Embedded Network Microcontroller with 10/100 MAC and 10Mbs PHY

The CS6208 Embedded Network Microcontroller is specifically targeted at applications that require network access. It incorporates a high-speed 4-cycle/ instruction 8051 compatible microcontroller core along with 65K bytes of on-chip program ROM memory and 32K bytes of on-chip data SRAM. External system memory is easily added to the CS6208, with up to 131K bytes of code memory and 98K bytes data memory facilitating large programs.

An on-chip crystal oscillator with programmable PLL provides a simple yet robust clocking scheme.

Network access is facilitated by moving packets through the transmit and receive buffers of the Media Access Control (MAC) block. There is a 32K byte packet buffer local to the MAC, shared with on-chip data memory. The on-chip Ethernet Transceiver allows direct connection to a twisted-pair network environment. Other physical network media such as Home PNA can be accessed by the 7-wire ENDEC interface using external transceivers.

The CS6208 also includes the standard port 1 and port 3 of the 8051. A full-duplex serial I/O port, 8 interrupt sources with three level priority, and 3 16-bit timers are shared with these ports. In addition two 8-bit wide programmable I/O ports are included for external control applications. Two programmable master/slave I^2C interfaces are included for communication with other hosts or slave devices such as EEPROM.

A four channel 7-Bit analog-to-digital converter hold allows the CS6208 to interface directly with analog circuits. Analog inputs can be converted singly or in a round-robbin fashion using the A/D input multiplexor.

All functional blocks of the CS6208 can be individually powered up/down to provide optimal power management using the power config register. An on-chip COP timer ensures reliable program operation and the power-on/off detection circuit keeps memory contents from being corrupted. The CS6208 operates under a wide voltage range from 3.0V to 5.0V, and in low power mode the CPU core can operate on only 1.8V.

CPU

- 4 cycle 80C51 instruction set compatible core.
- Programmable clock rate: DC/20MHz/40/60MHz.
- Single crystal operation.
- On-chip programmable PLL generates up to 60MHz processor clock.
- Two data pointers with auto-increment/decrement.
- Programmers model is 8051 compatible.

Memory Architecture

- 65KB on-chip program memory: ROM
- 32KB on-chip SRAM usable as data or packet buffer memory.
- Non-multiplexed external memory interface.
- External memory expansion up to 131K bytes code and 98K bytes data.
- In-system programming using external FLASH.
- Default memory map is 80C51 compatible.

Networking and I/O

- 10/100 Ethernet MAC with built-in Physical Layer, single chip networking.
- IEEE 802.3 7-wire ENDEC interface.
- 32KB on-chip TX/RX packet buffer memory shared with CPU.
- Half-duplex MAC operation.
- Hardware checksum capability speeds network protocol processing.
- Full-duplex serial port.
- Four 8-bit digital I/O ports, bi-directional.

Firmware

- On chip HTTPD server.
- TCP/IP network stack accessible by application programs.
- ARP, IP, ICMP, UDP, TCP, DHCP, and BOOTP protocols implemented.

Integrated System Resources

- High speed 7-bit analog-to-digital converter
- Programmable A/D clock rate.
- Two I^2C interfaces both programmable for master or slave operation.
- Three 16-bit on-chip timers.
- Eight interrupt sources with three priority levels.
- Full-Duplex UART.

Power Management

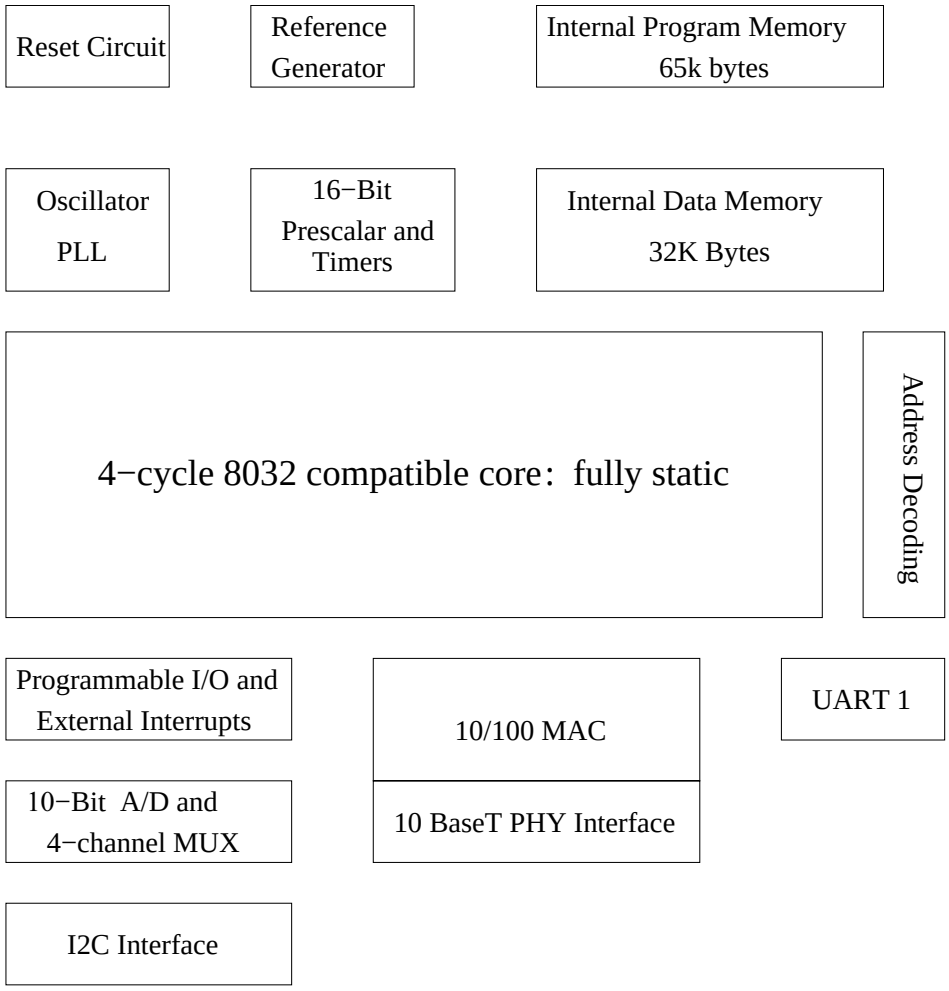
Embedded Network Microcontroller

GENERAL DESCRIPTION

FEATURES

- On-chip COP timer.
- Programmable power-fail detection/reset with interrupt.
- 3.3V and 5V I/O operation.
- 1.8V CPU core operation.
- Power on/off chip resources using power CONFIG register.

