INTRODUCTION TO ST8 FAMILY

ST8 CONCEPT

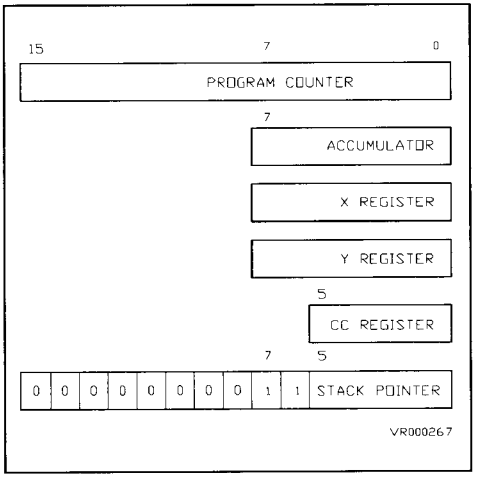
ST8 is defined as a family of high performance, low cost HCMOS 8 bit microcontrollers, using a modular block architecture design.

The ST8 concept is the association of an easy-to-use, powerful, high-speed 8 bit core with a library of function blocks such as ROM, RAM, EPROM, EEPROM, I/O PORTS, SERIAL INTERFACES, 8 and 16 bit TIMERS, 8 bit A/D CONVERTER, 8 bit D/A PWM's and possible CUSTOMER DEDICATED FUNCTIONS.

ST8 CORE

The ST8 core is an 8 bit Accumulator based machine. Two additionnal X and Y registers are dedicated to indexed addressing. A 16 bit Program counter allows addressing of to 64 K bytes of memory. The 6 bit stack pointer gives 64 levels of depth and can be upgraded to 8 bit (256 levels) in future versions. A condition Code Register includes 5 Condition bits that indicate the result of the last instruction executed.

ST8 CORE ARCHITECTURE



The instruction set is 8 bit data oriented and offers - in addition to the standard data movement, logic and arithmetic functions - bit oriented instructions and 8 by 8 multiplication. 7 main addressing modes allow programming, flexibility.

ST8 FUNCTION BLOCK LIST

ST8 blocks are hardware peripherals that can be placed around the core in order to compose an original single chip microcontroller, best suited for hi-integration low-cost applications. At the time of printing, the following blocks are available:

a) **ROM** = read only program memory. ROM can reach 32 K byte size on-chip.

b) **RAM** = random acces memory for data. Standard RAM sizes are 64, 128, 176, 256, 384 and 512 bytes

c) **EPROM** = erasable PROM.

Used especially for building EPROM emulation parts for the corresponding ROM devices. Standard EPROM sizes are 2K, 4K, 8K, 16K and 32K bytes.
EPROM window packages are dedicated to development and pre-series, while
OTP (One Time Programmable) versions can be used for small production

d) **EEPROM** = Electrically Erasable PROM. Used especially to keep data on-chip when power supply is off. Standard sizes are 48, 64, 128, 256 and 512 bytes. On-chip charge pump circuity provides the right internal voltage for EEPROM writing / erasing cycles.

e) **TIMER 1** = 16 bit free running counter system associated with one Input Capture Register / Input Capture Pin, and one Output Compare Register / Output Compare Pin.

f) **TIMER 2** = 16 bit free running counter system associated with two Input Capture Register / Input Capture Pin, and two Output Compare Register / Output Compare Pin.

g) **D / A CONVERTER** = dual Pulse Width Modulation systems able to generate two independant PWM outputs. The duty cycle is 8 bit programmable (256 steps from 0% to 100%). This unit must necessarily be associated with a TIMER in order to receive the proper clocking signals from a free running counter.

h) **A / D CONVERTER** = 8 bit analog to Digital converter with 1/2 lsb resolution within 32 machine cycles total conversion time. Two independant input pins allow input of external low and high analog references.

i) **SERIAL PERIPHERAL INTERFACE (SPI)** = synchronous serial interface, full duplex 3 wires system. Master & Slave modes are available.

j) **SERIAL COMMUNICATION INTERFACE (SCI)** = asynchronous serial interface, full duplex, NRZ format, programmable baud rates and standard error detection.

k) **WATCHDOG COUNTER** = resetable by 8 bit counter that can generate an on-chip Reset after a programmable time-out.

l) **I/ O PORTS** = standard output, input or bi-directionnal ports, TTL & CMOS compatible. The basic block has 8 lines.

8K ROM HCMOS MICROCONTROLLER

HARDWARE

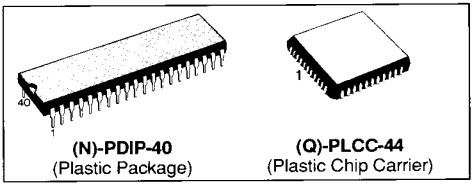
- HCMOS TECHNOLOGY.
- 8 BIT CORE & ARCHITECTURE.
- POWER SAVING WAIT, HALT AND RAM RETENTION MODES.
- 7744 BYTES OF USER ROM.
- 176 BYTES OF RAM.
- 24 BI-DIRECTIONNAL I/O LINES.
- 16 BIT FREE RUNNING COUNTER TIMER FEATURING ONE INPUT CAPTURE SYSTEM AND ONE OUTPUT COMPARE SYSTEM.
- SERIAL PERIPHERICAL INTERFACE (SYNCHRONOUS).
- SERIAL COMMUNICATION INTERFACE (ASYNCHRONOUS).
- EXTERNAL, TIMER, SCI AND SPI INTERRUPTS.
- MASTER RESET AND POWER ON RESET.
- SINGLE 3 TO 6 VOLTS SUPPLY.
- 2 VOLTS RAM RETENTION MODE
- USER MASK OPTIONS :
  - -RC or XTAL / CERAMIC oscillator option
  - -Interrupt trigger : edge or level & edge
  - -Internal clock for TIMER, SCI and SPI.
- 40 PIN DUAL-IN-LINE PACKAGE
- 44 LEAD PLCC

SOFTWARE

- 8 BIT DATA MANIPULATION.
- 74 BASIC INSTRUCTIONS.
- 8 BY 8 UNSIGNED MULTIPLY INSTRUCTION.
- 7 MAIN ADDRESSING MODES (IMMEDIATE / DIRECT/ EXTENDED / RELATIVE / INDEXED / INDIRECT / BIT).
- DEVELOPMENT SUPPORT ON REAL TIME EMULATOR, EPROM VERSION AND PC/DOS SOFTWARE (CROSS ASSEMBLER, DEBUGGER).

DESCRIPTION

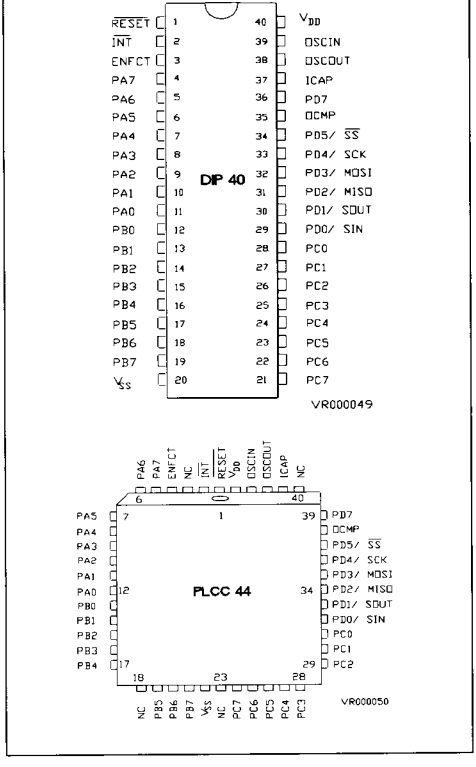
The ST8108 is a complete HCMOS microcontroller unit (MCU); The device includes an on-chip oscillator, CPU, ROM, RAM, I/O, two serial interface systems (SCI, SPI) and one TIMER. The fully static design allows frequency operation down to DC, reducing power consumption when needed.



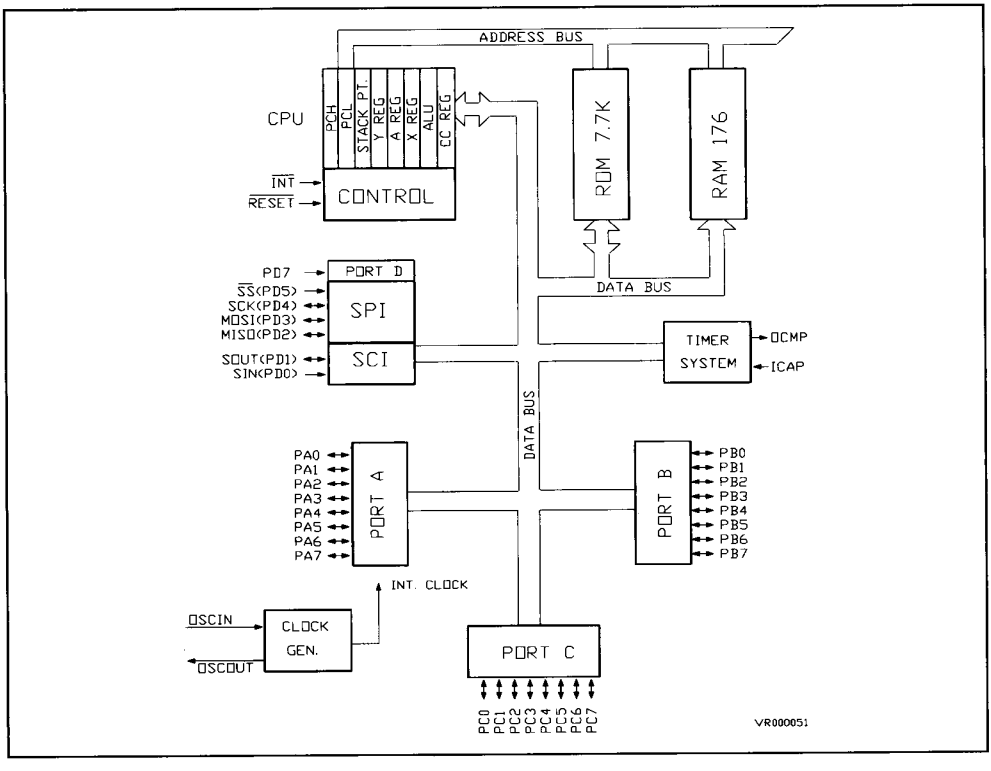
(N)-PDIP-40
(Plastic Package)

(Q)-PLCC-44
(Plastic Chip Carrier)

PIN CONNECTIONS



PART 1 . BLOCK DIAGRAM



PART 2. ST8108 PIN ASSIGNMENT

PIN DESCRIPTION

$\overline{\text{RESET}}$
The ST8108 can be initialized by the $\overline{\text{RESET}}$ input signal, active low. This event is considered as the first priority interrupt for the core. Refer to Part 5 (ST8108 INTERRUPT STRUCTURE) for more detailed information.

$\overline{\text{INT}}$
INT is the external, software maskable interrupt. It can be activated in two different ways (negative edge or negative edge & level sensitive) depending of the User Mask Option. Refer to part 5 (ST8108 INTERRUPT STRUCTURE) for more detailed information.

ENFCT
This pin is reserved for test purposes. The user can leave this pin non-connected, or connect it to Vss or to V_{DD}.

PA0-PA7 / PB0-PB7 / PC0-PC7
Standard bi-directionnal I/O lines. Port A, Port B and Port C are each made of 8 lines. Refer to Part 8 (I/O PORTS) for detailed information.

V_{DD}
Single power supply voltage 3 to 6 volts.

PD0-PD7
SEVEN I/O lines of Port D. PD0-PD5 lines are in alternate function with the SIN & SOUT signals from the Serial Communication Interface and MISO, MOSI, SCK & SS signals from the Serial Peripheral Interface.

Refer to Part 8 for Port D detailed information.
Refer to Part 11 (SCI) for SIN and SOUT signal description.
Refer to Part 10 (SPI) for MISO, MOSI, SCK and SS signal description.

OCMP
This Output Compare signal is provided by the Timer System Output Compare Logic. Refer to Part 9 (16 BIT TIMER) for detailed information.

ICAP
This Input Capture signal is used by the Timer system for signals / Events mesurement purposes. Refer to Part 9 (16 BIT TIMER) for detailed information.

OSCIN, OSCOUT
Oscillator input and output pins. These pins are to be connected to a parallel resonant crystal or ceramic. An external clock source can also be inputed thru OSCIN. Refer to part 6 (CLOCK) for additionnal information about oscillator characteristics.

V_{SS}
Ground

PART 3 . CENTRAL PROCESSING UNIT

3.1 INTRODUCTION

This CPU is an 8 bit microprocessor whose instruction set is defined in PART 4. The fully static design allows operation at frequencies down to DC, further reducing its low-power consumption.

3.1.1 HARDWARE FEATURES

- HCMOS Technology
- 8 bit architecture
- Up to 16 bit address bus
- Six internal registers :
- Accumulator (8 bits)
- 2 Index Registers (8 bits)
- Program counter (up to 16 bits)
- Stack Pointer (up to 8 bits)
- Code Condition Register
- Power saving HALT, WAIT and data retention modes
- Fully static operation

3.1.2 SOFTWARE FEATURES

- 74 basic instructions
- 8X8 unsigned multiply instruction
- 17 addressing modes
- True bit manipulation.
- Two power save standby modes (WAIT, HALT).
- Refer to PART 4. (ST8 instruction set) for a complete description of the instruction set.

3.2 CPU REGISTERS

The CPU contains six registers, as shown in the programming model of Figure 3.1.
Following an interrupt, the registers are stacked in the order shown in Figure 3.2. The Y index register is never stacked.

3.2.1 ACCUMULATOR (A)

The accumulator is an 8 bit general purpose register used to hold operands and results of the arithmetic calculations and to perform data manipulations.

Figure 3.1 . Programming Model

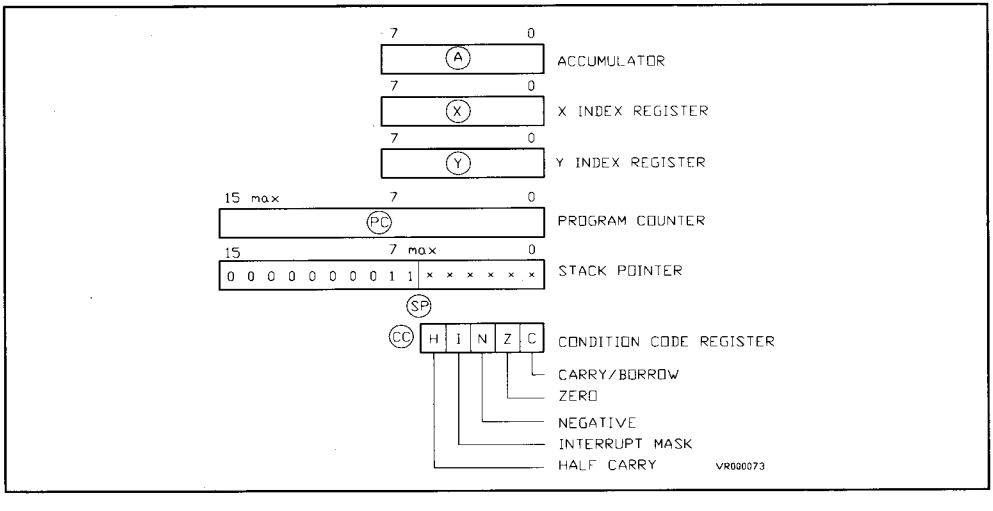
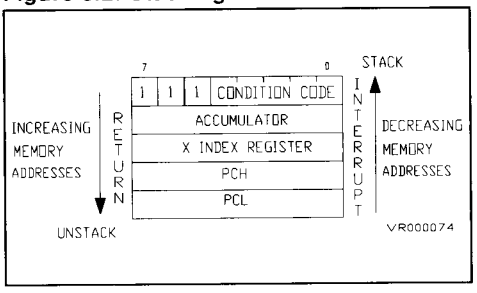


Figure 3.2: Stacking Order



3.2.2 INDEX REGISTER (X AND Y)
These 8-bit registers are used to create an effective address. They are also used for data manipulations with the read-modify-write type of instructions as well as a temporary storage register when not performing addressing operations. To indicate if an instruction refers to the Y index register and not to the X one, a precede instruction (PRE) is generated by the cross assembler. The Y index register is not pushed onto stack when an interrupt occurs.

3.2.3 PROGRAM COUNTER
The program counter is a register of 16 bits max containing the address of the next instruction to be executed by the processor.

3.2.4 STACK POINTER (SP)
The stack pointer is a register of 8 bits max containing the address of the next free location on the push-down/pop-up stack.

NOTE
The stack pointer can be placed either on page 0 ($0000 to $00ff) or on page N ($0N00 to $ONFF). Refer to 3.3 (Memory map) for stack position and depth inside the ram area. The stack area of RAM is used to store the return address on subroutine calls and the machine state during interrupts. During external reset and during a reset stack pointer (RSP) instruction, the stack pointer is set to its upper limit. Nested interrupt and/or subroutines may use up to 256 (decimal) locations, in the case of an 8 bit stack.

When the maximum number of location is exceeded, the stack pointer wraps around and points to its upper limit and loses the previously stored information. A subroutine call occupies two RAM bytes on the stack, while an interrupt uses five RAM bytes.

3.2.5 CONDITION CODE REGISTER (CC)
The condition code register is a 5-bit register which indicates the results of the instruction just executed as well as the state of the processor. These bits can be individually tested by a program and specified action taken as a result of their state. The following paragraphs describe each bit.

Half carry bit (H)
The H bit is set to 1 when a carry occurs between bits 3 and 4 of the ALU during an ADD or ACC instruction. The H bit is used in binary coded decimal subroutines.

Interrupt mask bit (I)
When the I bit is set, all interrupts are disabled. Clearing this bit enables the interrupts. If an external interrupt occurs while the I bit is set, the interrupts are latched and is processed after the I bit are next cleared ; therefore, no interrupts are lost.

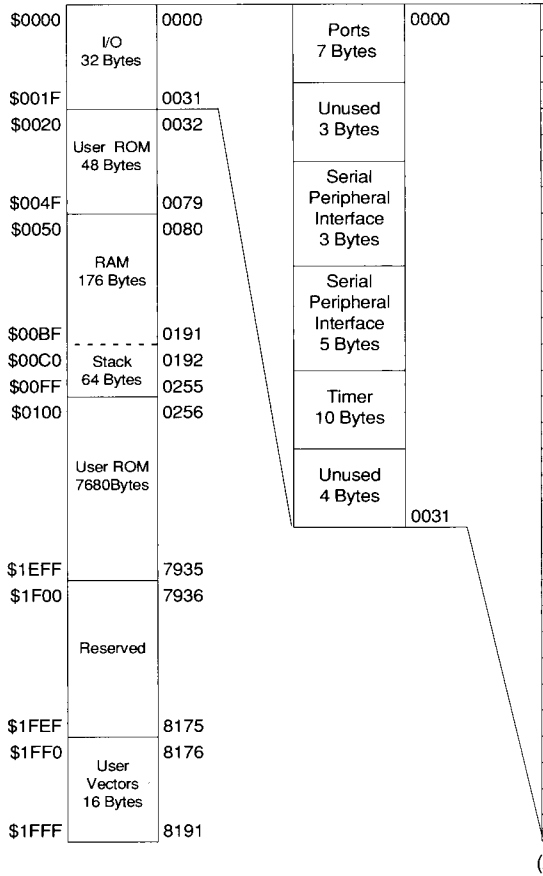
Negative (N)
When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation is negative (bit 7 in the result is a logic one).

Zero (Z)
When set, this bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

Carry/borrow (C)
Indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test, branch instructions, shifts and rotates.

3.3 MEMORY MAP
As shown in Figure 3.3, the MCU is capable of addressing 8192 bytes of memory and I/O registers with its program counter. ST8108 MCU has implemented 8185 bytes of these locations. The first 256 bytes of memory (page zero) include: 25 bytes of I/O features such as data ports, port DDRs, timer, serial peripheral interface (SPI), and serial communication interface (SCI); 48 bytes of user ROM, and 176 bytes of RAM. The next 7680 bytes complete the user ROM.
The highest address bytes contain the user defined reset and the interrupt vectors. The 176 bytes of user RAM include up to 64 bytes for the stack. Since most programs use only a small part of the allocated stack locations for interrupts and/or subroutine stacking purposes, the unused bytes are usable for program data storage.

3.3 MEMORY MAP



Port A Data Register	$00
Port B Data Register	$01
Port C Data Register	$02
Port D Fixed Input Register	$03
Port A Data Direction Reigster	$04
Port B Data Direction Register	$05
Port C Data Direction Register	$06
Unused	$07
Unused	$08
Unused	$09
Serial Peripheral Control Register	$0A
Serial Peripheral Status Register	$0B
Serial Peripheral Data I/O Register	$0C
Serial Communications Baud Rate Register	$0D
Serial Communications Control Register 1	$0E
Serial Communications Control Register 2	$0F
Serial Communications Status Register	$10
Serial Communications Data Register	$11
Timer Control Register	$12
Timer Status Register	$13
Input Capture High Register	$14
Input Capture Low Register	$15
Output Compare High Register	$16
Output Compare Low Register	$17
Counter High Register	$18
Counter Low Register	$19
Alternate Counter High Register	$1A
Alternate Counter Low Register	$1B
Unused	$1C
Unused	$1D
Unused	$1E
Unused	$1F

(All unused Registers are not available to the

PART 4 . INSTRUCTION SET

4.1 INSTRUCTION SET

Note: This chapter is an overview of the ST8 family instruction set. Refer to SGS-THOMSON appropriate documentation (ST8 macro assembler user's guide / ST8 programming manual) for detailed information.

THE ST8 INSTRUCTION SET IS AN 8 BIT DATA BASED INSTRUCTION SET THAT CAN BE DIVIDED INTO FIVE MAJOR GROUPS:

GROUP 1 = REGISTER / MEMORY AND ABSOLUTE JUMP GROUP
In this group of instructions, the operands can be the accumulator, the X index register, the Y index register or any "effective memory address" obtained by the different addressing modes.
Examples: - LD <ea>, a - means that the memory byte located at address <ea> is loaded with the 8 bit content of the accumulator a.

GROUP 2 = READ - MODIFY - WRITE GROUP
These instructions can read a register or a memory location, modify its content and write the new value back.
Example : - RRC <ea> - means that the content of the memory byte located at address <ea> is rotated right through the carry bit, result written in memory <ea> and carry bit.

GROUP 3 = BIT MANIPULATION AND TEST GROUP
Test instructions can test any bit of the first 256 memory locations and jump conditional within an 8 bit pc-relative displacement.
Example : - BTJT <ea>, #b, ee - corresponds to the relative jump (displacement = ee) if bit number #b of memory location <ea> is set. (Bit test and jump if true).

Bit manipulation instructions can set or reset any bit within the first 256 memory locations, except for ROM ($20-$4F) and registers located at addresses $03, $0B, $10, $13, $14 and $15.
Example: - BSET <ea>, #b - sets the bit #b of memory location <ea>.

GROUP 4 = PC-RELATIVE JUMP GROUP
These instructions execute a pc-relative jump (8bit displacement) depending on the state of flag bits inside the condition code register (H, I, N, Z, C flags).
Example: - JRC ee - jump relative if carry, displacement = ee

GROUP 5 = MISCELLANEOUS GROUP.
These instructions are mainly control instructions on registers, stack, interrupts, subroutines and power down modes. The multiply instruction is also included in this group. It performs an 8 by 8 bit unsigned multiplication between index register and accumulator, result given on 16 bits (on acc. and index registers.

4.2 ADDRESSING MODES:

The CPU uses 17 different addressing modes to provide the programmer with an opportunity to optimize the code to all situations. The various indexed addressing mode make it possible to locate data labels, code conversion tables and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) allow access of tables throughout memory. Short absolute (direct) and long absolute (extended) addressings are also included. Extended addressing permits jump instructions to reach all memory. Tables in 4.3 show the addressing modes for each instructions.

The term "effective address" (EA) is used in describing the various addressing modes, and is defined as the address to or from which the argument for an instruction is fetched or stored*. The 17 addressing modes of the processor are described below. Parentheses are used to indicate "content of" the memory location or register referred to e.g. (PC) indicates the content of the location pointed to by the PC.
*LSBEA is used to represent the least significant byte of EA and MSBEA the most significant one.

USING A PRE-BYTE
In order to extend the number of op-codes available for an eight bit CPU (256 op-codes), three "pre-byte" op-codes have been introduced. These pre-byte have to be seen as "pre-instructions" that modify the meaning of the following instruction.

The whole instruction become:

- n-1 End of previous instruction
- n Pre-byte
- n Op-code
- n ... (operand if needed)
- n+1 Pre-byte or Op-code of next instruction.

These pre-bytes introduce two possibilities:

- Use of Y as the index register instead of the X (pre-byte Y Direct = PDY = $90)
- Use of Indirect addressing mode. Each time the indirect memory addressing mode is selected, a pre-byte must precede the instruction OP-code ($91 or $92). Refer to 4.3 (OP-code tables) for detailed examples. Indirect indexed addressing can be also defined with X as index (PIX = $92) or with Y as index (PIY = $91)

4.2.1 . INHERENT

In inherent instructions, all the information needed to execute the instruction is contained in the Op-code. Operations specifying only the index register or accumulator, and no other arguments, are included in this mode.

4.2.2 . IMMEDIATE

In immediate addressing the operand is stored in the byte immediately following the Op-code.

4.2.3 . DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the Op-code byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two byte instruction.

4.2.4 .EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the op-code. Instructions with extended addressing modes are capable of referencing arguments anywhere in memory.

4.2.5 . INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in one of the 8-bit index register (X or Y). To access Y, the preceding instruction PRE is used prior to the instruction using the indexed, no offset addressing mode. Thus, this addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is used to move a pointer through a table or addressing of a frequently referenced RAM or I/0 location.

4.2.6 . INDEXED, 8BIT OFFSET

The EA is obtained by adding the contents of the second instruction byte to the appropriate index register. This mode allows addressing of 256 locations of the 511 lowest memory locations:

4.2.7. INDEXED, 16 BIT OFFSET

The EA is obtained by adding the 16-bits unsigned value composed by the second (MSB) and the third (LSB) instruction bytes to the appropriate index register. This mode allows addressing of 256 locations anywhere in the memory map.

4.2.8 .RELATIVE

The relative addressing mode is used for branch instruction (e.g. Branch on bit, Branch on condition, Branch subroutine). The branch address (new value of PC) is calculated by adding to the content of the PC, one 8 bit signed value (displacement). This means that the variation of PC value is in the range -126 to +129:

4.2.9 . BIT SET/ CLEAR

Bit Set / Clear mode is used to modify one single bit of a memory location in page zero. This is achieve by a read modify write mode. The position of the bit is given on 3 bits included in the op-code and the memory location is given by the second byte (direct addressing mode) .

