VERSA 1000: 64kB Embedded ISP/IAP FLASH 1kB RAM, 40 MHz, 8-Bit MCU

Datasheet Rev 1.6

Overview

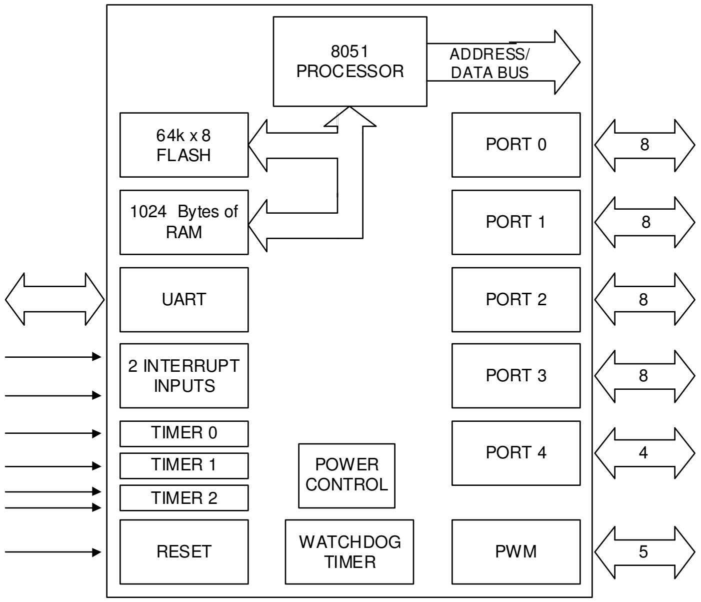
The VRS1000 is an 8-bit microcontroller with 64kB Flash and 1kB of RAM. It includes an In-System / In-Application Programmable (ISP/IAP) function and is based on the architecture of the standard 80C51 microcontroller.

Each device contains a small ISP program residing in the upper section of the Flash memory that allows In-System Programming of the Flash memory from the processor itself or through the UART interface.

The VRS1000's features and powerful instruction set make it a versatile and cost effective controller for applications that require non volatile data storage or that require the ability to perform its own Firmware code update.

The on-chip Flash memory can be programmed via a serial interface using the ISP feature. Programming of the VRS1000 is supported by programmers available from Goal Semiconductor or other 3rd party commercial programmers.
The VRS1000 is available in PLCC-44 or QFP-44 packages in industrial temperature range.

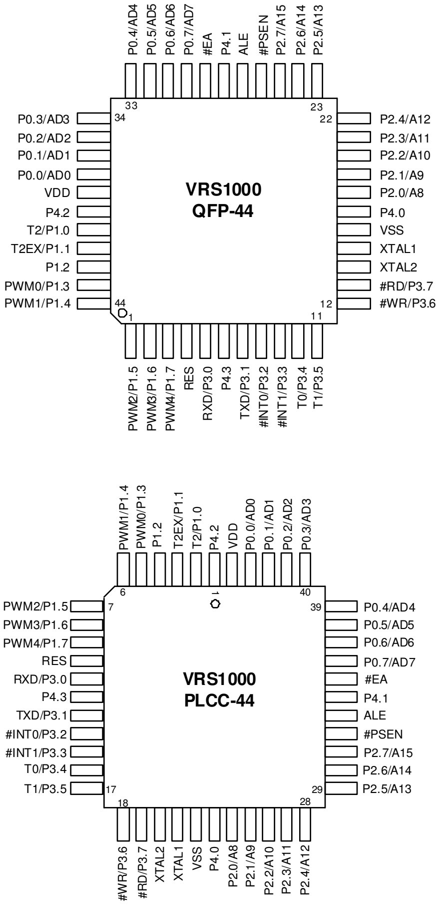
FIGURE 1: VRS1000 FUNCTIONAL DIAGRAM



Feature Set

- General 80C51/80C52 family compatible
- 64kB byte on-chip Flash memory
- In-System Programming (ISP) capability of Flash
- Program voltage: 5V
- 1024 bytes on chip data RAM
- Four 8-bit I/Os + one 4-bit I/O
- 5-Channel PWM on P1.3 to P1.7
- Full duplex serial port (UART)
- Three 16-bit Timers/Counters
- Watch Dog Timer
- Bit operation instruction
- 8-bit Unsigned Multiply and Division instructions
- BCD arithmetic
- Direct and Indirect Addressing
- Two Levels of Interrupt Priority and Nested Interrupts
- Power saving modes:
- Code protection function
- Low EMI (inhibit ALE)
- Industrial Temperature range (-40ºC to +85ºC)
- 5V version available

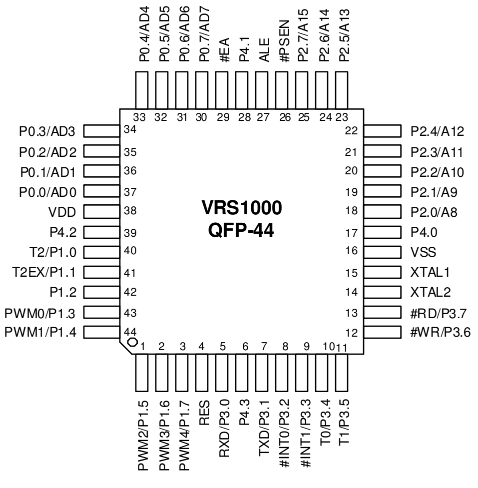
FIGURE 2: VRS1000 QFP-44 AND PLCC-44 PIN OUT DIAGRAMS

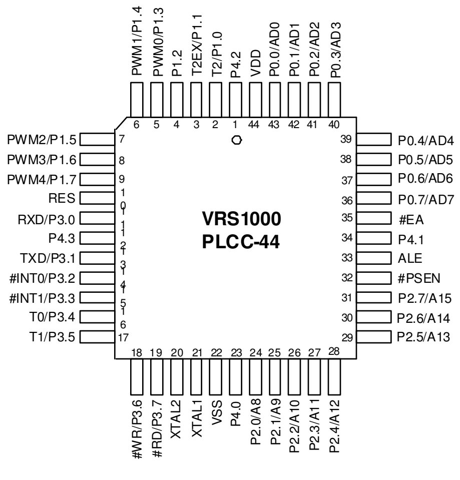


Pin Descriptions for QFP-44/PLCC-44

TABLE 1: PIN DESCRIPTIONS FOR QFP-44/PLCC-44

QFP - 44	PLCC - 44	Name	I/O	Function
1	7	PWM2	O	PWM Channel 2
		P1.5	I/O	Bit 5 of Port 1
2	8	PWM3	O	PWM Channel 3
		P1.6	I/O	Bit 6 of Port 1
3	9	PWM4	O	PWM Channel 4
		P1.7	I/O	Bit 7 of Port 1
4	10	RES	I	Reset
5	11	RXD	I	Receive Data
		P3.0	I/O	Bit 0 of Port 3
6	12	P4.3	I/O	Bit 3 of Port 4
7	13	TXD	O	Transmit Data &
		P3.1	I/O	Bit 1 of Port 3
8	14	#INT0	I	External Interrupt 0
		P3.2	I/O	Bit 2 of Port 3
9	15	#INT1	I	External Interrupt 1
		P3.3	I/O	Bit 3 of Port 3
10	16	T0	I	Timer 0
		P3.4	I/O	Bit 4 of Port 3
11	17	T1	I	Timer 1 & 3
		P3.5	I/O	Bit 5 of Port
12	18	#WR	O	Ext. Memory Write
		P3.6	I/O	Bit 6 of Port 3
13	19	#RD	O	Ext. Memory Read
		P3.7	I/O	Bit 7 of Port 3
14	20	XTAL2	O	Oscillator/Crystal Output
15	21	XTAL1	I	Oscillator/Crystal In
16	22	VSS	-	Ground
17	23	P4.0	I/O	Bit 0 of Port 4
18	24	P2.0	I/O	Bit 0 of Port 2
		A8	O	Bit 8 of External Memory Address
19	25	P2.1	I/O	Bit 1 of Port 2
		A9	O	Bit 9 of External Memory Address
20	26	P2.2	I/O	Bit 2 of Port 2
		A10	O	Bit 10 of External Memory Address
21	27	P2.3	I/O	Bit 3 of Port 2 &
		A11	O	Bit 11 of External Memory Address
22	28	P2.4	I/O	Bit 4 of Port 2
		A12	O	Bit 12 of External Memory Address
23	29	P2.5	I/O	Bit 5 of Port 2
		A13	O	Bit 13 of External Memory Address
24	30	P2.6	I/O	Bit 6 of Port 2
		A14	O	Bit 14 of External Memory Address
25	31	P2.7	I/O	Bit 7 of Port 2
		A15	O	Bit 15 of External Memory Address
26	32	#PSEN	O	Program Store Enable
27	33	ALE	O	Address Latch Enable
28	34	P4.1	I/O	Bit 1 of Port 4
29	35	#EA	I	External Access
30	36	P0.7	I/O	Bit 7 Of Port 0
		AD7	I/O	Data/Address Bit 7 of External Memory
31	37	P0.6	I/O	Bit 6 of Port 0
		AD6	I/O	Data/Address Bit 6 of External Memory
32	38	P0.5	I/O	Bit 5 of Port 0
		AD5	I/O	Data/Address Bit 5 of External Memory
33	39	P0.4	I/O	Bit 4 of Port 0
		AD4	I/O	Data/Address Bit 4 of External Memory
34	40	P0.3	I/O	Bit 3 Of Port 0
		AD3	I/O	Data/Address Bit 3 of External Memory
35	41	P0.2	I/O	Bit 2 of Port 0
		AD2	I/O	Data/Address Bit 2 of External Memory
36	42	P0. 1	I/O	Bit 1 of Port 0 & Data
		AD1	I/O	Address Bit 1 of External Memory
37	43	P0.0	I/O	Bit 0 Of Port 0 & Data
		AD0	I/O	Address Bit 0 of External Memory
38	44	VDD	-	VCC
39	1	P4.2	I/O	Bit 2 of Port 4
40	2	T2	I	Timer 2 Clock Out
		P1.0	I/O	Bit 0 of Port 1
41	3	T2EX	I	Timer 2 Control
		P1.1	I/O	Bit 1 of Port 1
42	4	P1.2	I/O	Bit 2 of Port 1
43	5	PWM0	O	PWM Channel 0
		P1.3	I/O	Bit 3 of Port 1
44	6	PWM1	O	PWM Channel 1
		P1.4	I/O	Bit 4 of Port 1





Instruction Set

The following table describes the instruction set of the VRS1000. The instructions are binary code compatible and perform the same functions as the industry standard 8051 ones.

TABLE 2: LEGEND FOR INSTRUCTION SET TABLE

Symbol	Function
A	Accumulator
Rn	Register R0-R7
Direct	Internal register address
@Ri	Internal register pointed to by R0 or R1 (except MOVX)
rel	Two's complement offset byte
bit	Direct bit address
#data	8-bit constant
#data 16	16-bit constant
addr 16	16-bit destination address
addr 11	11-bit destination address

TABLE 3: VRS570/VRS580 INSTRUCTION SET

Mnemonic	Description	Size (bytes)	Instr. Cycles
Arithmetic instructions			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	1
ADD A, @Ri	Add data memory to A	1	1
ADD A, #data	Add immediate to A	2	1
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	1
ADDC A, @Ri	Add data memory to A with carry	1	1
ADDC A, #data	Add immediate to A with carry	2	1
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	1
SUBB A, @Ri	Subtract data mem from A with borrow	1	1
SUBB A, #data	Subtract immediate from A with borrow	2	1
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	1
INC @Ri	Increment data memory	1	1
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	1
DEC @Ri	Decrement data memory	1	1
INC DPTR	Increment data pointer	1	2
MUL AB	Multiply A by B	1	4
DIV AB	Divide A by B	1	4
DA A	Decimal adjust A	1	1
Logical Instructions			
ANL A, Rn	AND register to A	1	1
ANL A, direct	AND direct byte to A	2	1
ANL A, @Ri	AND data memory to A	1	1
ANL A, #data	AND immediate to A	2	1
ANL direct, A	AND A to direct byte	2	1
ANL direct, #data	AND immediate data to direct byte	3	2
ORL A, Rn	OR register to A	1	1
ORL A, direct	OR direct byte to A	2	1
ORL A, @Ri	OR data memory to A	1	1
ORL A, #data	OR immediate to A	2	1
ORL direct, A	OR A to direct byte	2	1
ORL direct, #data	OR immediate data to direct byte	3	2
XRL A, Rn	Exclusive-OR register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	1
XRL A, @Ri	Exclusive-OR data memory to A	1	1
XRL A, #data	Exclusive-OR immediate to A	2	1
XRL direct, A	Exclusive-OR A to direct byte	2	1
XRL direct, #data	Exclusive-OR immediate to direct byte	3	2
CLR A	Clear A	1	1
CPL A	Compliment A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
Boolean Instruction			
CLR C	Clear Carry bit	1	1
CLR bit	Clear bit	2	1
SETB C	Set Carry bit to 1	1	1
SETB bit	Set bit to 1	2	1
CPL C	Complement Carry bit	1	1
CPL bit	Complement bit	2	1
ANL C,bit	Logical AND between Carry and bit	2	2
ANL C,#bit	Logical AND between Carry and not bit	2	2
ORL C,bit	Logical ORL between Carry and bit	2	2
ORL C,#bit	Logical ORL between Carry and not bit	2	2
MOV C,bit	Copy bit value into Carry	2	1
MOV bit,C	Copy Carry value into Bit	2	2
Data Transfer Instructions			
MOV A, Rn	Move register to A	1	1
MOV A, direct	Move direct byte to A	2	1
MOV A, @Ri	Move data memory to A	1	1
MOV A, #data	Move immediate to A	2	1
MOV Rn, A	Move A to register	1	1
MOV Rn, direct	Move direct byte to register	2	2
MOV Rn, #data	Move immediate to register	2	1
MOV direct, A	Move A to direct byte	2	1
MOV direct, Rn	Move register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	2
MOV direct, @Ri	Move data memory to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	2
MOV @Ri, A	Move A to data memory	1	1
MOV @Ri, direct	Move direct byte to data memory	2	2
MOV @Ri, #data	Move immediate to data memory	2	1
MOV DPTR, #data	Move immediate to data pointer	3	2
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	2
MOVC A, @A+PC	Move code byte relative PC to A	1	2
MOVX A, @Ri	Move external data (A8) to A	1	2
MOVX A, @DPTR	Move external data (A16) to A	1	2
MOVX @Ri, A	Move A to external data (A8)	1	2
MOVX @DPTR, A	Move A to external data (A16)	1	2
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange A and register	1	1
XCH A, direct	Exchange A and direct byte	2	1
XCH A, @Ri	Exchange A and data memory	1	1
XCHD A, @Ri	Exchange A and data memory nibble	1	1
Branching Instructions			
ACALL addr 11	Absolute call to subroutine	2	2
LCALL addr 16	Long call to subroutine	3	2
RET	Return from subroutine	1	2
RETI	Return from interrupt	1	2
AJMP addr 11	Absolute jump unconditional	2	2
LJMP addr 16	Long jump unconditional	3	2
SJMP rel	Short jump (relative address)	2	2
JC rel	Jump on carry = 1	2	2
JNC rel	Jump on carry = 0	2	2
JB bit, rel	Jump on direct bit = 1	3	2
JNB bit, rel	Jump on direct bit = 0	3	2
JBC bit, rel	Jump on direct bit = 1 and clear	3	2
JMP @A+DPTR	Jump indirect relative DPTR	1	2
JZ rel	Jump on accumulator = 0	2	2
JNZ rel	Jump on accumulator 1= 0	2	2
CJNE A, direct, rel	Compare A, direct JNE relative	3	2
CJNE A, #d, rel	Compare A, immediate JNE relative	3	2
CJNE Rn, #d, rel	Compare reg, immediate JNE relative	3	2
CJNE @Ri, #d, rel	Compare ind, immediate JNE relative	3	2
DJNZ Rn, rel	Decrement register, JNZ relative	2	2
DJNZ direct, rel	Decrement direct byte, JNZ relative	3	2
Miscellaneous Instruction			
NOP	No operation	1	1

Rn:	Any of the register R0 to R7
@Ri:	Indirect addressing using Register R0 or R1
#data:	immediate Data provided with Instruction
#data16:	Immediate data included with instruction
bit:	address at the bit level
rel:	relative address to Program counter from +127 to –128
Addr11:	11-bit address range
Addr16:	16-bit address range
#d:	Immediate Data supplied with instruction

Special Function Registers (SFR)

Addresses 80h to FFh of the SFR address space can be accessed in direct addressing mode only. The following table lists the VRS1000 Special Function Registers.

