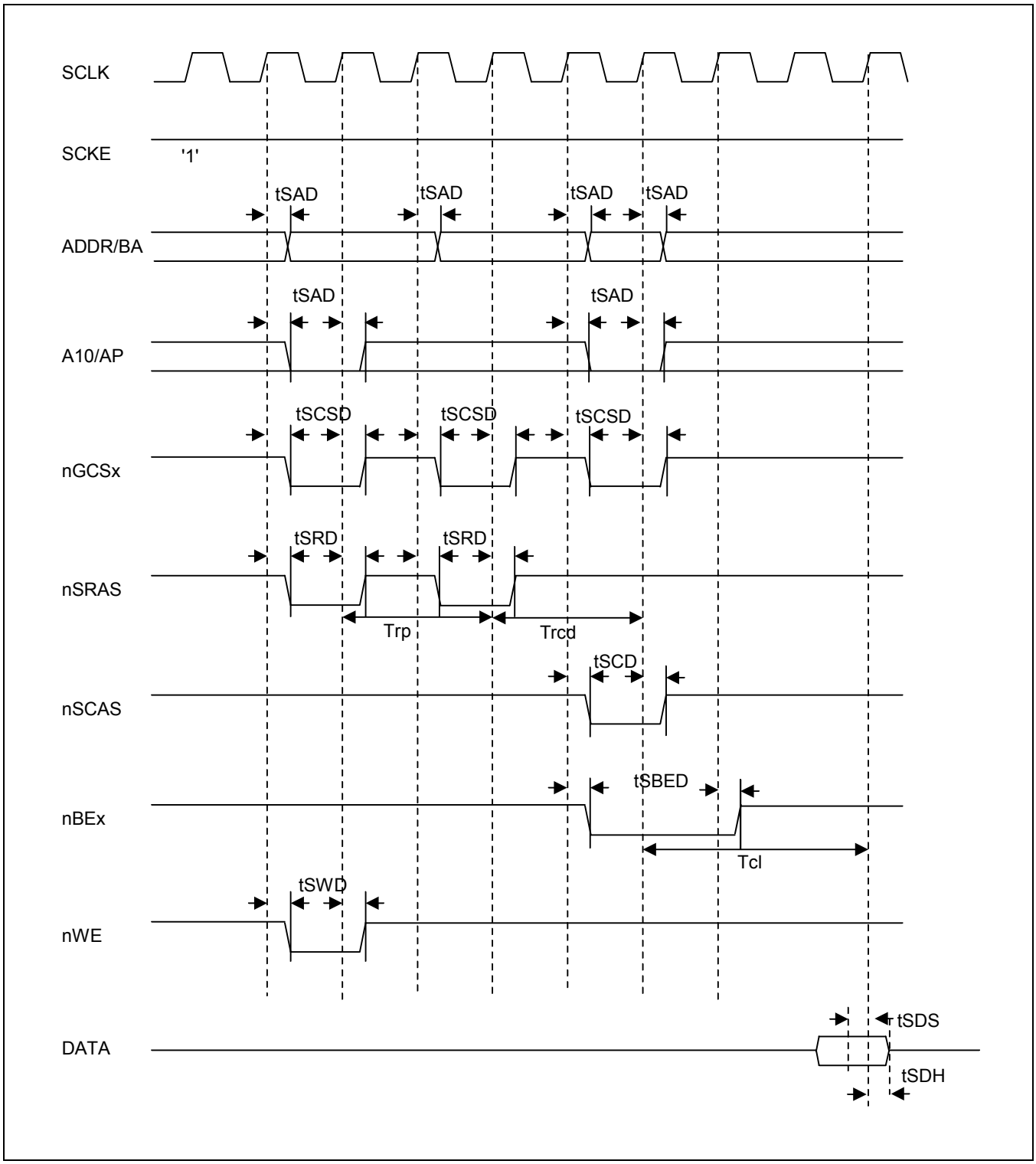
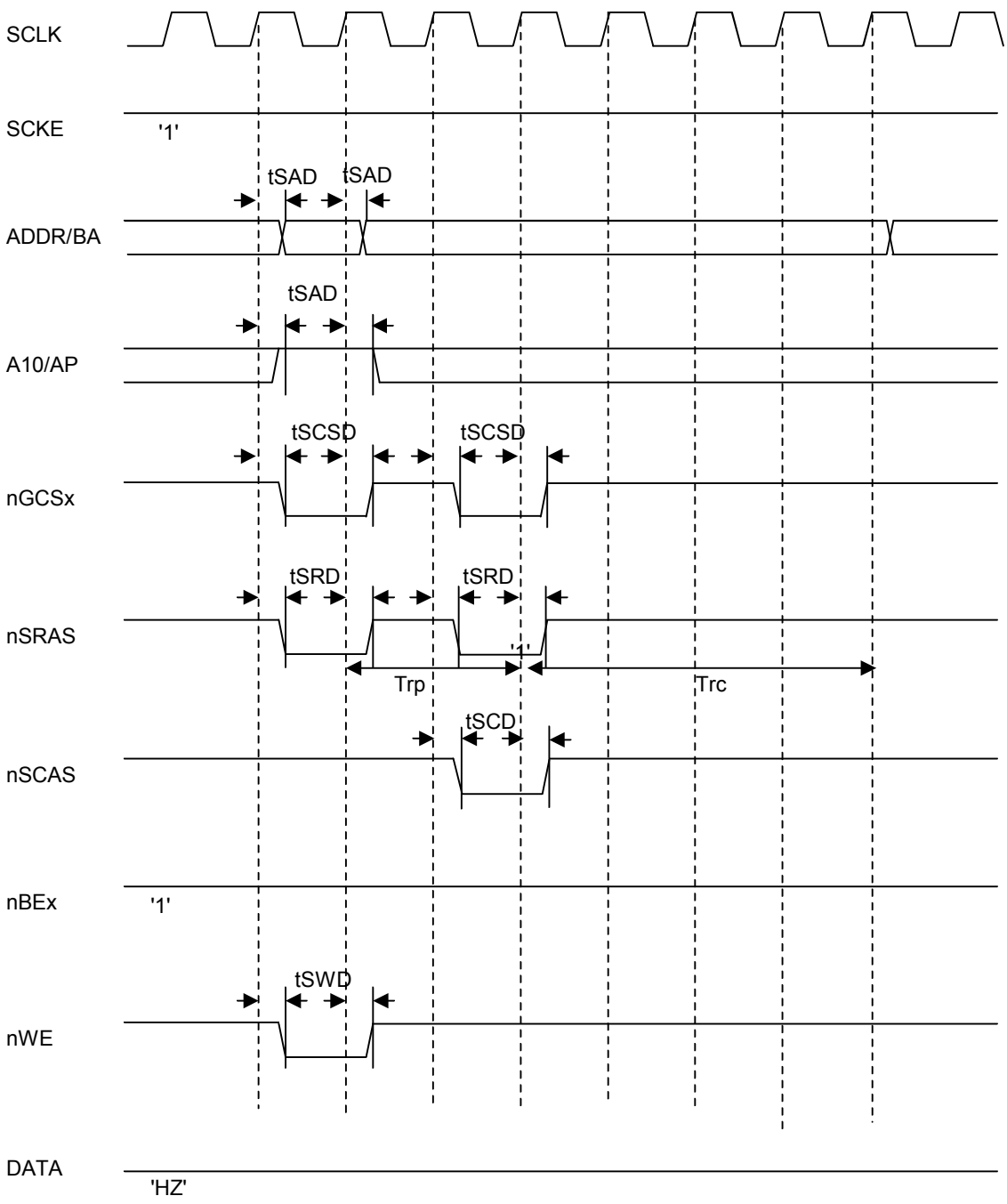