4.2.10 . BIT TEST AND BRANCH

Bit test and branch mode gives a relative branch according to the value of a single bit of a memory location in page zero (like Bit Set / Clear mode). Three bytes are needed to specifiy this kind of instruction:

- Op-code which contains the position of the tested bit in the Memory location concerned
- Memory location to be located EA1.
- Displacement (8 bit signed value), if test true, the displacement is added to the content of the PC .

4.2.11 . SHORT INDIRECT

For this mode the effective address is obtained in two steps :

The second byte of the instruction is used as a page zero address. The content of memory location pointed by this address is the effective address :

- LSBEA = ((PC + 1)) ; PC <---PC + 2
- MSBEA = 00

This mode could be called the "indirect direct" mode because the second step is the same as the one of direct mode.

4.2.12 . LONG INDIRECT

In the long indirect mode the second byte of the instruction is used as a zero page pointer. The most significant byte of the EA is the content of this location. The least significant byte is found in the content of the following zero page location (increment the pointer) :

- MSBEA = ((PC + 1)) ; PC <---PC + 2
- LSBEA = ((PC + 1) + 1)

This mode could be call "indirect extended" mode because, after picking the EA, it is similar to the previous extended mode. Three bytes including the pre-byte are needed to describe this mode.

4.2.13 . SHORT INDIRECT INDEXED

For this mode the effective address is obtained in three steps.

The second byte of the instruction is used as a page zero address. The content of the memory location pointed by this address is added to the value of the index register. The result is the EA.

- LSBEA = ((PC + 1)) + Index ; PC <---PC + 2
- MSBEA = Carry
- Index = X or Y according to the pre-byte (PIX or PIY)

4.2.14 . LONG INDIRECT INDEXED

In long indirect indexed mode the second byte of the instruction is used as a zero page pointer. A 16 bit word is read using this pointer (Most significant byte is byte pointed, Least significant byte is found in the content of the following location). The effective address is done by adding the index register value to these two bytes:

- MSBEA = ((PC + 1)) + Carry;
  PC <---PC + 2
- LSBEA = ((PC + 1) + 1) + Index
- Index = X or Y according to the pre-byte (PIX or PIY)

4.2.15 . INDIRECT RELATIVE

The indirect relative addressing mode is working as the relative mode but the displacement is not the content of the second byte of the instruction. This content is used as a zero page address where the displacement is located.

- LSBEA = LSBPC + 2 + ((PC + 1)) ;
  PC <---EA, if branch taken
- MSBEA = MSBPC + CARRY
- Otherwise, EA = PC <---PC + 2

4.2.16 . INDIRECT BIT SET / CLEAR

Indirect Bit Set / Clear mode is working like Bit Set / Clear mode except that the modified byte address is not the content of the second byte of the instruction. This content is used as a zero page address where the concerned byte is located.

4.2.17. INDIRECT BIT TEST AND BRANCH

Indirect Bit test and branch mode works as previous Test and branch mode but the tested byte is addressed as the modify byte in previous Indirect Bit Set / Clear.

4.3. MNEMONICS - OPCODES - CYCLES TABLES

The information given in this chapter is an overview of instruction possibilities. Refer to the "ST8 programming manual" document for complete information.

Each instruction of the five groups discussed in 4.1 is described in the next table.
In these tables the following symbols mean

		a	Accumulator a
		iX, X or Y	Index register (pre-byte 90 if Y)
EFFECTIVE ADDRESS SOURCE CODING		S	Stack pointer
#nn	Immediate	CC	Condition Codes register
ad8	Direct	nn	8 bit immediate value
ad16	Extended	a 8	8 bit address
(iX)	Indexed, no offset	a 16	16 bit address
(d8,iX)	Indexed, 8 bit offset	d8	8 bit signed offset
(d16,iX)	Indexed, 16 bit offset	ee	8 bit PC-relative displacement
(ad8) or (ad16)	Memory indirect, no index	ad	Source coding of address
((ad8) , iX) or ((ad16), iX)	Memory indirect, post indexed	b	3 bit , bit number
		< ea>	Effective address

EXAMPLE

FUNCTION	SOURCE CODING	ADDRESSING MODES Immediate
Load a with memory	LD a,< ea >	2 A6 2

# Bytes	OP-Code	# Cycles

GROUP 1 : REGISTER / MEMORY AND ABSOLUTE JUMP GROUP

FUNCTION	SOURCE CODING	ADDRESSING MODE					
		Immediate	Direct	Extended	Index 0	Index 8	Index 16
Load A with memory	LD a,< ea>	$_2$ A6 ^{2}	$_2$ B6 ^{3}	$_3$ C6 ^{4}	$_1$ F6 ^{3}	$_2$ E6 ^{4}	$_3$ D6 ^{5}
Load iX with memory	LD iX,< ea>	$_2$ AE ^{2}	$_2$ BE ^{3}	$_3$ CE ^{4}	$_1$ FE ^{3}	$_2$ EE ^{4}	$_3$ DE ^{5}
Load memory with A	LD < ea>,a	--	$_2$ B7 ^{4}	$_3$ C7 ^{5}	$_1$ F7 ^{4}	$_2$ E7 ^{5}	$_3$ D7 ^{6}
Load memory with iX	LD < ea>,iX	--	$_2$ BF ^{4}	$_3$ CF ^{5}	$_1$ FF ^{4}	$_2$ EF ^{5}	$_3$ DF ^{6}
Add memory to A	ADD a,< ea>	$_2$ AB ^{2}	$_2$ BB ^{3}	$_3$ CB ^{4}	$_1$ FB ^{3}	$_2$ EB ^{4}	$_3$ DB ^{5}
Add memory and carry to A	ADC a,< ea>	$_2$ A9 ^{2}	$_2$ B9 ^{3}	$_3$ C9 ^{4}	$_1$ F9 ^{3}	$_2$ E9 ^{4}	$_3$ D9 ^{5}
Subtract memory to A	SUB a,< ea>	$_2$ A0 ^{2}	$_2$ B0 ^{3}	$_3$ C0 ^{4}	$_1$ F0 ^{3}	$_2$ E0 ^{4}	$_3$ D0 ^{5}
Subtract memory with carry	SBC a,< ea>	$_2$ A2 ^{2}	$_2$ B2 ^{3}	$_3$ C2 ^{4}	$_1$ F2 ^{3}	$_2$ E2 ^{4}	$_3$ D2 ^{5}
And memory to A	AND a,< ea>	$_2$ A4 ^{2}	$_2$ B4 ^{3}	$_3$ C4 ^{4}	$_1$ F4 ^{3}	$_2$ E4 ^{4}	$_3$ D4 ^{5}
Or memory with A	OR a,< ea>	$_2$ AA ^{2}	$_2$ BA ^{3}	$_3$ CA ^{4}	$_1$ FA ^{3}	$_2$ EA ^{4}	$_3$ DA ^{5}
Exclusive OR	XOR a,< ea>	$_2$ A8 ^{2}	$_2$ B8 ^{3}	$_3$ C8 ^{4}	$_1$ F8 ^{3}	$_2$ E8 ^{4}	$_3$ D8 ^{5}
Arithmetic Compare A	CP a,< ea>	$_2$ A1 ^{2}	$_2$ B1 ^{3}	$_3$ C1 ^{4}	$_1$ F1 ^{3}	$_2$ E1 ^{4}	$_3$ D1 ^{5}
Arithmetic Compare iX	CP iX,< ea>	$_2$ A3 ^{2}	$_2$ B3 ^{3}	$_3$ C3 ^{4}	$_1$ F3 ^{3}	$_2$ E3 ^{4}	$_3$ D3 ^{5}
Bit compare A and memory	BCP a,< ea>	$_2$ A5 ^{2}	$_2$ B5 ^{3}	$_3$ C5 ^{4}	$_1$ F5 ^{3}	$_2$ E5 ^{4}	$_3$ D5 ^{5}
Absolute Jump	JP < ea>	--	$_2$ BC ^{2}	$_3$ CC ^{3}	$_1$ FC ^{2}	$_2$ EC ^{3}	$_3$ DC ^{4}
Call subroutine	CALL < ea>	--	$_2$ BD ^{5}	$_3$ CD ^{6}	$_1$ FD ^{5}	$_2$ ED ^{6}	$_3$ DD ^{7}

GROUP 2 : READ - MODIFY - WRITE GROUP

FUNCTION	ADDRESSING CODING	ADDRESSING MODES						
		Inh a	Inh iX	Direct	Memory indirect	INDEX 0	Index +d8	Index +[ad8]
Increment	INC <ea>	1 4C 3	1 5C 3	2 3C 5	3 923C 7	1 7C 5	2 6C 6	3 926C 8
(Y index)			2 905C 4			2 907C 6	3 906C 7	3 916C 8
Decrement	DEC <ea>	1 4A 3	1 5A 3	2 3A 5	3 923A 7	1 7A 5	2 6A 6	3 926A 8
(Y index)			2 905A 4			2 907A 6	3 906A 7	3 916A 8
Clear	CLR <ea>	1 4F 3	1 5F 3	2 3F 5	3 923F 7	1 7F 5	2 6F 6	3 926F 8
(Y index)			2 905F 4			2 907F 6	3 906F 7	3 916F 8
One's Complement	CPL <ea>	1 43 3	1 53 3	2 33 5	3 9233 7	1 73 5	2 63 6	3 9263 8
(Y index)			2 9053 4			2 9073 6	3 9063 7	3 9163 8
Negate (2's complement)	NEG <ea>	1 40 3	1 50 3	2 30 5	3 9230 7	1 70 5	2 60 6	3 9260 8
(Y index)			2 9050 4			2 9070 6	3 9060 7	3 9160 8
Rotate Left thru Carry	RLC <ea>	1 49 3	1 59 3	2 39 5	3 9239 7	1 79 5	2 69 6	3 9269 8
(Y index)			2 9059 4			2 9079 6	3 9069 7	3 9169 8
Rotate Right thru Carry	RRC <ea>	1 46 3	1 56 3	2 36 5	3 9236 7	1 76 5	2 66 6	3 9266 8
(Y index)			2 9056 4			2 9076 6	3 9066 7	3 9166 8
Shift Left Logical	SLL <ea>	1 48 3	1 58 3	2 38 5	3 9238 7	1 78 5	2 68 6	3 9268 8
(Y index)			2 9058 4			2 9078 6	3 9068 7	3 9168 8
Shift Right Logical	SRL <ea>	1 44 3	1 54 3	2 34 5	3 9234 7	1 74 5	2 64 6	3 9264 8
(Y index)			2 9054 4			2 9074 6	3 9064 7	3 9164 8
Shift Left Arithmetic	SLA <ea>	1 48 3	1 58 3	2 38 5	3 9238 7	1 78 5	2 68 6	3 9268 8
(Y index)			2 9058 4			2 9078 6	3 9068 7	3 9168 8
Shift Right Arithmetic	SRA <ea>	1 47 3	1 57 3	2 37 5	3 9237 7	1 77 5	2 67 6	3 9267 8
(Y index)			2 9057 4			2 9077 6	3 9067 7	3 9167 8
Test for Negative or Zero	TNZ <ea>	1 4D 3	1 5D 3	2 3D 4	3 923D 6	1 7D 4	2 6D 5	3 926D 7
(Y index)			2 905D 4			2 907D 5	3 906D 6	3 916D 7
Swap Nibbles	SWAP <ea>	1 4E 3	1 5E 3	2 3E 5	3 923E 7	1 7E 5	2 6E 6	3 926E 8
(Y index)			2 905E 4			2 907E 6	3 906E 7	3 916E 8

GROUP 3 : BIT MANIPULATION AND TEST GROUP

FUNCTION	SOURCE CODING	ADDRESSING MODES	
		Direct	Memory Indirect
		ad8	[ad8]
Bit Set	BSET < ea > , # b	$_2$ (10+2*b) [5]	$_3$ 92(10+2*b) [7]
Bit Reset	BRES < ea > , # b	$_2$ (11+2*b) [5]	$_3$ 92(10+2*b) [7]
Bit Test and Jump if True	BTJT < ea > , # b , ee	$_3$ (00+2*b) [5]	$_4$ 92(00+2*b) [7]
Bit Test and Jump if False	BTJF < ea > , # b , ee	$_3$ (01+2*b) [5]	$_4$ 92(01+2*b) [7]

GROUP 4 : PC-RELATIVE JUMP GROUP

FUNCTION	SOURCE CODING		ADDRESSING MODES	
			Direct	Memory Indirect
			ad	+[ad8] [ad8]
Jump Relative True	JRT	ee	$_2$ 20 [3]	$_3$ 9220 [5]
(Jump Relative always)	JRA	ee	$_2$ 20 [3]	$_3$ 9220 [5]
Jump Relative False	JRF	ee	$_2$ 21 [3]	$_3$ 9221 [5]
Jump Relative if Unsigned Greater than	JRUGT	ee	$_2$ 22 [3]	$_3$ 9222 [5]
Jump Relative if Unsigned Lower or Equal	JRULE	ee	$_2$ 23 [3]	$_3$ 9223 [5]
Jump Relative if No Carry	JRNC	ee	$_2$ 24 [3]	$_3$ 9224 [5]
Jump Relative if Unsigned Greater or Equal	JRUGE	ee	$_2$ 24 [3]	$_3$ 9224 [5]
Jump Relative if Carry	JRC	ee	$_2$ 25 [3]	$_3$ 9225 [5]
Jump Relative if Unsigned Lower than	JRULT	ee	$_2$ 25 [3]	$_3$ 9225 [5]
Jump Relative if Not Equal	JRNE	ee	$_2$ 26 [3]	$_3$ 9226 [5]
Jump Relative if Equal	JREQ	ee	$_2$ 27 [3]	$_3$ 9227 [5]
Jump Relative if Half Carry	JRH	ee	$_2$ 28 [3]	$_3$ 9228 [5]
Jump Relative if Not Half Carry	JRNH	ee	$_2$ 29 [3]	$_3$ 9229 [5]
Jump Relative if Plus	JRPL	ee	$_2$ 2A [3]	$_3$ 922A [5]
Jump Relative if Minus	JRMI	ee	$_2$ 2B [3]	$_3$ 922B [5]
Jump Relative if Not Interrupt Mask	JRNM	ee	$_2$ 2C [3]	$_3$ 922C [5]
Jump Relative if Interrupt Mask	JRM	ee	$_2$ 2D [3]	$_3$ 922D [5]
Jump Relative if Interrupt Line Low	JRIL	ee	$_2$ 2E [3]	$_3$ 922E [5]
Jump Relative if Interrupt Line High	JRIH	ee	$_2$ 2F [3]	$_3$ 922F [5]
Call Subroutine Relative	CALLR	ee	$_2$ AD [6]	$_3$ 92AD [8]

GROUP 5 : MISCELLANEOUS GROUP

FUNCTION	SOURCE CODING		NO INDEX OR X	Y INDEX
Multiply (iX, A = iX * A)	MUL	iX , a	1 42 [11]	2 9042 [12]
Load iX with acc. a content	LD	iX , a	1 97 [2]	2 9097 [3]
Load acc. a with iX content	LD	a , iX	1 9F [2]	2 909F [3]
Load Stack p. with acc. a content	LD	S , a	1 95 [2]	--
Load acc. a with Stack p. content	LD	a , S	1 9E [2]	--
Load Stack p. with iX content	LD	S , iX	1 94 [2]	2 9094 [3]
Load iX with Stack p. content	LD	iX , S	1 96 [2]	2 9096 [3]
Load X reg. with Y reg. content	LD	X , Y	1 93 [2]	--
Load Y reg. with X reg. content	LD	Y , X	--	2 9093 [3]
Push acc. a onto the Stack	PUSH	A	1 88 [3]	--
Pop acc. a from the Stack	POP	A	1 84 [4]	--
Push iX onto the stack	PUSH	iX	1 89 [3]	2 9089 [4]
Pop iX from the Stack	POP	iX	1 85 [4]	2 9085 [5]
Push Condition Codes onto the Stack	PUSH	CC	1 8A [3]	--
Pop Condition Codes from the Stack	POP	CC	1 86 [4]	--
Reset Carry Flag	RCF		1 98 [2]	--
Set Carry Flag	SCF		1 99 [2]	--
Reset Interrupt Mask	RIM		1 9A [2]	--
Set Interrupt Mask	SIM		1 9B [2]	--
Reset Stack Pointer	RSP		1 9C [2]	--
No Operation	NOP		1 9D [2]	--
Interrupt Routine Return	IRET		1 80 [9]	--
Subroutine Return	RET		1 81 [6]	--
Software Trap	TRAP		1 83 [10]	--
Halt	HALT		1 8E [2]	--
Wait For Interrupt	WFI		1 8F [2]	--

PART 5 . RESET AND INTERRUPTS

5.1 RESETS

The ST8108 has two reset modes : an active low external reset pin ($\overline{RESET}$) and a power-on reset function(POR). Refer to Figure 5.1 for timing sequence diagram.

5.1.1 RESET PIN

The Reset input pin is used to reset the MCU to provide an orderly software startup procedure. When using the external reset mode, the RESET pin must stay low for a minimum of one and one half t_{cyc}. The $\overline{RESET}$ pin contains an internal Schmitt Trigger as part of its input to improve noise immunity

5.1.2 POWER-ON $\overline{RESET}$

The power-On reset occurs when a positive transition is detected on V_{DD}. The power-on reset is strictly used for power up conditions and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuity provides for a 4096 t_{cyc} delay from the time that the oscillator becomes active. If the external $\overline{RESET}$ pin is low at the end of the 4096 t_{cyc} time out, the processor remains in the reset condition until $\overline{RESET}$ goes high. The user must ensure that V_{DD} has risen to a point where the MCU can operate properly prior to the time the 4096 POR reset cycles have elapsed. If there is doubt, the external $\overline{RESET}$ pin should remain low until V_{DD} has risen to the minimum operating voltage specified.

Table 5.2 shows the actions of the two resets on internal circuits, but not necessarily the order of occurence (X indicates that the condition occurs for the particular reset).

5.2 INTERRUPTS

Systems often require normal processing to be interrupted in order to handle external events. The ST8108 may be interrupted by one of five different methods : either one of four maskable hardware interrupts (INT, SPI, SCI, or TIMER) or a non-maskable software interrupt (TRAP).

Interrupts such as Timer, SPI, and SCI have several flags which will cause the interrupt.

Generally, interrupt flags are located in read-only status registers, whereas their equivalent enable bits are located in associated control registers. The interrupt flags and enable bits are never contained in the same register. If the enable bit is a logic zero it blocks the interrupt from occuring but does not inhibit the flag from being set. Reset clears all enable bits to disable interrupts during the reset procedure.

The general sequence for clearing an interrupt is a software sequence of first accessing the status register while the interrupt flag is set, followed by a read or write of an associated register. When any of these interrupts occur, and if the enable bit is a logic one, normal processing is suspended at the end of the current instruction execution.

Interrupts cause the processor registers to be saved on the stack and the interrupt mask (I bit) is set to prevent additional interrupts. The appropriate interrupt vector then points to the starting address of the interrupt service routine (refer to figure 5.3 for vector location). Upon completion of the interrupt service routine, the IRET instruction (which is normally a part of the service routine) causes the register contents to be recovered from the stack followed by a return to normal processing. The stack order is shown in Figure 3.2 (Part 3). The internal interrupt timing diagram is shown on Figure 5.7. The interrupt latency may be calculated as : 11 cycles plus the time to complete the current running instruction.

5.2.1 HARDWARE CONTROLLED INTERRUPT SEQUENCE

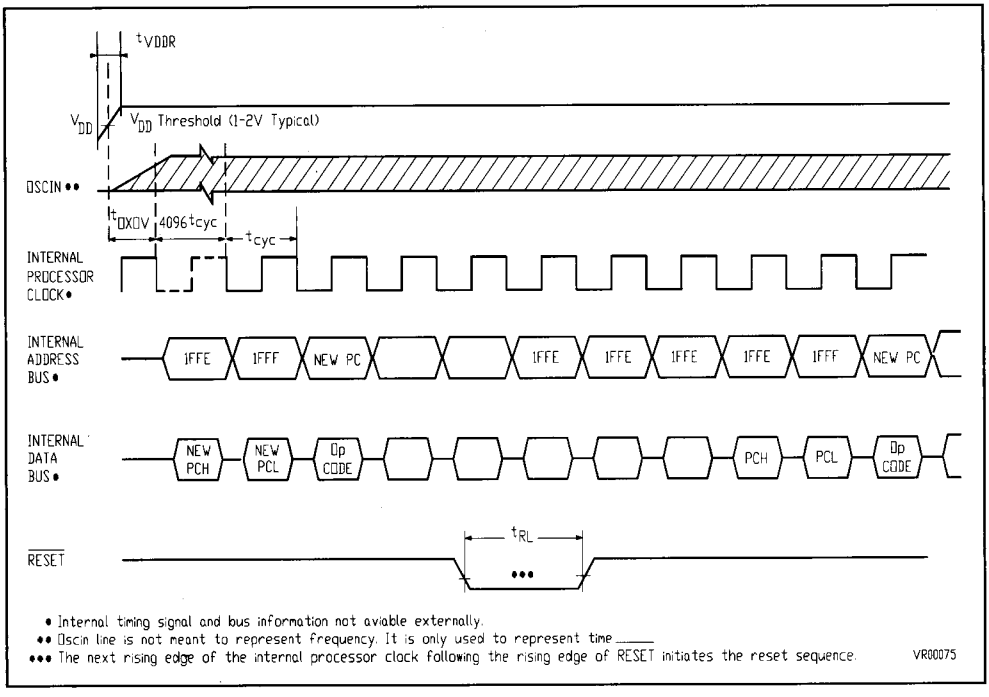
The three functions, $\overline{RESET}$, HALT and WFI, are not in the strictest sense interrupts ; however, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in Figure 5.4 and for HALT and WFI are provided in Figure 5.5

(a) A low input on the $\overline{RESET}$ input pin causes the program to vector to its starting address which is specified by the contents of memory locations $1FFE and $1FFF. The I bit in the condition code register is also set. Much of the MCU is configured to a known state during this type of reset as previouly described in paragraph 5.1.

(b) The HALT instruction causes the oscillator to be turned off and the processor to "sleep" until an external interrupt (INT) or reset occurs.

(c) The WFI instruction causes all processor clocks to stop, but leaves the Timer, SCI, and SPI clocks running. This "rest" state of the processor can be cleared by reset, external interrupt (INT) Timer Interrupt, SPI interrupt, or SCI interrupt.

Figure 5.1 . Power-on Reset And $\overline{\text{RESET}}$



• Internal timing signal and bus information not aviable externally.
•• Oscin line is not meant to represent frequency. It is only used to represent time.

Table 5.2 . Reset Action On Internal Circuit

Condition	RESET Pin	Power-On Reset
Timer Prescaler reset to Zero State.	X	X
Timer Counter Configured to $FFFC.	X	X
Timer Output Compare (OCMP) Bit Reset to Zero.	X	X
All Timer Interrupt Enable Bits Cleared (ICIE,OCIE and TOIE) to disable timer interrupts.	X	X
The OVL timer bit is cleared by reset.	X	X
All data direction registers cleared to zero (input).	X	X
Configure stack pointer to $00FF.	X	X
Force internal address bus to restart vector ($1FFE-$1FFF).	X	X
Set I bit in condition code register to a logic one.	X	X
Clear halt latch.	X*	X
Clear External Interrupt latch.	X	X
Clear WAIT latch.	X	X
Disable SCI (Serial control bits TE=0). Other SCI bits cleared by reset include : TIE,TCIE, RIE, ILIE, RWU, SBK, RDRF, IDLE, OR, NF, FE, SCP0 and SCP1.	X	X
Disable SPI (Serial output enable control bit SPE = 0). Other SPI bits cleared by reset include : SPIE, MSTR, SPIF, WCOL and MODF.	X	X
Set serial status bits TDRE and TC.	X	X
Clear serial interrupt enable bits (SPIE, TIE and TCIE).	X	X

5.2.2 . SOFTWARE INTERRUPT (TRAP)

The software TRAP is an executable instruction. The action of the instruction is similar to the hardware interrupt. TRAP is executed regardless of the state of the interruptmask (I bit) in the condition code register. The interrupt service routine address is specified by the contents of memory location $1FFC and $1FFD.

5.2.3 . EXTERNAL INTERRUPT

If the mask (I bit) of the condition code register has been cleared and the external interrupt pin ($\overline{INT}$) has gone low, then the external interrupt is recognized. When the interrupt is recognized, the current state of the CPU is pushed into the stack and the I bit is set, which masks further interrupts until the present one is serviced. The interrupt service routine address is specified by the contents of memory location $1FFA and $1FFB. A level-sensitive and negative edge-sensitive trigger, or a negative edge-sensitive only trigger are available as a mask option. Figures 5.6 and 5.7 show both a functional and mode timing diagram for the interrupt line. The first method shows single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the number of cycles required to execute the interrupt service routine plus 21 cycles. Once a pulse occurs, the next pulse should not occur until the MCU software has exited the routine (an IRET instruction occurs).

* Indicates that Timeout still occurs.

Table 5.3 . Vector Addresses for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interupts	Vector Address
N/A	N/A	Reset	$\overline{\text{RESET}}$	$1FFE-$1FFF
N/A	N/A	Software	TRAP	$1FFC-$1FFD
N/A	N/A	External Interrupt	$\overline{\text{INT}}$	$1FFA-$1FFB
Timer Status	ICF	Input Capture	TIMER	$1FF8-$1FF9
	OCF	Output Compare	"	"
	TOF	Timer Overflow	"	"
SCI Status	TDRE	Transmit Buffer Empty	SCI	$1FF6-$1FF7
	TC	Transmit Complete	"	"
	RDRF	Receiver Buffer Full	"	"
	IDLE	Idle Line Detect	"	"
	OR	Overrun	"	"
SPI Status	SPIF	Transfer Complete	SPI	$1FF4-$1FF5
	MODF	Mode fault	"	"

Figure 5.6 shows several interrupt lines "wire-ORed" to form the interrupts at the processor. Thus, if after servicing one interrupt the interrupt line remains low, then the next interrupt is recognized.The internal interrupt latch is cleared in the first part of the service routine; Therefore, one, and only one, external interrupt pulse could be latched during T_{ILIL} and serviced as soon as the I bit is cleared.

5.2.4 . TIMER INTERRUPT

There are three different timer interrupt flags that will cause a timer interrupt whenever they are set and enabled. These three interrupt flags are found in the three most significant bits of the timer status register (TSR, location $13) and all three will vector to the same interrupt service routine ($1FF8-$1FF9).

All interrupt flags have corresponding enable bits (ICIE, OCIE and TOIE) in the timer control register (TCR, location $12). Reset clears all enable bits, thus preventing an interrupt from occuring during the reset time period. The actual processor interrupt is generated only if the I bit in the condition code register is also cleared.When the interrupt is recognized the current machine state is pushed into the stack and I bit is set.

This masks further interrupts until the present one is serviced. The interrupt service routine address is specified by the contents of memory location $1FF8 and $1FF9. The general sequence for clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register. Refer to Part 9 "PROGRAMMABLE TIMER" for additional information about the timer circuitry.