TABLE 4: SPECIAL FUNCTION REGISTERS (SFR)

SFR Register	SFR Adrs	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Value
P0	80h	-	-	-	-	-	-	-	-	
SP	81h	-	-	-	-	-	-	-	-	
DPL	82h	-	-	-	-	-	-	-	-	
DPH	83h	-	-	-	-	-	-	-	-	
RCON	85h	-	-	-	-	-	-	RAMS1	RAMS0	******00b
DBANK	86h	BSE	-	-	-	BS3	BS2	BS1	BS0	0***0001b
PCON	87h	SMOD	-	-	-	GF1	GF0	PDOWN	IDLE	00000000b
TCON	88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000b
TMOD	89h	GATE1	C/T1	M1.1	M0.1	GATE0	C/T0	M1.0	M0.0	00000000b
TL0	8Ah	-	-	-	-	-	-	-	-	
TL1	8Bh	-	-	-	-	-	-	-	-	
TH0	8Ch	-	-	-	-	-	-	-	-	
TH1	8Dh	-	-	-	-	-	-	-	-	
P1	90h	-	-	-	-	-	-	-	-	
SCON	98h	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00000000b
SBUF	99h	-	-	-	-	-	-	-	-	
PWME	9Bh	PWM4E	PWM3E	PWM2E	PWM1E	PWM0E	-	-	-	00000***b
WDTC	9Fh	WDTE	-	CLEAR	-	-	PS2	PS1	PS0	0*0**000b
P2	A0h	-	-	-	-	-	-	-	-	
PWMC	A3h	-	-	-	-	-	-	PDCK1	PDCK0	******00b
PWMD0	A4h	PWMD0.4	PWMD0.3	PWMD0.2	PWMD0.1	PWMD0.0	NP0.2	NP0.1	NP0.0	00000000b
PWMD1	A5h	PWMD1.4	PWMD1.3	PWMD1.2	PWMD1.1	PWMD1.0	NP1.2	NP1.1	NP1.0	00000000b
PWMD2	A6h	PWMD2.4	PWMD2.3	PWMD2.2	PWMD2.1	PWMD2.0	NP2.2	NP2.1	NP2.0	00000000b
PWMD3	A7h	PWMD3.4	PWMD3.3	PWMD3.2	PWMD3.1	PWMD3.0	NP3.2	NP3.1	NP3.0	00000000b
IE	A8h	EA	-	ET2	ES	ET1	EX1	ET0	EX0	00000000b
PWMD4	ACh	PWMD4.4	PWMD4.3	PWMD4.2	PWMD4.1	PWMD4.0	NP4.2	NP4.1	NP4.0	00000000b
P3	B0h	-	-	-	-	-	-	-	-	
IP	B8h	-	-	PT2	PS	PT1	PX1	PT0	PX0	00000000b
SYSCON	BFh	WDR	-	-	-	-	IAPE	XRAME	ALEI	0****010b
T2CON	C8h	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00000000b
RCAP2L	CAh	-	-	-	-	-	-	-	-	00000000b
RCAP2H	CBh	-	-	-	-	-	-	-	-	
TL2	CCh	-	-	-	-	-	-	-	-	
TH2	CDh									
PSW	D0h	CY	AC	F0	RS1	RS0	OV	-	P	00000000b
P4	D8h	-	-	-	-	P4.3	P4.2	P4.1	P4.0	****1111b
ACC	E0h	-	-	-	-	-	-	-	-	
B	F0h	-	-	-	-	-	-	-	-	
IAPFADHI	F4h	FA15	FA14	FA13	FA12	FA11	FA10	FA9	FA8	00000000b
IAPFADLO	F5h	FA7	FA6	FA5	FA4	FA3	FA2	FA1	FA0	00000000b
IAPFDATA	F6h	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0	00000000b
IAPFCTRL	F7h	IAPSTART						IAPFCT1	IAPFCT0	0*****00b

VRS1000 Program Memory

The VRS1000 includes 64K of on-chip Flash memory that can be used as program memory but it can also be used as non-volatile data storage memory using the In-Application programming feature (IAP).

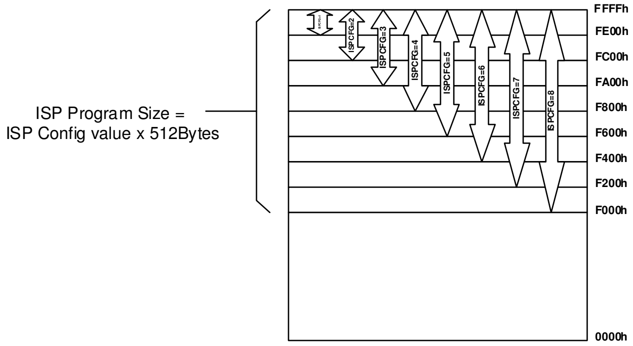
ISP Program Memory Zone

The upper portion of the VRS1000 Flash memory can be reserved to hold an ISP boot program.

This boot program can be used to perform the Flash memory programming using the serial interface or any other method by making use of the In-Application Programming (IAP) feature of the VRS1000 which permits the processor to load the program from an external device or system and program it into the Flash memory (See the ***VRS1000 IAP feature*** section)

The size of the memory block reserved for the ISP boot program (when activated) is adjustable from 512 Bytes up to 4k Bytes in increment of 512 Bytes.

FIGURE3: VRS1000-ISP PROGRAM SIZE VS ISP CONFIG. VALUE



Programming the ISP Boot Program

The ISP boot program must be programmed into the device using a parallel programmer such as the low cost VERSAMCU-PPR or a commercial parallel programmer supporting the VRS1000. The Flash memory reserved for the ISP program is defined in the parallel programmer software at the moment the device is programmed.

FIGURE 4: VERSAMCU-PPR PROGRAM INTERFACE WINDOW

Select Device: VRS1000

ISP Page Config: 0

Please put currect Parts or Device into Programmer Socket before programming.
To Avoid programmer damage,be sure that Dvice selectd and Device in

When programming the ISP boot program into the VRS1000, it is recommended to activate the "lock bit" option to protect the ISP flash memory zone from being inadvertently erased when the Flash Erase operations are performed (See the ***VRS1000 IAP feature*** section for details on IAP functions)

ISP Program Start Conditions

Setting the ISP page configuration to a value other than 0 will make the Processor jump to the base address of the ISP boot code when a hardware reset is performed, provided that the value FFh is present at program address 0000h.

It is also possible to call the ISP program by doing a program jump instruction to its start address

When the ISP page configuration is set to 0 at the moment the device is programmed using a parallel programmer, the ISP boot feature will be disabled.

Pre-Programmed VRS1000 ISP Program

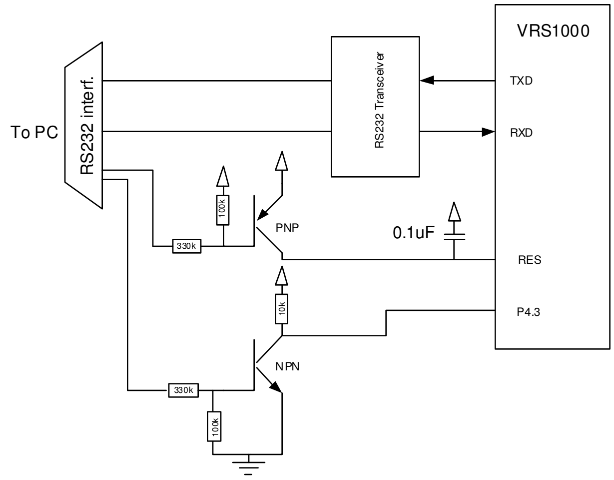
For your convenience, Goal Semiconductor Inc. has developed an ISP boot program for the VRS1000 that allows programming the device's Flash memory using the device's UART0 Serial Port and communicates with the GoalTender VRS1000 Program that runs under Windows™ operating system.

This ISP boot program allows you to program the VRS1000 on the final application PCB using the device's UART interface. The hardware interface to perform the VRS1000 Flash memory programming using the ISP boot program is shown below:

If you want to use the Goal Semiconductor parallel programmer to program the ISP onto the VRS1000, please specify at the moment you place your order.

For more information regarding features and use of the ISP Program developed by Goal Semiconductor Inc, please consult the "VRS1000 ISP Getting Started Guide".

FIGURE 5: VRS1000 INTERFACE FOR IN-SYSTEM PROGRAMMING



VRS1000 IAP feature

The VRS1000 IAP feature refers to the ability for the processor to self-program the Flash memory from within the user program.

Five SFR registers serve to control the IAP operation. The description of these registers is given below.

The System Control Register

By default upon reset, the IAP feature of the VRS1000 is de-activated. The IAPE bit of the SYSCON register is used to enable (and disable) the VRS1000 IAP function as shown below.

TABLE 5: SYSTEM CONTROL REGISTER (SYSCON) – SFR BFH

7	6	5	4	3	2	1	0
WDR	Unused				IAPE	XRAME	ALEI

Bit	Mnemonic	Description
7	WDR	This is the Watch Dog Timer reset bit. It will be set to 1 when the reset signal generated by WDT overflows.
6	Unused	-
5	Unused	-
4	Unused	-
3	Unused	-
2	IAPE	IAP function enable bit
1	XRAME	768 byte on-chip enable bit
0	ALEI	ALE output inhibit bit, which is used to reduce EMI.

IAP Flash Address and Data Registers

The IAPFADHI and IAPADLO registers are used to specify the address at which the IAP function will be performed.

TABLE 6:IAP FLASH ADDRESS HIGH - SFR F4H

7	6	5	4	3	2	1	0
IAPFADHI[15:8]							

TABLE 7:IAP FLASH ADDRESS LOW - SFR F5H

7	6	5	4	3	2	1	0
IAPFADLO[15:8]							

The IAPFDATA SFR register contains the Data byte required to perform the IAP function.

TABLE 8:IAP FLASH DATA REGISTER - SFR F6H

7	6	5	4	3	2	1	0
IAPFDATA[7:0]							

IAP Flash Control Register

The VRS1000 IAP function operation is controlled by the IAP Flash Control register, IAPFCTRL.

Setting the IAPSTART bit to 1, start the execution of the IAP command specified by the IAPFCT[1:0] bit of the IAP Flash Control register.

TABLE 9:IAP FLASH CONTROL REGISTER - SFR F7H

7	6	5	4	3	2	1	0
IAPFCTRL[15:8]							

Bit	Mnemonic	Description
7	IAPSTART	IAP Selected operation Start sequence
6	Unused	-
5	Unused	-
4	Unused	-
3	Unused	-
2	Unused	-
1	IAPFCT[1:0]	Flash Memory IAP Function
0		

The IAP sub-system handles four different functions. The IAP function to perform is defined by the IAPFCT bits value as shown below:

TABLE 10:IAP FUNCTIONS

IAPFCT[1:0] Bits value	IAP Function
00	Flash Byte Program
01	Flash Erase Protect
10	Flash Page Erase
11	Flash Erase

It is important to note that for security reasons the IAPSTART bit of the IAPFCTRL register is configured as read only by default.

In order to access the IAPSTART bit and to write a 1 into it the following operation sequence must be performed first:

```
MOV    IAPFDATA,#55h
MOV    IAPFDATA,#AAh
MOV    IAPFDATA,#55h
```

Then the IAPSTART bit can be set to 1.

Once the start bit is set to 1, the IAP sub-system will read the content of the IAP Flash Address and Data register and hold the VRS1000 program counter to its current value until the IAP operation is completed. When the IAP operation is complete, the IAPSTART bit is cleared and the program continues its execution.

IAP Byte Program Function

The IAP byte program function is used to program a byte into the specified Flash memory location under the control of the IAP feature. The following program example shows how to do it:

```
IAP_PROG:  MOV        IAPFDATA,#55H           ;Sequence to Enable Writing
           MOV        IAPFDATA,#0AAH          ; the IAPSTART bit
           MOV        IAPFDATA,#55H

           MOV        SYSCON,#04H             ;ENABLE IAP FUNCTION
           MOV        IAPFADHI, FADRSH        ;Set MSB of address  to program
           MOV        IAPFADLO,FADRSL         ;Set LSB of address  to program
           MOV        IAPFDATA,FDATA          ;Set Data to Program
           MOV        IAPFCTRL,#80H           ;Set the IAP Start bit

           ;**The program Counter will stop until the IAP function is completed
```

IAP Page Erase Function

Using the IAP feature it is possible to perform Page erase of the VRS1000 Flash memory at the exception of the memory area occupied by the ISP boot program. Each page is 512 Bytes in size.

To perform a given flash page erase, the page address is specified by the XY (hex) value written into the IAPFADHI register (The value 00h must be written into the IAPFADLO registers)

If the "Y" portion of the IAPFADHI register represents an even number, the page that will be erased corresponds to the range XY00h to X(Y+1)FFh

If the "Y" portion of the IAPFADHI register represents an odd number, the page that will be erased corresponds to the range X(Y-1)00h to XYFFh

The following example program shows how to erase the page corresponding to the address B000h-CFFFh

```
;** Erase Flash page located at address B000h to CFFFh.

PageErase: MOV        IAPFDATA,#55H           ;Sequence to Enable Writing
           MOV        IAPFDATA,#0AAH          ; the IAPSTART bit
           MOV        IAPFDATA,#55H

           MOV        SYSCON,#04H             ;ENABLE IAP FUNCTION
           MOV        IAPFADHI, #0B0h         ;Set MSB of  Page address to erase
           MOV        IAPFADLO,#00h           ;Set LSB of address  = 00
           MOV        IAPFCTRL,#82H           ;SET THE IAP START BIT
```

IAP Chip Erase Function

The IAP chip erase function will erase the entire flash memory content with the exception of the ISP boot program area. Running this function will also automatically unprotect the flash memory.

IAP Chip Protect Function

The chip protect function when executed makes the chip Flash memory content to read as 00h when an attempt is made to read it.

Program Status Word Register

The register below contains the program state flags. These flags may be read or written to by the user.

TABLE 11: PROGRAM STATUS WORD REGISTER (PSW) - SFR DOH

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	-	P

Bit	Mnemonic	Description
7	CY	Carry Bit
6	AC	Auxiliary Carry Bit from bit 3 to 4.
5	F0	User definer flag
4	RS1	R0-R7 Registers bank select bit 0
3	RS0	R0-R7 Registers bank select bit 1
2	OV	Overflow flag
1	-	-
0	P	Parity flag

RS1	RS0	Active Bank	Address
0	0	0	00h-07h
0	1	1	08h-0Fh
1	0	2	10h-17h
1	1	3	18-1Fh

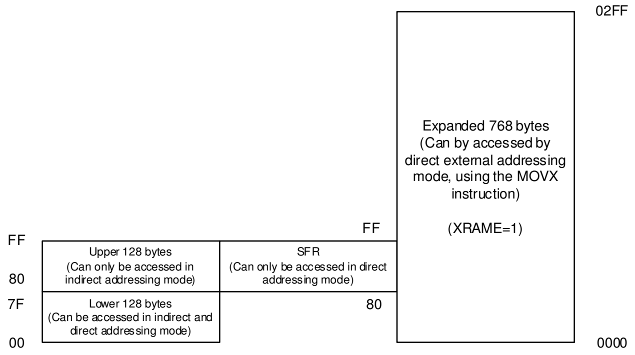
Data Pointer

The VRS1000 has one 16-bit data pointer. The DPTR is accessed through two SFR addresses: DPL located at address 82h and DPH located at address 83h.

Data Memory

The VRS1000 has 1K of on-chip RAM: 256 bytes are configured like the internal memory structure of a standard 80C52, while the remaining 768 bytes can be accessed using external memory addressing (MOVX).

FIGURE 6: VRS1000 DATA MEMORY



By default after reset, the expanded RAM area is disabled. It can be enabled by setting the XRAME bit of the SYSCON register located at address BFh in the SFR.

Lower 128 bytes (00h to 7Fh, Bank 0 & Bank 1)

The lower 128 bytes of data memory (from 00h to 7Fh) can be summarized in the following points:

- Address range 00h to 7Fh can be accessed in direct and indirect addressing modes.
- Address range 00h to 1Fh includes R0-R7 registers area.
- Address range 20h to 2Fh is bit addressable.
- Address range 30h to 7Fh is not bit addressable and can be used as general-purpose storage.

Upper 128 bytes (80h to FFh, Bank 2 & Bank 3)

The upper 128 bytes of the data memory ranging from 80h to FFh can be accessed using indirect addressing or by using the bank mapping in direct addressing mode.

Expanded RAM Access Using the MOVX @DPTR Instruction (0000-02FF, Bank4-Bank15)

The 768 bytes of the expanded RAM data memory occupy addresses 0000h to 02FFh. It can be accessed using external direct addressing (i.e. using the MOVX instruction) or by using bank mapping direct addressing. Note that in the case of indirect addressing using the *MOVX @DPTR* instruction, if the address is larger than 02FFh, the VRS1000 will generate the external memory control signal automatically.

Internal RAM Control Register

The 768 bytes of expanded RAM of the VRS1000 can also be accessed using the MOVX @Rn instruction (where n = 0,1). This instruction is only able to access data in a range of 256 bytes. The internal RAM Control Register RCON allows users to select which part of the expanded RAM will be targeted by the instruction, by configuring the value of the RAMS0 and RAMS1 bits.
The default setting of the RAMS1 and RAMS0 bits is 00 (page 0). Each page has 256 bytes.

TABLE 12: INTERNAL RAM CONTROL REGISTER (RCON) - SFR 85H

7	6	5	4	3	2	1	0
Unused						RAMS1	RAMS0

Bit	Mnemonic	Description
7	Unused	-
6	Unused	-
5	Unused	-
4	Unused	-
3	Unused	-
2	Unused	-
1	RAMS1	These two bits are used with Rn of instruction MOVX @Rn, n=1,0 for mapping (see section on extended 768 bytes) RAMS1, RAMS0 Mapped area 00 000h-0FFh 01 100h-1FFh 10 200h-2FFh 11 XY00h-XYFF* *Externally generated
0	RAMS0	

Example:

Suppose that RAMS1, RAMS0 are set to 0 and 1 respectively and Rn has a value of 45h.

Performing *MOVX @Rn, A*, (where n is 0 or 1) allows the user to transfer the value of A to the expanded RAM at address 145h (page 1).

It is important to note that when both RAMS1, RAMS0 are set to 1, the value of P2 defines the upper byte of the external address accessed. Rn defines the lower byte of the address. In this case, the VRS1000 will generate the external memory control signals automatically. This allows users to access externally mapped devices in the “P2value”00h to “P2value”FFh range.

Data Bank Control Register

The DBANK register allows the user to enable the Data Bank Select function and map the entire content of the RAM memory in the range of 40h to 7Fh for applications that would require direct addressing of the expanded RAM content.

The Data Bank Select function is activated by setting the Data Bank Select enable bit (BSE) to 1. Setting this bit to zero disables this function. The four least significant bits of this register controls the mapping of the entire 1K byte on-chip RAM space into the 040h-07Fh range.

TABLE 13: DATA BANK CONTROL REGISTER (DBANK) – SFR 86H

7	6	5	4	3	2	1	0
BSE	Unused			BS3	BS2	BS1	BS0

Bit	Mnemonic	Description
7	BSE	Data Bank Select Enable Bit BSE=1, Data Bank Select enabled BSE=0, Data Bank Select disabled
6	Unused	-
5	Unused	-
4	Unused	-
3	BS3	Allows the mapping of the 1K RAM into the 040h - 07Fh RAM space
2	BS2	
1	BS1	
0	BS0	

The windowed access to all the 1K on-chip RAM in the range of 40h-7Fh is described in the following table.