BLOCK DIAGRAM



PIN CONNECTION DIAGRAM

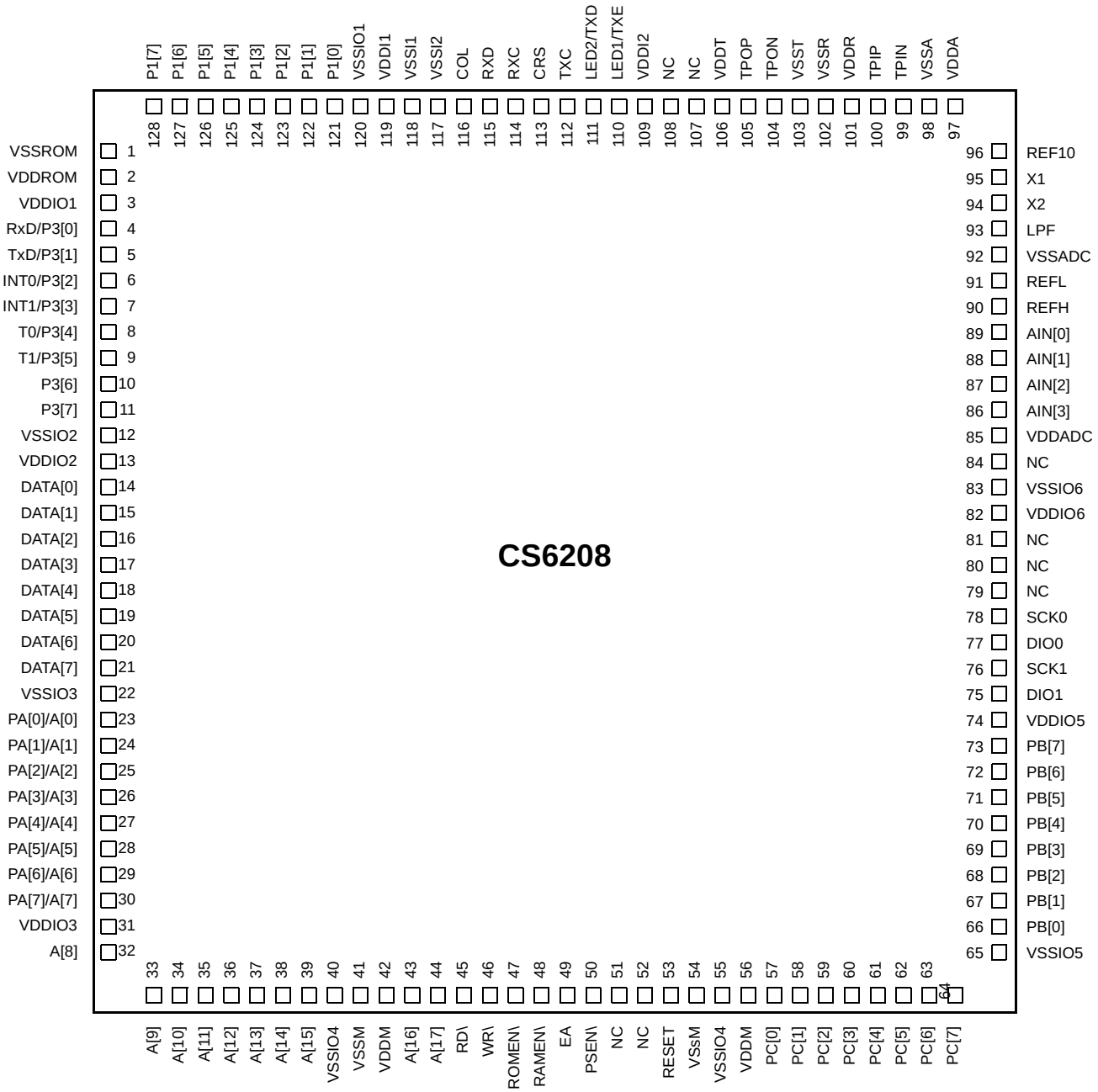


Figure-1 128-pin PQFP/LQFP

PIN DESCRIPTIONPort functions

Pin Name	I/O	Pin #	Description
// 8051 Ports			
Port1[7:0]	I/O	128-121	8051 I/O Port 1.
Port3[7:0]	I/O	11-4	8051 I/O Port 3. Multiplexed with: RD\, WR\, T1, T0, INT1, INT0, TxD, RxD. See port I/O section for further details.
PortA[7:0]	I/O	30-23	Programmable I/O Port A. Multiplexed with A[7:0]. Useable when the internal ROM is enabled.
PortB[7:0]	I/O	73-66	Programmable I/O Port B.
PortC[7:0]	I/O	64-57	Programmable I/O Port C.
A[17:0]	O	44-43/39-32/ 30-23	Address Bus Pins: (A17+A16+A[15:8]+A[7:0]). PortA[7:0] is multiplexed with A[7:0] and useable when the internal ROM is enabled.
Data[7:0]	I/O	21-14	Data Bus Pins.
RD\	O	45	External data memory read strobe. Active low. Connect to the output enable pin of external data memory.
WR\	O	46	External data memory write strobe. Active low. Connect to the write enable pin of external data memory.
ROMEN\	O	47	ROM Enable; Low active. Connect to the chip select pin of external program memory.
RAMEN\	O	48	RAM Enable; Low active. Connect to the chip select pin of external data memory.
EA	I	49	Memory map mode select.
PSEN\	O	50	Program Memory Output Enable; Low latched. Connect to the output enable pin of external program memory.
VDDIO1	P	3	Vdd for IO ports.
VDDIO2	P	13	Vdd for IO ports.
VDDIO3	P	31	Vdd for IO ports.
VDDIO4	P	55	Vdd for IO ports.
VDDIO5	P	74	Vdd for IO ports.
VDDIO6	P	82	Vdd for IO ports.
VSSIO1	P	120	Vss for IO ports.
VSSIO2	P	12	Vss for IO ports.
VSSIO3	P	22	Vss for IO ports.
VSSIO4	P	54	Vss for IO ports.
VSSIO5	P	65	Vss for IO ports.
VSSIO6	P	83	Vss for IO ports.
VDDI1	P	119	Vdd for 8051 core and I^2C digital blocks.