5.2.5 SERIAL COMMUNICATIONS INTERFACES (SCI) INTERRUPTS

An interrupt in the serial communications interface (SCI) occurs when one of the interrupt flag bits in the serial communications status register is set, provided the I bit in the condition code register is clear and the enable bit in the serial communications control register 2 (location $0F) is enabled. When the interrupt is recognized, the current state of the machine is pushed into the stack and the I bit in the condition code register is set. The I bit set masks further interrupts until the present one is serviced.

The SCI interrupt causes the program counter to vector to memory location $1FF6 and $1FF7 which contains the starting address of the interrupts service routine. Software in the serial interrupt service routine must determine the priority and cause of the SCI interrupt by examining the interrupt flags and the status bits located in the serial communications status register (location $10).

The general sequence for clearing an interrupt is a status register while the flag is set followed by a read or write of an associated register. Refer to part 11 "SERIAL COMMUNICATIONS INTERFACE" for a description of the SCI system and its interrupts.

5.2.6 . SERIAL PERIPHERAL INTERFACE (SPI) INTERRUPTS

An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag bits in the serial peripheral status register (location $0B) is set, provided the I bit in the condition code register is cleared and the enable bit in the serial peripheral control register (location $OA) is enabled.

When the interrupt is recognized, the current state of the machine is pushed into the stack and I bit in the condition code register is set. These masks further interrupt until the present one is serviced. The SPI interrupt causes the program counter to vector to memory location $1FF4 and $1FF5 which contains the starting address of the interrupt service routine.

Software in the serial peripheral interrupt service routine must determine the priority and cause of the SPI interrupt by examining the interrupt flag bits located in the SPI status register. The general sequence for clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register.

Refer to "SERIAL PERIPHERAL INTERFACE" for a description of the SPI system and its interrupts.

5.3 . LOW POWER MODES

5.3.1 . HALT INSTRUCTION

The HALT instruction places the ST8108 in its lowest power consumption mode. In the HALT mode the internal oscillator is turned off, causing all internal processing to be halted; During the HALT mode, the I bit in the condition code register is cleared to enable external interrupts.

All other registers and memory remain unaltered and all input/output line remains unchanged. This continues until an external interrupt (INT) or reset is sent. Then, the internal I oscillator is turned on. The external interrupt or reset causes the program counter to vector to memory location $1FFA and $1FFB or $1FFE and $1FFF which respectively contain the starting address of the interrupt or the reset service routine .

5.3.2 . WFI INSTRUCTION

The WFI instruction places the ST8108 in a low power consumption mode, but the WFI mode consumes somewhat more power than the HALT mode. In the WFI mode, the internal clock remains active, and all CPU processing is stopped; however, the programmable timer, serial peripheral interface, and serial communications interface systems remain active. During the WFI mode, the I bit in the condition code register is cleared to enable all interrupts. All other registers and memory remain unaltered and all parallel input/output lines remain unchanged.

This continues until any interrupt or reset is sensed. At this time the program counter branches to the memory location ($1FF4 through $1FFF) which contains the starting address of the interrupt or reset service routine.

5.4 . DATA RETENTION MODE

The contents of RAM and CPU registers are retained at supply voltage as low as 2.0 Vdc. This is referred as the data retention mode, where the RAM data is held, but the device is not guaranteed

Figure 5.4. Hardware Interrupt Flowchart

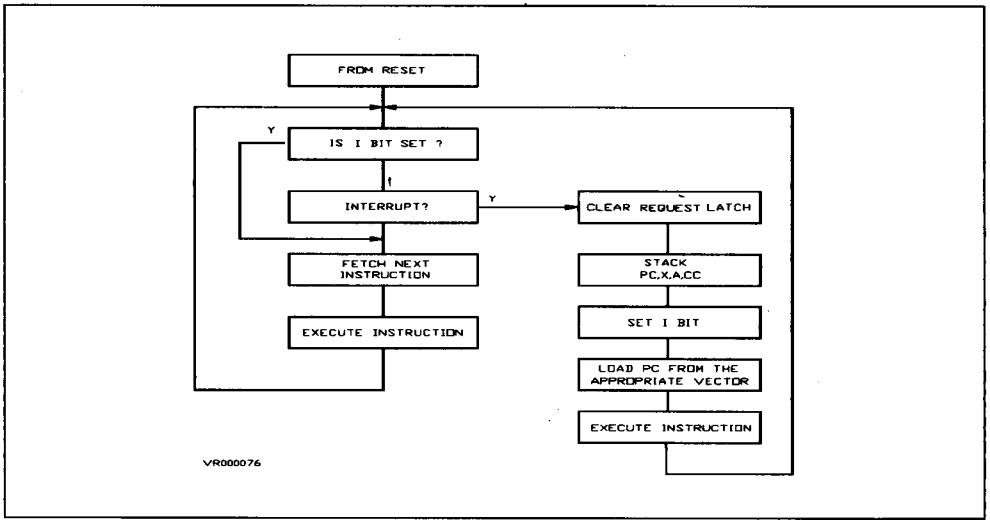


Figure 5.5 . HALT/WFI Flowchart

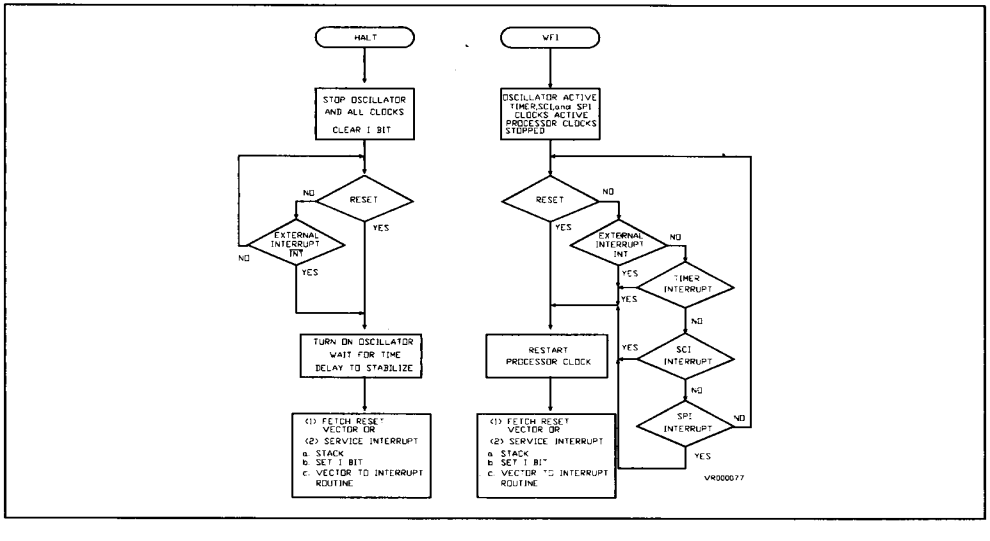


Figure 5.6. External Interrupt Function Diagram

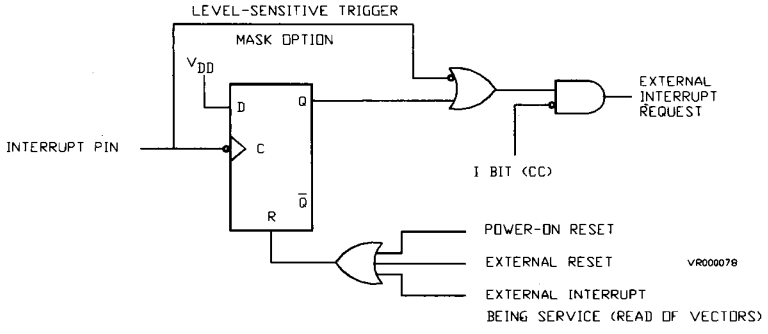
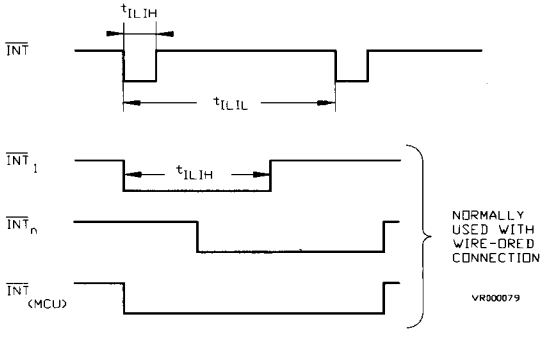


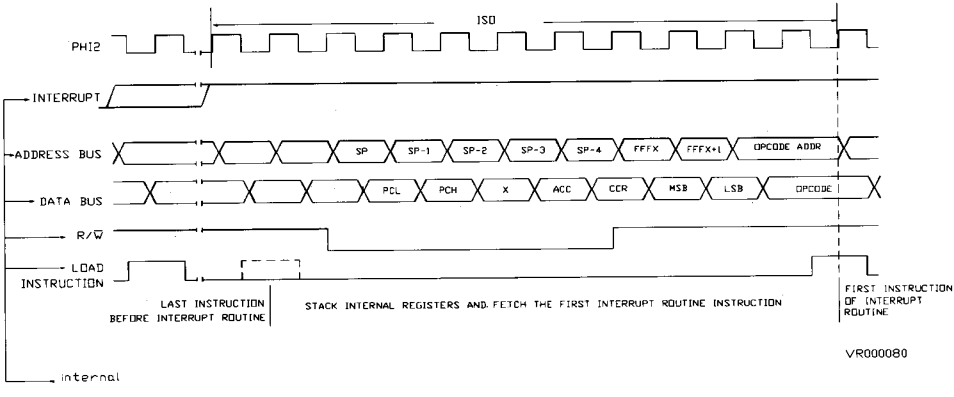
Figure 5.7 .External Interrupt Mode Diagram



Edge-Sensitive Trigger Condition: the minimun pulse width (t_{ILIH}) is either 125ns (V_{DD}=5V) or 250ns (V_{DD}=3V). The period t_{ILIL} should not be less than the number of t_{CYC} cycles it takes to execute the interupt service routine plus 21 t_{CYC} cycles.

Level-Sensitive Trigger Condition: if after servicing an interupt the $\overline{INT}$ main low, then the next interrupt is recognised.

Figure 5.8. Interrupt Timing Diagram



ISD (Interrupt Service Delay) depends of the current instruction :
ISD = 11 Cycles minimum
ISD = 22 Cycles maximum (if the current instruction interrupted is a multiplication)

PART 6 . CLOCK SYSTEM, WFI AND HALT MODES

6.1 . ON CHIP CLOCK SYSTEM - CHARACTERISTICS.

The ST8108 can be configured by mask option to accept either a Crystal/Ceramic resonator input or an RC network to control the internal oscillator. The internal clock (F_{OP}) is derived by a divide-by-two of the internal oscillator frequency (f_{OSC}).

6.1.1 . CRYSTAL

The circuit shown in Figure 6.1 (b) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{OSC}. Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. Refer to paragraph 12 for V_{DD} and maximum frequency specifications.

CCrystal

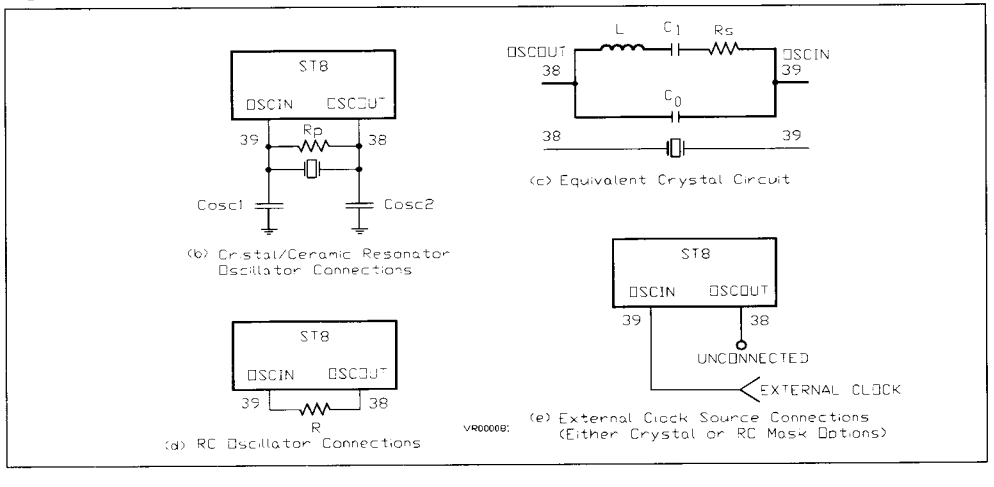
	2 MHZ	4 MHZ	8MHz	Units
R_{SMAX}	400	75	60	Ω
C_0	5	7	10	pF
C_1	8	12	15	nF
C_{OSCIN}	15-40	15-30	15-25	pF
C_{OSCOUT}	15-30	15-25	15-20	pF
R_P	10	10	10	MΩ
Q	30	40	60	K

Ceramic Resonator

	2-8 MHZ	Units
R_S (Typicall)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSCIN}	30	pF
C_{OSCOUT}	30	pF
R_P	1-10	MΩ
Q	1250	

(a) Crystal / Ceramic Resonator Parameters

Figure 6.1 . Oscillator Connections

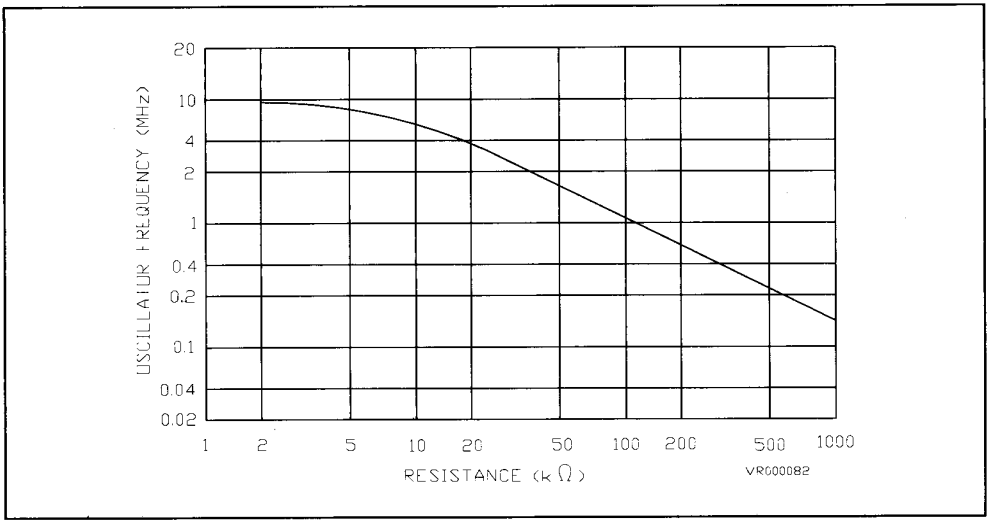


6.1.2 . CERAMIC RESONATOR
A ceramic may be used in place of the crystal in cost-sensitive applications. The circuit in Figure 6.1 (b) is recommended when using a ceramic resonator. Figure 6.1 (a) lists the recommended capacitance and feedback resistance values. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

6.1.3 . RC
If the RC oscillator is selected, then a resistor is connected to the oscillator pins as shown in Figure 6.1 (d). The relation between R and f_{OSC} is shown in Figure 6.6.

6.1.4 . EXTERNAL CLOCK
An external clock should be applied to the OSCIN input with the OSCOUT pin not connected, as shown in Figure 6.1(e). An external clock may be used with either the RC or crystal oscillator option. The t_{OXOV} ot t_{ILCH} specifications do not apply when using an external clock input. The equivalent specification of the external clock source should be used instead of t_{OXOV} or t_{ILCH}.

Figure 6.6 . Typical Frequency vs Resistance For RC Oscillator Option (V_{DD} = 5.0V)



6.2 . LOW POWER MODES

6.2.1 . HALT INSTRUCTION

The HALT instruction places the CPU in its lowest power consumption mode. In the HALT mode the internal oscillator is turned off, causing all internal processing to be halted. Refer to Figure 6.3. During the HALT mode, the I bit of the condition code register is cleared to enable external interrupts . All other registers remain unaltered. This continues until an external interrupt or RESET is sensed while the oscillator is turned on, but the CPU remains inactive . When the oscillator has correctly restarted, the first code operation is fetched at starting address of the interrupt or reset service routine.

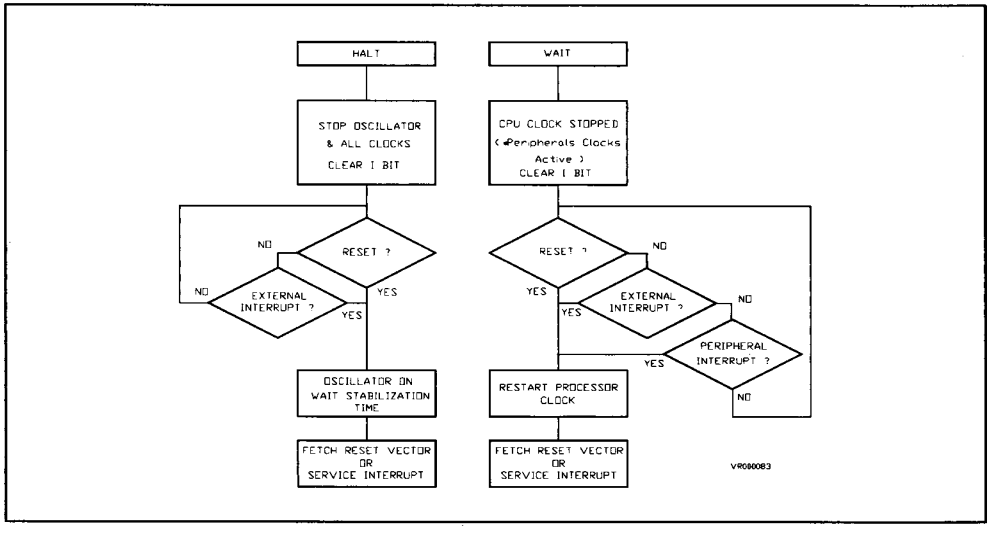
6.2.2 . WFI INSTRUCTION

The WFI instruction places the CPU in a low power consumption mode, but the WFI mode consumes somewhat more power than the HALT mode. In the WFI mode, the internal clock remains active, and all CPU processing is stopped. However, the peripherals remain active. Refer to Figure 6.3. During the WFI mode, the I bit of the condition code register is cleared to enable any interrupt. All other registers remain unaltered. This continues until any interrupt or reset is sensed. At the time the program couter vectors to the memory location which contains the starting address of the interrupt or reset service routine.

6.2.3 . DATA RETENTION MODE

The contents of the CPU registers are retained at supply voltages as low as 2.0V_{DC} . This is referred to as the data retention mode, where the data RAM is held, but the device is not guaranteed to operate.

Figure 6. 3 . HALT/WAIT Flowchart



PART 7 . ST8108 MICROPROCESSOR MODE

7.1 . NOTE ON MICROPROCESSOR MODE:
ST8108 has been designed to support also a microprocessor mode. In this case, a part of the I/O ports are modified into address/data/control signals in order to allow the ST8108 to address an external memory.

Nevertheless, at the time of printing, this mode is not released for production. Contact your local SGS-THOMSON sales offices for detailed information.

PART 8 . INPUT OUTPUT PORTS PROGRAMMING

8.1 . INPUT/OUTPUT PORTS PROGRAMMING

8.1.1 PARALLEL PORTS

Ports A, B and C may be programmed as input or output under software control. The direction of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Each 8-bit port having an associated DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero. At power-on or reset, all DDRs are cleared, which configures all port A, B and C pins as inputs. The data direction registers are capable of being written to or read by the processor. Refer to Figure 8.1 and table 8.2 During the programming output state, a read of the data register actually reads the value of the output data latch and not the I/O pin.

8.1.2 PA0 - PA7

These eight I/O lines compose the port A. The state of any pin is software programmable and all port A lines are configured as input during power-on or reset. Refer to INPUT/OUTPUT PROGRAMMING paragraph for a detailed description of I/O programming.

8.1.3 PB0 - PB7

These eight lines compose the port B. The state of any pin is software programmable and all port B lines are configured as input during power-on or reset. Refer to INPUT/OUTPUT PROGRAMMING paragraph for a detailed description of I/O programming.

8.1.4 PC0 - PC7

These eight lines compose the port C. The state of any pin is software programmable and all port C lines are configured as input during power-on or reset.Refer to INPUT/OUTPUT PROGRAMMING paragraph for a detailed description of I/O programming.

8.1.5 PORT D

Port D is a 7-bit input port (PD0-PD5, PD7) that continually monitors the external pins whenever the SPI or SCI systems are disabled. During power-on, reset or external reset all seven bits become valid input ports because all special function output drivers are disabled. For example, with the serial communications interface (SCI) system enabled, (RE = TE = 1) PD0 and PD1 inputs will read zero.

With the serial peripheral interface (SPI) system disabled (SPE = 0) PD2 through PD5 will read the state of the pin at the time of the read operation. No data register is associated with the port when it is used as an input.

It is recommended that all unused inputs and I/O port be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}).

8.1.6 SERIAL PORT (SCI OR SPI)

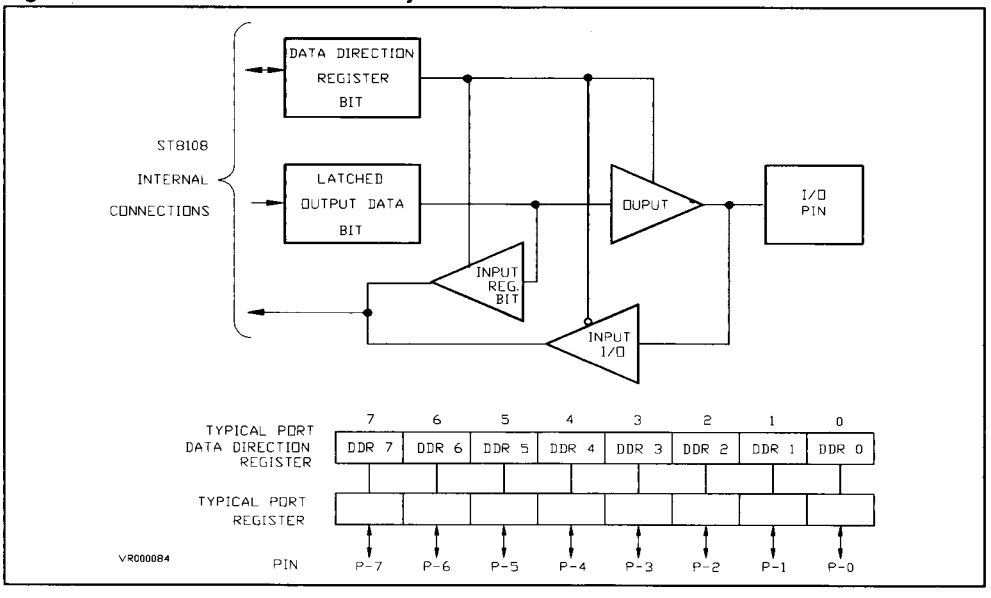
The serial communications interface (SCI) and serial peripheral interface (SPI) use port D pins for their functions. The SCI function requires two pins (PD0-PD1) for serial data input (SIN) and serial data output (SOUT), whereas the SPI function requires four pins (PD2-PD5) for its serial data input/output (MISO), serial data output/input (MOSI), system clock (SCK), and slave select ($\overline{SS}$) respectively. Refer to SERIAL COMMUNICATIONS INTERFACE and SERIAL PERIPHERAL INTERFACE for more detailed information.

I/O PIN FUNCTIONS

R/$\overline{W}$*	DDR	I/O pin functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

* R/W is an internal signal.

Figure 8.1 . Parallel Port I/O Circuitry



PART 9 . 16 BIT TIMER

9.1 . INTRODUCTION

9.1.1 . GENERAL

The 16-bit programmable timer can be used for many purposes including pulse length measurement of one input signal and generating one output signal waveform. Pulse lengths for both input and output signals can vary from several microseconds to many seconds. The timer is preceded by a manufacturing mask option programmable prescale

- Mask option 1 : The prescaler divides by 8 the internal processor clock
- Mask option 2 : The prescaler divides by 4 the internal processor clock
- Mask option 3 : The prescaler divides by 2 the internal processor clock.

Depending of the mask option, the timer is also capable of generating periodic interrupts or indicating passage of an arbitrary multiple of 2,4 or 8 numbers of MCU cycles. Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers.

These registers contain the "high byte" and "low byte" of that function. However an access of the "high byte" inhibits that specific Timer capability until the "low byte" is also accessed. The programmable Timer capabilities are provided by using the following 10 addressable 8-bit registers :

- Timer Control Register (TCR);
- Timer Status Register (TSR);
- Input Capture High Register (ICHR);
- Input Capture Low Register (ICLR);
- Output Compare High Register (OCHR);
- Output Compare Low Register (OCLR);
- Counter High Register (CHR);
- Counter Low Register (CLR);
- Alternate Counter High Register (ACHR);
- Alternate Counter Low Register (ACLR);

9.2 . CONNECTIONS

9.2.1 . CONNECTIONS TO EXTERNAL DEVICES

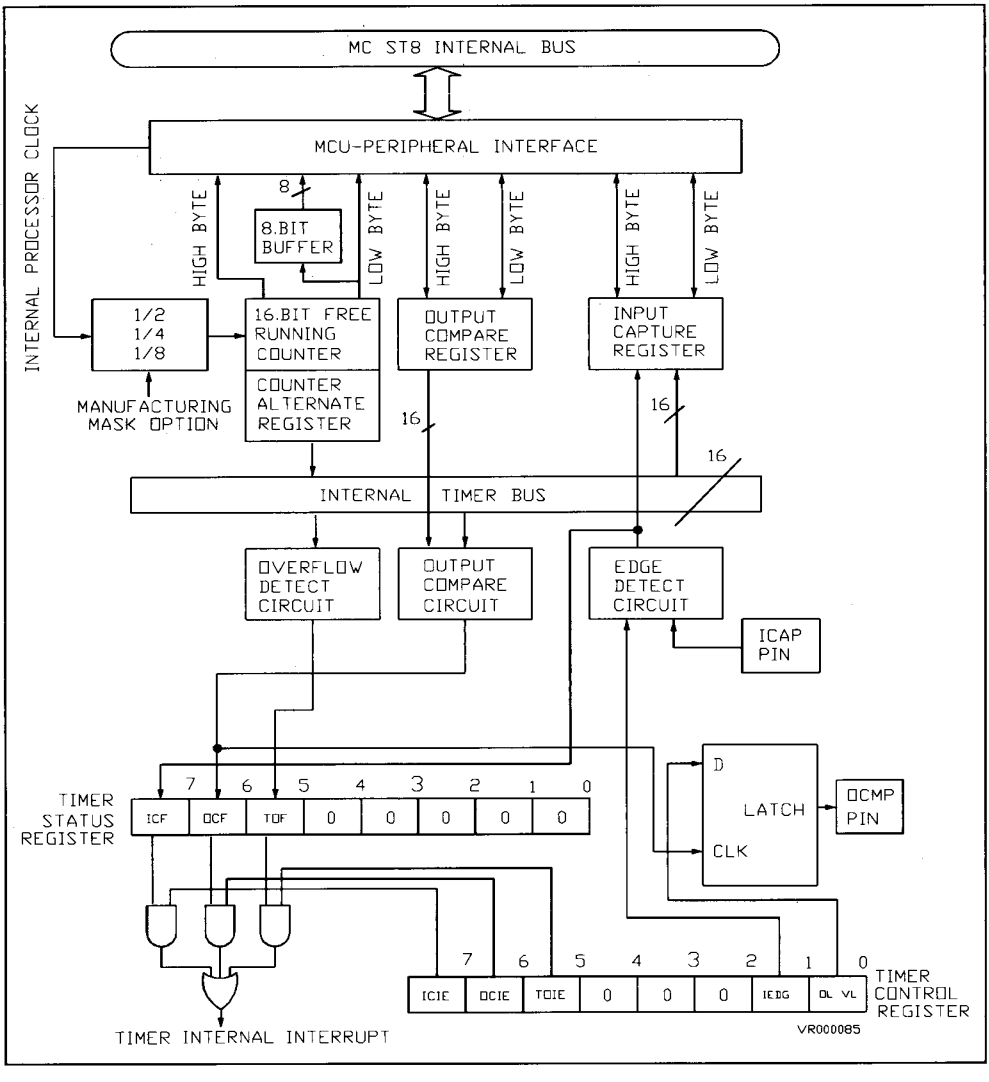
There are 2 connections to external devices

- Input Capture Pins (ICAP)
- Output Compare Pins (OCMP).