TABLE 14: BANK MAPPING DIRECT ADDRESSING MODE

BS3	BS2	BS1	BSO	040h~07fh mapping address	Note
0	0	0	0	000h-03Fh	Lower 128 byte RAM
0	0	0	1	040h-07Fh	Lower 128 byte RAM
0	0	1	0	080h-0BFh	Upper 128 byte RAM
0	0	1	1	0C0h-0FFh	Upper 128 byte RAM
0	1	0	0	0000h-003Fh	On-chip expanded 768 byte RAM
0	1	0	1	0040h-007Fh	On-chip expanded 768 byte RAM
0	1	1	0	0080h-00BFh	On-chip expanded 768 byte RAM
0	1	1	1	00C0h-00FFh	On-chip expanded 768 byte RAM
1	0	0	0	0100h-013Fh	On-chip expanded 768 byte RAM
1	0	0	1	0140h-017Fh	On-chip expanded 768 byte RAM
1	0	1	0	0180h-01BFh	On-chip expanded 768 byte RAM
1	0	1	1	01C0h-01FFh	On-chip expanded 768 byte RAM
1	1	0	0	0200h-023Fh	On-chip expanded 768 byte RAM
1	1	0	1	0240h-027Fh	On-chip expanded 768 byte RAM
1	1	1	0	0280h-02BFh	On-chip expanded 768 byte RAM
1	1	1	1	02C0h-02FFh	On-chip expanded 768 byte RAM

Example: User writes #55h to address 203h:

```
MOV   DBANK, #8CH                ;Set bank mapping 40h-07Fh to
                                 0200h-023Fh
MOV   A, #55H                    ;Store #55H to A
MOV   43H, A                     ;Write #55H to 0203h ;address
```

Description of Peripherals

System Control Register

The register represented in the next table is used for system control. Bit 7 indicates if the system has been reset due to the overflow of the Watch Dog Timer. It is for this reason that users should check the WDR bit whenever an unpredicted reset occurs.

The IAPE bit is used to enable and disable the IAP function. When set to 1, the XRAME bit allows the user to enable the on-chip expanded 768 bytes of RAM. Bit 0 of this register is the ALE output inhibit bit. Setting this bit to 1 will inhibit the Fosc/6Hz clock signal output to the ALE pin.

TABLE 15: SYSTEM CONTROL REGISTER (SYSCON) – SFR BFH

7	6	5	4	3	2	1	0
WDR	Unused				IAPE	XRAME	ALEI

Bit	Mnemonic	Description
7	WDR	This is the Watch Dog Timer reset bit. It will be set to 1 when the reset signal generated by WDT overflows.
6	Unused	-
5	Unused	-
4	Unused	-
3	Unused	-
2	IAPE	IAP function enable bit
1	XRAME	768 byte on-chip enable bit
0	ALEI	ALE output inhibit bit, which is used to reduce EMI.

Power Control Register

The VRS1000 provides two power saving modes: Idle and Power Down. These two modes serve to reduce the power consumption of the device.

In Idle mode, the processor is stopped but the oscillator is still running. The content of the RAM, I/O state and SFR registers are maintained. Timer operation is maintained, as well as the external interrupts.

This mode is useful for applications in which stopping the processor to save power is required. The processor will be woken up when an external event, triggering an interrupt, occurs.

In Power Down mode, the oscillator of the VRS1000 is stopped. This means that all the peripherals are disabled. The content of the RAM and the SFR registers, however, is maintained.

These power saving modes are controlled by the PDOWN and IDLE bits of the PCON register at address 87h.

TABLE 16: POWER CONTROL REGISTER (PCON) - SFR 87H

7	6	5	4	3	2	1	0
Unused						RAMS1	RAMS0

Bit	Mnemonic	Description
7	SMOD	1: Double the baud rate of the serial port frequency that was generated by Timer 1. 0: Normal serial port baud rate generated by Timer 1.
6		
5		
4		
3	GF1	General Purpose Flag
2	GF0	General Purpose Flag
1	PDOWN	Power down mode control bit
0	IDLE	Idle mode control bit

Input/Output Ports

The VRS1000 has 36 bi-directional lines grouped in four 8-bit I/O ports and one 4-bit I/O port. These I/Os can be individually configured as input or output

Except for the P0 I/Os, which are of the open drain type, each I/O is made of a transistor connected to ground and a dynamic pull-up resistor made of a combination of transistors.

Writing a 0 in a given I/O port bit register will activate the transistor connected to ground, this will bring the I/O to a LOW level.

Writing a 1 into a given I/O port bit register de-activates the transistor between the pin and ground. In this case the pull-up resistor will bring the PIN to a HIGH level.

To use a given I/O as an input, one must write a 1 into its associated port register bit.

By default, upon reset all the I/Os are configured as input.

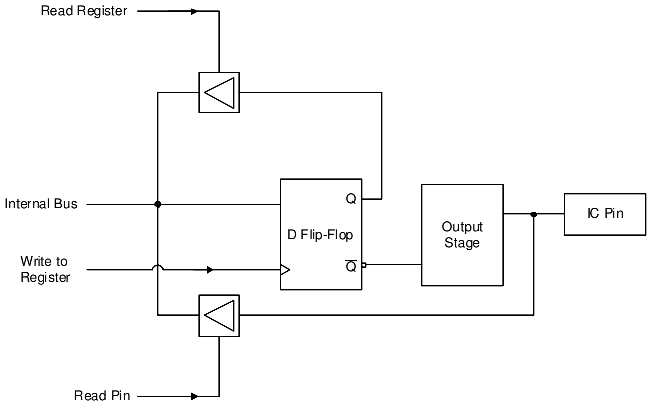
General Structure of an I/O Port

The following elements establish the link between the core unit and the pins of the microcontroller:

Special Function Register (same name as port)
Output Stage Amplifier (the structure of this element varies with its auxiliary function)

From the next figure one may see that the D flip-flop stores the value received from the internal bus after receiving a write signal from the core. Also, notice that the Q output of the flip-flop can be linked to the internal bus by executing a read instruction.
This is how one would read the content of the register. It is also possible to link the value of the pin to the internal bus. This is done by the "read pin" instruction. In short, the user may read the value of the register or the pin.

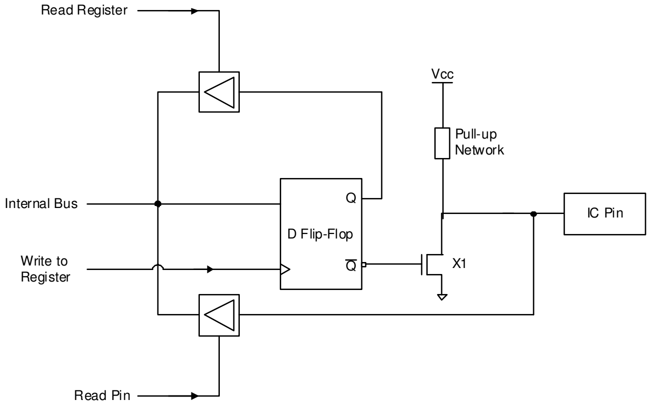
FIGURE 7: INTERNAL STRUCTURE OF ONE OF THE EIGHT I/O PORT LINES



Structure of the P1, P2 and P3 Ports

The following figure gives a general idea of the structure of one of the lines of the P1, P2 and P3 ports. For each port, the output stage is composed of a transistor (X1) and 3 other pull-up transistors. It is important to note that the figure below does not show the intermediary logic that connects the output of the register and the output stage together because this logic varies with the auxiliary function of each port.

FIGURE 8: GENERAL STRUCTURE OF THE OUTPUT STAGE OF P1, P2 AND P3



Each line may be used independently as a logical input or output. When used as an input, as mentioned earlier, the corresponding bit register must be high. This would correspond to Q=1 and (Q=0) in the above figure.

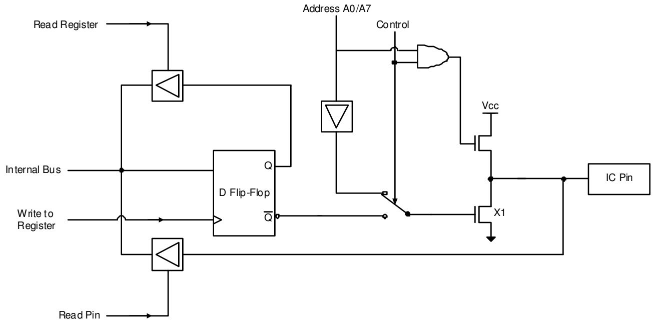
The transistor would be off (open-circuited) and current would flow from the VCC to the pin, generating a logical high at the output. Also, note that if an external device with a logical low value is connected to the pin, the current will flow out of the pin. In order to have a real bi-directional output, the input should be in a high impedance state. It is for this reason that we call ports P1, P2, P3 and P4 “quasi bi-directional”.

Structure of Port 0

The internal structure of P0 is shown in the next figure. The auxiliary function of this port requires a particular logic. As opposed to the other ports, P0 is truly bi-directional. In other words, when used as an input, it is considered to be in a floating logical state (high impedance state). This arises from the absence of the internal pull-up resistance. The pull-up resistance is actually replaced by a transistor that is only used when the port functions to access external memory/data bus (EA=0).

When used as an I/O port, P0 acts as an open drain port and the use of an external pull-up resistor is likely to be required for most applications.

FIGURE 9: PORT P0’S PARTICULAR STRUCTURE



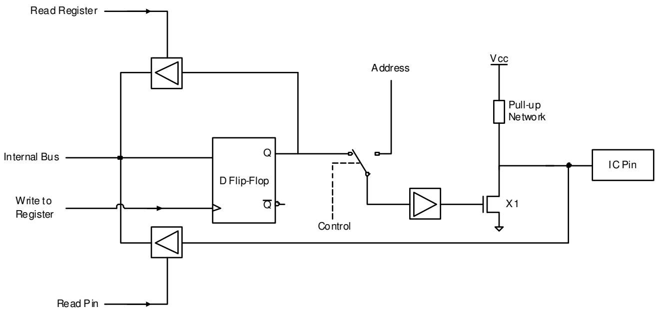
When P0 is used as an external memory bus input (for a MOVX instruction, for example), the outputs of the register are automatically forced to 1.

Port P0 and P2 as Address and Data Bus

The output stage may receive data from two sources

- The outputs of register P0 or the bus address itself multiplexed with the data bus for P0.
- The outputs of the P2 register or the high part (A8/A15) of the bus address for the P2 port.

FIGURE 10: P2 PORT STRUCTURE



When the ports are used as an address or data bus, the special function registers P0 and P2 are disconnected from the output stage. The 8 bits of the P0 register are forced to 1 and the content of the P2 register remains constant.

Auxiliary Port1 Functions

The Port1 I/O pins are shared with the PWM outputs, Timer 2 EXT and T2 inputs as shown below:

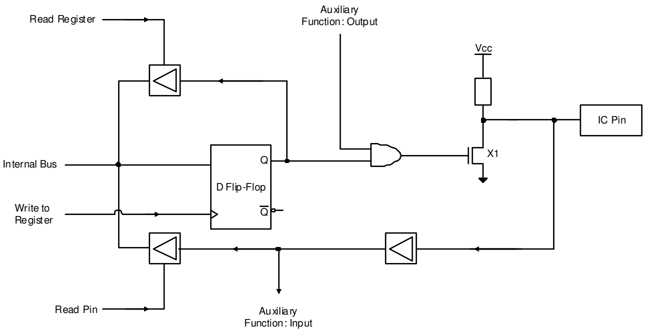
Pin	Mnemonic	Function
P1.0	T2	Timer 2 counter input
P1.1	T2EX	Timer2 Auxiliary input
P1.2		
P1.3	PWM0	PWM0 output
P1.4	PWM1	PWM1 output
P1.5	PWM2	PWM2 output
P1.6	PWM3	PWM3 output
P1.7	PWM4	PWM4 output

Auxiliary P3 Port Functions

The Port3 I/O pins are shared with the UART interface, INT0 and INT1 interrupts, Timer 0 and Timer 1 inputs and finally the #WR and #RD lines when external memory access is performed.

To maintain the correct functionality of the line in auxiliary function mode, it is necessary that the Q output of register is held stable at 1. Conversely, if the pull-down transistor continues conducting, it will set the IC pin at a voltage of approximately 0.

FIGURE 11: P3 PORT STRUCTURE



The following table describes the auxiliary function of the Port3 I/O pins.

TABLE 17: P3 AUXILIARY FUNCTION TABLE

Pin	Mnemonic	Function
P3.0	RXD	Serial Port: Receive data in asynchronous mode. Input and output data in synchronous mode.
P3.1	TXD	Serial Port: Transmit data in asynchronous mode. Output clock value in synchronous mode.
P3.2	$\overline{\text{INT0}}$	External Interrupt 0 Timer 0 Control Input
P3.3	$\overline{\text{INT1}}$	External Interrupt 1 Timer 1 Control Input
P3.4	T0	Timer 0 Counter Input
P3.5	T1	Timer 1 Counter Input
P3.6	$\overline{\text{WR}}$	Write signal for external memory
P3.7	$\overline{\text{RD}}$	Read signal for external memory

Port4

Port4 has four pins and its port address is located at 0D8H.

TABLE 18: PORT 4 (P4) - SFR D8H

7	6	5	4	3	2	1	0
Unused				P4.3	P4.2	P4.1	P4.0

Bit	Mnemonic	Description
7	Unused	-
6	Unused	-
5	Unused	-
4	Unused	-
3	P4.3	Used to output the setting to pins P4.3, P4.2, P4.1, P4.0 respectively.
2	P4.2	
1	P4.1	
0	P4.0	

Software Particularities Concerning the Ports

Some instructions allow the user to read the logic state of the output pin, while others allow the user to read the content of the associated port register. These instructions are called *read-modify-write* instructions. A list of these instructions may be found in the table below.

Upon execution of these instructions, the content of the port register (at least 1 bit) is modified. The other read instructions take the present state of the input into account. For example, the instruction ANL P3,#01h obtains the value in the P3 register; performs the desired logic operation with the constant 01h; and recopies the result into the P3 register. When users want to take the present state of the inputs into account, they must first read these states and perform an AND operation between the reading and the constant.

```
MOV A, P3; State of the inputs in the accumulator
ANL A, #01; AND operation between P3 and 01h
```

When the port is used as an output, the register contains information on the state of the output pins. Measuring the state of an output directly on the pin is inaccurate because the electrical level depends mostly on the type of charge that is applied to it. The functions shown below take the value of the register rather than that of the pin.

TABLE 19: LIST OF INSTRUCTIONS THAT READ AND MODIFY THE PORT USING REGISTER VALUES

Instruction	Function
ANL	Logical AND ex: ANL P0, A
ORL	Logical OR ex: ORL P2, #01110000B
XRL	Exclusive OR ex: XRL P1, A
JBC	Jump if the bit of the port is set to 0
CPL	Complement one bit of the port
INC	Increment the port register by 1
DEC	Decrement the port register by 1
DJNZ	Decrement by 1 and jump if the result is not equal to 0
MOV P.,C	Copy the held bit C to the port
CLR P.x	Set the port bit to 0
SETB P.x	Set the port bit to 1

Port Operation Timing

Writing to a Port (Output)

When an operation induces a modification of the content in a port register, the new value is placed at the output of the D flip-flop during the T12 period of the last machine cycle that the instruction needed to execute.

It is important to note, however, that the output stage only samples the output of the registers on the P1 phase of each period. It follows that the new value only appears at the output after the T12 period of the following machine cycle.

Reading a Port (Input)

The reading of an I/O pin takes place:
- During T9 cycle for P0, P1
- During T10 cycle for P2, P3

When the ports are configured as I/Os
In order to get sampled, the signal duration present on the I/O inputs must have a duration longer than Fosc/12.

I/O Ports Driving Capability

The maximum allowable continuous current that the device can sink on I/O port is defined by the following

Maximum sink current on one given I/O	10mA
Maximum total sink current for P0	26mA
Maximum total sink current for P1, 2, 3	15mA
Maximum total sink current for P4	20mA
Maximum total sink current on all I/O	91mA

On the VRS1000, the Port4 output buffers can sink up to 20mA, which render possible direct driving of LED displays.

It is not recommended to exceed the sink current expressed in the above table. Doing so is likely to make the low-level output voltage exceed the device's specification and it is likely to affect the device's reliability.

The VRS1000 I/O ports are not designed to source current.

Timers

The VRS1000 includes three 16-bit timers: T0, T1 and T2.

The timers can operate in two specific modes:

- Event counting mode
- Timer mode

When operating in event counting mode, the counter is incremented each time an external event, such as a transition in the logical state of the timer input (T0, T1, T2 input), is detected. When operating in timer mode, the counter is incremented by the microcontroller's direct clock pulse or by a divided version of this pulse.

Timer 0 and Timer 1

Timers 0 and 1 have four modes of operation. These modes allow the user to change the size of the counting register or to authorize an automatic reload when provided with a specific value. Timer 1 can even be used as a baud rate generator to generate communication frequencies for the serial interface.
Timer 1 and Timer 0 are configured by the TMOD and TCON registers.

Table 20: Timer Mode Control Register (TMOD) – SFR 89h

7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1.0	M0.0

Bit	Mnemonic	Description
7	GATE1	1: Enables external gate control (pin INT1 for Counter 1). When INT1 is high, and TRx bit is set (see TCON register), a counter is incremented every falling edge on the T1IN input pin.
6	C/T1	Selects timer or counter operation (Timer 1). 1 = A counter operation is performed 0 = The corresponding register will function as a timer.
5	M1.1	Selects mode for Timer/Counter 1
4	M0.1	Selects mode for Timer/Counter 1
3	GATE0	If set, enables external gate control (pin INT0 for Counter 0). When INT0 is high, and TRx bit is set (see TCON register), a counter is incremented every falling edge on the T0IN input pin.
2	C/T0	Selects timer or counter operation (Timer 0). 1 = A counter operation is performed 0 = The corresponding register will function as a timer.
1	M1.0	Selects mode for Timer/Counter 0.
0	M0.0	Selects mode for Timer/Counter 0.

The table below summarizes the four modes of operation of timers 0 and 1. The timer operating mode is selected by the bits M1 and M0 of the TMOD register.

