USER'S MANUAL

TCC720

32-bit RISC

Microprocessor

For

Digital Media Player

Preliminary Rev 0.51

TelleChips

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1 INTRODUCTION

TCC720 is system LSI for digital media player which based on ARM940T, ARM's proprietary 32-bit RISC CPU core. It can decode and encode MP3 or other types of audio/voice compression / decompression standards by software based architecture.

The on-chip USB controller enables the data transmission between a personal computer and storage of device such as NAND flash, HDD, CD etc, which can be controlled by TCC720.

TCC720 also includes on-chip stereo audio CODEC eliminates the need of expensive external audio CODEC. Using I2S port, TCC720 can also use the external audio CODEC by performance or other reason.

1.1 FEATURES

- 32bit ARM940TDMI RISC CPU core
- 8KB instruction/data cache
- Internal boot ROM of 4Kbytes for various boot procedure(NAND, UART) and security
- Internal SRAM of 64K bytes for general usage
- On-chip peripherals
  - Memory controller for various memories such as PROM, FLASH, SRAM, SDRAM
  - IDE Interface for HDD or USB device
  - 4 external interrupts, 9 internal interrupts
  - 4 timer/counters, 2 timers
  - USB1.1 device (Full speed)
  - UART(IrDA) for serial Host I/F
  - GPIO, GSIO
  - I2S interface for internal and external audio CODEC
  - I2S interface for CD-DSP
  - On chip audio CODEC with MIC input
  - General purpose ADC (3 input)
  - 1 Channel DMA for transferring a bulk of data
  - JTAG interface for code debugging
- 0.25um low power CMOS process
- 2.5V for core, 3.3V for I/O port
- 128-pin TQFP
- Operating up to 120MHz

1.2 APPLICATIONS

- Portable Digital Audio Encoder/Decoder
- MP3 Juke Box
- Digital Audio Encoder/Decoder
- Digital Internet Radio Server
- Multimedia Storage Device

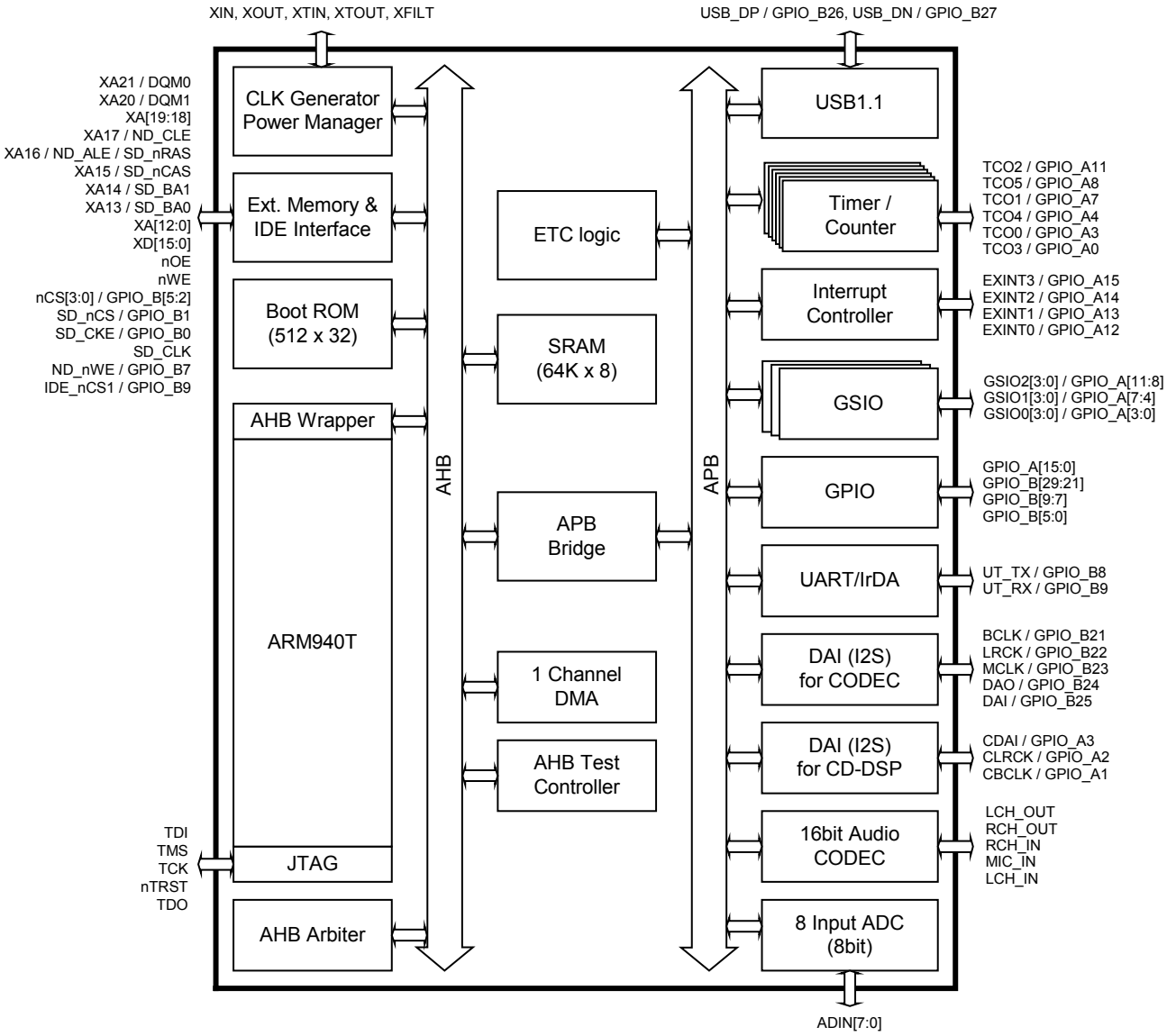


Figure 1.1 Functional Block Diagram

1.2 Pin Description

JTAG Interface

Signal Name	NUM	Type	Description
TDI	99	I	JTAG serial data input for ARM940T
TMS	100	I	JTAG Test mode select for ARM940T
TCK	101	I	JTAG test clock for ARM940T
TDO	102	O	JTAG serial data output for ARM940T
nTRST	103	I	Reset signal for boundary scan logic. Active low.

External Memory Interface

Signal	Shared Signal	NUM	Type	Description
SD_CKE	GPIO_B0	56	O	Clock enable signal for SDRAM, Active high. / GPIO_B1
SD_CLK	GPO	44	O	SDRAM clock
SD_nCS	GPIO_B1	46	O	Chip select signal for SDRAM, Active low. / GPIO_B0
XA[6:0]	-	23:17	O	Address bus for external memories.
XA[12:7]	-	31:26	O	
XA[13]	SD_BA0	34	O	Bank Address 0 for SDRAM / XA[13]
XA[14]	SD_BA1	35	O	Bank Address 1 for SDRAM / XA[14]
XA[15]	SD_nCAS	36	O	CAS for SDRAM / XA[15]
XA[16]	ND_ALE, SD_nRAS	37	O	ALE for NAND flash / RAS for SDRAM / XA[16]
XA[17]	ND_CLE	38	O	CLE for NAND flash / XA[17]
XA[19:18]	-	40:39	O	XA[19:18] for static memory / Bus Width configuration
XA[21:20]	DQM[0:1]	43:42	O	XA[21:20] / Data I/O mask
XD[15:9], XD[8:4], XD[3:0]	XD[15:9], XD[8:4], XD[3:0]	15:9, 6:2, 128:125	I/O	Data bus for external memory
nCS0 / ND_nOE0	GPIO_B2	47	O	External chip select 0 / NAND flash 0 OE / GPIO_B2
nCS1 / ND_nOE1	GPIO_B3	48	O	External chip select 1 / NAND flash 1 OE / GPIO_B3
nCS2 / ND_nOE2	GPIO_B4	49	O	External chip select 2 / NAND flash 2 OE / GPIO_B4
nCS3 / ND_nOE3	GPIO_B5	50	O	External chip select 3 / NAND flash 3 OE / GPIO_B5
IDE_nCS1	GPIO_B9 / UT_RX	61	O	IDE chip select 1. Active low. / GPIO_B9 / UART RX
ND_nWE	GPIO_B7	57	O	NAND flash WE. Active low. / GPIO_B7
nWE	nWE	58	O	Static memory write enable signal. Active low.
nOE	nOE	59	O	Static memory output enable signal. Active low.
READY	-	73	I	Ready information from external device.

*) XA[21:0] is used as system address bus for external memories such as SRAM, ROM.
XA[12:0] can be also used as RAS and CAS signals for SDRAM.
XD[15:0] is used as system data bus for all types of external memories contained.
SD_CLK is also used as general purpose output port by setting clock control flag. Refer to the chapter of memory controller for detail.

General Purpose I/O

Signal	Shared Signal	NUM	Type	Description
GPIO_A[15:12]	EXINT[3:0]	124:121	I/O	GPIO_A[15:12] / External Interrupt Source 3 ~ 0
GPIO_A[11:8]	GSIO2[3:0] (SDI_2, FRM_2, SCK_2, SDO_2)	118:115	I/O	GPIO_A[11:8] / General Purpose Serial I/O 2
GPIO_A[7:4]	GSIO1[3:0] (SDI_1, FRM_1, SCK_1, SDO_1)	114:113 111 108	I/O	GPIO_A[7:4] / General Purpose Serial I/O 1
GPIO_A[3:1]	GSIO0[3:1] (SDI_0, FRM_0, SCK_0) / CDIF[2:0] (CDAI, CLRCK, CBCLK)	107:105	I/O	GPIO_A[3:1] / General purpose serial I/O 0 / CD interface signals
GPIO_A[0]	GSIO0[0] (SDO_0)	104	I/O	GPIO_A[0] / General purpose serial out 0
GPIO_B[29:28]	-	54:53	I/O	GPIO_B[29:28]
GPIO_B[27:26]	USB_DP, USB_DN	52:51	I/O	GPIO_B[27:26] / USB_DP, USB_DN
GPIO_B[25:21]	DAI DAO MCLK LRCK BCLK	68 67 66 63 62	I/O	GPIO_B[25:21] / I2S Interface Signals
GPIO_B[9:8]	UT_RX / IDE_nCS1 UT_TX	61 60	I/O	GPIO_B[9:8] / UART Interface Signals IDE chip select 1
GPIO_B7	ND_nWE	57	I/O	GPIO_B7 / Write enable for NAND flash
GPIO_B[5:2]	nCS[3:0]	50:47	I/O	GPIO_B[5:2] / External Chip Select 3 ~ 0
GPIO_B1	SD_nCS	46	I/O	GPIO_B1 / Chip select for SDRAM
GPIO_B0	SD_CKE	56	I/O	GPIO_B0 / SDRAM clock control

USB / UART / IrDA Interface

Signal	Shared Signal	NUM	Type	Description
USB D+	GPIO_B26	51	I/O	USB Function D+ pin / GPIO_B26
USB D-	GPIO_B27	52	I/O	USB Function D- pin / GPIO_B27
UART_TXD	GPIO_B8	60	I/O	UART or IrDA TX data pin / GPIO_B8
UART_RXD	GPIO_B9 / IDE_nCS1	61	I/O	UART or IrDA RX data pin / GPIO_B9 IDE chip select 1

Audio Interface

Signal	Shared Signal	NUM	Type	Description
GPIO_B21	BCLK	62	I/O	I2S Bit clock (64fs) / GPIO_B21
GPIO_B22	LRCK	63	I/O	I2S Word clock / GPIO_B22
GPIO_B23	MCLK	66	I/O	I2S system clock (256fs or 384fs) / GPIO_B23
GPIO_B24	DAO	67	I/O	I2S digital audio data output / GPIO_B24
GPIO_B25	DAI	68	I/O	I2S digital audio data input / GPIO_B25
LCH_IN	LCH_IN	90	I	ADC left channel input of internal audio CODEC
RCH_IN	RCH_IN	91	I	ADC right channel input of internal audio CODEC
MIC_IN	MIC_IN	92	I	Mic input of internal audio CODEC
LCH_OUT	LCH_OUT	93	O	DAC left channel output of internal audio CODEC
RCH_OUT	RCH_OUT	94	O	DAC right channel output of internal audio CODEC
VREF	VREF	95	O	Reference voltage of internal audio CODEC

CD DSP Interface

Signal	Shared Signal	NUM	Type	Description
CBCLK	GPIO_A1	105	I/O	CD Data Bit Clock Input / GPIO_A1
CLRCK	GPIO_A2	106	I/O	CD Data Word Clock Input / GPIO_A2
CDAI	GPIO_A3	107	I/O	CD Data input / GPIO_A3

Clock Interface

Signal	Shared Signal	NUM	Type	Description
XIN	-	74	I	Main clock input for PLL
XOUT	-	75	O	Main clock output for PLL
XFILT	-	78	O	PLL filter output
XTIN	-	69	I	Sub clock input
XTOUT	-	70	O	Sub clock output

External Interrupt Interface

Signal	Shared Signal	NUM	Type	Description
EXINT[3:0]	GPIO_A15 GPIO_A14 GPIO_A13 GPIO_A12	124 123 122 121	I/O	External interrupt request [3:0] / GPIO_A[15:12]

General Purpose ADC Interface

Signal	Shared Signal	NUM	Type	Description
ADIN[2:0]	-	84:82	I	General purpose multi-channel ADC input

Mode Control

Signal	Shared Signal	NUM	Type	Description
MODE1	-	98	I	Mode Setting Input 1
nRESET	-	72	I	System Reset

Power

Signal	NUM	Type	Description
VDD3	112 76 64 33 16	PWR	Digital Power for I/O (3.3V)
VDD2D	119 109 87 71 41 24 7	PWR	Digital Power for Internal (2.5V)
VDD2A	89 81 77	PWR	Analog Power (2.5V)
VSS3D	97 65 45 32 1	PWR	Digital Ground for I/O
VSS2D	120 110 88 55 25 8	PWR	Digital Ground for Internal
VSSA	96 86 85 80 79	PWR	Analog Ground

1.3 Package Diagram

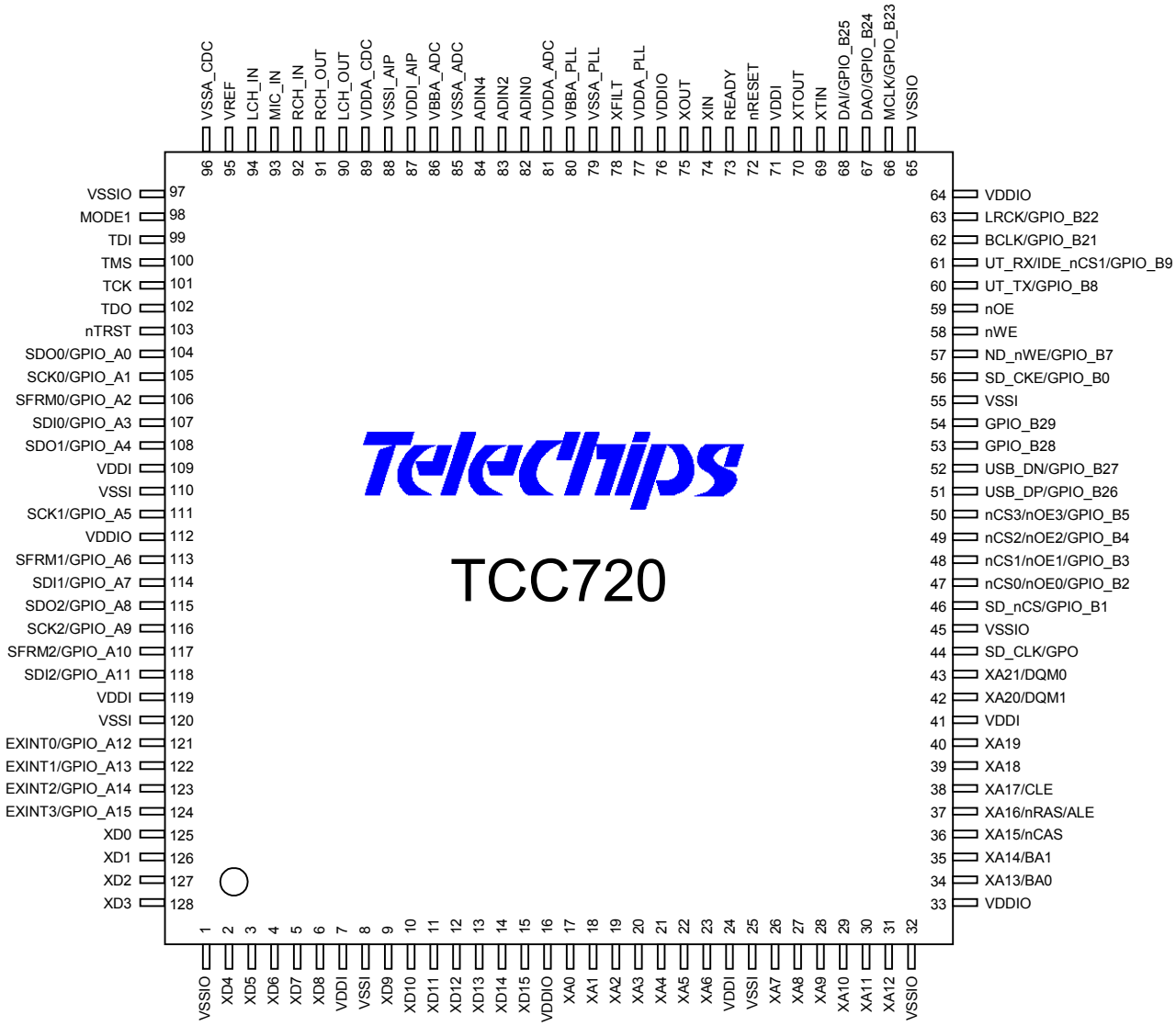


Figure 1.2 Package Diagram (128-TQFP-1414)

CHAPTER 2

ADDRESS & REGISTER MAP

2 ADDRESS & REGISTER MAP

2.1 Address Map

The TCC720 has fixed address maps for digital audio en-decoder system. The address space is separated MSB 4bits of address bus, the following table represents overall address space of TCC720 system.

Table 2.1 Address Allocation Map of TCC720

Address Space	Device Name
0x00000000 ~ 0x0FFFFFFF	internal or external ROM of chip select 3 (Remap == 0) Other type memory according to base value (Remap = 1) Internal SRAM when any other memory is not assigned.
0x10000000 ~ 0x1FFFFFFF	Not Used
0x20000000 ~ 0x2FFFFFFF	Initial area for SDRAM
0x30000000 ~ 0x3FFFFFFF	Area of internal SRAM
0x40000000 ~ 0x4FFFFFFF	Initial area for chip select 0 Initial configuration is for SRAM
0x50000000 ~ 0x5FFFFFFF	Initial area for chip select 1 Initial configuration is for IDE type device
0x60000000 ~ 0x6FFFFFFF	Initial area for chip select 2 Initial configuration is for NAND flash
0x70000000 ~ 0x7FFFFFFF	Initial area for chip select 3 Initial configuration is for ROM
0x80000000 ~ 0x8FFFFFFF	Various internal peripheral devices
0x90000000 ~ 0x9FFFFFFF	Not Used
0xA0000000 ~ 0xAFFFFFFF	
0xB0000000 ~ 0xBFFFFFFF	
0xC0000000 ~ 0xCFFFFFFF	
0xD0000000 ~ 0xDFFFFFFF	
0xE0000000 ~ 0xEFFFFFFF	Area for internal boot ROM
0xF0000000 ~ 0xFFFFFFFF	Area for memory controller configuration register space

The address space (0x00000000 ~ 0x0FFFFFFF) is initially allocated to internal or external PROM for booting procedure, and a special flag is exist in memory controller unit for remapping lower half space to other type memories. Refer to the description of memory controller for

detailed operation.

TCC720 has only one chip select for SDRAM, so its address space is dependent on SDRAM size attached to TCC720.

TCC720 has various peripherals for controlling a digital audio en-decoder system. These peripherals can be configured appropriately by it's own registers that can be accessed through specially allocated address. These address maps are represented in the following table. In case of memory controller, its space is separated for preventing illegal accessing.

Refer to corresponding sections for detail information of each peripheral.

Table 2.2 Address Allocation for Internal Peripherals (Base Address = 0x80000000)

Offset Address Space	Peripheral
0x000 ~ 0x0FF	DAI & CDIF
0x100 ~ 0x1FF	Interrupt Controller
0x200 ~ 0x2FF	Timer Counter
0x300 ~ 0x3FF	GPIO
0x400 ~ 0x4FF	Clock Generator & Power Management
0x500 ~ 0x5FF	USB Function
0x600 ~ 0x6FF	UART/IrDA
0x700 ~ 0x7FF	GSIO (General Purpose Serial Input/Output)
0x800 ~ 0x8FF	-
0x900 ~ 0x9FF	-
0xA00 ~ 0xAFF	Analog Control & Etc.
0xB00 ~ 0xBFF	-
0xC00 ~ 0xCFF	-
0xD00 ~ 0xDFF	-
0xE00 ~ 0xEFF	DMA Controller
0xF00 ~ 0xFFF	-

*) Address decoding logic only monitors base address (i.e. 0x8xxxxxxx), and bit11~bit8 of accessing address bus. So care must be taken not to modify these registers unintentionally.