**Figure 25-24. SDRAM Single READ Timing(II) (Trp=2, Trcd=2, Tcl=3)**




**NOTE:** Before executing auto/self refresh command, all banks must be in idle state.

**Figure 25-25. SDRAM Auto Refresh Timing (Trp=2, Trc=4)**

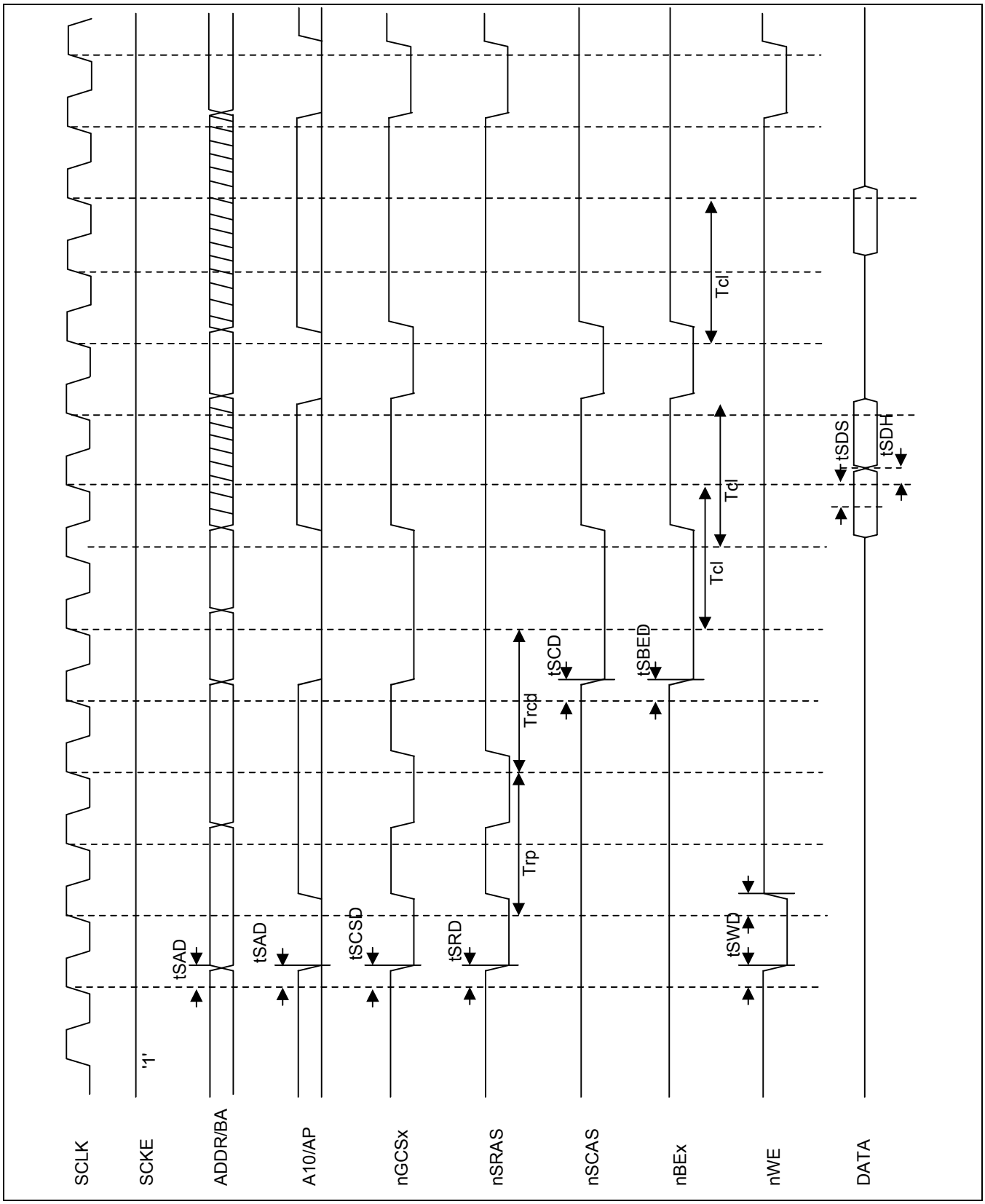
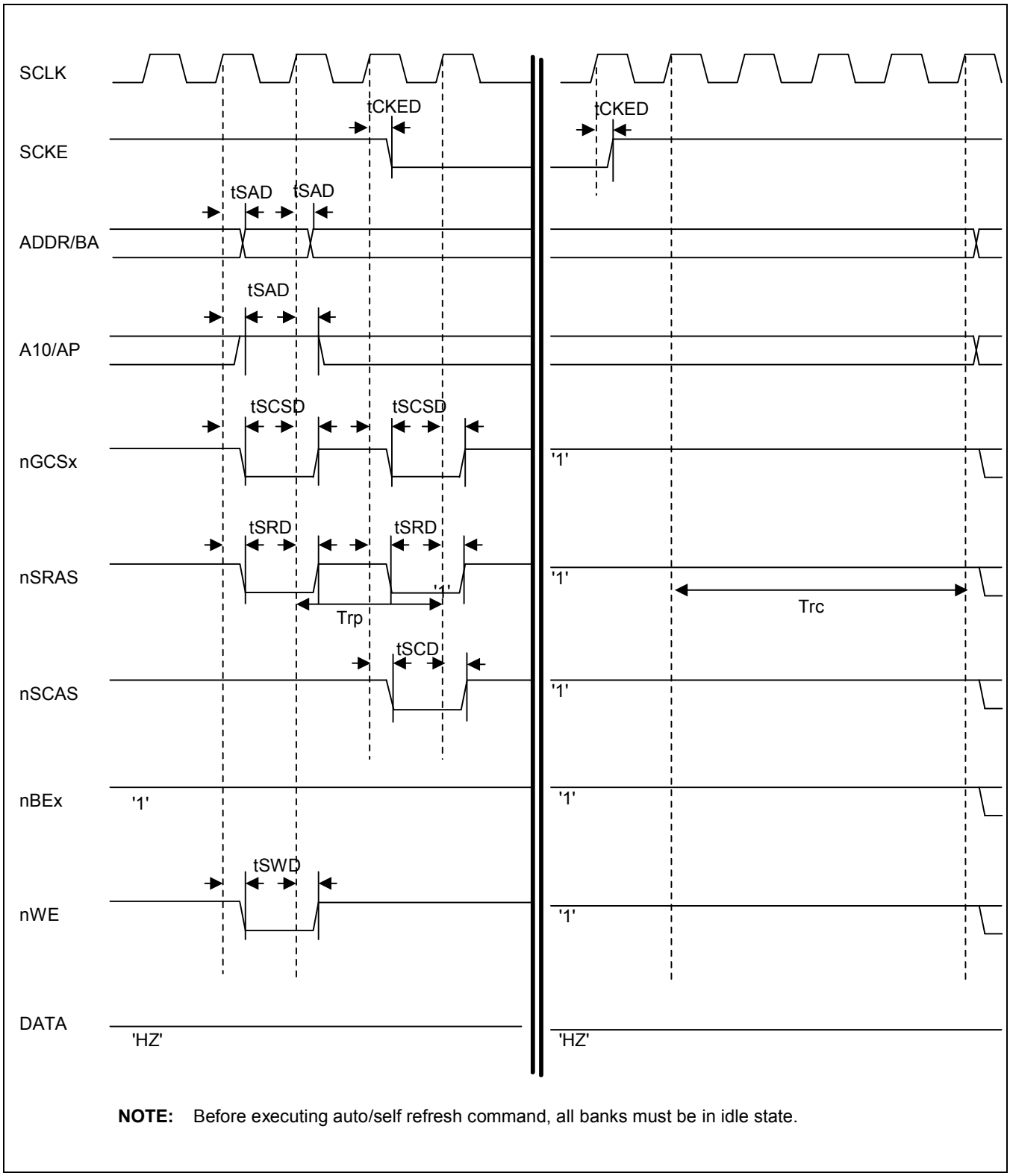



