Versa 8051 3.3V MCU with 64KB of IAP/ISP Flash

Overview

The VRS51L1050 is based on the standard 8051 microcontroller family architecture and is a pin compatible drop-in replacement for most 8051 MCUs.

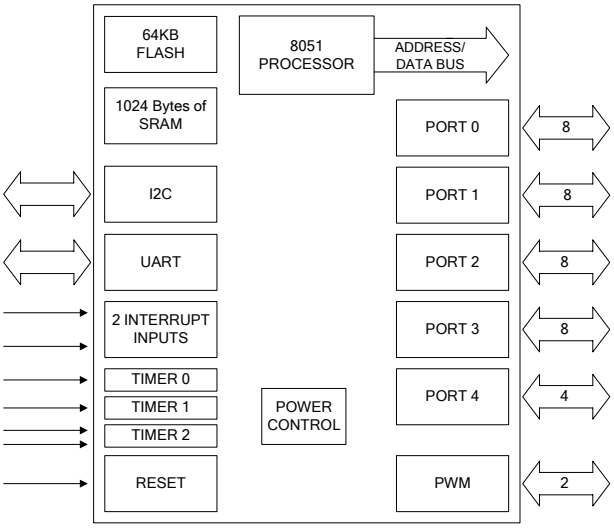
The VRS51L1050 is designed for applications that require a large amount of program/data memory with non-volatile data storage and/or code/field based firmware upgrade capability coupled with comprehensive peripheral support. It features 64KB of In-System/In-Application Programmable Flash memory, 1KB of SRAM, an I²C-compatible interface, 2 PWM output channels, a UART, three 16-bit timers and the ability to exit the power down mode upon assertion of an external interrupt (INT0 / INT1).

Ideal for battery-powered applications, the VRS51L1050 registers and I/Os maintain their current value in power down mode while the oscillator is disabled, enabling the supply current to drop below 20uA.

The VRS51L1050 is available with firmware that enables In-System Programming (firmware based boot-loader) of the Flash memory via the UART interface (ISPVx version). General Flash memory programming is supported by device programmers available from Ramtron or other 3rd party suppliers.

The VRS51L1050 is available in PLCC-44, QFP-44 and DIP-40 packages and functions over the commercial temperature range.

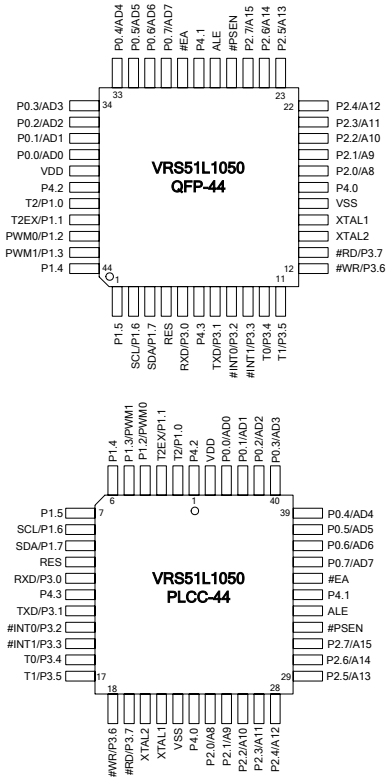
FIGURE 1: VRS51L1050 FUNCTIONAL DIAGRAM



Feature Set

- 8051/8052 pin compatible
- 64KB on-chip Flash memory
- In-System/In-Application Flash Programming (ISP/IAP)
- On-chip Charge Pump for Flash Programming
- 1024 Bytes on-chip Data SRAM
- Four 8-bit I/O Ports, one 4-bit I/O Port
- 2 PWM Outputs on P1.2 to P1.3 (8/5-bit resolution)
- 1 Full Duplex UART Serial Port
- I²C-compatible Interface
- Three 16-bit Timers/Counters
- Bit Operation Instruction
- 8-bit Unsigned Division and Multiply
- BCD Arithmetic
- Direct and Indirect Addressing
- 7 Interrupt Sources and 2 Levels of Interrupt Priority
- Power saving modes
- Wakeup from Power Down by Ext. Interrupt or Reset
- Code protection function
- Low EMI (inhibit ALE)
- Commercial Temperature Range (0°C to +70°C)
- 3.3V Operating Voltage

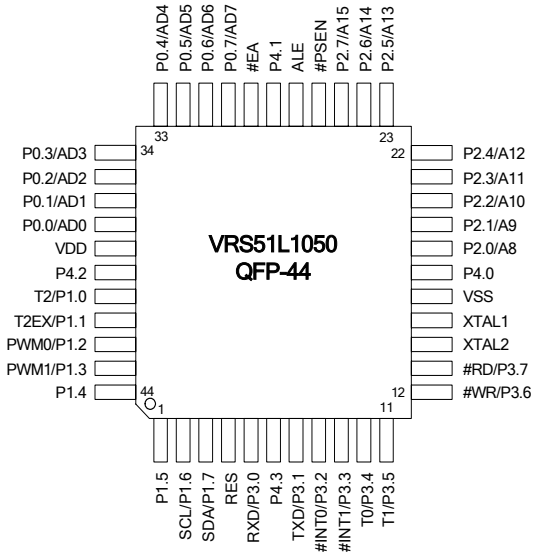
FIGURE 2: VRS51L1050 QFP-44 AND PLCC-44 PIN OUT DIAGRAMS

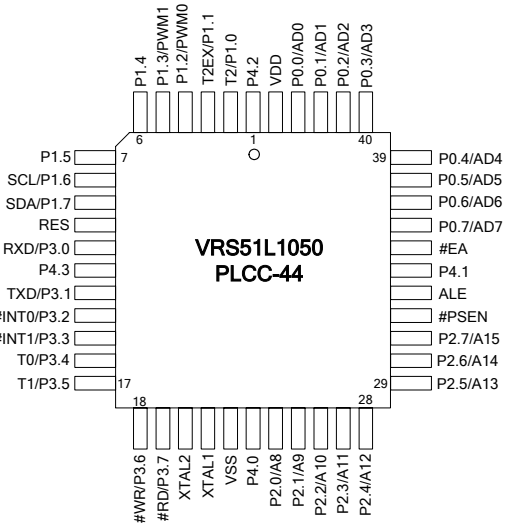


Pin Descriptions for QFP-44/PLCC-44

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

QFP - 44	PLCC - 44	Name	I/O	Function
1	7	P1.5	I/O	Bit 5 of Port 1
2	8	SCL	O	I²C SCL
		P1.6	I/O	Bit 6 of Port 1
3	9	SDA	O	I²C SDA
		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	PWM0		PWM Channel 0
		P1.2	I/O	Bit 2 of Port 1
43	5	PWM1	O	PWM Channel 1
		P1.3	I/O	Bit 3 of Port 1
44	6	P1.4	I/O	Bit 4 of Port 1

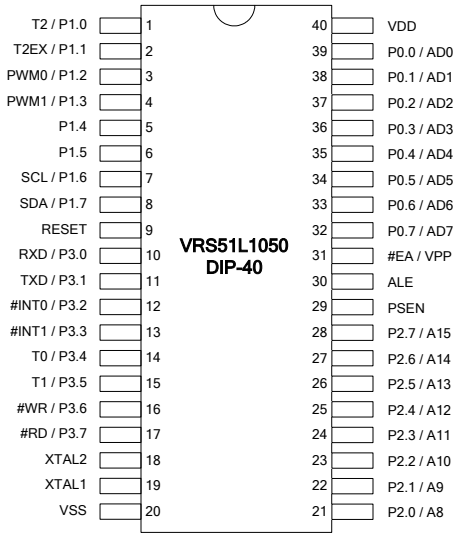




VRS51L1050 DIP-40 Pin Descriptions

TABLE 2: VRS51L1050 PIN DESCRIPTIONS FOR DIP40 PACKAGE

DIP40	Name	I/O	Function
1	T2	I	Timer 2 Clock Out
	P1.0	I/O	Bit 0 of Port 1
2	T2EX	I	Timer 2 Control
	P1.1	I/O	Bit 1 of Port 1
	PWM0		PWM Channel 0
3	P1.2	I/O	Bit 2 of Port 1
4	PWM1	O	PWM Channel 1
	P1.3	I/O	Bit 3 of Port 1
5	P1.4	I/O	Bit 4 of Port 1
6	P1.5	I/O	Bit 5 of Port 1
7	SCL	O	I²C SCL
	P1.6	I/O	Bit 6 of Port 1
8	SDA	O	I²C SDA
	P1.7	I/O	Bit 7 of Port 1
9	RESET	I	Reset
10	RXD	I	Receive Data
	P3.0	I/O	Bit 0 of Port 3
11	TXD	O	Transmit Data &
	P3.1	I/O	Bit 1 of Port 3
12	#INT0	I	External Interrupt 0
	P3.2	I/O	Bit 2 of Port 3
13	#INT1	I	External Interrupt 1
	P3.3	I/O	Bit 3 of Port 3
14	T0	I	Timer 0
	P3.4	I/O	Bit 4 of Port 3
15	T1	I	Timer 1 & 3
	P3.5	I/O	Bit 5 of Port
16	#WR	O	Ext. Memory Write
	P3.6	I/O	Bit 6 of Port 3
17	#RD	O	Ext. Memory Read
	P3.7	I/O	Bit 7 of Port 3
18	XTAL2	O	Oscillator/Crystal Output
19	XTAL1	I	Oscillator/Crystal In
20	VSS	-	Ground



DIP40	Name	I/O	Function
21	P2.0	I/O	Bit 0 of Port 2
	A8	O	Bit 8 of External Memory Address
22	P2.1	I/O	Bit 1 of Port 2
	A9	O	Bit 9 of External Memory Address
23	P2.2	I/O	Bit 2 of Port 2
	A10	O	Bit 10 of External Memory Address
24	P2.3	I/O	Bit 3 of Port 2 &
	A11	O	Bit 11 of External Memory Address
25	P2.4	I/O	Bit 4 of Port 2
	A12	O	Bit 12 of External Memory Address
26	P2.5	I/O	Bit 5 of Port 2
	A13	O	Bit 13 of External Memory Address
27	P2.6	I/O	Bit 6 of Port 2
	A14	O	Bit 14 of External Memory Address
28	P2.7	I/O	Bit 7 of Port 2
	A15	O	Bit 15 of External Memory Address
29	#PSEN	O	Program Store Enable
30	ALE	O	Address Latch Enable
31	#EA / VPP	I	External Access Flash programming voltage input
32	P0.7	I/O	Bit 7 Of Port 0
	AD7	I/O	Data/Address Bit 7 of External Memory
33	P0.6	I/O	Bit 6 of Port 0
	AD6	I/O	Data/Address Bit 6 of External Memory
34	P0.5	I/O	Bit 5 of Port 0
	AD5	I/O	Data/Address Bit 5 of External Memory
35	P0.4	I/O	Bit 4 of Port 0
	AD4	I/O	Data/Address Bit 4 of External Memory
36	P0.3	I/O	Bit 3 Of Port 0
	AD3	I/O	Data/Address Bit 3 of External Memory
37	P0.2	I/O	Bit 2 of Port 0
	AD2	I/O	Data/Address Bit 2 of External Memory
38	P0. 1	I/O	Bit 1 of Port 0 & Data
	AD1	I/O	Address Bit 1 of External Memory
39	P0.0	I/O	Bit 0 Of Port 0 & Data
	AD0	I/O	Address Bit 0 of External Memory
40	VDD	-	Supply input

Instruction Set

The following table describes the instruction set of the VRS51L1050. The instructions are function and binary code compatible with industry standard 8051s.

TABLE 3: 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 4: VRS51L1050 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 VRS51L1050 special function registers.

TABLE 5: 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	-	-	-	-	-	-	-	-	1111 1111b
SP	81h	-	-	-	-	-	-	-	-	0000 0111b
DPL	82h	-	-	-	-	-	-	-	-	0000 0000b
DPH	83h	-	-	-	-	-	-	-	-	0000 0000b
RCON	85h	-	-	-	-	-	-	RAMS1	RAMS0	0000 0000b
reserved	86h	-	-	-	-	-	-	-	-	0000 0001b
PCON	87h	SMOD	-	-	-	GF1	GF0	PDOWN	IDLE	0000 0000b
TCON	88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0010b
TMOD	89h	GATE1	C/T1	M1.1	M0.1	GATE0	C/T0	M1.0	M0.0	0000 0000b
TL0	8Ah	-	-	-	-	-	-	-	-	0000 0000b
TL1	8Bh	-	-	-	-	-	-	-	-	0000 0000b
TH0	8Ch	-	-	-	-	-	-	-	-	0000 0000b
TH1	8Dh	-	-	-	-	-	-	-	-	0000 0000b
P1	90h	-	-	-	-	-	-	-	-	1111 1111b
SCON	98h	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000b
SBUF	99h	-	-	-	-	-	-	-	-	0111 1111b
I2CPWME	9Bh	SDAE	SCLE	-	-	PWM1E	PWM1E	-	-	0000 0000b
P2	A0h	-	-	-	-	-	-	-	-	1111 1111b
IEN1	A8h	EA	-	ET2	ES	ET1	EX1	ET0	EX0	0000 0000b
IE1	A9h	-	-	-	-	-	-	EI2C	-	0000 0000b
IF1	AAh	-	-	-	-	-	-	I2CIF		0000 0000b
P3	B0h	-	-	-	-	-	-	-	-	1111 1011b
PWMD0	B3h	PWMD0.4	PWMD0.3	PWMD0.2	PWMD0.1	PWMD0.0	NP0.2	NP0.1	NP0.0	0000 0000b
PWMD1	B4h	PWMD1.4	PWMD1.3	PWMD1.2	PWMD1.1	PWMD1.0	NP1.2	NP1.1	NP1.0	0000 0000b
IP	B8h	-	-	PT2	PS	PT1	PX1	PT0	PX0	0000 0000b
IP1	B9h	-	-	-	-	-	-	PI2C	-	0000 0000b
SYSCON	BFh	-	-	-	PDWAKEUP	-	IAPE	XRAME	ALEI	0000 1010b
I2CSTATUS	C0h	I2CRXIF	I2CTXIF	I2CTXFAIL	I2CNOACKIF		I2CRXACK	I2CMASTER	I2CTXACK	0000 0000b
I2CADDR	C1h	I2CADDR7	I2CADDR6	I2CADDR5	I2CADDR4	I2CADDR3	I2CADDR2	I2CADDR1	MSBCOMP	1010 0000h
I2CCTRL1	C2h	I2CEN	-	-	-	I2CBUSY	I2CCK2	I2CCK1	I2CCK0	0000 0001b
I2CCTRL2	C3h	MATCH	SLAVERW	-	-	RESTART	-	-	MASTERRW	0000 0000h
I2CTX	C4h	-	-	-	-	-	-	-	-	0000 0000h
I2CRX	C5h	-	-	-	-	-	-	-	-	0000 0000h
T2CON	C8h	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	0000 0000b
T2MOD	C9h	-	-	-	-	-	-	T2OE	CDOWN	Xxxx xx00h
RCAP2L	CAh	-	-	-	-	-	-	-	-	0000 0000b
RCAP2H	CBh	-	-	-	-	-	-	-	-	0000 0000b
TL2	CCh	-	-	-	-	-	-	-	-	0000 0000b
TH2	CDh									0000 0000b
PSW	D0h	CY	AC	F0	RS1	RS0	OV	-	P	0000 0001b
PWMCTRL0	D3h						5BITE	PWMCK1	PWMCK0	0000 0000b
PWMCTRL1	D4h						5BITE	PWMCK1	PWMCK0	0000 0000b
P4	D8h	-	-	-	-	P4.3	P4.2	P4.1	P4.0	****1111b
ACC	E0h	-	-	-	-	-	-	-	-	
B	F0h	-	-	-	-	-	-	-	-	0000 0000b
IAPFADHI	F4h	FA15	FA14	FA13	FA12	FA11	FA10	FA9	FA8	0000 0000b
IAPFADLO	F5h	FA7	FA6	FA5	FA4	FA3	FA2	FA1	FA0	0000 0000b
IAPFDATA	F6h	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0	0000 0000b
IAPFCTRL	F7h	IAPSTART						**IAPFCT**1	**IAPFCT**0	0000 0000b

VRS51L1050 Program Memory

The VRS51L1050 includes 64KB of on-chip Flash memory that can be used as program memory or as general non-volatile data storage memory using the In-Application Programming (IAP) feature.

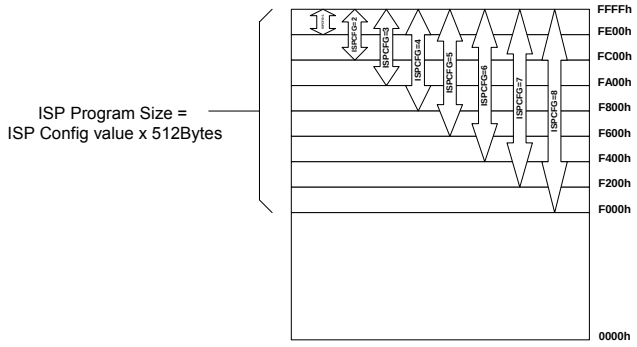
ISP Boot Program Memory Zone

The upper portion of the VRS51L1050 Flash memory can be reserved to store an ISP (In-System Programmable) boot loader program.