2.2 Register Map

DAI & CDIF Register Map (Base Address = 0x80000000)

Name	Address	Type	Reset	Description
DADI_L0	0x00	R	-	Digital Audio Left Input Register 0
DADI_R0	0x04	R	-	Digital Audio Right Input Register 0
DADI_L1	0x08	R	-	Digital Audio Left Input Register 1
DADI_R1	0x0C	R	-	Digital Audio Right Input Register 1
DADO_L0	0x10	R/W	-	Digital Audio Left Output Register 0
DADO_R0	0x14	R/W	-	Digital Audio Right Output Register 0
DADO_L1	0x18	R/W	-	Digital Audio Left Output Register 1
DADO_R1	0x1C	R/W	-	Digital Audio Right Output Register 1
DAMR	0x20	R/W	0x0000	Digital Audio Mode Register
DAVC	0x24	R/W	0x0000	Digital Audio Volume Control Register
CDDI_0	0x80	R	-	CD Digital Audio Input Register 0
CDDI_1	0x84	R	-	CD Digital Audio Input Register 1
CICR	0x88	R/W	0x0000	CD Interface Control Register

Interrupt Controller Register Map (Base Address = 0x80000100)

Name	Address	Type	Reset	Description
IEN	0x00	R/W	0x0000	Interrupt Enable Register
CREQ	0x04	W	-	Clear Interrupt Request Register
IREQ	0x08	R	0x0000	Interrupt Request Flag Register
IRQSEL	0x0C	R/W	0x0000	IRQ/FIQ Select Register
ICFG	0x10	R/W	0x0000	External Interrupt Configuration Register
MREQ	0x14	R	0x0000	Masked Interrupt Request Flag Register
IRQREQ	0x18	R	0x0000	IRQ Interrupt Request Flag Register
FIQREQ	0x1C	R	0x0000	FIQ Interrupt Request Flag Register

Timer/Counter Register Map (Base Address = 0x80000200)

Name	Address	Type	Reset	Description
TCFG0	0x0000	R/W	0x00	Timer/Counter 0 Configuration Register
TCNT0	0x0004	R/W	0x0000	Timer/Counter 0 Counter Register
TREF0	0x0008	R/W	0xFFFF	Timer/Counter 0 Reference Register
TMREF0	0x000C	R/W	0x0000	Timer/Counter 0 Middle Reference Register
TCFG1	0x0010	R/W	0x00	Timer/Counter 1 Configuration Register
TCNT1	0x0014	R/W	0x0000	Timer/Counter 1 Counter Register
TREF1	0x0018	R/W	0xFFFF	Timer/Counter 1 Reference Register
TMREF1	0x001C	R/W	0x0000	Timer/Counter 1 Middle Reference Register
TCFG2	0x0020	R/W	0x00	Timer/Counter 2 Configuration Register
TCNT2	0x0024	R/W	0x0000	Timer/Counter 2 Counter Register
TREF2	0x0028	R/W	0xFFFF	Timer/Counter 2 Reference Register
TMREF2	0x002C	R/W	0x0000	Timer/Counter 2 Middle Reference Register
TCFG3	0x0030	R/W	0x00	Timer/Counter 3 Configuration Register
TCNT3	0x0034	R/W	0x0000	Timer/Counter 3 Counter Register
TREF3	0x0038	R/W	0xFFFF	Timer/Counter 3 Reference Register
TMREF3	0x003C	R/W	0x0000	Timer/Counter 3 Middle Reference Register
TCFG4	0x0040	R/W	0x00	Timer/Counter 4 Configuration Register
TCNT4	0x0044	R/W	0x00000	Timer/Counter 4 Counter Register
TREF4	0x0048	R/W	0xFFFFF	Timer/Counter 4 Reference Register
TCFG5	0x0050	R/W	0x00	Timer/Counter 5 Configuration Register
TCNT5	0x0054	R/W	0x00000	Timer/Counter 5 Counter Register
TREF5	0x0058	R/W	0xFFFFF	Timer/Counter 5 Reference Register
TIREQ	0x0060	R/W	0x0000	Timer/Counter n Interrupt Request Register
TWDCFG	0x0070	R/W	0x0000	Watchdog Timer Configuration Register
TWDCLR	0x0074	W	-	Watchdog Timer Clear Register

GPIO Register Map (Base Address = 0x80000300)

Name	Addr	Type	Reset	Description
GDATA_A	0x00	R/W	0xFFFFFFFF	GPIO_A Data Register
GIOCON_A	0x04	R/W	0xFFFF0000	GPIO_A Direction Control Register
GSEL_A	0x08	R/W	0x00000000	GPIO_A Function Select Register
GTSEL_A	0x0C	R/W	0x00000000	GPIO_A Function Select Register 2
GDATA_B	0x10	R/W	0x3FFFFFFF	GPIO_B Data Register
GIOCON_B	0x14	R/W	0x001FFCFF	GPIO_B Direction Control Register
GSEL_B	0x18	R/W	0x3C0000BF	GPIO_B Function Select Register
GTSEL_B	0x1C	R/W	0x00000000	GPIO_B Function Select Register 2

Clock Generator Register Map (Base Address = 0x80000400)

Name	Address	Type	Reset	Description
CKCTRL	0x00	R/W	0x0003FFE	Clock Control Register
PLLMODE	0x04	R/W	0x03806	PLL Control Register
SCLKmode	0x08	R/W	0x082000	System Clock Control Register
DCLKmode	0x0C	R/W	0x0800	DCLK (DAI/CODEC) Control Register
EXTCLKmode	0x14	R/W	0x0000	EXTCLK (CD/Other) Control Register
UTCLKmode	0x18	R/W	0x01BE	UTCLK (UART) Control Register
UBCLKmode	0x1C	R/W	0x000	UBCLK (USB) Control Register
TCLKmode	0x24	R/W	0x000	TCLK (Timer) Control Register
GCLKmode	0x28	R/W	0x000	GCLK (GSIO) Control Register
SW_nRST	0x3C	R/W	0x000	Software Reset Control Register

USB Register Map (Base Address = 0x80000500)

Name	Address	Type	Reset	Description
UBFADR	0x00			Function Address Register
UBPWR	0x04			Power Management Register
UBIIR	0x08			In-Interrupt Register
UBOIR	0x10			Out-Interrupt Register
UBIR	0x18			Interrupt Register
UBIIEN	0x1C			In-Interrupt Enable Register
UBOIEN	0x24			Out-Interrupt Enable Register
UBIEN	0x2C			Interrupt Enable Register
UBFRM1	0x30			Frame Number 1 Register
UBFRM2	0x34			Frame Number 2 Register
UBIDX	0x38			Index Register
INMXPn	0x40			IN Max Packet Register
INCSR1n	0x44			IN CSR1 Register
INCSR2n	0x48			IN CSR2 Register
OMXPn	0x4C			OUT Max Packet Register
OCSR1n	0x50			OUT CSR1 Register
OCSR2n	0x54			OUT CSR2 Register
OFIFO1n	0x58			OUT FIFO Write Count 1 Register
OFIFO2n	0x5C			OUT FIFO Write Count 2 Register
EP0FIFO	0x80			EP0 FIFO Register
EP1FIFO	0x84			EP1 FIFO Register
EP2FIFO	0x88			EP2 FIFO Register

UART/IrDA Register Map (Base Address = 0x80000600)

Name	Address	Type	Reset	Description
RB	0x00	R	-	Receiver Buffer Register
THR	0x00	W	-	Transmitter Holding Register
DL	0x04	W	0x0000	Divisor Latch Register
IR	0x08	R/W	0x000	Interrupt Register
CR	0x0C	R/W	0x000	UART Control Register
LSR	0x10	R	0x0101	Status Register
IrDACFG1	0x14	R/W	0x0003	IrDA Configuration Register 1
IrDACFG2	0x18	R/W	0x4da1	IrDA Configuration Register 2

GSIO Register Map (Base Address = 0x80000700)

Name	Address	Type	Reset	Description
GSDO0	0x00	R/W		GSIO0 Output Data Register
GSDI0	0x04	R/W		GSIO0 Input Data Register
GSCR0	0x08	R/W		GSIO0 Control Register
GSICR	0x0C	R/W		GSIO Interrupt Control Register
GSDO1	0x10	R/W		GSIO1 Output Data Register
GSDI1	0x14	R/W		GSIO1 Input Data Register
GSCR1	0x18	R/W		GSIO1 Control Register
GSDO2	0x20	R/W		GSIO2 Output Data Register
GSDI2	0x24	R/W		GSIO2 Input Data Register
GSCR2	0x28	R/W		GSIO2 Control Register
GSDO3	0x30	R/W		GSIO3 Output Data Register
GSDI3	0x34	R/W		GSIO3 Input Data Register
GSCR3	0x38	R/W		GSIO3 Control Register

Analog Interface & ETC Register Map (Base Address = 0x80000A00)

Name	Address	Type	Reset	Description
ADCTR	0x00	R/W	0	ADC Control Register
ADDATA	0x04	R	-	ADC Data Register
CDCTR	0x08	R/W	0	Codec Control Register
CDCGAIN	0x0C	R/W	0	Codec Gain Register
LZC	0x10	R/W	-	Leading Zero Counter Register
USBCTR	0x14	R/W	0	USB Port Control Register
TSTSEL	0x18	R/W	0	Test Mode Register (must be remained zero)

DMA Controller Register Map (Base Address = 0x80000E00)

Name	Address	Type	Reset	Description
ST_SADR	0x00	R/W	-	Start Address of Source Block
SPARAM	0x04/0x08	R/W	-	Parameter of Source Block
C_SADR	0x0C	R	-	Current Address of Source Block
ST_DADR	0x10	R/W	-	Start Address of Destination Block
DPARAM	0x14/0x18	R/W	-	Parameter of Destination Block
C_DADR	0x1C	R	-	Current Address of Destination Block
HCOUNT	0x20	R/W	0x00000000	Initial and Current Hop count
CHCTRL	0x24	R/W	0x00000000	Channel Configuration

Memory Controller Register Map (Base Address = 0xF0000000)

Name	Address	Type	Reset	Description
SDCFG	0x00	R/W	0x4268A020	SDRAM Configuration Register
SDFSM	0x04	R	-	SDRAM FSM Status Register
MCFG	0x08	R/W	0xZZZZ_02	Miscellaneous Configuration Register
TST	0x0C	W	0x0000	Test mode register (must be remained zero)
CSCFG0	0x10	R/W	0x0B405601	External Chip Select 0 Configuration Register (Initially set to SRAM)
CSCFG1	0x14	R/W	0x0150569A	External Chip Select 1 Configuration Register (Initially set to IDE)
CSCFG2	0x18	R/W	0x0060569A	External Chip Select 2 Configuration Register (Initially set to NAND)
CSCFG3	0x1C	R/W	0x0A70569A	External Chip Select 3 Configuration Register (Initially set to NOR)

NAND flash Register Map (Base Address = N * 0x10000000)

Name	Address	Type	Reset	Description
CMD	0x00	R/W	-	Command Cycle Register
LADDR	0x04	W	-	Linear Address Cycle Register
BADDR	0x08	W	-	Block Address Cycle Register
IADDR	0x0C	W	-	Single Address Cycle Register
DATA	0x10	R/W	-	Data Access Cycle Register

***) N represents BASE field of CSCFGn registers.**

CHAPTER 3

DAI & CDIF

3 DAI (Digital Audio Interface) & CDIF (CD-DSP Interface)

3.1 DAI

The TCC720 provides digital audio interface that complies with IIS (Inter-IC Sound). The DAI has five input/output pins for IIS interface; MCLK, BCLK, LRCK, DAI, DAO. All DAI input/output pins are multiplexed with GPIO pins; GPIO_B<21:25>. The MCLK is the system clock pin that is used for CODEC system clock. The DAI provides 256fs, 384fs and 512fs as a system clock. 256fs means that 256 times of sampling frequency fs. The BCLK is the serial bit clock for IIS data exchange. The DAI can generate 64fs, 48fs and 32fs by dividing a system clock. The polarity of BCLK can be programmed. The LRCK is the frame clock for the audio channel Left and Right. The frequency of LRCK is the “fs” – sampling frequency. Generally, for audio application – such as MP3 Player , CD player, the fs can be set to 8kHz, 16kHz, 11.05kHz, 24kHz, 32kHz, 44.1kHz and 48kHz. For supporting the wide range of sampling frequency in audio application, the DCO function is very useful to generate a system clock. Refer Chap. 7 for detail information. All three clocks are selectable as master or slave. The DAI, DAO are the serial data input output pins respectively.

The DAI has two 8-word input/output buffers. The buffers can be read/written through the DADI/DADO. The hidden buffer pointers automatically increment when user read/write from/to the DADI/DADO. The maximum data word size is 24 bit. Data is justified to MSB of 32bits and zeros are padded to LSB.

There are 2 types of interrupt from IIS; transmit done interrupt, receive done interrupt. The transmit-done interrupt is generated when the 4 words are transferred successfully in the output buffer. At this interrupt, user should fill another 4 more words into the other part of the output buffer in the interrupt service routine (ISR). In this ISR routine, 4 consecutive stores of word data to the DADO is needed – sequence is that the left channel is the first and right channel. The receive-done interrupt is generated when the 4 words are received successfully in the input buffer. At this interrupt, user should read 4 received words from the input buffer using 4 consecutive load instructions from the DADI – sequence is that the left channel is the first.

DAI Register Map (Base Address = 0x80000000)

Name	Address	Type	Reset	Description
DADI_L0	0x00	R	-	Digital Audio Left Input Register 0
DADI_R0	0x04	R	-	Digital Audio Right Input Register 0
DADI_L1	0x08	R	-	Digital Audio Left Input Register 1
DADI_R1	0x0C	R	-	Digital Audio Right Input Register 1
DADO_L0	0x10	R/W	-	Digital Audio Left Output Register 0
DADO_R0	0x14	R/W	-	Digital Audio Right Output Register 0
DADO_L1	0x18	R/W	-	Digital Audio Left Output Register 1
DADO_R1	0x1C	R/W	-	Digital Audio Right Output Register 1
DAMR	0x20	R	0x0000	Digital Audio Mode Register
DAVC	0x24	R/W	0x0000	Digital Audio Volume Control Register

Digital Audio Mode Register (DAMR) **0x80000020**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	TE	RE	MD	SM	BM	FM	CC	BD<1:0>		FD<1:0>		BP	CM	MM	LB

EN [15]	DAI Master Enable
0	DAI disabled
1	DAI enabled

TE [14]	DAI Transmitter Enable
0	DAI transmitter disabled
1	DAI transmitter enabled

RE [13]	DAI Receiver Enable
0	DAI receiver disabled
1	DAI receiver enabled

MD [12]	DAI Bus Mode
0	DAI has IIS bus mode
1	DAI has MSB justified mode

SM [11]	DAI System Clock Master Select
0	DAI system clock is come from external pin
1	DAI system clock is generated by the clock generator block

BM [10]	DAI Bit Clock Master Select
0	DAI bit clock is come from external pin
1	DAI bit clock is generated by dividing DAI system clock

FM [9]	DAI Frame Clock Master Select
0	DAI frame clock is come from external pin
1	DAI frame clock is generated by dividing DAI bit clock

CC [8]	CDIF Clock Select
0	CDIF Clock master mode disabled
1	CDIF Clock master mode enabled.

BD [7:6]	DAI Bit Clock Divider select
00	Div 4 (256fs->64fs)
01	Div 6 (384fs->64fs)
10	Div 8 (512fs->64fs, 384fs->48fs , 256fs->32fs)
11	Div16 (512fs->32fs)

FD [5:4]	DAI Frame Clock Divider select
00	Div 32 (32fs->fs)
01	Div 48 (48fs->fs)
10	Div 64 (64fs->fs)

BP [3]	DAI Bit Clock Polarity
0	Data is captured at positive edge of bit clock
1	Data is captured at negative edge of bit clock

CM [2]	CDIF Monitor Mode
0	CDIF monitor mode is disabled
1	CDIF monitor mode is enabled. Data bypass from CDIF

MM [1]	DAI Monitor Mode
0	DAI monitor mode is disabled
1	DAI monitor mode is enabled. TE should be enabled

LB [0]	DAI Loop-back Mode
0	DAI Loop back mode is disabled
1	DAI Loop back mode is enabled

Digital Audio Volume Control Register (DAVC) **0x80000024**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												VC<3:0>			

VC [3:0]	DAI Volume control
0000	0dB
0001	-6dB
0010	-12dB
0011	-18dB
0100	-24dB
0101	-30dB
0110	-36dB
0111	-42dB
1000	-48dB
1001	-54dB
1010	-60dB
1011	-66dB
1100	-72dB
1101	-78dB
1110	-84dB
1111	-90dB

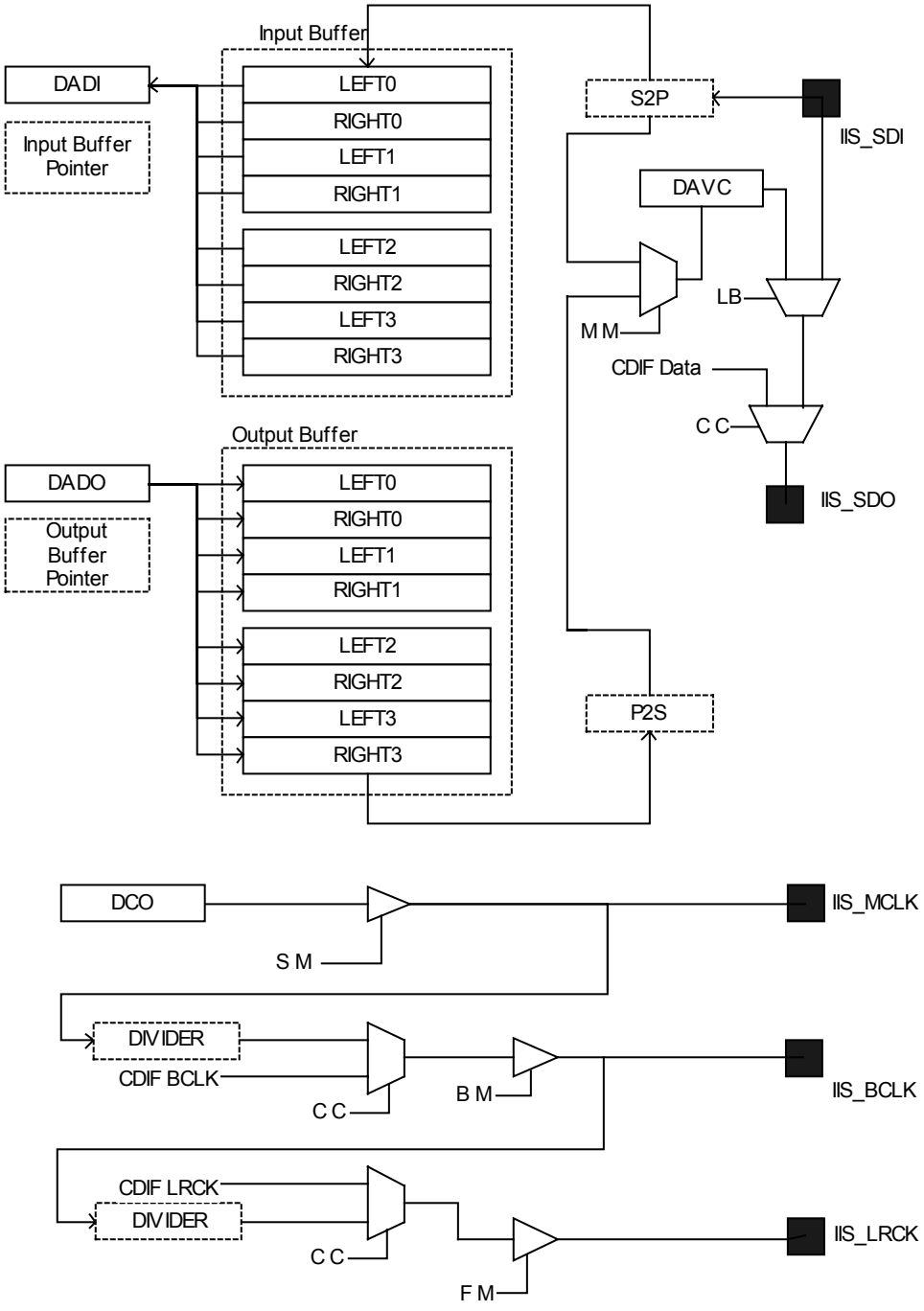
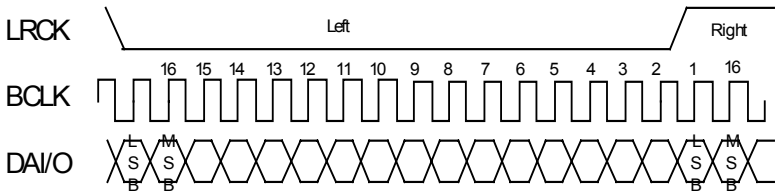
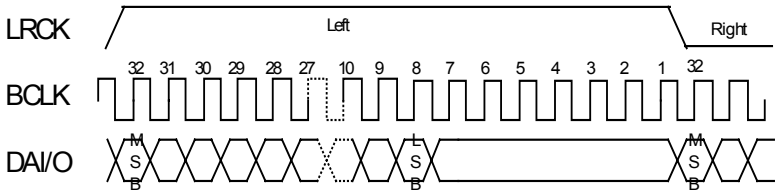


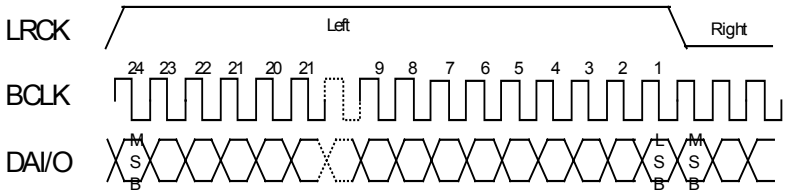
Figure 3.1 DAI Block Diagram



MD=0 (IIS mode), BP=0, BCLK = 32fs



MD=1(MSB justified mode), BP=0, BCLK=64fs



MD=1(MSB justified mode), BP=1, BCLK=48fs

Figure 3.2 DAI Bus Timing Diagram

3.2 CDIF

The TCC720 provides CD-ROM interface for feasible implementation of CD-ROM application such as CD-MP3 player. The CDIF supports the industry standard IIS format and the LSB justified format that is used as the most popular format for CD-ROM interface by Sony and Samsung. The CDIF has three pins for interface; CBCLK, CLRCK, CDAI that are multiplexed with GPIO_B14, GPIO_B15 and GPIO_B16, respectively. The CBCLK is the bit clock input pins of which frequency can be programmed by CICR for selection of 48fs and 32fs. The CLRCK is the frame clock input pin that indicates the channel of CD digital audio data. The CDAI is the input data pin.

The CDIF has three registers; CDDI_0, CDDI_1 and CICR. The CDDI_0 and the CDDI_1 are the banked read only registers for access of data input buffer. The data input buffer is composed of four 32 bit wide registers of which upper 16 bit is left channel data and lower is right channel data. The CDIF receive the serial data from CDAI pin and store the data into the buffer through the serial to parallel register. Whenever the half of buffer is filled, the receive interrupt is generated. Only the half of input buffer can be accessible through the CDDI_0 and the CDDI_1.

CDIF Register Map (Base Address = 0x80000000)

Name	Address	Type	Reset	Description
CDDI_0	0x80	R		CD Digital Audio Input Register 0
CDDI_1	0x84	R		CD Digital Audio Input Register 1
CICR	0x88	R/W	0x0000	CD Interface Control Register

CD Data Input (CDDI0) **0x80000080**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Data Input (CDDI1) **0x80000084**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Left Channel Data															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Right Channel Data															

CD Interface Control Register (CICR) **0x80000088**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EN	Reserved			BS		MD	BP

EN [7]	CDIF Enable
0	CDIF disabled
1	CDIF enabled

BS [3:2]	CDIF Bit Clock select
00	64fs
01	32fs
10	48fs

MD [1]	Interface Mode select
0	IIS format
1	LSB justified format

BP [3]	CDIF Bit Clock Polarity
0	Data is captured at positive edge of bit clock
1	Data is captured at negative edge of bit clock

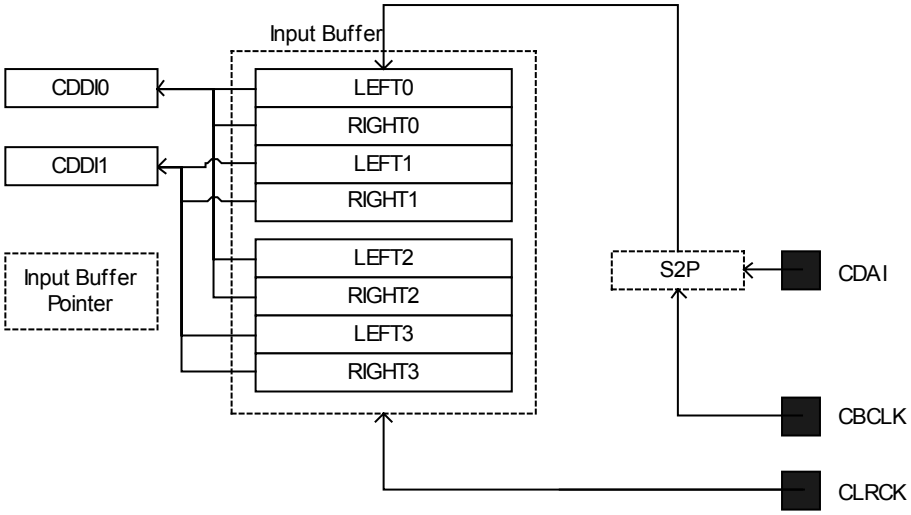
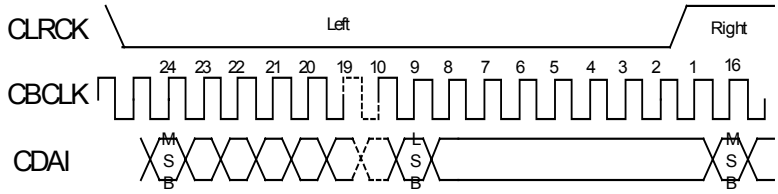
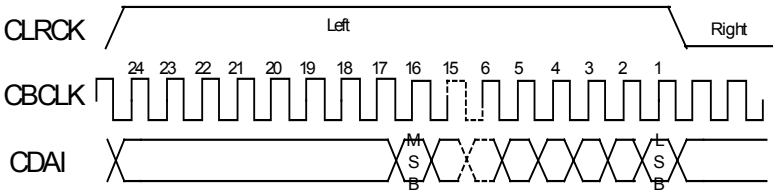


Figure 3.3. CDIF Block Diagram



MD=0 (IIS mode), BP=0, CBCLK=48fs



MD=1(LSB justified mode), BP=0, CBCLK=48fs

Figure 3.3 CDIF Bus Timing Diagram

CHAPTER 4

INTERRUPT CONTROLLER

4 INTERRUPT CONTROLLER

4.1 Overview

Interrupt controller can manage up to 16 interrupt sources. In TCC720, there are 4 external interrupt sources that can be detected various kind of method, that is a rising edge/ falling edge / level high / level low detection can be set for external interrupt sources. External interrupt sources can be reliably managed with noise filtering up to 100 ~ 400 us.