Table 21: Timer/Counter Mode Description Summary

M1	M0	Mode	Function
0	0	Mode 0	13-bit Counter
0	1	Mode 1	16-bit Counter
1	0	Mode 2	8-bit auto-reload Counter/Timer. The reload value is kept in TH0 or TH1, while TL0 or TL1 is incremented every machine cycle. When TLx overflows, the value of THx is copied to TLx.
1	1	Mode 3	If Timer 1 M1 and M0 bits are set to 1, Timer 1 stops.

Timer 0 /Timer 1 Counter / Timer Functions

Timing Function

When operating as a timer, the counter is automatically incremented at every machine cycle. A flag is raised in the event that an overflow occurs and the counter acquires a value of zero. These flags (TF0 and TF1) are located in the TCON register.

TABLE 22: TIMER 0 AND 1 CONTROL REGISTER (TCON) –SFR 88H

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit	Mnemonic	Description
7	TF1	Timer 1 Overflow Flag. Set by hardware on Timer/Counter overflow. Cleared by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
6	TR1	Timer 1 Run Control Bit. Set/cleared by software to turn Timer/Counter on or off.
5	TF0	Timer 0 Overflow Flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
4	TR0	Timer 0 Run Control Bit. Set/cleared by software to turn Timer/Counter on or off.
3	IE1	Interrupt Edge Flag. Set by hardware when external interrupt edge is detected. Cleared when interrupt processed.
2	IT1	Interrupt 1 Type Control Bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
1	IE0	Interrupt 0 Edge Flag. Set by hardware when external interrupt edge is detected. Cleared when interrupt processed.
0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Counting Function

When operating as a counter, the timer's register is incremented at every falling edge of the T0, T1 and T2 signals located at the input of the timer. In this case, the signal is sampled at the T10 phase of each machine cycle for Timer 0, Timer 1 and T9 for Timer 2. When the sampler sees a high immediately followed by a low in the next machine cycle, the counter is incremented. Two machine cycles are required to detect and record an event. This reduces the counting frequency by a factor of 24 (24 times less than the oscillator's frequency).

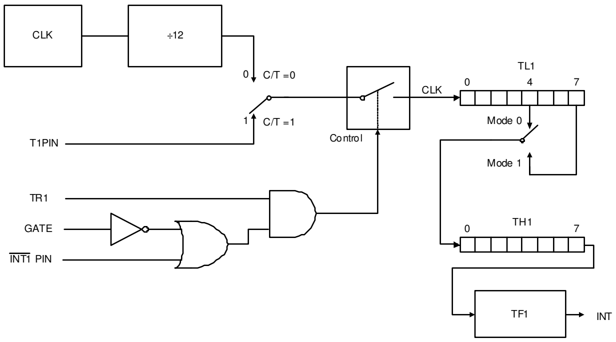
Timer 0 / Timer 1 Operating Modes

The user may change the operating mode by varying the M1 and M0 bits of the TMOD SFR.

Mode 0

A schematic representation of this mode of operation can be found in the figure below. In Mode 0, the timer operates as an 8-bit counter preceded by a divide-by-32 prescaler made of the 5LSB of TL1. The register of the counter is configured to be 13 bits long. When an overflow causes the value of the register to roll over to 0, the TFx interrupt signal goes to 1. The count value is validated as soon as TRx goes to 1 and the GATE bit is 0, or when INTx is 1.

FIGURE 12: TIMER/COUNTER 1 MODE 0: 13-BIT COUNTER



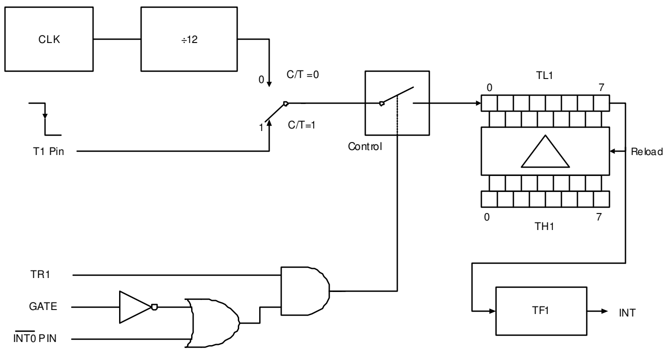
Mode 1

Mode 1 is almost identical to Mode 0. They differ in that, in Mode 1, the counter uses the full 16 bits and has no prescaler.

Mode 2

In this mode, the register of the timer is configured as an 8-bit automatically re-loadable counter. In Mode 2, it is the lower byte TLx that is used as the counter. In the event of a counter overflow, the TFx flag is set to 1 and the value contained in THx, which is preset by software, is reloaded into the TLx counter. The value of THx remains unchanged.

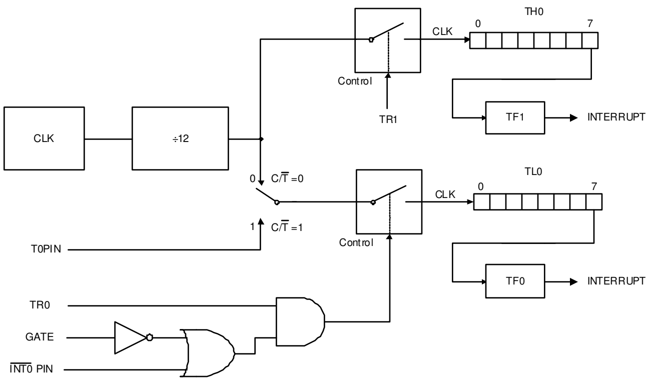
FIGURE 13: TIMER/COUNTER 1 MODE 2: 8-BIT AUTOMATIC RELOAD



Mode 3

In Mode 3 the Timer 1 is blocked as if its control bit, TR1, was set to 0. In this mode, Timer 0's registers TL0 and TH0 are configured as two separate 8-bit counters. Also, the TL0 counter uses Timer 0's control bits C/T, GATE, TR0, INT0, TF0 and the TH0 counter is held in Timer Mode (counting machine cycles) and gains control over TR1 and TF1 from Timer 1. At this point, TH0 controls the Timer 1 interrupt.

FIGURE 14: TIMER/COUNTER 0 MODE 3



Timer 2

Timer 2 of the VRS1000 is a 16-bit Timer/Counter. Similar to timers 0 and 1, Timer 2 can operate either as an event counter or as a timer. The user may switch functions by writing to the C/T2 bit located in the T2CON special function register. Timer 2 has three operating modes: "Auto-Load" "Capture", and "Baud Rate Generator". The T2CON SFR configures the modes of operation of Timer 2. The table below describes each bit in the T2CON special function register.

TABLE 23: TIMER 2 CONTROL REGISTER (T2CON) –SFR C8H

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

Bit	Mnemonic	Description
7	TF2	Timer 2 Overflow Flag: Set by an overflow of Timer 2 and must be cleared by software. TF2 will not be set when either RCLK =1 or TCLK =1.
6	EXF2	Timer 2 external flag change in state occurs when either a capture or reload is caused by a negative transition on T2EX and EXEN2=1. When Timer 2 is enabled, EXF=1 will cause the CPU to Vector to the Timer 2 interrupt routine. Note that EXF2 must be cleared by software.
5	RCLK	Serial Port Receive Clock Source. 1: Causes Serial Port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. 0: Causes Timer 1 overflow to be used for the Serial Port receive clock.
4	TCLK	Serial Port Transmit Clock. 1: Causes Serial Port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. 0: Causes Timer 1 overflow to be used for the Serial Port transmit clock.
3	EXEN2	Timer 2 External Mode Enable. 1: Allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the Serial Port. 0: Causes Timer 2 to ignore events at T2EX.
2	TR2	Start/Stop Control for Timer 2. 1: Start Timer 2 0: Stop Timer 2
1	C/$\overline{T2}$	Timer or Counter Select (Timer 2) 1: External event counter falling edge triggered. 0: Internal Timer (OSC/12)
0	CP/$\overline{RL2}$	Capture/Reload Select. 1: Capture of Timer 2 value into RCAP2H, RCAP2L is performed if EXEN2=1 and a negative transitions occurs on the T2EX pin. The capture mode requires RCLK and TCLK to be 0. 0: Auto-reload reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2=1. When either RCK =1 or TCLK =1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

The possible combinations of control bits that may be used for the mode selection of Timer 2 are shown below:

TABLE 24: TIMER 2 MODE SELECTION BITS

RCLK + TCLK	$CP/\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-Reload Mode
0	1	1	16-bit Capture Mode
1	X	1	Baud Rate Generator Mode
X	X	0	Off

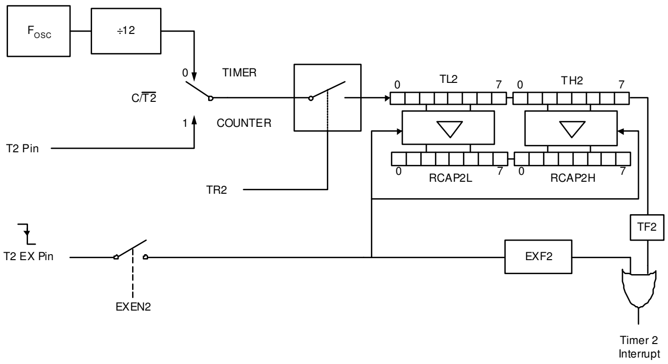
The details of each mode are described below.

Timer 2 Capture Mode

In Capture Mode the EXEN2 bit value defines if the external transition on the T2EX pin will be able to trigger the capture of the timer value.

When EXEN2 = 0, Timer 2 acts as a 16-bit timer or counter, which, upon overflowing, will set bit TF2 (Timer 2 overflow bit). This overflow can be used to generate an interrupt.

FIGURE 15: TIMER 2 IN CAPTURE MODE



When EXEN2 = 1, the above still applies. In addition, it is possible to allow a 1 to 0 transition at the T2EX input to cause the current value stored in the Timer 2 registers (TL2 and TH2) to be captured into the RCAP2L and RCAP2H registers. Furthermore, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt.

Note that both EXF2 and TF2 share the same interrupt vector.

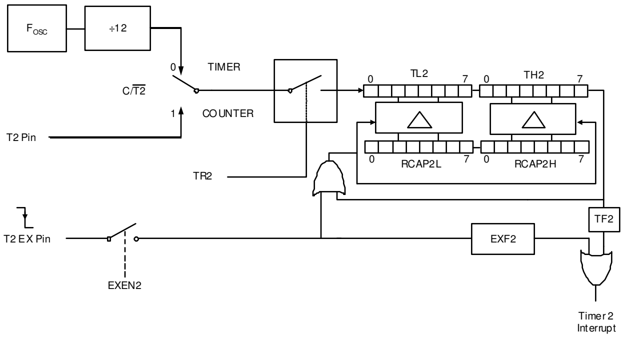
Timer 2 Auto-Reload Mode

In this mode, there are also two options. The user may choose either option by writing to bit EXEN2 in T2CON.

If EXEN2 = 0, when Timer 2 rolls over, it not only sets TF2, but also causes the Timer 2 registers to be reloaded with the 16-bit value in the RCAP2L and RCAP2H registers previously initialised. In this mode, Timer 2 can be used as a baud rate generator source for the serial port.

If EXEN2=1, then Timer 2 still performs the above operation, but a 1 to 0 transition at the external T2EX input will also trigger an anticipated reload of the Timer 2 with the value stored in RCAP2L, RCAP2H and set EXF2.

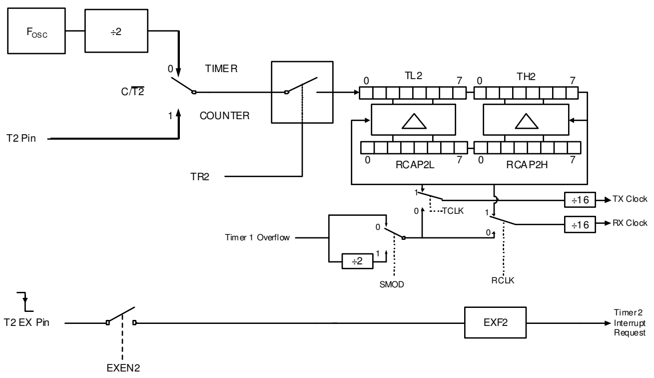
FIGURE 16: TIMER 2 IN AUTO-RELOAD MODE



Timer 2 Baud Rate Generator Mode

This mode is activated when RCLK is set to 1 and/or TCLK is set to 1. This mode will be described in the serial port section.

FIGURE 17: TIMER 2 IN AUTOMATIC BAUD GENERATOR MODE



UART Serial Port

The serial port on the VRS1000 can operate in full duplex; in other words, it can transmit and receive data simultaneously. This occurs at the same speed if one timer is assigned as the clock source for both transmission and reception, and at different speeds if transmission and reception are each controlled by their own timer.

The serial port receive is buffered, which means that it can begin reception of a byte even if the one previously received byte has not been retrieved from the receive register by the processor. However, if the first byte still has not been read by the time reception of the second byte is complete, the byte present in the receive buffer will be lost.

One SFR register, SBUF, gives access to the transmit and receive registers of the serial port. When users read from the SBUF register, they will access the receive register. When users write to the SBUF, the transmit register will be loaded.

UART Control Register

The serial port control register and status register (SCON) contain the 9th data bit for transmit and receive (TB8 and RB8) and all the mode selection bits. SCON also contains the serial port interrupt bits (TI and RI).

TABLE 25: SERIAL PORT CONTROL REGISTER (SCON) – SFR 98H

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Bit	Mnemonic	Description
7	SM0	Bit to select mode of operation (see table below)
6	SM1	Bit to select mode of operation (see table below)
5	SM2	Multiprocessor communication is possible in modes 2 and 3. In modes 2 or 3 if SM2 is set to 1, RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received.
4	REN	Serial Reception Enable Bit This bit must be set by software and cleared by software. 1: Serial reception enabled 0: Serial reception disabled
3	TB8	9th data bit transmitted in modes 2 and 3 This bit must be set by software and cleared by software.
2	RB8	9th data bit received in modes 2 and 3. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, this bit is not used. This bit must be cleared by software.
1	TI	Transmission Interrupt flag. Automatically set to 1 when: • The 8th bit has been sent in Mode 0. • Automatically set to 1 when the stop bit has been sent in the other modes. This bit must be cleared by software.
0	RI	Reception Interrupt flag Automatically set to 1 when: • The 8th bit has been received in Mode 0. • Automatically set to 1 when the stop bit has been sent in the other modes (see SM2 exception). This bit must be cleared by software.

TABLE 26: SERIAL PORT MODES OF OPERATION

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	$F_{osc}/12$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$F_{osc}/64$ or $F_{osc}/32$
1	1	3	9-bit UART	Variable

UART Operating Modes

The VRS1000's serial port can operate in four different modes. In all four modes, a transmission is initiated by an instruction that uses the SBUF SFR as a destination register. In Mode 0, reception is initiated by setting RI to 0 and REN to 1. An incoming start bit initiates reception in the other modes provided that REN is set to 1. The following paragraphs describe the four modes.

UART Operation in Mode 0

In this mode, the serial data exits and enters through the RXD pin. TXD is used to output the shift clock. The signal is composed of 8 data bits starting with the LSB. The baud rate in this mode is 1/12 the oscillator frequency.

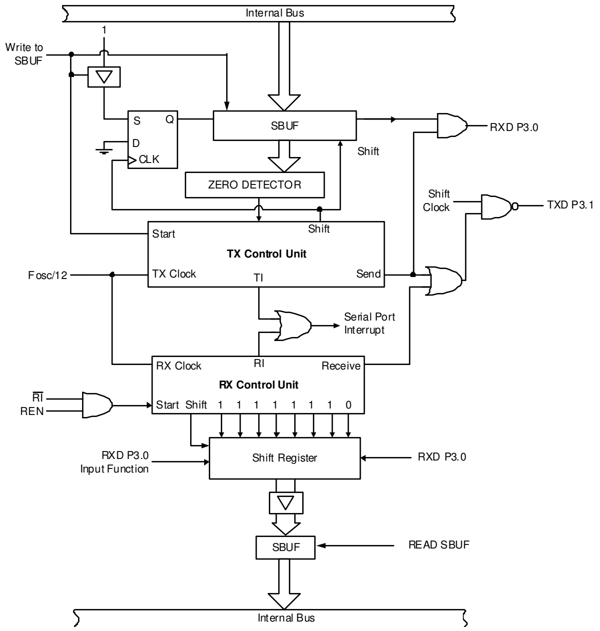


FIGURE 18: SERIAL PORT MODE 0 BLOCK DIAGRAM

UART Transmission in Mode 0

Any instruction that uses SBUF as a destination register may initiate a transmission. The "write to SBUF" signal also loads a 1 into the 9th position of the transmit shift register and tells the TX control block to begin a transmission. The internal timing is such that one full machine cycle will elapse between a write to SBUF instruction and the activation of SEND.

The SEND signal enables the output of the shift register to the alternate output function line of P3.0 and enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is high during T11, T12 and T1, T2 and T3, T4 of every machine cycle and low during T5, T6, T7, T8, T9 and T10. At T12 of every machine cycle in which SEND is active, the contents of the transmit shift register are shifted to the right by one position.

Zeros come in from the left as data bits shift out to the right. The TX control block sends its final shift and de-activates SEND while setting T1 after one condition is fulfilled: When the MSB of the data byte is at the output position of the shift register; the 1 that was initially loaded into the 9th position is just to the left of the MSB; and all positions to the left of that contain zeros. Once these conditions are met, the de-activation of SEND and the setting of T1 occur at T1 of the 10th machine cycle after the "write to SBUF" pulse.

UART Reception in Mode 0

When REN and R1 are set to 1 and 0 respectively, reception is initiated. The bits 11111110 are written to the receive shift register at T12 of the next machine cycle by the RX control unit. In the following phase, the RX control unit will activate RECEIVE.

SHIFT CLOCK to the alternate output function line of P3.1 is enabled by RECEIVE. At every machine cycle, SHIFT CLOCK makes transitions at T5 and T11. The contents of the receive shift register are shifted one position to the left at T12 of every machine in which RECEIVE is active. The value that comes in from the right is the value that was sampled at the P3.0 pin at T10 of the same machine cycle.