9.2.2 . CONNECTIONS TO INTERNAL MCU BUS

- PHI2 : Internal Processor Clock
- $\overline{CS}$: Chip Select
- R/W : Read / Write
- POR : Power On Reset
- $\overline{INT}$: Timer Interrupt.
- DATA BUS
- ADDRESS BUS

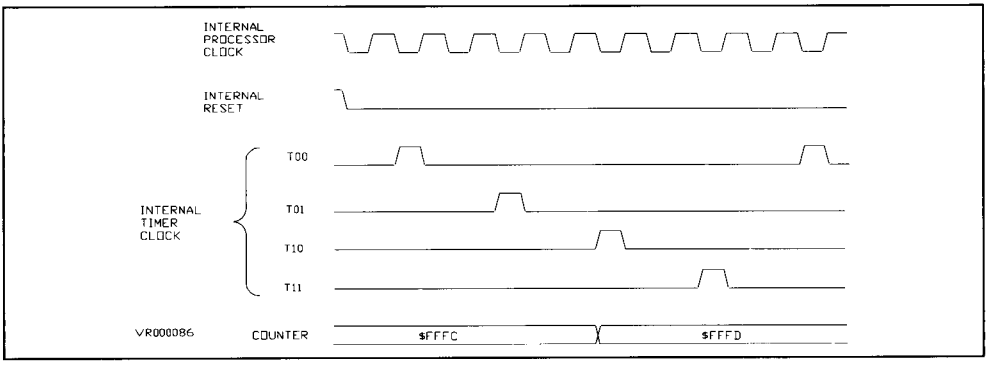
Figure 9.1 . Programmable Timer Block Diagram



9.3 . HARDWARE FUNCTIONAL DESCRIPTION

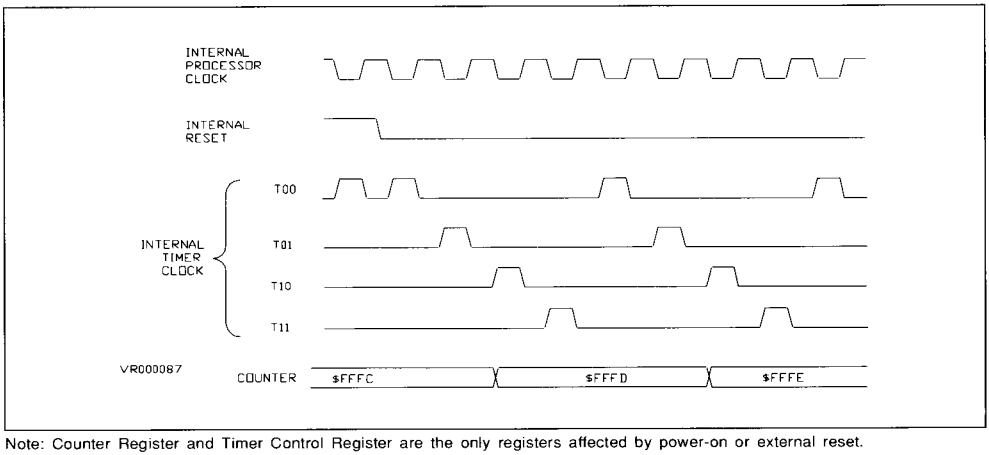
9.3.1 . TIMER STATE TIMING DIAGRAM FOR RESET

Mask option 1 : Internal Processor clock divided by 8



Note: Counter Register and Timer Control Register are the only registers affected by power-on or external reset.

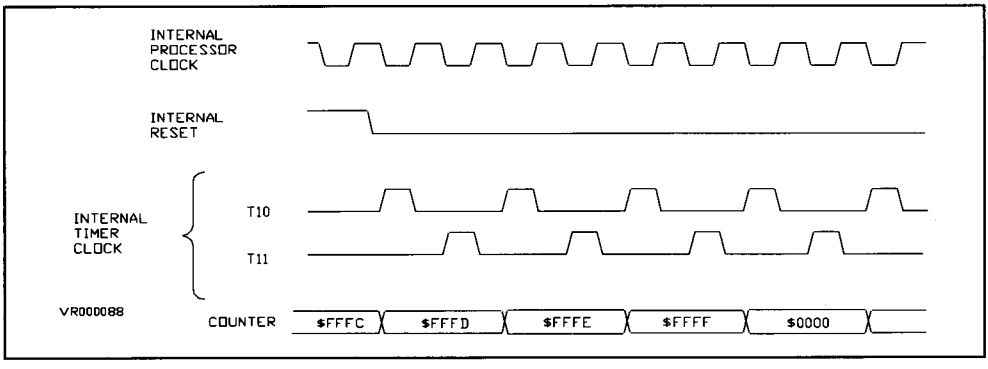
Mask Option 2 : Internal Processor Clock Divided by 4



Note: Counter Register and Timer Control Register are the only registers affected by power-on or external reset.

9.3.1 . (Continued)

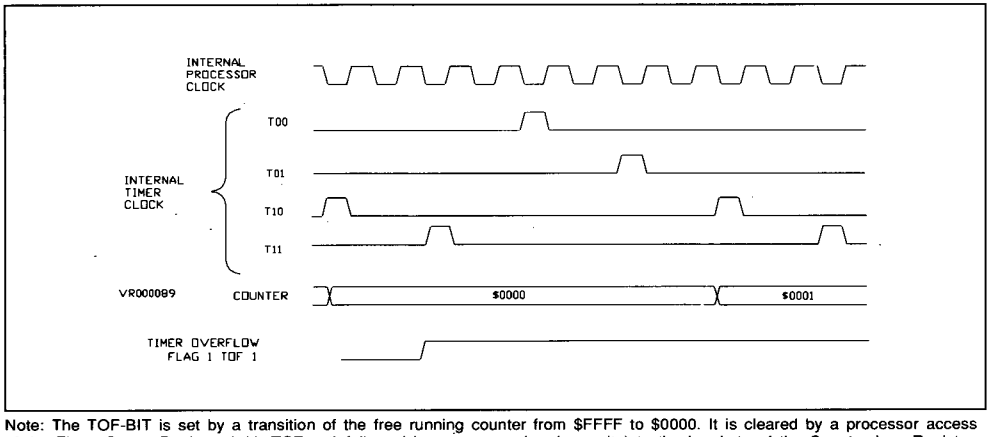
Mask Option 3 : Internel Processor Clock Divided by 2



Note: Counter Register and Timer Control Register are the only registers affected by power-on or external reset.

9.3.2 . TIMER STATE TIMING DIAGRAM FOR TIMER OVERFLOW

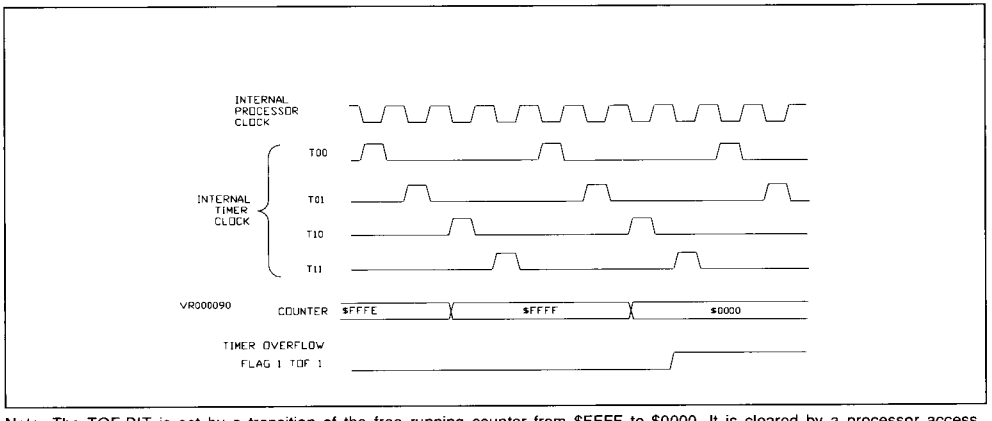
Mask option 1 : Internal Processor clock divided by 8



Note: The TOF-BIT is set by a transition of the free running counter from $FFFF to $0000. It is cleared by a processor access of the Timer Status Register (with TOF set) followed by an access (read or write) to the low byte of the Counter Low Register (CLR).

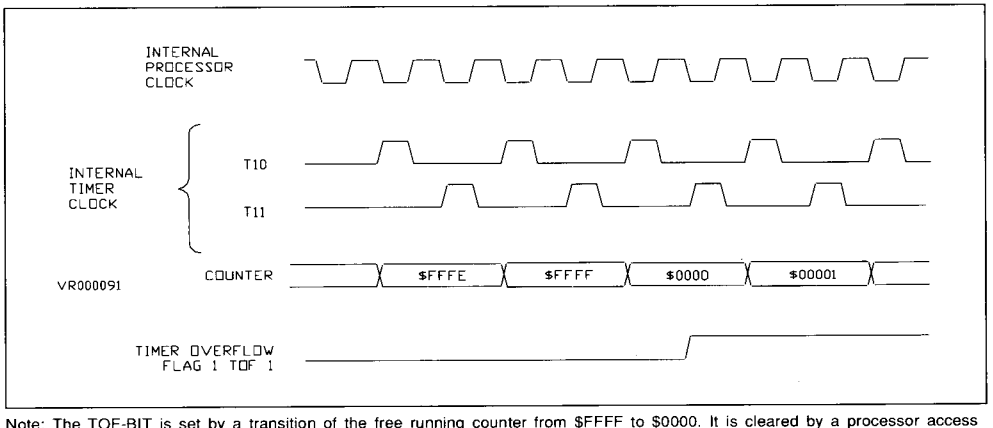
9.3.2 . (Continued)

Mask Option 2 : Internal Processor Clock Divided by 4



Note: The TOF-BIT is set by a transition of the free running counter from $FFFF to $0000. It is cleared by a processor access of the Timer Status Register (with TOF set) followed by an access (read or write) to the low byte of the Counter Low Register (CLR).

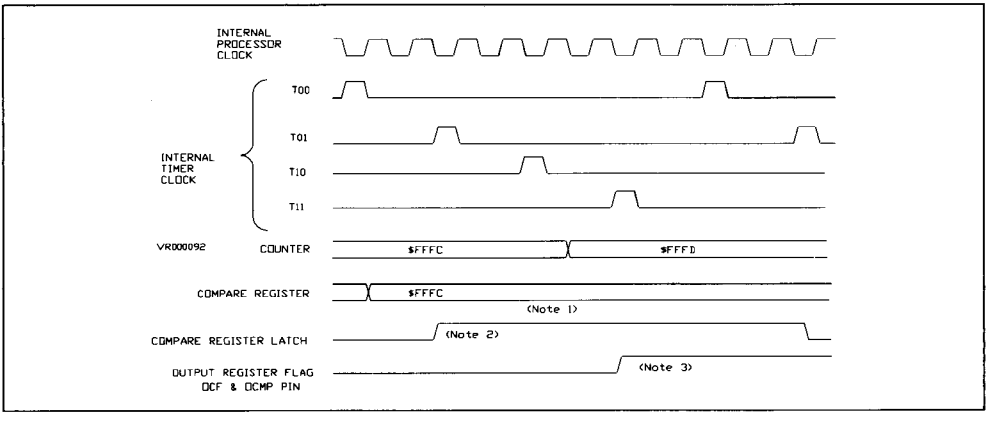
Mask Option 3 : Internel Processor Clock Divided by 2



Note: The TOF-BIT is set by a transition of the free running counter from $FFFF to $0000. It is cleared by a processor access of the Timer Status Register (with TOF set) followed by an access (read or write) to the low byte of the Counter Low Register (CLR).

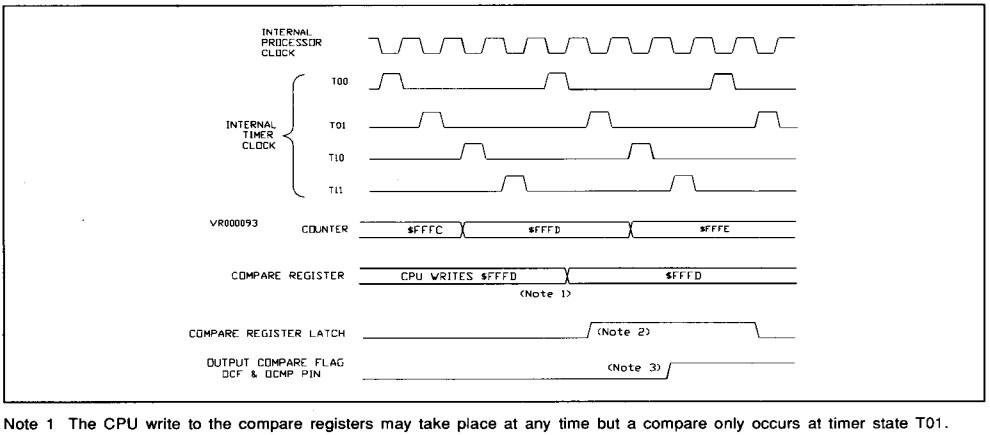
9.3.3 . TIMER STATE TIMING DIAGRAM FOR OUTPUT COMPARE

Mask option 1 : Internal Processor clock divided by 8



Note 1 The CPU write to the compare registers may take place at any time but a compare only occurs at timer state T01. Thus a 8-cycles difference may exist between the write to the compare register and the actual compare.
Note 2 :Internal compare takes place during timer state T01.
Note 3 :OCF is set at timer state T11 which follows the comparison match ($FFFC in this example).

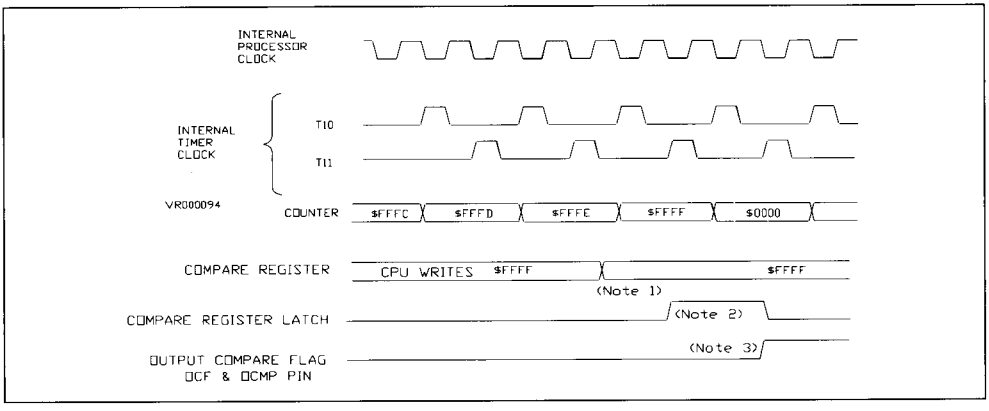
Mask Option 2 : Internal Processor Clock Divided by 4



Note 1 The CPU write to the compare registers may take place at any time but a compare only occurs at timer state T01. Thus a 4-cycles difference may exist between the write to the compare register and the actual compare.
Note 2 :Internal compare takes place during timer state T01.
Note 3 :OCF is set at timer state T11 which follows the comparison match ($FFFD in this example).

9.3.3 . (Continued)

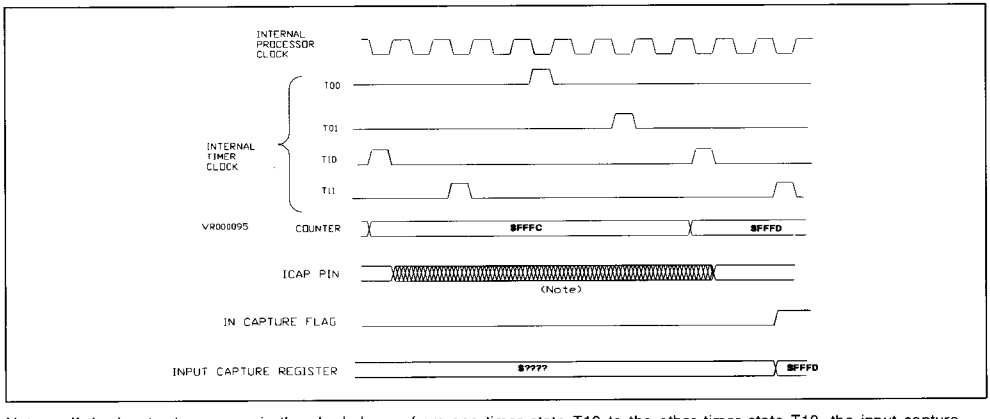
Mask Option 3 : Internal Processor Clock Divided by 2



Note 1 The CPU write to the compare registers may take place at any time but a compare only occurs at timer state T11. Thus a 2-cycles difference may exist between the write to the compare register and the actual compare.
Note 2 :Internal compare takes place during timer state T11.
Note 3 :OCF is set at timer state T11 which follows the comparison match ($FFFF in this example).

9.3.4 . TIMER STATE TIMING DIAGRAM FOR INPUT CAPTURE

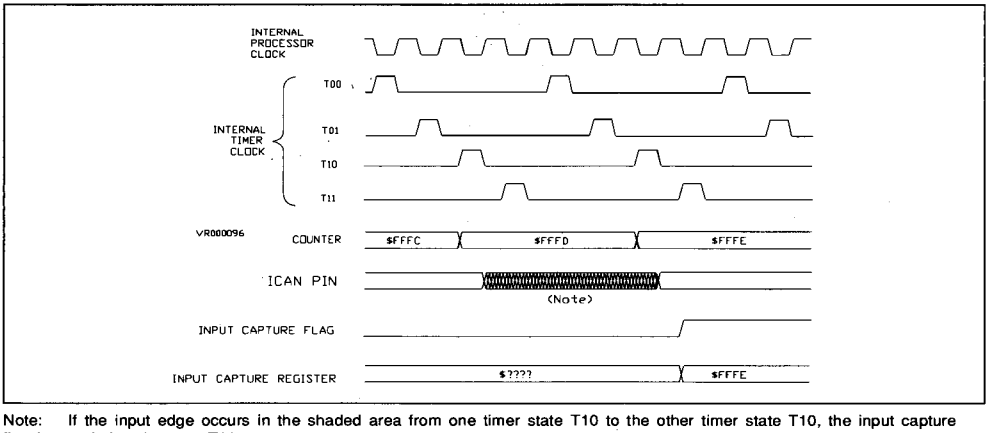
Mask option 1 : Internal Processor clock divided by 8



Note: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10, the input capture flag is set during the next T11 state.

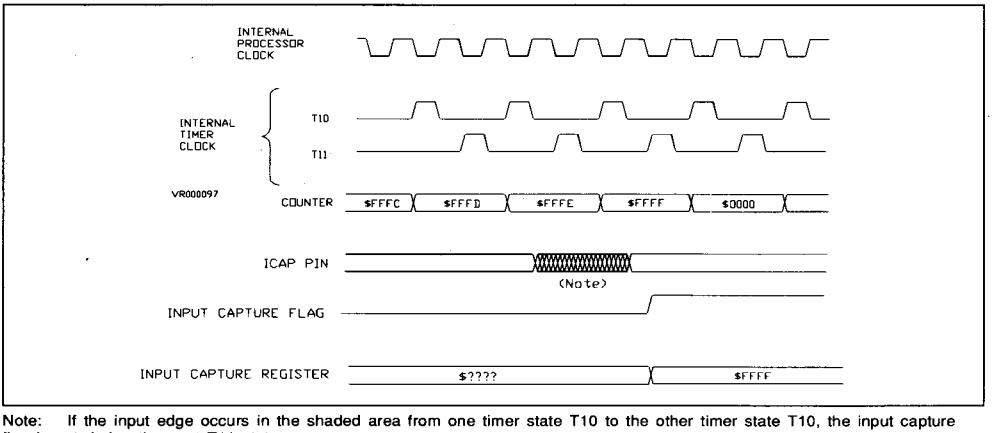
9.3.4 . (Continued)

Mask Option 2 : Internal Processor Clock Divided by 4



Note: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10, the input capture flag is set during the next T11 state.

Mask Option 3 : Internel Processor Clock Divided by 2



Note: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10, the input capture flag is set during the next T11 state.

9.4 SOFTWARE FUNCTIONAL DESCRIPTION

9.4.1 . COUNTER

The key element of the programmble Timer is a 16-bit free running counter or counter register, preceded by a fixed prescaler which divides the internal processor clock by two, four or height according to the manufacturing mask option. The prescaler gives a Timer resolution of 0.5 microsecond for option 3, 1 microsecond for option 2 and 2 microsecond for option 1, with a 4 MHz processor clock. The counter is clocked to increasing values during the low portion of the internal processor clock. Software can read the counter at any time without affecting its value.

The double byte free running counter can be read from two locations called "Counter Register" or "Alternate Counter Register". A read sequence containing only a read of the least significant byte of the free running counter will receive the count value at the time of the read. If a read of the free running counter or counter alternate register first addresses the most significant byte it causes the least significant byte to be transferred into a buffer. This buffer value remains unchanged after the most significant byte is "read" even if the user reads the most significant byte several times. This buffer is accessed when reading the free running counter or counter alternate register least significant byte (CLR or ACLR), and thus completes the read sequence of the total counter value. Note that in reading either the free running counter or counter alternate register, if the most significant byte is read, the least significant byte must also be read in order to complete the sequence.

The free running counter is configured to $FFFC during external reset and is always a read-only register. During a power-on reset (POR), the counter is also configured to $FFFC and begins running the oscillator startup delay. Because the 16 bit free running counter is preceded by a mask option programmable divide by 2, 4 or 8 prescaler, the value in the free running counter repeats respectively every 131072, 262144, or 524288 internal processor clock.

When the counter rolls over from $FFFF to $0000, the Timer Overflow flag (TOF) bit is set (bit-5 of TSR). Timer interrupt is then enabled by setting the TOIE bit (bit 5 of TCR).

9.4.2 . INPUT CAPTURE

Input Capture High Register :

CTH7	CTH6	CTH5	CTH4	CTH3	CTH2	CTH1	CTH0

Input Capture Low register:

CTL7	CTL6	CTL5	CTL4	CTL3	CTL2	CTL1	CTL0

The Input Capture Register (ICR) is a 16-bit register, which is made up of two 8-bit register : the most significant byte register (ICHR) and the least significant byte register (ICLR).

These registers are read only and are used to latch the value of the free running counterafter a defined transition is sensed by the input capture edge detector at pin ICAP. When an input capture occurs, the corresponding flag ICF (bit-7) in Timer Status Register (TSR) is set. The level transition of the input capture pin ICAP which trigggers the counter transfer for ICR is defined by the corresponding input edge bit IEDG (bit-1) of the Timer Control Register (TCR).

- IEDG1 = 0 = Negative Edge Sensitive
- IEDG1 = 1 = Positive Edge Sensitive.

Hardware circuitry has to provide protection from generating a wrong input capture when changing the edge sensitivity option of ICAP pin (IEDG-bit). The Input Capture Register is undetermined at power-on and is not affected by an external reset. An interrupt can also accompany an input capture provided the corresponding interrupt enable bit, ICIE (bit-7 of Timer Control Register) is set.

The result obtained by an input capture will be one more than the value of the free running counter on the rising edge of the internal processor clock preceding the external transition (refer to 9. 3.4). This delay is required for internal synchronisation.

The free running counter is transferred to the input capture Register on each proper signal transition regardless of whether the Input Capture Flag ICF (bit-7 of Timer Status Register) is set or clear. The Input Capture Register always contains the free running counter value which corresponds to the most recent input capture.

After a read of the most significant byte of the Input Capture Register (ICHR), counter transfer of input capture is inhibited until the least significant byte of input capture Register (ICLR) is also read. This characteristic forces the minimum pulse period attainabe to be determined by the time used in the capture software routine and its interaction with the main program.

A read of the least significant byte of the input Capture Register (ICLR) does not inhibit the free running transfer. Again, minimum pulse periods are the ones which allow software to read the least significant byte and perform needed operations. There is no conflict between the read of the Input Capture Register and the running counter transfer since they occur on opposite edges of the internal processor clock.

9.4.3 . OUTPUT COMPARE REGISTERS

Output Compare Registers can be used for several purposes such as controlling an output waveform or indicating when a period of time has elapsed. The Output Compare Registers are unique because all bits are readable and writable and are not affected by the Timer hardware.

Power-on or external reset does not affect the contents of these registers, and if the compare functions are not utilised, the two bytes of the Output Compare Registers can be used as storage locations.

Output Compare High Register :

CMH7	CMH6	CMH5	CMH4	CMH3	CMH2	CMH1	CMH0

Output Compare Low Register 1 :

CML7	CML6	CML5	CML4	CML3	CML2	CML1	CML0

The Output Compare Register (OCR) is a 16-bit register, which is made up of two 8-bit registers: The most significant byte register (OCHR) and the least significant byte register (OCLR. The contents of the Output Compare Register are compared with the contents of the free running counter once during every 2,4 or 8 internal processor clock periods according to the timer clock source mask option 1 (Fop /2), 2 (Fop /4) or 3 (Fop /8). If match is found, the Output Compare Flag OCF (bit-6 of Timer Status Register) is set and the corresponding output level OLVL1-bit (bit 0 of the Timer Control Register) is copied to an output level latch connected to the OCMP pin. The value in the output compares register and the output level bit register. It could be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit OCIE (bit-6 of the Timer Control Register), is set. After a processor write cycle to the Output Compare Register containing the most significant byte, the output compare function is inhibited until the least significant byte is also written. The user must write both bytes (locations) if the most significant byte is writen first. A write made to only the least significant byte will not inhibit the compare function. The minimum time required to update the Output Compare Register is a function of the software program rather than the internal hardware.The output level bit (OLVL) is copied to the corresponding output level latch and hence, to the OCMPpin regardless of whether the Output Compare Flag (OCF) is set or not.