**Figure 25-26. SDRAM Page Hit-Miss READ Timing (Trp=2, Trcd=2, Tcl=2)**




**NOTE:** Before executing auto/self refresh command, all banks must be in idle state.

**Figure 25-27. SDRAM Self Refresh Timing (Trp=2, Trc=4)**

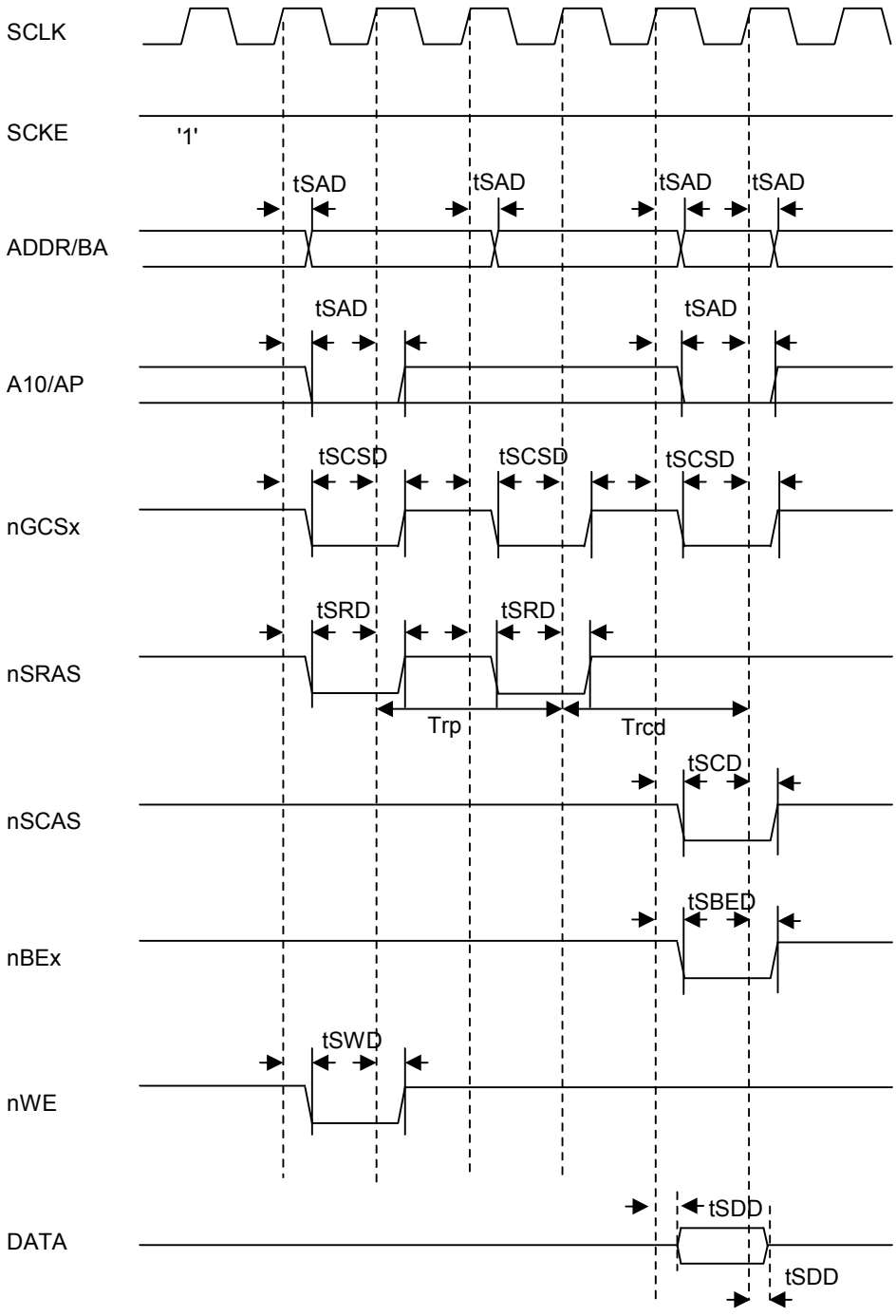



**Figure 25-28. SDRAM Single Write Timing (Trp=2, Trcd=2)**

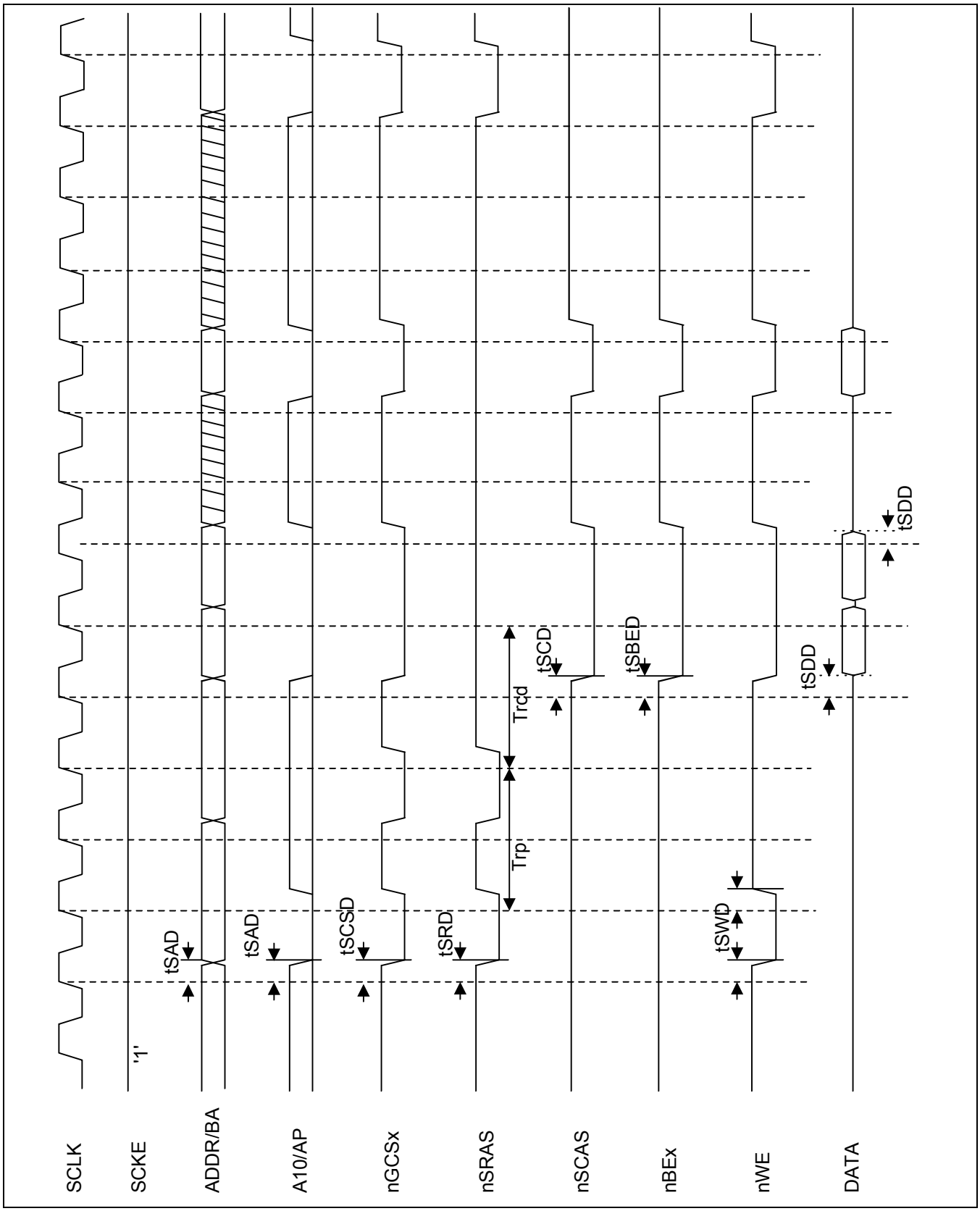



**Figure 25-29. SDRAM Page Hit-Miss Write Timing (Trp=2, Trcd=2, Tcl=2)**

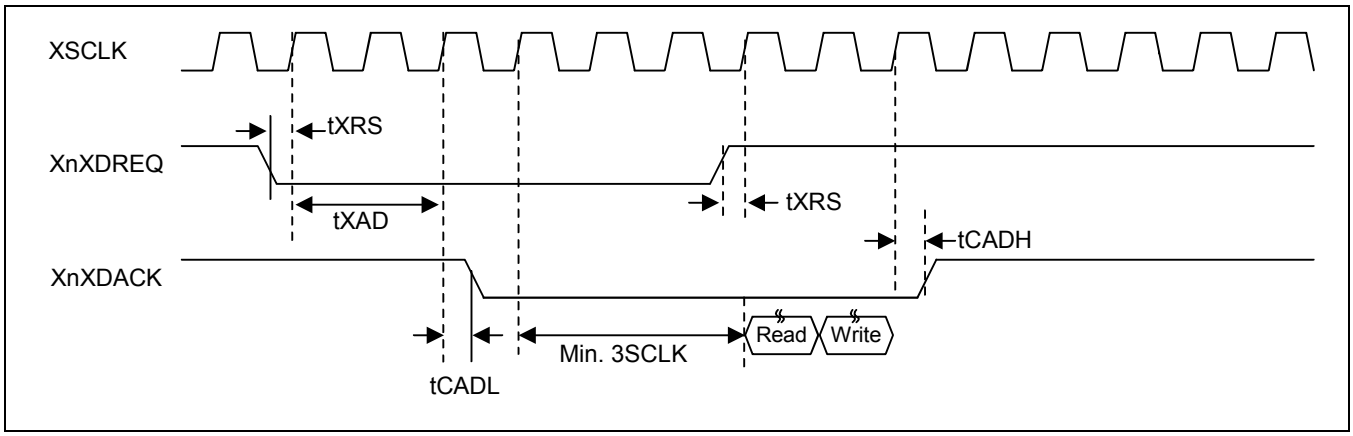



**Figure 25-30. External DMA Timing (Handshake, Single transfer)**

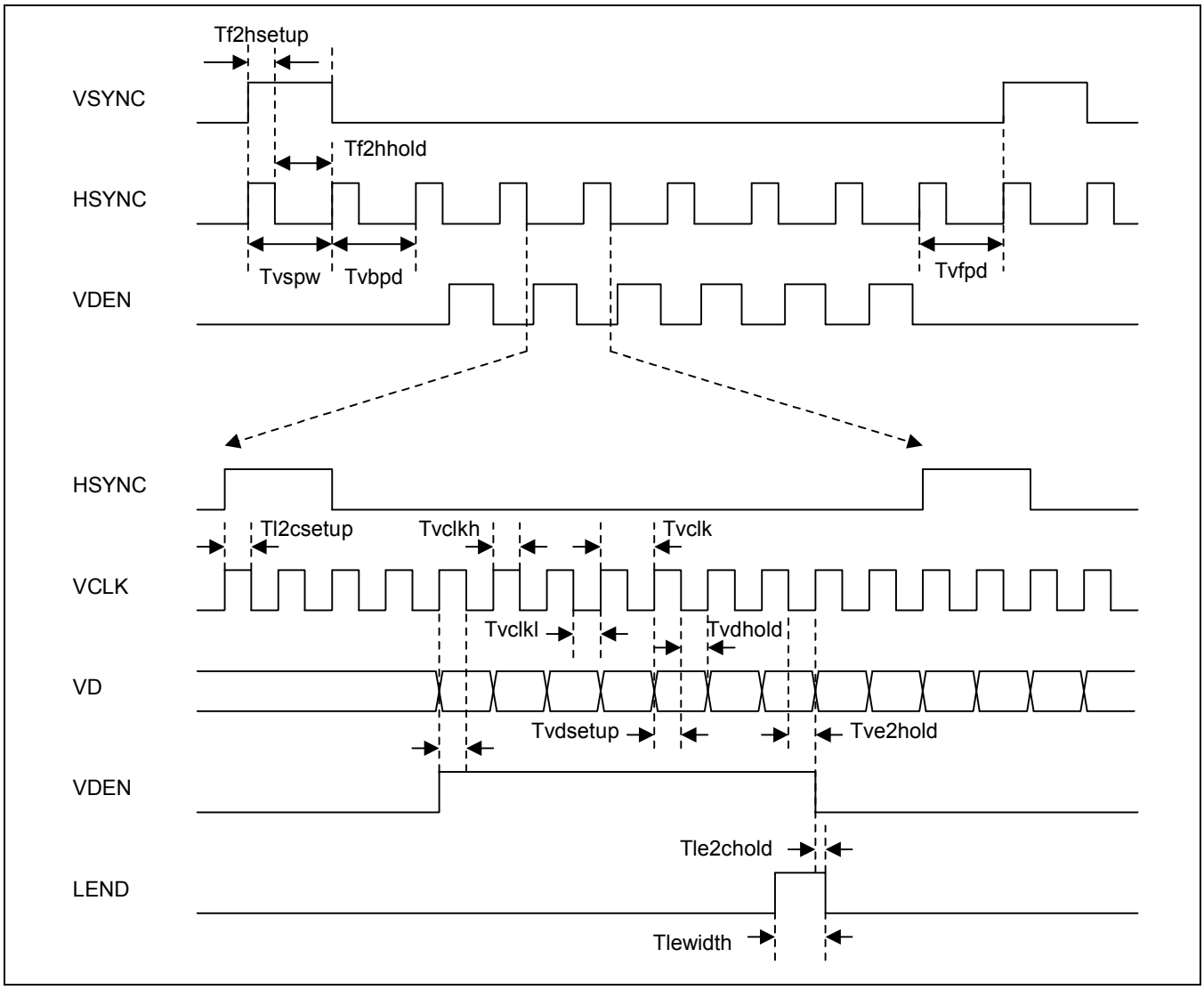



**Figure 25-31. TFT LCD Controller Timing**

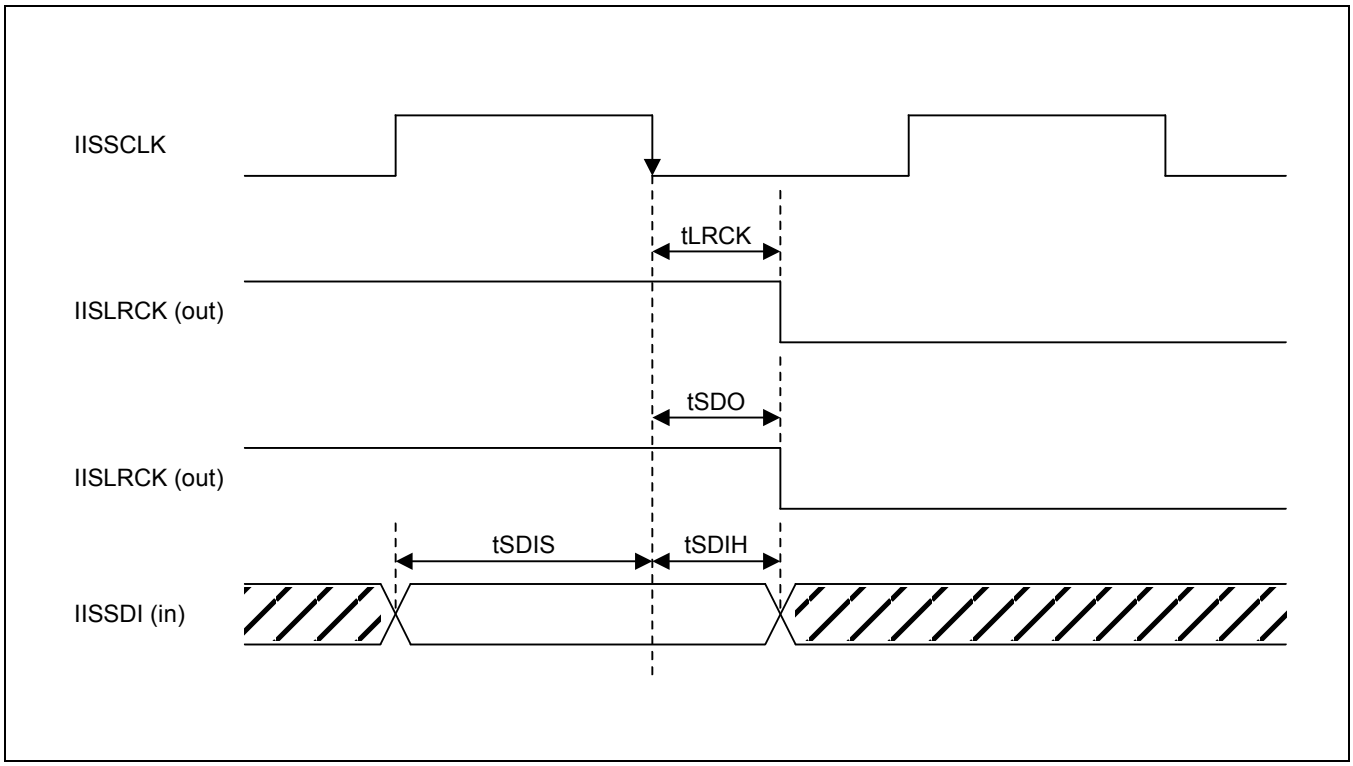



**Figure 25-32. IIS Interface Timing**

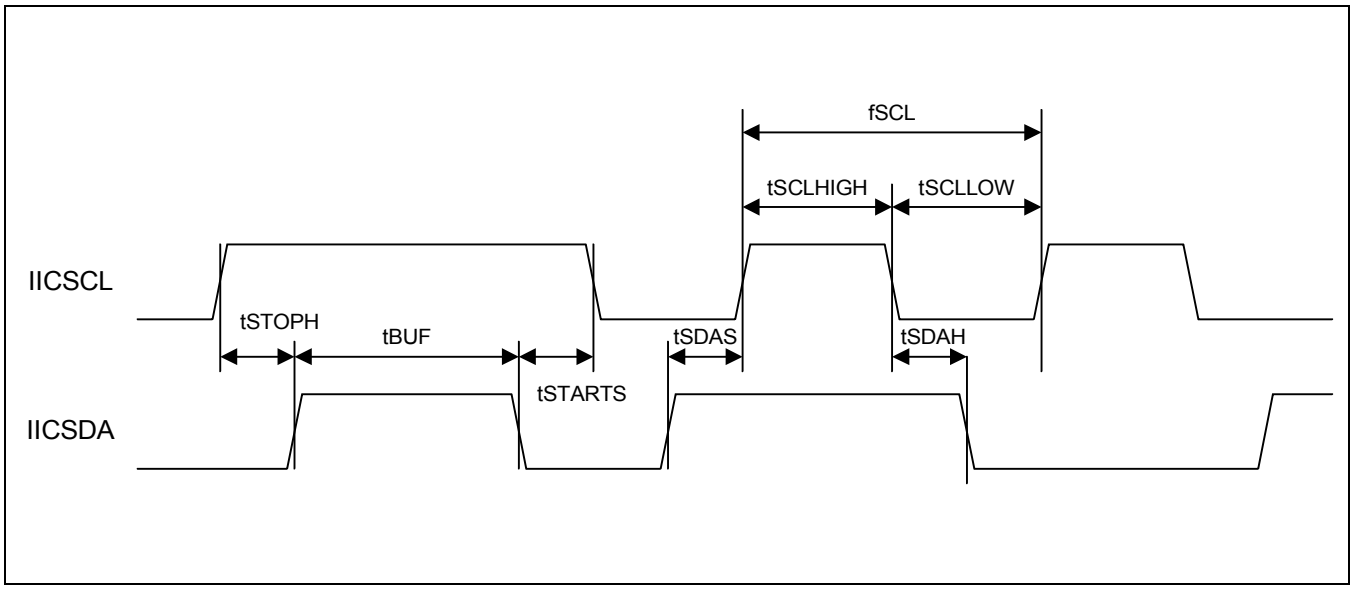



**Figure 25-33. IIC Interface Timing**

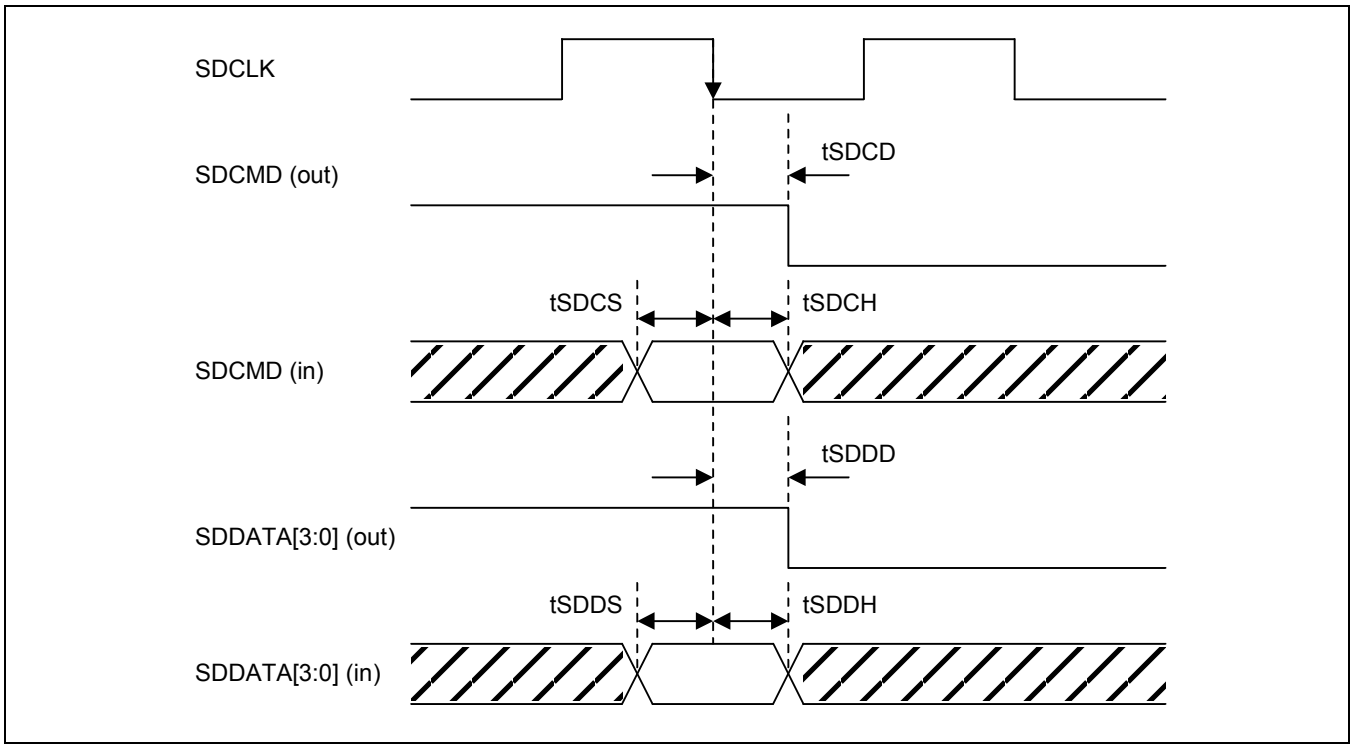



**Figure 25-34. SD/MMC Interface Timing**

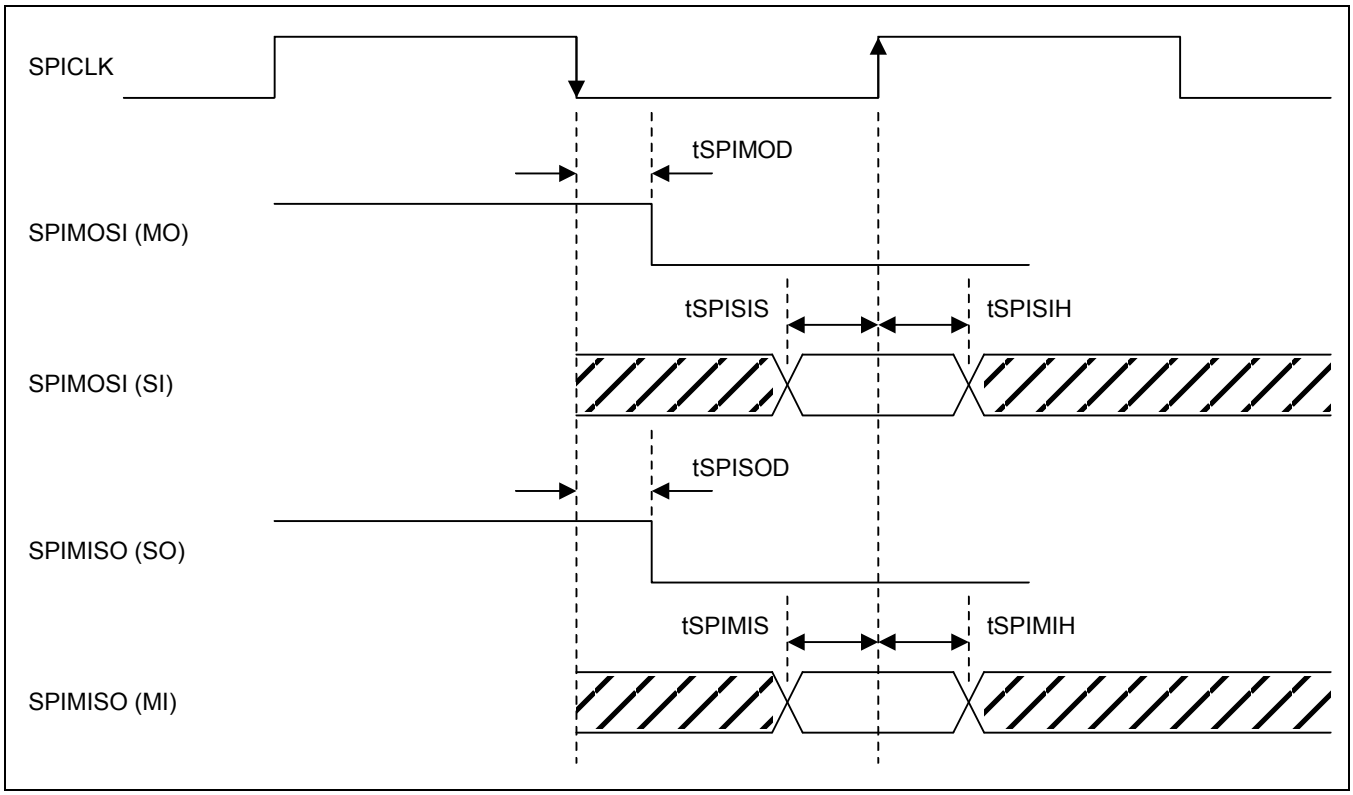