There are two types of interrupt in ARM940T, IRQ type, FIQ type.
Interrupt controller can manage these two types for each interrupt sources separately.

The following figure represents the block diagram of interrupt controller.

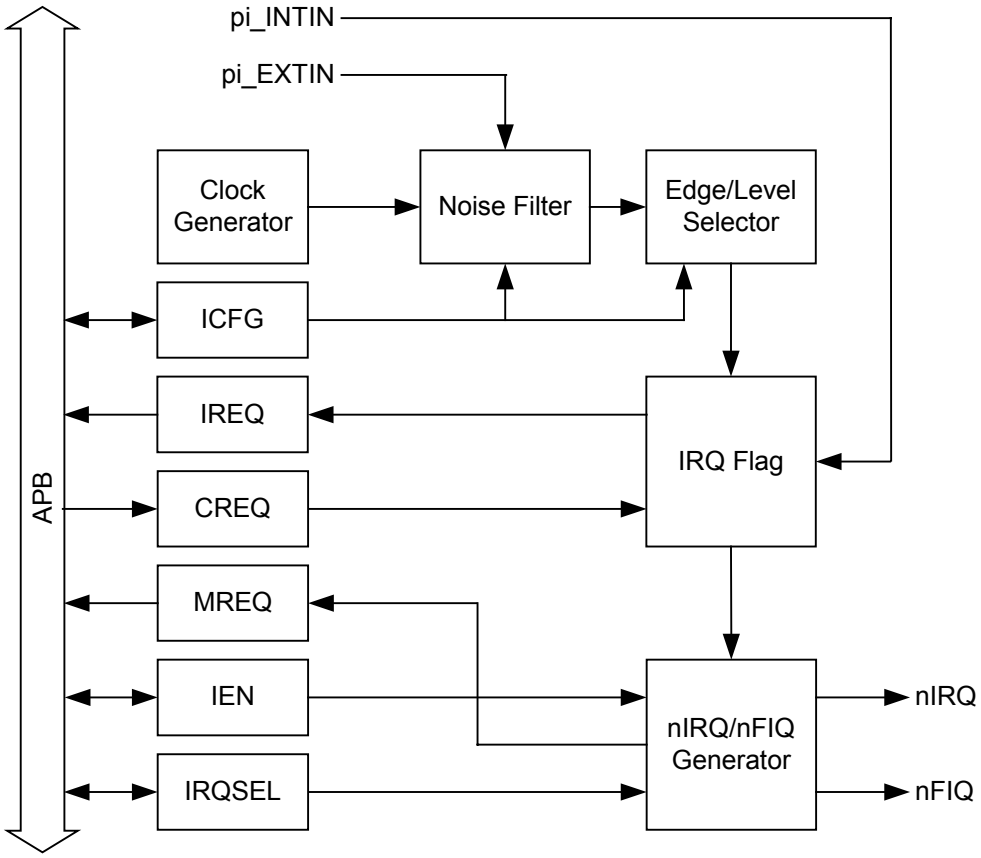


Figure 4.1 Interrupt Controller Block Diagram

4.2 Register Description

Interrupt Controller Register Map (Base Address = 0x80000100)

Name	Address	Type	Reset	Description
IEN	0x00	R/W	0x0000	Interrupt Enable Register
CREQ	0x04	W	-	Clear Interrupt Request Register
IREQ	0x08	R	0x0000	Interrupt Request Flag Register
IRQSEL	0x0C	R/W	0x0000	IRQ/FIQ Select Register
ICFG	0x10	R/W	0x0000	External Interrupt Configuration Register
MREQ	0x14	R	0x0000	Masked Interrupt Request Flag Register
TSTREQ	0x18	R/W	0x0000	Test Mode Register (must be remained zero)

Interrupt Enable Register (IEN) **0x80000100**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEN	-	DMA	LCD	CDIF	-	GS	UB	UT	TC	I2T	I2R	E3	E2	E1	E0

MEN [15]	Master Enable
0	All interrupts are disabled
1	All interrupt enabled by corresponding bit[14:0] can be passed to CPU

Bit Field	Each Interrupt Request Control 1 = Interrupt enabled, 0 = Interrupt disabled
DMA [13]	DMA interrupt control
LCD [12]	LCD interrupt control
CDIF [11]	CDIF interrupt control
[10]	Not used
GS [9]	GSIO interrupt control
UB [8]	USB interrupt control
UT [7]	UART/IrDA interrupt control
TC [6]	Timer/Counter interrupt control
I2T [5]	I2S TX interrupt control
I2R [4]	I2S RX interrupt control
E3 [3]	External interrupt request 3 control
E2 [2]	External interrupt request 2 control

E1 [1]	External interrupt request 1 control
E0 [0]	External interrupt request 0 control

Clear Interrupt Request Register (CREQ) **0x80000104**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	DMA	LCD	CDIF	-	GS	UB	UT	TC	I2T	I2R	E3	E2	E1	E0

*) When writing “1” to each field, the interrupt request flag of corresponding interrupt is cleared.

Interrupt Request Register (IREQ) **0x80000108**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	DMA	LCD	CDIF	-	GS	UB	UT	TC	I2T	I2R	E3	E2	E1	E0

*) When each field is “1”, the corresponding interrupt has been requested and not cleared.

IRQ Interrupt Select Register (IRQSEL) **0x8000010C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	DMA	LCD	CDIF	-	GS	UB	UT	TC	I2T	I2R	E3	E2	E1	E0

*) When each field is “1”, the corresponding interrupt is considered as IRQ interrupt, otherwise as FIQ interrupt.

External Interrupt Configuration Register (ICFG) **0x80000110**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FE3	DTYPE3		FT3	FE2	DTYPE2		FT2	FE1	DTYPE1		FT1	FE0	DTYPE0		FT0

FE3~FE0	Filter Enable
0	Noise filter is enabled (in case of DTYPEn != 3)
1	Noise filter is disabled (in case of DTYPEn != 3)

*) If DTYPEn == 3, noise filter is always enabled, and this field sets which level generates the interrupt. If it is set to 1, level high triggers interrupt, and if it is set to 0, level low triggers interrupt.

DTYPE3~0	Detection Type
0	Falling edge triggered external interrupt
1	Rising edge triggered external interrupt
2	Both edge triggered external interrupt
3	Level high / low triggered external interrupt FEn field determines which level triggers the interrupt. If FEn == 1, level high triggers the interrupt and FEn == 0, level low triggers the interrupt.

FT3~FT0	Filter Type
0	Clock based filter is used. The filter delay is proportional to PCLK period as the following equations. Filter Delay = T_{PCLK} * 64 If PCLK has 25MHz, then the filter delay has about 16us.
1	Delay cell based filter is used. The filter delay varies on the operating conditions, like voltage, temperature, etc. The nominal delay is about 120ns. This type of filter must be selected when the PCLK has to be stopped, as like as stop mode etc.

Masked Interrupt Request Register (MREQ) **0x80000114**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	DMA	LCD	CDIF	-	GS	UB	UT	TC	I2T	I2R	E3	E2	E1	E0

*) Same meaning as IREQ except that it represents only the enabled interrupt's request.

CHAPTER 5

TIMER / COUNTER

5 TIMER / COUNTER

5.1 Overview

The TCC720 has four 16bit and two 20bit timer counter. Each timer counter has 3 registers for various operation modes. Refer to register description table for details. When operating in counter modes, External interrupt pin is used as counting clock for that counter.
The main clock frequency of timer counter can be configured by setting TCLK frequency. (Refer to Clock generator block) With the 12bit internal basic counter, the timer counter can generate various intervals from micro-seconds to seconds unit.

The following figure represents the block diagram of timer counter.

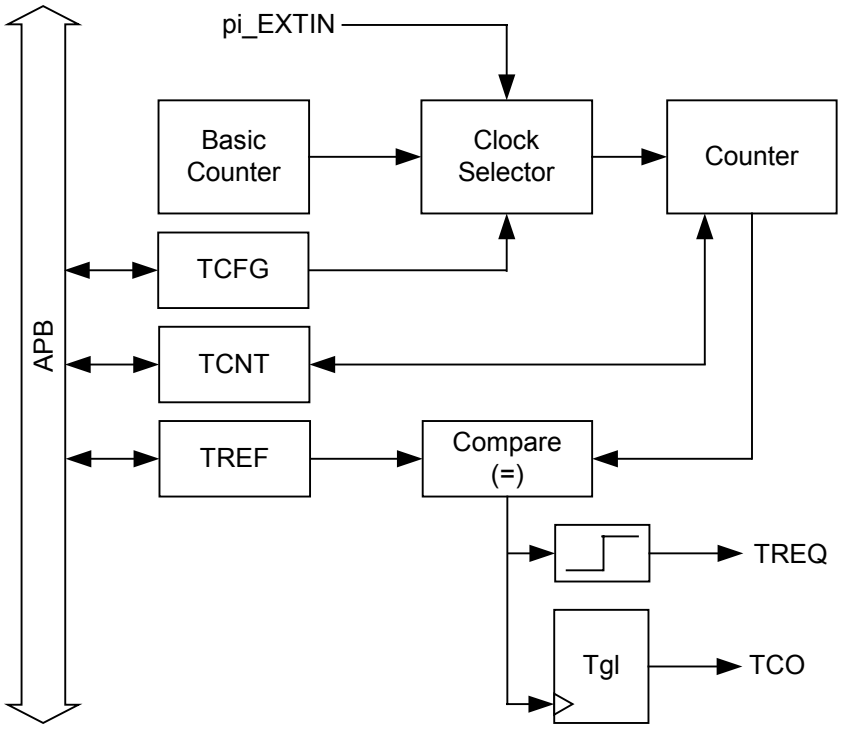


Figure 5.1 Timer Counter Block Diagram

The following table explains the three registers of each timer counter. The address of each timer counter is 16bytes aligned. The base address of timer counter is 0x80000200.

The number n represents for each timer/counter. In case of timer/counter 4, 5 (that is n = 4 or 5)

the TREF, TCNT register has 20bit resolution. It can be used for generation of a long time period.

5.2 Register Description

Timer/Counter Register Map (Base Address = 0x80000200)

Name	Address	Type	Reset	Description
TCFG0	0x0000	R/W	0x00	Timer/Counter 0 Configuration Register
TCNT0	0x0004	R/W	0x0000	Timer/Counter 0 Counter Register
TREF0	0x0008	R/W	0xFFFF	Timer/Counter 0 Reference Register
TMREF0	0x000C	R/W	0x0000	Timer/Counter 0 Middle Reference Register
TCFG1	0x0010	R/W	0x00	Timer/Counter 1 Configuration Register
TCNT1	0x0014	R/W	0x0000	Timer/Counter 1 Counter Register
TREF1	0x0018	R/W	0xFFFF	Timer/Counter 1 Reference Register
TMREF1	0x001C	R/W	0x0000	Timer/Counter 1 Middle Reference Register
TCFG2	0x0020	R/W	0x00	Timer/Counter 2 Configuration Register
TCNT2	0x0024	R/W	0x0000	Timer/Counter 2 Counter Register
TREF2	0x0028	R/W	0xFFFF	Timer/Counter 2 Reference Register
TMREF2	0x002C	R/W	0x0000	Timer/Counter 2 Middle Reference Register
TCFG3	0x0030	R/W	0x00	Timer/Counter 3 Configuration Register
TCNT3	0x0034	R/W	0x0000	Timer/Counter 3 Counter Register
TREF3	0x0038	R/W	0xFFFF	Timer/Counter 3 Reference Register
TMREF3	0x003C	R/W	0x0000	Timer/Counter 3 Middle Reference Register
TCFG4	0x0040	R/W	0x00	Timer/Counter 4 Configuration Register
TCNT4	0x0044	R/W	0x00000	Timer/Counter 4 Counter Register
TREF4	0x0048	R/W	0xFFFFF	Timer/Counter 4 Reference Register
TCFG5	0x0050	R/W	0x00	Timer/Counter 5 Configuration Register
TCNT5	0x0054	R/W	0x00000	Timer/Counter 5 Counter Register
TREF5	0x0058	R/W	0xFFFFF	Timer/Counter 5 Reference Register
TIREQ	0x0060	R/W	0x0000	Timer/Counter n Interrupt Request Register
TWDCFG	0x0070	R/W	0x0000	Watchdog Timer Configuration Register
TWDCLR	0x0074	W	-	Watchdog Timer Clear Register

Timer/Counter n Configuration Register (TCFGn) **0x800002n0**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							CC	POL	TCKSEL			IEN	PWM	CON	EN

CC [8]	Clear Count
0	TCNTn hold its value.
1	TCNTn is cleared to zero.

POL [7]	TCK Polarity
0	TCNTn is incremented at rising edge of the selected counting clock
1	TCNTn is incremented at falling edge of the selected counting clock

TCKSEL [6:4]	TCK Select
k = 0 ~ 4	TCK is internally generated from divider circuit. It is driven by PCLK, and this value determines the division factor of this circuit. Division factor is $2^{(k+1)}$.
k = 5, 6	TCK is internally generated from divider circuit. It is driven by PCLK, and this value determines the division factor of this circuit. Division factor is 2^{2k}
k = 7	TCK is the external pin shared by external interrupt signal. In TCC721, there are 4 external pins for this purpose, so this configuration is valid only for timer/counter 3 ~ 0. (not for timer/counter 5, 4)

IEN [3]	Interrupt Enable
1	Timer/Counter interrupt is enabled

PWM [2]	PWM Mode Enable
1	Timer/Counter Output is changed at every time the TCNTn is equal to TREFn and TMREFn value. It can be used to generate PWM waveform, by changing TMREFn while fixing TREFn. (where, TREFn > TMREFn)

CON [1]	Continue Counting
0	TCNTn is stop counting at the time TCNTn is equal to TREFn
1	When the TCNTn is reached to TREFn, TCNTn continues counting from 0 at the next pulse of selected clock source

EN [0]	Timer/Counter Enable
1	Timer counter is enabled. TCNTn value is cleared at the same time.

Timer/Counter n Counting Register (TCNTn) **0x800002n4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												TCNTn[19:16]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCNTn[15:0]															

*) TCNTn is increased by 1 at every pulse of selected clock source. TCNTn can be set to any value by writing the value to this register. In case of timer 4 and timer 5, it has 20bit.

Timer/Counter n Counting Reference Register (TREFn) **0x800002n8**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												TREFn[19:16]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREFn[15:0]															

*) When TCNTn is reached at TREFn, the TCNTn is cleared to 0. According to the TCFGn settings, various kinds of operations may be done. In case of timer 4 and timer 5, it has 20bit.

Timer/Counter Interrupt Request Register (TIREQ) **0x80000260**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	TWF	TF5	TF4	TF3	TF2	TF1	TF0	0	TW	TI5	TI4	TI3	TI2	TI1	TI0

TWF	Watchdog Timer Flag
1	Watchdog timer has reached to its reference value.

TFn	Timer/Counter n Flag
1	Timer/counter n has been overflowed.

TWI	Watchdog Timer Interrupt Request Flag
1	Watchdog timer has generated its interrupt.

TIn	Timer/Counter n Interrupt Request Flag
1	Timer/counter n has generated its interrupt.

*) if a timer n has reached its reference value, the TFn is set. (bit n represents for Timer n). If its interrupt request is enabled by set bit 3 of TCFGn register, the TIn is set. And if the TC bit of IEN register is set, the timer interrupt is really generated, and this TIREQ register is used to determine which timer has requested the interrupt. After checking these flags, user can clear these TFn, TIn field by writing “1” to corresponding TIn bit field.

Watchdog Timer Configuration Register (TWDCFG) **0x80000270**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								0	TCKSEL			IEN	0	RST	EN

Watchdog timer is used for the system not to be stuck by generating a reset pulse automatically when the watchdog timer counter overflows to zero.

The programmer must clear the watchdog counter before it overflows by writing any value to TWDCLR register.

TCKSEL [6:4]	TCK Select
k = 0 ~ 4	TCK is internally generated from divider circuit. It is driven by PCLK, and this value determines the division factor of this circuit. Division factor is $2^{(k+1)}$.
k = 5, 6	TCK is internally generated from divider circuit. It is driven by PCLK, and this value determines the division factor of this circuit. Division factor is 2^{2k}
k = 7	Undefined. Should not be used.

IEN [3]	Interrupt Enable
1	Watchdog Timer Interrupt is initiated. This field is valid only if RST field is set to 0.

RST [1]	Reset Enable
0	Watchdog timer does not generate reset signal although it reaches to the reference value, and it continue counting from 0.
1	Watchdog timer generates the reset signal when it reaches to the reference value, the reset signal is applied to every component in the chip.

EN [0]	Watchdog Timer Enable
1	Watchdog timer is enabled. If the watchdog timer is disabled, its counter goes to 0xffe0, so when it is first enabled, user must clear the counter by writing to TWDCLR register.

Watchdog Timer Clear Register (TWDCLR) **0x80000274**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
any value															

*) The watchdog timer counter can be cleared to 0 by writing any value to this register. If it is not cleared before it overflows, the watchdog timer generate reset signal to the entire component of chip.

CHAPTER 6

GPIO PORT

6 GPIO (General Purpose I/O) PORT

6.1 Functional Description

The TCC720 has a lot of general purpose I/Os that can be programmed by setting internal registers. All I/Os are set to input mode at reset. The block diagram of GPIO is in the following figure.

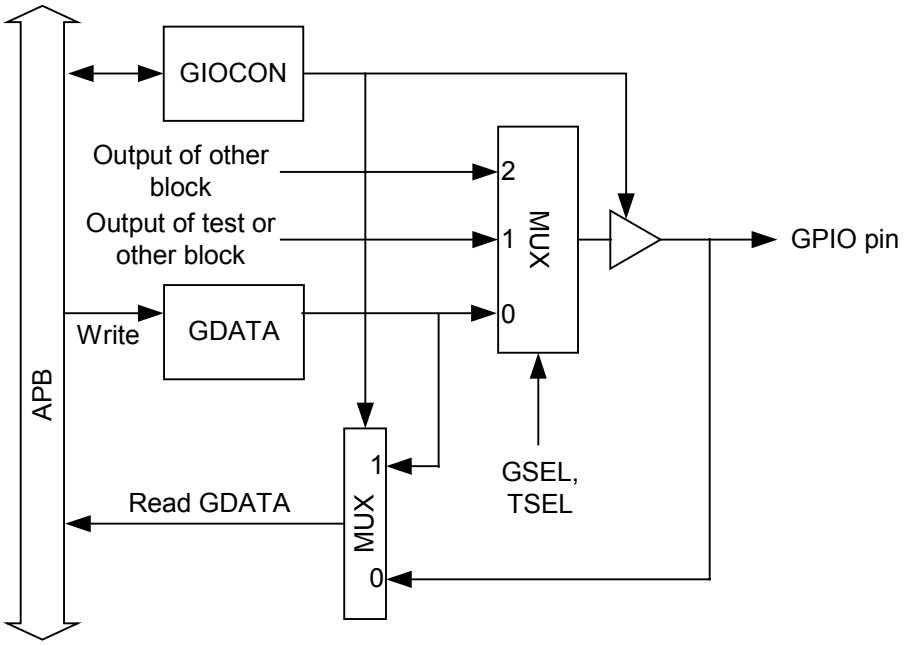


Figure 6.1 GPIO Block Diagram

The I/O mode can be set by the state of GIOCONn register.
If a bit of GIOCONn register is 1, the corresponding GPIO pin has come to output mode, and if 0, which is the default state of GIOCON register, the corresponding GPIO pin is set to input mode.

If GPIO pin is set to input mode, GPIO pin's state can be fed to CPU by reading GDATAn register and when output mode, GPIO pin's state can be controlled by the state of the corresponding bit of GDATAn register.

If GDATAn register is read when the mode is output mode, the value that CPU gets is the one

that CPU has written before.

In TCC720, there are various kinds of peripherals that generate its own control signals. These peripherals can occupy the dedicated GPIO pins. This option is controlled by the state of the GSELx register. If a bit of these GSELx is 1, the corresponding GPIO pin is entered to other function mode, so used by other peripherals not by GPIO block. The direction control method of GPIO pins in the other function mode is determined case by case. One of them follows the normal direction control method using GDDR register, the other method uses a dedicated direction control signals.

6.2 Register Description

GPIO Register Map (Base Address = 0x80000300)

Name	Addr	Type	Reset	Description
GDATA_A	0x00	R/W	0xFFFFFFFF	GPIO_A Data Register
GIOCON_A	0x04	R/W	0x00000000	GPIO_A Direction Control Register
GSEL_A	0x08	R/W	0x00000000	GPIO_A Function Select Register 1
GTSEL_A	0x0C	R/W	0x00000000	GPIO_A Function Select Register 2
GDATA_B	0x10	R/W	0x3FFFFFFF	GPIO_B Data Register
GIOCON_B	0x14	R/W	0x001FFCFF	GPIO_B Direction Control Register
GSEL_B	0x18	R/W	0x3C0000BF	GPIO_B Function Select Register 1
GTSEL_B	0x1C	R/W	0x00000000	GPIO_B Function Select Register 2

GPIO_A Data Register (GDATA_A) **0x80000300**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0xFFFF															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data for GPIO_A[15:0] pin															

GPIO_A Direction Control Register (GIOCON_A) **0x80000304**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0xFFFF															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direction control for GPIO_A[15:0] pin															

*) if a bit is set to 1, the corresponding GPIO pin is set to output mode.

GPIO_A Function Select Register (GSEL_A) **0x80000308**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					GS2[2:0]			-	GS1[2:0]			-	GS0[2:0]		

*) if a bit is set to 1, the corresponding GPIO pin is used by the other dedicated function blocks.

GSn[2:0]	GPIO_A[10:8], GPIO_A[6:4], GPIO_A[2:0] Function
0	GPIO_A[10:8], GPIO_A[6:4], GPIO_A[2:0] pin is working as Normal GPIO Function
GS2[2] = 1	GPIO_A[10] : FRM signal of GSIO2 block
GS2[1] = 1	GPIO_A[9] : SCK signal of GSIO2 block
GS2[0] = 1	GPIO_A[8] : SDO signal of GSIO2 block
GS1[2] = 1	GPIO_A[6] : FRM signal of GSIO1 block
GS1[1] = 1	GPIO_A[5] : SCK signal of GSIO1 block
GS1[0] = 1	GPIO_A[4] : SDO signal of GSIO1 block
GS0[2] = 1	GPIO_A[2] : FRM signal of GSIO0 block
GS0[1] = 1	GPIO_A[1] : SCK signal of GSIO0 block
GS0[0] = 1	GPIO_A[0] : SDO signal of GSIO0 block

*) SDI signal for GSIO2, GSIO1, GSIO0 block is always fed through GSIO_A[11], GSIO_A[7], GSIO_A[3] pin regardless of these GS[2:0] bit. But these pins must be set to input mode.

GPIO_A Test Select Register (GTSEL_A) **0x8000030C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				TC2	0		TC5	TC1	0		TC4	TC0	0		TC3

*) if a bit is set to 1, and the corresponding bit of GSEL_A is 0, GPIO pin is used by the other dedicated function blocks.

TC5 ~ TC0	GPIO_A[11,8,7,4,3,0] Function Select
0	GPIO_A[11,8,7,4,3,0] pin is working as Normal GPIO Function
1	GPIO_A[11,8,7,4,3,0] is the output of 6 timer/counter

*) this bit field is only valid only if the corresponding bit of GSEL_A is set to 0

GPIO_B Data Register (GDATA_B) **0x80000310**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		GPIO_B[29:21]									0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						GPIO_B[9:7]			0	GPIO_B[5:0]					

GPIO_B Direction Control Register (GIOCON_B) **0x80000314**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		GIO_B[29:21]									0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						GIO_B[9:7]			0	GIO_B[5:0]					

*) if a bit is set to 1, the corresponding GPIO pin is set to output mode.