9.4.4 . TIMER CONTROL REGISTER

The Timer Control Register (TRC) is an 8 bit read/write register. Three of these bits in this control register are the interrupts associated with the three flag bits found in the Timer status register (see section 9.4.5). One bit controls which edge is significant edge detector for the input capture (negative or positive). One bit controls the next values to be copied to the output level latches in response to successful output compares.

Timer Control Register :

ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL

Bit-7 : **ICIE**

If ,the Input Capture Interrupt Enable (ICIE) is set, a Timer interrupt is enabled whenever the ICF status flag (in the Timer Status Register) is set. If the ICIE-bit is clear, the interrupt is inhibited. The ICIE-bit is cleared by power-on or external reset.

Bit-6 : **OCIE**

If the Output Compare Interrupt Enable (OCIE) is set, a Timer interrupt is enabled whenever the OCF status flag (in the Timer Status Register) is set. If the OCIE-bit is clear, the interrupt is inhibited. The OCIE-bit is cleared by power-on or external reset.

Bit-5 : **TOIE**
If the Timer Overflow Interrupt Enable (TOIE) is set, a Timer interrupt is enable whenever the TOF status flag (in the Timer Status Register) is set. If the TOIE-bit is clear, the interrupt is inhibited.
The TOIE-bit is cleared by power-on or external reset.

Bit-1 : **IEDG**
The value of the IEDG-bit (Input Edge) determines which level transition on pin ICAP will trigger a free running counter transfer to the Input Capture Register .
The IEDG-bit is undetermined at power-on or external reset.

- IEDG = 0 = Negative Edge
- IEDG = 1 = Positive Edge

Bit-0 : **OLVL**
The value of OLVL-bit (output Level) is copied into the output level latch by the next successful output compare and will appear at pin OCMP.
The OLVL-bit and the output level latch are cleared by power-on or external reset.

- OLVL = 0 = Low Output
- OLVL = 1 = High Output.

9.4.5 . TIMER STATUS REGISTER
The Timer Status Register (TSR) is an 8-bit register of which the three most significant bits contain read-only status information. These five bits indicates the following.

a) A proper transition has been taken at pin ICAP with an accompanying transfer of the free running counter content to the corresponding Input Capture Register.

b) A match has been found between the free running counter and the Output compare Register OCR.

c) A free running counter transition from $FFFF to $0000 has been sensed (Timer Overflow).

The timer Status Register is illustrated below and followed by a definition of each bit.

Timer Status Register :

ICF	OCF	TOF	0	0	0	0	0

Bit-7 : **ICF**
The Input Capture Flag (ICF) is set when a proper edge has been sensed by the input capture edge detector at pin ICAP. The edge is selected by the IEDG-bit in the Timer Control Register. It is cleared by a processor access of the Timer Status Register (with ICF set) followed by accessing (read or write) the low byte of the Input Capture Register (ICLR). The Input Capture Flag is undetermined at power-on, and is not affected by an external reset.

Bit-6 : **OCF**
The Output Compare Flag (OCF) is set when the content of the free running counter matches the content of the Output Compare Register. It is cleared by a processor access of the Timer Status Register (with OCF set) followed by accessing (read or write) the low byte of the Output Compare Register (OCLR).
The Output Compare Flag is undetermined at power-on, and is not affected by an external reset.

Bit-5 : **TOF**
The Timer Overflow Flag (TOF) is set by a transition of the free running counter from $FFFF to $0000. It is cleared by a processor access of the Timer Status Register (with TOF set) followed by accessing (read or write) the low byte of the counter low register (CLR).

The Timer Overflow Flag is undetermined at power-on, and is not affected by an external reset.
Note : An access to the Alternate Counter Register (ACLR or ACHR) does not affect the TOF-bit.

Accessing the Timer Status Register satisfies the first condition required to clear any status bits which happen to be set during the access. The remaining step is to access the register which is associated with the status bit. Typically, this presents no problem for the input capture and output compare function.

A problem may occur when using the timer overflow function and reading the free running counter at random times to measure on elapsed time. Without incorporating the proper precautions into software, the Timer Overflow Flag (TOG-bit) could unintentionally be cleared if :

a) The Timer Status Register is accessed when TOF is set.

b) The least significant byte of the free running counter (CLR) is read but not for the purpose of servicing the flag.

9.4.6 . TIMER INTERRUPTS

There are three different Timer Interrupt Flags (ICF, OCF, TOF) that will cause a Timer interrupt whenever they are set and enable. These three interrupt flags are found in the most significant bits of the Timer Status Register (TSR).

There are three corresponding enable bits : ICIE for ICF, OCIE for OCF and TOIE for TOF. These enable bits are located in the Timer Control Regster (TCR). Power-on and external reset clears all enable bits, thus preventing an interrupt from occuring during the reset time period. The general sequence for clearing an interrupt is a software sequence of accessing by a read or write of the low byte associated register.

9.4.7 . POWER-ON AND EXTERNAL RESET INFLUENCE

The timer Control Regsister and the free running counter are the only sections of the timer affected by a power-on or an external reset.

9.4.7.1 . Output Compare Functions

The OCMP output latch is forced low during reset and stays low until valid compares change them to a high level. Because the Output Compare Flags(OCF) and Output Compare Registers are undeterminate at power-on, and are not affected by an external reset, care must be exercise when initializing the output compare functions with software. The following procedure is recommended.

a) Write the "high" byte at the output compare register to inhibit further compares until the low byte is written.

b) Read the Timer Status Register to arm the OCF, bit if it is already set.

c) Write the Output Compare Register "low" byte to enable the output compare function with the flag clear.

The purpose of this procedure is to prevent the OCF, bit from being set between the time it is read and the write to the corresponding Output Compare Register.

9.4.8 WFI AND HALT

During the WFI mode the TIMER continues to operate normally and may generate an interrupt to trigger the CPU out of the WFI state.

During the HALT mode the TIMER holds its current state retaining all data, and resumes operations from this point when an external interrupt is received. If an external reset is received the TIMER value will be set at $FFFC. A power-on detect has the same effect.

Another feature of the programmable timer is that in the HALT mode, if a least one valid input capture edge occurs at the ICAP pin, the corresponding input capture detect circuitry is armed. This action does not set any timer flags nor "wake-up" the MCU. But, when the MCU does wake up, there is an active input capture flag (and data) from the first valid edge that occured during the HALT mode.

If the HALT mode is exited by an external reset then no such input capture flag or data action takes place even if there was a valid input capture edge (at the ICAP pin) during the MCU HALT mode.

PART 10 . SERIAL PERIPHERAL INTERFACE

10.1 . INTRODUCTION AND FEATURES

10.1.1 . INTRODUCTION

The serial peripheral interface (SPI) is an interface which allows several MCUs to be interconnected within a single "black box" or on the same printed circuit board. In a serial peripheral interface (SPI), separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may be configured as containing one master MCU and several slave MCUs, or as a system in which a MCU is capable of being either a master or a slave.

10.1.2 . FEATURES

- Full duplex, three-wire synchronous transfers
- Master or slave operation
- 2 MHz (maximum) master bit frequency
- 4 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transmission interrupt flag
- Write collision flag protection
- Master -Master mode fault protection capability.

10.2 . FEATURES

10.2.1 . SIGNAL DESCRIPTION

The four basic signals (MOSI, MISO, SCK and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both the master and slave mode.

10.2.2 . MASTER OUT SLAVE IN (MOSI)

The MOSI pin is configured as a data output in the master (mode) device and as a data input in the slave (mode) device.

In this manner data is transferred serially from a master to a slave on this line, most significant bit first, least significant bit last. The timing diagram of Figure 10.2 summarizes the SPI timing diagram shown in the electrical specifications (timing tables, Part 12); and show the relation ship between data and clock (SCK).

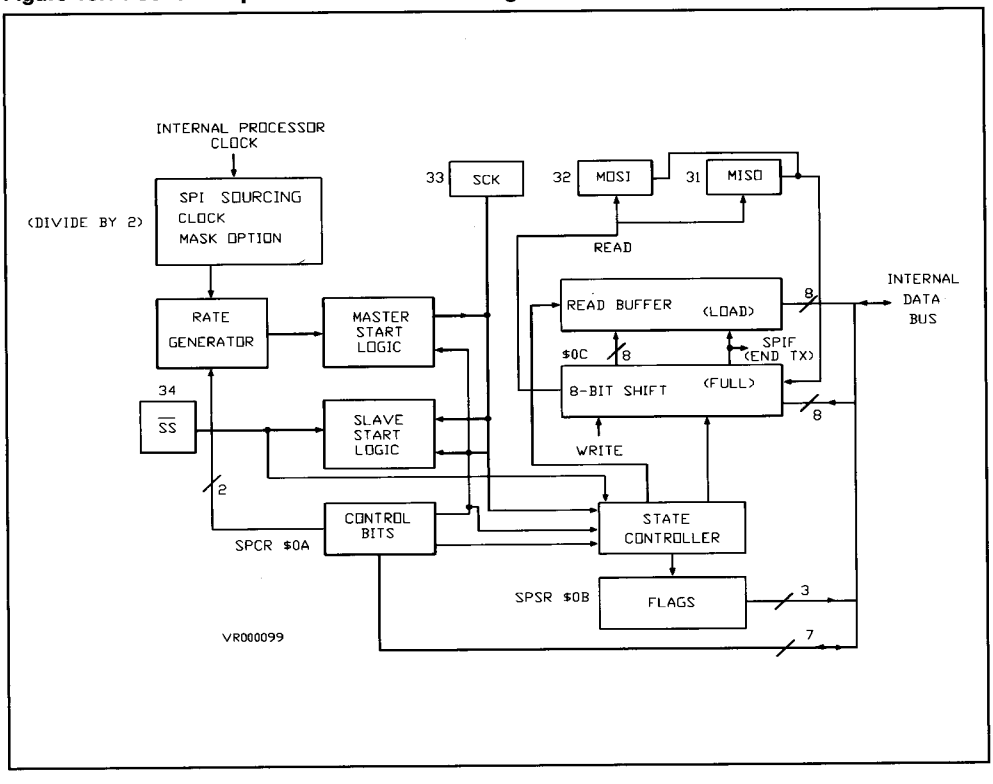
As shown in Figure 10.2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the clock edge (SCK) to allow the slave device to latch the data.

Both the slave device (s) and a master device must be programmed to similar modes for proper data transfer.

When the master device transmits data to a second (slave) device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) is used to signify that the I/O operation is complete.

Configuration of the MOSI pin is a function of the MSTR bit in the serial peripheral control register (SPCR, location $0A). When operating as a master, the user should set the MSTR bit to a logic one, giving the MOSI pin as an output.

Figure 10.1 . Serial Peripheral Interface Block Diagram



Note : The user can select by mask option if the internal processor clock signal is divided by 2 or not before entering the rate generator.

10.2.3 . MASTER IN SLAVE OUT (MISO)
The MISO pin is configured as an input in a master (mode) device and as an output in a slave (mode) device. In this manner data is transferred serially from a slave to a master on this line, most significant bit first, least significant bit last. The MISO pin of a slave device is placed in the high-impedance state if it is not selected by the master, i.e., its $\overline{SS}$ pin is a logic one. The timing diagram of Figure 10.2 shows the relationship between data and clock (SCK). As shown in Figure 10.2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the clock edge (SCK) in order for the slave device to latch the data.

When the master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device.) Thus, the byte transmitted is replaced by the byte received and eliminates the need for seperate transmit-empty and receiver-full bits. A single status bit (SPIF) in the serial peripheral status register (SPSR), location $0B is used to signify that the I/O operation is complete.

In the master device, the MSTR control bit in the serial peripheral control register (SPCR, location $0A) should be to a logic one (by the program) to allow the master device to receive data on its MISO pin. In the slave device, its MISO pin is enabled by the logic 1 level of the $\overline{SS}$ pin, i.e., if $\overline{SS}$ = 1, then the MISO pin is placed in the high-impedance state, whereas if $\overline{SS}$ = 0 the MISO pin is an output for the slave device.

10.2.4 . SLAVE SELECT ($\overline{SS}$)
The slave select ($\overline{SS}$) pin is a fixed input (PD5, pin 34), which receives an active low signal that is generated by the master device to enable slave devices to accept data.

To ensure that data will be accepted by a slave device, the $\overline{SS}$ signal line must be a logic low prior to occurence of SCK (system clock) and must remain low until after the last (eighth) SCK cycle.

Figure 10.2 illustrates the relationship between SCK and the data for two different level combinations of CPHA, when $\overline{SS}$ is pulled low. These are :

a) CPHA = 1 or 0, the first bit of data is applied to the MISO line for transfer, and,

b) when CPHA = 0 the slave device is prevented from writing to its data register. For further information on the effect the $\overline{SS}$ input and the CPHA have on the I/O data register, refer to the WCL status flag in the "serial peripheral status register description" (location $0B). A high level $\overline{SS}$ signal forces the MISO (master in slave out) line to the high-impedance state. Also, SCK and the MOSI (master out slave in) line are ignored by a slave device when its $\overline{SS}$ signal is high.

When a device is a master, it constantly monitors its $\overline{SS}$ signal input for a logic low. The master device will become a slave device any time its $\overline{SS}$ signal input is detected low. This ensures that there is only one master controlling the $\overline{SS}$ line for a particular system.

When the $\overline{SS}$ line is detected low, the master clears the MSTR control bit (serial peripheral control register, location $0A). Also, control bit SPE in the serial peripheral control register is cleared which causes the serial peripheral interface (SPI) to be disabled (port D SPI pins become inputs). The MODF flag bit in the serial peripheral status register (location $0B) is also set to indicate to the master device that another device is attempting to become a master. Two devices attempting to be outputs are normally the result of a software error, however, a system could be configured containing a default master which would automatically "take-over" and restart the system.

10.2.5 . SERIAL CLOCK (SCK)

The serial clock is used to synchronize the movement of data both in and out of the device through its MOSI and MISO pins. The master and slave devices are capable of exchanging a data byte of information during a sequence of eight clock pulses. Since the SCK is generated by the master device, the SCK line becomes an input on all slave devices and synchronizes slave data transfer. The type of clock and its relationship to data are controlled by the CPOL and CPHA bits in the serial peripheral control register (location $0A) discussed below. Refer to Figure 10.2 for timing.

The master device generates the SCK through a circuit driven by the internal processor clock. Two bits (SPR0 and SPR1) in the serial peripheral control register (location $0A) of the master device select the clock rate. The master device uses the SCK to latch incoming slave device data on the MISO line and shifts out data to the slave device on the MOSI line. Both master and slave devices must be operated in the same timing mode as controlled by the CPOL and CPHA bit in the serial peripheral control register. In the slave device, SPR0, SPR1 have no effect on the operation of the serial peripheral interface. Timing is shown in Figure 10.2.

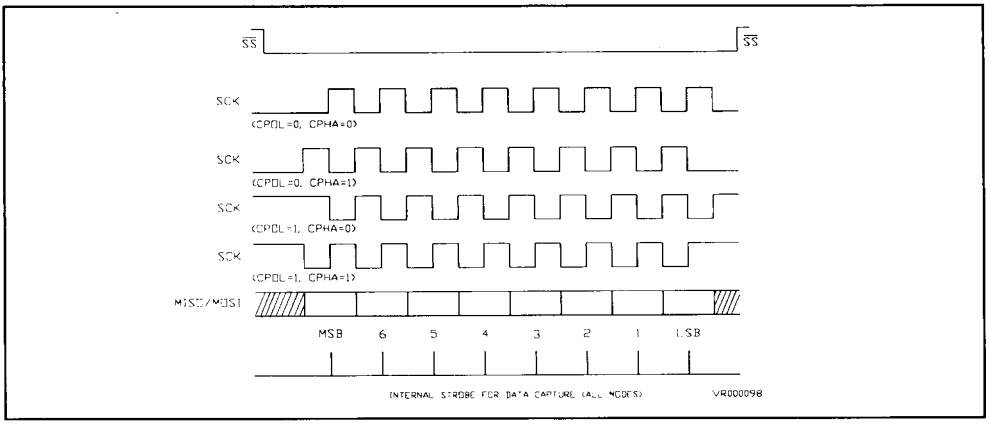
10.3 . FUNCTIONAL DESCRIPTION

A block diagram of the serial peripheral interface (SPI) is shown in Figure 10.1. In a master configuration, the master start logic receives an input from the CPU (in the form of a write to the SPI rate generator data register) and originates the system clock (SCK) based on the internal processor clock. This clock is also used internally to control the state controller as well as the 8 bit shift register.

As a master device, data is parallel loaded into the 8 bit shift register (from the internal bus) during a write cycle and then shifted out serially to the MOSI pin for application to the slave device(s). During a read cycle, data is applied serially from a slave device via the MISO pin to the 8 bit shift register. After the 8 bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low (from a master device) at the $\overline{SS}$ pin and a system clock input (from the same master device) at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave MOSI pin and loads the 8 bit shift register.

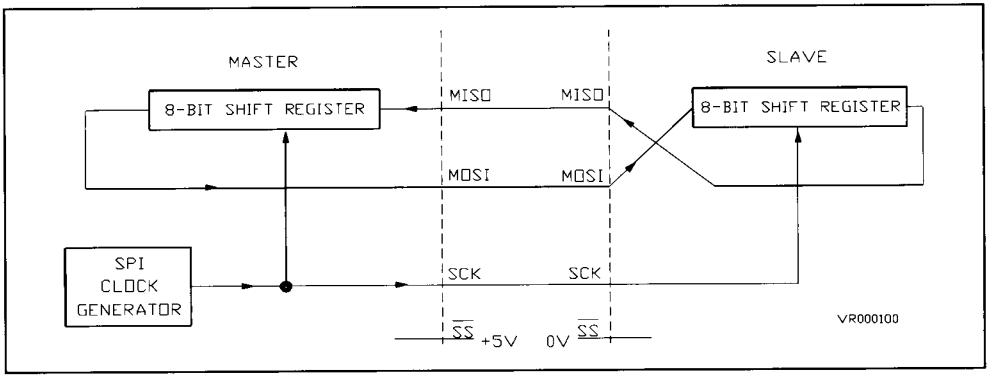
Figure 10.2 . Data Clock Timing Diagram



After the 8 bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle. During a write cycle, data is parallel loaded into the 8 bit shift register from the internal data bus and then shifted out serially to the MISO pin for application to the master device.

Figure 10.3 illustrates the MOSI, MISO and SCK master-slave interconnections. Note that the master $\overline{SS}$ pin is tied to a logic high and the slave $\overline{SS}$ pin is a logic low.

Figure 10.3 . Serial Peripheral Interface Master-Slave Interconnections



10.4 . REGISTERS

There are three registers in the SPI interface which provide control, status, and data storage functions. These registers which include the serial peripheral control register (SPCR, location $0A), the serial peripheral status register (SPSR, location $0B), and the serial peripheral data I/O register (SPDR, location $0C) are described below.

10.4.1 . SERIAL PERIPHERAL CONTROL REGISTER (SPCR)

7	6	5	4	3	2	1	0
SPIE	SPE	–	MSTR	CPOL	CPHA	SPR1	SPR0

The serial peripheral control register bits are defined as follows.

Bit-7 : **SPIE**
When the serial peripheral interrupts enable bit is high, it allows the occurence of a processor interrupt, and forces the proper vector to be loaded into the program counter if the serial peripheral status register flag bit (SPIF and /or MODF) is set to a logic one. It does not inhibit the setting of a status bit. The SPIE bit is cleared by reset.

Bit-6 : **SPE**
when the serial peripheral output enable control bit is set, all output drive is applied to the external pins and the system is enabled. When the SPE bit is set, it enables the SPI system by connecting it to the external pins thus allowing it to interface with the external SPI bus. The pins that are defined as output depend on which mode (master or slave) the device is in . Because the SPE bit is cleared by reset, the SPI system is not connected to the external pins upon reset.

Bit-4 : **MSTR**
The master bit determines whether the device is a master or a slave. If the MSTR bit is a logic zero it indicates a slave device and a logic one denotes a master device. If the master mode is selected, the function of the SCK pin changes from an input to an output and the function of the MISO and MOSI pins are reversed. This allows the user to wire device pins MISO to MOSI, and MOSI to MOSI, and SCK to SCK without incident. The MSTR bit is cleared by reset, therefore, the device is always placed in the slave mode during reset.

Bit-3 : **CPOL**
The clock polarity bit controls the normal or steady state value of the clock when no data is being transferred. The CPOL bit affects both the master and slave modes. It must be used in conjunction with the clock phase control bit (CPHA) to produce the wanted clock-data relationship between a master and a slave device.When the CP0L bit is a logic zero, it produces a steady state low value at the SCK pin of the master device. If the CPOL bit is a logic one, a high value is produced at the SCK pin of the master device when data is not being transferred. The CPOL bit is not affected by reset. Refer to Figure 10.2.

Bit-2 : **CPHA**
The clock phase bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control has effect in both the master or slave modes. It must be used in conjunction with the clock polarity control bit (CPOL) to produce the wanted clock-data relationship. In general the CPHA bit selects the clock edge which captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the first data capture edge. The CPHA bit is not affected by reset. Refer to Figure 10.2

Bit-1 : **SPR1** and Bit-0 : **SPR0**
These two serial peripheral rate bits select one of four baud rates to be used as SCK if the device is a master. However, these 2 bits have no effect in the slave mode. The slave device is capable of shifting data in and out at a maximum rate which is equal to the CPU clock. A rate table is given below for the generation of the SCK from the master. The SPR1 and SPR0 bits are not affected by reset.

SPR1	SPR2	Internal Processor Clock Divide*
0	0	2
0	1	4
1	0	16
1	1	32

* In this case, the SPI sourcing clock mask option (divider by 2) has not been selected.

10.4.2 . SERIAL PERIPPHERAL STATUS REGISTER (SPSR)

7	6	5	4	3	2	1	0
SPIF	WCOL	–	MODF	–	–	–	–

The status flags which generate a serial peripheral interface (SPI) interrupt may be blocked by the SPIE control bit in the serial peripheral control register. The WCOL bit does not cause an interrupt. The serial peripheral status register bits are defined as follows :

Bit-7 : **SPIF**
The serial peripheral data transfer flag bit notifies the user that a data transfer between the device and an external device has been completed. With the completion of the data transfer, SPIF is set, and if SPIE is set, a serial peripheral interrupt (SPI) is generated. During the clock cycle SPIF is being set, a copy of the received data byte in the shift register is moved to a buffer. When the data register is read, it is the buffer that is read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not responded to the first SPIF, only the first byte sent is contained in the receiver buffer and all other bytes are lost.

The transfer of data is initiated by the master device writing to its serial peripheral data register.

Clearing the SPIF bit is accomplished by a software sequence of accessing the serial peripheral status register while SPIF is set and followed by a write to or a read of the serial peripheral data register.

While SPIF is set, all writes to the serial peripheral data register are inhibited until the serial peripheral status register is read. This occurs in the master device. In the slave device, SPIF can be cleared (using a similar sequence) during a second transmission ; however, it must be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

Bit-6 : **WCOL**

The function of the write collision status bit is to notify the user that an attempt was made to write the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted ; therefore, a write will be unsuccessful. A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation. If a "write collision" occurs, WCOL is set but no SPI interrupt is generated. The WCOL bit is a status flag only.

Clearing the WCOL bit is accomplished by a software sequence of accessing the serial peripheral status register while WCOL is set, followed by :
1) A read of the serial peripheral data register prior to the SPIF bit being set, or
2) A read or write of the serial peripheral data register after the SPIF bit is set. A write to the serial peripheral data register (SPDR) prior to the SPIF bit being set, will result in generation of another WCOL status flag. Both the SPIF and WCOL bits will be cleared in the same sequence. If a second transfer has started while trying to clear (the previously set) SPIF and WCOL bits with a clearing sequence containing a write to the serial peripheral data register, only the SPIF bit will be cleared.

A collision of a write to the serial peripheral data register while an external data transfer is taking place can occur both in the master mode and the slave mode, although with proper programming the master device should have sufficient information to preclude this collision.

Collision in the master device is defined as a write of the serial peripheral data register while the internal rate clock (SCK) is in the process of transfer.

The signal on the $\overline{SS}$ pin is always high on the master device.

A collision in a slave device is defined in two separate modes. One problem arises in a slave device when the CPHA control bit is a logic zero.
When CPHA is logic zero, data is latched with the occurence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when it attempts to write the serial peripheral data register after its $\overline{SS}$ pin has been pulled low. The $\overline{SS}$ pin of the slave device freezes the data in its serial peripheral data register and does not allow it to be altered if the CPHA bit is a logic zero.

The master device must raise the $\overline{SS}$ pin of the slave device high between each byte it transfers to the slave device

The second collision mode is defined for the state of the CPHA control bit being a logic one. With the CPHA bit set, the slave device will be receiving a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device I/O register and allow the msb on to the external MISO pin of the slave device. The $\overline{SS}$ pin low state enables the slave device but the drive onto the MISO pin does not take place until the first data transfer clock edge. The WCOL bit will only be set if the I/O register is accessed while a transfer is taking place. By definition of the second collision mode, a master device might hold a slave device $\overline{SS}$ pin low during a transfer of several bytes of data without a problem.

Unlike other SPI interfaces, there is no special case of collision undetected by the WCOL bits. The WCOL bit is totally reliable for collision detection.

Since the slave device is operating asynchronously with the master device, the WCOL bit may be used as an indicator of a collision occurence. This helps alleviate the user from a strict real-time programming effort. The WCOL is cleared by reset.

Bit-4 : **MODF**

The function of the mode fault flag is defined for the master mode (device). If the device is a slave device the MODF bit will be prevented from toggling from a logic zero to a logic one; however, this does not prevent the device from being in the slave mode with the MODF bit set.

The MODF bit is normally a logic zero and is set only when the master device has its $\overline{SS}$ pin pulled low. Toggling the MODF bit to a logic one affects the internal serial peripheral interface (SPI) system in the following ways :
a) MODF is set and SPI interrupt is generated if SPIE = 1.
b) The SPE bit is forced to a logic zero. This blocks all output drive from the device and, disables the SPI system.
c) The MSTR bit is forced to a logic zero, thus forcing the device into the slave mode.

Clearing the MODF is accomplished by a software sequence of accessing the serial peripheral status register while MODF is set followed by a write to the serial peripheral control register.

To avoid any multi slave conflict in the case of a system of several MCUs, the $\overline{SS}$ pin must be pulled high during the clearing sequence of MODF. Control bits SPE and MSTR may be restored to their original set state during this clearing sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bit while MODF is a logic one unless it is during the proper clearing sequence. The MODF flag bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF bit is cleared by reset.

10.4.3 . SERIAL PERIPHERAL DATA I/O REGISTER (SPDR)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

The serial peripheral data I/O is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. A slave device writing to its data I/O regiter will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices.

A write or read of the serial peripheral data I/O register, after accessing the serial peripheral status register with SPIF set, will clear SPIF.