Pin Name	I/O	Pin #	Description
VSSI1	P	118	Vss for 8051 core and I^2C digital blocks.
// I^2C Interface			
SCK0	I/O	78	Master/Slave I^2C Interface Clock.
DIO0	I/O	77	Master/Slave I^2C Data I/O.
SCK1	I	76	Slave I^2C Interface Clock.
DIO1	I/O	75	Slave I^2C Data I/O.
// REF, Xtal, PLL, ROM			
RST	I	53	Reset Input.
X1	I	95	Crystal Input. Reference Application Note AN60.
X2	O	94	Crystal Output. Reference Application Note AN60.
LPF	O	93	PLL low pass filter pin. Reference Application Note AN60.
VDDA	P	97	Vdd for REF, Xtal, PLL blocks.
VSSA	P	98	Vss for REF, Xtal, PLL blocks.
VDDROM	P	2	Vdd for ROM block.
VSSROM	P	1	Vss for ROM block.
// T10PMD			
TPOP	O	105	Twist-pair output signal.
TPON	O	104	Twist-pair output signal.
REF10	O	96	Add an external 56K ohm pull-down resistor for waveform amplitude fine tuning.
VDDT	P	106	Vdd for T10PMD output driver.
VSST	P	103	Vss for T10PMD output driver.
TPIP	I	100	Twist-pair input signal.
TPIN	I	99	Twist-pair input signal.
VDDR	P	101	Vdd for T10PMD analog blocks.
VSSR	P	102	Vss for T10PMD analog blocks.
// 7-wire signals			
// Default: Int_Phy=1, used internal T10Pmd, All 7-wire pins are output.			
// If clear Int_Phy=0, NIC connect with 7-wire pins, and TXE/TXD are output,			

Pin Name	I/O	Pin #	Description
// TXC/CRS/COL/RXC/RXD are input.			
LED1/TXE	O/O	110	Transmit Enable.
LED2/TXD	O/O	111	TRansmit Data.
TXC	O/I	112	Transmit CLock.
CRS	O/I	113	Carrier Sense.
RXC	O/I	114	Receive Clock.
RXD	O/I	115	Receive Data.
COL	O/I	116	Collision Indicator.
VDDI2	P	109	Vdd for 7-wire IO pads and T10PMD's Digital blocks.
VSSI2	P	117	Vss for 7-wire IO pads and T10PMD's Digital blocks.
// SRAM power pins			
VDDM	P	42/56	Vdd for SRAM.
VSSM	P	41/54	Vss for SRAM.
// ADC			
AIN[3:0]	I	86-89	Analog Input signals.
REFH	I	90	High level voltage reference. Connect to 2.0V source.
REFL	I	91	Low level voltage reference. Connect to 1.2V source.
VDDADC	P	85	Vdd for ADC.
VSSADC	P	92	Vss for ADC.

Pins 51, 52, 79, 80, 81, 84, 107, and 108 are No Connects (NC) and must be left floating.

Port Functions:

P0: Standard 8051 data bus expansion, always dedicated to DATA[7:0] on the CS6208. Replaced by port C.

P1 and P3: Standard 8051 I/O ports.

P2: Standard 8051 high address output, always dedicated to ADDR[15:8] on the CS6208. Replaced by port B.

P5: Standard 8051 low address output, dedicated to ADDR[7:0] when using expanded memory mode. Can be used as an 8bit programmable I/O port A when in expanded memory mode (pin EA = 0).

Technical Specifications

The CS6208 is an 8051 compatible device that gives the user the features of the original 8051, but with improved speed, power consumption characteristics, large on-chip xdata SRAM, and integrated Ethernet networking. It has the same instruction set as the 8051 family; with one addition, DEC DPTR (op-code A5). While the original 8051 family was designed to operate at 12 clock periods per machine cycle, the CS6208 operates at a much reduced clock rate of only 4 clock periods per machine cycle.

This naturally speeds up the execution of the instructions. Consequently, the CS6208 can run at a higher speed as compared to the original 8051, even if the same crystal is used. Since the CS6208 is a fully static CMOS design, it can also be operated at a lower crystal clock, giving the same throughput in terms of instruction execution, yet reducing the power consumption.

The 4 clocks per machine cycle feature in the CS6208 is responsible for a three-fold increase in execution speed. The CS6208 has all the standard features of the 8051 and has many extra peripherals and features as well.

Integrated Ethernet MAC:

The CS6208 includes an integrated 10/100Mbs ethernet MAC unit with 7-wire output allowing the system designer freedom to choose the specific physical layer best suited for their application.

Integrated Ethernet Physical Layer:

An on-chip 10Mbs ethernet PHY reduces the system package count for ethernet only applications. A few external passive components are al that is needed to complete a working ethernet interface.

Integrated Hardware Checksum:

A hardware checksum register is provided to speed processing of network protocols.

Integrated PLL Clock Generator:

An on-chip programmable phase locked loop generates clocks for all hardware blocks of the CS6208.

32k-byte On-chip XDATA SRAM:

The CS6208 includes 32k-bytes of on-chip SRAM usable as XDATA memory space. In addition this memory block in tightly integrated with the on-chip Ethernet MAC, providing buffer memory for incoming/outgoing Ethernet packets. A small part of this memory is reserved for external on-chip SFR registers.

Dual MASTER/SLAVE I^2C Interfaces:

Accessing external peripherals such as serial eeprom or A/D converters is greatly facilitated by using the on-chip I^2C interfaces in either MASTER or SLAVE mode.

Four I/O Ports:

The CS6208 has four 8-bit ports which total up to 32 lines. Port 0 is dedicated for use as a Address/Data bus and

cannot be used as an I/O port. It has strong pull-ups and pull-downs, and does not need any external pull-ups. Port 2 is always used as the upper 8-bits of the Address bus. It has strong pull-ups and pull-downs. Ports 1, 3, B, and C act as I/O ports. When in expanded memory mode (pin EA = 1) the ADDR[7:0] pins become I/O Port A.

Serial I/O:

The CS6208 has one serial port that is functionally similar to the serial port of the original 8051 family. However the serial port on the CS6208 can operate in different modes in order to obtain timing similarity as well. The Serial port has the enhanced features of Automatic Address recognition and Frame Error detection.

Timers/Counters:

The CS6208 has three 16-bit timers that are functionally and similar to the timers of the 8051 family. When used as timers, they can be set to run at either 4 clocks or 12 clocks per count, thus providing the user the option of operating in a mode that emulates the timing of the original 8051.

The CS6208 has an additional feature, Watch Dog timer. This timer is used as a System Monitor or as a very long time period timer.