This boot program can be used to program the Flash memory via the serial interface (or any other method) with the VRS51L1050's In-Application Programming feature. This allows the processor to load the program from an external device or system and program it into the Flash memory (see the **VRS51L1050 IAP feature** section).

The size of the memory block reserved for the ISP boot loader program (when activated) is adjustable from 512 bytes up to 4KB in increments of 512 bytes.

FIGURE3: VRS51L1050-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 Ramtron's VERSAMCU-PPR or a commercial programmer that supports the VRS51L1050). The Flash memory reserved for the ISP program is defined by the parallel programmer software at the moment the device is programmed.

When programming the ISP boot program into the VRS51L1050, the "lock bit" option should be activated in order to:

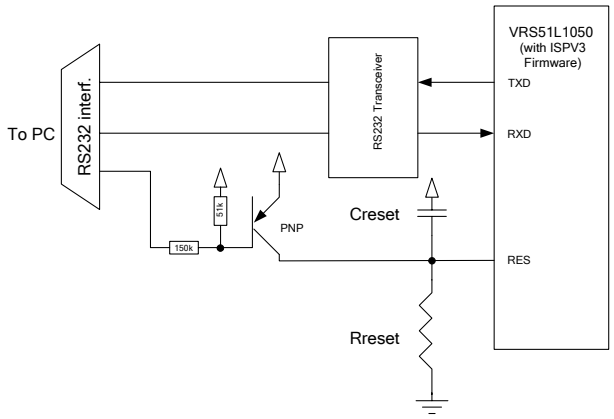
- Protect the ISP Flash memory zone from being inadvertently erased (this can happen when the Flash Erase operations are performed under the control of the ISP boot program),
- Prevent the VRS51L1050 Flash from being read back using a parallel programmer.

If an erase operation is performed using a parallel programmer, the entire Flash memory, including the ISP boot program memory zone, will be erased.

VRS51L1050 ISPV3 Firmware Boot Program

An ISP boot loader program is available for the VRS51L1050 (ISPVx Firmware, x = revision, see Ramtron web site for latest revision). The ISPVx firmware enables In-System-Programming of the VRS51L1050 on the final application PCB using the device's UART interface. See the following figure for a hardware configuration example (other configurations are also possible).

FIGURE 4: VRS51L1050 INTERFACE FOR IN-SYSTEM PROGRAMMING



The VRS51L1050 is available with or without the ISPVx boot loader firmware (see ordering information on page 50). The ISPVx boot loader firmware can also be programmed into the VRS51L1050 by the user. Source code is included with Ramtron's Windows™-based Versa Ware ISP application software, which allows communication with the ISPVx firmware.

Visit the Ramtron web site to download the Versa Ware ISP software. For more information on the ISPVx firmware, consult the "Versa Ware ISP - VRS51L1050 ISPVx User Guide.pdf", also available on the Ramtron web site.

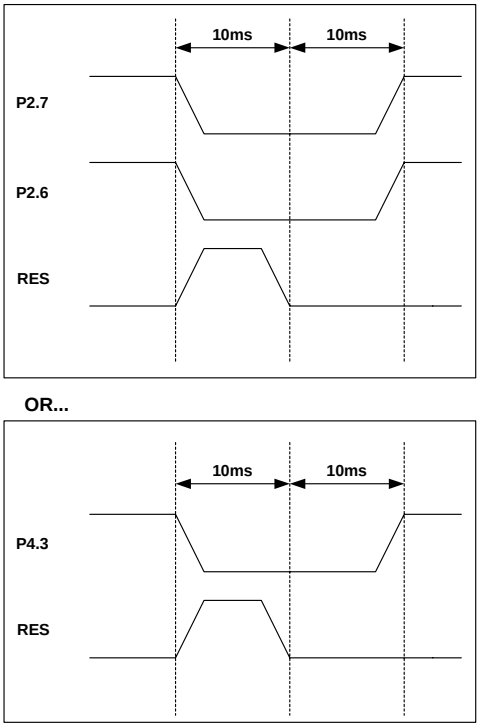
ISP Program Start Conditions

Setting the ISP page configuration to a value other than 0 will cause the processor to 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).

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.

An alternate way to force the VRS51C1050 to jump to the ISP boot program is to maintain pins P2.6 and P2.7 or pin P4.3 at a low logic level during a hardware reset, as shown in the diagram below:

FIGURE 5: VRS51C1050 ALTERNATE ISP BOOT PROGRAM ACCESS



The ISP boot program can also be accessed via the LJMP instruction. When the ISP page configuration is set to 0 while the device is being programmed with a parallel programmer, the ISP boot feature will be disabled.

VRS51L1050 IAP feature

The VRS51L1050 IAP feature refers to the processor's ability to self-program the Flash memory from within the user program. Five SFR registers control the IAP operation. The description of these registers is provided in the following sections.

System Control Register

By default, upon reset the IAP feature of the VRS51L1050 is deactivated. The IAPE bit of the SYSCON register is used to enable (and disable) the VRS51L1050 IAP function.

TABLE 6: SYSTEM CONTROL REGISTER (SYSCON) – SFR BFH

7	6	5	4	3	2	1	0
			PDWAKEUP		IAPE	XRAME	ALEI

Bit	Mnemonic	Description
7	Unused	-
6	Unused	-
5	Unused	-
4	PDWAKEUP	Power down wakeup from INT0 / INT1 0 = Deactivated 1 = Device can wakeup from power down from external interrupt
3	Unused	-
2	IAPE	IAP function enable bit 0 = IAP function is deactivated 1 = IAP function is activated
1	XRAME	768 byte on-chip enable bit 0 = Enabled 1 = Disabled
0	ALEI	ALE output inhibit bit, used to reduce EMI. 0 = ALE pin is active 1 = ALE is inhibited

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 7:IAP FLASH ADDRESS HIGH - SFR F4H

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

TABLE 8: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 9:IAP FLASH DATA REGISTER - SFR F6H

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

IAP Flash Control Register

The VRS51L1050 IAP function operation is controlled by the IAP Flash control register, IAPFCTRL. Setting the IAPSTART bit to 1 starts the execution of the IAP command specified by the IAPFCT[1:0] bits of the IAP Flash control register.

TABLE 10: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 subsystem handles four different functions. Which are controlled by the IAPFCT bits as follows:

TABLE 11: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.

To set the IAPSTART bit to 1, the following operation sequence must be performed:

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

Once the start bit is set to 1, the IAP subsystem will read the contents of the IAP Flash address and data registers and hold the VRS51L1050 program counter at its current value until the IAP operation is complete. When it is complete, the IAPSTART bit will be cleared and the program will continue executing.