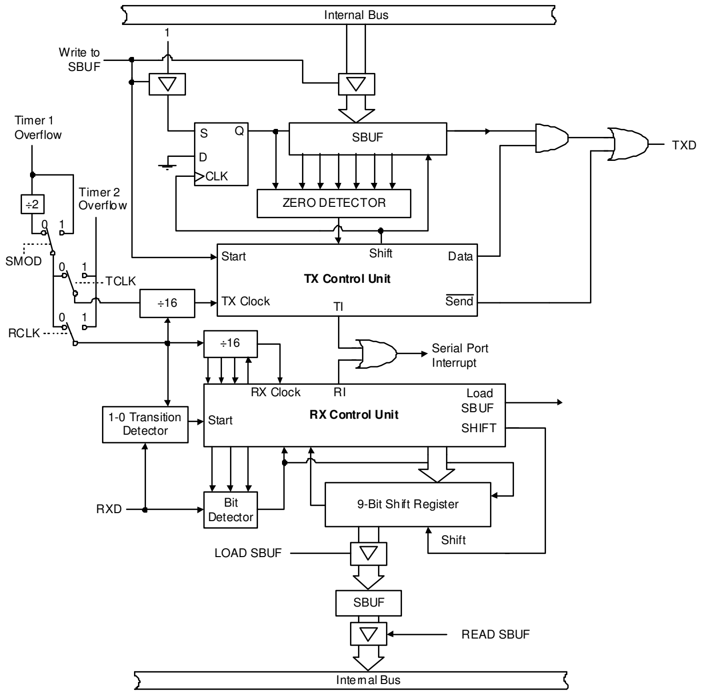
1’s are shifted out to the left as data bits are shifted in from the right. The RX control block is flagged to do one last shift and load SBUF when the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register.

UART Operation in Mode 1

For an operation in Mode 1, 10 bits are transmitted (through TXD) or received (through RXD). The transactions are composed of: a Start bit (Low); 8 data bits (LSB first) and one Stop bit (high). The reception is completed once the Stop bit sets the RB8 flag in the SCON register. Either Timer 1 or Timer 2 controls the baud rate in this mode.

The following diagram shows the serial port structure when configured in Mode 1.

FIGURE 19: SERIAL PORT MODE 1 AND 3 BLOCK DIAGRAM



UART Transmission in Mode 1

Transmission is initiated by any instruction that makes use of SBUF as a destination register. The 9th bit position of the transmit shift register is loaded by the “write to SBUF” signal. This event also flags the TX Control Unit that a transmission has been requested.

It is after the next rollover in the divide-by-16 counter when transmission actually begins at T1 of the machine cycle. It follows that the bit times are synchronized to the divide-by-16 counter and not to the “write to SBUF” signal.

When a transmission begins, it places the start bit at TXD. Data transmission is activated one bit time later. This activation enables the output bit of the transmit shift register to TXD. One bit time after that, the first shift pulse occurs.

In this mode, zeros are clocked in from the left as data bits are shifted out to the right. When the most significant bit of the data byte is at the output position of the shift register, the 1 that was initially loaded into the 9th position is to the immediate left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control Unit to shift one more time.

UART Reception in Mode 1

One to zero transitions at RXD initiate reception. It is for this reason that RXD is sampled at a rate of 16 multiplied by the baud rate that has been established. When a transition is detected, 1FFh is written into the input shift register and the divide-by-16 counter is immediately reset. The divide-by-16 counter is reset in order to align its rollovers with the boundaries of the incoming bit times.

In total, there are 16 states in the counter. During the 7th, 8th and 9th counter states of each bit time; the bit detector samples the value of RXD. The accepted value is the value that was seen in at least two of the three samples. The purpose of doing this is for noise rejection. If the value accepted during the first bit time is not zero, the receive circuits are reset and the unit goes back to searching for another one to zero transition. All false start bits are rejected by doing this. If the start bit is valid, it is shifted into the input shift register, and the reception of the rest of the frame will proceed.

For a receive operation, the data bits come in from the right as 1's shift out on the left. As soon as the start bit arrives at the leftmost position in the shift register, (9-bit register), it tells the RX control block to perform one last shift operation: to set RI and to load SBUF and RB8. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

- Either SM2 = 0 or the received stop bit = 1
- RI = 0

If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. If one of these conditions is not met, the received frame is completely lost. At this time, whether the above conditions are met or not, the unit goes back to searching for a one to zero transition in RXD.

UART operation in Mode 2

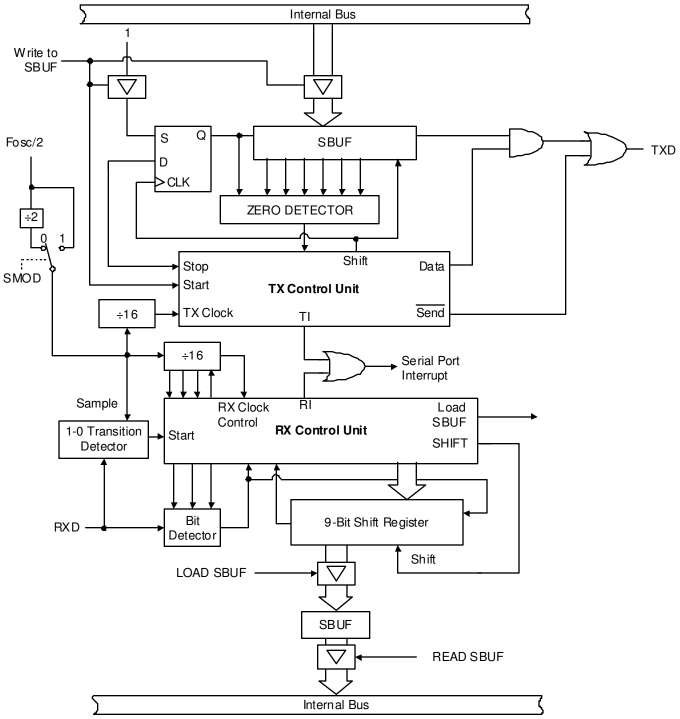
In Mode 2 a total of 11 bits are transmitted (through TXD) or received (through RXD). The transactions are composed of: a Start bit (Low), 8 data bits (LSB first), a programmable 9th data bit, and one Stop bit (High).

For transmission, the 9th data bit comes from the TB8 bit of SCON. For example, the parity bit P in the PSW could be moved into TB8.

In the case of receive, the 9th data bit is automatically written into RB8 of the SCON register.

In Mode 2, the baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

FIGURE 20: SERIAL PORT MODE 2 BLOCK DIAGRAM

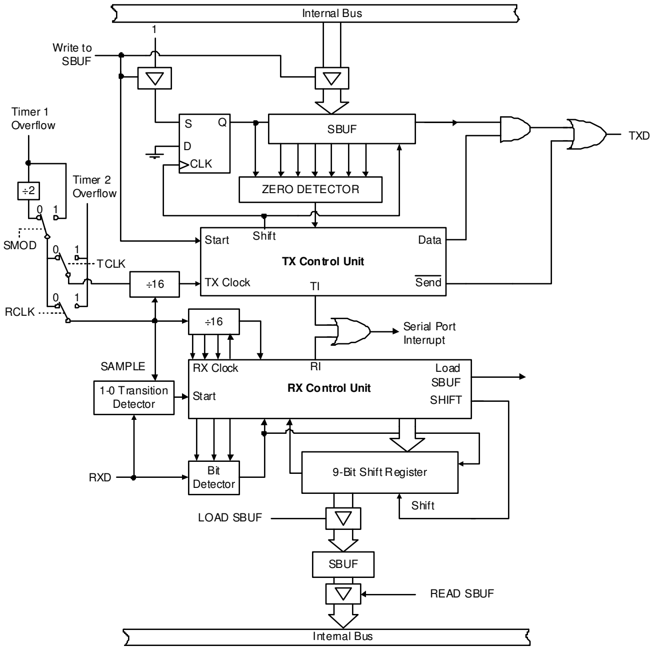


UART Operation in Mode 3

In Mode 3, 11 bits are transmitted (through TXD) or received (through RXD). The transactions are composed of: a Start bit (Low), 8 data bits (LSB first), a programmable 9th data bit, and one Stop bit (High).

Mode 3 is identical to Mode 2 in all respects but one: the baud rate. Either Timer 1 or Timer 2 generates the baud rate in Mode 3.

FIGURE 21: SERIAL PORT MODE 3 BLOCK DIAGRAM



UART in Mode 2 and 3: Additional Information

As mentioned earlier, for an operation in modes 2 and 3, 11 bits are transmitted (through TXD) or received (through RXD). The signal comprises: a logical low Start bit, 8 data bits (LSB first), a programmable 9th data bit, and one logical high Stop bit.
On transmit, (TB8 in SCON) can be assigned the value of 0 or 1. On receive; the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from either Timer 1 or Timer 2 depending on the states of TCLK and RCLK.

UART Transmission in Mode 2 and Mode 3

The transmission is initiated by any instruction that makes use of SBUF as the destination register. The 9th bit position of the transmit shift register is loaded by the "write to SBUF" signal. This event also informs the TX control unit that a transmission has been requested. After the next rollover in the divide-by-16 counter, a transmission actually begins at T1 of the machine cycle. It follows that the bit times are synchronized to the divide-by-16 counter and not to the "write to SBUF" signal, as in the previous mode.

Transmissions begin when the SEND signal is activated, which places the Start bit at TXD. Data is activated one bit time later. This activation enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that.

The first shift clocks a Stop bit (1) into the 9th bit position of the shift register to TXD. Thereafter, only zeros are clocked in. Thus, as data bits shift out to the right, zeros are clocked in from the left. When TB8 is at the output position of the shift register, the stop bit is just to the left of TB8, and all positions to the left of that contain zeros. This condition signals to the TX control unit to shift one more time and set TI, while deactivating SEND. This occurs at the 11th divide-by-16 rollover after "write to SBUF".

UART Reception in Mode 2 and Mode 3

One to zero transitions at RXD initiate reception. It is for this reason that RXD is sampled at a rate of 16 multiplied by the baud rate that has been established. When a transition is detected, the 1FFh is written into the input shift register and the divide-by-16 counter is immediately reset.

During the 7th, 8th and 9th counter states of each bit time; the bit detector samples the value of RXD. The accepted value is the value that was seen in at least two of the three samples. If the value accepted during the first bit time is not zero, the receive circuits are reset and the unit goes back to searching for another one to zero transition. If the start bit is valid, it is shifted into the input shift register, and the reception of the rest of the frame will proceed.

For a receive operation, the data bits come in from the right as 1's shift out on the left. As soon as the start bit arrives at the leftmost position in the shift register (9-bit register), it tells the RX control block to do one more shift, to set RI, and to load SBUF and RB8. The signal to set RI and to load SBUF and RB8 will be generated if, and only if, the following conditions are satisfied at the instance when the final shift pulse is generated:

- Either SM2 = 0 or the received 9th bit is equal to 1
- RI = 0

If both conditions are met, the 9th data bit received goes into RB8, and the first 8 data bits go into SBUF. If one of these conditions is not met, the received frame is completely lost. One bit time later, whether the above conditions are met or not, the unit goes back to searching for a one to zero transition at the RXD input. Please note that the value of the received stop bit is unrelated to SBUF, RB8 or RI.

UART Baud Rates

In Mode 0, the baud rate is fixed and can be represented by the following formula:

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{12}$$

In Mode 2, the baud rate depends on the value of the SMOD bit in the PCON SFR. From the formula below, we can see that if SMOD = 0 (which is the value on reset), the baud rate is 1/32 the oscillator frequency.

$$\text{Mode 2 Baud Rate} = \frac{2^{SMOD} \text{ x (Oscillator Frequency)}}{64}$$

The Timer 1 and/or Timer 2 overflow rate determines the baud rates in modes 1 and 3.

Generating UART Baud Rate with Timer 1

When Timer 1 functions as a baud rate generator, the baud rate in modes 1 and 3 are determined by the Timer 1 overflow rate.

$$\text{Mode 1,3 Baud Rate} = \frac{2^{SMOD} \text{x Timer 1 Overflow Rate}}{32}$$

Timer 1 must be configured as an 8-bit timer (TL1) with auto-reload with TH1 value when an overflow occurs (Mode 2). In this application, the Timer 1 interrupt should be disabled.

The two following formulas can be used to calculate the baud rate and the reload value to put in the TH1 register.

$$\text{Mode 1,3 Baud Rate} = \frac{2^{SMOD} \text{x Fosc}}{32 \text{ x } 12(256 - TH1)}$$

The value to put into the TH1 register is defined by the following formula:

$$TH1 = 256 - \frac{2^{SMOD} \text{ x Fosc}}{32 \text{ x } 12\text{x (Baud Rate)}}$$

It is possible to use Timer 1 in 16-bit mode to generate the baud rate for the serial port. To do this, leave the Timer 1 interrupt enabled, configure the timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and use the Timer 1 interrupt to perform a 16-bit software reload. This can achieve very low baud rates.

Generating UART Baud Rates with Timer 2

Timer 2 is often preferred to generate the baud rate, as it can be easily configured to operate as a 16-bit timer with auto-reload. This allows for much better resolution than using Timer 1 in 8-bit auto-reload mode.

The baud rate using Timer 2 is defined as:

$$\text{Mode 1,3 Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The timer can be configured as either a timer or a counter in any of its 3 running modes. In most typical applications, it is configured as a timer (C/T2 is set to 0).

To make the Timer 2 operate as a baud rate generator the TCLK and RCLK bits of the T2CON register must be set to 1.

The baud rate generator mode is similar to the auto-reload mode in that an overflow in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software. However, when Timer 2 is configured as a baud rate generator, its clock source is Osc/2.

The following formula can be used to calculate the baud rate in modes 1 and 3 using the Timer 2:

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32\text{x}[65536 - (\text{RCAP2H, RCAP2L})]}$$

The formula below is used to define the reload value to put into the RCAP2h, RCAP2L registers to achieve a given baud rate.

$$(\text{RCAP2H, RCAP2L}) = 65536 - \frac{\text{Fosc}}{32\text{x}[\text{Baud Rate}]}$$

In the above formula, RCAP2H and RCAP2L are the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Because of this, the Timer 2 interrupt does not have to be disabled when Timer 2 is configured in baud rate generator mode.

Also, if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from RCAP2x to Tx2. Therefore, when Timer 2 is used as a baud rate generator, T2EX can be used as an extra external interrupt.

Furthermore, when Timer 2 is running (TR2 is set to 1) as a timer in baud rate generator mode, the user should not try to read or write to TH2 or TL2. When operating under these conditions, the timer is being incremented every state time and the results of a read or write command may be inaccurate.

The RCAP2 registers, however, may be read but should not be written to, because a write may overlap a reload operation and generate write and/or reload errors. In this case, before accessing the Timer 2 or RCAP2 registers, be sure to turn the timer off by clearing TR2.

Interrupts

The VRS1000 has 8 interrupt sources (9 if we include the WDT) and 7 interrupt vectors (including reset) to handle them.

The interrupt can be enabled via the IE register shown below:

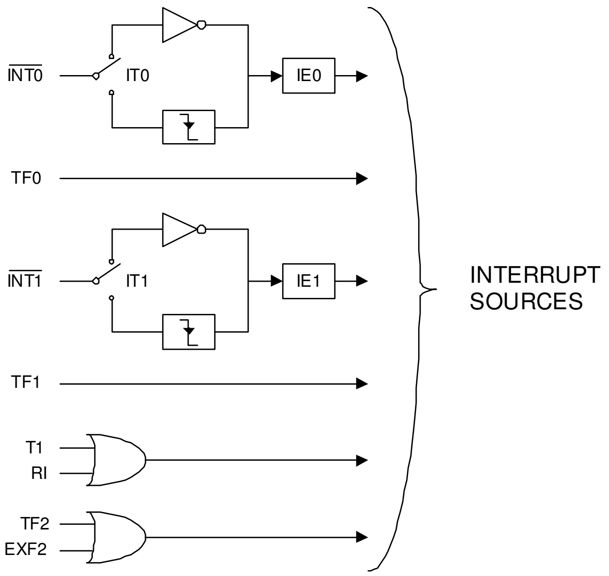
TABLE 27: IE INTERRUPT ENABLE REGISTER –SFR A8H

7	6	5	4	3	2	1	0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

Bit	Mnemonic	Description
7	EA	Disables All Interrupts 0: no interrupt acknowledgment 1: Each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
6	-	Reserved
5	ET2	Timer 2 Interrupt Enable Bit
4	ES	Serial Port Interrupt Enable Bit
3	ET1	Timer 1 Interrupt Enable Bit
2	EX1	External Interrupt 1 Enable Bit
1	ET0	Timer 0 Interrupt Enable Bit
0	EX0	External Interrupt 0 Enable Bit

The following figure illustrates the various interrupt sources on the VRS1000.

FIGURE 22: INTERRUPT SOURCES



Interrupt Vectors

The table shown below specifies each interrupt source, its flag and its vector address.

TABLE 28: INTERRUPT VECTOR ADDRESS

Interrupt Source	Flag	Vector Address
RESET (+ WDT)	WDR	0000h*
INT0	IE0	0003h
Timer 0	TF0	000Bh
INT1	IE1	0013h
Timer 1	TF1	001Bh
Serial Port	RI+TI	0023h
Timer 2	TF2+EXF2	002Bh

*If location 0000h = FFh, the PC jump to the ISP program.

External Interrupts

The VRS1000 has two external interrupt inputs named INT0 and INT1. These interrupt lines are shared with P3.2 and P3.3.

The bits IT0 and IT1 of the TCON register determine whether the external interrupts are level or edge sensitive.

If ITx = 1, the interrupt will be raised when a 1-> 0 transition occurs at the interrupt pin. The duration of the transition must be at least equal to 12 oscillator cycles.

If ITx = 0, the interrupt will occur when a logic low condition is present on the interrupt pin.

The state of the external interrupt, when enabled, can be monitored using the flags, IE0 and IE1 of the TCON register that are set when the interrupt condition occurs.

In the case where the interrupt was configured as edge sensitive, the associated flag is automatically cleared when the interrupt is serviced.

If the interrupt is configured as level sensitive, then the interrupt flag must be cleared by the software.

Timer 0 and Timer 1 Interrupt

Both Timer 0 and Timer 1 can be configured to generate an interrupt when a rollover of the timer/counter occurs (except Timer 0 in Mode 3).

The TF0 and TF1 flags serve to monitor timer overflow occurring from Timer 0 and Timer 1. These interrupt flags are automatically cleared when the interrupt is serviced.