**Figure 25-35. SPI Interface Timing (CPHA=1, CPOL=1)**

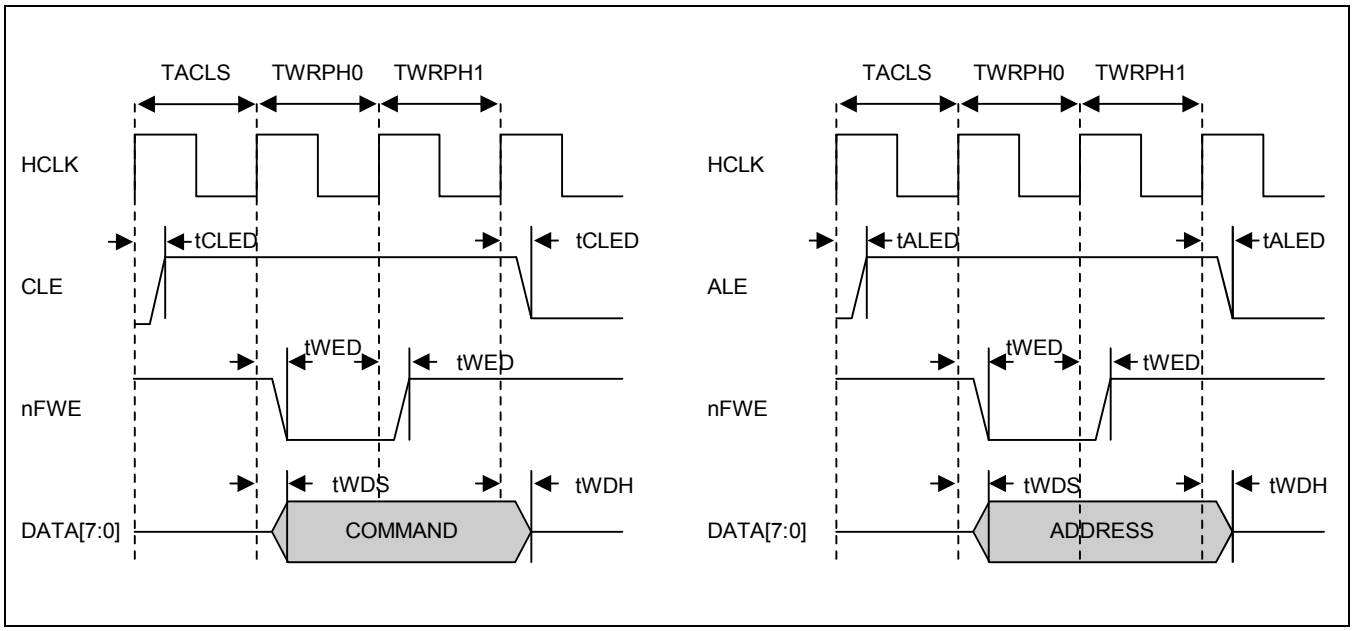



**Figure 25-36. NAND Flash Address/Command Timing**

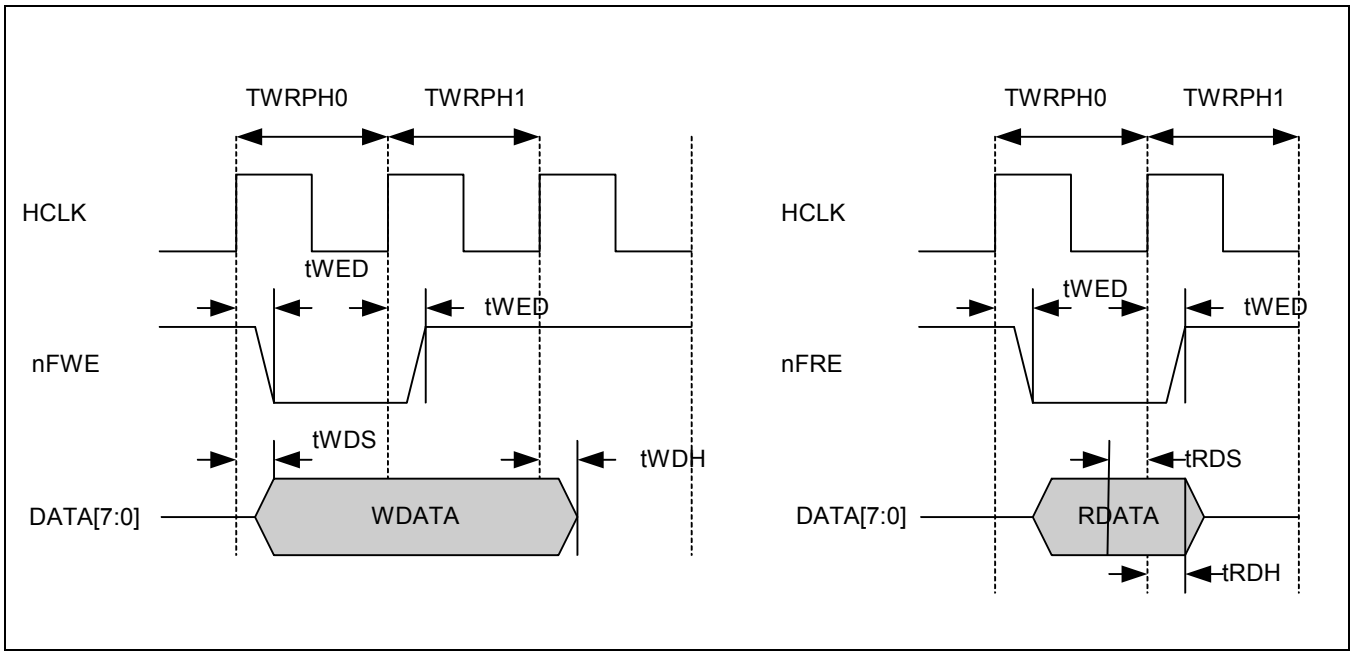



**Figure 25-37. NAND Flash Timing**

**Table 25-7. Clock Timing Constants**

($V_{DDi}$, $V_{DDalive}$, $V_{DDiarm}$ = 1.2 V ± 0.1 V, $T_A$ = 0 to 70 °C, $V_{DDMOP}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Crystal clock input frequency | $f_{XTAL}$ | 10 | – | 20 | MHz |
| Crystal clock input cycle time | $t_{XTALCYC}$ | 50 | – | 100 | ns |
| External clock input frequency | $f_{EXT}$ | – | – | 66 | MHz |
| External clock input cycle time | $t_{EXTCYC}$ | 15.0 | – | – | ns |
| External clock input low level pulse width | $t_{EXTLOW}$ | 7 | – | – | ns |
| External clock to HCLK (without PLL) | $t_{EX2HC}$ | 3 | – | 7 | ns |
| HCLK (internal) to CLKOUT | $t_{HC2CK}$ | 3 | – | 8 | ns |
| HCLK (internal) to SCLK | $t_{HC2SCLK}$ | 0 | – | 1 | ns |
| External clock input high level pulse width | $t_{EXTHIGH}$ | 4 | – | – | ns |