IAP Byte Program Function

The IAP byte program function is used to program a byte into a specified Flash memory location under the control of the IAP feature (see the following program example):

```
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 a page erase of the VRS51L1050 Flash memory (the memory area occupied by the ISP boot program cannot be page erased). Each page is 512 bytes in size.

To perform a 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 to 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 to be erased corresponds to the range X(Y-1)00h to XYFFh.

The following program example erases 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 & Chip Protect Functions

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

When the chip protect function is enabled, values read back from Flash memory will be 00h.

Program Status Word Register

The PSW register is a bit-addressable register that contains the status flags (CY, AC, OV, P), user flag (F0) and register bank select bits (RS1, RS0) of the 8051 processor.

TABLE 12: 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 Register bank select bit 0
3	RS0	R0-R7 Register 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

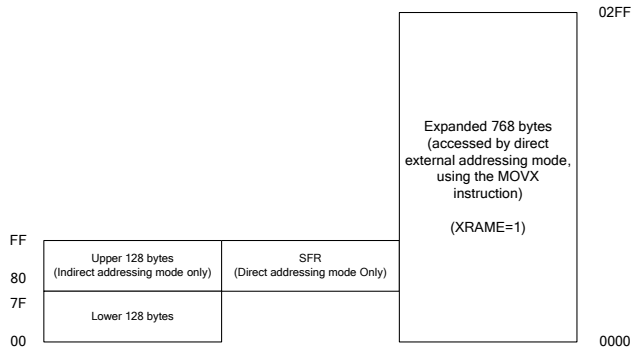
Data Pointer

The VRS51L1050 has one 16-bit data pointer. The DPTR is accessed via two SFR addresses: DPL located at address 82h and DPH located at address 83h.

Data Memory

The VRS51L1050 has 1KB of on-chip SRAM: 256 bytes are mapped into the internal memory bus as is standard for 8052 MCUs. The remaining 768 bytes (expanded SRAM) can be accessed using external memory addressing via the MOVX instruction.

FIGURE 6: VRS51L1050 DATA MEMORY



By default after reset, only the 256 bytes of SRAM mapped to internal memory is accessible (access to the remaining 768 bytes of SRAM is disabled). This 768 bytes 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)

Details of the lower 128 bytes of data memory (from 00h to 7Fh) are summarized as follows:

- Address range 00h to 7Fh can be accessed in direct and indirect addressing modes.
- Address range 00h to 1Fh includes R0-R7 register areas.
- 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 data memory ranging from 80h to FFh can be accessed using indirect addressing or by using bank mapping in direct addressing mode.

Stack Pointer

The stack pointer is a register located at address 81h of the SFR register area whose value corresponds or points to the address of the last item placed on the processor stack. The stack pointer contents are incremented each time new data is placed on the stack.

By default, the stack pointer value is 07h, but it is possible to program the processor stack pointer to point anywhere in the 00h to FFh range of SRAM. When a function call is performed or an interrupt is serviced, the 16-bit return address (two bytes) is stored on the stack. Data can be placed manually on the stack by using the PUSH and POP functions.

Expanded SRAM Access Using the MOVX @DPTR Instruction

The 768 bytes of expanded SRAM data memory occupies addresses 0000h to 02FFh. This can be accessed using external direct addressing (i.e. the MOVX instruction). Note that in the case of indirect addressing using the *MOVX @DPTR* instruction, if the address is larger than 02FFh, the VRS51L1050 will access off-chip memory in the external memory space using the external memory control signals.

Internal SRAM Control Register

The 768 bytes of expanded SRAM can also be accessed using the MOVX @Rn instruction (where n = 0 or 1). This instruction can only access data in a range of 256 bytes. The internal SRAM control register (RCON) allows users to select which part of the expanded SRAM will be accessed by this 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 13: INTERNAL SRAM 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 OVX @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 SRAM at address 145h (page 1).

Note that when both RAMS1 and RAMS0 are set to 1, the value of P2 defines the upper byte and Rn defines the lower byte of the external address. In this case, the device will access the off-chip memory in the external memory space using the external memory control signals. Off-chip peripherals can, therefore, be mapped into the "P2value"00h to "P2value"FFh address range.

Description of Peripherals

System Control Register

The following table describes the system control register (SYSCON).

TABLE 14: SYSTEM CONTROL REGISTER (SYSCON) – SFR BFH

7	6	5	4	3	2	1	0
			PDWAKEUP		IAPE	XRAME	ALEI

Bit	Mnemonic	Description
7	Unused	-
6	Unused	-
5	Unused	-
4	**PDWAKEUP**	Power down wakeup from INT0/INT1 0 = Deactivated 1 = Device can exit power down from the external interrupt
3	Unused	-
2	IAPE	IAP function enable bit 0 = IAP function is deactivated 1 = IAP function is activated
1	XRAME	768 byte on-chip enable bit 0 = Enabled 1 = Disabled
0	ALEI	ALE output inhibit bit, which is used to reduce EMI. 0 = ALE pin is active 1 = ALE is inhibited

Bit 4 of the SYSCON register is the PDWAKEUP bit that, when set to 1, allows the device to exit power down mode from external interrupt INT0/INT, provided it is activated. If the PDWAKEUP bit is cleared, the external INT0/INT1 will not wake up the processor.

The IAPE bit is used to enable and disable the IAP function.

The XRAME bit allows the user to enable the on-chip expanded 768 bytes of SRAM by setting the XRAME bit to 1. By default, upon reset the XRAME bit is set to 0.

Bit 0 of the SYSCON register is the ALE output inhibit bit. Setting this bit to 1 will inhibit the Fosc/6 clock signal output to the ALE pin.

Power Control Register

The VRS51L1050 provides two power saving modes, idle and power down, which are controlled by the PDOWN and IDLE bits of the PCON register at address 87h.

TABLE 15: POWER CONTROL REGISTER (PCON) - SFR 87H

7	6	5	4	3	2	1	0
SMOD				GF1	GF0	PDOWN	IDLE

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

The SMOD bit of the PCON register controls the oscillator divisor applied to Timer 1 when used as a baud rate generator for the UART. Setting this bit to 1 doubles the UART's baud rate generator frequency.

In idle mode, the processor is disabled and the oscillator continues operating. The contents of the SRAM, I/O state and SFR registers are maintained and the timer and external interrupts remain operational. The processor will be woken up when an external event, triggering an interrupt, occurs.

In power down mode, the oscillator and peripherals are disabled. The contents of the SRAM and the SFR registers, however, are maintained.

Exiting Power Down

The VRS51L1050 features two options for exiting power down mode:

- Hardware Reset
- Triggering External Interrupts INT0 or INT1

For the VRS51L1050 to exit power down mode from an external interrupt (INT0 or INT1), the PDWAKEUP bit of the SYSCON register must be set to 1 and the external interrupt must be activated and configured to be edge or level sensitive.

Since the oscillator is disabled in power down mode, when an interrupt is received there will be a delay before the system restarts. The length of the delay before the device exits power down mode will be 65-75K oscillator cycles, may vary from device to device and depends on the crystal used (approximately 3.1ms for a 22.1184MHz crystal and 6.2ms for a 11.0592MHz crystal).

When the VRS51L1050 exits power down mode as a result of an external interrupt, the program counter will jump to its associated interrupt service routine. Upon completion of the interrupt service routine, the processor will return to the main program and execute the next instruction following the one that put the device into power down mode. When the VRS51L1050 is in power down mode, its current consumption drops below 20uA.

Input/Output Ports

The VRS51L1050 has 36 bi-directional lines grouped into four 8-bit I/O ports and one 4-bit I/O port. These I/Os can be individually configured as inputs or outputs. The VRS51L1050 I/O pins are not 5V tolerant.

Except for the P0 I/Os, which are of the open drain type, each I/O consists of a transistor connected to ground and a weak pull-up resistor (transistor-based).

Writing a 0 in a given I/O port bit register will activate the transistor connected to Vss. This will bring the I/O to a low level.

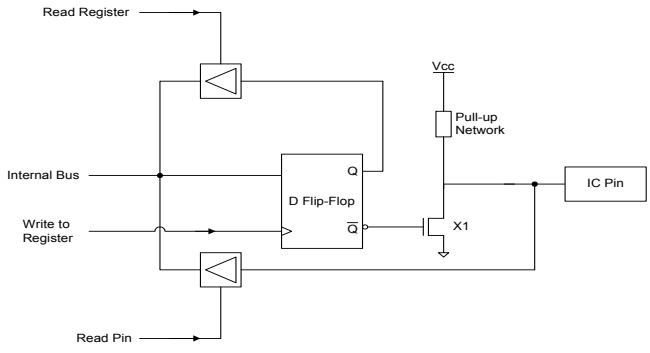
Writing a 1 into a given I/O port bit register deactivates the transistor between the pin and ground. In this case, an internal weak pull-up resistor will bring the pin to a high level (except for Port 0 which is open-drain).

To use a given I/O as an input, a 1 must be written into its associated port register bit. By default, upon reset all I/Os are configured as inputs. The VRS51L1050 I/O ports are not designed to source current.

Structure of the P1, P2, P3 and P4 Ports

The following figure provides the general structure of the P1, P2 and P3 port I/Os. For these ports, the output stage is composed of a transistor (X1) and a transistor set configured as a weak pull-up. Note that the figure below does not show the intermediary logic that connects the register's output and the output stage because this logic varies with the auxiliary function of each port.

FIGURE 7: GENERAL STRUCTURE OF THE OUTPUT STAGE OF P1, P2, P3 AND P4



Each I/O may be used independently as a logical input or output. When used as an input, as mentioned previously, the corresponding bit register must be high. This corresponds to #Q=0 in the above figure.

The transistor will be off (open-circuited) and current will flow from the VCC to the pin, generating a logical high at the output. Note that if an external device with a logical low value is connected to the pin, current will flow out of the pin.

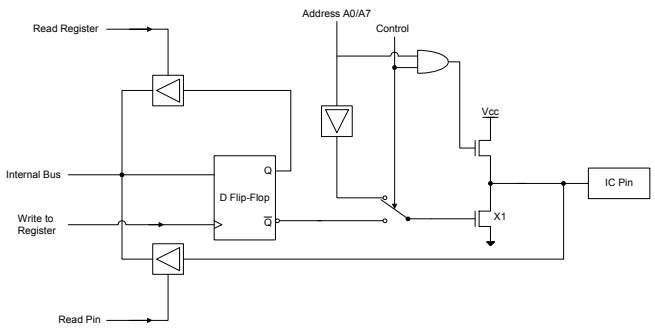
The presence of the pull-up resistance, even when the I/Os are configured as inputs, means that a small current is likely to flow from the VRS51L1050 I/O's pull-up resistors to the driving circuit when the inputs are driven low. For this reason, the VRS51L1050 I/O ports P1, P2, P3 and P4 are called "quasi bi-directional".

Structure of Port 0

The internal structure of P0 is shown in the next figure. 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 is configured for accessing 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 will likely be required for most applications.

FIGURE 8: 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.

The bit-addressable P0 register, located at address 80h, controls the P0 pin directions when used as an I/O (see the following table).

TABLE 16: PORT 0 REGISTER (P0) - SFR 80H

7	6	5	4	3	2	1	0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

Bit	Mnemonic	Description
7	P0.7	Each bit of the P0 register corresponds to an I/O line: 0: Output transistor pulls the line to 0V 1: Output transistor is blocked so the pull-up brings the I/O to 3.3V
6	P0.6	
5	P0.5	
4	P0.4	
3	P0.3	
2	P0.2	
1	P0.1	
0	P0.0	

Port 2

Port P2 is similar to Port 1 and Port 3, the difference being that P2 is used to drive the A8-A15 lines of the address bus when the EA line of the VRS51L1050 is held low at reset time, or when a MOVX instruction is executed.

Like the P0, P1 and P3 registers, the P2 register is bit-addressable.

TABLE 17: PORT 2 REGISTER (P2) - SFR A0H

7	6	5	4	3	2	1	0
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0

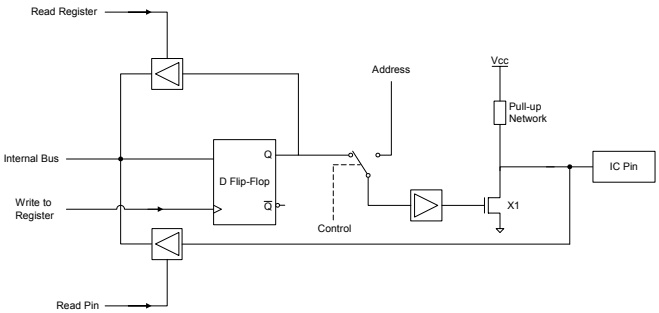
Bit	Mnemonic	Description
7	P2.7	Each bit of the P2 register corresponds to an I/O line: 0: Output transistor pulls the line to 0V 1: Output transistor is blocked so the pull-up brings the I/O to 3.3V
6	P2.6	
5	P2.5	
4	P2.4	
3	P2.3	
2	P2.2	
1	P2.1	
0	P2.0	

Port P0 and P2 as Address and Data Bus

The output stage may derive its 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 byte (A8 through A15) of the bus address for the P2 port.

FIGURE 9: P2 PORT STRUCTURE



When the ports are used as an address or data bus, 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 contents of the P2 register remain constant.

Port 1

The P1 register controls the direction of the Port 1 I/O pins. Writing a 1 into the P1.x bit (see the following table) of the P1 register configures the bit as an output, presenting a logic 1 to the corresponding I/O pin, or enables use of the I/O pin as an input. Writing a 0 activates the output “pull-down” transistor which will force the corresponding I/O line to a logic low.

TABLE 18: PORT 1 REGISTER (P1) - SFR 90H

7	6	5	4	3	2	1	0
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0

Bit	Mnemonic	Description
7	P1.7	Each bit of the P1 register corresponds to an I/O line: 0: Output transistor pulls the line to 0V 1: Output transistor is blocked so the pull-up brings the I/O to 3.3V
6	P1.6	
5	P1.5	
4	P1.4	
3	P1.3	
2	P1.2	
1	P1.1	
0	P1.0	

Auxiliary Port 1 Functions

The Port 1 I/O pins are shared with the I²C-compatible interface, 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	PWM0	PWM0 output
P1.3	PWM1	PWM1 output
P1.4		
P1.5		
P1.6	SCL	I²C SCL
P1.7	SDA	I²C SDA

Port 3

The structure of Port 3 is similar to that of Port 1.

TABLE 19: PORT 3 REGISTER (P3) - SFR B0H

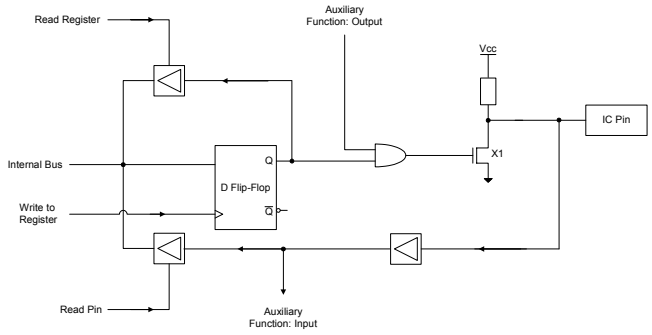
7	6	5	4	3	2	1	0
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0

Bit	Mnemonic	Description
7	P3.7	Each bit of the P3 register corresponds to an I/O line: 0: Output transistor pulls the line to 0V 1: Output transistor is blocked so the pull-up brings the I/O to 3.3V To configure P3 pins as inputs or use alternate P3 functions, the corresponding bit must be set to 1.
6	P3.6	
5	P3.5	
4	P3.4	
3	P3.3	
2	P3.2	
1	P3.1	
0	P3.0	

Auxiliary P3 Port Functions

The Port 3 I/O pins are shared with the UART interface, INT0 and INT1 interrupts, Timer 0 and Timer 1 inputs, and the #WR and #RD lines when external memory accesses are performed.

FIGURE 10: P3 PORT STRUCTURE



The following table describes the auxiliary functions of the Port 3 I/O pins.

TABLE 20: 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

Port 4

Port 4 has four related I/O pins and its port address is located at 0D8H.

TABLE 21: 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 Port Control

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

Upon execution of these instructions, the contents of the port register (at least 1 bit) are 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 re-copies 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 read value 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 use the value of the register rather than that of the pin.

TABLE 22: 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 results in a modification of the content in a port register, the new value is placed at the output of the D flip-flop (see figure) during the last machine cycle of the executed instruction.

Reading a Port (Input)

In order to be sampled, the signal duration present on the I/O inputs must be longer than Fosc/12.

I/O Ports Driving Capability

The maximum allowable continuous current that the device can sink on an I/O port is described in the following table:

Nominal Port 0 pin sink current (0.4V out)	4 to 8 mA
Nominal ports 1, 2, 3, 4 pin sink current (0.4V out)	3 to 6mA
Maximum sink current on a given I/O pin	10mA
Maximum total sink current for P0	26mA
Maximum total sink current for P1, 2, 3,4	15mA
Maximum total sink current on all I/O	71mA

It is not recommended to exceed the above values for sink current as doing so may cause the low-level output voltage to exceed the device's specification and affect device reliability. VRS51L1050 I/O ports are not designed to source current.

VRS51L1050 Timers

The VRS51L1050 includes three 16-bit timers: Timer 0, Timer 1 and Timer 2.

The timers can operate in two 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 system clock (Fosc/12) or by a divided version of it.