Interrupts:

The Interrupt structure in the CS6208 is slightly different from that of the standard 8051. Due to the presence of additional features and peripherals, the number of interrupt sources and vectors has been increased. There are provisions to set the interrupt priority level between integrated peripherals. The INSTAR ESFR (0xFFEB) is applicable for the following integrated peripherals: RAM Checksum, I^2C2, I^2C1, and Ethernet Mac. Use the INSTAR ESFR to determine which integrated peripheral(s) may use the interrupt system. These share hardware interrupts with standard 8051 functions as shown in the following table.

Interrupt Vector:

RESET	0000
INT0	0003 (External Interrupt 0, priority 1.)
T0_INT	000B (Timer 0 Overflow, priority 2.)
INT1	0013 (External Interrupt 1, priority 3.)
T1_INT	001B (Timer 1 Overflow, priority 4.)
SCON	0023 (Serial port, priority 5.)
T2_INT	002B (Timer 2 Overflow, priority 6.)
T2EX_INT	002B (Negative transition on T2EX.)
PF_INT	0033 (Power fail interrupt, priority 0.)
WG_INT	0063 (Watchdog timer interrupt, priority 8.)
CHKSUM_INT	Shared with INT0. Priority selectable.
I2C1_INT	Shared with INT0. Priority selectable.
I2C2_INT	Shared with INT0. Priority selectable.
MAC_INT	Shared with INT1. Always highest priority.

See the Interrupt Pseudocode in Appendix A at the end of this document.

Data Pointers:

The original 8051 had only one 16-bit Data Pointer (DPL, DPH). In the CS6208, there is an additional 16-bit Data Pointer (DPL1, DPH1). This new Data Pointer uses two unused SFR locations in the original 8051. In addition there is an added instruction DEC DPTR (op-code A5), which helps in improving the programming flexibility of the user.

Power Management:

Like the standard 80C51, the CS6208 also has the IDLE and POWERDOWN modes of operation. In the IDLE mode, the clock to the CPU is stopped while the timers, serial port and interrupt block continue to operate. In the POWERDOWN mode, all the clock are stopped and the chip operation is completely stopped. This is the lowest power consumption state. In addition, the CS6208 has a power-fail interrupt, which can indicate an imminent power failure, giving the user time to save whatever critical data that may be needed.

Power-On reset:

The CS6208 has an on-chip Power-On Reset facility. This does away with the need for an external capacitor-resistor network that was needed in the standard 8051.

CORE CPU ARCHITECTURE:

The CS6208 is based on the standard 8051 device. The CS6208 is built around a 8-bit ALU that uses internal registers for temporary storage and control of the peripheral devices. It can execute the standard 8051 instruction set. A brief description of the internal blocks follows:

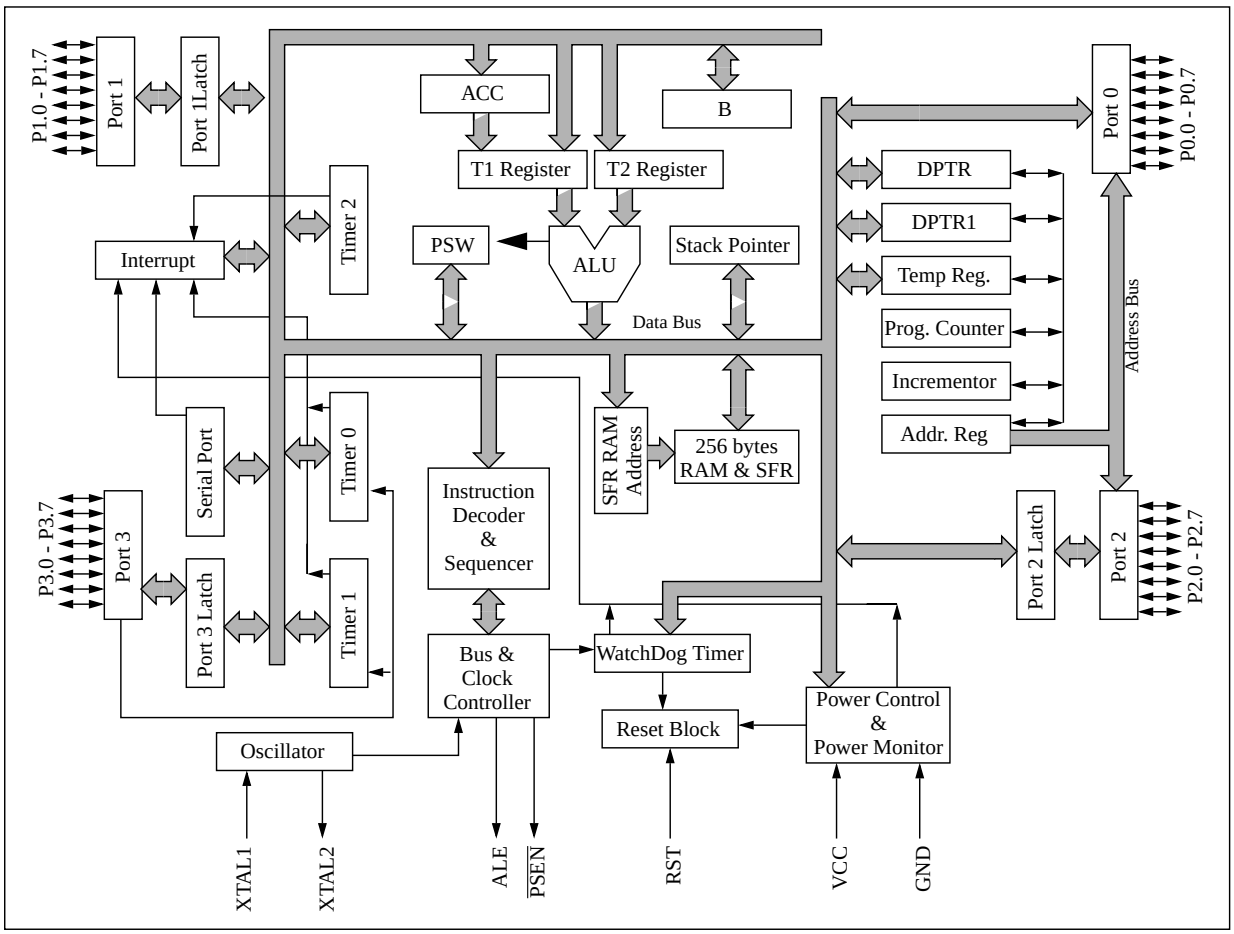


Figure-2 Architecture of the cs6208 processor core.