During the clock cycle the SPIF bit is being set, a copy of the received data byte in the shift register is being moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not internally responded to clear the first SPIF, only the first byte is contained in the receive buffer of the slave device; all others are lost. The user may read the buffer at any time. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

The ability to access the serial peripheral data I/O register is limited when a transmission is taking place. It is important to read the discussion defining the WCOL and SPIF status bits to understand the utilization limits of the serial peripheral data I/O register.

10.5 . SINGLE MASTER AND MULTIMASTER CONFIGURATIONS

There are two types of SPI systems, single master system and multi-master systems.

A typical single master system may be configured, using a MCU as the master and fours MCUs as slaves. The MOSI, MISO and SCK pins are all wired to equivalent pins on each of the five devices. The master device generates the SCK clock,the slave devices all receive it. Since the MCU master device is the bus master, it internally controls the function of its MOSI and MISO lines, thus writing data to the slave devices on the MOSI and reading data from the slave devices on the MISO lines.

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices. A slave device is selected when the master device pulls its $\overline{SS}$ pin low. The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices. Note that the slave devices do not have to be enabled in a mutually exclusive fashion except to prevent bus contention on the MISO lines. For example, three slave devices, enabled for a transfer, are permissible if only one has the capability of being read by the master. An example of this is a write to several display drivers to clear a display with a single I/O operation.

To ensure that proper data transmission is occuring between the master device and a slave device, the master device may have the slave device respond with a previously received data byte (this data byte could be inverted or at least be a byte that is different from the last one sent by the master device). The master device will always receive the previous byte back from the slave device if all MISO and MOSI lines are connected and the slave has not written its data I/O register. Other transmission security methods might be defined using ports for handshake lines or data bytes with command fields.

A multi-master system may also be configured by the user. An exchange of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system. The major device control that plays a part in this system is the MSTR bit in the serial peripheral control register and the MODF bit in the serial peripheral status register.

PART 11 . SERIAL COMMUNICATION INTERFACE

11.1 . INTRODUCTION

The SCI (Serial Communications Interface) provides full duplex, asynchronous communication in NRZ code at variable transmission speeds. The transmitter and the receiver are functionally independent but operate at the same baud rate.
Serial data format is "Mark/Space" (NRZ)-type which corresponds to one START bit, eight or nine data bits and one STOP bit. The RDI and TDO pins are the received input and transmitted data output pins.

11.1.1 . SCI COMMUNICATION CHARACTERISTICS

- NRZ standard format (Mark/Space).
- Frame Error detection and detection of any noise longer than 1/16 th of a bit based on a 1/16 th bit clock rate (RT).
- Two-way transmission
- Software-programmable transmission and reception speed (32 speeds available).
- Software-programmable word length (eight or nine bits).
- Different enable bits for the transmitter and the receiver.
- The SCI can be interrupt-driven.
- The various interrupts can be enabled by four different bits.

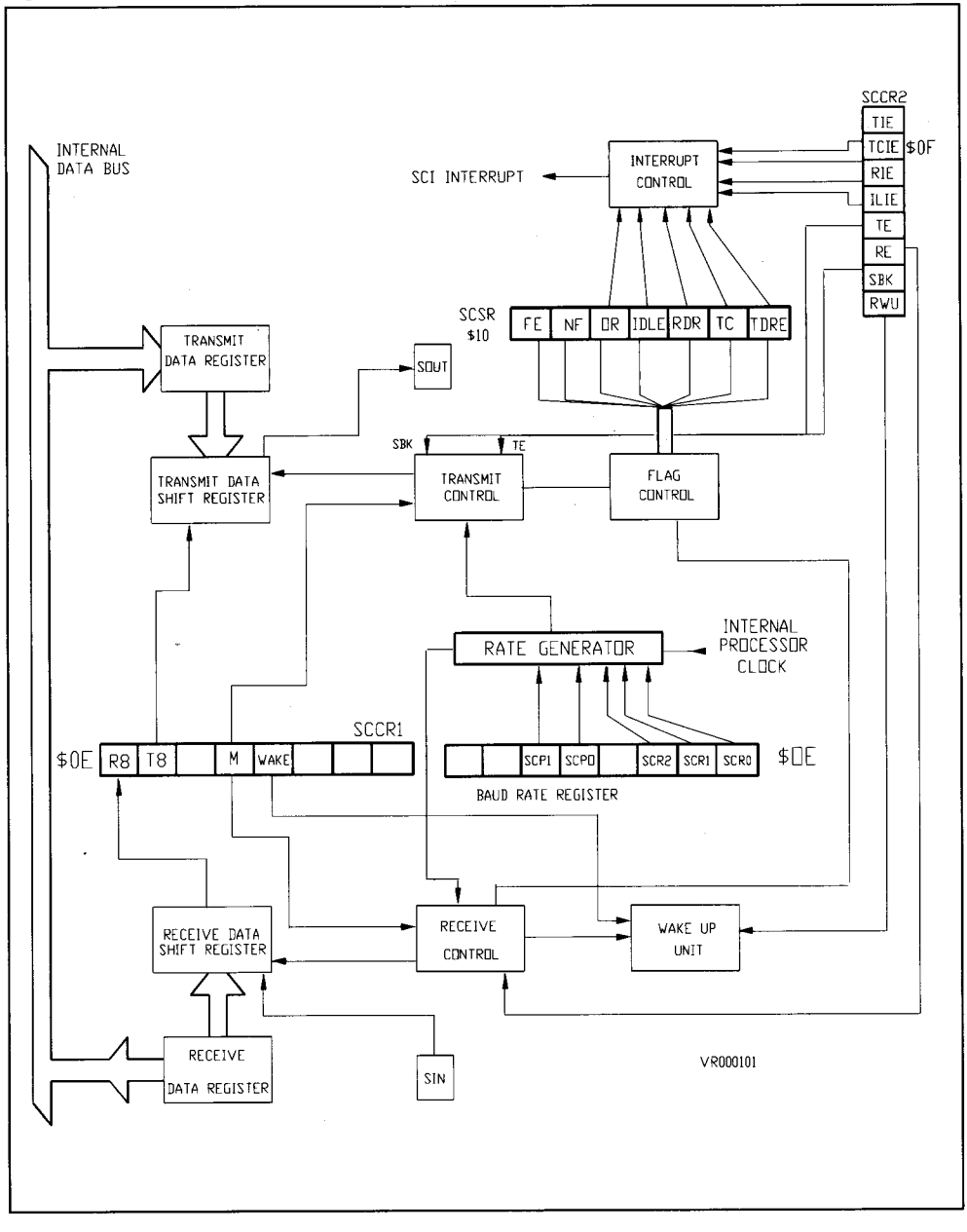
11.1.2 . RECEIVER CHARACTERISTICS

- Wake up function by the most significant bit or by an idle line.
- Idle line detection.
- Frame error detection.
- Noise detection.
- Overrun detection.
- Receiver data full flag.

11.1.3 . TRANSMITTER CHARACTERISTICS

- Transmission register empty flag.
- End of transmission flag.
- Break transmission (all bits at 0 including STOP bit).

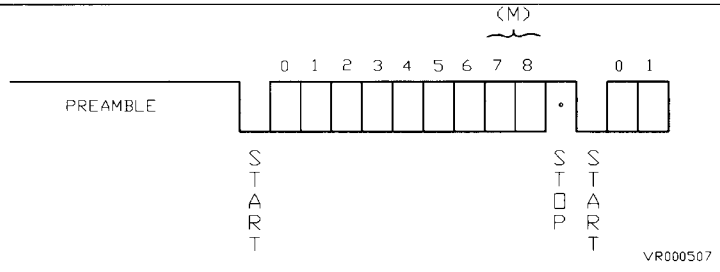
Figure 11.1 . Serial Communication Interface Block Diagram



11.2 . DATA FORMAT

Transmission or reception data format conforms to the Figure below :

Figure 11.2 . Data Format



Note: Words are sent LSB first,
A START bit is transmitted at the beginning of each word,
The START bit must be preceded by a logic high level,
A STOP bit = "1" is sent in 8th or 9th position except for transmission of a break,
The quiescent line is at "1"; this status precedes transmission.
A break is defined by 10 or 11 bits at "0".
A preamble is defined by 10 or 11 bits at "1".

(M) Control bit M : if M = 1, 9 bit data. If M = 0, 8 bit data.

* the STOP bit is at "0" for a break transmission

11.3 . MUTING FUNCTION

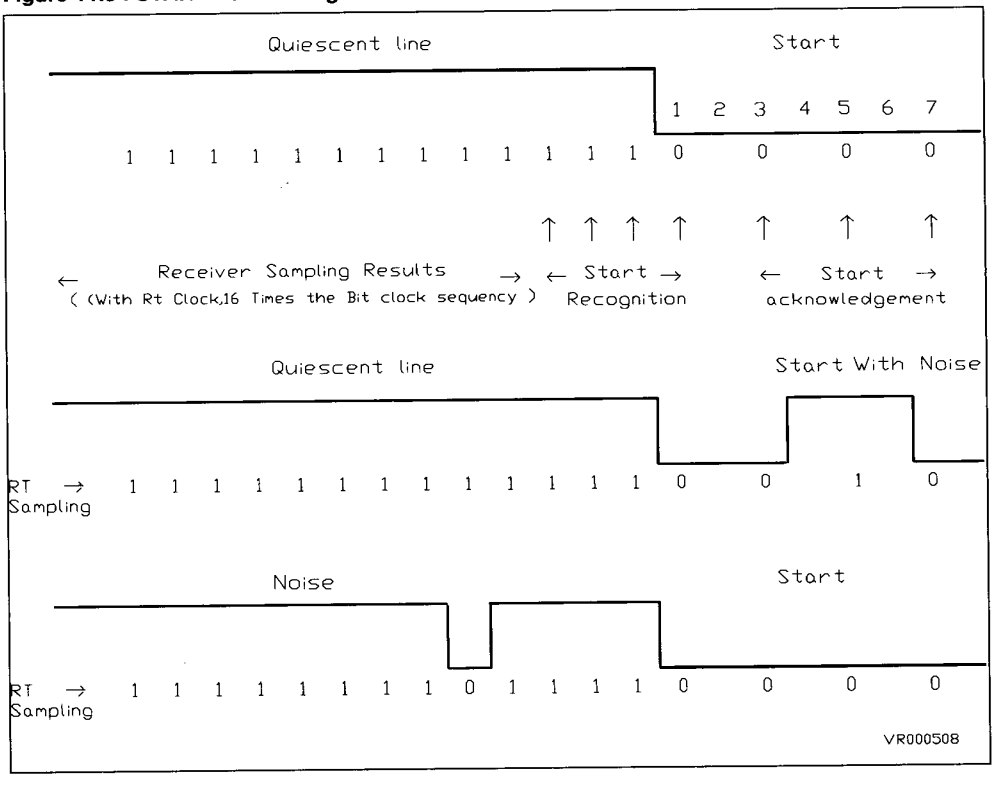
In a multiprocessor configuration, communication protocols generally provide for an address at the beginning of the messages. In order to tell MPUs not concerned by the current message to ignore the message, a muting function is included which inhibits all flags and interrupts on reception. The receiver is woken up either by a quiescent line (10 or 11 consecutive "1"s) or by reception of a "1" on the most significant bit (8th bit for M = "0", 9th for M= "1"); the choice is software programmable.

11.4 . DATA RECEPTION

The receiver samples data at a rate 16 times faster than the bit rate (see figure below). This clock will be referred to by the symbol RT. It is controlled by the BRR register which selects transmitter and receiver clock rates.

The START bit will be recognized at RT clock rate if at least three consecutive "1"s followed by the first "0" are received and a majority of "0"s are recognized in the three samples located at the 3rd, 5th and 7th RT clock signals.

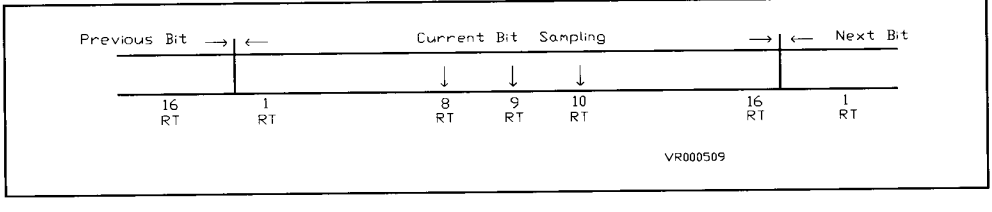
Figure 11.3 . START Acknowledgment



All bits making up a frame, including start and stop bits, are sampled at the 8th, 9th and 10th RT clock signals as soon as the start bit is acknowledged as in the previous paragraph.

The value assigned to bits will be the value returned by the majority of the three samples which raise the noise flag to "1" if all three are not equal.

Figure 11.4 . Bit Sampling at Reception

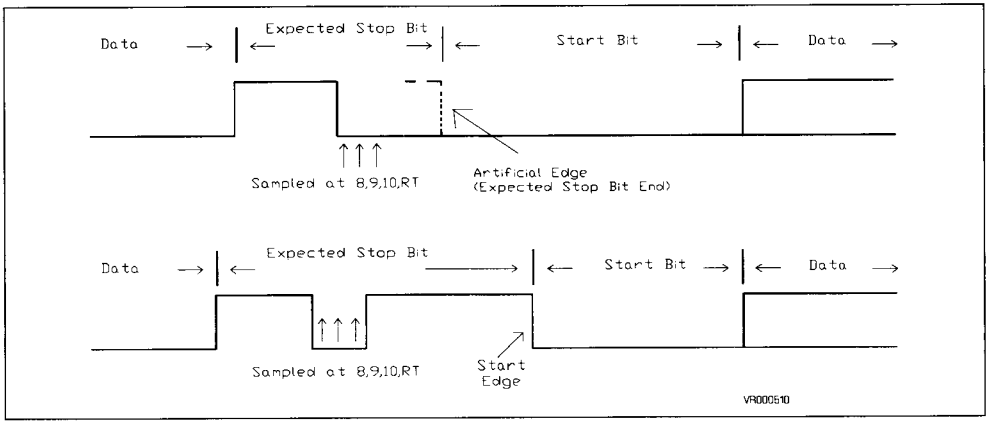


11.5 . START DETECTION FOLLOWING A FRAME ERROR

If a frame error has occured without a break (stop bit sampled at "0" for a word different to "0"), the circuit continues operating as if a stop bit had been received.

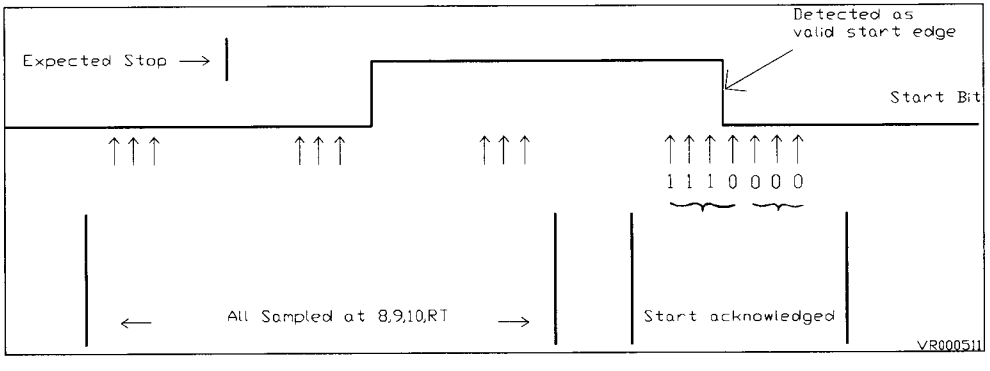
The falling edge corresponding to the end of the STOP bit is restored to its expected position. If the STOP bit is longer than expected, the START will be synchronous with the real falling edge.

Figure 11.5. START Acknoledgement After a Frame Error



A BREAK is a "0" data with a "0" STOP bit. When a break is received, the receiver does not generate a falling edge; a "1" bit must first be received before the start is recognized.

Figure 11.6. START Acknowledgement at the End of a BREAK



11.6. DATA TRANSMISSION

The format of output data is similar to input data format, described above.
When the transmitter is enabled but nothing is to be transmitted the serial output pin (SOUT) goes to high (idle) state.
When the transmitter is disabled, the serial output pin (SOUT) goes to high impedance state.

11.7. REGISTERS

The SCI has 5 registers. Their internal configurations are explained below.

11.7.1. DATA REGISTER (SCDAT)
The SCI data register (SCDAT) is mapped at address $11. It performs a double function (read and write) since it is composed of two registers, one for transmission and the other for reception: (TDR) and (RDR) respectively.

The TDR register provides the parallel interface between the internal bus and the output shift register. The RDR register provides the parallel interface between the input shift register and the internal bus.

When the SCDAT is read, the RDR is accessed and its content is transferred to the bus . The RDRF (RDR Full flag) is set to "1" as soon as the word in the reception shift register is transferred to the RDR.

A write operation in the SCDAT addresses the TDR which receives the data on the bus. The TDRE flag (TDR Empty) is set to "1" as soon as the word in the TDR is transferred to the output shift register.

11.7.2. CONTROL REGISTER 1 (SCCR1)

Figure 11.7 . SCCR1

7	6	5	4	3	2	1	0	ADDRESS
R8	T8		M	WAKE				= $0E

The SCCR1 :

- Supplies control bits which :
  determine word length (8 or 9 bits)
  select the wake up mode used.
- Stores the 9th transmission or reception bits.

B7 => R8 : if bit M is at "1", B7 will be used to store the 9th bit on reception. Reinitialization of the system does not effect this bit.

B6 => T8 : if the M is at "1", B6 will be used to store the 9th bit on transmission. Reinitialization of the system does not effect this bit.

B4 => M : this bit gives the option on word length. Reinitialization of the system does not effect this bit.

- M = "0" => 1 start bit, 8 data bits and 1 stop bit.
- M = "1" => 1 start bit, 9 data bits and 1 stop bit.

B3 => WAKE : this bit selects the wake up mode. If WAKE = "0" an idle line will activate wake up allowing reception to begin at the next word received. If WAKE = "1" the system will wait for a word with the most significant bit at "1" to be received to activate wake up by starting reception as from the same word which can therefore be used as an address word. Reinitialization of the system does not effect bit M.

TABLE 11.1. BITS 3 AND 4 OF SCCR1

Bit 3 (Wake)	Bit 4 (M)	Receiver wake up mode
0	X	Wake Up On Detection Of a Quiescent (idle) Line
1	0	Wake Up On Detection Of The 8th Bit At "1"
1	1	Wake Up On Detection Of The 9th Bit At "1"

Note : To avoid all risks of loosing bits T8 and R8 in 9 bit mode, R8 must be read before reading the RDR and T8 written before writing into the TDR.

11.7.3. CONTROL REGISTER 2 (SCCR2)

Figure 11.8 . SCCR2

7	6	5	4	3	2	1	0	ADDRESS = $0F
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	

The SCCR2 supplies the command bits which :

- activate or deactivate the transmitter or receiver.
- enable system interrupts,
- enable the muting function or cause a break to be sent.

B7 => TIE : the TIE bit at "1" authorizes an interrupt when the TDRE (transmission register empty) flag is at "1" indicating that the last word has been transmitted. When TIE bit is at "0" this interrupt is disabled. Reinitialization of the system resets the TIE bit to "0".

B6 = > TCIE : this bit at "1" enables an interrupt when the TC flag (transmission completed) changes to "1". The TCIE bit at "0" disables this interrupt. Reinitialization of the system resets this bit to "0".

B5 = > RIE : this bit at "1" enables an interrupt when either the RDRF flag (receive data register full) or the OR flag (overspeed on reception) is at "1". The RIE bit at "0" disables this interrupt. Reinitialization of the system resets the RIE bit to "0".

B4 => ILIE : this bit at "1" enables an interrupt if the IDLE flag changes to "1", i.e. which corresponds to an idle line on reception. The interrupt cannot occur if the ILIE bit is at "0". Reinitialization of the system resets the ILIE bit to "0".

B3 => TE : this bit at "1" enables the transmitter. At start-up, the transmitter sends a preamble (ten or eleven "1"s).During transmission, a "0" pulse on the TE bit (a "0" write followed by a "1" write) sends a preamble after the current word. Setting the TE bit to "0" switches the output line to high impedance state at the end of the word currently being transmitted. Reinitialization of the system resets the TE bit to "0".

B2 => RE : the RE bit at "1" enables the receiver which begins searching for a START bit. The RE bit at "0" deactivates the receiver, and resets the associated status bits to zero (RDRF, IDLE, OR, NF and FE). Reinitialization of the system resets the RE bit to "0".

B1 = > RWU : the RWU bit at "1" mutes the receiver. The wake-up mode is determined by the WAKE bit (bit B3 in the SCCR1 register). As long as RWU remains at "1", the flags relative to the receiver cannot raise to "1".

Software write of a "0" to RWU forces the exit from the muting state.
As soon as the wake up sequence is recognized, the RWU bit is forced to "0". If the wake up mode selected corresponds to the reception of a preamble, the RWU bit cannot be set to "1" for as long as the reception line remains idle. If the selected wake up mode corresponds to the reception of a "1" on the most significant bit, the reception of this particular word wakes up the receiver and sets the RDRF flag to "1" which allows the receiver to receive this word normally and use it as an address word. Reinitialization of the system resets the RWU bit to "0".

B0 = > SBK : this bit at "1" tells the transmitter to send a whole number of BREAKS (all bits at "0" including the STOP bit). At the end of the last BREAK the transmitter inserts an extra "1" bit so that the START bit is acknowledged. If the SBK bit is set to "1" then to "0", the transmitter will send a BREAK word at the end of the current word. Reinitialization of the system resets the SBK bit to "0".

Figure 11.9 . SCSR

7	6	5	4	3	2	1	0	ADDRESS = $10
TDRE	TC	RDRF	IDLE	OR	NF	FE		

11.7.4. STATUS REGISTER (SCSR)

The SCSR register supplies the flag generating SCI interrupts NF (reception noise) and FE (frame error) flags.

SCSR flags relative to reception are reset to "0" by an access to SCSR followed by a read operation of the reception data register (RDR). SCSR flags relative to transmission are reset to "0" by an access to SCSR followed by a write operation into the transmission data register (RDR). Except tor TC, the only bits reset to "0" will be the ones that were actually at "1" at the time the SCSR was read.

Note :

- One or several accesses of the SCSR alone do not cancel flags.
- One or several read operations of the reception data register without prior access of the SCSR do not lower corresponding flags. This causes the same word to be read, finally leading to reception overrun.
- One or several write operations into the transmission data register without prior access of the SCSR do not lower corresponding flags. This causes loss of the first words written into the register and none of them are sent.

B7 = > TDRE : the TDRE bit at "1" indicates that the content of the transmission data register has been transferred into the shift register. If the TDRE bit is at "0", it indicates that transfer has not yet occured and that a write operation into the data register would overwrite previous data. The TDRE bit is reset to "0" by an SCSR access to the followed by a write operation into the transmission data register. Data will not be transmitted into the shift register as long as the TDRE bit is not reset to "0". Reset sets the TDRE bit to "1".

B6 = > TC : the TC bit is set to "1" at the end of a frame whether it includes data, a preamble or a BREAK if :

- TE = "1", TDRE = "1", no words are currently being transmitted and no preamble or BREAK are awaiting transmission,
- TE = "0" and the current word, preamble or BREAK has been transmitted.

The TC bit is a flag indicating that one of the above sequences has occurred. This bit is reset to "0" by an access to the SCSR followed by a write operation into the data register or when TDRE is reset to "0". Transmitter operation is in no way modified whatever the status of this bit. Reinitialization of the system sets the TC bit to "1".

B5 = > RDRF : this bit at "1" indicates that the content of the reception data register has been transferred into the data register. If a frame error or noise has been detected during reception of the word, the corresponding flags will be raised at the time of transfer. The RDRF bit is reset to "0" by an access to the SCSR followed by a data register read operation. Reset clears the RDRF bit.

B4 = > IDLE : When the idle line detect bit is set, it indicates that a receiver idle line is detected (receipt of a minimum number of ones to constitute the number of bits in the frame format). The minimum number of ones needed will be 10 (M = 0) or 11 (M = 1). This allows a receiver that is not in the wake-up mode to detect the end of a message, detect the preamble of a new message, or to resynchronize with the transmitter. The IDLE bit is cleared by accessing the serial communications status register (with IDLE set) followed by a read of the serial communications data register. The IDLE bit will not be set again until after an RDRF has been set ; i.e., a new idle line occurs. The IDLE bit is not set by an idle line when the receiver "wakes-up" from the wake-up mode. Reset clears the IDLE bit.

B3 = > OR : this bit is set to "1" when the word currently being received in the shift register is ready to be transferred into the data register while the latter is already full (RDRF = "1"). All transfers will remain disabled as long as RDRF stays at "1". Data register content will not be lost ,but the content of the reception shift register will be overwritten. The OR bit is reset to "0" by an access to the SCSR followed by a data register read operation. Reset clears the OR bit.

B2 = > NF : the NF bit is set to "1" when noise is detected on an acknowledge START bit, a data or a STOP bit. The NF bit is not set to "1" as soon as the noise is detected, but only when the RDRF bit changes to "1"; the NF bit is therefore representative of the word present in the data register. This bit does not generate interrupts since it is set to "1" at the same time as the RDRF bit which generates an interrupt itself. The NF bit is set to "0" by a SCSR read ope-ration followed by a data register read operation. Reset clears the NF bit.

B1 = > FE : the FE bit is set to "1" when the STOP bit is not recognized on reception at the expected moment, following either a de-synchronization , excessive noise or when a break is received. The word will, however, still be transferred to the data register. As in the case of the NF bit, the FE bit does not generate interrupts since it is set to "1" at the same time as the RDRF bit. If the word currently being received causes both a frame error and reception overspeed, it will not be transferred and only OR bit will be set to "1". The FE bit is reset to "0" by a SCSR read operation followed by a data register read operation. Reset clears the FE bit.

11.7.5. BAUD RATE SELECTION REGISTER (BRR)

Figure 11.10. BRR

7	6	5	4	3	2	1	0	ADDRESS = $OD
		SCP1	SCP0		SCR2	SCR1	SCR0	

This register contains data used to determine the baud rates. Bits SCP0 and SCP1 define a prescaling factor and a set of 8 frequencies. The transmitter and receiver baud rate is then chosen among the 8 according to SCR2, SCR1, SCR0. This register must not be changed or even written to while the transmitter or the receiver are enabled.

B5 = SCP1:
B4 = SCP0
these 2 prescaling bits make it possible to obtain several standard clock rate ranges.
reinitialization of the system sets these bits to 0.

TABLE 11.2.