The GPIO_B[29:28] and GPIO_B[27:26] pin is unable to be set to different I/O mode. That is, GPIO_B[29] have always same direction with GPIO_B[28], and it is same as GPIO_B[27] and GPIO_B[26], so to make GPIO_B[27:26] output port, you must set both GIOCON_B[27] and GIOCON_B[26] to 1.

GPIO_B Function Select Register (GSEL_B) **0x80000318**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				USB[1:0]		0	DAI[3:0]				0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							UTX	NWE	0	CS[3:0]				SCS	CKE

*) if a bit is set to 1, and the corresponding GPIO pin is set to output mode, the output of internal peripherals occupy the corresponding GPIO pin.

JSB[1:0] [27:26	GPIO_B[27:26] Function Select
0	GPIO_B[27:26] pin is working as Normal GPIO Function
3	GPIO_B[27] pin is working as USB D- Port GPIO_B[26] pin is working as USB D+ Port

DAI[3:0] [24:21]	GPIO_B[24:21] Function Select
0	GPIO_B[24:21] pin is working as Normal GPIO Function
DAI[3] = 1	GPIO_B[24] pin is working as DAO signal of DAI block
DAI[2] = 1	GPIO_B[23] pin is working as MCLK signal of DAI block
DAI[1] = 1	GPIO_B[22] pin is working as LRCK signal of DAI block
DAI[0] = 1	GPIO_B[21] pin is working as BCLK signal of DAI block

UTX [8]	GPIO_B[8] Function Select
0	GPIO_B[8] pin is working as Normal GPIO Function
1	GPIO_B[8] : UART TX signal of UART block

NWE [7]	GPIO_B[7] Function Select
0	GPIO_B[7] pin is working as Normal GPIO Function
1	GPIO_B[7] : ND_nWE (write enable for NAND flash) of memory controller

CS[3:0] [5:2]	GPIO_B[5:2] Function Select
0	GPIO_B[5:2] pin is working as Normal GPIO Function
CS[3] = 1	GPIO_B[5] : nCS3 or ND_nOE3 of memory controller
CS[2] = 1	GPIO_B[4] : nCS2 or ND_nOE2 of memory controller
CS[1] = 1	GPIO_B[3] : nCS1 or ND_nOE1 of memory controller
CS[0] = 1	GPIO_B[2] : nCS0 or ND_nOE0 of memory controller

SCS [1]	GPIO_B[1] Function Select
0	GPIO_B[1] pin is working as Normal GPIO Function
1	GPIO_B[1] : SD_nCS (chip select for SDRAM) of memory controller

CKE [0]	GPIO_B[0] Function Select
0	GPIO_B[0] pin is working as Normal GPIO Function
1	GPIO_B[0] : SD_CKE (clock enable for SDRAM) of memory controller

GPIO_B Test Select Register (GTSEL_B) **0x8000031C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0							EXT	GST[2:0]			0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						IDE									

*) if a bit is set to 1, and the corresponding bit of GSEL_B is 0, GPIO pin is used by the other dedicated function blocks.

EXT [24]	GPIO_B[24] Function Select
0	GPIO_B[24] pin is working as Normal GPIO Function or I2S Data Output
1	GPIO_B[24] pin is working as EXTCLK from Clock Controller

IDE [9]	GPIO_B[9] Function Select
0	GPIO_B[9] pin is working as Normal GPIO Function or UART Data Input
1	GPIO_B[9] pin is working as chip select 1 for IDE

GST[2:0] [23:21	GPIO_B[23:21] Function Select
0	GPIO_B[23:21] pin is working as Normal GPIO Function
GST[2] = 1	GPIO_B[23] pin is working as FRM of 1 of 4 GSIO blocks
GST[1] = 1	GPIO_B[22] pin is working as SCK of 1 of 4 GSIO blocks
GST[0] = 1	GPIO_B[21] pin is working as SDO of 1 of 4 GSIO blocks

*) this bit field is only valid only if the corresponding bit of GSEL_B is set to 0

CHAPTER 7

CLOCK GENERATOR

7 CLOCK GENERATOR

7.1 Functional Description

In TCC720, there are a lot of peripherals for which has different operating frequency. To support an appropriate stable clock to each other peripherals, TCC720 has clock generator unit and for considering power consumption there is also power management unit that can manage several operating modes, such as initialization mode, normal operation mode, idle mode, stop mode.

The simple block diagram of clock generator is as followings.

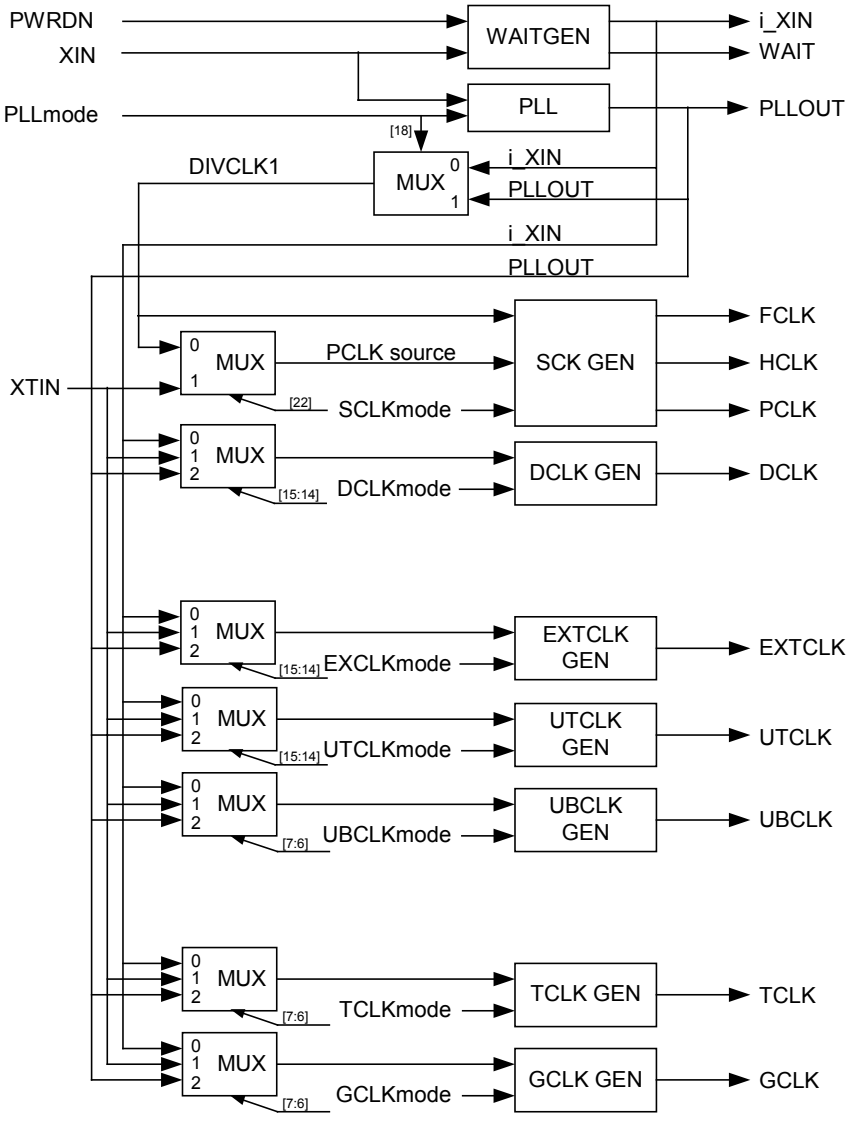


Figure 7.1 Clock Generator Block Diagram

WAITGEN cell is for waiting until oscillation is stabilized. It blocks internal clocks until about 2^18 number of XIN transition occurs after reset is released. If frequency of XIN is 16MHz, the wait time is about 16.4 ms

The DIVCLK1 are used as main clock of TCC720 and it can be either an oscillator output or PLL output clock. It is the source of system clocks (FCLK, HCLK, PCLK). The PCLK can be also driven by XTIN. The other clocks each can be driven by one of 3 clock sources XIN, PLLOUT, XTIN independently by its mode register.

DCO Control

DCLK is used as the master clock of DAI (Digital Audio Interface) block when it's mode is set to master mode. EXTCLK is used for external usage especially for CD application. UTCLK is used as the main clock of UART controller.

These clocks are generated by 14bit DCO (Digital Controlled Oscillator) that can generate a stable and variable frequency as long as its frequency is below about one tenth of the divisor clock. For reliable operation of DAI, divisor clock frequency must be higher than about 200 MHz. But maximum frequency of ARM940T is lower than 120MHz, the division factor for FCLK must be greater than 2.

The target frequency can be acquired by writing the phase value calculated by the following equation to the each PHASE register.

$$D\_PHASE = 2^{14} * f_{DCLK} / f_{DIV}$$

$$CV\_PHASE = 2^{14} * f_{CVCLK} / f_{DIV}$$

$$EXT\_PHASE = 2^{14} * f_{EXTCLK} / f_{DIV}$$

$$UT\_PHASE = 2^{14} * f_{UTCLK} / f_{DIV}$$

For example, when you use 44.1KHz sampling rate and want to set DCLK as 256fs, the target frequency of DCLK is 256 * 44.1k = 11.2896 MHz, and if you set PLL to 266MHz, the D_PHASE value must be set to 696 (~= 2^{14} * 11.2896 / 266).

7.2 Register Description

Clock Generator Register Map (Base Address = 0x80000400)

Name	Address	Type	Reset	Description
CKCTRL	0x00	R/W	0x0003FFE	Clock Control Register
PLLMODE	0x04	R/W	0x03806	PLL Control Register
SCLKmode	0x08	R/W	0x082000	System Clock Control Register
DCLKmode	0x0C	R/W	0x0800	DCLK (DAI/CODEC) Control Register
EXTCLKmode	0x14	R/W	0x0000	EXTCLK (CD/Other) Control Register
UTCLKmode	0x18	R/W	0x01BE	UTCLK (UART) Control Register
UBCLKmode	0x1C	R/W	0x00	UBCLK (USB) Control Register
TCLKmode	0x24	R/W	0x00	TCLK (Timer) Control Register
GCLKmode	0x28	R/W	0x00	GCLK (GSIO) Control Register
SW_nRST	0x3C	R/W	0x3FFF	Software Reset for each peripherals

Clock Control Register (CKCTRL) **0x80000400**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PDN	IDLE	0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	XTIN	PLL	-	GCK	TCK	-	USB	UART	EXT	-	CDC	DAI	PCK

This controls various sources of clocks fed to each peripherals. If each control bit is set to 1, the corresponding clock is disabled and the peripherals use that clock are also disabled. To enable the clock, clear the control bit to 0.

Power down and Idle mode bit are write-only register, and it is always 0 when read CKCTRL register.

PDN [25]	Power Down Mode
1	TCC720 goes to power down mode. All blocks disabled.

IDLE [24]	Idle Mode
1	TCC720 goes to idle mode. Only ARM is disabled.

XTIN [12]	Sub Clock Control
1	Disable XTIN Clock

PLL [11]	PLL Control
1	Disable PLL block

GCK [9]	GSIO Control
1	Disable GSIO block

TCK [8]	Timer Control
1	Disable Timer block

USB [6]	USB Control
1	Disable USB block

UART [5]	UART Control
1	Disable UART block

EXT [4]	EXT Clock Control
1	Disable External Clock Output (EXCLK pin)

CDC [2]	CODEC Control
1	Disable internal CODEC block. If DAI is disabled, internal CODEC is also disabled.

DAI [1]	DAI Control
1	Disable DAI block If DAI is disabled, internal CODEC is also disabled.

PCK [0]	PCLK Control
1	Disable PCLK clock (Interrupt Control block, EFM, CIRC block, GSIO block, and ADC block are disabled)

PLL Control Register (PLLmode) **0x80000404**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0												XTE	DIV1	S	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M								0	0	P					

XTE [19]	XTIN mode select
0	XTIN is disabled when power down is requested.
1	XTIN is only controlled by XTIN bit of CKCTRL register

DIV1 [18]	Divisor Clock1 Select
0	Use Oscillator as DIVCLK1
1	Use PLL output as DIVCLK1

S/M/P	PLL Frequency Setting
S/M/P	$f_{PLL} = (M + 8) * f_{Xin} / ((P + 2) * 2^S)$

The TCC720 has one PLL for generating of internal main clock. This internal PLL can generate the required frequency by setting internal register. The desired frequency can be acquired by the following equation.

$$f_{PLL} = (M + 8) * f_{Xin} / ((P + 2) * 2^S)$$

Where, M, P, S can be set by PLLmode register. M has 8bit resolution, P has 6bit resolution, and S has 2bit resolution.

PLL has standby mode for minimizing power consumption that can be controlled by PLL bit of CKCTRL register.

System Clock Control Register (SCLKmode) **0x80000408**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								PS	XTI	P_PHASE					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS	0	H_PHASE						0	0	F_PHASE					

It generates FCLK, HCLK, PCLK for system operation. FCLK is dedicated for ARM940T processor, HCLK is used as internal AHB bus clock, and PCLK is for APB bus clock. Each clock is generated by 6bit DCO (Digital Controlled Oscillator) that can generate a stable and variable frequency as long as its frequency is below about 0.1 times of that of divisor clock. For reliable operation, keep the n power of 2 relationships with divisor clock.

The target frequency can be acquired by writing the phase value calculated by the following equation to the PHASE register.

$$PHASE = 2^6 * f_{SCLK} / f_{DIV}$$

PS,XTI [23:22]	PCLK Clock Select
00	use DIVCLK1 as a divisor clock of PCLK generator
01	use XTIN pin as a divisor clock of PCLK generator
1x	use FCLK as a divisor clock of PCLK generator

P_PHASE [21:16	PCLK Frequency Select
n (!= 0)	$f_{PCLK} = f_{DIV} * n / 2^6$
0	$f_{PCLK} = f_{DIV}$ or f_{XTIN} (depends on PS, XTI bit)

*) The DIVCLK1 is selected by DIV1 bit of PLLmode register.

HS [15]	HCLK Clock Select
0	use DIVCLK1 as a divisor clock of HCLK generator
1	use FCLK as a divisor clock of HCLK generator

H_PHASE [13:8]	HCLK Frequency Select
n (!= 0)	$f_{HCLK} = f_{DIV} * n / 2^6$
0	$f_{PCLK} = f_{DIV}$ or f_{FCLK} (depends on HS bit)

F_PHASE [5:0]	FCLK Frequency Select
n (!= 0)	$f_{FCLK} = f_{DIV} * n / 2^6$
0	$f_{PCLK} = f_{DIV}$

DCLK (DAI/CODEC) Control Register (DCLKmode) **0x8000040C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVD		D_PHASE[13:0]													

DIVD [15:14]	DCLK Divisor Clock Select
0	use XIN as a divisor clock of DCLK generator
1	use PLL output as a divisor clock of DCLK generator
2, 3	use XTIN pin as a divisor clock of DCLK generator

D_PHASE [13:0]	DCLK Clock Frequency Select
d (!= 0)	$f_{DCLK} = f_{DIV} * d / 2^{14}$
0	$f_{DCLK} = f_{DIV}$

*) The divisor clock is selected by DIVD field of PLLmode register. DCLK is also controlled by DAI bit of CKCTRL register that can enable or disable DCLK. If this bit is set to high, DCLK is disabled and if it is low, DCLK is enabled.

DCLK is for DAI and internal CODEC requires 512*fs frequency. To make DCLK of this frequency, first set the frequency of PLL (f_{DIV}) more higher than 512*fs and set D_PHASE according to the above formulae. It is recommended to set the frequency of PLL by the n power of 2, than the duty ratio of DCLK is only dependant of that of PLL clock.

EXTCLK Control Register (EXTCLKmode) **0x80000414**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVXT		EX_PHASE[13:0]													

DIVXT [15:14]	EXTCLK Divisor Clock Select
0	use XIN pin as a divisor clock of EXTCLK generator
1	use PLL output as a divisor clock of EXTCLK generator
2, 3	use XTIN pin as a divisor clock of EXTCLK generator

EX_PHASE [13:0	EXTCLK Clock Frequency Select
e (!= 0)	$f_{EXTCLK} = f_{DIV} * e / 2^{14}$
0	$f_{EXTCLK} = f_{DIV}$

*) The divisor clock is selected by DIVXT bit of EXTCLKmode. EXTCLK is also controlled by EXT bit of CKCTRL register that can enable or disable EXTCLK. If this bit is set to high, EXTCLK is disabled and if it is low, EXTCLK is enabled.

External clock is user-programmable clock that can be used various purposes, it is not used by internal peripherals, and by setting GPIO registers, GPIO_B24 pin can output this clock to user application board. Care must be taken not to use too high frequency that the GPIO_B24 pin cannot cope with this signals, or the GPIO_B24 pin show no clock signal out.

UTCLK (UART) Control Register (UTCLKmode) **0x80000418**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVUT		UT_PHASE[13:0]													

DIVUT [15:14]	UTCLK Divisor Clock Select
0	use XIN pin as a divisor clock of UTCLK generator
1	use PLL output as a divisor clock of UTCLK generator
2, 3	use XTIN pin as a divisor clock of UTCLK generator

JT_PHASE [13:(	UTCLK Clock Frequency Select
u (!= 0)	$f_{UTCLK} = f_{DIV} * u / 2^{14}$
0	$f_{UTCLK} = f_{DIV}$

*) The divisor clock is selected by DIVUT bit of UTCLKmode. UTCLK is also controlled by UART bit of CKCTRL register that can enable or disable UTCLK. If this bit is set to high, UTCLK is disabled and if it is low, UTCLK is enabled

This clock is used by UART. For reliable communication with host side, this clock has the frequency of 3.6864MHz or so. The UART clock is then divided by DL register in UART block, it is not so important to maintain the duty ratio of 50%.

UBCLK (USB) Control Register (UBCLKmode) **0x8000041C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						DIVUB		0	0	UB_PHASE[5:0]					

UBCLK is used as the main clock of USB block. It is generated by a DCO that has 6bit resolution, and its frequency is set by writing the phase value calculated by the following equation to the UB_PHASE register.

$$UB\_PHASE = 2^6 * f_{UBCLK} / f_{DIV}$$

UBCLK is also controlled by USB bit of CKCTRL register that can enable or disable UBCLK. If this bit is set to low, UBCLK is enabled and if it is high, UBCLK is disabled.

DIVUB [9:8]	UBCLK Divisor Clock Select
0	use XIN pin as a divisor clock of UBCLK generator
1	use PLL output as a divisor clock of UBCLK generator
2, 3	use XTIN pin as a divisor clock of UBCLK generator

JB_PHASE [5:0	UBCLK Clock Frequency Select
ub (!= 0)	$f_{UBCLK} = f_{DIV} * ub / 2^6$
0	$f_{UBCLK} = f_{DIV}$

*) The divisor clock is selected by DIVUB bit of UBCLKmode. UBCLK is also controlled by USB bit of CKCTRL register that can enable or disable UBCLK. If this bit is set to high, UBCLK is disabled and if it is low, UBCLK is enabled

TCLK (Timer) Control Register (TCLKmode) **0x80000424**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						DIVT		0	0	TC_PHASE[5:0]					

DIVT [9:8]	TCLK Divisor Clock Select
0	use XIN pin as a divisor clock of TCLK generator
1	use PLL output as a divisor clock of TCLK generator
2, 3	use XTIN pin as a divisor clock of TCLK generator

ΓC_PHASE [5:0	TCLK Clock Frequency Select
tc (!= 0)	$f_{TC} = f_{DIV} * tc / 2^6$
0	$f_{TC} = f_{DIV}$

*) The divisor clock is selected by DIVT field of TCLKmode.

GCLK (GSIO) Control Register (GCLKmode) **0x80000428**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						DIVG		0	0	GC_PHASE[5:0]					

DIVG [9:8]	GCLK Divisor Clock Select
0	use XIN pin as a divisor clock of GCLK generator
1	use PLL output as a divisor clock of GCLK generator
2, 3	use XTIN pin as a divisor clock of GCLK generator

GC_PHASE [5:0	GCLK Clock Frequency Select
gc (!= 0)	$f_{GC} = f_{DIV} * gc / 2^6$
0	$f_{GC} = f_{DIV}$

*) The divisor clock is selected by DIVG field of GCLKmode.

Software Reset Register (SW_nRST) **0x8000043C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	DMA	1	ETC	1	1	1	GS	UT	UB	GP	TC	IC	DAI

DMA [12]	DMA Block Reset Control
1	Reset for DMA is released
0	Reset for DMA is generated

ETC [10]	Miscellaneous Block Reset Control
1	Reset for Miscellaneous Block is released
0	Reset for Miscellaneous Block is generated

*) Miscellaneous block contains ADC and CODEC control register and leading zero counter register, etc.

GS [6]	GSIO Block Reset Control
1	Reset for GSIO is released
0	Reset for GSIO is generated

UT [5]	UART/IrDA Block Reset Control
1	Reset for UART/IrDA is released
0	Reset for UART/IrDA is generated

UB [4]	USB Block Reset Control
1	Reset for USB is released
0	Reset for USB is generated

GP [3]	GPIO Block Reset Control
1	Reset for GPIO is released
0	Reset for GPIO is generated

TC [2]	Timer/Counter Block Reset Control
1	Reset for Timer/Counter is released
0	Reset for Timer/Counter is generated

IC [1]	Interrupt Controller Block Reset Control
1	Reset for Interrupt Controller is released
0	Reset for Interrupt Controller is generated

DAI [0]	DAI/CDIF Block Reset Control
1	Reset for DAI/CDIF is released
0	Reset for DAI/CDIF is generated

CHAPTER 8

USB CONTROLLER

8 USB (Universal Serial Bus) CONTROLLER

8.1 Overview

The TCC720 supports a fully compliant to USB 1.1 specification, full-speed (12 Mbps) functions and suspend/resume signaling. The USB controller is compatible with both OpenHCI and Intel UHCI standards. The USB function controller has an endpoint EP0 for control and two in/output endpoints EP1/EP2 for bulk data transaction. The endpoint EP0 has a single 16 byte FIFO; Max packet size is 16 bytes. And the endpoint EP1 and EP2 have a dual 128 byte FIFO, respectively; Max packet size of EP1 and EP2 is 64 bytes.

There are 4 types of internal registers; IN_CSR (IN Control Status Register), OUT_CSR (OUT Control Status Register), IN_MAXP (IN Maximum Packet size Register), and OUT WRITE COUNT. Interrupt (Status) and Interrupt Enable registers are broken down into 2 banks: Endpoint Interrupts, USB Interrupts. The MAXP, ENDPOINT INTERRUPT and ENDPOINT INTERRUPT ENABLE registers are used regardless of the direction of the endpoint. The associated CSR registers correspond to the direction of endpoint.

8.2 Register Description

USB Register Map (Base Address = 0x80000500)

Name	Address	Type	Reset	Description
NON INDEXED REGISTERS				
UBFADR	0x00	R/W		Function Address Register
UBPWR	0x04	R/W		Power Management Register
UBEIR	0x08			Endpoint Interrupt Register
UBIR	0x18			USB Interrupt Register
UBEIEN	0x1C			Endpoint Interrupt Enable Register
UBIEN	0x2C			Interrupt Enable Register
UBFRM1	0x30			Frame Number 1 Register
UBFRM2	0x34			Frame Number 2 Register
UBIDX	0x38			Index Register
COMMON INDEXED REGISTER				
MAXP	0x40			IN Max Packet Register
IN INDEXED REGISTERS				
INCSR1	0x44			IN CSR1 Register (EP0 CSR Register)
INCSR2	0x48			IN CSR2 Register
OUT INDEXED REGISTERS				
OCSR1	0x50			OUT CSR1 Register
OCSR2	0x54			OUT CSR2 Register
OFIFO1	0x58			OUT FIFO Write Count 1 Register
OFIFO2	0x5C			OUT FIFO Write Count 2 Register
FIFO REGISTERS				
EP0FIFO	0x80		-	EP0 FIFO Register
EP1FIFO	0x84		-	EP1 FIFO Register
EP2FIFO	0x88		-	EP2 FIFO Register

Function Address Register (UBFADR) **0X80000500**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								UP	FADR						

UP [7]	Function Address Update
UP = 0	Function address doesn't be updated
UP = 1	Function address can be updated with FADR

* The MCU sets this bit whenever it updates the FADR field. This bit is write only register.