| Reset assert time after clock stabilization | $t_{RESW}$ | 4 | – | – | XTIpll or EXTCLK |
| PLL Lock Time | $t_{PLL}$ | 300 | – | | uS |
| Sleep mode return oscillation setting time | $t_{OSC2}$ | – | – | 65536 | XTIpll or EXTCLK |
| The interval before CPU runs after nRESET is released. | $t_{RST2RUN}$ | – | 7 | – | XTIpll or EXTCLK |

**Table 25-8. ROM/SRAM Bus Timing Constants**

($V_{DDi}$, $V_{DDalive}$, $V_{DDiarm}$ = 1.2 V ± 0.1 V, $T_A$ = 0 to 70 °C, $V_{DDMOP}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| ROM/SRAM Address Delay | $t_{RAD}$ | 3 | – | 7 | ns |
| ROM/SRAM Chip select Delay | $t_{RCD}$ | 2 | – | 7 | ns |
| ROM/SRAM Output enable Delay | $t_{ROD}$ | 2 | – | 5 | ns |
| ROM/SRAM read Data Setup time. | $t_{RDS}$ | 1 | – | 3 | ns |
| ROM/SRAM read Data Hold time. | $t_{RDH}$ | 0 | – | 1 | ns |
| ROM/SRAM Byte Enable Delay | $t_{RBED}$ | 2 | – | 6 | ns |
| ROM/SRAM Write Byte Enable Delay | $t_{RWBED}$ | 2 | – | 6 | ns |
| ROM/SRAM output Data Delay | $t_{RDD}$ | 2 | – | 7 | ns |