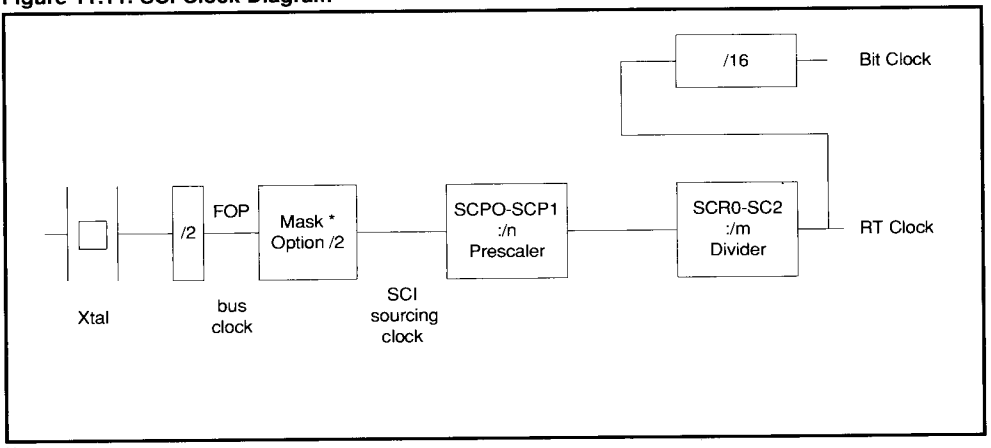
SCP1	SCP0	n Prescaling Factor
0	0	1
0	1	3
1	0	4
1	1	13

B2 = SCR2, B1 = SCR1, B0 = SCR0: these 3 bits in conjunction with the 2 previous bits define the total division applied to the bus clock before it becomes the bit clock.

TABLE 11.3.

SCR2	SCR1	SCR0	m Dividing Factor
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Figure 11.11. SCI Clock Diagram



Note:
1 *If the SCI sourcing clock mask option is enabled (selected by user), the bus clock is divided by 2 before entering the n prescaler.
2 The mask optionnal divider (division by 2) allows software compatibility with MCU using a XTAL to bus cycle ratio of four, while doubling the computing power.

TABLE 11.4. MAXIMUM BIT FREQUENCY (HZ) DEPENDING ON PRESCALER RATIO

SCP bits		Divided by	XTAL Frequency in MHz				
1	0		**8.0**	**4.1943**	4.0	**2.4576**	**1.8432**
0	0	1	250,000	131,072	125,000	76,800	57,600
0	1	3	83,333	43,691	41,666	25,600	19,200
1	0	4	62,500	32,768	31,250	19,200	14,400
1	1	13	19,231	10,082	9,615	5,908	4,430

TABLE 11.5. SCI BIT FREQUENCY

SCP bits			Divided by	Maximum bit frequency choosen by the prescaler ratio (Hz)				
2	1	0		**250,000**	**32,768**	**76,800**	**19,200**	**9,600**
0	0	0	1	250,000	32,768	76,800	19,200	9,600
0	0	1	2	125,000	16,384	38,400	9,600	4,800
0	1	0	4	62,500	8,192	19,200	4,800	2,400
0	1	1	8	31,250	4,096	9,600	2,400	1,200
1	0	0	16	15,625	2,048	4,800	1,200	600
1	0	1	32	7,813	1,024	2,400	600	300
1	1	0	64	3,906	512	1,200	300	150
1	1	1	128	1,953	256	600	150	75

Remark: the values obtained in Table 11.5 take into account the four divisions applied to the external clock before it becomes the SCI bit clock (XTAL divided by 2, prescaling in Table 11.4, division in table 11.5 followed by division by 16).
The extra mask optionnal division by 2 has not been selected.

11.8.1. SCI OPERATION SUMMARY

This summary will provide the programmer with the essential information on SCI control and behaviour in a few lines.

11.8.2. STATUS AND CONTROL BITS

SCCR1 register (R8, T8, M, WAKE)
R8: 9th bit received if M ="1"
T8: 9th bit transmitted if M ="1"
M: word lengh
M = "1" = nine data bits
M = "0" = eight data bits
WAKE : Wake up mode selection:

- WAKE = "0" = wake up by a preamble or quiescent line
- Wake = "1" = wake up by most significant bit at "1".

SCCR2 register (TIE, TCIE, RIE, ILIE, TE, RE, RWU, SBK)
TIE: authorizes interrupt for transmission data register empty (TDRE = "1")
TCIE: authorizes interrupt for transmission completed (TC = "1")
RIE: authorizes interrupt for reception overspeed or reception register full (OR = "1" or RDRF = "1")
ILIE: authorizes interrupt for a quiescent line (IDLE = "1")
TE: enables the transmitter
RE: enables the receiver
RWU: enter (1) or exit (0) the muting function
SBK: BREAK send request

SCSR (TDRE, TC, RDRF, IDLE, OR, NF, FE)
TDRE: transmission register empty
TC: transmission completed
RDRF: reception register full
IDLE: indicates reception of a quiescent line (high status)
OR: reception overspeed
NF: noise on reception
FE: frame error

TABLE 11.6. TRANSMISSION

Action on Command Bits			Effect at the End of The Frame Currently Transmitted
TE	SBK	TDRE	
0	x	x	The Line Changes to High Impedence
1	0	0	Data is Transmitted
1	0	1	Quiescent Line (High Level)
1	1 or	x	Break Sent
	0	x	Preamble Sent (ten or eleven "1"s)

Notes :

 Indicates that the TE bit changes to "0" then to "1" during the current frame in transmission.

 Indicates that the SBK bit changes to "1" then to "0" during the current frame in transmission.

TABLE 11.7. REGISTER TABLE

Add	Register	7	6	5	4	3	2	1	0
$0D	BRR			SCP1	SCP0		SCR2	SCR1	SCR0
$0E	SCCR1	R8	T8		M	WAKE			
$0F	SCCR2	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
$10	SCSR	TDRE	TC	RDRF	IDLE	OR	NF	FE	
$11	RDR	R7	R6	R5	R4	R3	R2	R1	R0
$11	TDR	T7	T6	T6	T4	T3	T2	T1	T0

PART 12. ST8108 ELECTRICAL SPECIFICATIONS

12.1. MAXIMUM RATINGS

The ST8108 device contains circuitry to protect the inputs againts damage due to high static voltage or electric field. Never the less it is advised to take normal precautions and to avoid applying to this high impedance voltage circuit any voltage higher than the maximum rated voltages. It is recommended for proper operation that V_{IN} and V_{OUT} be constrained to the range :

- $V_{SS} \leq V_{IN}$ or $V_{OUT} \leq V_{DD}$

To enhance reliability of operation, it is recommended to connect unused inputs to an appropriate logic voltage level such as V_{SS} or V_{DD}.

All the voltage in the following tables are referenced to V_{SS}

TABLE 12.1. MAXIMUM RATINGS (Voltage Referenced to Vss)

Symbol	Ratings	Value	Unit
V_{DD}	Supply Voltage	-0.3 to +7.0	V
V_{IN}	Input Voltage	V_{SS} -0.3 to V_{DD} + 0.3	V
I	Current Drain Per Pin Excluding V_{DD} and V_{SS}	25	mA
T_A	Operating Temperature Range	T_L to T_H	°C
	ST8108B1 (Standard)	0 to +70	°C
	ST8108B6 (Extended)	-40 to +85	°C
	ST8108B3 (Automotive)	-40 to +125	°C
T_{STG}	Storage Temperature Range	-65 to +150	°C

TABLE 12.2. THERMAL CHARACTERISTICS

Symbol	Characteristics	Value	Unit
θJA	Thermal Resistance		°C/W
	Ceramic	50	
	Plastic	60	
	Plastic leaded Chip Carrier (PLCC)	70	

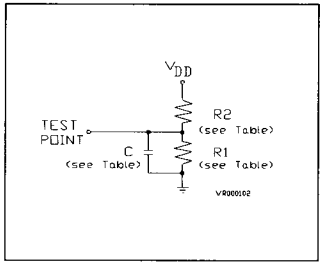
Figure 12.1. Equivalent Test Load

V_{DD} = 4.5V

Pins	R1	R2	C
PA0-PA7	3.26kΩ	2.38kΩ	50pF
PB0-PB7			
PC0-PC7			
PD6			
PD1-PD4	1.9kΩ	2.26kW	200pF

V_{DD}= 3.0V

Pins	R1	R2	C
PA0-PA7	10.91kΩ	6.32kΩ	50pF
PB0-PB7			
PC0-PC7			
PD6			
PD1-PD4	6kΩ	6kW	200pF



12.2. POWER CONSIDERATIONS

T_J, the average chip-junction temperature in Celsius can be calculated from the following equation:

$$T_J = T_A + (P_D \cdot \theta J_A) \quad (1)$$

Where:

- T_A is the Ambient Temperature in °C,
- θJ_A ithe Package Thermal Resistance, Junction-to-Ambient in °C/W,
- P_D the sum of P_{INT} and $P_{I/O}$,
- P_{INT} equals I_{CC} time V_{CC}, Watts-Chip Internal Power
- $P_{I/O}$ the Power Dissipation on Input and Output Pins, User Determined.

For most applications $P_{I/O} < P_{INT}$ and can be neglected .

P_{PORT} may be significant if the device is configured to drive Darlington bases or sink LED Loads.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Therefore :

$$K = P_D \cdot (T_A + 273°C) + \theta J_A \cdot P_D^2 \quad (3)$$

Where K is constant pertaining to the particular part, K can be determined from equation (3) by measuring P_D (at equilibrium) for a kwon T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A

Table 12.4. DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 V_{dc} ± 10%, V_{SS} = 0 V_{dc}, T_A = T_L to T_H unless otherwise noted)

Symbol	Characteristics	Min.	Typ.	Max.	Unit
V_{OL}	Output Voltage, Load ≤ 10.0 µA			0.1	V
V_{OH}		V_{DD}-0.1			V
	Output High Voltage				
V_{OH}	(I LOAD = 0.8 mA) PA0-PA7,PB0-PB7,PC0-PC7,OCMP (See Fig.9.2)	V_{DD}-0.8			V
V_{OH}	(I LOAD = 1.6 mA)PD1-PD4 (See Fig. 9.3)	V_{DD}-0.8			V
V_{OL}	Output Low Voltage (See Fig. 9.4)				V
	(I LOAD = 1.6 mA) PA0-PA7,PB0-PB7,PC0-PC7,PD1-PD4,OCMP			0.4	
V_{IH}	Input High Voltage				
	PA0-PA7,PB0-PB7,PC0-PC7,PD0-PD5,PD7,ICAP,$\overline{IRQ}$,$\overline{RESET}$,OSC1	0.7xV_{DD}		V_{DD}	V
V_{IL}	Input Low Voltage				
	PA0-PA7,PB0-PB7,PC0-PC7,PD0-PD5,PD7,ICAP,$\overline{IRQ}$,$\overline{RESET}$,OSC1	V_{SS}		0.2xV_{DD}	V
V_{RM}	Data Retention Mode (0 to 70°C)	2			V
	Supply Current (See Notes)				
I_{DD}	Run (Fosc = 4.2 MHz)		3.5	7.0	mA
	(Fosc = 8 MHz)		6	12.0	mA
I_{DD}	Wait (Fosc = 4.2 MHz)		1.6	4.0	mA
	(Fosc = 8 MHz)		3	8.0	mA
	Stop				
I_{DD}	-40 to 85°C		1	10	µA
I_{DD}	-40 to 125°C		1	10	µA
I_{IL}	I/O Ports Hi-Z Leakage Current				µA
	PA0-PA7,PB0-PB7,PC0-PC7,PD1-PD4			± 10	
I_{IN}	Input Current : $\overline{RESET}$, $\overline{IRQ}$, ICAP, OSC1, PD0, PD5, PD7			± 1	µA
C_{OUT}	Capacitance : Ports (as Input or Output)			12	pF
C_{IN}	$\overline{RESET}$, $\overline{IRQ}$, ICAP, OSC1, PD0-PD5, PD7			8	pF

Notes:
1 All values show reflect average measurements
2 Typical values at midpiont of voltage range, 25°C only
3 Wait I_{DD} : only timer system active (SPE=TE=RE=0) if SPI, SCI active (SPE= TE= RE=1) add 10% current draw.
4 Run (Operating) I_{DD}, wait I_{DD}: measured using external square wave clock source (f $_{OSC}$ = 4.2 MHz) all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20 pF on OSCOUT.
5 Wait, stop I_{DD}: all ports configured as inputs, V_{IL} = 0.2V, V_{IH}= V_{DD}-0.2V
6 Stop I_{DD} : all ports measured with OSCIN = V_{SS}
7 Standard temperature range is 0° to 70°C. Extended temperature (-40° to 85°C, -40° to 125°C) version and a 25°C only version available.
8 Wait I_{DD} is affected linearly by OSCIN capacitance.
9 Typical curves of I_{DD} (supply current) versus f_{OP} (internal frequency) are given on figures 12.5., 12.6.a, and 12.6.b..

Table 12.5. DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 3.3 V_{dc} ± 10%, V_{SS} = 0 V_{dc}, T_A = T_L to T_H unless otherwise noted)

Symbol	Characteristics	Min.	Typ.	Max.	Unit
V_{OL}	Output Voltage, Load ≤ 10.0 µA			0.1	V
V_{OH}		V_{DD}-0.1			V
	Output High Voltage				
V_{OH}	(I_{LOAD} = 0.2 mA) PA0-PA7,PB0-PB7,PC0-PC7,OCMP (See Fig.9.2)	V_{DD}-0.3			V
V_{OH}	(I_{LOAD} = 0.4 mA)PD1-PD4 (See Fig. 9.3)	V_{DD}-0.3			V
V_{OL}	Output Low Voltage (See Fig. 9.4)				V
	(I_{LOAD} = 0.4 mA) PA0-PA7,PB0-PB7,PC0-PC7,PD1-PD4,OCMP			0.3	
V_{IH}	Input High Voltage				
	PA0-PA7,PB0-PB7,PC0-PC7,PD0-PD5,PD7,ICAP,$\overline{IRQ}$,$\overline{RESET}$,OSC1	0.7xV_{DD}		V_{DD}	V
V_{IL}	Input Low Voltage				
	PA0-PA7,PB0-PB7,PC0-PC7,PD0-PD5,PD7,ICAP,$\overline{IRQ}$,$\overline{RESET}$,OSC1	V_{SS}		0.2xV_{DD}	V
V_{RM}	Data Retention Mode (0 to 70°C)	2			V
	Supply Current (See Notes)				
I_{DD}	Run (Fosc = 2.1 MHz)		1.0	2.5	mA
I_{DD}	Wait (Fosc = 2.1 MHz)		0.5	1.4	mA
	Stop				
I_{DD}	-40 to 85°C		1	5	µA
I_{DD}	−40 to 125°C		1	5	µA
I_{IL}	I/O Ports Hi-Z Leakage Current				µA
	PA0-PA7,PB0-PB7,PC0-PC7,PD1-PD4			± 10	
I_{IN}	Input Current : $\overline{RESET}$, $\overline{IRQ}$, ICAP, OSC1, PD0, PD5, PD7			± 1	µA
C_{OUT}	Capacitance : Ports (as Input or Output)			12	pF
C_{IN}	$\overline{RESET}$, $\overline{IRQ}$, ICAP, OSC1, PD0-PD5, PD7			8	pF

Notes:
1 All values show reflect average measurements
2 Typical values at midpiont of voltage range, 25°C only
3 Wait I_{DD} : only timer system active (SPE=TE=RE=0) if SPI, SCI active (SPE= TE= RE=1) add 10% current draw.
4 Run (Operating) I_{DD}, wait I_{DD}: measured using external square wave clock source (f $_{OSC}$ = 2.1 MHz) all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20 pF on OSCOUT.
5 Wait, stop I_{DD}: all ports configured as inputs, V_{IL} = 0.2V, V_{IH}= V_{DD}-0.2V
6 Stop I_{DD} : all ports measured with OSCIN = V_{SS}
7 Standard temperature range is 0° to 70°C. Extended temperature (-40° to 85°C, -40° to 125°C) version.
8 Wait I_{DD} is affected linearly by OSCIN capacitance.
9 Typical curves of I_{DD} (supply current) versus f_{OP} (internal frequency) are given on figures 12.5., 12.6.a, and 12.6.b..

Figure 12.5 . Typical Current vs Internal Frequency for Run and Wait Modes

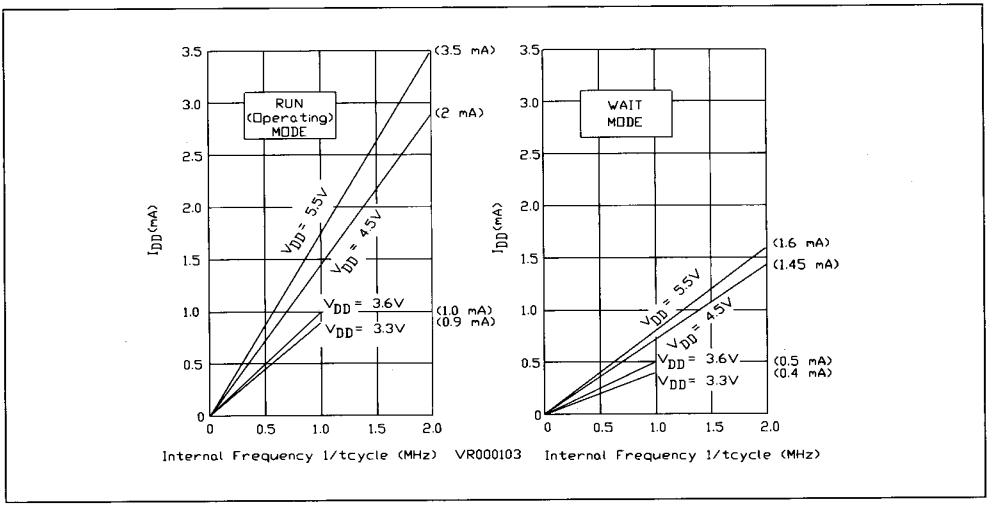


Figure 12.6a. Maximun IDD vs Frequency for VDD = 5.0 Vdc

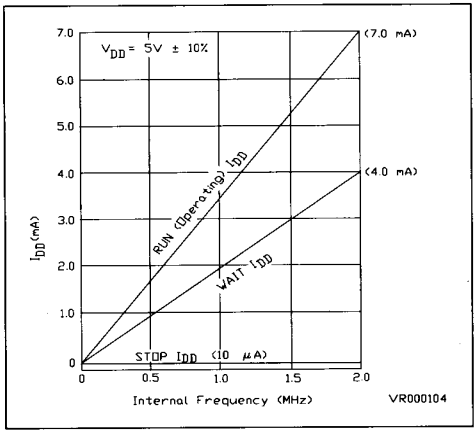


Figure 12.6b. Maximun IDD vs Frequency for VDD = 3.3 Vdc

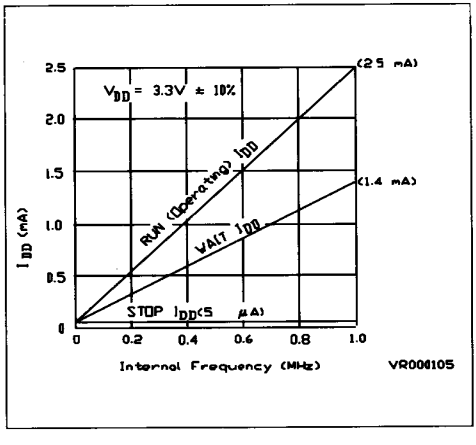


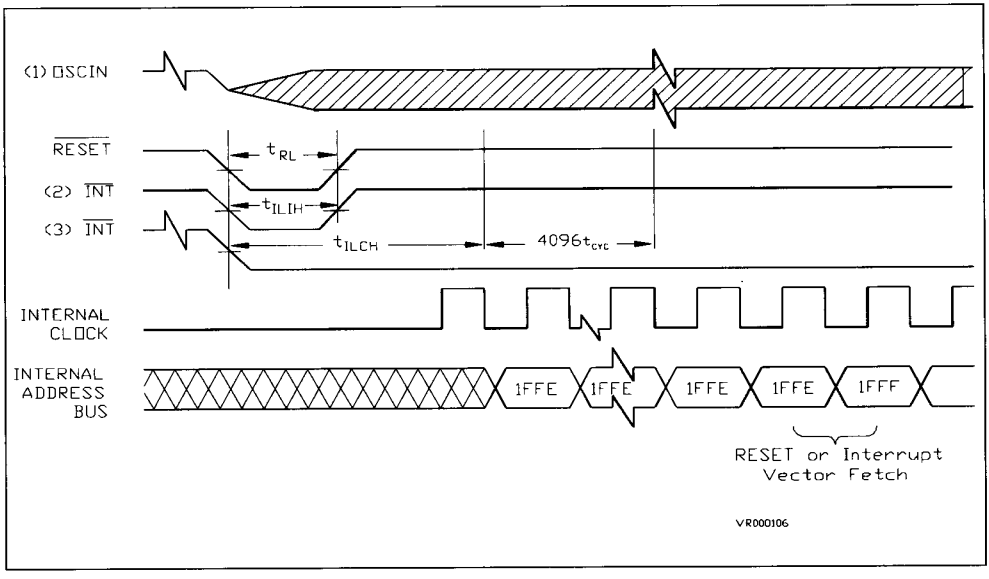
Table 12.6. CONTROL TIMING (Maximum Bus Speed = 4MHz)

(V_{DD} = 5.0 V_{dc} ± 10%, V_{SS} = 0 V_{dc}, T_A = T_L to T_H)

Symbol	Characteristics	Min.	Typ.	Max.	Unit
	Frequency of option				
f_{OSC}	Crystal Option			8	MHz
f_{OSC}	External Option	dc		8	MHz
	Internal Operating Frequency				
f_{OP}	Crystal (f_{OSC} ÷ 2)			4	MHz
f_{OP}	External Clock (fOSC ÷ 2)	dc		4	MHz
t_{CYC}	Clock Time	125			ns
t_{OXOV}	Crystal Oscillator Start-up Time (See Fig.5.1)			50	ms
t_{ILCH}	Stop Recovery Start-up Time (Crystal oscillator) (See Fig.12.2)			50	ms
t_{RL}	RESET Pulse Width (See Fig.5.1)	1.5			t_{CYC}
	Timer				
t_{RESL}	Resolution**	2/4/8			t_{CYC}
t_{TH},t_{TL}	Input Capture Pulse Width (See Fig.12.3)	125			ns
t_{TLTL}	Input Capture Pulse Period (See Fig.12.3)	***			t_{CYC}
t_{ILIH}	Interrupt Pulse Width Low (Edge-triggered) (See Fig.5.6)	125			ns
t_{ILIL}	Interrupt Pulse Period (See Fig.5.6)	*			t_{CYC}
t_{OH},t_{OL}	OSCIN Pulse Width	45			ns

Notes:
* The minimum preiod t_{ILIL} should not be less than the number of cycles times it takes to execute the interrupt service routine plus 21t_{CYC}.
** Depending of the timer input clock mask option (f_{OP}/2, f_{OP}/4, f_{OP}/8) the resolution can be 2 t $_{CYC}$, 4 t_{CYC}, 8 t_{CYC}.
*** The minimun period t_{TLTL} should not be less than the number of cycles times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.

Figure 12.2. Stop Recovery Timing Diagram



Notes:
1 Represents the internal gating of the OSCIN pin.
2 INT pin edgde-sensitive mask option.
3 INT pin level and edge sensitive mask option.

Figure 12.3. Timer Relationship

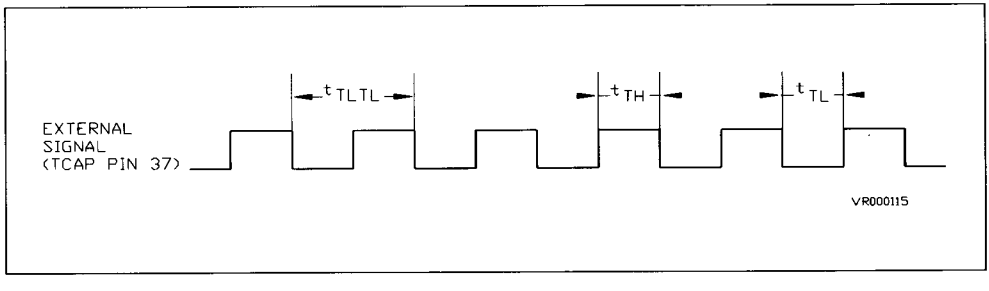
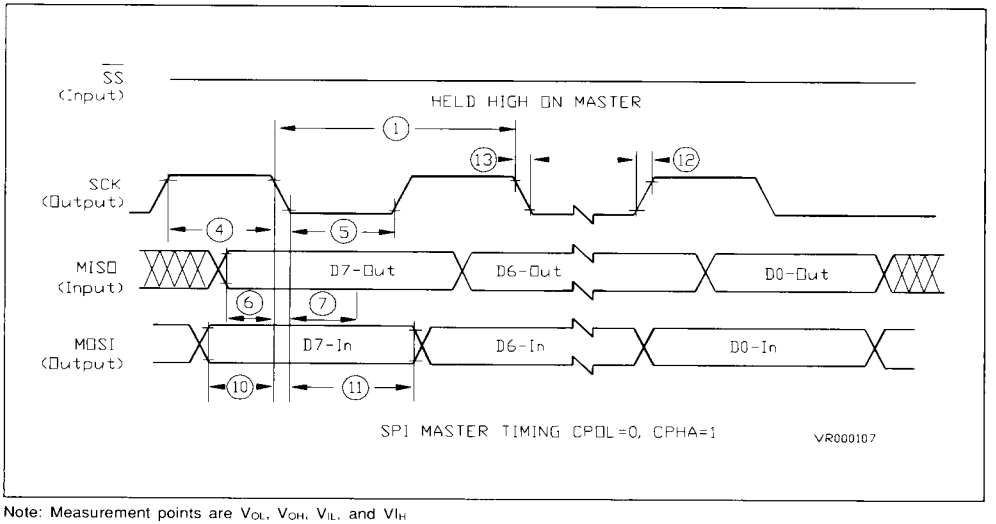
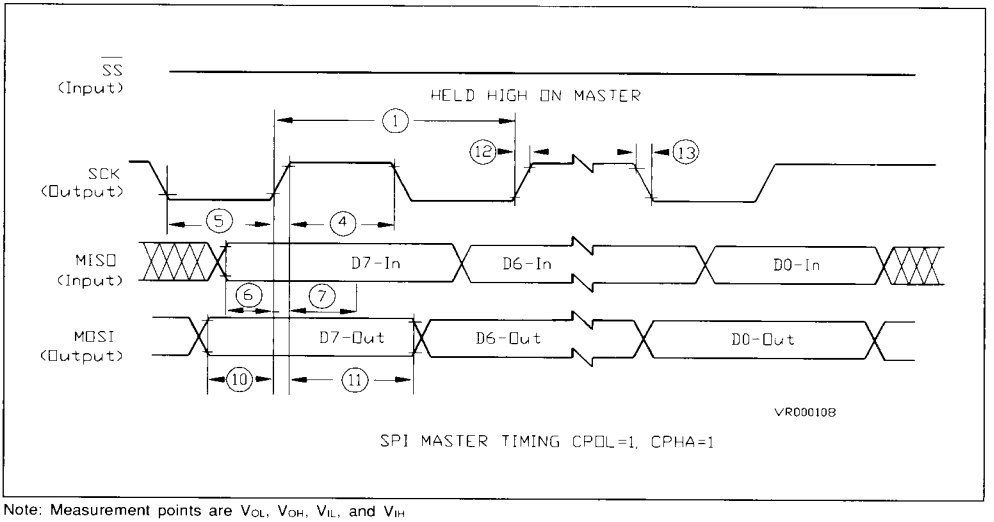