FADR [6:0]	Function Address
n	Function address

This register maintains the USB Device Address assigned by the host. The control program should write the value received through a SET_ADDRESS descriptor from host to this register. The address is used for the next token. The UP bit field should be set whenever the FADR field is written. The FADR field is used after the Status phase of a Control transfer, which is signaled by the clearing of the DATA_END bit in the endpoint EP0 CSR.

Power Management Register (UBPWR) **0x80000504**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								-	-	-	-	RST	RSM	SP	ENSP

RST [3]	Type	Reset Signal
1	R	Indicates that 1 reset signaling is received from the host

RSM [2]	Type	Resume Signal
1	R/W	Initiates a resume signaling (10 ~ 15 ms duration)

SP [1]	Type	Suspend Mode
1	R	Indicates that the USB enters suspend mode

ENSP [0]	Type	Enable Suspend Mode
0	R/W	Disable Suspend Mode
1	R/W	Enable Suspend Mode

This register is used for suspend, resume and reset signaling. If ENSP filed is zero, the device will not enter suspend mode. The SP bit field is set by the USB when it enters suspend mode. It is cleared when you clear the RSM bit field by writing zero or when the resume signal from host is received. The USB generates resume signaling while RSM bit is set in suspend mode. The RST bit field is set by USB when reset signal is received from the host.

Endpoint Interrupt Register (UBEIIR) **0x80000508**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								-	-	-	-	-	EP2	EP1	EP0

EP[2:0] [2:0]	Type	EP Interrupt Flag
if bit n is 1	R	Indicates that the USB EP interrupt has been generated

USB Interrupt Register (UBIR) **0x80000518**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								-	-	-	-	-	RST	RSM	SP

RST [2]	Type	Reset Interrupt Flag
1	R	Indicates that the USB has received reset signaling

RSM [1]	Type	Resume Interrupt Flag
1	R	Indicates that the USB has received resume signaling in suspend mode

SP [0]	Type	Suspend Interrupt Flag
1	R	Indicates that the USB has received suspend signalizing Suspend signal is implicit signal that is generated if there is no activity for 3ms.

The USB controller has two interrupt registers: Endpoint interrupt register and USB interrupt register. These registers act as status registers when interrupt is generated. Once interrupt generated, it is needed to read all the interrupt registers and write back to all the registers to clear the interrupt. The endpoint interrupt register UBEIR has three bit fields that correspond to the respective endpoints.

The EP0 interrupt is generated under the following conditions:

1. OUT Packet is ready. ORDY field is set in the CSR register
2. IN Packet is ready. IRDY field is set in the CSR register
3. SENT STALL is set
4. SETUP END is set
5. DATA END is cleared (End of control transfer)

The EP1/E2 interrupt is generated under the following conditions:

For IN endpoints

1. IRDY field is cleared in the CSR register
2. FIFO is flushed
3. SENT STALL is set

For OUT endpoints

1. ORDY field is set in the CSR register.
2. SENT STALL is set

The suspend interrupt is generated when the USB receives suspend signaling. The SP bit field of the UBIR is set whenever there is no activity for 3ms on the bus. This interrupt is disabled in default. The resume interrupt is generated by a USB reset in suspend mode. The USB reset interrupt is generated when USB controller receives the reset signaling from the host.

Endpoint Interrupt Enable Register (UBEIEN) **0x8000051C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								-	-	-	-	-	EP2	EP1	EP0

USB Interrupt Enable Register (UBIEN) **0x8000052C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								-	-	-	-	-	RST	RSM	SP

Corresponding to each interrupt register, there is an INTERRUPT ENABLE register (except resume interrupt enable). By default, the USB reset interrupt is enabled.

If bit = 0, the interrupt is disabled.

If bit = 1, the interrupt is enabled.

Frame Number 1 Register (UBFRM1) **0x80000530**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FRM1							

Frame Number 2 Register (UBFRM2) **0x80000534**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FRM2							

There are two registers, UBFRM1 and UBFRM2, which inform the frame number received from the host. The UBFRM1 denotes the lower byte of frame number. The UBFRM2 denotes the higher byte of frame number.

Frame number = [UBFRM2[7:0] : UBFRM1[7:0]]

USB Index Register (UBIDX) **0x80000538**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IDX							

This Index register is used to indicate the endpoint number while accessing the indexed registers: MAXP, INCSR1/2, OCSR1/2, OFIFO1/2.

*) The following registers denoted by suffix letter of 'n' are index register. Index register means that its address is shared by each end point blocks. So if you want to access the indexed registers of EP0, write 0 to the index register ahead, and for EP1 write 1 to the index register, and so on.

Max Packet Register (MAXPn) **0x80000540**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								-	-	-	-	-	MAXP		

MAXP[2:0] [2:0]	Type	Max Packet Number
n	R/W	Max packet is 8*n

IN CSR1 Register (INCSR1n) **0x80000544**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								-	CTGL	STST	ISST	FLFF	-	FNE	IRDY

CTGL [6]	Type	Clear Data Toggle Bit
1	W	The data toggle bit is cleared

STST [5]	Type	STALL Handshake Issued
1	R	Indicates that the STALL handshake is issued
0	W	Clear by writing 0

ISST [4]	Type	Issue STALL Handshake
1	R/W	Start issuing a STALL Handshake
0	R/W	Clear to end STALL condition

FLFF [3]	Type	Issue FIFO Flush
1	R/W	IN FIFO is flushed
0	R	This bit is cleared by the USB when the FIFO is flushed. The interrupt is generated when this happens. If a token is in progress, the USB waits until the transmission is complete before the FIFO is flushed. If two packets are loaded into the FIFO, only the top-most packet (one that was intended to be sent to the host) is flushed, and the corresponding IRDY bit for that packet is cleared.

FNE [1]	Type	IN FIFO Not Empty
0	R/W	Indicates that no packet of data is in IN-FIFO
1	R/W	Indicates that at least one packet of data is in IN-FIFO

IRDY [0]	Type	IN Packet Ready
0	R	Indicates that the packet has been successfully sent to host
1	W	After writing a packet of data into the IN-FIFO, set this bit to 1.

EP0 CSR Register (EP0CSR) **0x80000544**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CLSE	CLOR	ISST	CEND	DEND	STAL	IRDY	ORDY

*) EP0 CSR register can access by writing “0” to UBIDX register, and use same address as INCSR1.

CLSE [7]	Type	Clear Setup End Bit
1	W	The SEND bit is cleared

CLOR [6]	Type	Clear Output Packet Ready Bit
1	W	The ORDY bit is cleared

ISST [5]	Type	Issue STALL Handshake
1	R/W	Start issuing a STALL Handshake. At the same time, it clears ORDY bit if it decodes an invalid token
0	W	End the STALL condition

CEND [4]	Type	Control Setup End
1	R	Indicates that the control transfer ends before DEND bit is set
0	R	Indicates that the CLSE is written by “1”. At the same time, the USB flushes the FIFO, and invalidates access to the FIFO. That is, when the access to the FIFO is invalidated, this bit is cleared.

DEND [3]	Type	Data End
1	R	Indicates that the one of the following conditions matched. - after loading the last packet of data into the FIFO. (at the same time IRDY is set) - while it clears ORDY after unloading the last packet of data. - for a zero length data phase (at the same time, it clears ORDY and sets IRDY)

STAL [2]	Type	IN Packet Ready
1	R	Indicates that a control transaction is ended due to a protocol violation

IRDY [1]	Type	IN Packet Ready
0	R	Indicates that the packet has been successfully sent to host
1	W	After writing a packet of data into EP0 FIFO, set this bit to 1.

ORDY [0]	Type	OUT Packet Ready
0	R	Indicates that the CLOR has been set to “1”
1	R	Indicates that a valid token is written to the FIFO

IN CSR2 Register (INCSR2n) **0x80000548**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								-	ASET	ISO	MDIN	-	-	-	-

ASET [6]	Auto Set
0	User set IRDY flag when interrupt.
1	IRDY is set automatically.

ISO [5]	ISO Select
0	Configures endpoint to Bulk mode
1	Configures endpoint to ISO mode (Not support)

MDIN [4]	IN/OUT Select
0	Corresponding EPn is configured as OUT Mode
1	Corresponding EPn is configured as IN Mode

OUT CSR1 Register (OCSR1n) **0x80000550**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CTGL	STST	ISST	FLFF	-	-	FFL	ORDY

CTGL [7]	Type	Data Toggle Bit
1	R	The data toggle sequence bit is reset to DATA0

STST [6]	Type	STALL Handshake Issued
1	R	Indicates that the OUT token is ended with a STALL handshake

ISST [5]	Type	Issue STALL Handshake
1	R/W	Start issuing a STALL Handshake
0	R/W	End the STALL Condition

FLFF [4]	Type	Issue FIFO Flush
1	R/W	OUT FIFO is flushed
0	R/W	Stop flushing FIFO

FFL [1]	Type	OUT FIFO Full
1	R	Indicates that no more packets can be accepted

ORDY [0]	Type	OUT Packet Ready
0	R	Indicates that once the MCU reads the FIFO for all the packet
1	R	Once it has loaded a packet of data into the FIFO.

OUT CSR2 Register (OCSR2n) **0x80000554**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ACLR	-	-	-	-	-	-	-

ACLR [7]	Type	Auto Clear
1	R/W	Whenever the MCU reads data from the OUT FIFO, ORDY of OCSR1n will automatically be cleared by the core, without any intervention from MCU.

OUT FIFO Write Count 1 Register (OFIFO1n) **0x80000558**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OFIFO1n							

OUT FIFO Write Count 2 Register (OFIFO2n) **0x8000055C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OFIFO2n							

There are two register, OFIFO1n and OFIFO2n, which maintain the write count. OFIFO1n maintains the lower bytes, while OFIFO2n maintains the higher byte. When ORDY bit of OCSR1n is set for OUT endpoints, these registers maintain the number of bytes in the packet due to be unloaded by the MCU.

EP0 FIFO Register (EP0FIFO) **0x80000580**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FIFO							

EP1 FIFO Register (EP1FIFO) **0x80000584**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FIFO							

EP2 FIFO Register (EP2FIFO) **0x80000588**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FIFO							

CHAPTER 9

UART/IrDA CONTROLLER

9 UART / IrDA

9.1 Functional Description

The TCC720 has 1 simple UART module that can be used in programming the system software or IrDA interfacing. The block diagram of UART is in the following figure.

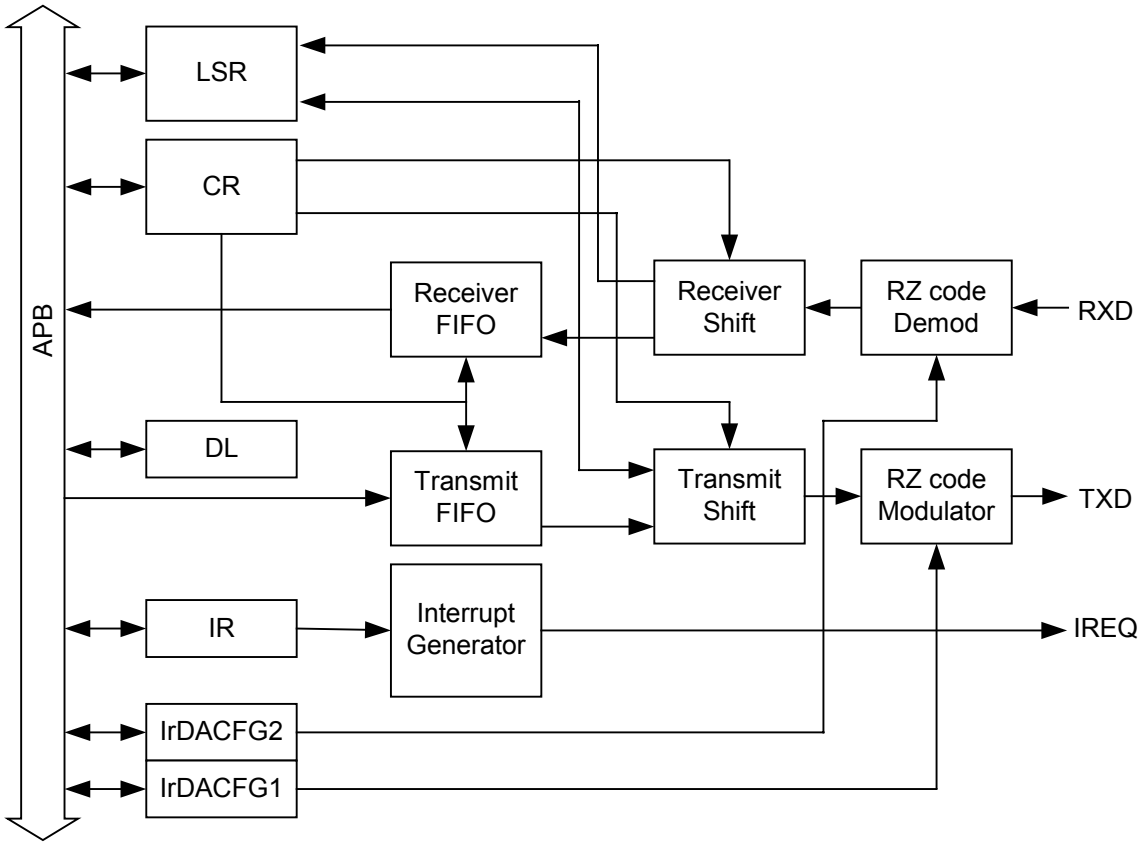


Figure 9.1 UART Block Diagram

This UART is simplified version of UART16550, it provides only a simple interface (TXD, RXD) between host system and TCC720 system.
In the UART, there are two FIFO blocks for transmission and receiving link. Transmission FIFO has 4 bytes depth, receiving FIFO has 8 bytes depth.

UART can also be used as IrDA interfacing. There is a signal transformer between IrDA signal and UART signal.

9.2 Register Desciption

UART/IrDA Register Map (Base Address = 0x80000600)

Name	Address	Type	Reset	Description
RXD	0x00	R	-	Receiver Buffer Register
TXD	0x00	W	-	Transmitter Holding Register
DL	0x04	R/W	0x0000	Divisor Latch Register
IR	0x08	R/W	0x000	Interrupt Register
CR	0x0C	R/W	0x000	UART Control Register
LSR	0x10	R	0x0101	Status Register
IrDACFG1	0x14	R/W	0x0003	IrDA Configuration Register 1
IrDACFG2	0x18	R/W	0x4da1	IrDA Configuration Register 2

Receiver Buffer Register (RXD) **0x80000600**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Received Data (when reading)							

Whenever FRX flag of IR register is set, or RA flag of LSR register is set, reading of this register gets the 1 byte of received data.

Transmitter Holding Register (TXD) **0x80000600**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0								Transmitting Data (when writing)							

When the transmission FIFO is not full, writing of this register fills that data to transmission FIFO. Checking TF flag of LSR register can monitor the status of a transmission FIFO.

Divisor Latch Register (DL) **0x80000604**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Divisor Latch Value															

This is for generation of the desired baud rate clock. This register is set to 0 at reset, UART is disabled until this register is set by non-zero value. The value should be equal to (UART clock speed) / (16 * desired baud rate). The UART clock is generated by clock generator block. It is recommended that the frequency of UART clock is set to 3.6864MHz, so the desired baud rate can be acquired by writing (230400/baud rate) to DL register.

Interrupt Register (IR) **0x80000608**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					ERS	ETX	ERX	0	FRS	FTX	FRX	0	QRS	QTX	QRX

ERS [10]	Receiver Line Status Interrupt
0	disabled
1	enabled

ETX [9]	Transmitter Holding Register Empty Interrupt
0	disabled
1	enabled

ERX [8]	Receiver Data Available Interrupt
0	disabled
1	enabled

FRS [2]	Flag for Receiver Line Status Interrupt
0	Interrupt has not generated
1	Interrupt has generated, but not cleared

FTX [1]	Flag for Transmitter Holding Register Empty Interrupt
0	Interrupt has not generated
1	Interrupt has generated, but not cleared

FRX [0]	Flag for Receiver Data Available Interrupt
0	Interrupt has not generated
1	Interrupt has generated, but not cleared

*) FLS, FTX, FRX is set or cleared regardless of each enable settings.

QRS [2]	Request for Receiver Line Status Interrupt
0	Interrupt has not generated
1	Interrupt has generated, but not cleared

QTX [1]	Request for Transmitter Holding Register Empty Interrupt
0	Interrupt has not generated
1	Interrupt has generated, but not cleared

QRX [0]	Request for Receiver Data Available Interrupt
0	Interrupt has not generated
1	Interrupt has generated, but not cleared

*) QRLS, QTHE, QRDA is only set when each enable bit is set to 1.

UART/IrDA Control Register (CR) **0x8000060C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0						NO	BK	TF	RF	FIFO		PR		ST	B7

NO [9]	Start Bit Width Check
0	Check if the pulse width of start bit is more than 0.5 bit duration of baud rate
1	Don't check the pulse width of start bit (used only for test or boot mode)

BK [8]	Break Control Bit
0	Normal operation
1	Bit '0' is transmitted regardless of THR

TF [7]	Reset Transmitter FIFO
1	The transmitter FIFO is cleared

RF [6]	Reset Receiver FIFO
1	The receiver FIFO is cleared

FIFO [5:4]	RX FIFO Level Select
n	0 = 1byte FIFO, 1 = 2 byte FIFO 2 = 4 byte FIFO, 3 = 7 byte FIFO

*) This field controls the RDA(Receive Data Available) flag or interrupt only, that is the actual FIFO depth can't be modified and fixed to 8. If this field is set to 1, it means that the RDA flag or interrupt is influenced when the number of received data in the RX FIFO is 2. It is recommended that this field is set to 0, so right after reception of some data, the RDA flag or interrupt can be generated.

PR [3:2]	Parity Bit Select
0	Even parity
1	Odd parity
2, 3	Parity is disabled

ST [1]	Stop Bit
0	1 Stop bit
1	2 Stop bit

B7 [0]	Number of Bits per Character
0	8 bit
1	7 bit

Line Status Register (LSR) **0x80000610**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0											TE	TF	FE	PE	RA

TE [4]	Transmitter FIFO
0	Not empty
1	Empty

*) Transmitter FIFO depth is fixed to 4.

TF [3]	Transmitter FIFO
0	Not full
1	Full

*) Transmitter FIFO depth is fixed to 4.

FE [2]	Framing Status
0	Correct stop bit is received
1	The received data in the FIFO don’t have valid stop bit

PE [1]	Parity Status
0	Correct parity bit is received
1	The received data in the FIFO don’t have valid parity bit

RA [0]	Received FIFO Status
0	No data has received
1	At least 1 received data is in the FIFO

IrDA Configuration Register 1 (IrDACFG1) **0x80000614**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	P1	POL	LB	0								PW			

EN [15]	IrDA TX Enable
0	IrDA TX is disabled, UART mode is used
1	IrDA TX is enabled

P1 [14]	Transmit Pulse Type
0	Pulse width is proportional to selected baud speed
1	Pulse width is proportional to UART base clock speed

POL [13]	Transmit Pulse Polarity
0	TX ‘0’ data is converted to level high pulse
1	TX ‘0’ data is converted to level low pulse

LB [12]	Loopback
0	Normal operation
1	Transmitted data is fed back to RX port.

PW [3:0]	IrDA RZ Pulse Width
n	Represents pulse width of TX ‘0’ data. If n = 3, during 3/16 of its 1 bit period or 3 * 3686400^{-1} sec, the high pulse is generated

IrDA Configuration Register 2 (IrDACFG2) **0x80000618**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	P1	POL	0	DEC				MAX1				MIN1			

EN [15]	IrDA RX Enable
0	IrDA RX is disabled, UART mode is used
1	IrDA RX is enabled

P1 [14]	Receiver Pulse Type
0	Received pulse width is proportional to selected baud speed
1	Received pulse width is proportional to UART base clock speed

POL [13]	Receive Pulse Polarity
0	The polarity of received data is not inverted
1	The polarity of received data is inverted

DEC [11:8]	RX Data Decision Time
n	The decision point for receiving data, its unit has 1/16 of baud rate.

MAX1 [7:4]	Maximum number of “1”s
n	The maximum number of “1”s to decide the received IrDA (RZ) signal as 0. If P1 is set to 1, MAX1 has the unit of 1/1843200 sec, or if P1 is set to 0, the unit of MAX1 has 1/16 of baud rate.

MIN1 [3:0]	Minimum number of “1”s
n	The minimum number of “1”s to decide the received IrDA (RZ) signal as 0. If P1 is set to 1, MIN1 has the unit of 1/1843200 sec, or if P1 is set to 0, the unit of MIN1 has 1/16 of baud rate.

CHAPTER 10

GSIO PORT

10 GSIO (General Purpose Serial Input/Output) PORT

The TCC720 has three GSIOs for communication between the TCC720 and other devices that have serial interface. All the pins in the GSIOs are multiplexed with GPIOs. User can program what these multiplexed pins are used for. The GSIO block has 4 pins; SDI, SDO, SCK, FRM. The SDO is the serial data output pin, the SDI is the serial data input pin, the SCK is the serial clock pin and the FRM is frame pin. The base clock is generated by dividing the PCLK programming the GSIO control register GSCR. The SCK is generated from the basic clock in every data transfers. Various types of serial interface can be programmed using GSIO control field in the GSCR. There are 5 control registers for GSIOs; GSCR0, GSCR1, GSCR2, GSCR3, and GSICR. The start time of transfer can be controlled with programming the delay counter field in the GSCRn. The base counter increments at every base clock right after writing the data into the GSDRn. The serial data starts to come out when delay counter value are same to base counter value. The word size of transfer can be programmed from 1 bit to 16 bits. The frame1 and the frame2 fields specify the start and end point of transition based on base counter. The frame polarity defines whether the frame signal is low active or high active signal. The Last Clock mask filed is for special serial interface, which makes the last clock pulse masked.