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 enable an automatic reload when encountering a specific count value. Timer 1 can also be used as a baud rate generator to generate communication frequencies for the serial interface.

Timer 1 and 0 are configured by the TMOD and TCON registers.

TABLE 23: TIMER MODE CONTROL REGISTER (TMOD) – SFR 89H

7	6	5	4	3	2	1	0
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0

Bit	Mnemonic	Description
7	GATE1	1: Enables external gate control (pin INT1 for Counter 1). When INT1 is high, and the 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	T1M1	Selects the operating mode of Timer/Counter 1
4	T1M0	
3	GATE0	If set, enables external gate control (pin INT0 for Counter 0). When INT0 is high, and the 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	T0M1	Selects the operating mode of Timer/Counter 0.
0	T0M0	

The table below summarizes the four modes of operation of timers 0 and 1. The timer operating mode

is selected by bits T1M1/T1M0 and T0M1/T0M0 of the TMOD register.

TABLE 24: 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 Timer 1 or Timer 0 is configured to operate as a timer, its value is automatically incremented at every machine cycle. Once the timer value rolls over, a flag is set and the counter is set to zero. The overflow flags (TF0 and TF1) are located in the TCON register.

The TR0 and TR1 bits of the TCON register gate the corresponding timer operation. In order for the timer to run, the corresponding TRx bit must be set to 1. The IT0 and IT1 bits of the TCON register control the event that will trigger the external interrupt as follows:

IT0 = 0: INT0, if enabled, occurs if a low level is present on P3.2

IT0 = 1: INT0, if enabled, occurs if a high to low transition is detected on P3.2

IT1 = 0: INT1, if enabled, occurs if a low level is present on P3.3

IT1 = 1: INT1, if enabled, occurs if a high to low transition is detected on P3.3

The IE0 and IE1 bits of the TCON register are external flags that indicate that a transition has been detected on the INT0 and INT1 interrupt pins, respectively.

If the external interrupt is configured as edge sensitive, the corresponding IE0 and IE1 flag is automatically cleared when the corresponding interrupt is serviced.

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

TABLE 25: 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 and T1 signals located at the input of the timer.

When the sampling circuit 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. To be properly sampled, the duration of the event presented to the timer input should be greater than 1/24 of the oscillator frequency.

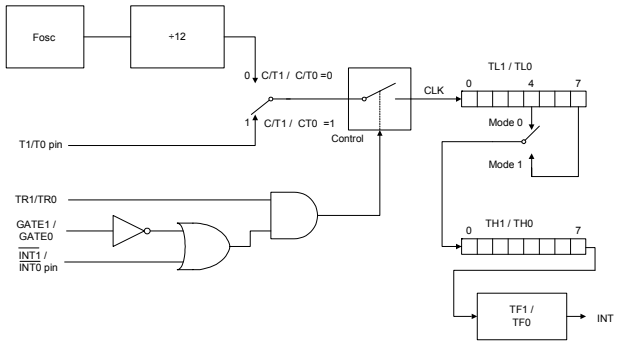
Timer 0 / Timer 1 Operating Modes

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

Mode 0

A schematic representation of this mode of operation is presented in the following figure. In Mode 0, the timer operates as 13-bit counter made up of 5 LSBs from the TLx register and 8 upper bits from the THx register. 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 11: TIMER/COUNTER 1 MODE 0: 13-BIT COUNTER



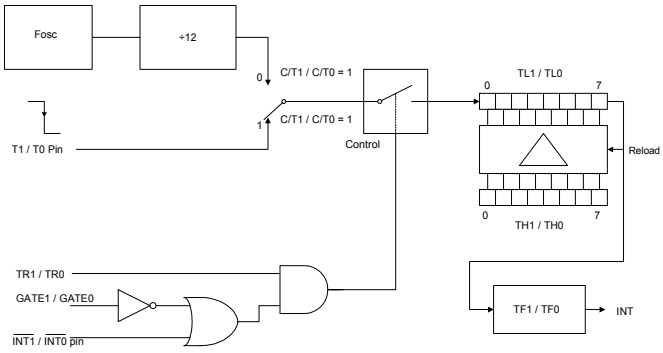
Mode 1

Mode 1 is almost identical to Mode 0, with the difference being that in Mode 1, the counter/timer uses the timer's entire 16 bits.

Mode 2

In this mode, the timer register is configured as an 8-bit auto-re-loadable counter/timer and TLx is used as a 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.

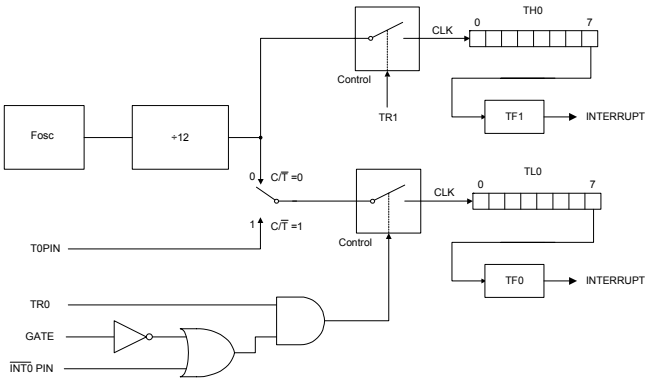
FIGURE 12: TIMER/COUNTER 1 MODE 2: 8-BIT AUTOMATIC RELOAD



Mode 3

In Mode 3, 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. 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 13: TIMER/COUNTER 0 MODE 3



Timer 2

Timer 2 of the VRS51L1050 is a 16-bit timer/counter and is similar to timers 0 and 1 in that it operates as either an event counter or a timer. This is controlled by the C/T2 bit in the T2CON special function register. Timer 2 has three operating modes - auto-load, capture and baud rate generator. These modes are selected via the T2CON SFR. The following table describes T2CON special function register bits:

TABLE 26: 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-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 Timer 2 mode selection bits and their function are described in the following table.

TABLE 27: 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	Timer 2 Stops

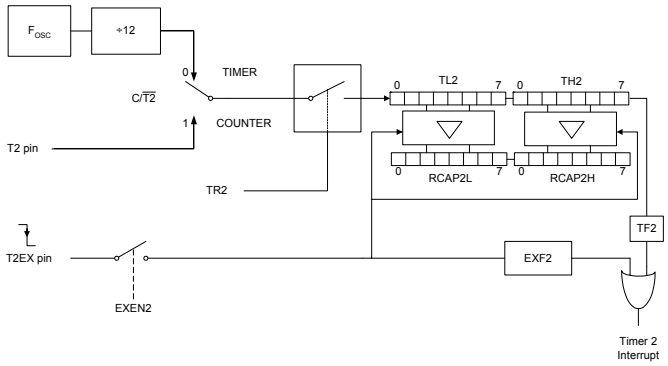
The modes are discussed in the following sections.

Timer 2 Capture Mode

In Capture Mode, the EXEN2 bit of the T2CON register controls whether an external transition on the T2EX pin will trigger a capture of the timer value.

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

FIGURE 14: TIMER 2 IN CAPTURE MODE



When EXEN2 = 1, the above still applies. However, it is also 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 in 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

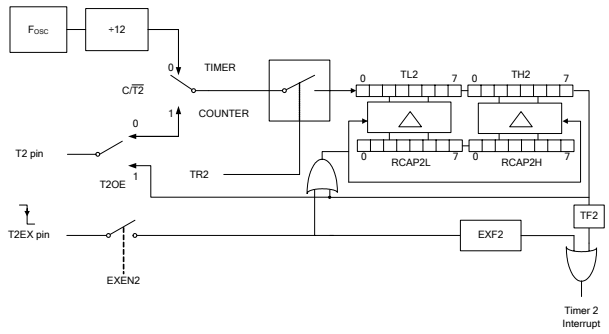
Additionally in this mode, there are two options controlled by the EXEN2 bit in the T2CON register.

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 initialized. In this mode, Timer 2 can be used as a baud rate generator source for the serial port.

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

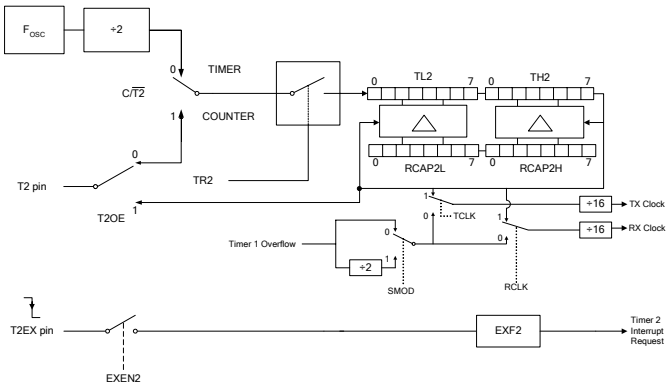
FIGURE 15: TIMER 2 IN AUTO-RELOAD MODE



Timer 2 Baud Rate Generator Mode

Timer 2 can be used for UART baud rate generation. This mode is activated when RCLK is set to 1 and/or TCLK is set to 1. This mode is described further in the serial port section.

FIGURE 16: TIMER 2 IN AUTOMATIC BAUD GENERATOR MODE



Special Features of Timer 2

Timer 2 on the VRS51L1050 incorporates unique features uncommon to standard 8051s. These include:

- Timer 2 Output
- Optional Down Counting

These special features can be activated through the T2MOD register located at SFR address C9h.

TABLE 28: TIMER 2 SPECIAL MODE REGISTER (T2MOD) –SFR C9H

7	6	5	4	3	2	1	0
						T2OE	CDOWN

Bit	Mnemonic	Description
7:2	-	
1	T2OE	Timer 2 Output Enable 0: Timer 2 Output Disabled 1: Timer 2 output Disabled
0	CDOWN	Timer 2 Down Count Enable 0: Timer 2 Counts Up 1: Timer 2 Counts Down

The T2OE bit, when set to 1, will configure the T2 (P1.0) Pin as a Timer 2 output. If the T2OE bit is cleared, the T2 Pin acts as a Timer 2 event counter input if the C/T2 bit of T2CON is set to 1.

When T2OE is set to 1 the T2 Pin will toggle each time Timer 2 overflows.

The Timer 2 output function is incompatible with the Timer 2 event counter mode. When Timer 2 is configured as an event counter (C/T2 = 0), the T2OE must be cleared.

The CDOWN bit, when set to 1, will cause Timer 2 to count down from FFFF to 0000h. However, if Timer 2 is configured in auto-reload mode, it will count from FFFF down to the reload value stored in the RCAP2H, RCAP2L registers.

When the CDOWN bit is cleared, Timer 2 counts from the reload value up to FFFFh.

UART Serial Port

The VRS51L1050's serial port operates in full duplex mode (transmits and receives 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 VRS51L1050 serial port includes a double buffer for the receiver, which allows reception of a byte even if the processor has not retrieved the previously received byte from the receive register. 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.

The SBUF register provides access to the transmit and receive registers of the serial port. Reading from the SBUF register will access the receive register, while a write to the SBUF loads the transmit register.

Serial Port Control Register

The SCON (serial port control) register contains control and status information, and includes the 9th data bit for transmit/receive (TB8/RB8 if required), mode selection bits and serial port interrupt bits (TI and RI).

TABLE 29: 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 following table)
6	SM1	Bit to select mode of operation (see following table)
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, 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 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. • 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. • The stop bit has been sent in the other modes (see SM2 exception). This bit must be cleared by software.

TABLE 30: 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

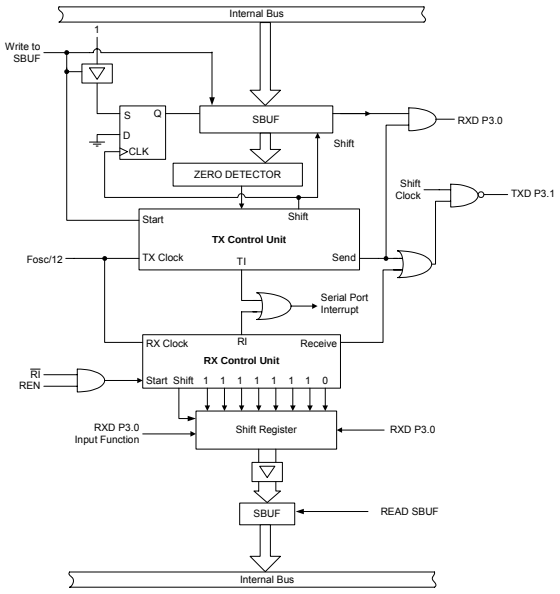
UART Operating Modes

The VRS51L1050's serial port operates in four different modes. In all four modes, a transmission is initiated by an instruction that uses the SBUF register 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 sections describe these four modes.

UART Operation in Mode 0

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

FIGURE 17: 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 informs 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.

At every machine cycle in which SEND is active, the contents of the transmit shift register is 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 deactivates 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 deactivation 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. Bits 11111110 are written to the receive shift register at the end of the next machine cycle by the RX control unit. In the following phase, the RX control unit will activate RECEIVE.

The contents of the receive shift register are shifted one position to the left at the end of every machine cycle during which RECEIVE is active. The value that comes in from the right is the value that was sampled at the P3.0 pin.

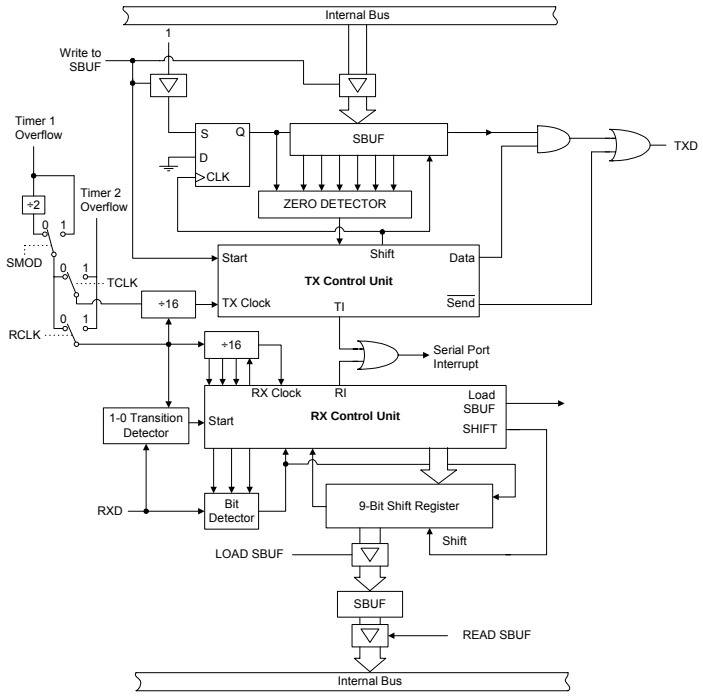
1's are shifted out to the left as data bits are shifted in from the right. The RX control block is flagged to do one last shift and load SBUF when the 0 that was initially loaded into the rightmost position arrives at the leftmost position in the shift register.

UART Operation in Mode 1

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 18: SERIAL PORT MODE 1 AND 3 BLOCK DIAGRAM



UART Transmission in Mode 1

Transmission in this mode 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/informs the TX control unit that a transmission has been requested.

After the next rollover in the divide-by-16 counter, transmission actually begins. 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

A 1 to 0 transition at pin RXD will initiate reception. For this reason, RXD is sampled at a rate of 16 multiplied by the established baud rate. When a transition is detected, 1FFh is written into the input shift register and the divide-by-16 counter is immediately reset (this is done 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. This is done for noise rejection.

If the value accepted during the first bit time is not zero, the receive circuits are reset and the unit returns to searching for another 1 to 0 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 causes the UART's receive controller block to perform one last shift operation: to set RI and 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 enters 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 returns to searching for a 1 to 0 transition in RXD.

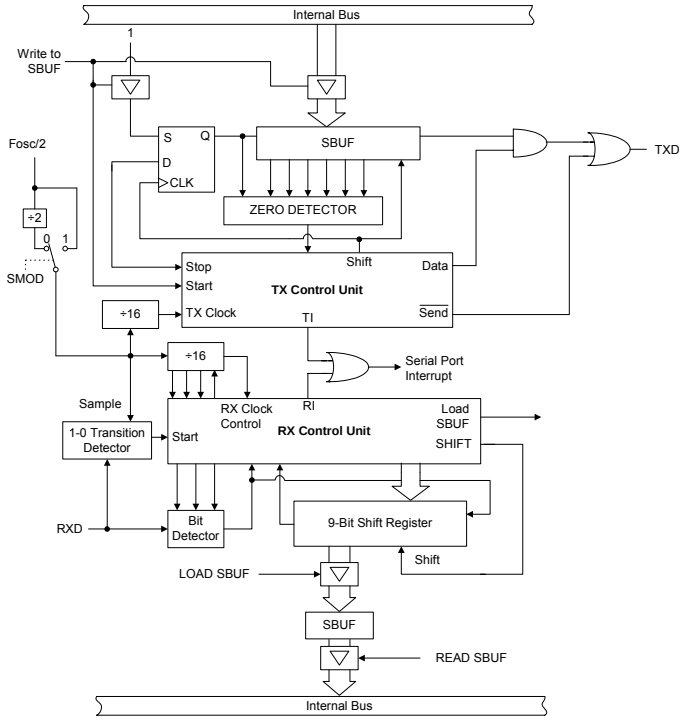
UART Operation in Mode 2

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. For reception, 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 19: SERIAL PORT MODE 2 BLOCK DIAGRAM

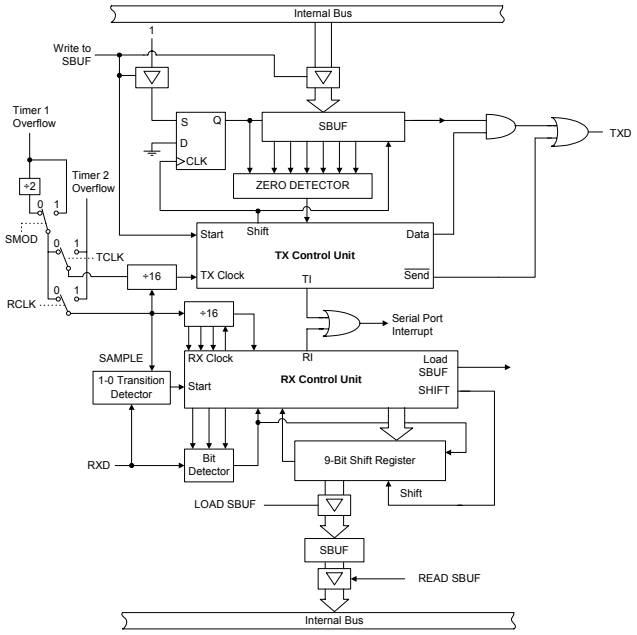


UART Operation in Mode 3

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 20: SERIAL PORT MODE 3 BLOCK DIAGRAM



UART in Mode 2 and 3: Additional Information

As mentioned previously, 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 enters 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 UART transmission control unit that a transmission has been requested. After the next rollover in the divide-by-16 counter, a transmission actually starts at the beginning 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 on the TXD pin. Data is activated one bit time later. This activation enables the output bit of the transmit shift register to the TXD pin. 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 on 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 0 transitions on the RXD pin initiate reception. For this reason, RXD is sampled at a rate of 16 multiplied by the established baud rate. 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 1 to 0 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 instructs 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 = 1
- RI = 0

If both conditions are met, the 9th data bit received enters RB8, and the first 8 data bits enter 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 returns to searching for a 1 to 0 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. The formula below demonstrates that if SMOD = 0 (which is the value on reset), the baud rate is 1/32 the oscillator frequency.

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

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.

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

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

The following formulas can be used to calculate the baud rate and the reload value to be written into the TH1 register.

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

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

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

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 better resolution compared to 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 three running modes. In typical applications, it is configured as a timer (C/T2 is set to 0).

To operate Timer 2 as a baud rate generator, the TCLK and RCLK bits of the T2CON register must be set to 1. Baud rate generator mode is similar to 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 the 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 Timer 2:

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

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

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

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, Timer 2 interrupt does not have to be disabled when Timer 2 is configured in baud rate generator mode.

Furthermore, when Timer 2 is configured as a UART baud rate generator and running (TR2 is set to 1), the user should not try to perform read or write operations to the TH2 or TL2 and RCAP2H, RCAP2L registers.

Timer 1 Reload Value in Modes 1 and 3 for UART Baud Rate

The following table provides examples of the Timer 1, 8-bit reload value when it is used as a UART baud rate generator and the SMOD bit of the PCON register is set to 1.

	22.184MHz	16.000MHz	14.745MHz	12.000MHz	11.059MHz	8.000MHz	3.57MHz
115200bps	FFh	-	-	-	-	-	-
57600bps	Feh	-	-	-	FFh	-	-
38400bps	FDh	-	FEh	-	-	-	-
31250bps	-	-	-	FEh	-	-	-
19200bps	FAh	-	FCh	-	FDh	-	-
9600bps	F4h	-	F8h	-	FAh	-	-
2400bps	D0h	DDh	E0h	E6h	E8h	-	-
1200bps	A0h	BBh	C0h	CCh	D0h	DDh	-
300bps	-	-	00h	30h	40h	75h	C2h

Timer 2 Reload Value in Modes 1 and 3 for UART Baud Rate

The following are examples of [RCAP2H, RCAP2L] reload values for Timer 2 when it is used as baud rate generator for the VRS51L1050 UART.

	22.184MHz	16.000MHz	14.745MHz	12.000MHz	11.059MHz	8.000MHz	3.57MHz
230400bps	FFFDh	-	FFFEh	-	-	-	-
115200bps	FFFAh	-	FFFCh	-	FFFDh	-	-
57600bps	FFF4h	-	FFF8h	-	FFFAh	-	-
38400bps	FFEEh	FFF3h	FFF4h	-	FFF7h	-	-
31250bps	FFEAh	FFF0h	FFF1h	FFF4h	FFF5h	FFF8h	-
19200bps	FFDCh	FFE6h	FFE8h	-	FFEEh	FFF3h	
9600bps	FFB8h	FFCCh	FFD0h	FFD9h	FFDCh	FFE6h	-
2400bps	FEE0h	FF30h	FF40h	FF64h	FF70h	FF98h	FFD1h
1200bps	FDC0h	FE5Fh	FE80h	FEC7h	FEE0h	FF30h	FFA3h
300bps	F700h	F97Dh	FA00h	FB1Eh	FB80h	FCBEh	FE8Bh

UART Initialization in Mode 3 Using Timer 1