Figure 12.4a. SPI Master Timing Diagram CPOL=0, CPHA=1



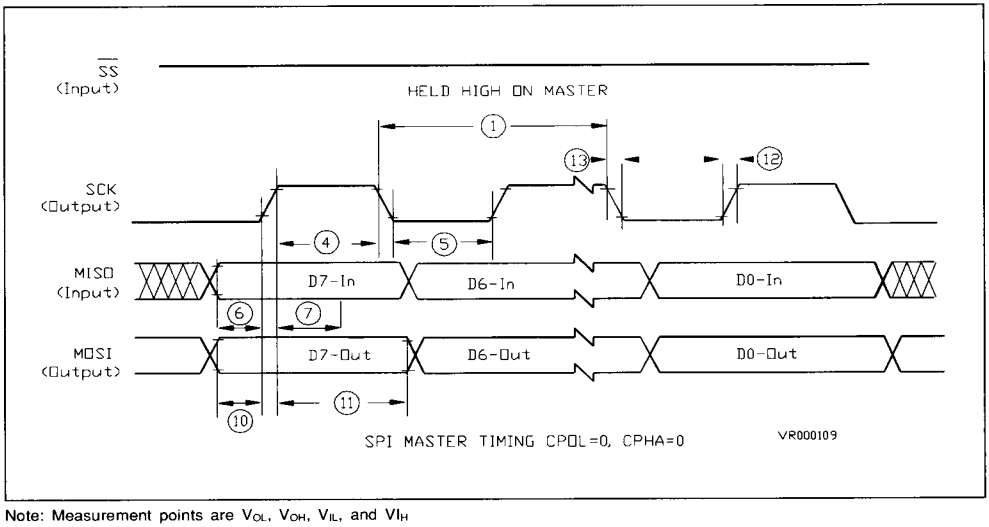
Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

Figure 12.4b. SPI Master Timing Diagram CPOL=1, CPHA=1



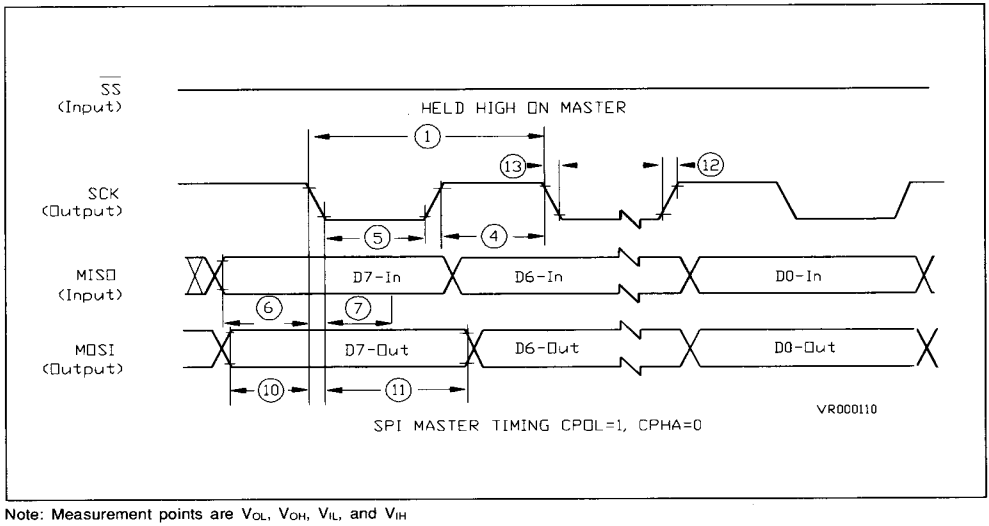
Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

Figure 12.4c. SPI Master Timing Diagram CPOL=0, CPHA=0



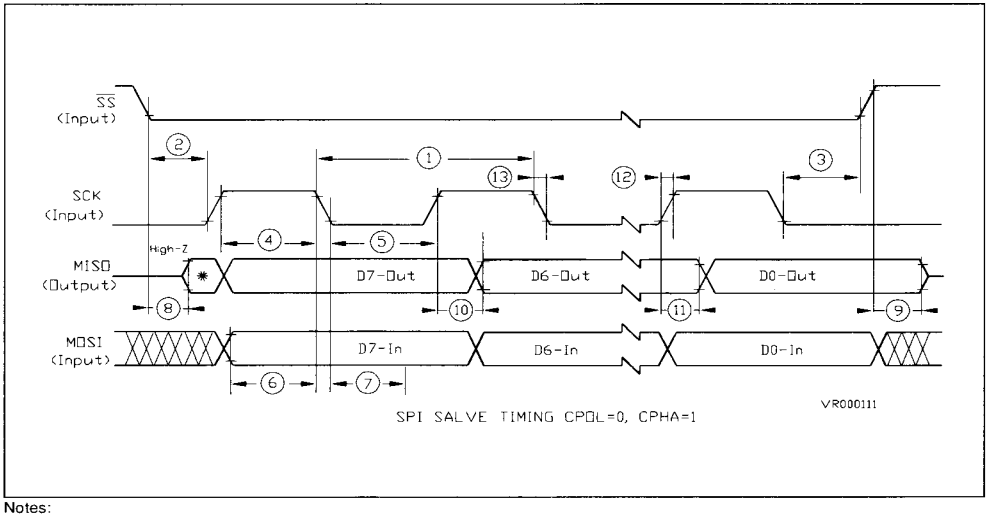
Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

Figure 12.4d. SPI Master Timing Diagram CPOL=1, CPHA=0



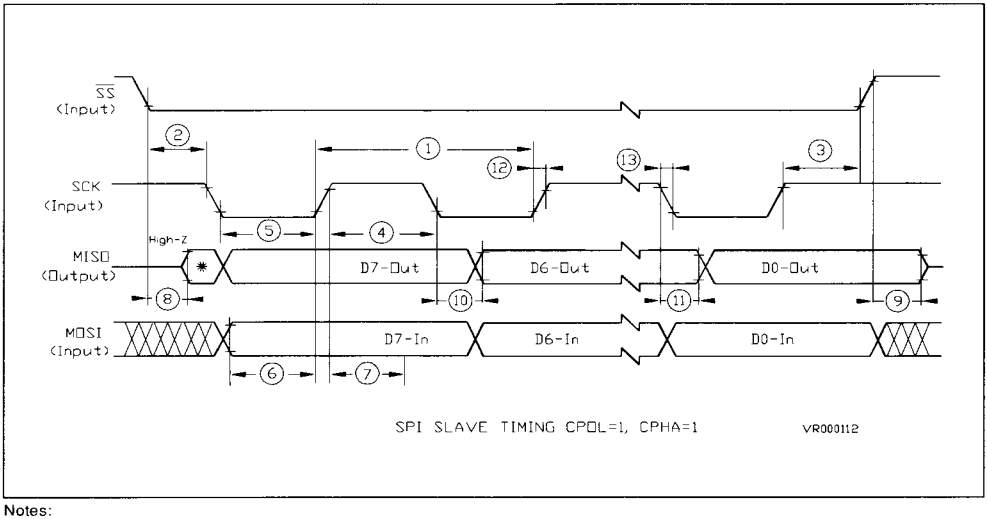
Note: Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}

Figure 12.4e. SPI Slave Timing Diagram CPOL=0, CPHA=1



Notes:
1 Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}.
2 * Denotes undefined, either high or low.

Figure 12.4f. SPI Slave Timing Diagram CPOL=1, CPHA=1



Notes:
1 Measurement points are V_{OL}, V_{OH}, V_{IL}, and V_{IH}.
2 * Denotes undefined, either high or low.

Figure 12.4g. SPI Slave Timing Diagram CPOL=0, CPHA=0

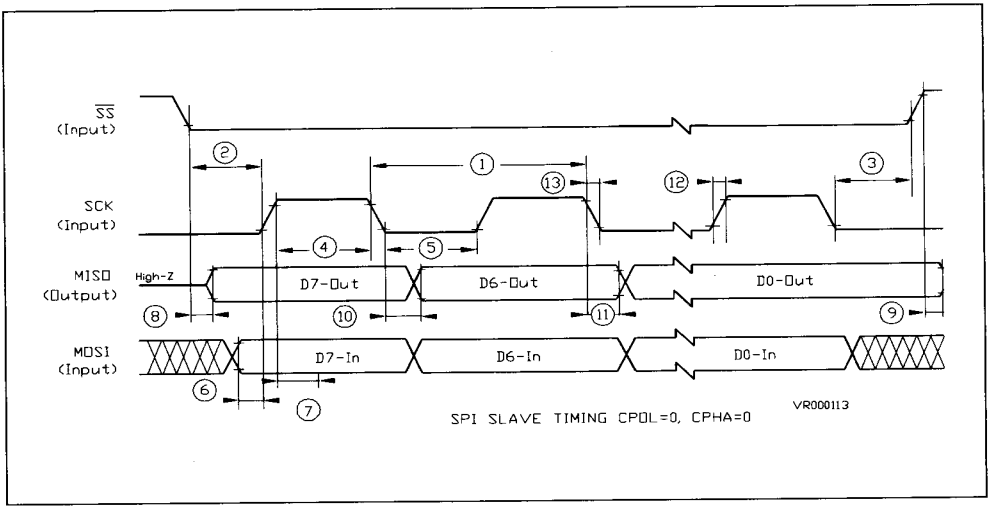


Figure 12.4h. SPI Slave Timing Diagram CPOL=1, CPHA=0

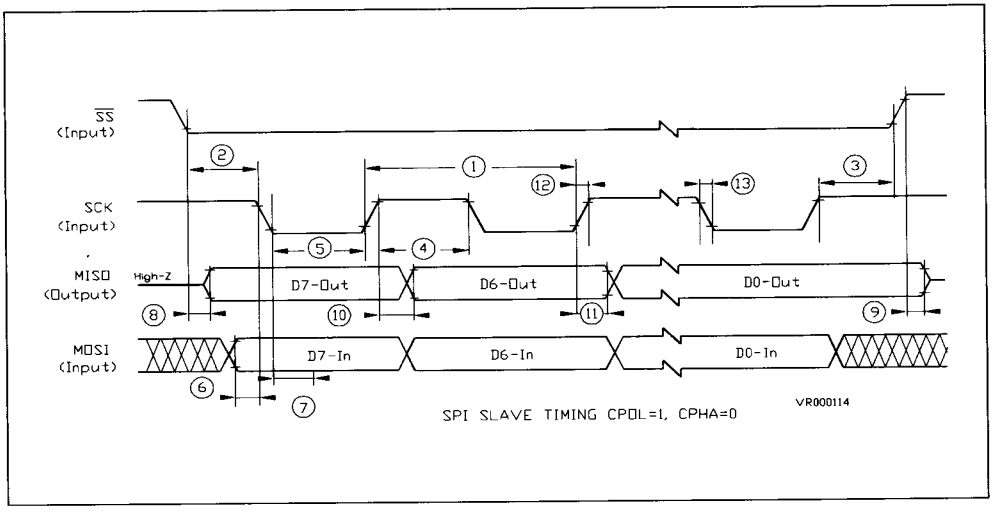


Table 12.7. CONTROL TIMING

(V_{DD} = 3.3 V_{dc} ± 10%, V_{SS} = 0 V_{dc}, T_A = T_L to T_H)

Symbol	Characteristics	Min.	Typ.	Max.	Unit
	Frequency of Option				
f_{OSC}	Crystal Option			2	MHz
f_{OSC}	External Option	dc		2	MHz
	Internal Operating Frequency				
f_{OP}	Crystal (f_{OSC} ÷ 2)			1	MHz
f_{OP}	External Clock (fOSC ÷ 2)	dc		1	MHz
t_{CYC}	Clock Time	1000			ns
t_{OXOV}	Crystal Oscillator Start-up Time (See Fig.5.1)			50	ms
t_{ILCH}	Stop Recovery Start-up Time (Crystal oscillator) (See Fig.12.2)			50	ms
t_{RL}	$\overline{\text{RESET}}$ Pulse Width (See Fig.5.1)	1.5			t_{CYC}
	Timer				
t_{RESL}	Resolution**	2/4.0/8			t_{CYC}
t_{TH},t_{TL}	Input Capture Pulse Width (See Fig.12.3)	250			ns
t_{TLTL}	Input Capture Pulse Period (See Fig.12.3)	***			t_{CYC}
t_{ILIH}	Interrupt Pulse Width Low (Edge-triggered) (See Fig.5.6)	250			ns
t_{ILIL}	Interrupt Pulse Period (See Fig.5.6)	*			t_{CYC}
t_{OH},t_{OL}	OSCIN Pulse Width	200			ns

Notes:
* The minimum preiod t_{ILIL} should not be less than the number of cycles times it takes to execute the interrupt service routine plus 21t_{CYC}.
** Depending of the timer input clock mask option (f_{OP} 12, f_{OP} 14, f_{OP}18) the resolution can be 2 t $_{CYC}$, 4 t_{CYC}, 8 t_{CYC}.
*** The minimun period t_{TLTL} should not be less than the number of cycles times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.

Table 12.8. SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 5.0 V_{dc} ± 10%, V_{SS} = 0 V_{dc}, T_A = T_L to T_H)

Num.	Symbol	Characteristics	Fosc = 4.2MHz		Fosc = 8.0MHz		Unit
			Min.	Max.	Min.	Max.	
		Operating Frequency = $F_{OSC/2}$ = F_{OP}					
	f_{OP}(m)	Master	dc	0.5	dc	0.5	f_{OP}
	f_{OP}(s)	Slave	dc	2.1	dc	4.0	MHz
1		Cycle Time					
	$t_{CYC(m)}$	Master	2.0		2.0		tCYC
	$t_{CYC(s)}$	Slave	480		240		ns
2		Enable Lead Time					ns
	$t_{lead(m)}$	Master	*		*		
	$t_{lead(s)}$	Slave	240		120		
3		Enable Lag time					ns
		Master	*		*		
		Slave	240		120		
4		Clock (SCK)High Time					
	$t_{W(SCKH)m}$	Master	340		100		ns
	$t_{W(SCKH)s}$	Slave	190		90		ns
5		Clock (SCK) Low Time					
	$t_{W(SCKL)m}$	Master	340		100		ns
	$t_{W(SCKL)s}$	Slave	190		90		ns
6		Data Set-up Time					
	$t_{SU(m)}$	Master	100		100		ns
	$t_{SU(s)}$	Slave	100		100		ns
7		Data Hold Time (Inputs)					
	$t_{H(m)}$	Master	100		100		ns
	$t_{H(s)}$	Slave	100		100		ns
8	t_A	Access Time (Time to Data Active from High Impedance State)					ns
		Slave	0	120	0	120	
9	t_{DIS}	Disable Time (Hold Time to High Impedance State)					ns
		Slave		240		240	
10		Data Valid					
	$t_{V(m)}$	Master (Before Capture Edge)	0.25		0.25		$t_{CYC(m)}$
	$t_{V(s)}$	Slave (After Enable Edge) **		240		120	ns
11		Data Hold Time (Outputs)					
	$t_{HO(m)}$	Master (Before Capture Edge)	0.25		0.25		$t_{CYC(m)}$
	$t_{HO(s)}$	Slave (After Enable Edge)	0		0		ns
12		Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200pF)					
	t_{RM}	SPI Outputs (SCK, MOSI, MISO)		100		100	ns
	t_{RS}	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)		2.0		2.0	µs
13		Fall Time (70% V_{DD} to 20% V_{DD}, C_L					
	t_{FM}	SPI Outputs (SCK, MOSI, MISO)		100		100	ns
	t_{FS}	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)		2.0		2.0	µs

Notes:

Table 12.9. SERIAL PERIPHERAL INTERFACE (SPI) TIMING

(V_{DD} = 3.3 V_{dc} ± 10%, V_{SS} = 0 V_{dc}, T_A = T_L to T_H)

Num.	Symbol	Characteristics	Min.	Max.	Unit
		Operating Frequency			
	$f_{OP(m)}$	Master	dc	0.5	f_{OP}
	$f_{OP(s)}$	Slave	dc	1.0	MHz
1		Cycle Time			
	$t_{CYC(m)}$	Master	2.0		tCYC
	$t_{CYC(s)}$	Slave	1.0		µs
2		Enable Lead Time			ns
	$t_{LEAD(m)}$	Master	*		
	$t_{LEAD(s)}$	Slave	500		
3		Enable Lag time			ns
		Master	*		
		Slave	500		
4		Clock (SCK)High Time			
	$t_{W(SCKH)m}$	Master	720		ns
	$t_{W(SCKH)s}$	Slave	400		ns
5		Clock (SCK) Low Time			
	$t_{W(SCKL)m}$	Master	720		ns
	$t_{W(SCKL)s}$	Slave	400		ns
6		Data Set-up Time			
	$t_{SU(m)}$	Master	200		ns
	$t_{SU(s)}$	Slave	200		ns
7		Data Hold Time (Inputs)			
	$t_{H(m)}$	Master	200		ns
	$t_{H(s)}$	Slave	200		ns
8	t_A	Access Time (Time to Data Active from High Impedance State)			ns
		Slave	0	250	
9	t_{DIS}	Disable Time (Hold Time to High Impedance State)			ns
		Slave		500	
10		Data Valid			
	$t_{V(m)}$	Master (Before Capture Edge)	0.25		$t_{CYC(m)}$
	$t_{V(s)}$	Slave (After Enable Edge) **		500	ns
11		Data Hold Time (Outputs)			
	$t_{HO(m)}$	Master (Before Capture Edge)	0.25		$t_{CYC(m)}$
	$t_{HO(s)}$	Slave (After Enable Edge)	0		ns
12		Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200pF)			
	t_{RM}	SPI Outputs (SCK, MOSI, MISO)		200	ns
	t_{RS}	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)		2.0	µs
13		Fall Time (70% V_{DD} to 20% V_{DD}, C_L			
	t_{FM}	SPI Outputs (SCK, MOSI, MISO)		200	ns
	t_{FS}	SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)		2.0	µs

Notes:
1 * Signal production depends on software.
2 **Assumes 200pF load on all SPI pins.
3 See figures 12.4. for SPI timing diagram.

Table 12.10. DC ELECTRICAL CHARACTERISTICS FOR LOW VOLTAGE OPERATION

(V_{DD} = 2.4 V_{dc} – 3.6 V_{dc} , V_{SS} = 0 V_{dc}, T_A = T_L to T_H unless otherwise noted)

Symbol	Characteristics	Min.	Typ.	Max.	Unit
V_{OL}	Output Voltage, Load ≤ 10.0 µA			0.1	V
V_{OH}		V_{DD}-0.1			V
	Output High Voltage				
V_{OH}	(I LOAD = 0.2 mA) PA0-PA7,PB0-PB7,PC0-PC7,OCMP (See Fig.9.2)				V
V_{OH}	(I LOAD = 0.4 mA)PD1-PD4 (See Fig. 9.3)	V_{DD}-0.3			V
V_{OL}	Output Low Voltage (See Fig. 9.4)	VDD-0.3			V
	(I LOAD = 0.4 mA) PA0-PA7,PB0-PB7,PC0-PC7,PD1-PD4,OCMP			0.3	
V_{IH}	Input High Voltage				
	PA0-PA7,PB0-PB7,PC0-PC7,PD0-PD5,PD7,ICAP,$\overline{IRQ}$,$\overline{RESET}$,OSC1	0.7xV_{DD}		V_{DD}	V
V_{IL}	Input Low Voltage				
	PA0-PA7,PB0-PB7,PC0-PC7,PD0-PD5,PD7,ICAP,$\overline{IRQ}$,$\overline{RESET}$,OSC1	V_{SS}		0.2xV_{DD}	V
V_{RM}	Data Retention Mode (0 to 70°C)	20			V
	Supply Current (2.4Vdc at 500kHz)				
I_{DD}	Run (See Fig.12.5)			750	µA
I_{DD}	Wait (See Fig.12.5)			400	µA
	Stop (See Fig.12.5)				
I_{DD}	0 to 70°C		1	5.0	µA
I_{IL}	I/O Ports Hi-Z Leakage Current				µA
	PA0-PA7,PB0-PB7,PC0-PC7,PD1-PD4			± 10	
I_{IN}	Input Current : $\overline{RESET}$, $\overline{IRQ}$, ICAP, OSC1, PD0, PD5, PD7			± 1	µA
C_{OUT}	Capacitance : Ports (as Input or Output)			12	pF
C_{IN}	$\overline{RESET}$, $\overline{IRQ}$, ICAP, OSC1, PD0-PD5, PD7			8	pF

Notes:
1 All values show reflect average measurements
2 Typical values at midpiont of voltage range, 25°C only
3 Wait I_{DD} : only timer system active (SPE=TE=RE=0) if SPI, SCI active (SPE= TE= RE=1) add 10% current draw.
4 Run (Operating) I_{DD}, wait I_{DD}: measured using external square wave clock source (f_{OSC} = 1.0 MHz) all inputs 0.2V from rail, no DC loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5 Wait, stop I_{DD}: all ports configured as inputs, V_{IL} = 0.2V, V_{IH}= V_{DD}-0.2V
6 Stop I_{DD} : all ports measured with OSC1 = V_{SS}
7 Standard temperature range is 0° to 70°C. Extended temperature (-40° to 85°C) range is available.
8 Wait I_{DD} is affected linearly by OSC2 capacitance.

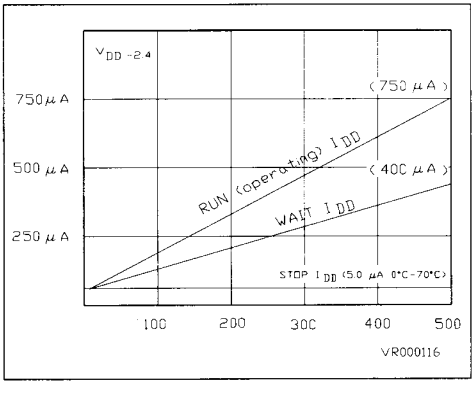
Table 12.12. CONTROL TIMING FOR LOW VOLTAGE OPERATION

(V_{DD} = 2.4 V_{dc} – 3.6 V_{dc} , V_{SS} = 0 V_{dc}, T_A = T_L to T_H)

Symbol	Characteristics	Min.	Typ.	Max.	Unit
f_{OSC}	Frequency of option				MHz
	Crystal Option			1.0	
	External Option	dc		1.0	
f_{OP}	Internal Operating Frequency				MHz
	Crystal (f_{OSC} ÷ 2)			0.5	
	External Clock (f_{OSC} ÷ 2)	dc		0.5	
t_{CYC}	Clock Time	2000			ns
t_{OXOV}	Crystal Oscillator Start-up Time (See Fig.5.1)			50	ms
t_{ILCH}	Stop Recovery Start-up Time (Crystal oscillator) (See Fig.12.2)			50	ms
t_{RL}	$\overline{\text{RESET}}$ Pulse Width (See Fig.5.1)	1.5			t_{CYC}
	Timer				
t_{RESL}	Resolution**	2/4.8			t_{CYC}
t_{TH},t_{TL}	Input Capture Pulse Width (See Fig.12.3)	250			ns
t_{TLTL}	Input Capture Pulse Period (See Fig.12.3)	***			t_{CYC}
t_{ILIH}	Interrupt Pulse Width Low (Edge-triggered) (See Fig.5.6)	500			ns
t_{ILIL}	Interrupt Pulse Period (See Fig.5.6)	*			t_{CYC}
t_{OH},t_{OL}	OSCIN Pulse Width	400			ns

Notes:
* The minimum period t_{ILIL} should not be less than the number of cycles times it takes to execute the interrupt service routine plus 21t_{CYC}.
** Depending of the timer input clock mask option (f_{OP}/2, f_{OP}/4, f_{OP}/8) the resolution can be 2 t $_{CYC}$, 4 t_{CYC}, 8 t_{CYC}.
*** The minimun period t_{TLTL} should not be less than the number of cycles times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.

Figure 12.5. Maximum I_{DD} vs Frequency for V_{DD} = 2.4 V_{dc}



ST8108 ORDERING FORM

User ROM Content

The hexadecimal object file is recommended to be sent in an hexadecimal file (**assembler output**), stored on a PC/DOS, 360K Double Sided 5 1/4 floppy disk.

If the floppy media can not be provided by the customer, the ROM file can be also sent in a 27xx EPROM memory, where all unused bytes must be left erased to "FF".

Mask Option List & application information

Oscillator :
- ❑ Crystal/Resonator
- ❑ Resistor

Timer internal clock source :
- ❑ Standard (Fop/4)
- ❑ Slow (Fop/8)
- ❑ Fast (Fop/2)

SPI :
- ❑ used
- ❑ not used

SPI internal clock source :
- ❑ Standard (Fop)
- ❑ Slow (Fop/2)

Halt Mode :
- ❑ Used
- ❑ Not Used

Voltage range :
- ❑ standard 4.5V/5.5V
- ❑ low 2.7V/3.3V
- ❑ extended 3V/6V
- ❑ other .....V/.....V

Interrupt trigger :
- ❑ Edge only sensitive
- ❑ Level & Edge sensitive

Frequency of operation (Fop) :

Oscillator frequency = Fosc = ...MHz

Internal operation Fop = Fosc/2 = ...MHz

SCI :
- ❑ used
- ❑ not used

SCI internal clock used :
- ❑ Standard (Fop)
- ❑ Slow (Fop/2)

Wait Mode :
- ❑ Used
- ❑ Not Used

Temperature :
- ❑ standard 0/70 C
- ❑ extended -40/85 C
- ❑ other ...../..... C

Packaging
- ❑ DIL plastic 40
- ❑ PLCC 44
- ❑ SDIP 42
- ❑ QFP 44
- ❑ Wafer on membrane
- ❑ Sawed dice (waffle box)

Marking

ST LOGO, Date Code, Assy code
ST8108XX/YY (ST part number)
. . . . . .
(customer id, 11 characters).

COMPANY/NAME : . . . . . SIGNATURE : . . . . . .

ORDERING INFORMATION

PART NUMBER	PACKAGE	TEMPERATURE
ST8108-B1	PLASTIC DIL 40	0°C/70°C
ST8108-B6	PLASTIC DIL 40	-40°C/+85°C
ST8108-C1	PLCC 44	0°C/70°C
ST8108-C6	PLCC 44	-40°C/+85°C
ST8108-S1	SDIP 42	0°C/70°C
ST8108-S6	SDIP 42	-40°C/+85°C
ST8108-Q1	QFP 44	0°C/70°C
ST8108-Q6	QFP 44	-40°C/+85°C

* NOTE : Each ROM content is identified by two alphabetic characters xx to be added to the sales type (i.e. ST8108 B1/xx).

SGS-THOMSON Microelectronics GROUP OF COMPANIES
Australia - Brazil - China - France - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands - Singapore - Spain - Sweden - Switzerland - Taiwan - United Kingdom - U.S.A. - West Germany

Printed by NTI - AGL Partenaires - Belcodène - France

SGS-THOMSON MICROELECTRONICS