| ROM/SRAM external Wait Setup time | $t_{WS}$ | 1 | – | 3 | ns |
| ROM/SRAM external Wait Hold time | $t_{WH}$ | 0 | – | 1 | ns |
| ROM/SRAM Write enable Delay | $t_{RWD}$ | 3 | – | 6 | ns |

**Table 25-9. Memory Interface Timing Constants (3.3V)**

($V_{DDi}$, $V_{DDalive}$, $V_{DDiarm}$ = 1.2 V ± 0.1 V, $T_A$ = 0 to 70 °C, $V_{DDMOP}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SDRAM Address Delay | $t_{SAD}$ | 2 | – | 7 | ns |
| SDRAM Chip Select Delay | $t_{SCSD}$ | 2 | – | 5 | ns |
| SDRAM Row active Delay | $t_{SRD}$ | 2 | – | 5 | ns |
| SDRAM Column active Delay | $t_{SCD}$ | 2 | – | 5 | ns |
| SDRAM Byte Enable Delay | $t_{SBED}$ | 2 | – | 6 | ns |
| SDRAM Write enable Delay | $t_{SWD}$ | 2 | – | 6 | ns |
| SDRAM read Data Setup time | $t_{SDS}$ | 1 | – | 3 | ns |
| SDRAM read Data Hold time | $t_{SDH}$ | 0 | – | 1 | ns |
| SDRAM output Data Delay | $t_{SDD}$ | 2 | – | 8 | ns |
| SDRAM Clock Enable Delay | $T_{cked}$ | 2 | – | 5 | ns |