ALU:

The ALU is the heart of the CS6208. It is responsible for the arithmetic and logical functions. It is also used in decision making, in case of jump instructions, and is also used in calculating jump addresses. The user cannot directly use the ALU, but the Instruction Decoder reads the op-code, decodes it, and sequences the data through the ALU and its associated registers to generate the required result. The ALU mainly uses the ACC which is a special function register (SFR) on the chip. Another SFR, namely B register is also used in Multiply and Divide instructions. The ALU generates several status signals which are stored in the Program Status Word register (PSW).

Accumulator:

The Accumulator (ACC) is the primary register used in arithmetic, logical and data transfer operations in the CS6208. Since the Accumulator is directly accessible by the CPU, most of the high-speed instructions make use of the ACC as one argument.

B Register:

This is an 8-bit register that is used as the second argument in the MUL and DIV instructions. For all other instructions it can be used simply as a general purpose register.

Program Status Word:

This is an 8-bit SFR, that is used to store the status bits of the ALU. It holds the Carry flag, the Auxiliary Carry flag, General purpose flags, the Register Bank Select, the Overflow flag, and the Parity flag.

Data Pointers:

The Data Pointers are used in MOVX instructions that transfer data to and from off-chip External Data. These pointers hold the address of the External Memory location with which data is to be transferred. Since data can be moved to and from this external memory, the CS6208 has provided two separate Data Pointers. The user can switch between either of the two with minimum soft-ware overhead, but thereby greatly increasing the system throughput.The CS6208 also has a decrement data pointer instruction (DEC DPTR) with opcode 0xA5.

Scratch-pad RAM:

The CS6208 has a 256 byte on-chip scratch-pad RAM. This can be used by the user for temporary storage during program execution. A certain section of this RAM is bit addressable, and can be directly addressed for this purpose.

Stack Pointer:

The CS6208 has a 8-bit Stack Pointer which points to the top of the Stack. This stack resides in the Scratch Pad RAM in the CS6208. Hence the size of the stack is limited by the size of this RAM.

I/O Ports:

The CS6208 offers four 8-bit general purpose I/O ports. Each I/O port has a 8-bit SFR associated with it that can be written or read. In general the port pins are read during a read operation, however a few exceptions do exist. Each port is represented by a SFR or ESFR location. I/O port1 and I/O port 3 from the standard 8051 are included, in addition 16 general purpose I/O pins (PortB and PortC) are included which replace the standard 8051 ports 0 and 2. Standard ports 0 and 2 are used for the demultiplexed data/address bus on the CS6208.

Timers/Counters:

The CS6208 has three 16-bit Timer/Counters. Each timer is contained in two SFR locations that can be written or read by software. There are also some other SFRs associated with the timers that control their mode and operation.

UART:

The CS6208 provides one serial I/O port, which can operate in synchronous as well as asynchronous mode. The UART has several SFR locations associated with it which can be both read as well as written to.

MEMORY ORGANIZATION:

The CS6208 separates the memory into two separate regions, the Program Memory and the Data Memory. The Program Memory is used to store the instruction op-codes, while the Data Memory is used to store data or for memory mapped devices.

Program Memory:

The Program Memory on the CS6208 can be expanded up to 131K bytes using memory banking. Memory may be completely external to the CS6208, or it may include the 65K byte on-chip ROM. All instructions are fetched for execution from this memory area, which may also be accessed using the MOVC instruction.

Program Memory Map: Pin EA = 0, External 131K byte Program Memory.

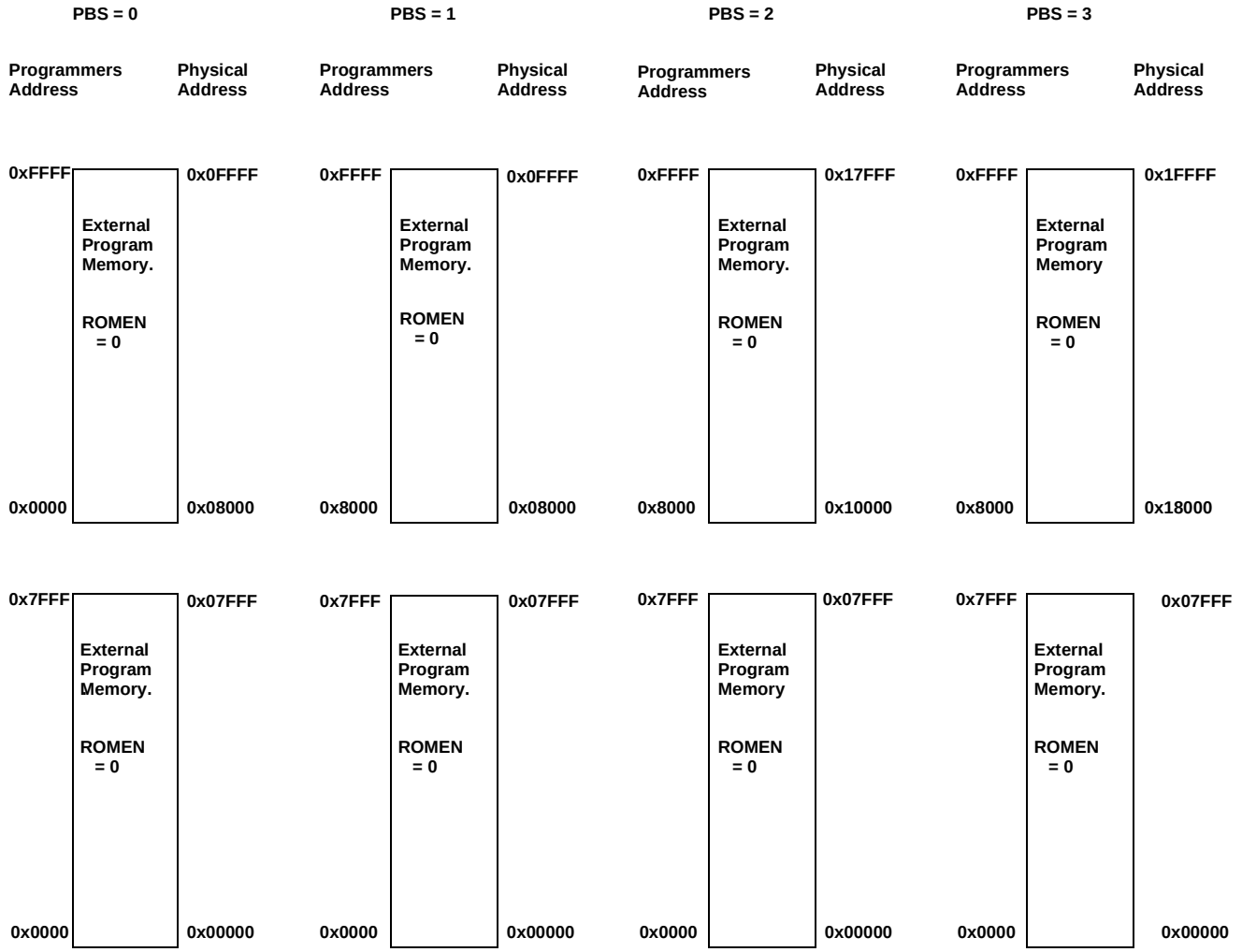


Figure 1 - ProgramMemory Map, pin EA = 0.