Timer 2 interrupt

Timer 2 interrupt can occur if TF2 and/or EXF2 flags are set to 1 and if the Timer 2 interrupt is enabled.

The TF2 flag is set when a rollover of Timer 2 Counter/Timer occurs. The EXF2 flag can be set by a 1->0 transition on the T2EX pin by the software.

Note that neither flag is cleared by the hardware upon execution of the interrupt service routine. The service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt. These flag bits will have to be cleared by the software.

Every bit that generates interrupts can either be cleared or set by the software, yielding the same result as when the operation is done by the hardware. In other words, pending interrupts can be cancelled and interrupts can be generated by the software.

Serial Port Interrupt

The serial port can generate an interrupt upon byte reception or once the byte transmission is completed.

Those two conditions share the same interrupt vector and it is up to the interrupt service routine to find out what caused the interrupt by looking at the serial interrupt flags RI and TI.

Note that neither of these flags is cleared by the hardware upon execution of the interrupt service routine. The software must clear these flags.

Execution of an Interrupt

When the processor receives an interrupt request, an automatic jump to the desired subroutine occurs. This jump is similar to executing a branch to a subroutine instruction: the processor automatically saves the address of the next instruction on the stack. An internal flag is set to indicate that an interrupt is taking place, and then the jump instruction is executed. An interrupt subroutine must always end with the RETI instruction. This instruction allows users to retrieve the return address placed on the stack.

The RETI instruction also allows updating of the internal flag that will take into account an interrupt with the same priority.

Interrupt Enable and Interrupt Priority

When the VRS1000 is initialized, all interrupt sources are inhibited by the bits of the IE register being reset to 0. It is necessary to start by enabling the interrupt sources that the application requires. This is achieved by setting bits in the IE register, as discussed previously.

This register is part of the bit addressable internal RAM. For this reason, it is possible to modify each bit individually in one instruction without having to modify the other bits of the register. All interrupts can be inhibited by setting EA to 0.

The order in which interrupts are serviced is shown in the following table:

TABLE 29: INTERRUPT PRIORITY

Interrupt Source
RESET + WDT (Highest Priority)
IE0
TF0
IE1
TF1
RI+TI
TF2+EXF2 (Lowest Priority)

Modifying the Order of Priority

The VRS1000 allows the user to modify the natural priority of the interrupts. One may modify the order by programming the bits in the IP (Interrupt Priority) register. When any bit in this register is set to 1, it gives the corresponding source a greater priority than interrupts coming from sources that don't have their corresponding IP bit set to 1.

The IP register is represented in the table below.

TABLE 30: IP INTERRUPT PRIORITY REGISTER –SFR B8H

7	6	5	4	3	2	1	0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

Bit	Mnemonic	Description
7	-	
6	-	
5	PT2	Gives Timer 2 Interrupt Higher Priority
4	PS	Gives Serial Port Interrupt Higher Priority
3	PT1	Gives Timer 1 Interrupt Higher Priority
2	PX1	Gives INT1 Interrupt Higher Priority
1	PT0	Gives Timer 0 Interrupt Higher Priority
0	PX0	Gives INT0 Interrupt Higher Priority

Watch Dog Timer

The Watch Dog Timer (WDT) is a 16-bit free-running counter that generates a reset signal if the counter overflows. The WDT is useful for systems that are susceptible to noise, power glitches and other conditions that can cause the software to go into infinite dead loops or runaways. The WDT function gives the user software a recovery mechanism from abnormal software conditions. The WDT is different from Timer 0, Timer 1 and Timer 2 of the standard 80C52.

Once the WDT is enabled, the user software must clear it periodically. In the case where the WDT is not cleared, its overflow will trigger a reset of the VRS1000.

The user should check the WDR bit of the SYSCON register whenever an unpredicted reset has taken place.

The WDT timeout delay can be adjusted by configuring the clock divider input for the time base source clock of the WDT. To select the divider value, bit2-bit0 (PS2~PS0) of the Watch Dog Timer Control Register (WDTC) should be set accordingly.

To enable the WDT, the user must set bit 7 (WDTE) of the WDTC register to 1. Once WDTE has been set to 1, the 16-bit counter will start to count with the selected time base source clock configured in PS2~PS0. The Watch Dog Timer will generate a reset signal if an overflow has taken place. The WDTE bit will be cleared to 0 automatically when VRS1000 has been reset by either the hardware or a WDT reset.

Clearing the WDT is accomplished by setting the CLR bit of the WDTC to 1. This action will clear the contents of the 16-bit counter and force it to restart.

Watch Dog Timer Registers: WDTC and SYSCON

Two of the registers of the VRS1000 are associated with the Watch Dog Timer: WDTC and SYSCON. The WDTC register allows the user to enable the WDT, to clear the counter and to divide the clock source. The WDR bit of the SYSCON register indicates whether the Watch Dog Timer has caused the device reset.

TABLE 31: WATCH DOG TIMER REGISTERS: WDTC – SFR 9FH

7	6	5	4	3	2	1	0
WDTE	Unused	CLR	Unused		PS2	PS1	PS0

Bit	Mnemonic	Description
7	WDTE	Watch Dog Timer Enable Bit
6	Unused	-
5	CLR	Watch Dog Timer Counter Clear Bit
[4:3]	Unused	-
2	PS2	Clock Source Divider Bit 2
1	PS1	Clock Source Divider Bit 1
0	PS0	Clock Source Divider Bit 0

The next table gives examples of what timeout period the user will obtain for different values of the PSx bits of the Watch Dog Timer Register.

TABLE 32: TIME PERIOD AT 40MHZ, 22.184MHZ AND 11.059MHZ

PS [2:0]	Divider (OSC in)	WDT Period 40MHz	WDT Period 22.18MHz	WDT Period 12MHz
000	8	13.11	23.63	43.69
001	16	26.21	47.27	87.38
010	32	52.43	94.53	174.76
011	64	104.86	189.07	349.53
100	128	209.72	378.14	699.05
101	256	419.43	756.28	1398.10
110	512	838.86	1512.55	2796.20
111	1024	1677.72	3025.10	5592.41

TABLE 33: WATCH DOG TIMER REGISTER-SYSTEM CONTROL REGISTER (SYSCON)–SFR BFH

7	6	5	4	3	2	1	0
WDR	Unused				IAPE	XRAME	ALEI

Bit	Mnemonic	Description
7	WDR	Watch Dog Timer Reset Bit
[6:3]	Unused	-
2	IAPE	ISP Overall Enable Bit 1: Enables ISP Function 0: Disables ISP Function
1	XRAME	
0	ALEI	1: Enable Electromagnetic Interference Reducer 0: Disable Electromagnetic Interference Reducer

As mentioned earlier, bit 7 (WDR) of SYSCON is the Watch Dog Timer Reset bit. It will be set to 1 when a reset signal is generated by the WDT overflow. The user should check the WDR bit whenever an unpredicted reset has taken place.

Reduced EMI Function

The VRS1000 can also be set up to reduce its EMI (electromagnetic interference) by setting bit 0 (ALEI) of the SYSCON register to 1. This function will inhibit the Fosc/6Hz clock signal output to the ALE pin.

Pulse Width Modulation (PWM)

The Pulse Width Modulation (PWM) module has five 8-bit channels. Each channel uses an 8-bit PWM data register (PWMD) to set the number of continuous pulses within a PWM frame cycle.

PWM Function Description:

Each 8-bit PWM channel is composed of an 8-bit register that consists of a 5-bit PWM (5 MSBs) and a 3-bit (LSBs) Narrow Pulse Generator (NP). The 5-bit PWM determines the duty cycle of the output. The 3-bit NPx generates and inserts narrow pulses among the PWM frame made of 8 cycles.

The number of pulses generated is equal to the number programmed in the 3-bit NP. The NP is used to generate an equivalent 8-bit resolution PWM type DAC with a reasonably high repetition rate through a 5-bit PWM clock speed. The PDCK[1:0] settings of the PWMC (A3h) register is used to derive the PWM clock from Fosc.

$$\text{PWM Clock} = \frac{F_{osc}}{2^{(PDCK\,[1:0]\,+1)}}$$

The PWM output cycle frame repetition rate (frequency) is calculated using the following formula:

$$\text{PWM Clock} = \frac{F_{osc}}{32 \times 2^{(PDCK\,[1:0]\,+1)}}$$

PWM Registers - Port1 Configuration Register

TABLE 34: PORT1 CONFIGURATION REGISTER (PWME, $9B)

7	6	5	4
PWM4E	PWM3E	PWM2E	PWM1E

3	2	1	0
PWM0E	Unused		

Bit	Mnemonic	Description
7	PWM4E	When bit is set to one, the corresponding PWM pin is active as a PWM function. When the bit is cleared, the corresponding PWM pin is active as an I/O pin. These five bits are cleared upon reset.
6	PWM3E	
5	PWM2E	
4	PWM1E	
3	PWM0E	
[2:0]	Unused	-

PWM Registers -PWM Control Register

The table below represents the PWM Control Register.

TABLE 35: PWM CONTROL REGISTER (PWMC) – SFR A3H

7	6	5	4	3	2	1	0
Unused						PDCK1	PDCK0

Bit	Mnemonic	Description
[7:2]	Unused	-
1	PDCK1	Input Clock Frequency Divider Bit 1
0	PDCK0	Input Clock Frequency Divider Bit 0

The following table shows the relationship between the values of PDCK1/PDCK0 and the value of the divider. Numerical values of the corresponding frequencies are also provided.

PDCK1	PDCKO	Divider	PWM clock, Fosc=20MHz	PWM clock, Fosc=24MHz
0	0	2	10MHz	12MHz
0	1	4	5MHz	6MHz
1	0	8	2.5MHz	3MHz
1	1	16	1.25MHz	1.5MHz

PWM Data Registers

The tables below show the PWM Data Registers. The PWMDx bits hold the content of the PWM Data Register and determine the duty cycle of the PWM output waveform. The NP[2:0] bits will insert narrow pulses in the 8-PWM-cycle frame.

TABLE 36: PWM DATA REGISTER 0 (PWMD0) – SFR A4H

7	6	5	4
PWMD0.4	PWMD0.3	PWMD0.2	PWMD0.1

3	2	1	0
PWMD0.0	NP0.2	NP0.1	NP0.0

Bit	Mnemonic	Description
7	PWMD0.4	Contents of PWM Data Register 0 Bit 4
6	PWMD0.3	Contents of PWM Data Register 0 Bit 3
5	PWMD0.2	Contents of PWM Data Register 0 Bit 2
4	PWMD0.1	Contents of PWM Data Register 0 Bit 1
3	PWMD0.0	Contents of PWM Data Register 0 Bit 0
2	NP0.2	Inserts Narrow Pulses in a 8-PWM-Cycle Frame
1	NP0.1	
0	NP0.0	

TABLE 37: PWM DATA REGISTER 1 (PWMD1) – SFR A5H

7	6	5	4
PWMD1.4	PWMD1.3	PWMD1.2	PWMD1.1

3	2	1	0
PWMD1.0	NP1.2	NP1.1	NP1.0

Bit	Mnemonic	Description
7	PWMD1.4	Contents of PWM Data Register 1 Bit 4
6	PWMD1.3	Contents of PWM Data Register 1 Bit 3
5	PWMD1.2	Contents of PWM Data Register 1 Bit 2
4	PWMD1.1	Contents of PWM Data Register 1 Bit 1
3	PWMD1.0	Contents of PWM Data Register 1 Bit 0
2	NP1.2	Inserts Narrow Pulses in a 8-PWM-Cycle Frame
1	NP1.1	
0	NP1.0	

TABLE 38: PWM DATA REGISTER 2 (PWMD2) – SFR A6H

7	6	5	4
PWMD2.4	PWMD2.3	PWMD2.2	PWMD2.1

3	2	1	0
PWMD2.0	NP2.2	NP2.1	NP2.0

Bit	Mnemonic	Description
7	PWMD2.4	Contents of PWM Data Register 2 Bit 4
6	PWMD2.3	Contents of PWM Data Register 2 Bit 3
5	PWMD2.2	Contents of PWM Data Register 2 Bit 2
4	PWMD2.1	Contents of PWM Data Register 2 Bit 1
3	PWMD2.0	Contents of PWM Data Register 2 Bit 0
2	NP2.2	
1	NP2.1	Inserts Narrow Pulses in a 8-PWM-Cycle
0	NP2.0	

TABLE 39: PWM DATA REGISTER 3 (PWMD1) – SFR A7H

7	6	5	4
PWMD3.4	PWMD3.3	PWMD3.2	PWMD3.1

3	2	1	0
PWMD3.0	NP3.2	NP3.1	NP3.0

Bit	Mnemonic	Description
7	PWMD3.4	Contents of PWM Data Register 3 Bit 4
6	PWMD3.3	Contents of PWM Data Register 3 Bit 3
5	PWMD3.2	Contents of PWM Data Register 3 Bit 2
4	PWMD3.1	Contents of PWM Data Register 3 Bit 1
3	PWMD3.0	Contents of PWM Data Register 3 Bit 0
2	NP3.2	Inserts Narrow Pulses in a 8-PWM-Cycle Frame
1	NP3.1	
0	NP3.0	

TABLE 40: PWM DATA REGISTER 4 (PWMD1) – SFR ACH

7	6	5	4
PWMD4.4	PWMD4.3	PWMD4.2	PWMD4.1

3	2	1	0
PWMD4.0	NP4.2	NP4.1	NP4.0

Bit	Mnemonic	Description
7	PWMD4.4	Contents of PWM Data Register 4 Bit 4
6	PWMD4.3	Contents of PWM Data Register 4 Bit 3
5	PWMD4.2	Contents of PWM Data Register 4 Bit 2
4	PWMD4.1	Contents of PWM Data Register 4 Bit 1
3	PWMD4.0	Contents of PWM Data Register 4 Bit 0
2	NP4.2	Inserts Narrow Pulses in a 8-PWM-Cycle Frame
1	NP4.1	
0	NP4.0	

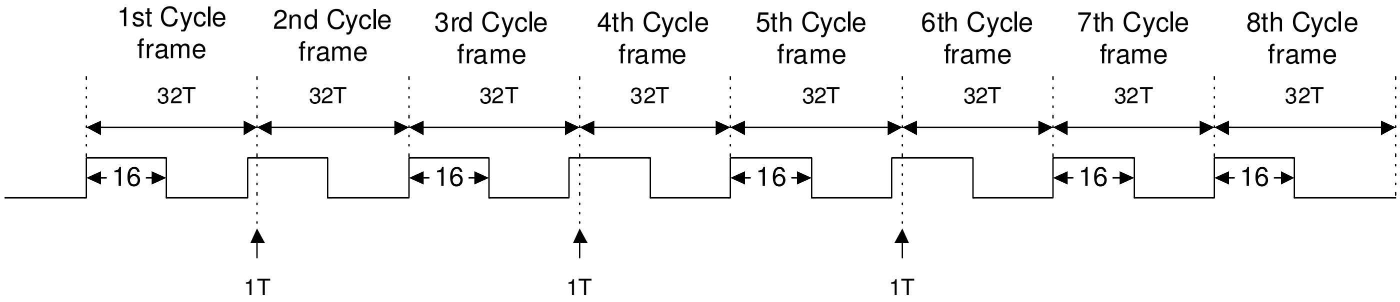
The table below shows the number of PWM cycles inserted in an 8-cycle frame when we vary the NP number.

N = NP[4:0][2:0]	Number of PWM cycles inserted in an 8-cycle frame
XX1	1
X1X	2
1XX	3

Example of PWM Timing Diagram

```
MOV PWMD0 #83H          ; PWMD04:0]=10h (=16T high, 16T low), NP02:0] = 3
MOV PWME, #08H          ; Enable P1.3 as PWM output pin
```

FIGURE 23: PWM TIMING DIAGRAM



(Narrow pulse inserted by NP0[2:0]=3)

PWM clock= 1/T= Fosc / 2^(PDIV+1)
The SPWM output cycle frame frequency = SPWM clock/32 = [Fosc/2^(PDIV+1)]/32

If Fosc = 20MHz, PDCK[1:0] of PWMC = #03H, then PWM clock = 20MHz/2^4 = 20MHz/16 = 1.25MHz. PWM output cycle frame frequency = (20MHz/2^4)/32 = 39.1 kHz.

Crystal consideration

The crystal connected to the VRS1000 oscillator input should be of a parallel type, operating in fundamental mode.

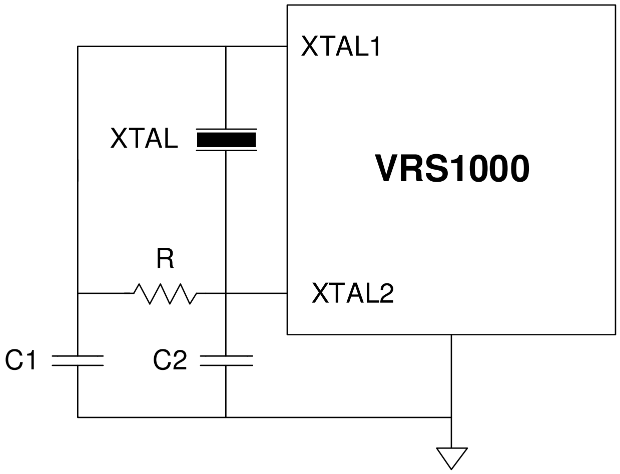
The following table shows the value of capacitors and feedback resistor that must be used at different operating frequencies.

Valid for VRS1000				
XTAL	3MHz	6MHz	9MHz	12MHz
C1	30 p	30 p	30 p	30 p
C2	30 p	30 p	30 p	30 p
R	open	open	open	open
XTAL	16MHz	25MHz	33MHz	40MHz
C1	30 pF	15 pF	10 pF	2 pF
C2	30 pF	15 pF	10 pF	2 pF
R	open	62KΩ	6.8KΩ	4.7KΩ

Note: Oscillator circuits may differ with different crystals or ceramic resonators in higher oscillation frequency.

Crystals or ceramic resonator characteristics vary from one manufacturer to the other.

The user should check the specific crystal or ceramic resonator technical literature available or contact the manufacturer to select the appropriate values for the external components.