```
;*** INTIALIZE THE UART @ 9600BPS, Fosc=11.0592MHz

INISER0T1I: MOV   A,T2CON          ;RETRIEVE CURRENT VALUE OF T2CON
            ANL   A,#11001111B     ;RCLK & TCLK BIT = 0 -> TO USE TIMER1
            MOV   T2CON,A          ;BAUD RATE GENERATOR SOURCE FOR UART
            MOV   PCON,#80H        ;SET THE SMOD BIT TO 1
            MOV   TL1,#0FAH        ;CONFIG TIMER1 AT 8BIT WITH AUTO-RELOAD
            MOV   TH1,#0FAH        ;CALCULATE THE TIMER 1 RELOAD VALUE
                                   ;TH1 = [(2^SMOD) * Fosc] / (32 * 12 * Fcomm)
                                   ;TH1 FOR 9600BPS @ 11.059MHz = FAh
            MOV   SCON,#05Ah       ;CONFIG SCON_0 MODE_1
            MOV  TMOD,#00100000B ;CONFIG TIMER 1 IN MODE 2, 8BIT
                                   ; + AUTO RELOAD
            MOV   TCON,#01000000B  ;START TIMER1


            CLR   SCON.0           ;CLEAR UART RX, TX FLAGS
            CLR   SCON.1

            MOV   SBUF,#DATA       ;SEND ONE BYTE ON THE SERIAL PORT
```

UART Initialization in Mode 3, Using Timer 2

```
;*** INTIALIZE THE UART @57600BPS, Fosc=11.0592MHz

INISER0T2I: MOV   SCON,#05Ah       ;CONFIG SCON_0 MODE_1,
                                   ;CALCULATE RELOAD VALUE WITH T2
                                   ;RCAP2H,RCAP2L = 65536 - [ Fosc / (32*Fcomm)]
            MOV   RCAP2H,#0FFh     ;RELOAD VALUE 57600bps,  11.059MHz  =FFFAh
            MOV   RCAP2L,#0DCh     ;