The ROMEN, RAMEN, and ADDR[17:15] signals are used to determine if on-chip memory or off-chip memory is being accessed. The ROMEN signal indicates an access by the internal program memory (ROMEN = 1), or an access by the external program memory (ROMEN = 0) and should be used together with any external memory decoding circuitry to differentiate between an internal/external program memory access.

Note that the programmer always “sees” the same address range when writing application software as shown in figure1 and 1.1. The physical address which corresponds to the currently selected bank is different and is also shown in figures 1 and 1.1. When using external memory banking note that some of the banks overlap. This is shown beneath the “Physical Address” column is figures 1, 1.1 and 2.

Program Memory Map: Pin EA = 1, Internal 65K byte and External 98K byte Program Memory.

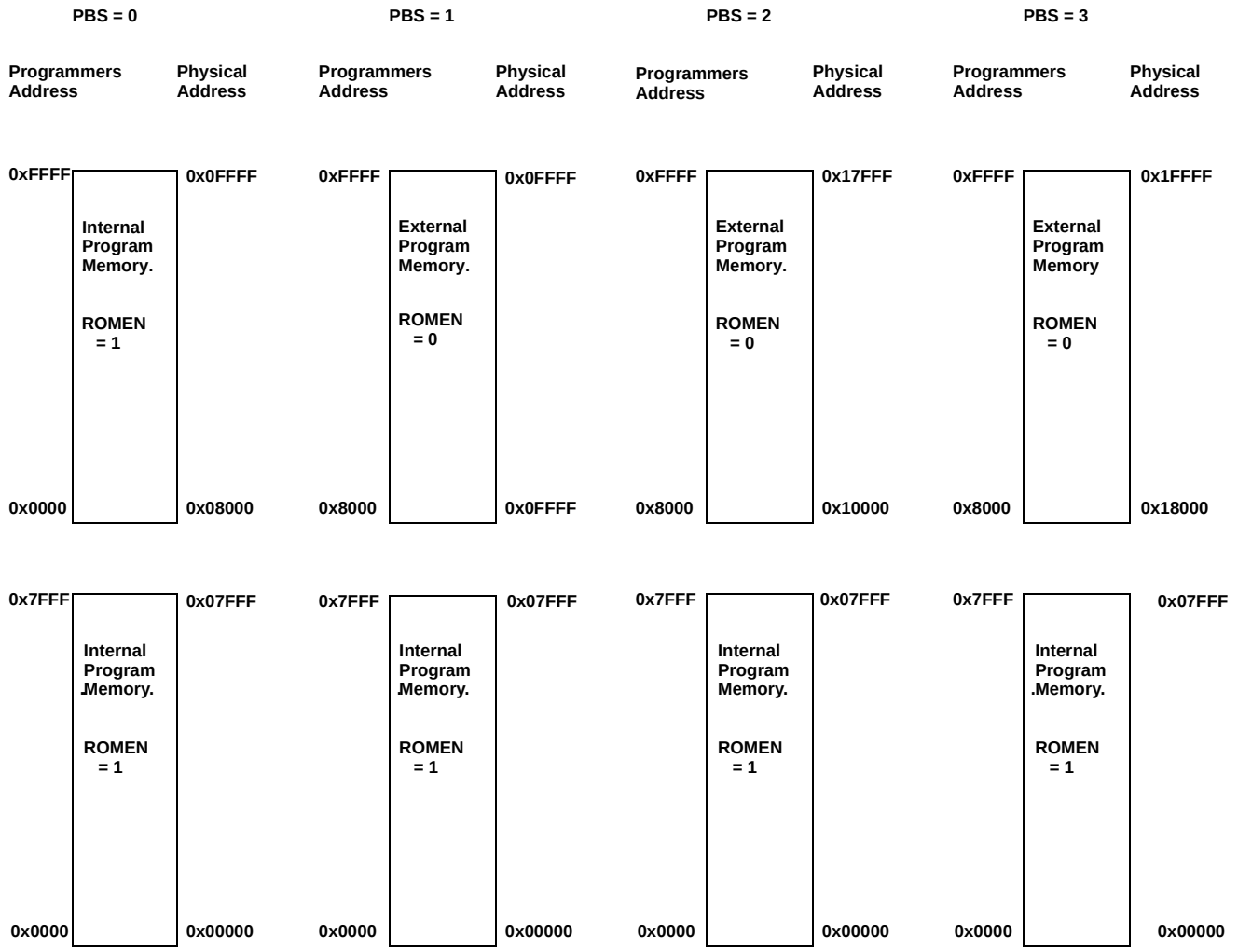


Figure 1.1 - Program Memory Map, pin EA = 1.

Data Memory:

The CS6208 can access up to 98K bytes of external Data Memory using memory banking. This memory region is accessed using the MOVX instruction. In addition the CS6208 has 32K bytes of on-chip RAM, which may be accessed either by direct addressing or by indirect addressing. There are also External Special Function Registers (ESFRs) which can only be accessed by direct addressing, and are used to configure the integrated on-chip.peripherals. The RAMEN signal indicates an access by the internal data memory (RAMEN = 1), or an access by the external data memory (RAMEN = 0) and should be used together with any external memory decoding circuitry to differentiate between an internal/external data memory access. Note that the programmer always "sees" the same address range when writing application software as shown in figure 2. The physical address which corresponds to the currently selected bank is different and is also shown in figure 2.

Data Memory Map: Internal 32K byte SRAM and external 65K byte SRAM.