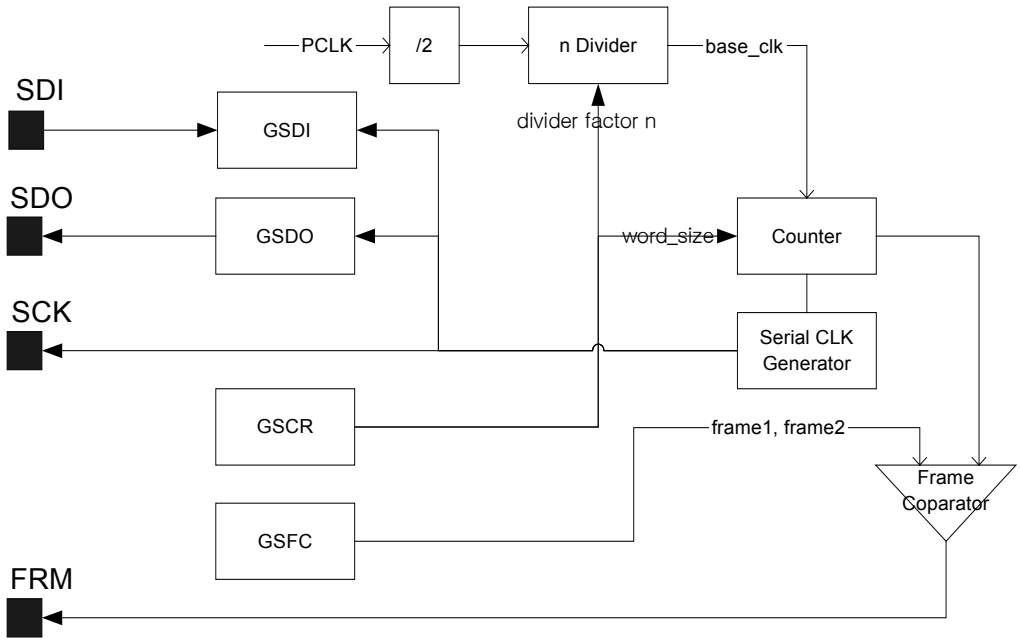


Figure 10.1 GSIO Block Diagram

GSIO Register Map (Base Address = 0x80000700)

Name	Address	Type	Reset	Description
GSDO0	0x00	R/W		GSIO0 Output Data Register
GSDI0	0x04	R/W		GSIO0 Input Data Register
GSCR0	0x08	R/W		GSIO0 Control Register
GSGCR	0x0C	R/W		GSIO Global Control Register
GSDO1	0x10	R/W		GSIO1 Output Data Register
GSDI1	0x14	R/W		GSIO1 Input Data Register
GSCR1	0x18	R/W		GSIO1 Control Register
GSDO2	0x20	R/W		GSIO2 Output Data Register
GSDI2	0x24	R/W		GSIO2 Input Data Register
GSCR2	0x28	R/W		GSIO2 Control Register
GSDO3	0x30	R/W		GSIO3 Output Data Register
GSDI3	0x34	R/W		GSIO3 Input Data Register
GSCR3	0x38	R/W		GSIO3 Control Register

GSIOn Output Data Register (GSDO0, GSDO1, GSDO2, GSDO3) **0x800007x0**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												WORD[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data to GSIO Output Pin															

WORD[3:0] [19:16]	GSIO word size
n	GSIO data has (n+1) bit unit, n = 0 ~ 15

*) This field is valid only if WS of GSCRn register is set to 1.

GSIOn Input Data Register (GSDI0, GSDI1, GSDI2, GSDI3) **0x800007x4**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data from GSIO Input Pin															

*) These registers is updated every writing to GSDO registers.

GSIOn Control Register (GSCR0, GSCR1, GSCR2, GSCR3) **0x800007x8**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	MS	WORD				WS	DIV							CP	CM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DELAY		FP	FRM1						FRM2					

EN [31]	GSIO Enable
0	GSIO Disabled
1	GSIO Enabled

MS [30]	First Bit Select
0	LSB first
1	MSB first

WORD [29:26]	GSIO word size
n	GSIO data has (n+1) bit unit, n = 0 ~ 15

WS [25]	Word Size Select
0	GSIO word size is determined by WORD of GSCRn register
1	GSIO word size is determined by BW of GSDO register

DIV [24:18]	GSIO base clock speed control
n	GSIO base clock has 1/(2n+2) of PCLK frequency, n = 0 ~ 127

CP [17]	GSIO clock polarity
0	SDO changes at SCK falling
1	SDO changes at SCK rising

CM [16]	Last clock mask
0	No mask. GSIO clock is generated for every SDO.
1	GSIO clock is masked at the last SDO period.

DELAY [14:13]	Initial delay for serial transmission
n	GSIO transmission starts after n base clock has generated.

FP [12]	Frame pulse polarity
0	FRM has low active pulse
1	FRM has high active pulse

FRM1 [11:6]	Frame pulse start position
n	Frame pulse starts after n base clock has generated

FRM2 [5:0]	Frame pulse end position
n	Frame pulse ends after n base clock has generated

GSIO Global Control Register (GSGCR) **0x8000070C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G3	G2	G1	G0	IEN3	IEN2	IEN1	IEN0	FLG3	FLG2	FLG1	FLG0	Busy3	Busy2	Busy1	Busy0

G[3:0] [15:12]	GPIO_B[23:21] Other Function Signal Select
if bit n is 1	FRM, SCK, SDO output of GSIOn is come out from GPIO_B[23:21]

*) If multiple bit of G[3:0] is set to 1, the output of each GSIO is orred and come out from GPIO_B[23:21]

IEN[3:0] [11:8]	GSIO Interrupt Enable
if bit n is 1	GSIOn Interrupt is enabled
0	GSIOn Interrupt is disabled

FLG[3:0] [7:4]	R/W	GSIO Interrupt Flag
if bit n is 1	R	GSIOn operation (read/write) has been completed.
if bit n is 1	W	Clear FLG[n] field

*) If an interrupt of a GSIO is enabled, GSIO interrupt is generated when the GSIO operation is completed. These FLGn can be used to distinguish which GSIO has generated the interrupt. These flags are cleared by writing “1” at the corresponding flag.

Busy[3:0] [3:0]	GSIO Cycle Busy Flag
if bit n is 0	GSIOn transmission has finished, and can transmit another serial data.
if bit n is 1	GSIOn transmission is in operation, so it cannot accept another serial data.

The following figures represent some kinds of various GSIO operations.

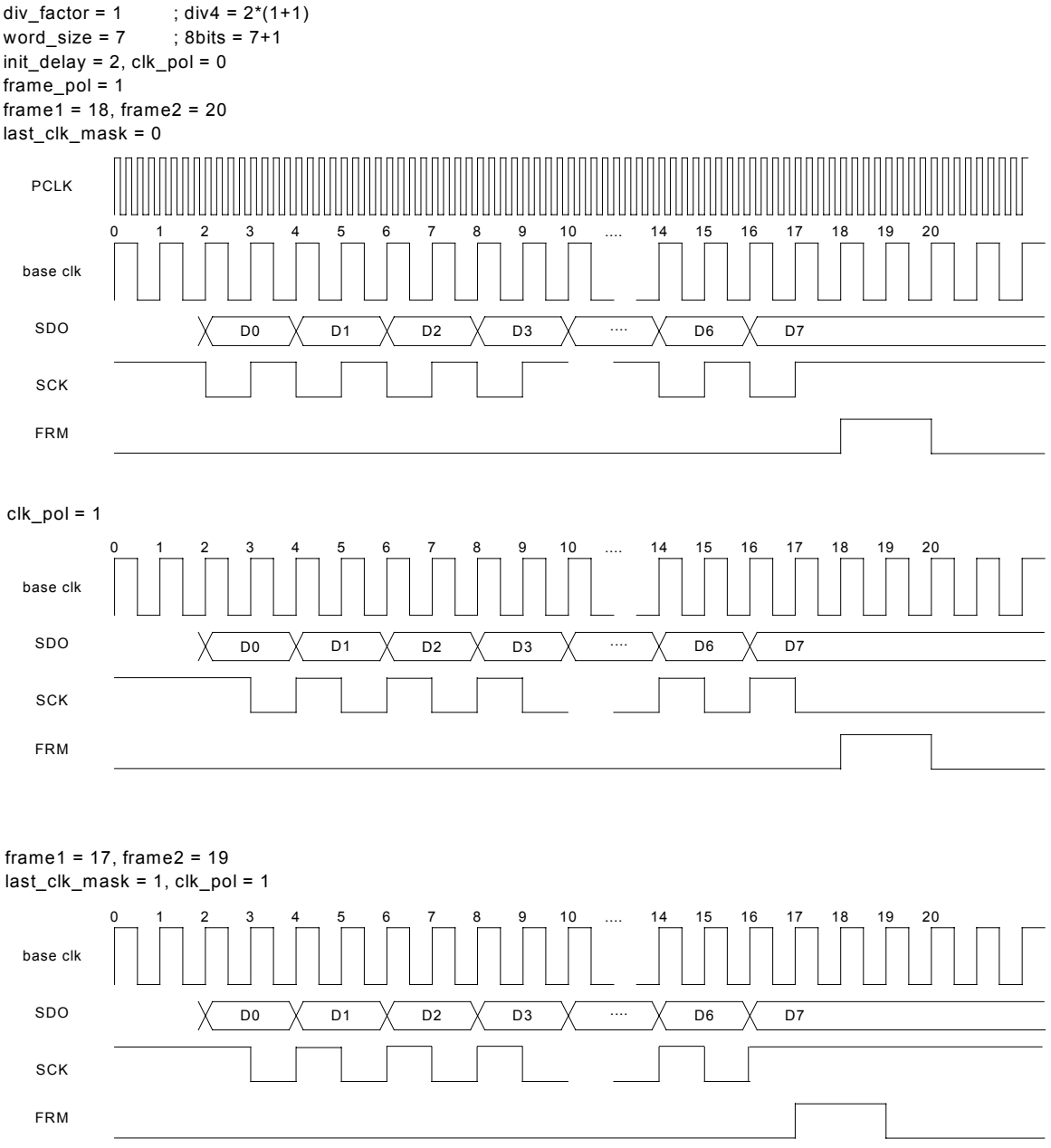


Figure 10.2 GSIO operation

CHAPTER 11

MISCELLANEOUS PERIPHERALS

11 MISCELLANEOUS PERIPHERALS

11.1 ADC

The TCC720 has 3-input general purpose low-power ADC for battery level detection, remote control interface, touch screen interface, etc. It is a CMOS type 8bit/10bit changeable A/D converter which combines suitable blocks for various purpose such as an analog input multiplexer, auto offset calibration comparator, 8bit/10bit changeable successive approximation register (SAR), etc.

Various operating option can be set by using ADCCON register, it can convert up to 8 analog input and be operated as 10bit ADC at about 200ksps rates, as well as 8bit ADC at about 250ksps rates. It has standby mode for power consumption also.

The output of ADC can be read from the ADCDATA register.

ADC Control Register (ADCCON) **0x80000A00**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							ADEN				STB	M8	ASEL		

ADEN [8:5]	ADC Sample Rate Select
adc	Sample Rate = f_{PCLK} / ((adc+1) * 16)

STB [4]	ADC Standby Select
1	ADC goes to standby mode
0	ADC starts operating

M8 [3]	ADC Bit Select
1	Select 8bit conversion mode
0	Select 10bit conversion mode

ASEL [2:0]	ADC Input Select
n	ADINn pin is selected as ADC input signal (n = one of 0, 2, 4)

ADC Data Register (ADCDATA) **0x80000A04**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					ADATA										FLG

ADATA [10:1]	ADC Data
adc	When 8bit mode, lsb 2bit must be ignored. When 10bit mode, ADC data = adc

FLG [0]	ADC Status Flag
1	A/D conversion is finished, data is stable
0	A/D conversion is on processing, data is unstable

11.2 CODEC

The TCC720 has on-chip sigma delta type 16bit audio stereo CODEC for high grade digital audio en-decoder systems. It contains various blocks such as compensation filter, digital volume attenuator, de-emphasis filter, FIR filter, sinc filter, digital sigma-delta modulator, analog postfilter, anti-image filter, etc.

CODEC Control Register (CDC_CTRL) **0x80000A08**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				ZID	ZC	AIS	EMP	IIS	FSEL		RST	DA		AD	

ZID [11]	Zero Input Detection Control
0	DAC Zero Input Detection is enabled
1	DAC Zero Input Detection is disabled

*) If the input data has the condition where the lower 4bits of the input data are DC and the remaining upper bits are all “0” or all “1” has continued 8192 cycles of LRCK (=32fs), then zero input is detected, and the analog postfilter output will be immediately forced to VREF.

ZC [10]	Zero Cross Enable Control
0	DAC Zero Cross Enable Control is disabled
1	DAC Zero Cross Enable Control is enabled

*) If DAC postfilter output data has the condition where it is cross VREF reference level, DAC programmable gain amplifier control register is up-dated. It is used to improve click and pop-noise. If ZC is 0, DAC Programmable Gain Control Register is modified by CDC_GAIN register.

AIS [9]	Analog Input Selection
0	LCH_IN and RCH_IN input is processed
1	MIC_IN input is processed

EMP [8]	De-emphasis Control
0	De-emphasis is disabled
1	De-emphasis is enabled

*) This bit is only useful when 44.1KHz mode.

IIS [7]	Data Format Select
0	16bit Right Justified Mode is selected
1	16bit IIS Mode is selected

FSEL [6:5]	Sample Frequency Select
0, 3	32KHz, 44.1KHz, 48KHz mode (System clock must be 256*fs)
1	16KHz, 22.05KHz mode (System clock must be 512*fs)
2	8KHz, 11KHz mode (System clock must be 512*fs)

RST [4]	Reset Signal
0	ADC, DAC reset is released
1	ADC, DAC reset is generated

DA [3:2]	DAC Mode Selection
00	DAC normal operation mode
01	DAC mute OFF, DAC power down ON
10	DAC mute ON , DAC power down OFF
11	DAC mute ON , DAC power down ON

AD [1:0]	ADC Mode Selection
00	ADC normal operation mode
01	ADC mute OFF, DAC power down ON
10	ADC mute ON , DAC power down OFF
11	ADC mute ON , DAC power down ON

CODEC Gain Control Register (CDC_GAIN) **0x80000A0C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0										ADR		DATA			

ADR [5:4]	Gain Register Select
00	ADC Left Channel is selected
01	ADC Right Channel is selected
10	DAC Left Channel is selected
11	DAC Right Channel is selected

DATA [3:0]	Gain Data
n	When ADR field selects ADC Gain, ADC gain = 1.5 * n dB When ADR field selects DAC Gain, DAC gain = - 2.0 * n dB

11.3 ETCETERA

Count Leading Zero Register (CLZ) **0x80000A10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLZ[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLZ[15:0]															

When X is written to CLZ register, the number of zero counting from MSB of X can be calculated by reading CLZ register.

If the value returned by reading CLZ register is Y, the number of zero counting from MSB of X is 32 – Y.

USB Port Control Register (USB_CTRL) **0x80000A14**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												PSL	CNT	OVR	SW

*) Must be remained to 0

TEST Mode Register (TSTSEL) **0x80000A18**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0											US1	US0	B2	B1	AC

*) Must be remained 0

CHAPTER 12

DMA CONTROLLER

12 DMA CONTROLLER

12.1 Functional Description

TCC720 has a simple 1-channel DMA controller for data transfer. It can be used to transfer data from some kind of memory block to other kind of memory block.

The block diagram of DMA controller is in the following figure.

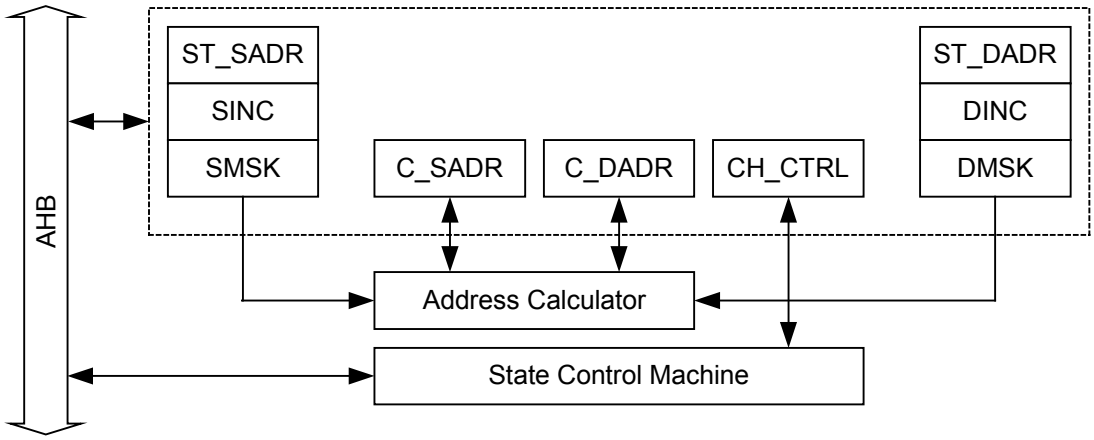


Figure 12.1 DMA Controller Block Diagram

There are various kinds of transfer modes for DMA operation. The following table represents each type of transfer according to CHCTRL register.

Table 12.1 Type of DMA transfer

LOCK (CHCTRL[11])	TYPE (CHCTRL[10:8])	Description
0	000	SINGLE type transfer without LOCK
1	000	SINGLE type transfer with LOCK
0	001	HW_ARBIT type transfer without LOCK
1	001	HW_ARBIT type transfer with LOCK
0 or 1	101	HW_BURST type transfer
0	010	SW_ARBIT type transfer without LOCK
1	010	SW_ARBIT type transfer with LOCK
0 or 1	110	SW_BURST type transfer

In SINGLE type transfer, 1 Hop of transfer occurs only once at every DMA requests. The 1 Hop of transfer means 1 burst read followed by 1 burst write. 1 burst means 1, 2 or 4 consecutive read or write cycles defined by CHCTRL[7:6] field.

Hardware type transfer (HW_ARBIT, HW_BURST) means that the DMA transfer triggered by external or internal hardware blocks selected by CHCTRL[28:16] field. This field has same mapping with interrupt enable flag of interrupt controller, so the DMA transfer can be occurred as like as interrupt is generated.

Software type transfer (SW_ARBIT, SW_BURST) means that the DMA transfer triggered by CHCTRL[0] flag (enable flag). When this flag is set to 1, the DMA transfer begins at the same time.

Arbitration type transfer (HW_ARBIT, SW_ARBIT) means that at the end of every HOP transfer, the AHB bus is released from DMA channel so other master can occupy the bus when the master has requested the bus.

Burst type transfer (HW_BURST, SW_BURST) means that once the DMA transfer occurs, all of transfers are executed without further DMA requests.

Lock field controls the LOCK signal (refer to AHB specification), so that when the LOCK is set to 1, the corresponding transfer doesn't be bothered by other AHB masters like LCD controller, ARM etc. This field is only meaningful for non-burst type of transfers.

12.2 Register Description

DMA Controller Register Map (Base Address = 0x80000E00)

Name	Address	Type	Reset	Description
ST_SADR	0x00	R/W	-	Start Address of Source Block
SPARAM	0x04/0x08	R/W	-	Parameter of Source Block
C_SADR	0x0C	R	-	Current Address of Source Block
ST_DADR	0x10	R/W	-	Start Address of Destination Block
DPARAM	0x14/0x18	R/W	-	Parameter of Destination Block
C_DADR	0x1C	R	-	Current Address of Destination Block
HCOUNT	0x20	R/W	0x00000000	Initial and Current Hop count
CHCTRL	0x24	R/W	0x00000000	Channel Configuration
CLRDRQ	0x28	W	-	Clear End of DMA flag

Start Source Address Register (ST_SADR) **0x80000E00**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST_SADR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_SADR[15:0]															

*) This register contains the start address of source block for DMA transfer. The transfer begins reading data from this address.

Start Destination Address Register (ST_DADR) **0x80000E10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ST_DADR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_DADR[15:0]															

*) This register contains the start address of destination block for DMA transfer.

Source Block Parameter Register (SPARAM) **0x80000E04 / 0x80000E08**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMASK[23:8]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMASK[7:0]								SINC[7:0]							

SMASK [23:8]	Source Address Mask Register
0	non-masked
1	masked

*) Each bit field controls the corresponding bit of source address field. That is, if SMASK[23] is set to 1, the 28th bit of source address is masked, and if SMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is unchanged during DMA transfer. This function can be used to generate circular buffer address.

SINC [7:0]	Source Address Increment Register
sinc	Source address is added by amount of sinc at every write cycles. sinc is represented as 2's complement, so if SINC[7] is 1, the source address is decremented.

The addresses of DMA transfer have 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

Destination Block Parameter Register (DPARAM) **0x80000E14 / 0x80000E18**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMASK[23:8]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASK[7:0]								DINC[7:0]							

DMASK [23:8]	Destination Address Mask Register
0	non-masked
1	masked

*) Each bit field controls the corresponding bit of source address field. That is, if DMASK[23] is set to 1, the 28th bit of source address is masked, and if DMASK[22] is set to 1, the 27th bit of source address is masked, and so on. If a bit is masked, a corresponding bit of address bus is unchanged during DMA transfer. This function can be used to generate circular buffer address.

DINC [7:0]	Destination Address Increment Register
dinc	Destination address is added by amount of dinc at every write cycles. dinc is represented as 2's complement, so if DINC[7] is 1, the destination address is decremented.

The addresses of DMA transfer have 32bit wide, but the upper 4bit of them are not affected during DMA transfer. If the source or destination address reaches its maximum address space like 0x7FFFFFFF or 0x2FFFFFFF, the next transfer is starting from 0x70000000 or 0x20000000 not from 0x80000000 or 0x30000000.

Current Source Address Register (C_SADR) **0x80000E0C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_SADR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C_SADR[15:0]															

*) This register contains current source address of DMA transfer.

Current Destination Address Register (C_DADR) **0x80000E1C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_DADR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C_DADR[15:0]															

*) This register contains current destination address of DMA transfer.

HOP Count Register (HCOUNT) **0x80000E20**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C_HCOUNT[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_HCOUNT[15:0]															

C_HCOUNT [31:16]	Current Hop Count
cn	Represent cn number of Hop transfer remains

ST_HCOUNT [15:0]	Start Hop Count
sn	Represent sn number of Hop transfer is transferred.

*) At the beginning of transfer, the C_HCOUNT is stored by ST_HCOUNT register. And at the end of every hop transfer, this is decremented by 1 until reached to 0.

Channel Control Register (CHCTRL) **0x80000E24**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			DMASEL[12:0]												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONT	PRI			LOCK	TYPE			BSIZE		WSIZE		FLAG	IEN	REP	EN

DMASEL [28:16]	Select Source of DMA Request
non-zero	Each bit field selects corresponding signal as a source for DMA request. The bit-map of this register is identical with the IEN of interrupt controller. So if you want to use EXINT0 pin as a source of DMA request, set DMASEL[0] as 1 and select HW_ARBIT or HW_BURST type transfer. If multiple bits of this register is set, all the corresponding signal can generate DMA request for this channel.

CONT [15]	Issue Locked Transfer
0	DMA transfer begins from ST_SADR / ST_DADR address
1	DMA transfer begins from C_SADR / C_DADR address It must be used after the former transfer has been executed, so that C_SADR and C_DADR contains meaningful values.

PRI [14:12]	Priority
0	Priority 0 is equal to disable DMA transfer
non-zero	DMA channel is enabled only when have non-zero priority.

LOCK [11]	Issue Locked Transfer
0	DMA transfer executed without lock property
1	DMA transfer executed with lock property

*) Lock field controls the LOCK signal (refer to AHB specification), so that when the LOCK is set to 1, the corresponding transfer doesn't be bothered by other AHB masters like LCD controller, ARM etc. This field is only meaningful for non-burst type of transfers.

TYPE [10:8]	Transfer Type
000	SINGLE transfer
001	HW_ARBIT transfer
101	HW_BURST transfer
010	SW_ARBIT transfer
110	SW_BURST transfer

*) Please refer to table 12.1 for detailed information of each transfer types.

BSIZE [7:6]	Burst Size
0	1 Hop transfer consists of 1 pair of read and write cycle.
1	1 Hop transfer consists of 2 pair of read and write cycle
2, 3	1 Hop transfer consists of 4 pair of read and write cycle

WSIZE [5:4]	Word Size
0	byte transfer
1	half word transfer
2, 3	word transfer

FLAG [3]	DMA Flag
1	Represents that all hop of transfers are fulfilled

IEN [2]	Interrupt Enable
1	At the same time FLAG goes to 1, DMA interrupt request is generated.

*) To generate IRQ or FIQ interrupt, the corresponding enable bit in the interrupt controller must be set to 1 ahead.

CONT [1]	Continuous Transfer
0	After all of hop transfer has executed, the DMA channel is disabled
1	The DMA channel remains enabled, so when another DMA request has occurred, the DMA channel start transfer data again with the same manner (type, address, increment, mask) as the latest transfer of that channel.

*) This bit is only valid if the transfer type is hardware and non-burst type transfer.

EN [0]	DMA Channel Enable
0	DMA channel is disabled or terminated. Once terminated, user must make HCOUNT to 0 not to continue transfer after channel is re-enabled.
1	DMA channel is enabled. If software type transfer is selected, this bit generates DMA request directly, or if hardware type transfer is used, the interrupt request generates DMA request.