            MOV  T2CON,#034h       ;SERIAL PORT0, TIMER2 RELOAD START


            CLR   SCON.0           ;CLEAR UART RX, TX FLAGS
            CLR   SCON.1

            MOV   SBUF,#DATA       ;SEND ONE BYTE ON THE SERIAL PORT
```

I²C-Compatible Interface

The VRS51L1050 provides an I²C-compatible interface that operates in master and slave modes. In master mode, the transaction speed is adjustable and can reach speeds of up to 400kbps.

The VRS51L1050's I²C interface can simultaneously accommodate a number of devices connected on the same I²C bus, as long as the driving capacity load does not exceed 400pF.

A complete set of SFR registers control the I²C interface. The I²C interface shares lines SCL and SDA (respectively) of the P1.6 and P1.7 I/O ports. The I²C controls these I/O lines when bits 6 and 7 of the I2CPWME SFR registers are set to 1.

TABLE 31: I2CPWME CONFIGURATION REGISTER (PWME) - SFR -9BH

7	6	5	4
SDAE	SCLE	-	-

3	2	1	0
PWM1E	PWM0E		

Bit	Mnemonic	Description
7	SDAE	I²C SDA Enable 0: P1.7 I/O operate as regular I/O 1: P1.7I/O is dedicated to I²C SDA
6	SCLE	I²C SCL Enable 0: P1.6 I/O operate as regular I/O 1: P1.6I/O is dedicated to I²C SCL
5	-	
4	-	
3	PWM1E	PWM1 Enable Register 0: PWM1 module is deactivated 1: PWM1 module is activated on P1.3
2	PWM0E	PWM1 Enable Register 0: PWM1 module is deactivated 1: PWM1 module is activated on P1.3
1	-	
0	-	

I²C Control Registers

The primary I²C control registers are SFR registers, I2CCTRL1 and I2CCTRL2, described below.

TABLE 32: I²C CONTROL REGISTER 1 (I2CCTRL1) – SFR C2H

7	6	5	4
I2CEN	-	-	-

3	2	1	0
I2CBUSY	I2CCK[2:0]		

Bit	Mnemonic	Description
7	I2CEN	I²C Interface Enable 0 : I²C interface is disabled 1 : I²C Interface is enabled
6	-	
5	-	
4	-	
3	I2CBUSY	I²C Bus Status 0: I²C bus is idle 1 I²C bus is busy
2	I2CCK2	I²C Clock Speed Configuration (see table below)
1	I2CCK1	
0	I2CCK0	

In order for the I²C interface module to operate, it must first be enabled by setting the I2CEN bit of the I2CCTRL1 register to 1.

The BUSY bit indicates the current state of the I²C bus. It is set to 1 when a start condition is detected on the bus and is cleared when a stop condition is detected. Before initiating a transaction on the I²C bus, make sure the BUSY bit is cleared (I²C bus is free).

The I2CCKx bits of the I2CCTRL1 register define the communication speed of the I²C interface when it operates in master mode. By default, upon reset, the I²C communication speed is set to Fosc/64.

TABLE 33: I²C CMMUNICATION SPEED IN MASTER MODE VX I2CCK[2:0]

I2CCK[2:0] bit value	I²C communication. speed	Com. speed @Fosc 25MHz	Com. speed @Fosc 11.05
000	Fosc / 32	-s	346 kbps
001	Fosc / 64 (default)	390 kbps	173 kbps
010	Fosc / 128	195 kbps	84.4 kbps
011	Fosc / 256	97.6 kbps	43.2 kbps
100	Fosc / 512	48.8 kbps	21.6 kbps
101	Fosc / 1024	24.41 kbps	10.8 kbps
110	Fosc / 2048	12.21 kbps	5.4kbps
111	Fosc / 4096	6.10 kbps	2.7kbps

TABLE 34: I²C CONTROL REGISTER 2 (I2CCTRL2) - SFR –3H

7	6	5	4
MATCH	SLAVERW	-	-

3	2	1	0
RESTART			MASTERRW

Bit	Mnemonic	Description
7	MATCH	I²C Received address vs. I2CADDR match indicator 0: No match between I²C address and I2CADDR register content 1: Last I²C address received matches value present in the I2CADDR register
6	SLAVERW	Slave Mode Operation 0: Slave mode read (data received) 1: Slave mode write (data transmitted)
5	-	
4	-	
3	RESTART	Master Mode Restart Signal 0: No action 1: The I²C interface will send a start followed by I2CADDR content
2	-	
1	-	
0	MASTERRW	Master Mode Data Direction 0: Master mode write 1: Master mode read

The MATCH bit of the I2CCTRL2 register is used for slave I²C transactions. When the received data following a start equals the value present in the I2CADDR register, the MATCH bit will be set. In the case where the MSBCOMP bit is set to 1, the MATCH bit will be set when the upper four bits of the received address correspond to the upper four bits in the I2CADDR register. The processor can monitor the MATCH bit to detect the beginning of an I²C transaction addressed to it.

The SLAWERW bit is used in slave mode to inform the processor of the data direction. This bit is updated after the calling address is received in slave mode.

If data is going to be received, the SLAVERW bit will be 0. If data is going to be transmitted, the SLAVERW bit will be set to 1.

The SLAVERW is especially useful in programs using the interrupt to manage I²C slave transactions. The SLAVERW bit is cleared upon device reset.

The RESETART is only active in master mode. When this bit is set to 1, the I²C interface will generate a start condition after the current acknowledge phase, and then send the content of the I2CADDR register to the I²C bus. If the addressed slave device fails to acknowledge, the I2CTXFAIL bit of the I2CSTATUS register will be set to 1, the RESTART bit will be cleared and the I²C interface will release the bus. The RESTART bit is automatically cleared after the I²C interface has generated the start condition and after a device reset.

The MASTERRW bit of the I2CCTRL2 register defines the data direction in master mode. This bit serves as bit 0 of the I²C address that will be sent to the I²C bus in master mode.

To perform a read operation, the MASTERRW bit must be set to 1. To perform a write operation, the MASTERRW bit must be cleared.

The I²C Status Register

The I2CSTATUS register provides most of the indicators for the I²C interface. The four upper bits of this register contain the interrupt flags and the lower three bits are used for I²C interface control and monitoring.

TABLE 35: I²C STATUS REGISTER 1 (I2CSTATUS) – SFR C0H

7	6	5	4
I2CRXIF	I2CTXIF	I2CTXFAIL	I2CNOACKIF

3	2	1	0
	I2CRXACK	I2CMASTER	I2CTXACK

Bit	Mnemonic	Description
7	I2CRXIF	I²C Reception Interrupt Flag
6	I2CTXIF	I²C Transmission Interrupt Flag
5	I2CTXFAILIF	I²C Transmission Fail Interrupt Flag
4	I2CNOACKIF	I²C No Acknowledge Received interrupt Flag
3	-	
2	I2CRXACK	I²C Reception Acknowledge
1	I2CMASTER	I²C Master mode
0	I2CTXACK	I²C Transmission Acknowledge

The I2CRXIF flag will be set to 1 by the I²C upon the reception of new data in the I2CRX register. Once the data is loaded into the I2CRX register, the I2CRXIF flag will be set. No new data received on the I²C interface can be loaded into the I2CRX until the processor retrieves the data already in the I2CRX register. The I2CRXIF flag will be automatically cleared when the processor reads the I2CRX. This bit can also be cleared manually by the processor.

The I2CTXIF flag will be set to 1 by the I²C once the data present in the I2CTX register is sent to the interface's shift register and the I2CTX register is ready to receive the next data byte to be transmitted. The I2CTXIF flag will be automatically cleared when new data is written into the I2CTX register. It can also be manually cleared by the processor.

The I2CTXFAILIF flag will be set to 1 if the data transmission fails. The I2CTXFAILIF flag will also be reset if an arbitration loss condition is detected by the I²C interface in master mode. The arbitration loss condition occurs when the master tries to transmit a 1 on the SDA line but it detects a 0 there. The I2CTXFAILIF flag must be cleared manually.

The I2CNOACKIF bit is only set in master mode when an acknowledge signal has not been detected after a data transmission. The I2CNOACKIF flag must be cleared manually.

The I2CRXACK bit is a read-only, active low flag that, when cleared, indicates that an acknowledge signal has been received from the master after an 8-bit data transmission is completed in slave mode.

The RXACK bit will be set to 1 after a reset or when no acknowledge signal is detected during the acknowledge phase of slave data transmission. In this case, the I²C interface will release the SDA line in order to allow the bus master to generate a stop or another start condition.

When the I2CMASTER is set to 1 by the processor, it will force the I²C interface into master mode and immediately initiate a transaction beginning with a start and followed by the address stored in the I2CADDR (a read or a write operation). In the case of a write operation, the value present in the I2CTX register will be sent to the bus, provided that a valid acknowledge signal from the slave device is received after the address transmission.

When the I2CMASTER is cleared either by the software or the I2CNACKIF flag, the I²C interface will generate a stop on the I²C bus after the current byte transmission is complete. Any data present in the I2CTX register that was not transmitted will not be transmitted.

In the case where the I2CTXIF bit is set after a data transmission fails, the I²C interface will immediately release the SCL and SDA lines.

The I2CTXACK is the acknowledge status bit. The value of I2CTXACK defines the value to be put in SDA during the acknowledge phase of a slave data reception. If I2CTXACK is set to 1, it indicates that no acknowledge signal was sent to the master. If the I2CTXACK bit is cleared, a valid acknowledge will be sent to the master. This feature is useful for informing the master device on the I²C bus that the VRS51L1050 is busy. The I2CTXACK is automatically cleared at reset and can be set/cleared manually by the processor.

I²C Address Register

The I2CADDR register contains the device address that will be transmitted in master mode. The MASTERRW bit of the I2CCTRL2 register holds the value of address bit 0 (read/write operation) to be sent by the master following the start condition.

In slave mode, the content of the I2CADDR register is compared with the incoming address sent by the I²C bus master.

TABLE 36: I²C ADDRESS REGISTER (I2CADDR) - SFR –C1H

7	6	5	4
I2CADDR7	I2CADDR6	I2CADDR5	I2CADDR4

3			
I2CADDR3	I2CADDR2	I2CADDR1	MSBCOMP

Bit	Mnemonic	Description
7:1	I2CADDR[7 :1]	I²C Address to be sent in master mode I²C Slave address in slave mode
0	MSBCOMP	I²C Address compare 0: 7 address bits are compared in slave mode 1: Compare only the four most significant bits in slave mode

The MSBCOMP bit is used in slave mode. When this bit is set to 1, the I²C interface will send an acknowledge signal to the general call address (00h) and a compare between the received address and the value of the I2CADDR register will be made on the four most significant bits.

When the MSBCOMP bit is cleared, the I²C interface will only acknowledge to the calls that have an address matching the upper seven bit of the I2CADDR register.

I2CTX and I2CTX Registers

The I2CTX register contains the data to be transmitted on the I²C interface.

In master mode, the content of the I2CTX register will be sent to the interface's shift register when the receive acknowledge signal is received from the slave device (I2CRXACK = 0).

In slave mode, the content of the I2CTX register will be sent to the interface's shift register when a matching address is received (MATCH = 1) and bit 0 of the incoming address is 1 (read operation).

As soon as the contents of the I2CTX register is sent to the interface's shift register the I2CTXIF flag of the I2CSTATUS will be set to 1 and an I²C interrupt will be triggered if it was enabled.

TABLE 37: I²C TRANSMIT REGISTER (I2CTX) - SFR –C4H

7	6	5	4
I2CTX7	I2CTX6	I2CTX5	I2CTX4

3	2	1	0
I2CTX3	I2CTX2	I2CTX1	I2CTX0

Bit	Mnemonic	Description
7:0	I2CTX[7:0]	I²C Transmit Register

In the case where the I2CTX register is not updated in time, the I²C interface will hold the I²C SCL line down after the acknowledge phase until new data is written into the I2CTX register. When new data arrives in the I2CTX register, it will be immediately transferred to the I²C shift register for transmission and the I²C module will release the SCL line. Simultaneously, the I2CTXIF interrupt flag will be raised to request new data from the processor.

In slave mode, if the master device does not acknowledge after a byte transmission from the I²C module, the I2CRXACK bit will remain at 1 forcing the I²C interface to release the SDA line so the master can generate a stop condition on the bus.

The I2CRX register contains the data received on the I²C interface.

TABLE 38: I²C RECEIVE REGISTER (I2CRX) - SFR –C5H

7	6	5	4
I2CRX7	I2CRX6	I2CRX5	I2CRX4

3	2	1	0
I2CRX3	I2CRX2	I2CRX1	I2CRX0

Bit	Mnemonic	Description
7:0	I2CRX[7:0]	I²C Transmit Register

In the slave mode, if the MATCH bit equals 1, the I2CRX register will contain the last received data. If the MATCH bit equals 0, the I2CRX register will contain the device address called by the I²C bus master.

The I2CRX register will be updated with the new data received as soon its reception is complete and provided the previously received data has been retrieved by the processor. In the case where the I2CRX register contents have not been retrieved by the processor, the I²C interface will pull the SCL line low to stop any further data reception until the I2CRX register is read by the processor.

As soon as a new data byte is available in the I2CRX register, the I2CRXIF bit of the I2CSTATUS register will be set to 1. Once the data is retrieved by the processor, the I2CRXIF flag will be automatically cleared and new data can be received into the I2CRX register.

If the I²C interrupt is enabled, it will be triggered as soon as the I2CRXIF flag is set to 1.

I²C Example Program

Basic EEPROM interface program

The following shows a basic I²C interface program for an EEPROM device

```
//-------------------------------------------------------------------------------------------------------------------//
// VRS51L1050_I²C_24xx64.c //
//-------------------------------------------------------------------------------------------------------------------//
// DESCRIPTION:          24xx64 EERPOM basic interface Demonstration Program.
//
// Target Device:        VRS51L1050
//-------------------------------------------------------------------------------------------------------------------//
#include <VRS51L1050_SDCC.h>

#define OK  0x01;
#define BUG 0x00;
#define BUSY 0x00;

//--EEPROM I²C Functions
char EE_I²C_Busy( char );
char EE_I²C_ByteWrite( char, int, char);
char EE_I²C_RandomByteRead( char, int);
char EE_I²C_Read( char );
void I²C_MConfig(void );

//-------------------------------------------------------------------------------------------------------------------//
//          MAIN FUNCTION
//-------------------------------------------------------------------------------------------------------------------//
void main (void) {
        char x;
        //--Configure the I²C
        I²C_MConfig( );

         //--Write Data Byto to FRAM
//       x = EE_I²C_ByteWrite( 0x00, 0x0302, 0x0F);

//      if(x == 0x00)
//              while(!EE_I²C_Busy(0x00));      //--wait Device to be ready

        //Read the Data byte from the EEPROM
        x = EE_I²C_RandomByteRead( 0x00, 0x0302);

        while(1);
}// End of main

//-------------------------------------------------------//
//;       EE_I²C_Busy
//-------------------------------------------------------//
char EE_I²C_Busy( char device)
        {
        I2CNOACKF = 0;
        I2CTXACK = 0;
        while((I2CCTRL1 & 0x08) != 0x00){};     //--Wait Bus idle

        //Configure I²C ID and device number
        device = (device << 2)& 0x0E;
        I2CADDR = 0xA0 + device;

        //Configure master mode Data direction = Write
        I2CCTRL2 &= 0xFE;                       //I²C Master Write
        //Start I²C
        I2CMASTER = 1;                          //Start Transaction
        while((I2CTXIF) == 0x00){};             //wait TXIF flag to get set
        I2CMASTER = 0;                          //Generate a stop condition

        while((I2CCTRL1 & 0x08) != 0x00){};     //--Wait Bus idle

        if(I2CNOACKF)
                return BUSY
        else
                return OK;
        }//end of EE_I²C_Busy
```

```
//-------------------------------------------------//
//              EE_I²C_ByteWrite
//-------------------------------------------------//
char EE_I²C_ByteWrite( char device, int address, char txdata)
        {
        Int         adrstemp = address;


        //--Wait Bus idle
        while((I2CCTRL1 & 0x08) != 0x00){};

        //Configure I²C ID and device number
        device = (device << 2)& 0x0E;
        I2CADDR = 0xA0 + device;

        //Configure master mode Data direction = Write
        I2CCTRL2 &= 0xFE;       //I²C Master Write

        //Start I²C
        I2CMASTER = 1;

        //--Send MSB of address
        while((I2CSTATUS & 0x40) == 0x00){};  //wait TXIF flag to get set
        I2CTX = (adrstemp >> 8)& 0x1F;        //Send lower 5 bit of MCB address

        //--Send LSB of address
        while(!I2CTXIF){};                    //wait TXIF flag to get set
        I2CTX = address & 0x00FF;             //Send ILSB of address

        //--Send Data
        while(!I2CTXIF){};                    //wait TXIF flag to get set
        I2CTX = txdata;                       //Send  Data

        while(!I2CTXIF){};                    //wait TXIF flag to get set
        I2CSTATUS &= 0xFD;                    //Clear MASTER -> Generate STOP

        if(I2CTXACK)
                return OK                     //if NAKIF Flag == 0 -> ACK received
                else
                                return BUG;   //else device is nt responding
        }//end of EE_I²C_ByteWrite


//-----------------------------------------------------------------//
//;              EE_I²C_RandomByteRead
//-----------------------------------------------------------------//
char EE_I²C_RandomByteRead( char device, int address)
        {
        int         adrstemp =  address;

        while((I2CCTRL1 & 0x08) != 0x00){};   //--Wait Bus idle

        //Configure I²C ID and device number + Write address
        device = (device << 1)& 0x0E;
        I2CADDR = 0xA0 + device;

        I2CMASTER = 1;                        //Start I²C Transaction

        //--Send MSB of address
        while(!I2CTXIF){};                    //wait TXIF flag to get set
        I2CTX = (adrstemp >> 8)& 0x1F;        //Send lower 5 bit of MCB address

        //--Send LSB of address
        while(!I2CTXIF){};                    //wait TXIF flag to get set
        I2CTX = address & 0x00FF;             //Send ILSB of address

        while(!I2CTXIF){};                    //wait TXIF flag to get set

        I2CCTRL2 = 0x09;                      //Set device in Read mode and restart

//--Wait for Data Byte to be loaded into I2CTX
        while(!I2CRXIF){};                    //wait RXIF flag to get set

        //Generate a Stop
        I2CSTATUS = 0x00;
        return I2CRX ;

        }//end of FRAM_I²C_RandomByteRead
```

```
//-------------------------------------------------//
//;              EE_I²C_Read
//-------------------------------------------------//
char EE_I²C_Read( char device)
        {

        //--Wait Bus idle
        while((I2CCTRL1 & 0x08) != 0x00){};

        //Configure I²C ID and device number
        device = (device << 1)& 0x0E;
        I2CADDR = 0xA0 + device;
        I2CCTRL2 = 0x01;                      //Set device in Read mode and restart
        I2CMASTER = 1;
//--Wait for Data Byte to be loaded into I2CTX
        while(!I2CRXIF){};                    //wait RXIF flag to get set

        //Generate a Stop
        I2CMASTER = 0;
        return I2CRX ;

        }//end of EE_I²C_RandomByteRead


//-------------------------------------------------//
//              I²C_MCONFIG
//-------------------------------------------------//

void I²C_MConfig(void )
        {
        //Activate I²C SCL and SDA access to P1
        I2CPWME = 0xC0;