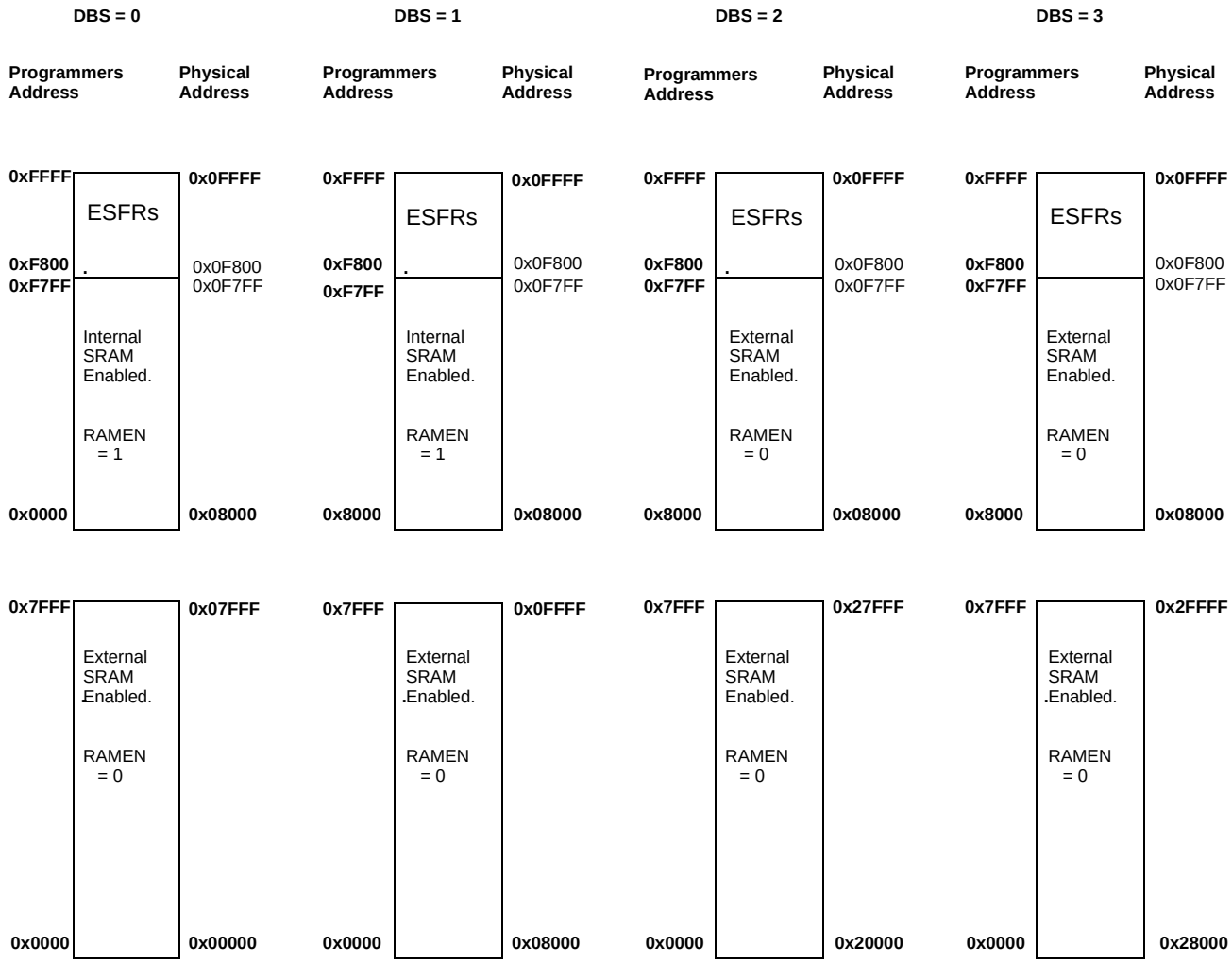


Figure 2 - Data Memory Map

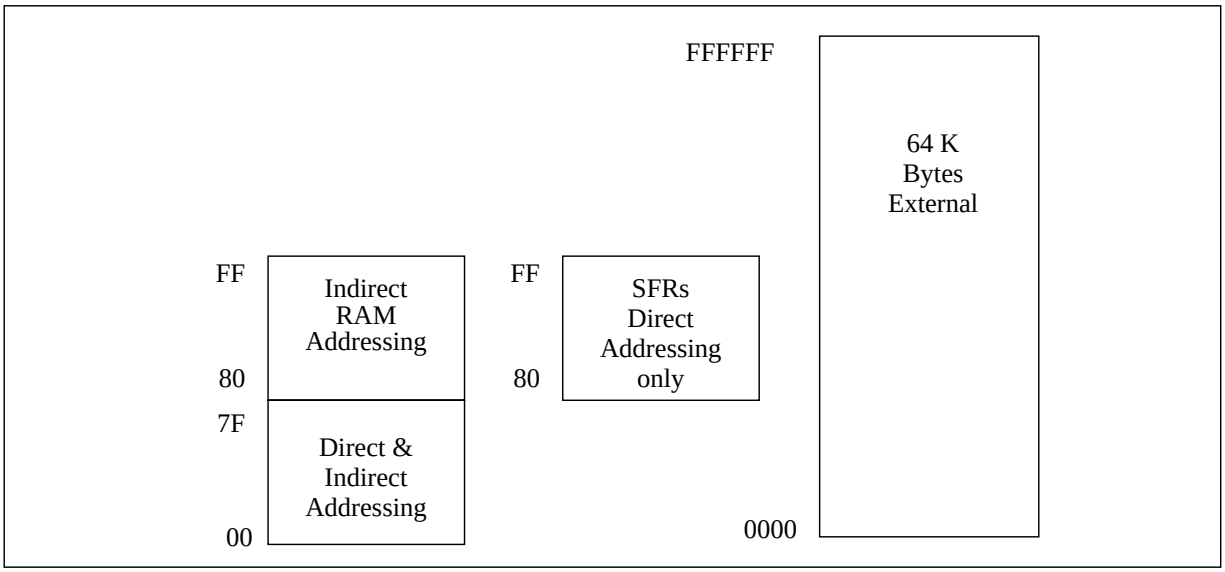


Figure-3 Internal Register and Memory Map

External Control Register (SFR) Address Map and Detailed Description (located in on-chip xdata SRAM)

All register contents are set to default at power-on or by external RESET.