**Table 25-10. External Bus Request Timing Constants**

($V_{DD}$ = 1.2 V ± 0.1 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|
| eXternal Bus Request Setup time | $t_{XnBRQS}$ | 2 | – | 4 | ns |
| eXternal Bus Request Hold time | $t_{XnBRQH}$ | 0 | – | 1 | ns |
| eXternal Bus Ack Delay | $t_{XnBACKD}$ | 4 | – | 10 | ns |
| HZ Delay | $t_{HZD}$ | 3 | – | 8 | ns |

**Table 25-11. DMA Controller Module Signal Timing Constants**

($V_{DD}$ = 1.2 V ± 0.1 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|
| eXternal Request Setup | $t_{XRS}$ | 2 | – | 4 | ns |
| aCcess to Ack Delay when Low transition | $t_{CADL}$ | 4 | – | 9 | ns |
| aCcess to Ack Delay when High transition | $t_{CADH}$ | 4 | – | 9 | ns |
| eXternal Request Delay | $t_{XAD}$ | 2 | – | – | SCLK |

**Table 25-12. TFT LCD Controller Module Signal Timing Constants**

($V_{DD}$ = 1.8 V ± 0.15 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Vertical sync pulse width | Tvspw | VSPW + 1 | – | – | Phclk (note1) |
| Vertical back porch delay | Tvbpd | VBPD+1 | – | – | Phclk |
| Vertical front porch dealy | Tvfpd | VFPD+1 | – | – | Phclk |
| VCLK pulse width | Tvclk | 1 | – | – | Pvclk (note2) |
| VCLK pulse width high | Tvclkh | 0.5 | – | – | Pvclk |
| VCLK pulse width low | TvclkI | 0.5 | – | – | Pvclk |
| Hsync setup to VCLK falling edge | Tl2csetup | 0.5 | – | – | Pvclk |
| VDEN set up to VCLK falling edge | Tde2csetup | 0.5 | – | – | Pvclk |
| VDEN hold from VCLK falling edge | Tde2chold | 0.5 | – | – | Pvclk |
| VD setup to VCLK falling edge | Tvd2csetup | 0.5 | – | – | Pvclk |
| VD hold from VCLK falling edge | Tvd2chold | 0.5 | – | – | Pvclk |
| VSYNC setup to HSYNC falling edge | Tf2hsetup | HSPW + 1 | – | – | Pvclk |
| VSYNC hold from HSYNC falling edge | Tf2hhold | HBPD + HFPD + HOZVAL + 3 | – | – | Pvclk |

**NOTES:**
1. HSYNC period
2. VCLK period

**Table 25-13. IIS Controller Module Signal Timing Constants**

($V_{DD}$ = 1.2 V ± 0.1 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|
| IISLRCK delay time | $t_{LRCK}$ | 0 | – | 3 | ns |
| IISDO delay time | $t_{SDO}$ | 0 | – | 2 | ns |
| IISDI Input Setup time | $t_{SDIS}$ | 5 | – | 10 | ns |
| IISDI Input Hold time | $t_{SDIH}$ | 0 | – | 1 | ns |
| CODEC clock frequency | $f_{CODEC}$ | 1/16 | – | 1 | $f_{IIS\_BLOCK}$ |

**Table 25-14. IIC BUS Controller Module Signal Timing**

($V_{DD}$ = 1.8 V ± 0.15 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|
| SCL clock frequency | $f_{SCL}$ | – | – | std. 100<br>fast 400 | kHz |
| SCL high level pulse width | $t_{SCLHIGH}$ | std. 4.0<br>fast 0.6 | – | – | μs |
| SCL low level pulse width | $t_{SCLLOW}$ | std. 4.7<br>fast 1.3 | – | – | μs |
| Bus free time between STOP and START | $t_{BUF}$ | std. 4.7<br>fast 1.3 | – | – | μs |
| START hold time | $t_{STARTS}$ | std. 4.0<br>fast 0.6 | – | – | μs |
| SDA hold time | $t_{SDAH}$ | std. 0<br>fast 0 | – | std. - fast<br>0.9 | μs |
| SDA setup time | $t_{SDAS}$ | std. 250<br>fast 100 | – | – | ns |
| STOP setup time | $t_{STOPH}$ | std. 4.0<br>fast 0.6 | – | – | μs |

**NOTES:** Std. means Standard Mode and fast means Fast Mode.

1. The IIC data hold time(tSDAH) is minimum 0ns.
   (IIC data hold time is minimum 0ns for standard/fast bus mode in IIC specification v2.1.)
   Please check the data hold time of your IIC device if it's 0 nS or not.
2. The IIC controller supports only IIC bus device(standard/fast bus mode), not C bus device.

**Table 25-15. SD/MMC Interface Transmit/Receive Timing Constants**

($V_{DD}$ = 1.2 V ± 0.1 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|
| SD Command output Delay time | $t_{SDCD}$ | 0 | – | 1 | ns |
| SD Command input Setup time | $t_{SDCS}$ | 5 | – | 11 | ns |
| SD Command input Hold time | $t_{SDCH}$ | 0 | – | 1 | ns |
| SD Data output Delay time | $t_{SDDD}$ | 0 | – | 1 | ns |
| SD Data input Setup time | $t_{SDDS}$ | 5 | – | 11 | ns |
| SD Data input Hold time | $t_{SDDH}$ | 0 | – | 1 | ns |

**Table 25-16. SPI Interface Transmit/Receive Timing Constants**

($V_{DD}$ = 1.2 V ± 0.1 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|
| SPI MOSI Master Output Delay time | $t_{SPIMOD}$ | 0 | – | 1 | ns |
| SPI MOSI Slave Input Setup time | $t_{SPISIS}$ | 0 | – | 1 | ns |
| SPI MOSI Slave Input Hold time | $t_{SPISIH}$ | 0 | – | 1 | ns |
| SPI MISO Slave output Delay time | $t_{SPISOD}$ | 6 | – | 18 | ns |
| SPI MISO Master Input Setup time | $t_{SPIMIS}$ | 5 | – | 15 | ns |
| SPI MISO Master Input Hold time | $t_{SPIMIH}$ | 0 | – | 1 | ns |

**Table 25-17. USB Electrical Specifications**

($V_{DD}$ = 1.8 V ± 0.15 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| **Supply Current** | | | | | |
| Suspend Device | ICCS | | | 10 | µA |
| **Leakage Current** | | | | | |
| Hi-Z state Input Leakage | ILO | 0V < VIN < 3.3V | -10 | 10 | µA |
| **Input Levels** | | | | | |
| Differential Input Sensitivity | VDI | \| (D+) – (D-) \| | 0.2 | | V |
| Differential Common Mode Range | VCM | Includes VDI range | 0.8 | 2.5 | |
| Single Ended Receiver Threshold | VSE | | 0.8 | 2.0 | |
| **Output Levels** | | | | | |
| Static Output Low | VOL | RL of 1.5Kohm to 3.6V | | 0.2 | V |
| Static Output High | VOH | RL of 15Kohm to GND | 2.8 | 3.6 | |
| **Capacitance** | | | | | |
| Transceiver Capacitance | CIN | Pin to GND | | 20 | pF |

**Table 25-18. USB Full Speed Output Buffer Electrical Characteristics**

($V_{DD}$ = 1.8 V ± 0.15 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| **Driver Characteristics** | | | | | |
| Transition Time<br>Rise Time<br>Fall Time | <br>TR<br>TF | <br>CL = 50pF<br>CL = 50pF | <br>4.0<br>4.0 | <br>20<br>20 | <br>ns<br> |
| Rise/Fall Time Matching | Trfm | (TR / TF ) | 90 | 111.1 | % |
| Output Signal Crossover Voltage | Vcrs | | 1.3 | 2.0 | V |
| Drive Output Resistance | Zdrv | Steady state drive | 28 | 44 | ohm |

**Table 25-19. USB Low Speed Output Buffer Electrical Characteristics**

($V_{DD}$ = 1.8 V ± 0.15 V, $T_A$ = 0 to 70 °C, $V_{EXT}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| **Driver Characteristics** | | | | | |
| Rising Time | TR | CL = 50pF | 75 | | ns |
| | | CL = 350pF | | 300 | |
| Falling Time | TF | CL = 50pF | 75 | | |
| | | CL = 350pF | | 300 | |
| Rise/Fall Time Matching | Trfm | (Tr / Tf ) | 80 | 125 | % |
| Output Signal Crossover Voltage | Vcrs | | 1.3 | 2.0 | V |

**Note:** All measurement conditions are in accordance with the Universal Serial Bus Specification 1.1 Final Draft Revision.