        //Set I²C Speed=Fosc/512 (43kHz @ 22.1184MHz)
        I2CCTRL1 = 0x84;

        while((I2CCTRL1 & 0x08) == 0x08){};           //Wait I2CBUSY == 0

        I2CCTRL2 = 0x00;
                //configure I²C in master Transmit

        I2CADDR = 0xA0;                       //EEPROM Device Address
        }//end of I²C_MConfig
```

Pulse Width Modulation (PWM)

The VRS51L1050 provides two PWM outputs that are shared with the P1.2 and P1.3 I/O pins. Each PWM can be configured to operate with a resolution of 8 or 5 bits, with the 5-bit mode allowing a faster PWM output rate.

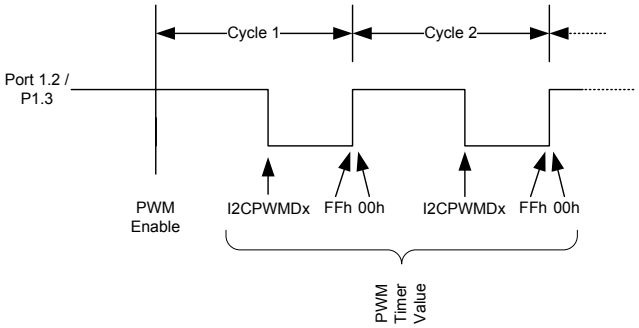
Each PWM module is composed of a free running timer/counter and a comparator. The comparator compares the PWM free running timer to the PWM data register. The comparator output will remain high as long as the value of the free running timer is lower than the value present in the PWMDx register.

Once the value of the PWM timer equals the value in the PWMDx register, the PWM output will be set to 0 and will remain in that state until the PWM timer overflows.

The maximum timer value before an automatic overflow and restart occurs depends on the configuration of the PWM module. When the PWM module is configured to operate at 8 bits, the overflow will occur at FFh. When the PWM module is configured to operate in 5-bit mode, the overflow will occur at 1Fh.

The following diagram demonstrates the relationship between the PWM output vs. the PWM timer, when the PWM module is configured in 8-bit mode.

FIGURE 21: PWM OUTPUT VS. PWM TIMER VALUE



There is no interrupt associated with the PWM modules.

PWM Control Registers

The PWMCTRL0 and PWMCTRL1 registers control the operating frequency and the resolution of each PWM module.

The following two tables describe the PWM control registers' bit assignment:

TABLE 39: PWM0 CONTROL REGISTER PWMCTRL0–SFR D3H

7	6	5	4	3	2	1	0
-					5BITE	PWMCK1	PWMCK0

Bit	Mnemonic	Description
[7:3]	-	-
2	5BITE	5 Bit PWM operation enable 0 = PWM0 resolution is 8 bit 1 = PWM0 resolution is 5 bit
1	PWMCK1	Clock Frequency Divider Bit 1 for PWM0
0	PWMCK0	Clock Frequency Divider Bit 0 for PWM0

TABLE 40: PWM1 CONTROL REGISTER (PWMCTRL1–SFR D4H)

7	6	5	4	3	2	1	0
-					5BITE	PWMCK1	PWMCK0

Bit	Mnemonic	Description
[7:3]	-	-
2	5BITE	5 Bit PWM operation enable 0 = PWM1 resolution is 8 bit 1 = PWM1 resolution is 5 bit
1	PWMCK1	Clock Frequency Divider Bit 1 for PWM1
0	PWMCK0	Clock Frequency Divider Bit 0 for PWM1

The following table describes the relationship between the values of PWMCK1, PWMCK0 and the numerical divider values of the corresponding frequencies.

PWMCK1	PWMCKO	Divider
0	0	4
0	1	8
1	0	16
1	1	32

The PWM output frequency is calculated using the following two formulas:

$$\text{8-bit PWM Rate} = \frac{F_{osc}}{[256 \times 2^{(PWMCK[1:0]+2)}]}$$

$$\text{5-bit PWM Rate} = \frac{F_{osc}}{[32 \times 2^{(PWMCK[1:0]+2)}]}$$

PWM Data Registers

The following tables describe the PWM0 and PWM1 data registers. The PWMDx bits hold the contents of the PWM data register and determine the duty cycle of the PWM output waveforms.

TABLE 41: PWM0 DATA REGISTER (PWMD0) – SFR B3H

7	6	5	4
PWMD0.7	PWMD0.6	PWMD0.5	PWMD0.4

3	2	1	0
PWMD0.4	PWMD0.3	PWMD0.2	PWMD0.1

Bit	Mnemonic	Description
7	PWMD0[7:0]	PWM0 Data Register

TABLE 42: PWM1 DATA REGISTER (PWMD1) – SFR B4H

7	6	5	4
PWMD1.7	PWMD1.6	PWMD1.5	PWMD1.4

3	2	1	0
PWMD1.4	PWMD1.3	PWMD1.2	PWMD1.1

Bit	Mnemonic	Description
7	PWMD1[7:0]	PWM1 Data Register

PWM Modules Activation

The PWM1 and PWM0 outputs are activated by setting bits 3 and 2 of the I2CPWME register.

TABLE 43: I2CPWME CONFIGURATION REGISTER (I2CPWME, $9B)

7	6	5	4
SDAE	SCLE	-	-

3	2	1	0
PWM1E	PWM0E	-	-

Bit	Mnemonic	Description
7	SDAE	I²C SDA Enable 0: P1.7I/O is dedicated to I²C SDA 1: P1.7 I/O operate as regular I/O
6	SCLE	I²C SCL Enable 0: P1.6I/O is dedicated to I²C SCL 1: P1.6 I/O operate as regular I/O
5	-	
4	-	
3	PWM1E	PWM1 Enable Register 0 = PWM1 module is deactivated 1 = PWM1 module is activated on P1.3
2	PWM0E	PWM1 Enable Register 0 = PWM0 module is deactivated 1 = PWM0 module is activated on P1.3
1	-	
0	-	

Notes on PWM Use

To activate the PWM modules follow the procedure below:

1. Set the PWM data register
2. Set the PWM operating frequency and PWM resolution by setting the PWMCTRLx register
3. Activate the PWM output by setting the corresponding PWMxE bit of the PWMCTRLx register

The selected PWM module operation will begin if:

- The Port 1 I/O pins associated with the PWM outputs are configured to output a logic 0.
- The PWM module is enabled before the PWM data is configured. The PWM output will momentarily go to 0 for a duration equal to the time the PWM was enabled and data will be written into the I2CPWMDx register.
- If the I2CPWMDx register is set to FFh, the PWM will output 1 the entire time, but the output will go low momentarily (1.5 us) on each PWM cycle.

The PWM modules are not data buffered. As such, updating the PWM data register will have an immediate impact on the PWM output.

Interrupts

The VRS51L1050 has seven interrupt sources. The interrupts are enabled via the IE and IEN1 registers shown below:

TABLE 44: 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	Global Interrupt Controller 0: Inhibit all interrupts 1: Each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
6	-	-
5	ET2	Timer 2 Interrupt Enable Bit 0: Timer 2 interrupt is disabled 1: Timer 2 interrupt is enabled
4	ES	UART Serial Port Interrupt Enable Bit 0: UART interrupt is disabled 1: UART interrupt is enabled
3	ET1	Timer 1 Interrupt Enable Bit 0: Timer 1 interrupt is disabled 1: Timer 1 interrupt is enabled
2	EX1	External Interrupt 1 Enable Bit 0: INT1 interrupt is disabled 1: INT1 interrupt is enabled
1	ET0	Timer 0 Interrupt Enable Bit 0: Timer 0 interrupt is disabled 1: Timer 0 interrupt is enabled
0	EX0	External Interrupt 0 Enable Bit 0: INT0 interrupt is disabled 1: INT0 interrupt is enabled

The EA bit of the IE register is the global interrupt controller bit. When the EA bit is cleared, all interrupts will be inhibited. Setting the EA bit to 1 will allow all activated interrupts to reach the interrupt controller.

Bits 5 to 0 of the IE register, as well as bit 2 of the IEN1 register, are individual interrupt enable bits for each peripheral of the VRS51L1050, which can generate an interrupt. Setting one of these bits to 1 will activate the corresponding peripheral interrupt.

TABLE 45: IEN1 INTERRUPT ENABLE REGISTER 1–SFR A9H

7	6	5	4	3	2	1	0
						EI2C	

Bit	Mnemonic	Description
7:2	-	
1	EI2C	I²C Interrupt Enable Bit 0: I²C interrupt is disabled 1: I²C interrupt is enabled
0	-	

The IF1 register holds the I²C interrupt flag.

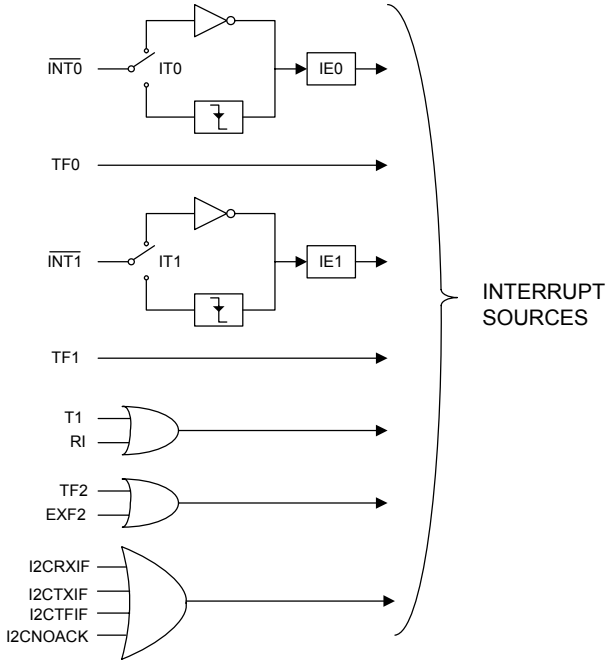
TABLE 46: IF1 I2C INTERRUPT FLAG REGISTER 1–SFR A9H

7	6	5	4	3	2	1	0
						I2CIF	

Bit	Mnemonic	Description
7:2	-	
1	I2CIF	I²C Interrupt Flag
0	-	

The following figure illustrates the various interrupt sources on the VRS51L1050.

FIGURE 22: INTERRUPT SOURCES



Interrupt Vectors

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

TABLE 47: INTERRUPT VECTOR ADDRESS

Interrupt Source	Flag	Vector Address
RESET	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
I²C	I2CRXIF+ I2CTXIF+ I2CTFIF+ I2CNOACK	003Bh

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

External Interrupts

The VRS51L1050 has two external interrupt inputs (INT0 and INT1). These interrupt lines are shared with the P3.2 and P3.3 I/Os. 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 to 0 transition occurs at the interrupt pin.
- If ITx = 0, the interrupt will occur when a logic low condition is present on the interrupt pin. The duration of the low state must be equal to at least 12 oscillator cycles.

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

- If the interrupt is configured as edge sensitive, the associated flag is automatically cleared when the interrupt is serviced.
- If the interrupt is configured as level sensitive, 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 in timers 0 and 1. These interrupt flags are automatically cleared when the interrupt is serviced.

Timer 2 interrupt

A 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 the Timer 2 Counter/Timer occurs. The EXF2 flag can be set by a 1 to 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.

Bits that generate an interrupt can be cleared or set by the software, yielding the same result as when this operation is done by the hardware.

UART Serial Port Interrupt

The serial port can generate an interrupt upon byte reception or once the byte transmission is complete. Those two conditions share the same interrupt vector and it is up to the user-developed interrupt service routine software to ascertain the cause of the interrupt by surveying 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.

I²C Interrupt

One interrupt vector is dedicated to the I²C interface. Either one of the following events can trigger an I²C interrupt if activated:

- I²C data byte received (I2CRXIF)
- I²C data byte transmitted (I2CTXIF)
- I²C data transmission failed (I2CTFIF)
- No acknowledge received (I2CNOACK)

Once the interrupt is serviced, the program should retrieve the I2CSTATUS register to determine which of the events above triggered the I²C interrupt. Once the interrupt source(s) has been identified, the corresponding interrupt flag should be cleared.

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 VRS51L1050 is initialized, all interrupt sources are inhibited by resetting the bits of the IE register to 0. It is necessary to start by enabling the interrupt sources that the application requires by setting bits in the IE register, as discussed previously.

This IE register is part of the bit addressable SFR. 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 the EA bit to 0.

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

TABLE 48: INTERRUPT PRIORITY

Interrupt Source
RESET (Highest Priority)
IE0
TF0
IE1
TF1
RI+TI
TF2+EXF2
I2CRXIF+ I2CTXIF+ I2CTFIF+ I2CNOACK (Lowest Priority)

Modifying the Order of Priority

The VRS51L1050 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 priority over interrupts coming from sources that don't have their corresponding IP bits set to 1.

The IP and IP1 register are represented in the tables below.

TABLE 49: IP INTERRUPT PRIORITY REGISTER –SFR B8H

7	6	5	4	3	2	1	0
-	-	PT2	PS	PT1	PX1	PT0	PX0

Bit	Mnemonic	Description
7	-	
6	-	
5	PT2	Gives Timer 2 interrupt higher priority when set to 1
4	PS	Gives serial port interrupt higher priority when set to 1
3	PT1	Gives Timer 1 interrupt higher priority when set to 1