Operating Conditions

TABLE 41: OPERATING CONDITIONS

Symbol	Description	Min.	Typ.	Max.	Unit	Remarks
TA	Operating temperature	-40	25	+85	ºC	Ambient temperature under bias
TS	Storage temperature	-55	25	155	ºC	
VCC5	Supply voltage	4.5	5.0	5.5	V	
Fosc 40	Oscillator Frequency	3.0	-	40	MHz	For 5V application

DC Characteristics

TABLE 42: DC CHARACTERISTICS

Symbol	Parameter	Valid	Min.	Max.	Unit	Test Conditions
VIL1	Input Low Voltage	Port 0,1,2,3,4,#EA	-0.5	1.0	V	VCC=5V
VIL2	Input Low Voltage	RES, XTAL1	0	0.8	V	VCC=5V
VIH1	Input High Voltage	Port 0,1,2,3,4,#EA	2.0	VCC+0.5	V	VCC=5V
VI H2	Input High Voltage	RES, XTAL1	70% VCC	VCC+0.5	V	VCC=5V
VOL1	Output Low Voltage	Port 0, ALE, #PSEN		0.45	V	IOL=3.2mA
VOL2	Output Low Voltage	Port 1,2,3,4		0.45	V	IOL=1.6mA
VOH1	Output High Voltage	Port 0	2.4		V	IOH=-800uA
			90%VCC		V	IOH=-80uA
VOH2	Output High Voltage	Port 1,2,3,4,ALE,#PSEN	2.4		V	IOH=-60uA
			90% VCC		V	IOH=-10uA
IIL	Logical 0 Input Current	Port 1,2,3,4		-75	uA	Vin=0.45V
ITL	Logical Transition Current	Port 1,2,3,4		-650	uA	Vin=2.OV
ILI	Input Leakage Current	Port 0, #EA		±10	uA	0.45V<Vin<VCC
R RES	Reset Pull-down Resistance	RES	50	300	Kohm	
C^{-}10	Pin Capacitance			10	pF	Fre=1 MHz, Ta=25°C
ICC	Power Supply Current	VDD		20	mA	Active mode, 40MHz
				15	mA	Active mode 25MHz
				10	mA	Active mode 16MHz
				10	mA	Idle mode, 40MHz
				7.5	mA	Idle mode 25MHz
				6	mA	Idle mode, 16MHz
				150	uA	Power down mode

FIGURE 24: ICC ACTIVE MODE TEST CIRCUIT

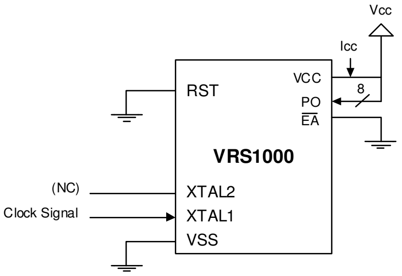
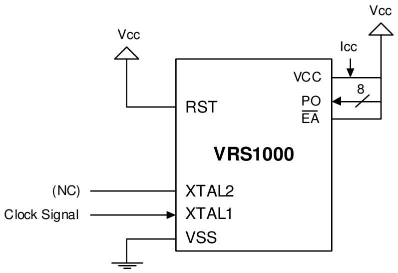


FIGURE 25: ICC IDLE MODE TEST CIRCUIT



AC Characteristics

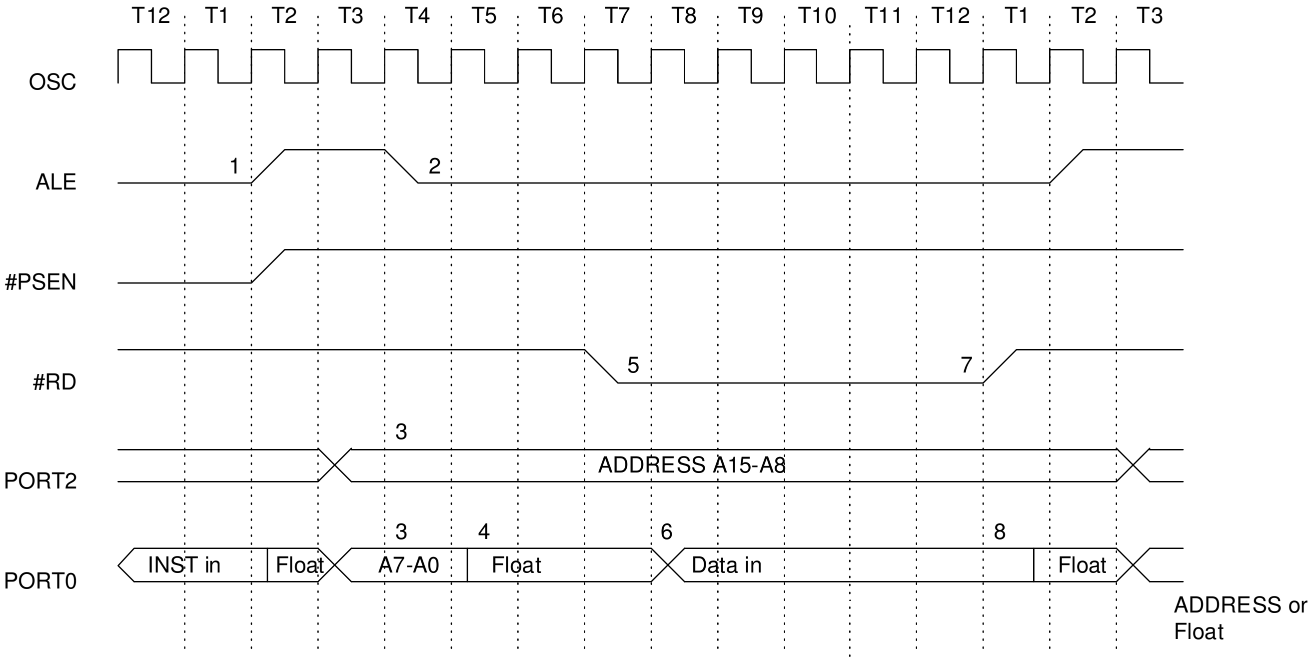
TABLE 43: AC CHARACTERISTICS

Symbol	Parameter	Valid Cycle	Fosc 16			Variable Fosc			Unit
			Min.	Type	Max.	Min.	Type	Max.	
T LHLL	ALE Pulse Width	RD/WRT	115			2xT - 10			nS
T AVLL	Address Valid to ALE Low	RD/WRT	43			T - 20			nS
T LLAX	Address Hold after ALE Low	RD/WRT	53			T - 10			nS
T LLIV	ALE Low to Valid Instruction In	RD			240			4xT - 10	nS
T LLPL	ALE Low to #PSEN low	RD	53			T - 10			nS
T PLPH	#PSEN Pulse Width	RD	173			3xT - 15			nS
T PLIV	#PSEN Low to Valid Instruction In	RD			177			3xT -10	nS
T PXIX	Instruction Hold after #PSEN	RD	0			0			nS
T PXIZ	Instruction Float after #PSEN	RD			87			T + 25	nS
T AVI V	Address to Valid Instruction In	RD			292			5xT - 20	nS
T PLAZ	#PSEN Low to Address Float	RD			10			10	nS
T RLRH	#RD Pulse Width	RD	365			6xT - 10			nS
T WLWH	#WR Pulse Width	WRT	365			6xT - 10			nS
T RLDV	#RD Low to Valid Data In	RD			302			5xT - 10	nS
T RHDX	Data Hold after #RD	RD	0			0			nS
T RHDZ	Data Float after #RD	RD			145			2xT + 20	nS
T LLDV	ALE Low to Valid Data In	RD			590			8xT - 10	nS
T AVDV	Address to Valid Data In	RD			542			9xT - 20	nS
T LLYL	ALE low to #WR High or #RD Low	RD/WRT	178		197	3xT - 10		3xT + 10	nS
T AVYL	Address Valid to #WR or #RD Low	RD/WRT	230			4xT - 20			nS
T QVWH	Data Valid to #WR High	WRT	403			7xT - 35			nS
T QVWX	Data Valid to #WR Transition	WRT	38			T - 25			nS
T WHQX	Data Hold after #WR	WRT	73			T + 10			nS
T RLAZ	#RD Low to Address Float	RD						5	nS
T YALH	#W R or #RD High to ALE High	RD/WRT	53		72	T -10		T+10	nS
T CHCL	Clock Fall Time								nS
T CLCX	Clock Low Time								nS
T CLCH	Clock Rise Time								nS
T CHCX	Clock High Time								nS
T , T CLCL	Clock Period		63				1/fosc		nS

Data Memory Read Cycle Timing

The following timing diagram shows what occurs at each signal during a Data Memory Read Cycle.

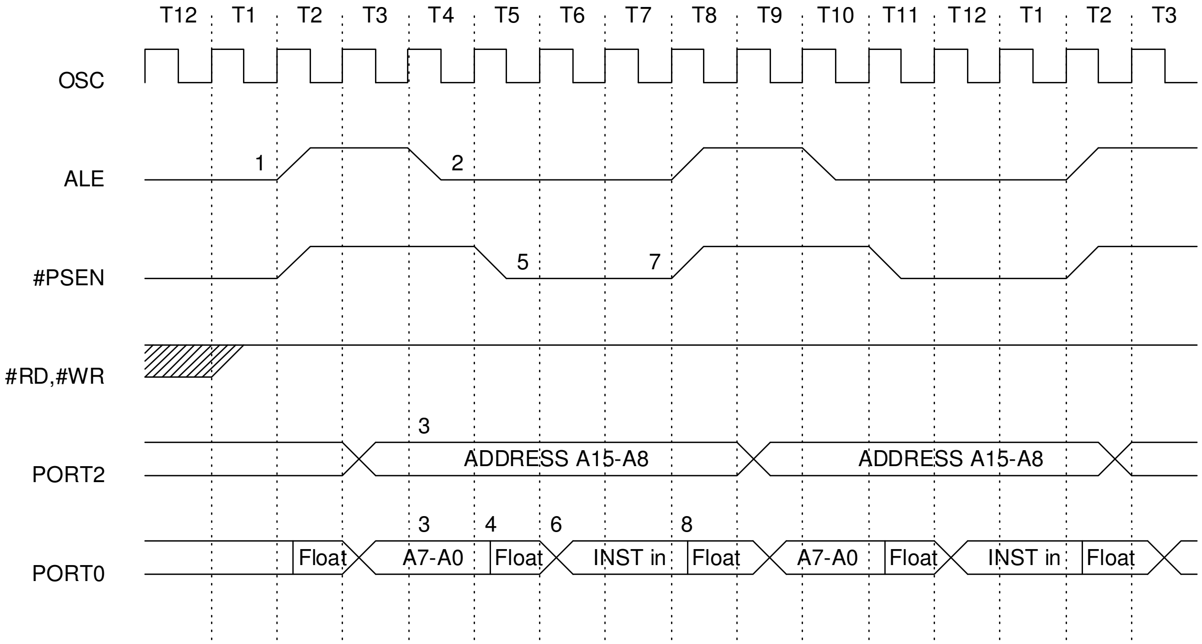
FIGURE 26: DATA MEMORY READ CYCLE TIMING



Program Memory Read Cycle Timing

The following timing diagram shows what occurs at each signal during a Program Memory Read Cycle.

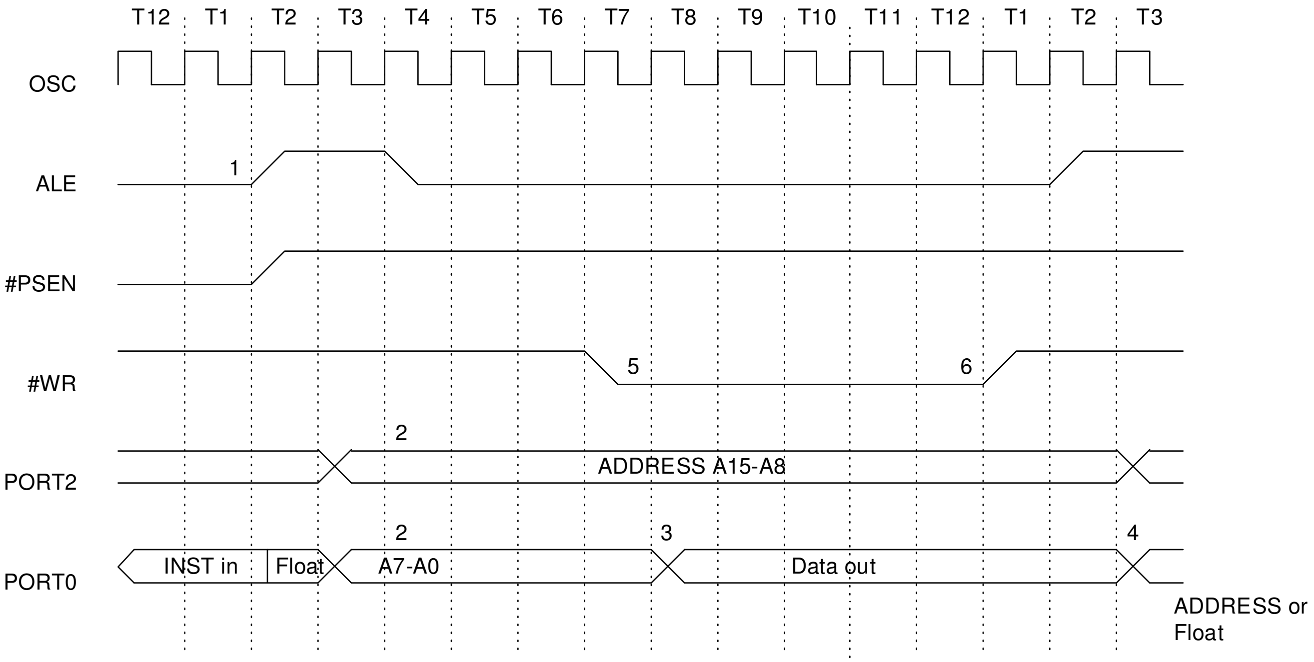
FIGURE 27: PROGRAM MEMORY READ CYCLE



Data Memory Write Cycle Timing

The following timing diagram shows what occurs at each signal during a Data Memory Write Cycle.

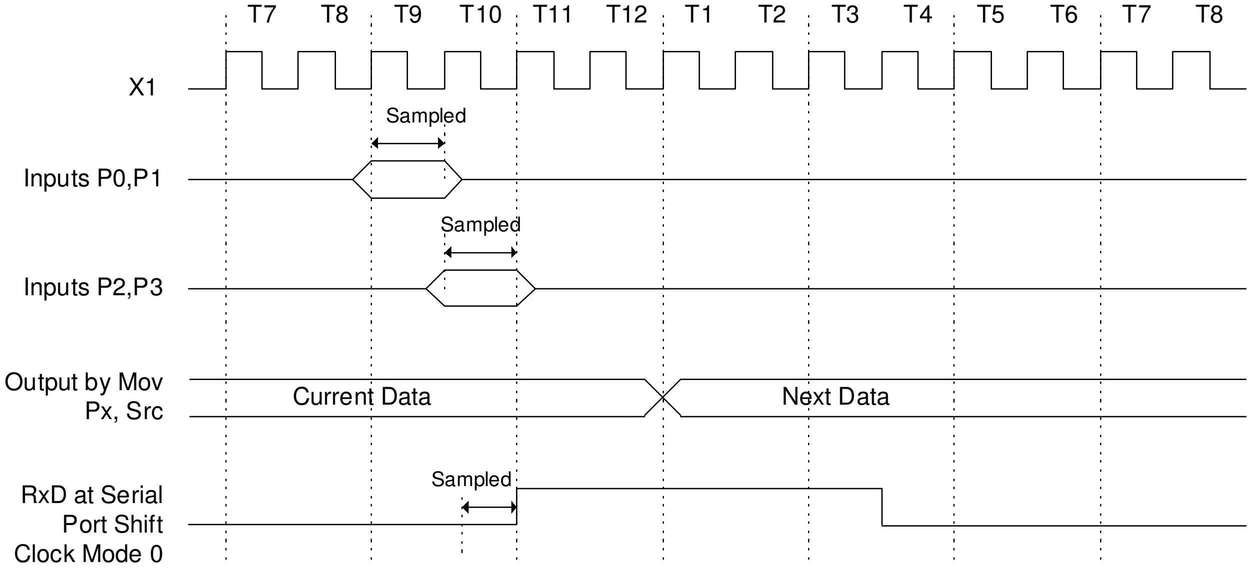
FIGURE 28: DATA MEMORY WRITE CYCLE TIMING



I/O Ports Timing

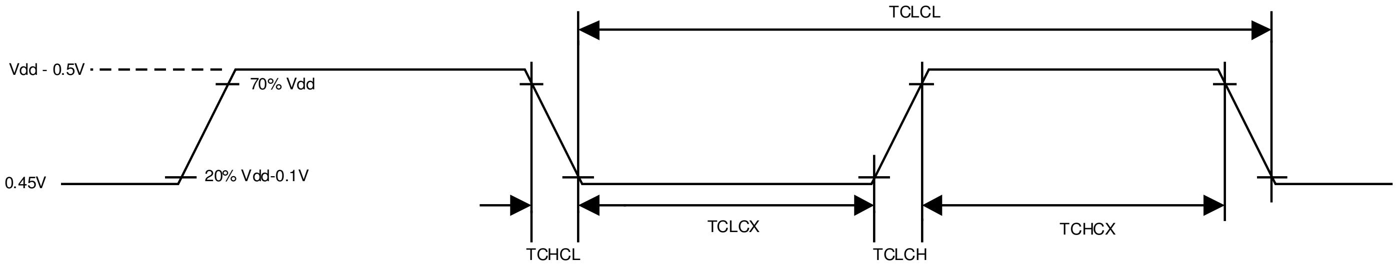
The following timing diagram shows what occurs during I/O Port Timing.