CHAPTER 13

MEMORY CONTROLLER

13 MEMORY CONTROLLER

13.1 Overview

TCC720 has a memory controller for various kind of memory for digital media en-decoding system. It can manipulate SDRAM, Flash (NAND, NOR type), ROM, SRAM type memories, and also support the IDE interface for HDD or USB2.0 device. It has configurable data bus width through the GPIO pin or each configuration register. The data bus width can be configured for each chip select separately

The memory controller provide the power saving function for SDRAM (self refresh).

The following figure represents the block diagram of memory control unit.

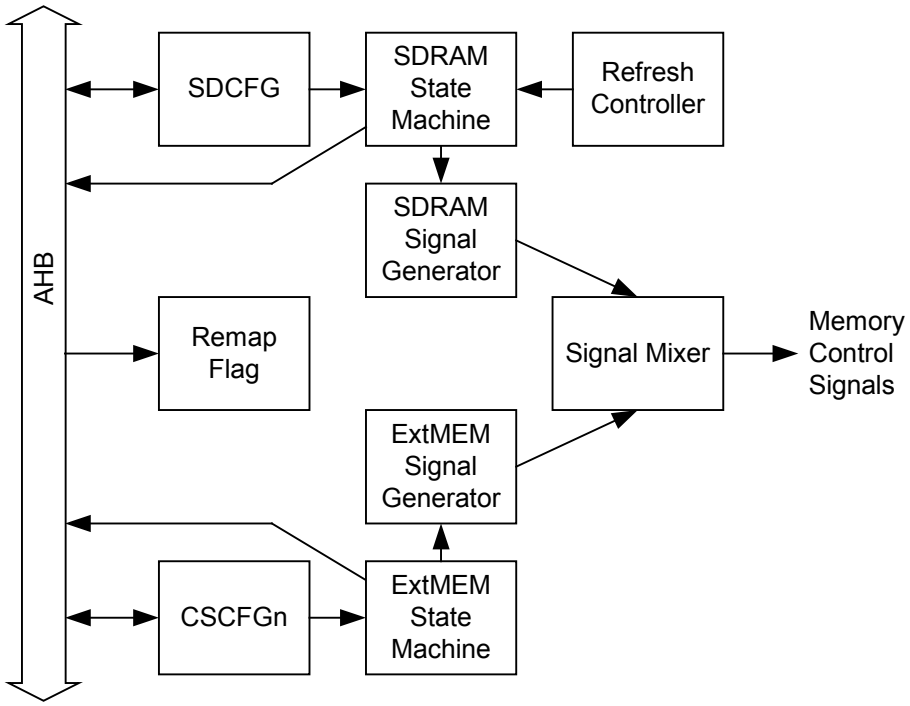


Figure 13.1 Memory Controller Block Diagram

The registers for memory controller block have the base address of 0xF0000000.

Memory Controller Register Map (Base Address = 0xF0000000)

Name	Address	Type	Reset	Description
SDCFG	0x00	R/W	0x4268A020	SDRAM Configuration Register
SDFSM	0x04	R	-	SDRAM FSM Status Register
MCFG	0x08	R/W	0xZZZZ_02	Miscellaneous Configuration Register
TST	0x0C	W	0x0000	Test mode register (must be remained zero)
CSCFG0	0x10	R/W	0x0B405601	External Chip Select 0 Configuration Register (Initially set to SRAM)
CSCFG1	0x14	R/W	0x0150569A	External Chip Select 1 Configuration Register (Initially set to IDE)
CSCFG2	0x18	R/W	0x0060569A	External Chip Select 2 Configuration Register (Initially set to NAND)

NAND flash Register Map (Base Address = N * 0x10000000)

Name	Address	Type	Reset	Description
CMD	0x00	R/W	-	Command Cycle Register
LADDR	0x04	W	-	Linear Address Cycle Register
BADDR	0x08	W	-	Block Address Cycle Register
IADDR	0x0C	W	-	Single Address Cycle Register
DATA	0x10	R/W	-	Data Access Cycle Register

***) N represents BASE field of CSCFGn registers.**

13.2 SDRAM Controller

SDRAM controller can control from 64Mbit up to 256Mbit SDRAM. In TCC720 system, the SDRAM contains almost parts for system operation. (program, data, ESP buffer, etc is located in SDRAM).

The SDRAM parameter such as size, refresh period, RAS to CAS delay, refresh to idle delay can be programmed by internal register.

The registers for SDRAM controller is as the followings.
Refer to SDRAM cycle diagram in figure 13.2

SDRAM Configuration Register (SDCFG) **0xF0000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CL	BW	CW		SDBASE				RC			RCD			RD[2:1]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RD0	RP			RW	Refresh										

*) The reset value means the following configuration.

CL=2cycle, CW=8bit, BW=16bit, SDBASE=2, RC =3, RCD=2, RD=1, RP=2, RW=12bit, Refresh=0x20

CL [31]	CAS Latency (tCL)
0	CAS latency is 2 cycle
1	CAS latency is 3 cycle

BW [30]	Bus Width Select
0	Bus width for SDRAM is 32 bit
1	Bus width for SDRAM is 16 bit

CW [29:28]	CAS Width
0, 1	8 bit is used for CAS address
2	9 bit is used for CAS address
3	10 bit is used for CAS address

SDBASE [27:24]	SDRAM Base Address
N	Indicates the MSB 4bit of SDRAM area. That is SDRAM base = 0x**N**0000000
RC [23:21]	Delay of Refresh to Idle (tRC)
n	n number of HCLK cycle is used to meet the refresh to idle delay time

RCD [20:18]	Delay of RAS to CAS (tRCD)
n	(n+1) number of HCLK cycle is used to meet the RAS to CAS delay time

RD [17:15]	Delay of Read to Precharge (tRD)
n	n number of HCLK cycle is used to meet the read to precharge time

RP [14:12]	Delay of Precharge to Refresh (tRP)
n	(n+1) number of HCLK cycle is used to meet the precharge to refresh time

RW [11]	RAS Width
0	12bit is used for RAS address bus
1	13bit is used for RAS address bus

Refresh [10:0]	Refresh Cycle
n	Every (n * 16 + 15) number of HCLK cycle has passed, the SDRAM refresh request is generated. If on going cycle has finished, the refresh cycle starts. Real refresh period depends on the period of HCLK.

SDRAM FSM Status Register (SDFSM) **0xF0000004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0				SDFSM											

*) Represents current status of finite state machine in the SDRAM controller.

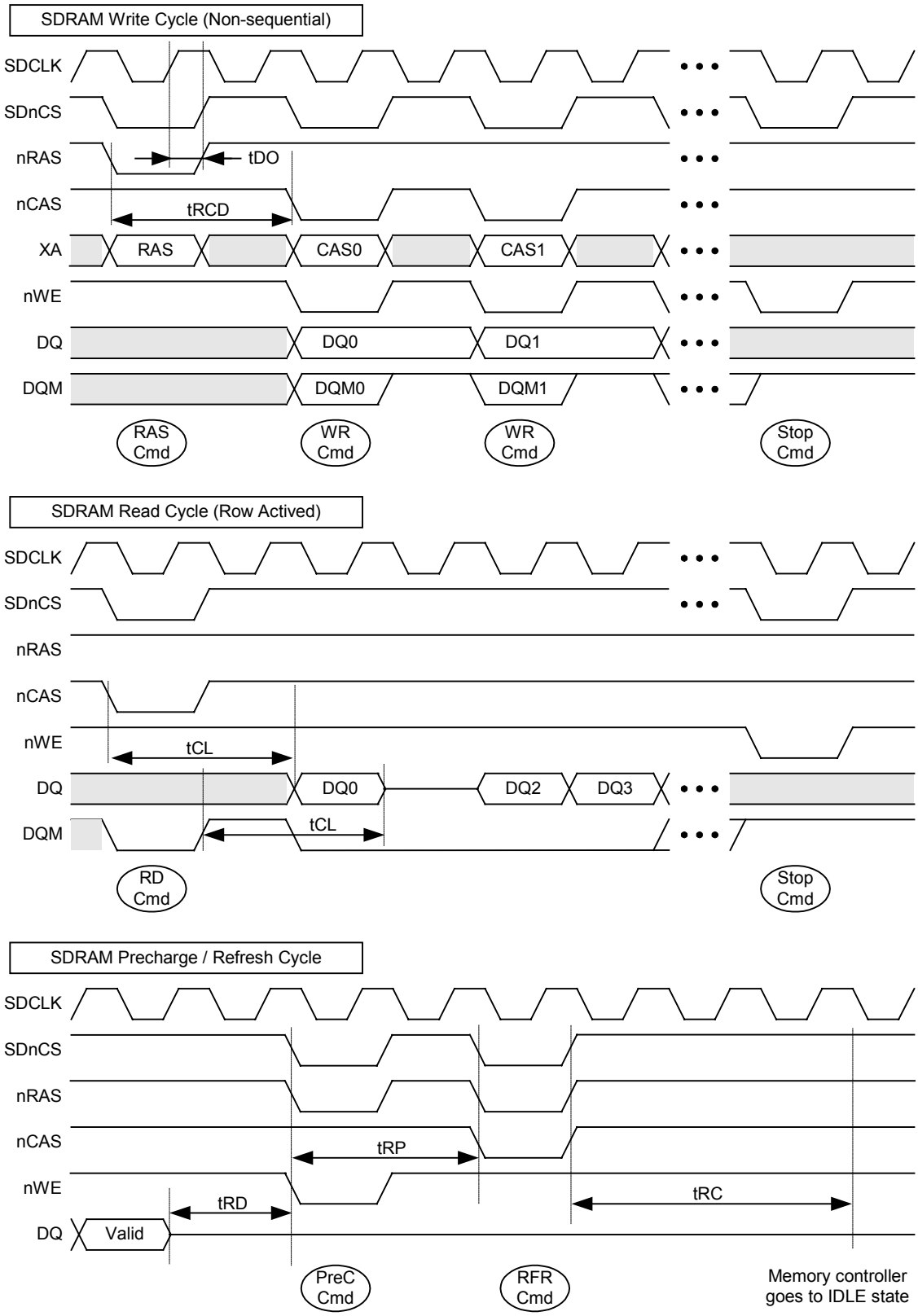


Figure 13.2 SDRAM Cycle Diagram

13.3 Miscellaneous Configuration

In this register, there is various special flag for TCC720 system.

One of them is for supporting boot PROM. In initialization, the lower address space (0x00000000 ~ 0x0FFFFFFF) is mapped to internal or external boot ROM but after initialization, a kind of RAM must be mapped to these space as the system program including interrupt vector table is located in this area. To satisfy this requirement, TCC720 provide RM flag.

BM flag is used to select the boot procedure between the 7 kinds of them. Refer to chapter of boot mode for details. BM flag is determined at the rising edge of the nRESET pin, and contains the state of GPIO_A[10:8] pin.

BW flag is used to know the initial system bus width configuration. This flag is read-only, and contains the state of GPIO_A[5:4] pin at the rising edge of nRESET pin. So user can control the bus width by pulling up or down the GPIO_A[5:4] pin.

Miscellaneous Configuration Register (MCFG) **0xF0000008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDY	0	0	BW		BM			0		JTEN	SDEN	SDS	IM	GPO	RM

RDY [15]	Type	Bus Width Flag
0	R	The state of READY pin is low.
1		The state of READY pin is high.

*) READY pin is used to extend the access cycle for the external memories, it controls directly the cycle of external memory access by setting the URDY bit of each configuration register or can be used as a flags by polling the state of this bit, especially it can be used as a ready signal of NAND flash.

bw* [12:11]	Type	Bus Width Flag
00, 01	R	The corresponding memory is configured by 32bit data bus.
10		The corresponding memory is configured by 16bit data bus.
11		The corresponding memory is configured by 8bit data bus.

*) bw is calculated by xoring the BW field of MCFG register and BW field of CSCFGn register, that is bw = BW(of MCFG) ^ BW(of CSCFGn). BW(of MCFG) is determined by status of GPIO_A[5:4] at the rising edge of nRESET signal.

BM [10:8]	Type	Boot Mode
000	R	Booting procedure begins at the external memory attached at nCS3
001		Booting for downloading firmware by UART port using XIN as main clock
010		Booting for downloading firmware by UART port using XTIN as main clock
011		Booting from NAND flash without decryption process.
100		Booting from NAND flash with decryption process.
101		Booting from NOR flash with decryption process.
110		Booting from HPI bus interface.
111		Development mode: JTAG and SDRAM is enabled, and the base address of SDRAM is set to 0. The TCC721 is waiting for JTAG connection while toggling the GPIO_A[0] output.

*) Except the case of BM == 0, the booting sequence always starts from the internal boot ROM. Refer to chapter of boot mode for more detailed information about booting procedure.

JTEN [5]	Type	Master of Internal Memory Select
0	R/W	JTAG port is disabled
1		JTAG port is enabled

SDEN [4]	Type	Master of Internal Memory Select
0	R/W	SDRAM controller is disabled
1		SDRAM controller is enabled

SDS [3]	Type	SD_CLK output select
0	R/W	SDRAM Clock is out from SD_CLK pin
1		SD bit is out from SD_CLK pin

GPO [1]	Type	SD_CLK output
0 / 1	R/W	When SDS bit is high, this bit is out through SD_CLK pin

IM [2]	Type	SD_CLK output select
0	R/W	Memory controller automatically into idle state, when there is no memory request during 4 cycle of HCLK. If memory request occur, memory controller can serve that request immediately.
1		Memory controller is always active regardless of request state, unless power down or idle state begins.

RM [0]	Type	Remap Flag
0	R/W	The area 0 (0x00000000 ~ 0x0FFFFFFF) space is mapped to internal / external boot ROM
1		The area 0 space is released from boot ROM

*) If external boot ROM is used, it is considered as default that it is attached to nCS3 chip select pin.

In initialization, RM flag direct that the lower address space is mapped to internal or external boot PROM, as program running, the program contained in the internal or external boot ROM must set the RM flag to 1. After this flag is set to 1, the lower address space is released from boot PROM. This lower address space can be mapped to other memories including SDRAM or Flash by changing the base address of that memories. The RM flag can be restored to 0 by clearing bit [0] of 0xF0000008. The lower address space is remapped to boot ROM. Care must be taken not to illegally change the RM flag.

13.4 External Memory Controller

External memory controller can control external memories such as NAND or NOR type flash memory and ROM, SRAM type memory. These memories are selected by nCS3 ~ nCS0 pins. The cycle parameter for accessing external memory can be configured by internal registers. In case of NAND flash, additional parameters for address, command, data cycles are provided.

External Chip Select n Configuration Register (CSCFGn) 0xF0000010 + n*4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			EPW	BW		MTYPE		CSBASE				JRDY	RDY	0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	AMSK	PSIZE		CLADR			STP			PW			HLD		

*) The reset value means the following configuration for each chip select.

Chip Select 0 : 16bit, SRAM, Base = 0x40000000, tSTP=0, tPW=1, tHLD=1

Chip Select 1 : 32bit, IDE, Base = 0x50000000, not use Ready, tSTP=2, tPW=4, tHLD=2

Chip Select 2 : 32bit, NAND, Base = 0x60000000, AMSK=1, PSIZE=1, cLADR=3, tSTP=2, tPW=4, tHLD=2

Chip Select 3 : 16bit, NOR, Base = 0x70000000, tSTP=2, tPW=4, tHLD=2

*bw [27:26]	Bus Width Select
0, 1	Bus width = 32 bit
2	Bus width = 16 bit
3	Bus width = 8 bit

*) The bw is determined by xoring the BW field of CSCFGn register with the BW field of MCFG register.

MTYPE [25:24]	Type of External Memory
0	NAND type
1	IDE type
2	SMEM_0 type (Byte write is not permitted. Ex : ROM, NOR flash)
3	SMEM_1 type (Byte write is permitted. Ex : SRAM)

CSBASE [23:20	Chip Select n Base Address
M	Indicates the MSB 4bit of nCS[n] area. That is nCS[n] base = M * 0x10000000

URDY [19]	Use Ready
1	Ready / Busy signal monitoring is enabled The memory controller waits until the state of READY pin indicate that its access request has accomplished.

RDY [18]	Ready / Busy Select
0	The selected GPIO pin indicating the READY signal. The memory controller waits until this pin goes to high state.
1	The selected GPIO pin indicating the BUSY signal. The memory controller waits until this pin goes to low state.

AMSK [14]	Address Mask Bit
0	Upper half of data bus is masked to zero.

*) In case of 16bit width NAND flash, the upper half byte must be held low, during address cycles. This bit must be set to zero. But if the system uses multiple NAND flashes by sharing a chip select but separating each data to 16 or 32bit data bus of TCC720, the AMSK must be set to 1, so the address can be fed to each NAND flashes.

PSIZE [13:12]	Page size of NAND Flash
psize	The size of one page for NAND type flash. It represents byte per page calculated by the following equation. 1 Page = 256 * 2^{psize}

CLADR [11:9]	Number of Cycle for Linear Address
N	The number of linear address command cycle for NAND type flash. (N+1) cycle is used for generating linear address command.

STP [8:6]	Number of Cycle for Setup Time (tSH)
N	N cycle is issued between the falling edge of nCS[n] and nOE / nWE.

EPW,PW [5:3]	Number of Cycle for Pulse Width (tPW)
N (= 0~15)	(N+1) cycle is issued between the falling and rising edge of nOE / nWE.

HLD [2:0]	Number of Cycle for Hold Time (tHLD)
N	N cycle is issued between the rising edge of nOE / nWE and nCS[n].

The following figure displays the element cycle diagram for external memories.

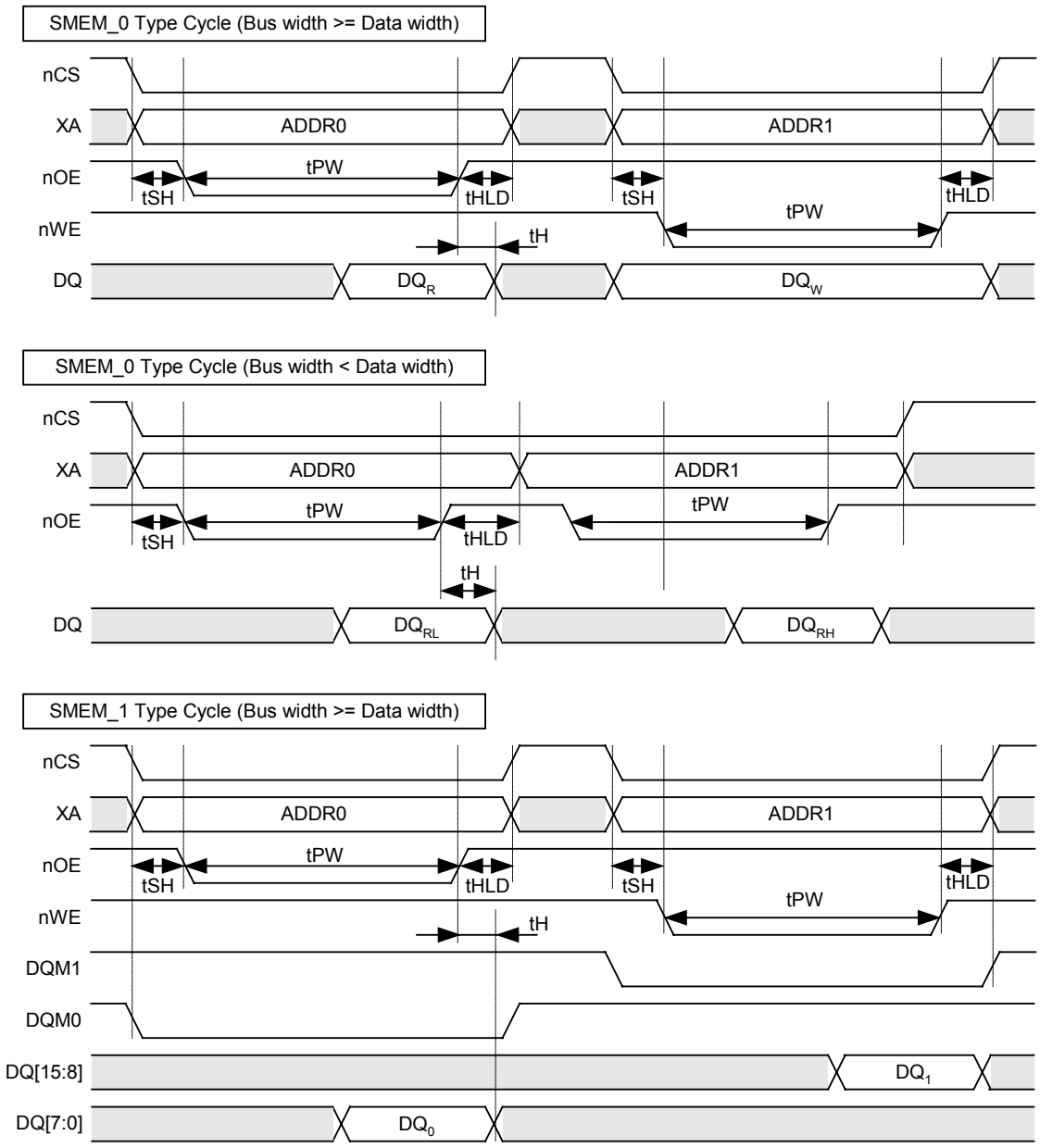


Figure 13.3 Basic Timing Diagram for External Memories

In case of IDE type memories, there are two chip enable signals for it. In TCC720, each enable can be controlled by offset address space. ‘nCS0’ reflects that the offset address range of 0 ~ 0x1F is accessed, ‘nCS1’ reflects that 0x20 ~ 0x3F is accessed. For larger address than 0x3F, if bit5 of address value means which enable signal is activated. (0 to ‘nCS0’, 1 to ‘nCS1’)

In case of NAND flash type memories, there are several sub-registers for accessing.
The followings are these sub-registers. (M is base field of CSCFGn register)

Command Cycle Register (CMD) **0xM0000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMD3								CMD2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD1								CMD0							

*) If bus width of NAND flash is more than 8bit, the CMD1 ~ 3 may be used as command register, otherwise only CMD0 is used as command register. The following values are an example commands for NAND flash of SAMSUNG.

0x00/0x01 : Page Read Command

0x50 : Spare Read Command

0x80 : Page Program Command

0x60 : Block Erase Command

0x70 : Status Read Command

(status read command is generated by reading 0xM0000700 address, not 0xM0000000)

*) Refer to corresponding datasheet of NAND flash chip for detailed command list.

Linear Address Cycle Register (LADDR) **0xM0000004**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LADDR[15:0]															

*) LADDR is used as the linear address for accessing NAND flash data. The number of cycle is determined by CLADR of CSCFGn register. Memory controller assumes that the byte per page is 512, so from the second cycle of address, LADDR[31:9] value is fed to NAND flash.

Block Address Cycle Register (BADDR) **0xM0000008**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BADDR[15:0]															

*) BADDR is used as the block address for accessing NAND flash data with a block unit. The number of cycle is determined by CLADR and PSIZE of CSCFGn register.

Single Address Cycle Register (IADDR) **0xM000000C**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IADDR							

*) When CPU writes to this register, one cycle of address cycle is generated.

Data Register (DATA) **0xM0000010**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3								DATA2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1								DATA0							

*) DATA3~1 may be used as the value of data register, otherwise only DATA0 is used as data register. The number of data cycle is dependent on the bus width of NAND flash and the data size of access cycle.

14.5 Internal Memories

In TCC720, there is 64Kbytes of SRAM for general purposes and 4Kbytes of ROM for system initialization. SRAM area is dedicated to area 3 (0x30000000 ~ 0x3FFFFFFF), and also accessed by area 0 (0x00000000 ~ 0x0FFFFFFF) when there are no devices assigned to area 0. ROM area is dedicated to area E (0xE0000000 ~ 0xEFFFFFFF), and also accessed by area 0 (0x00000000 ~ 0x0FFFFFFF) when RM flag of MCFG register is cleared to 0.

In case of internal ROM, the access speed is not enough to cope with that of system bus (AHB). So when the system bus clock is higher than about 40MHz, the ROM access cycle must be extended by inserting 1 wait cycle. This wait cycle is determined by writing any value to ROM area.