2	PX1	Gives INT1 interrupt higher priority when set to 1
1	PT0	Gives Timer 0 interrupt higher priority when set to 1
0	PX0	Gives INT0 interrupt higher priority when set to 1

TABLE 50: IP1 INTERRUPT PRIORITY REGISTER 1–SFR B9H

7	6	5	4	3	2	1	0
						PI2C	

Bit	Mnemonic	Description
7	-	
6	-	
5	-	
4	-	
3	-	
2		
1	PI2C	Gives I²C interrupt higher priority when set to 1
0	-	

If the interrupt of more than one peripheral is configured with a high priority level and more than one of these interrupt occurs simultaneously, the natural priority among those interrupt will apply in the interrupt servicing.

Reduced EMI Function

The VRS51L1050 can also be set up for reduced 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.

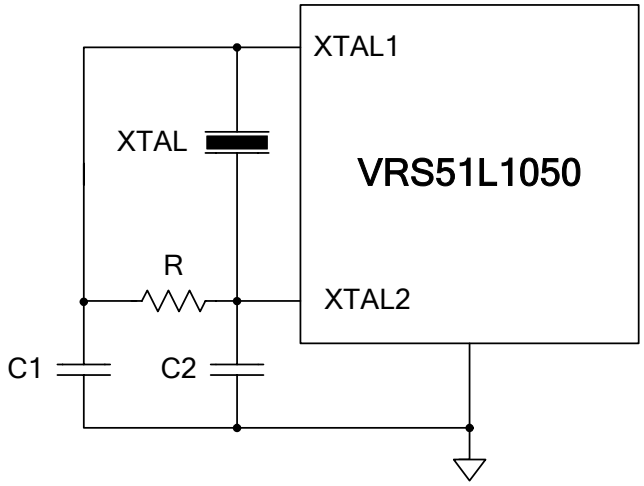
Crystal consideration

The crystal connected to the VRS51L1050 oscillator input should be of a parallel type, operating in fundamental mode. The following table provides suggested capacitor and resistor feedback values for different operating frequencies:

Valid for VRS51L1050				
XTAL	3MHz	6MHz	9MHz	12MHz
C1	30 pF	30 pF	30 pF	22 pF
C2	30 pF	30 pF	30 pF	22 pF
R	-	-	-	-
XTAL	16MHz	25MHz		
C1	30 pF	15 pF		
C2	30 pF	15 pF		
R	-	-		

Note: Oscillator circuits may differ with different crystals or ceramic resonators in higher oscillator frequencies. Crystals or ceramic resonator characteristics vary from one manufacturer to the other.

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



Operating Conditions

TABLE 51: OPERATING CONDITIONS

Symbol	Description	Min.	Typ.	Max.	Unit	Remarks
TA	Operating temperature	0	25	+70	ºC	Ambient temperature, operating
TS	Storage temperature	-55	25	155	ºC	Possible damage to devices
VCC5	Supply voltage	3.0	3.3	3.3	V	
Fosc 40	Oscillator Frequency	-	-	25	MHz	At 3.3V
IOLpin	Maximum Output current IOL per I/O pin			10	mA	
IOLP0	Maximum Output current IOL Port 0 all I/O pins			26	mA	
IOLP1234	Maximum Output current IOL Port 1,2,3,4 all I/O pins			15mA		
iOLALLIO	Maximum Output current IOL ALL I/O pins			71	mA	

DC Characteristics

TABLE 52: DC CHARACTERISTICS

Symbol	Parameter	Valid	Min.	Max.	Unit	Test Conditions
VIL1	Input Low Voltage	Port 0,1,2,3,4,#EA	-0.5	0.8	V	
VIL2	Input Low Voltage	RES, XTAL1	0	0.8	V	
VIH1	Input High Voltage	Port 0,1,2,3,4,#EA	2.0	VCC+0.2	V	
VI H2	Input High Voltage	RES, XTAL1	70% VCC	VCC+0.2	V	
VOL1	Output Low Voltage	Port 0, ALE, #PSEN		0.4	V	IOL=3.2mA
VOL2	Output Low Voltage	Port 1,2,3,4		0.4	V	IOL=1.6mA
VOH1	Output High Voltage	Port 0, ,ALE,#PSEN	2.4		V	IOH=-300uA
VOH2	Output High Voltage	Port 1,2,3,4	2.4		V	IOH=-20uA
IIL	Logical 0 Input Current	Port 1,2,3,4 (except P1.6, P1.7)		-50	uA	Vin=0.45V
IIL2	Logical 0 Input Current	Port 0 and P1.6, P1.7		-650	uA	Vin=0.45V
ITL	Logical Transition Current	Port 1,2,3,4		-650	uA	Vin=1.5V
ILI	Input Leakage Current			±10	uA	0.45V<Vin<VCC
ISK1	Sink current Port 1,2,3,4	VIN = 0.4V	3	6		mA
ISK2	Sink current Port0, ALE,#PSEN	VIN = 0.4V	4	8		mA
ISRC1	Source current Port 1,2,3,4	VIN = 2.4V	-40	-80		uA
ISRC2	Source current Port0, ALE,#PSEN	VIN = 2.4V	-4	-8		mA
R RES	Reset Pull-down Resistance	RES	50	300	Kohm	VCC = 3.6V
C˜10	Pin Capacitance			10	pF	Fre=1 MHz, Ta=25˚C
IC C	Power Supply Current	VDD		10	mA	Active mode 25MHz, 3.6V
				5	mA	Idle mode 12MHz
				20	uA	Power down mode (all outputs pin disconnected)

Note:

- The supply current is measured with all output disconnected and XTAL1 driven with 5ns rise/fall time with amplitude of 0.5V to VCC+0.5V. XTAL2 pin not connected, EA, RESET,Port0 connected to VDD
- The I/O port pins source a transition current when they are externally driven from high to low and the transition current reaches a maximum of around 2V.
- Capacitive load on Port 0 and Port 2 may cause spurious noise to occur on ALE and other I/O ports when they output on logic low level. In some cases (Cload > 100pF), the noise on ALE may exceed 0.8V. In those cases, it is recommended to buffer the ALE with a Schmitt trigger type logic device or use an address with a Schmitt trigger input.

AC Characteristics

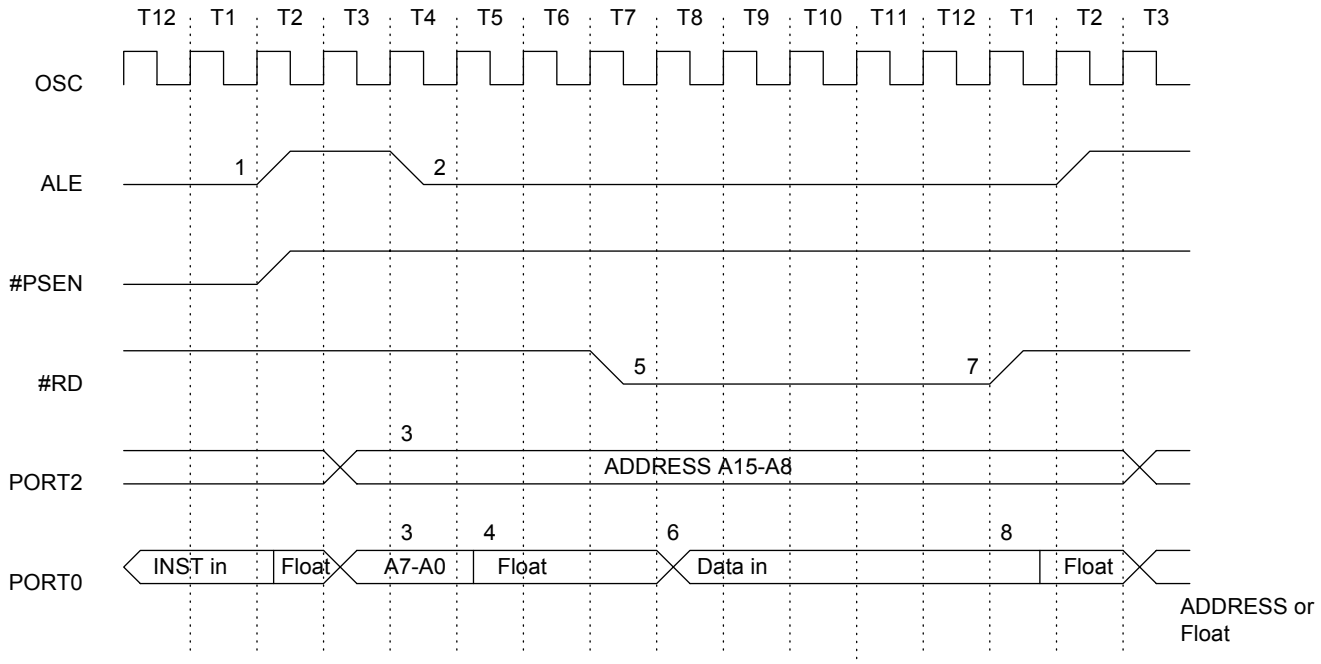
TABLE 53: AC CHARACTERISTICS

Symbol	Parameter	Valid Cycle	Variable Fosc			Unit
			Min.	Type	Max.	
T LHLL	ALE Pulse Width	RD/WRT	2xT - 40			nS
T AVLL	Address Valid to ALE Low	RD/WRT	T - 40			nS
T LLAX	Address Hold after ALE Low	RD/WRT	T - 30			nS
T LLIV	ALE Low to Valid Instruction In	RD			4xT - 100	nS
T LLPL	ALE Low to #PSEN low	RD	T - 30			nS
T PLPH	#PSEN Pulse Width	RD	3xT - 45			nS
T PLIV	#PSEN Low to Valid Instruction In	RD			3xT -105	nS
T PXIX	Instruction Hold after #PSEN	RD	0			nS
T PXIZ	Instruction Float after #PSEN	RD			T - 25	nS
T AVI V	Address to Valid Instruction In	RD			5xT - 105	nS
T PLAZ	#PSEN Low to Address Float	RD			10	nS
T RLRH	#RD Pulse Width	RD	6xT - 100			nS
T WLWH	#WR Pulse Width	WRT	6xT - 100			nS
T RLDV	#RD Low to Valid Data In	RD			5xT - 165	nS
T RHDX	Data Hold after #RD	RD	0			nS
T RHDZ	Data Float after #RD	RD			2xT - 70	nS
T LLDV	ALE Low to Valid Data In	RD			8xT - 150	nS
T AVDV	Address to Valid Data In	RD			9xT - 165	nS
T LLYL	ALE low to #WR High or #RD Low	RD/WRT	3xT - 50		3xT + 50	nS
T AVYL	Address Valid to #WR or #RD Low	RD/WRT	4xT - 130			nS
T QVWH	Data Valid to #WR High	WRT	7xT - 150			nS
T QVWX	Data Valid to #WR Transition	WRT	T - 50			nS
T WHQX	Data Hold after #WR	WRT	T - 50			nS
T RLAZ	#RD Low to Address Float	RD			0	nS
T YALH	#W R or #RD High to ALE High	RD/WRT	T -40		T+40	nS
	Serial Clock Time		12xT			
	Output data setup to clock rising edge		10xT-133			nS
	Output data hold after clock rising edge		2xT – 117			nS
	Input data hld after clock rising edge		0			nS
	Clock rising edge to input data valid				10xT-133	nS
T,TCLCL	Clock Period			1/fosc		nS

Data Memory Read Cycle Timing

The following timing diagram provides data memory read cycle timing information.

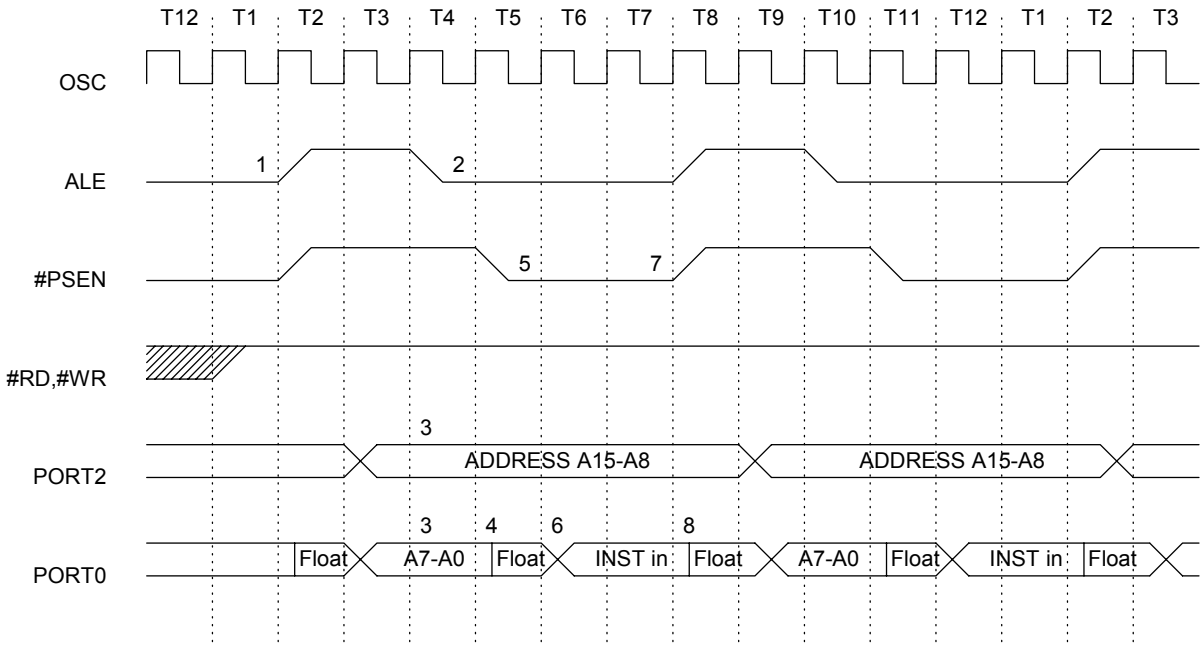
FIGURE 23: DATA MEMORY READ CYCLE TIMING



Program Memory Read Cycle Timing

The following timing diagram provides program memory read cycle timing information

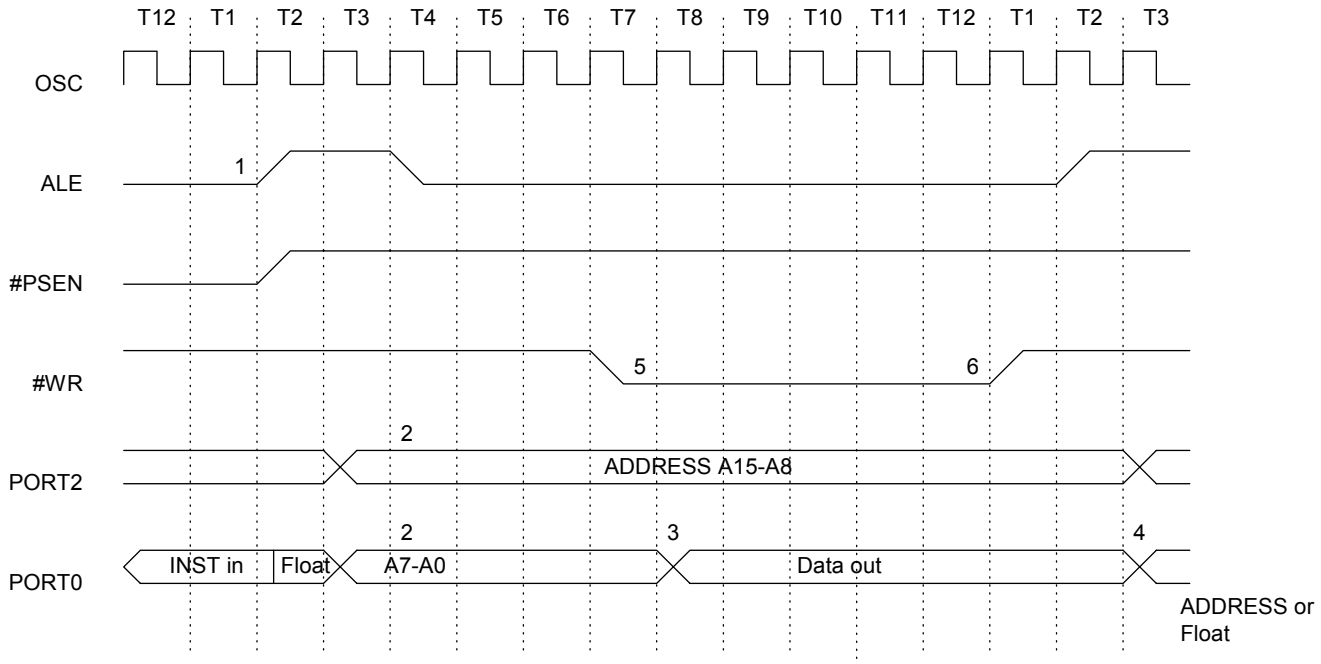
FIGURE 24: PROGRAM MEMORY READ CYCLE



Data Memory Write Cycle Timing

The following timing diagram provides data memory write cycle timing information.

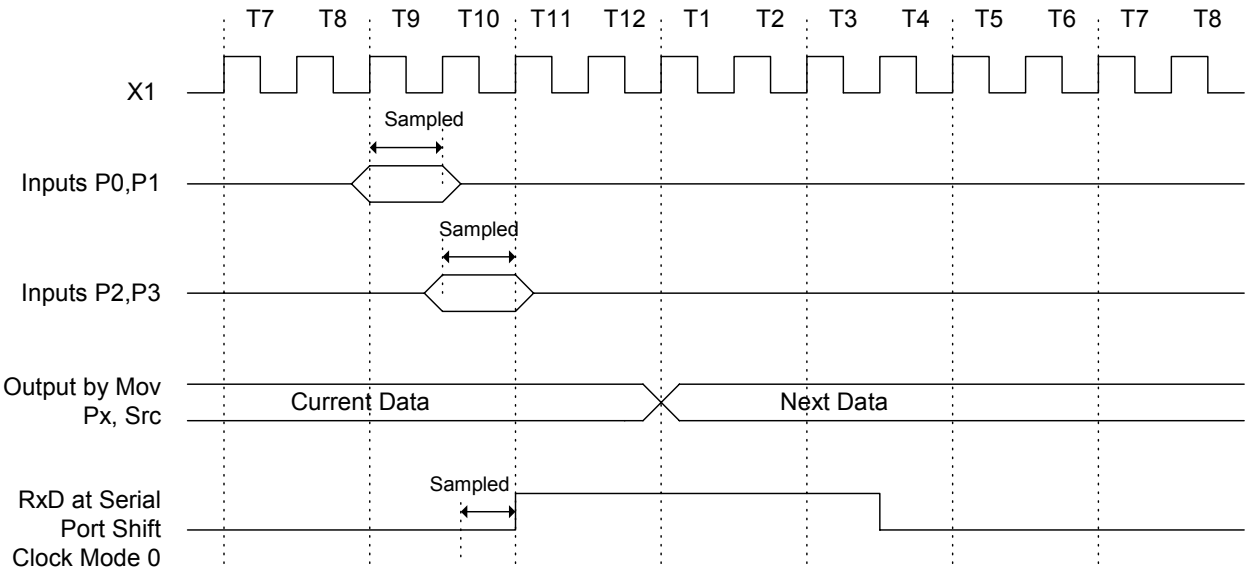
FIGURE 25: DATA MEMORY WRITE CYCLE TIMING



I/O Port Timing

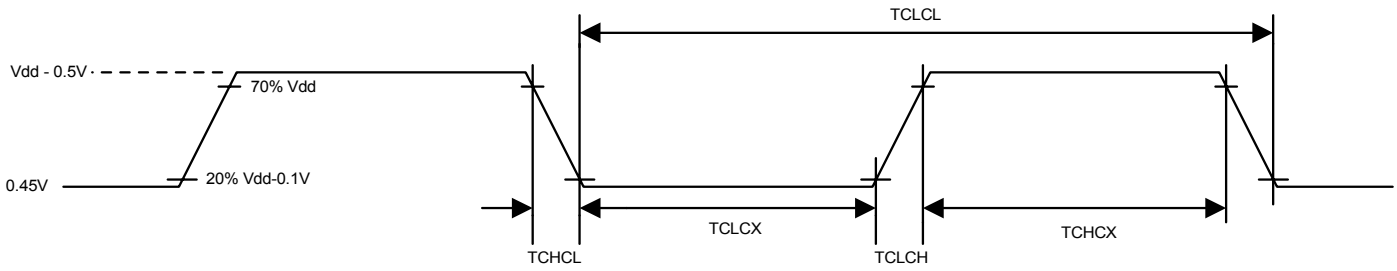
The following timing diagram provides I/O port timing information.

FIGURE 26: I/O PORTS TIMING



Timing Requirement for External Clock (VSS = 0v Assumed)

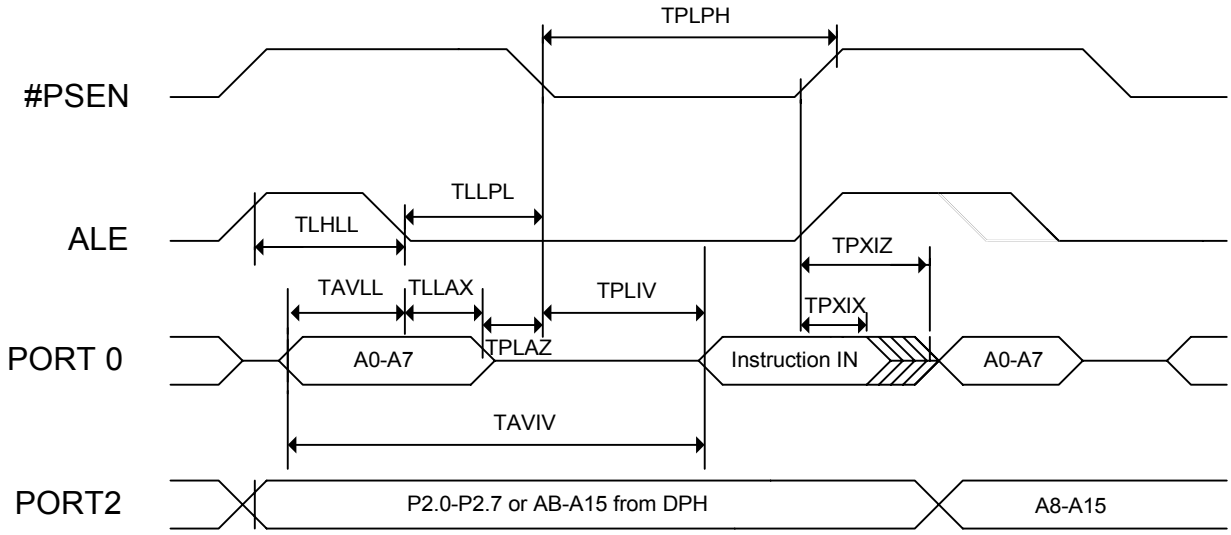
FIGURE 27: TIMING REQUIREMENT OF EXTERNAL CLOCK (VSS= 0.0V IS ASSUMED)



External Program Memory Read Cycle

The following timing diagram provides external program memory read cycle timing information.

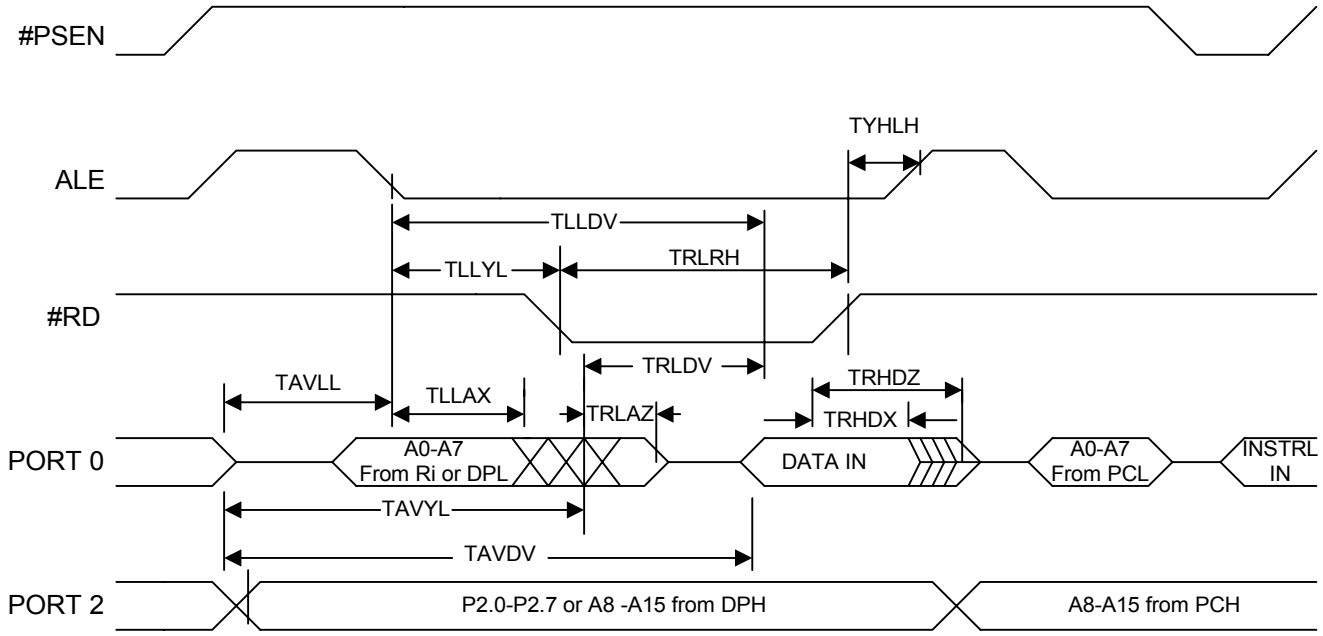
FIGURE 28: EXTERNAL PROGRAM MEMORY READ CYCLE



External Data Memory Read Cycle

The following timing diagram provides external data memory read cycle timing information.

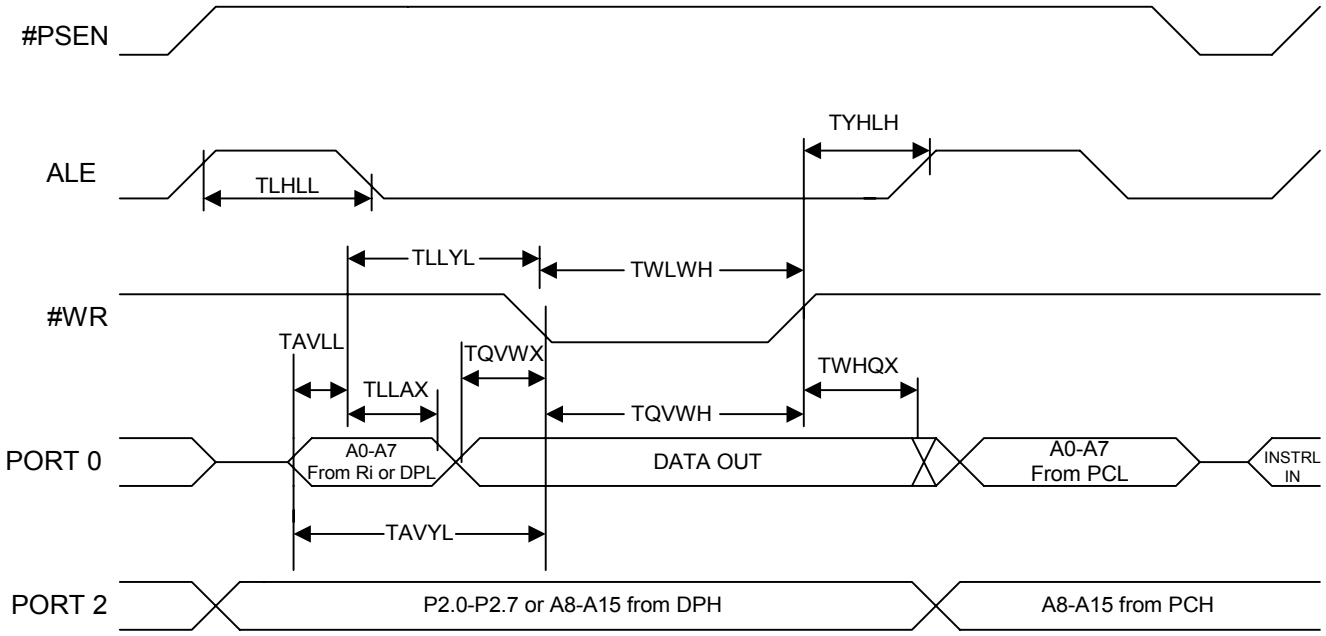
FIGURE 29: EXTERNAL DATA MEMORY READ CYCLE



External Data Memory Write Cycle

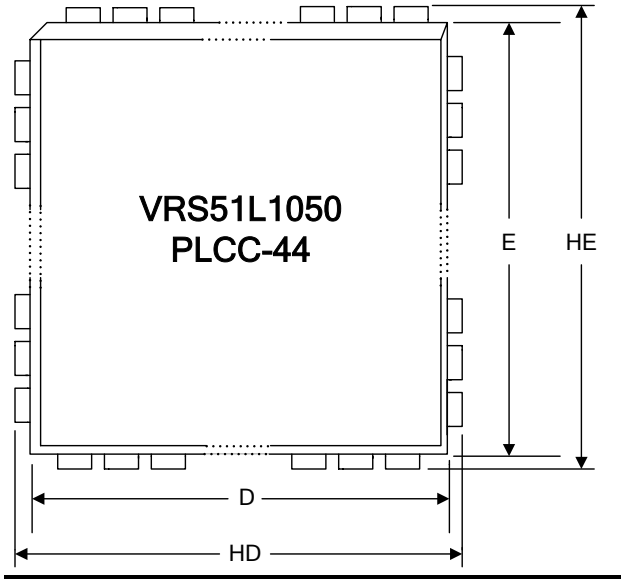
The following timing diagram provides external data memory write cycle timing information.

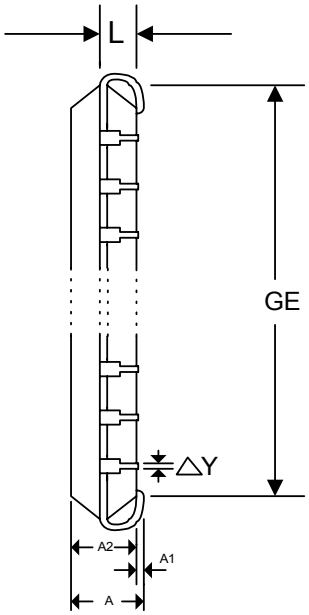
FIGURE 30: EXTERNAL DATA MEMORY WRITE CYCLE

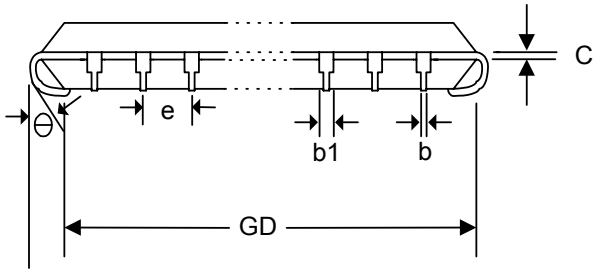


.

Plastic Chip Carrier (PLCC-44)





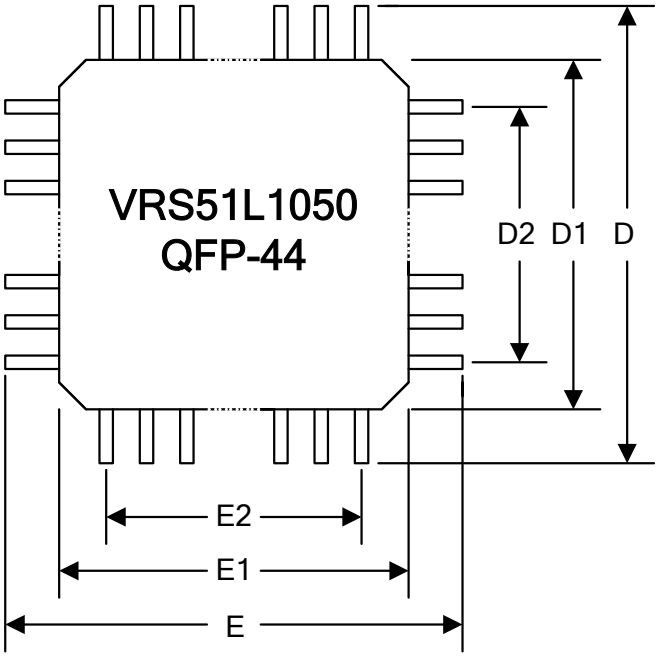


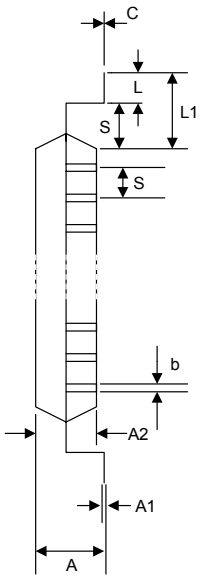
Note:
1. Dimensions D & E do not include interlead Flash.
2. Dimension B1 does not include dambar protrusion/intrusion.
3. Controlling dimension: Inch
4. General appearance spec should be based on final visual inspection spec.

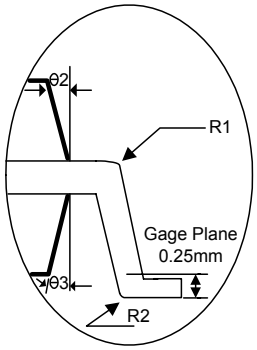
TABLE 54: DIMENSIONS OF PLCC-44 CHIP CARRIER

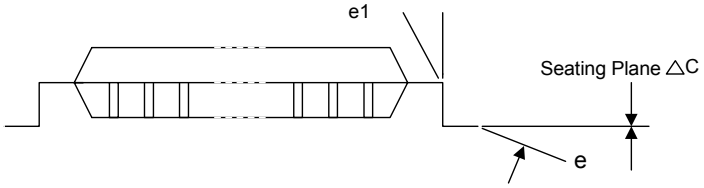
Symbol	Dimension in inch	Dimension in mm
	Minimal/Maximal	Minimal/Maximal
A	-/0.185	-/4.70
Al	0.020/-	0.51/
A2	0.145/0.155	3.68/3.94
bl	0.026/0.032	0.66/0.81
b	0.016/0.022	0.41/0.56
C	0.008/0.014	0.20/0.36
D	0.648/0.658	16.46/16.71
E	0.648/0.658	16.46/16.71
e	0.050 BSC	1.27 BSC
GD	0.590/0.630	14.99/16.00
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
∆y	/	/

Plastic Quad Flat Package (QFP-44)









Note:

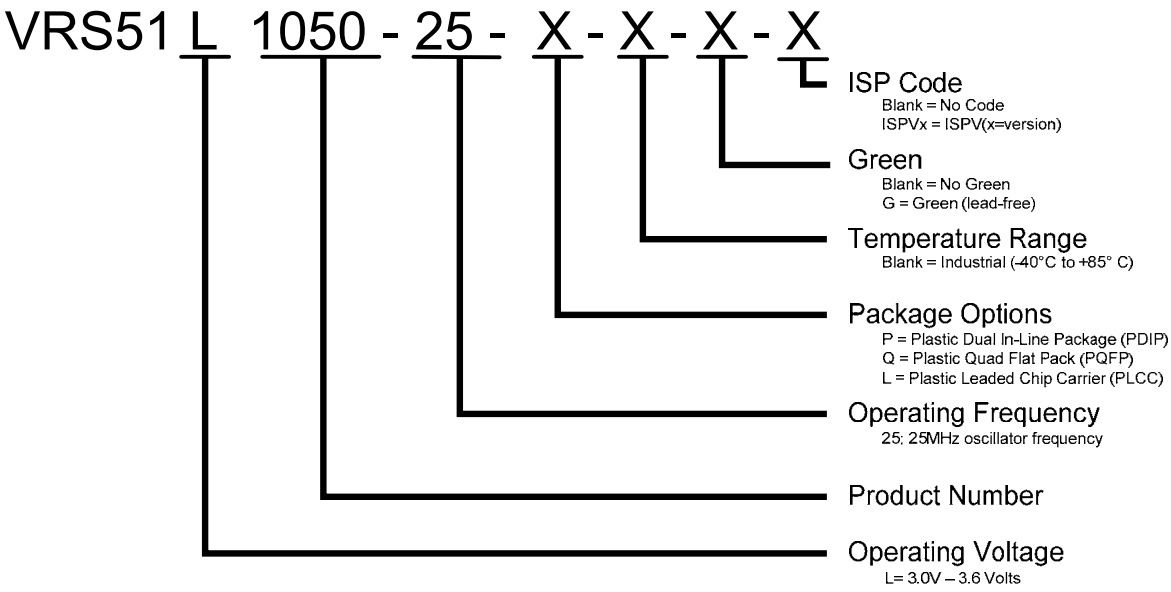
1. Dimensions D1 and E1 do not include mold protrusion.
2. Allowance protrusion is 0.25mm per side.
3. Dimensions D1 and E1 do not include mold mismatch and are determined datum plane.
4. Dimension b does not include dambar protrusion.
5. Allowance dambar protrusion shall be 0.08 mm total in excess of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

TABLE 55: 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

Ordering Information

Device Number Structure



VRS51L1050 Ordering Options (No ISPVx Firmware preprogrammed)

Device Number	Flash Size	SRAM Size	Package Option	Voltage	Temperature	Frequency
VRS51L1050-25-L	64KB	1KB	PLCC-44	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-Q	64KB	1KB	QFP-44	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-P	64KB	1KB	DIP-40	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-LG	64KB	1KB	PLCC-44	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-QG	64KB	1KB	QFP-44	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-PG	64KB	1KB	DIP-40	3.0V to 3.6V	0°C to +70°C	25MHz

VRS51L1050 Ordering Options (With ISPVx Firmware preprogrammed). See Ramtron web site for latest version - x).

Device Number	Flash Size	SRAM Size	Package Option	Voltage	Temperature	Frequency
VRS51L1050-25-L-ISPV3	64KB	1KB	PLCC-44	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-Q-ISPV3	64KB	1KB	QFP-44	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-P-ISPV3	64KB	1KB	DIP-40	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-LG-ISPV3	64KB	1KB	PLCC-44	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-QG-ISPV3	64KB	1KB	QFP-44	3.0V to 3.6V	0°C to +70°C	25MHz
VRS51L1050-25-PG-ISPV3	64KB	1KB	DIP-40	3.0V to 3.6V	0°C to +70°C	25MHz

Disclaimers

Right to make change - Ramtron 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 - Ramtron 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 Ramtron parts. Ramtron assumes no liability for applications assistance or customer product design.
Life support – Ramtron products are not designed for use in life support systems or devices. Ramtron customers using or selling Ramtron products for use in such applications do so at their own risk and agree to fully indemnify Ramtron for any damages resulting from such applications.

I²C is a trademark of Koninklijke Philips Electronics NV.