**Table 25-20. NAND Flash Interface Timing Constants**

($V_{DDi}$, $V_{DDalive}$, $V_{DDiarm}$ = 1.8 V ± 0.15 V, $T_A$ = 0 to 70 °C, $V_{DDIO}$ = 3.3V ± 0.3V)

| Parameter | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| NFCON Chip Enable delay | $t_{CED}$ | – | 5.3 | ns |
| NFCON CLE delay | $t_{CLED}$ | – | 5.7 | ns |
| NFCON ALE delay | $t_{ALED}$ | – | 5.5 | ns |
| NFCON Write Enable delay | $t_{WED}$ | – | 5.6 | ns |
| NFCON Read Enable delay | $t_{RED}$ | – | 5.9 | ns |
| NFCON Write Data Setup time | $t_{WDS}$ | – | 5.4 | ns |
| NFCON Write Data Hold time | $t_{WDH}$ | 3.7 | 5.5 | ns |
| NFCON Read Data Setup requirement time | $t_{RDS}$ | 3 | – | ns |
| NFCON Read Data Hold requirement time | $t_{RDH}$ | 0.3 | – | ns |

# NOTES

# 26 MECHANICAL DATA

## PACKAGE DIMENSIONS

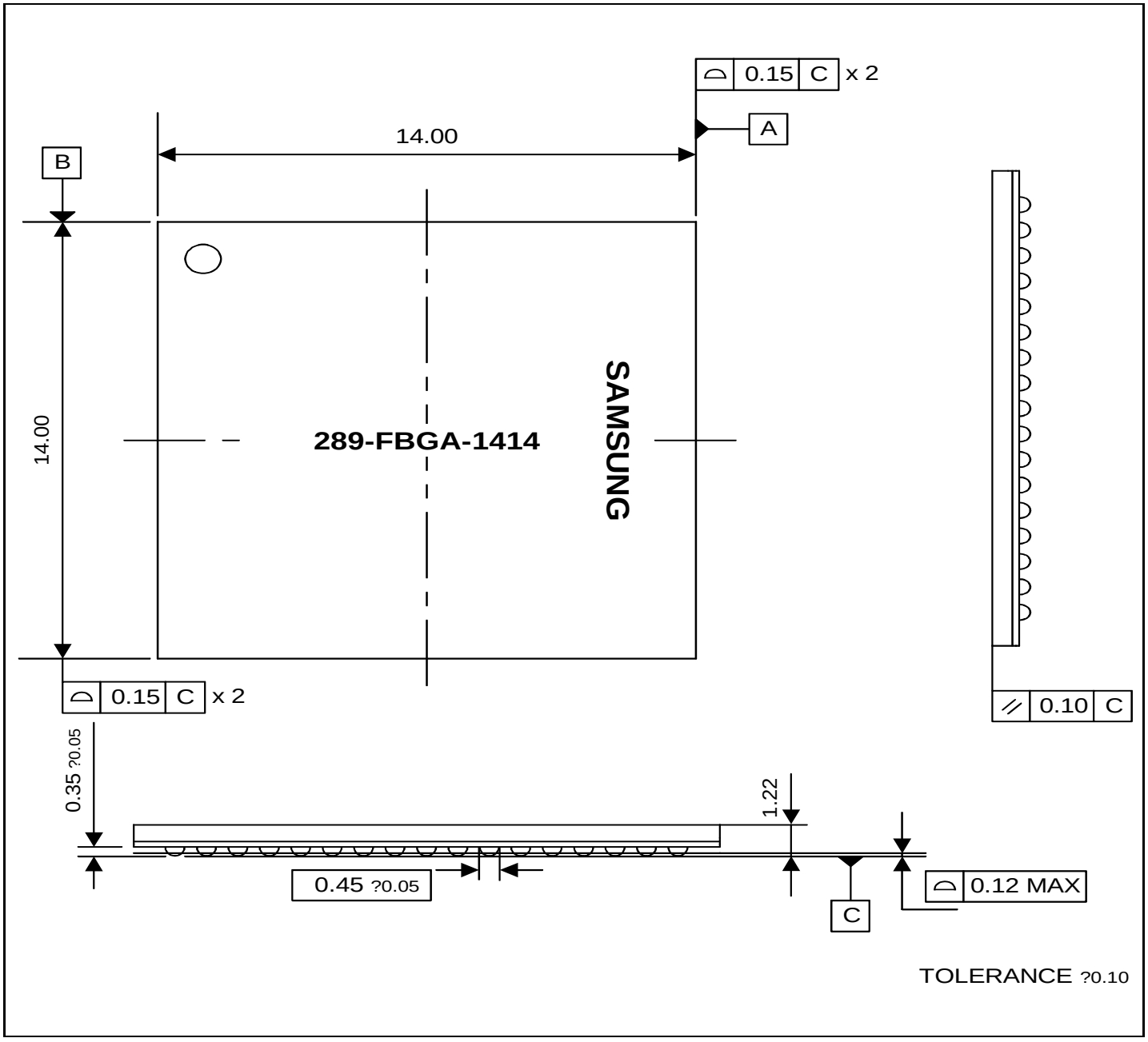



**Figure 26-1. 289-FBGA-1414 Package Dimension 1 (Top View)**

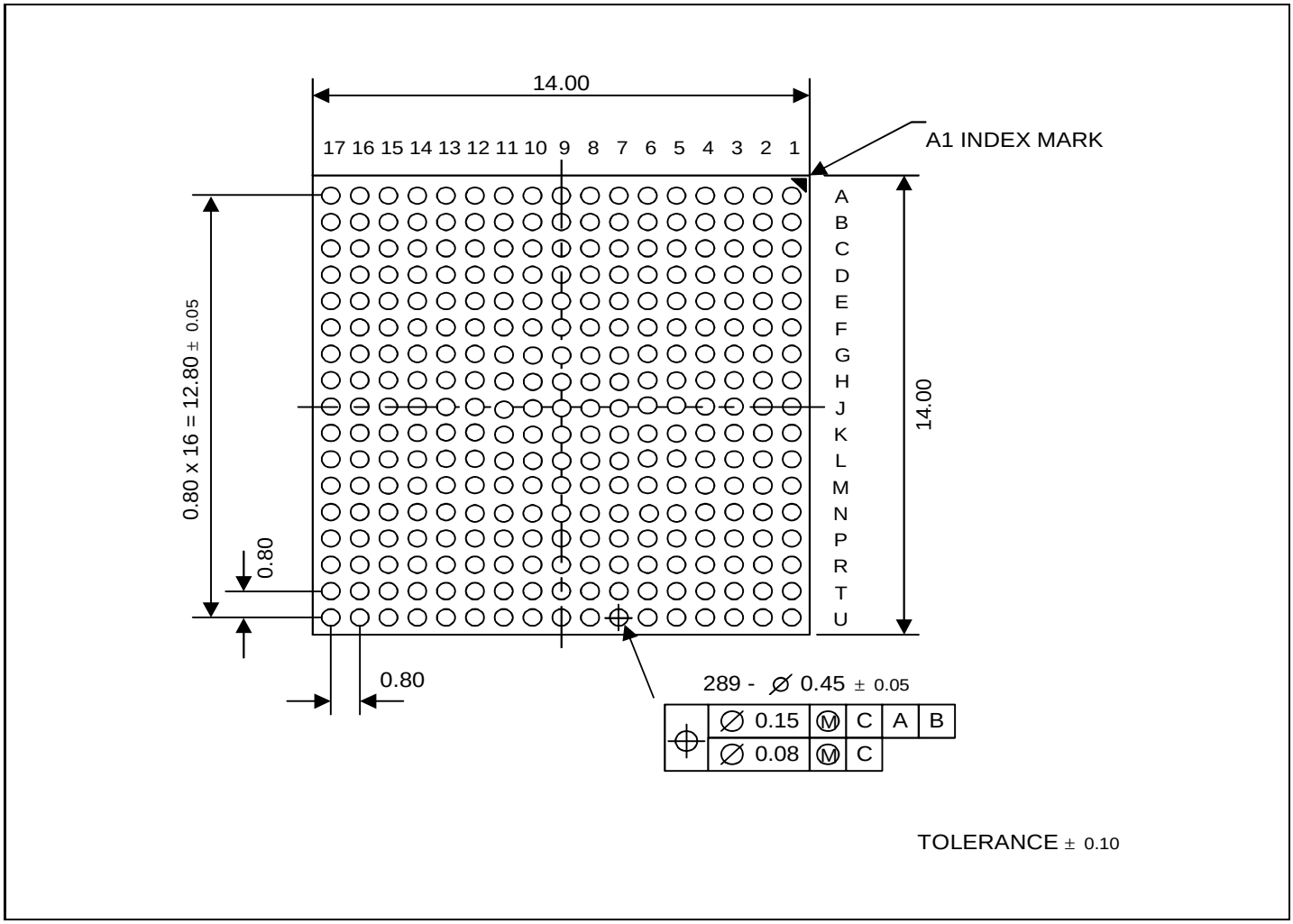



**Figure 26-2. 289-FBGA-1414 Package Dimension 2 (Bottom View)**

The recommended land open size is 0.39 – 0.41mm diameter.