When writing to address the bit 2 of which is 1 (such as 0xE0000004, 0xE000000C, 0xE0000014, …) , the wait cycle is to be inserted from the next ROM access cycle. On the other hand writing to address the bit 2 of which is 0 (such as 0xE0000000, 0xE0000008, 0xE0000010, …), the wait cycle is to be removed from the next ROM access cycle.

The access time of internal SRAM is faster than that of internal ROM, so there is no need to extend access cycle for SRAM.

CHAPTER 14

BOOTING PROCEDURE

14 BOOTING PROCEDURE

In the TCC720, there is a internal boot ROM for system initialization process. It contains the fundamental routines for system initialization or firmware upgrading through various interface such as UART, HPI(Host Port Interface) BUS.

There are 8 modes for booting procedure. It is selected by the state of GPIO_A[10:8] at nRESET going to high. The following table represents the boot mode of TCC720.

Table 14.1 Booting Mode of TCC720

BM	Description
1	F/W download from UART interface with XIN clock source
2	F/W download from UART interface with XTIN clock source
3	NAND boot with non-security NAND must be attached to chip select 2, and use only XD[7:0]
4	NAND boot with security NAND must be attached to chip select 2, and use only XD[7:0]
5	NOR boot with security NOR must be attached to chip select 3, and bus width can be configured by BW
6	HPI boot Processing for HPI bus cycle from HOST processor
7	Development mode JTAG and SDRAM is enabled, and the base address of SDRAM is set to 0. The TCC720 is waiting for JTAG connection while toggling the GPIO_A[0] output.
0	External boot External ROM must be attached to chip select 3

14.1 External Boot

It support an external boot ROM.

When external boot mode, the sequence begins from external ROM that is attached to nCS3.

The bus width of external boot ROM can be determined by state of GPIO_A[5:4] at the rising edge of nRESET pin. If GPIO_A[5:4] == 0, the bus width is 16bit, if GPIO_A[5:4] == 1, it is 8bit, otherwise, it is 32bit. (Refer to the chapter of memory controller for more details)

14.2 UART Boot

For the flexibility and safety of code transfer, there are 2 modes in TCC720 for selecting UART clock. One is using XIN clock divided by 8, the other is using XTIN clock.
This can be selected by setting GPIO_A[10:8] appropriately as described in table 14.1.

Normally, the frequency of UART need to be about 3.6864MHz. But that of XIN may be variable according to the applications. To compromise the difference of clock between TCC720 and host (most case it is a desktop PC), the code is transferred by 1 bit per 1 byte transmission. That is, although the UART transmission is accomplished by 1 byte unit, TCC720 take it as 1 bit transfer, so if the received character is not 0xFF it is regarded as 0, and if 0xFF is received, it is regarded as 1. So, to receive 32bit value, a host must transmit 32 bytes with MSB first order.
Figure 14.1 illustrates the transmission of one 32bit value.

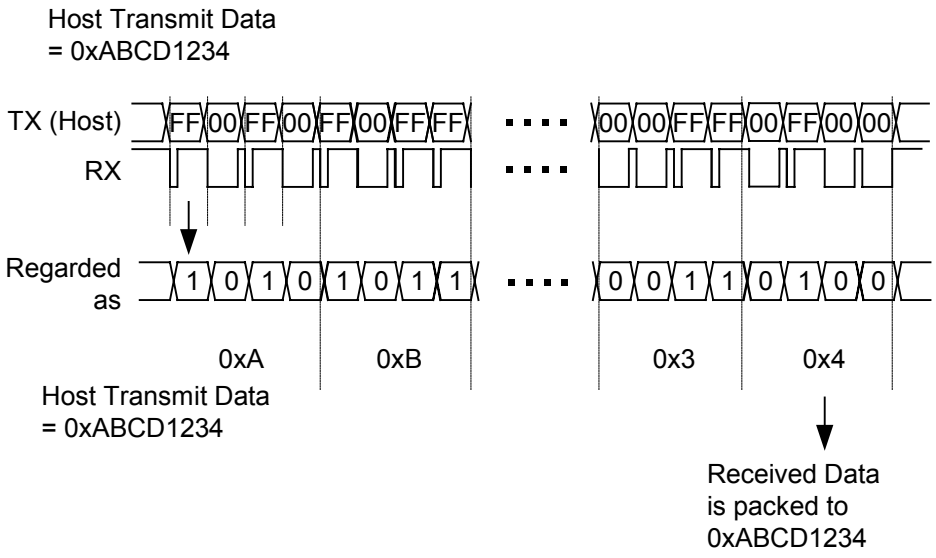


Figure 14.1 The waveform of UART transmission

Because of TCC720 always regard none 0xFF data as ‘0’, it is more robust to set the baud rate of a host UART faster than that of TCC720. The baud rate of a host UART must be as fast as that the duration of the start bit is shorter than that of TCC720 and longer than the period of UART clock (XIN/8 or XTIN). But between each transfer cycles, it is recommended to make sufficient delay times for TCC720 to receive each data correctly.

The procedure of boot code transmission is like as follows. Remember each bit is transferred by

1 byte unit. (0xFF or others)

i) TCC720 enables UART as 9600 baud, none parity bit, 1 stop bit, and 7 data bits.

ii) It receives initial code size of 16bit.

iii) Receive a code of 32bit with order of MSB first and then 1 bit even parity information.

iv) If parity check is succeeded, TCC720 transfer an acknowledgement of 0xFF, or it transfers 0x00, so a host must check if the transfer is succeeded or not.

v) After all of codes are transferred TCC720 branches to address 0x00000000.

This procedure is illustrated in figure 14.2.

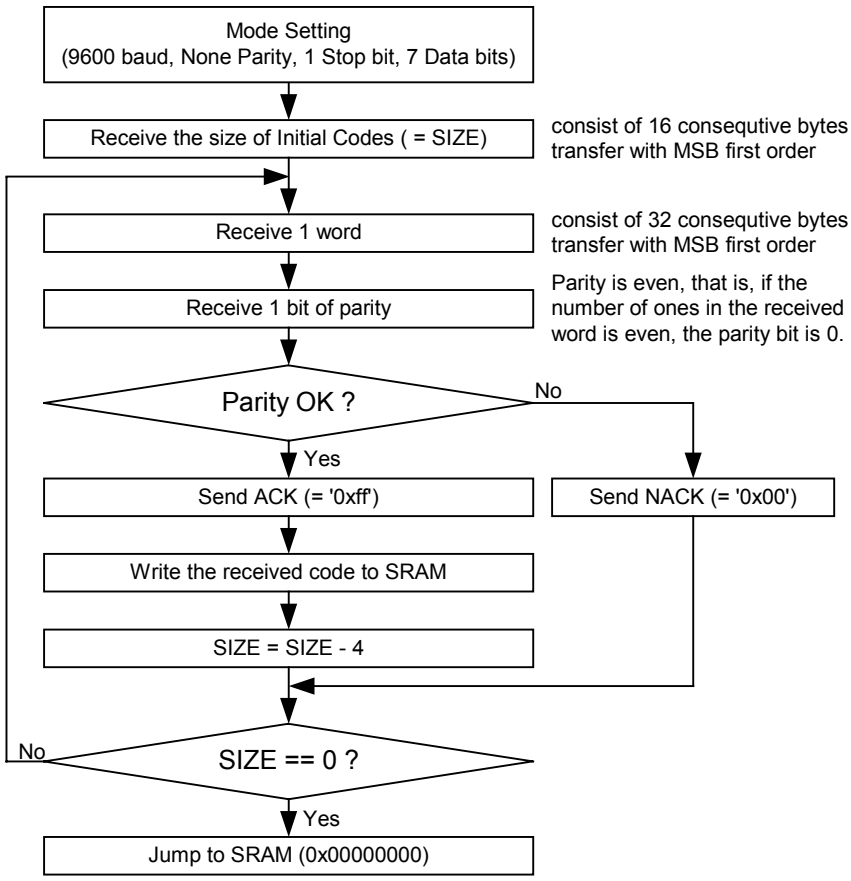


Figure 14.2 UART booting procedure

14.3 NAND Boot

There are 2 modes in TCC720 for booting from NAND flash. One is booting from NAND flash containing a pure F/W code, the other is booting from one containing an encrypted F/W code. This can be selected by setting GPIO_A[10:8] appropriately as described in table 14.1.

The NAND flash is considered to be connected with nCS2, and the bus width is 8 bit regardless of bus width configuration through GPIO_A[5:4].

The supported NAND flash types are as follows.

Table 14.2 Supported NAND flash types

Size (bytes)	Size of Page (bytes)	Number of Page	CADR*	Device ID
1M	128	4K	3	6E
2M	256	4K	3	EA / 64
4M	256	8K	3	E3 / E5
8M	512	8K	3	E6
16M	512	16K	3	73
32M	512	32K	3	75
64M	512	64K	4	76
128M	512	128K	4	79
256M	2048	64K	5	AA / DA

At first, TCC720 checks if the second byte of each spare area is ‘0xC4’ or not starting from the last page to first page. It considers the page of containing ‘0xC4’ at the second byte in that spare area as the start page of containing the initialization codes, so it copies those codes from NAND to internal SRAM. The amount of codes to be transferred is the size of page – 4. The last 4 bytes mean the start number of page which containing the main F/W codes. (TCC720 considers all memories as little endian. So the byte located first means least significant byte in 32bit number, and so on.)

Regardless of encryption option, this initialization codes are not encrypted, so there is no need of decryption and TCC720 directly jump to the code just copied to internal SRAM(0x00000000). At this point, the register R0 contains the number of start page that contains the main F/W codes, and R5 contains page size. If you want change these value, modify these registers

before returning from the initialization code.

The initialization code must be encapsulated by the entrance command of 'STR LR, [SP - #4]!', and the exit command of 'LDR PC, [SP], #4'. This code may contain various routines such as memory configuration or user customized booting code itself.

After the initialization code finishes, and the code returns by the above exit code, the main F/W copy code begins copy from the start page contained in R0 register. TCC720 copies the size of page – 8 bytes of codes per every page to the area starting from the address of '0x00000000'. You must configure this area appropriately before returning from the initialization code. In case of encrypted F/W code, it is decrypted and then copied to as like as in non-encrypted case.

The next page number is consisted of 4 bytes and located at (the size of page – 7) ~ (the size of page – 4) in current page of data to be copied. The last 4 bytes in page of data are reserved for future use.

If the next page number is equal to '0xFFFFFFFF', that page is the last page containing F/W code. TCC720 copy F/W code until this number is acquired.
Figure 14.3 illustrates the organization of NAND flash.

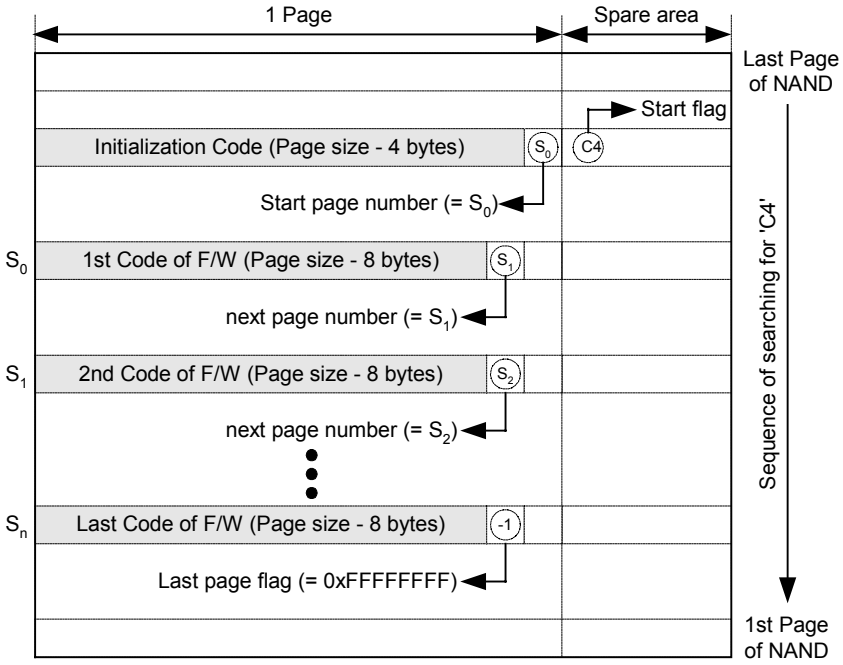


Figure 14.3 The boot code structure in NAND flash

The procedure of booting from NAND flash is displayed in figure 14.4

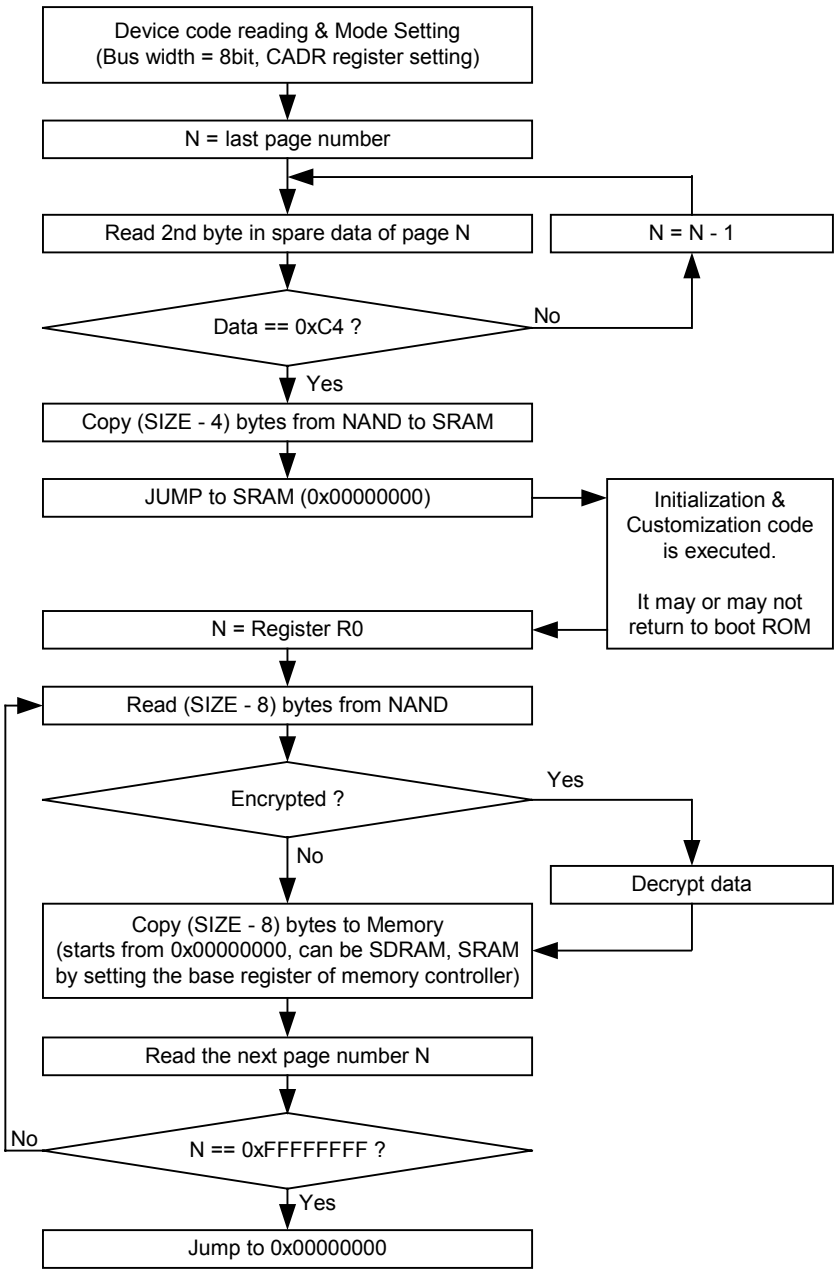


Figure 14.4 NAND boot procedure

14.4 NOR Boot with Security

There are two modes for booting from NOR flash. One is booting from external ROM that is commonly used for various applications, and the other is booting from encrypted external ROM that can be used to hide one's code from the non-authorized user.
It can be selected by setting GPIO_A[10:8] appropriately as described in table 14.1.

In both cases, the NOR flash must be connected via nCS3 signal.
In case of non-encrypted mode, the F/W code in external flash is directly fetched to TCC720 without any other intermediate processing, but in case of encrypted mode, the F/W code in external flash is stored in encrypted value, so boot ROM must decrypt it ahead and then copy these codes to the other random accessible memories.

Figures 14.5 illustrate the allocation map of encrypted F/W code in NOR flash.

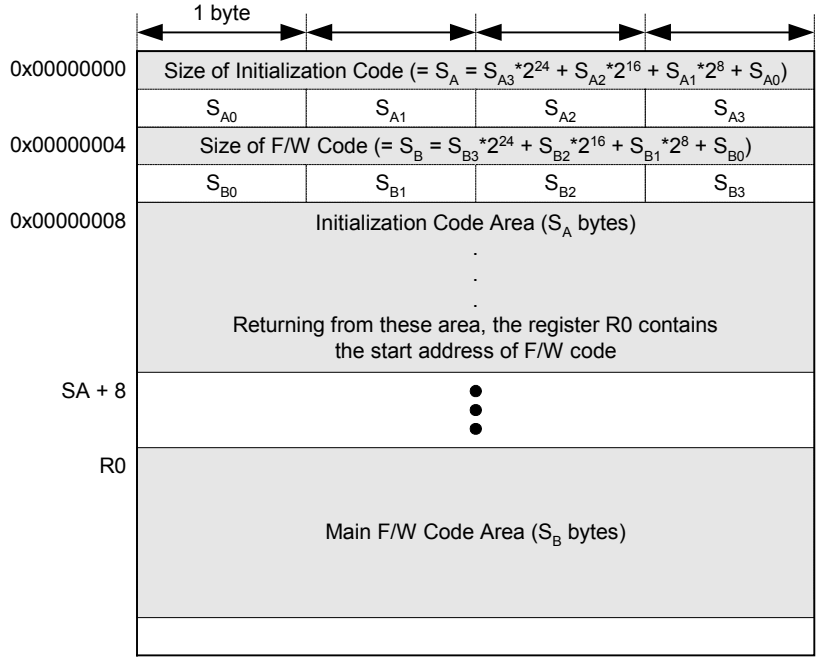


Figure 14.5 Allocation of encrypted F/W code

14.5 Development mode

To ease the effort for starting development withTCC720, TCC720 provides development mode in booting. In this mode, JTAG interface is enabled and set cache & protection unit of TCC720 appropriately.

The table 14.4 describes the region setting in this mode.

Table 14.4 Region Settings in Development Mode

Region #	Start	End	I Cache	D Cache	Buffer	Protection
0	0x00000000	0xFFFFFFFF	Enabled	Enabled	Enabled	Full Access
1	0x20000000	0x3FFFFFFF	Enabled	Disabled	Enabled	Full Access
2	0x40000000	0x4FFFFFFF	Enabled	Disabled	Enabled	Full Access
3	0x50000000	0x5FFFFFFF	Enabled	Disabled	Enabled	Full Access
4	0x60000000	0x6FFFFFFF	Enabled	Disabled	Enabled	Full Access
5	0x70000000	0x7FFFFFFF	Enabled	Disabled	Enabled	Full Access
6	0x80000000	0xFFFFFFFF	Enabled	Disabled	Enabled	Full Access
7	0x3000F000	0x3000FFFF	Enabled	Disabled	Enabled	Full Access

*) The region of higher number has higher priority than that of the lower regions. That is, region 7 has highest priority and region 0 has lowest priority.

After region setting finishes, it goes into a infinite loop toggling GPIO_A[0].

CHAPTER 15

JTAG DEBUG INTERFACE

15 JTAG DEBUG INTERFACE

The TCC720 has the ARM940T core as main controller, and JTAG interface for developing the application programs. It can be connected with Multi-ICE of ARM or other third party's in-circuit emulator supporting for ARM940T core.

With the use of in-circuit emulator, the user can easily develop the program for his or her own system. It provides hardware breakpoints, internal register monitoring, memory dump, etc. Refer to user's manual of in-circuit emulator for more detail functions of it.

CHAPTER 16

PACKAGE DEMENSION

16 PACKAGE DEMENSION

16.1 128-Pin TQFP

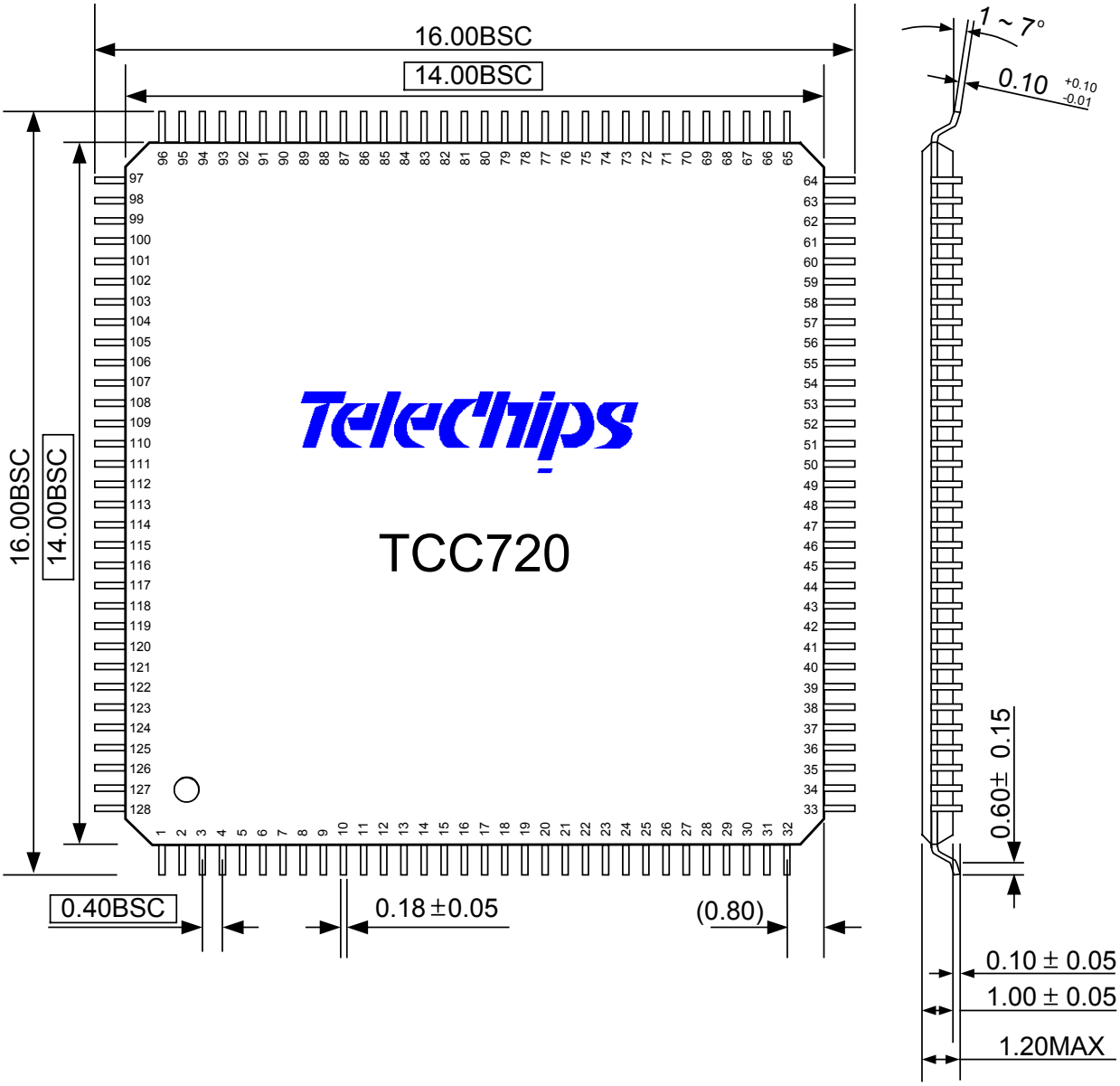


Figure 16.1 Package Dimension of 128-TQFP-1414