FIGURE 29: I/O PORTS TIMING



Timing Requirement of the External Clock (VSS = 0v is assumed)

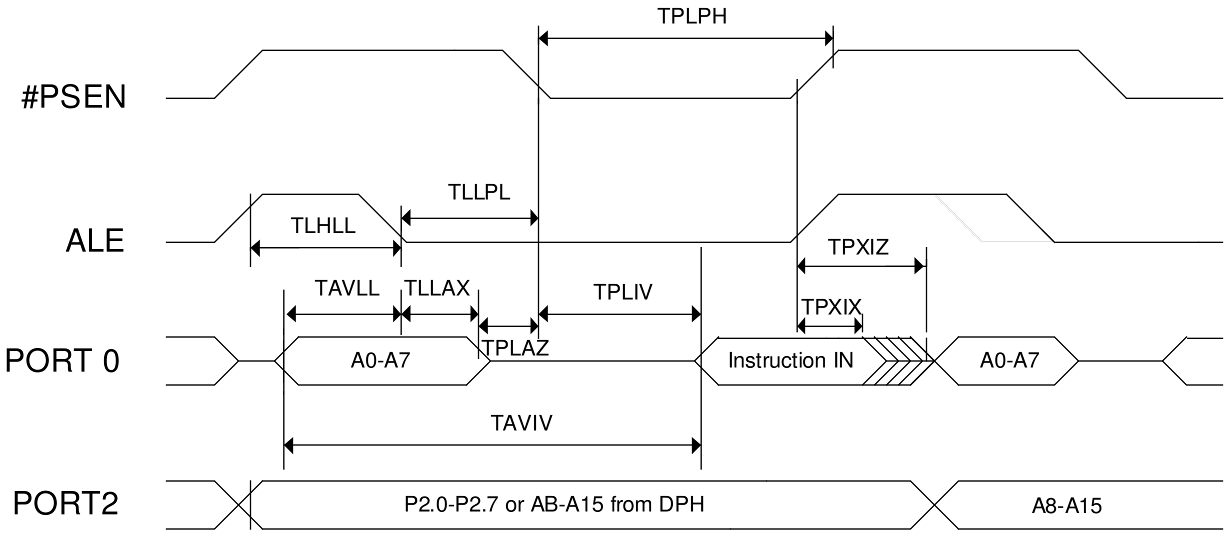
FIGURE 30: TIMING REQUIREMENT OF EXTERNAL CLOCK (VSS= 0.0V IS ASSUMED)



External Program Memory Read Cycle

The following timing diagram shows what occurs at each signal during an External Program Memory Read Cycle.

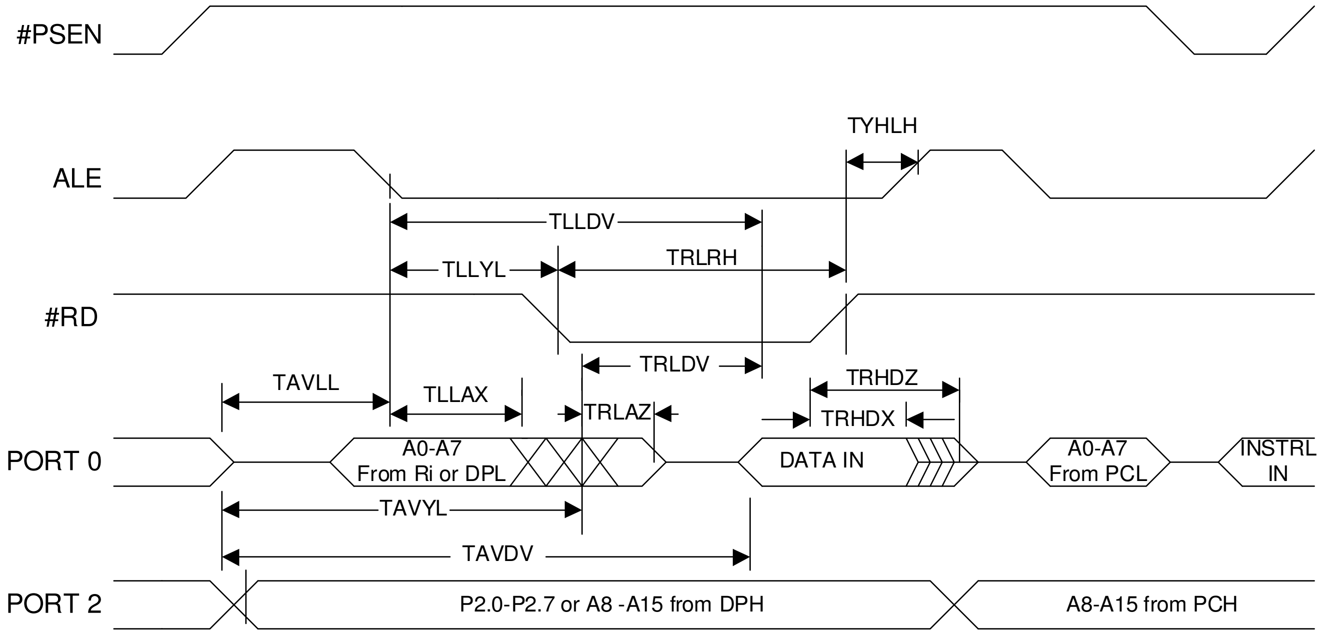
FIGURE 31: EXTERNAL PROGRAM MEMORY READ CYCLE



External Data Memory Read Cycle

The following timing diagram shows what occurs at each signal during an External Data Memory Read Cycle.

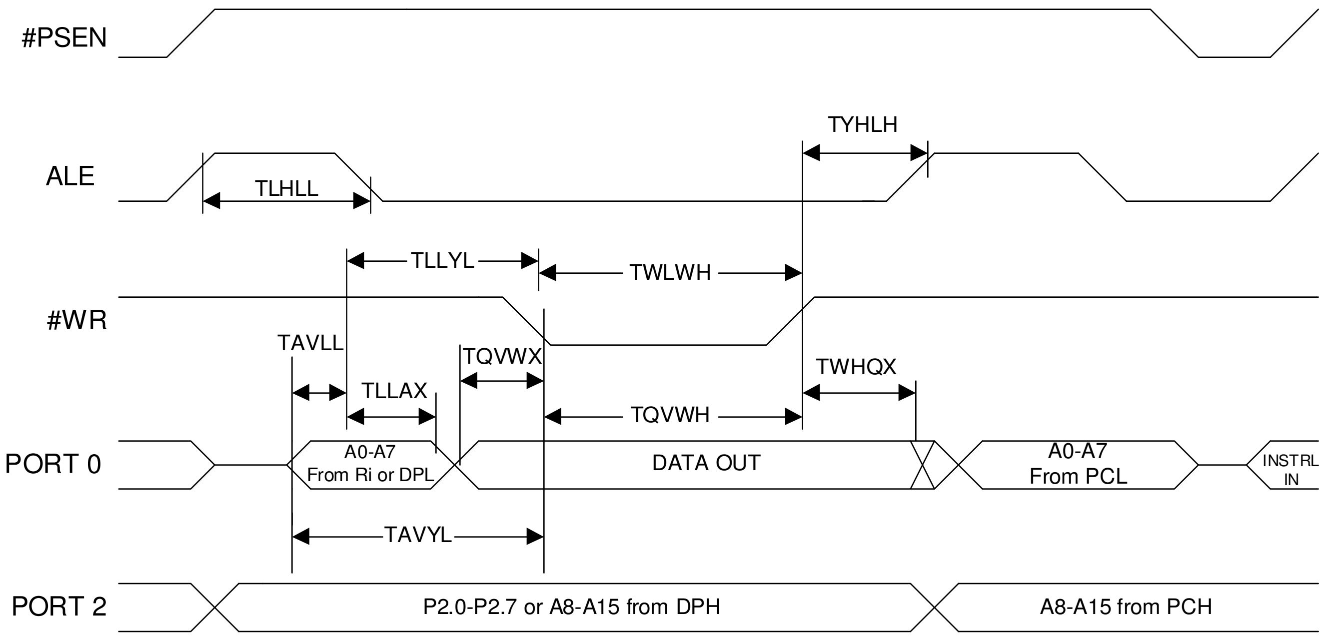
FIGURE 32: EXTERNAL DATA MEMORY READ CYCLE



External Data Memory Write Cycle

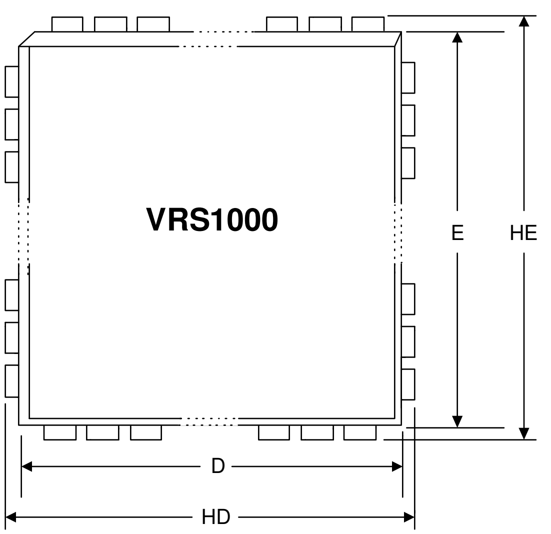
The following timing diagram shows what occurs at each signal during an External Data Memory Write Cycle.

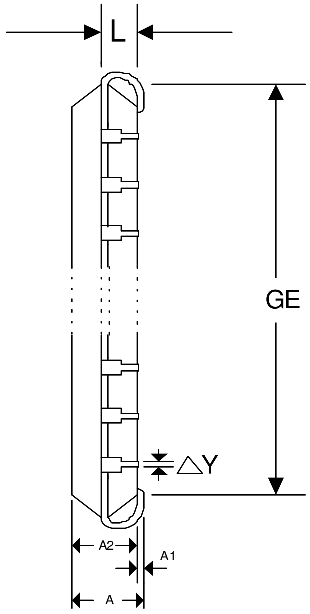
FIGURE 33: EXTERNAL DATA MEMORY WRITE CYCLE

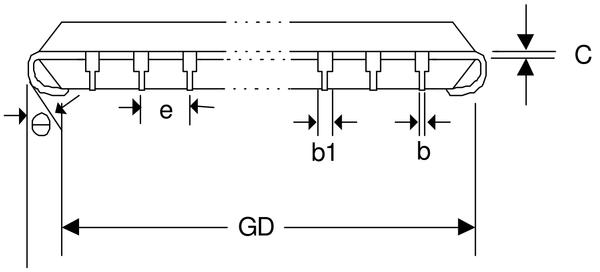


.

Plastic Chip Carrier (PLCC)





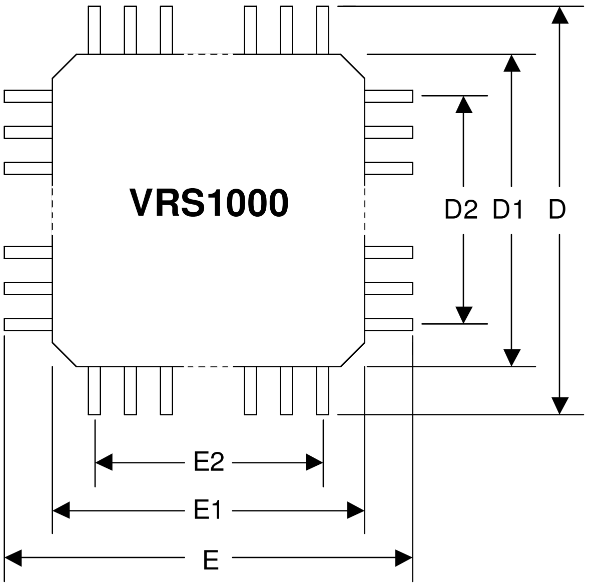


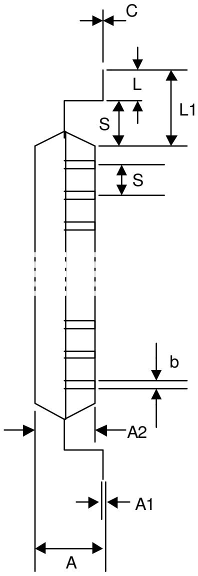
Note:
1. Dimensions D & E do not include interlead Flash.
2. Dimension B1 does not include dambar protrusion/intrusion.
3. Controlling dimension: Inch
4. General appearance spec should be based on final visual inspection spec.

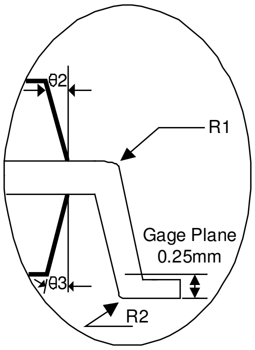
TABLE 44: DIMENSIONS OF PLCC-44 CHIP CARRIER

Symbol	Dimension in inch	Dimension in mm
	Minimal/Maximal	Minimal/Maximal
A	-/0.185	-/4.70
Al	0.020/-	0.51/
A2	0.145/0.155	3.68/3.94
bl	0.026/0.032	0.66/0.81
b	0.016/0.022	0.41/0.56
C	0.008/0.014	0.20/0.36
D	0.648/0.658	16.46/16.71
E	0.648/0.658	16.46/16.71
e	0.050 BSC	1.27 BSC
GD	0.590/0.630	14.99/16.00
GE	0.590/0.630	14.99/16.00
HD	0.680/0.700	17.27/17.78
HE	0.680/0.700	17.27/17.78
L	0.090/0.110	2.29/2.79
θ	-/0.004	-/0.10
Δy	/	/

Plastic Quad Flat Package (QFP)







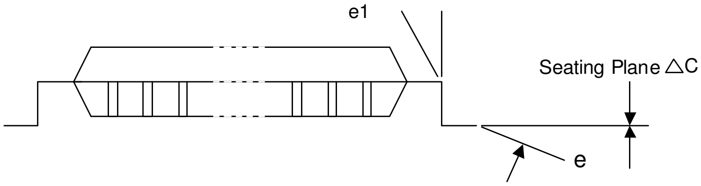


TABLE 45: DIMENSIONS OF QFP-44 CHIP CARRIER

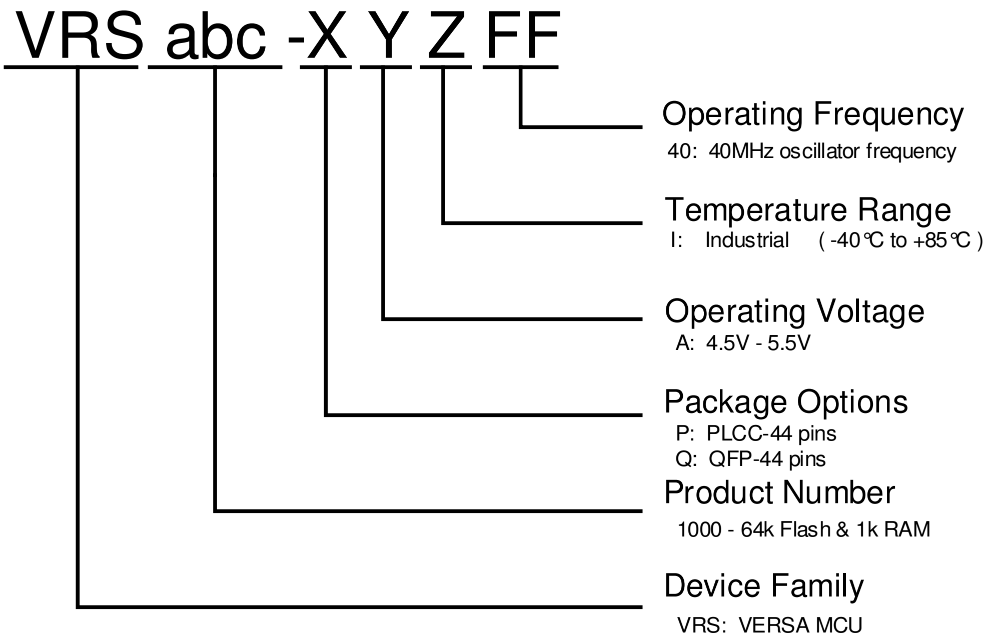
Symbol	Dimension in in.	Dimension in mm
	Minimal/Maximal	Minimal/Maximal
A	-/0.100	-/2.55
Al	0.006/0.014	0.15/0.35
A2	0.071 / 0.087	1.80/2.20
b	0.012/0.018	0.30/0.45
c	0.004 / 0.009	0.09/0.20
D	0.520 BSC	13.20 BSC
D1	0.394 BSC	10.00 BSC
D2	0.315	8.00
E	0.520 BSC	13.20 BSC
E1	0.394 BSC	10.00 BSC
E2	0.315	8.00
e	0.031 BSC	0.80 BSC
L	0.029 / 0.041	0.73/1.03
L1	0.063	1.60
R1	0.005/-	0.13/-
R2	0.005/0.012	0.13/0.30
S	0.008/-	0.20/-
0	0˚/7˚	as left
θ 1	0˚/ -	as left
θ 2	10˚ REF	as left
θ 3	7˚ REF	as left
∆C	0.004	0.10

Note:

1. Dimensions D1 and E1 do not include mold protrusion.
2. Allowance protrusion is 0.25mm per side.
3. Dimensions D1 and E1 do not include mold mismatch and are determined datum plane.
4. Dimension b does not include dambar protrusion.
5. Allowance dambar protrusion shall be 0.08 mm total in excess of the b dimension at maximum material condition. Dambar cannot be located on the lower radius of the lead foot.

Ordering Information

Device Number Structure



VRS1000 Ordering Options

Device Number	Flash Size	RAM Size	Package Option	Voltage	Temperature	Frequency
VRS1000-PAI40	64k	1k	PLCC-44	4.5V to 5.5V	-40°C to +85°C	40MHz
VRS1000-QAI40	64k	1k	QFP-44	4.5V to 5.5V	-40°C to +85°C	40MHz

Disclaimers

Right to make change - Goal Semiconductor reserves the right to make changes to its products - including circuitry, software and services - without notice at any time. Customers should obtain the most current and relevant information before placing orders.
Use in applications - Goal Semiconductor assumes no responsibility or liability for the use of any of its products, and conveys no license or title under any patent, copyright or mask work right to these products and makes no representations or warranties that these products are free from patent, copyright or mask work right infringement unless otherwise specified. Customers are responsible for product design and applications using Goal Semiconductor parts. Goal Semiconductor assumes no liability for applications assistance or customer product design.
Life support – Goal Semiconductor products are not designed for use in life support systems or devices. Goal Semiconductor customers using or selling Goal products for use in such applications do so at their own risk and agree to fully indemnify Goal Semiconductor for any damages resulting from such applications.