Register Address	Register Name	Description
FFFFh	CCR	Clock Control Register
FFFEh	PLLFCR	PLL Frequency Control Register
FFFDh	ADCCR	ADC Control Register
FFFCh	ADCCOR	ADC Conversion Output Register
FFFBh	ADCSR	ADC Control Switch
FFFAh	ADCAR	ADC address. (XX_ADDR)
FFF9h	ADCDAR	ADC data. (XX_DATA)
FFF8h	ADCDIVR	ADC Clock Divider
FFF7h	PBR	Port B Register
FFF6h	PBOENR	Port B Output Enable Register
FFF5h	PCR	Port C Register
FFF4h	PCOENR	Port C Output Enable Register
FFF3h	MBSR	Memory Bank Select Register
FFF2h	HRCSA	High byte of RAM Check Sum Start Address
FFF1h	LRCSA	Low byte of RAM Check Sum Start Address
FFF0h	HRCSBC	High byte RAM Check Sum offset
FFEFh	LRCSBC	Low byte RAM Check Sum offset
FFEEh	HRCSD	High byte of RAM Check Sum result
FFEDh	LRCSD	Low byte of RAM Check Sum result
FFECh	RCSCR	RAM check Sum Control Register
FFEBh	INSTAR	Interrupt Status Register
FFEAh	PHYCTRL	Internal 10MHz PHY Control
FFE9h	B10CTRL	Current fine tune for Internal 10MHz PHY
FFE8h	PAR	Port A Register
FFE7h	PAOENR	Port A Output Enable Register
FFE6h	Reserved	Reserved
FFE5h	Reserved	Reserved
FFE4h	Reserved	Reserved
FFE3h	I2C1SR	I2C1 Status Register
FFE2h	I2C1RBR	I2C1 Receive Buffer Register
FFE1h	I2C2SR	I2C2 Status Register
FFE0h	I2C2RBR	I2C2 Receive Buffer Register
FFDFh	I2C1CR	I2C1 Control Register
FFDEh	I2C1TBR	I2C1 Transmit Buffer Register
FFDDh	SI2CICR	Slave I2C1 Control Register
FFDCh	SI2CIAR	Slave I2C1 Address Register
FFDBh	I2C2CR	I2C2 Control Register
FFDAh	I2C2TBR	I2C2 Transmit Buffer Register

Register Address	Register Name	Description
FFD9h	SI2C2CR	Slave I2C2 Control Register
FFD8h	SI2C2AR	Slave I2C2 Address Register

CCR - Clock Control Register: FFFFh

Bit	Default	Description
7-4	100, R/W	Set the delay time when switching between slow clock and PLL clock source OSC waiting cycles (OSCWC) (2ms ~ 300ms) System can switch to fast clock source after waiting N slow clock cycles. OSCWC 000 N = 511 (2ms) 111 N = OSCWC * 4096 -1 (20ms * OSCWC) others N = OSCWC * 4096 -1 (300ms)
3	0, R/W	port A IO mode (paiom): 0: lower address byte out from port A 1: programmable IO
2	1, R/W	port B IO mode (pbiom): 0: PLL out on port B[0] 1: programmable IO
1-0	0, R/W	clock switch control **In normal operated mode:** 00: system using fast OSC and enable slow OSC 01: system using fast OSC and disable slow OSC 1x: system using slow OSC and disable PLL **In power down mode:** xx: disable fast OSC, PLL and slow OSC

PLLFCR - PLL Frequency Control Register: FFFEh

Bit	Default	Description
7-5	1000, R/W	High nibble of PLL divisor frequency.
4-0	1001, R/W	Low nibble of PLL divisor frequency. System Clock Frequency (Mhz) = External Crystal Frequency (Mhz) * (PLLFC[3:0] + 1) / (PLLFC[7:4] + 1) External Crystal Frequency must = 20Mhz to use ethernet.

ADCCR - ADC Control Register: FFFDh

Bit	Default	Description
7	0, R/W	XX_WRITE, pulse high to let data be written.
6	0, R/W	XX_CAL, ADC doing calibration when XX_CAL keeps at high
5	0, R/W	ADC Enable when signal keeps at high.
4	0, R	XX_READY (Read Only). 1: XX_DATA and XX_ADDR are ready to be modified
3	0, R	CBR_DONE (Read Only), high to mean calibration is done
2	0, R	EOC end of conversion (Read Only), digital output data is valid when pulse high

Bit	Default	Description
1-0	00, R/W	ADC conversion results [9:8]

ADCCOR - ADC Conversion Output Register: FFFCh

Bit	Default	Description
7-0	8'h00, R	ADC conversion results [7:0]

ADCSR - ADC Control Switch: FFFBh

Bit	Default	Description
7-5	000, R	Reserved
4	0, R	Clock switches acknowledge (Read Only). 1: done, 0: in progress
3-0	0001, R/W	ADC analog input control switches; 0001 AIN[0] 0010 AIN[1] 0100 AIN[2] 1000 AIN[3]

ADCAR - ADC address: FFFAh

Bit	Default	Description
7-0	8'h00, R/W	XX_ADDR (R/W), Identified address bus when writing data.

ADCDAR - ADC data: FFF9h

Bit	Default	Description
7-0	8'h00, R/W	XX_DATA (R/W), write/read data bus

ADCDIVR - ADC Clock Divider: FFF8h

Bit	Default	Description
7-0	8'h07, R/W	0: ADC clock off 1~ff: ADC clock frequency = PLL output frequency / (ADCDIV +1) Maximum ADC clock frequency = 10 MHz

PBR - Port B Register: FFF7h

Bit	Default	Description
7-0	8'hFF, R/W	Read: p0in = PBOENR ? PBR: port B Write: PBR = PBOENR ? P0in: port B

PBOENR - Port B Output Enable Register: FFF6h

Bit	Default	Description
7-0	8'h00, R/W	Port B = PBOENR ? PBR: peripheral inputs

PCR - Port C Register: FFF5h

Bit	Default	Description
7-0	8'hFF, R/W	Read: p0in = PCOENR ? PCR: port C Write: PCR = PCOENR ? P0in: port C

PBOENR - Port C Output Enable Register: FFF4h

Bit	Default	Description
7-0	8'h00, R/W	Port C = PCOENR ? PCR: peripheral inputs

MBSR - Memory Bank Select: FFF3h

Bit	Default	Description
7	0, R/W	Reserved
6-5	00, R/W	External Data Bank Select (DBS).
4-3	00, R/W	External Program Bank Select (PBS).
2-0	000, R/W	Reserved

Ram Check Sum operations apply to internal 32k-byte XDATA RAM only.

HRCSA - High Byte of RAM Check Sum start address: FFF2h

Bit	Default	Description
7-0	8'h00, R/W	High byte of RAM Check Sum start address

LRCSA - Low Byte of RAM Check Sum start address: FFF1h

Bit	Default	Description
7-0	8'h00, R/W	Low byte of RAM Check Sum start address

HRCSBC - High Byte of RAM Check Sum offset: FFF0h

Bit	Default	Description
7-0	8'h00, R/W	High byte of RAM Check Sum offset

LRCSBC - Low Byte of RAM Check Sum offset: FFEFh

Bit	Default	Description
7-0	8'h00, R/W	The ending address for Check Sum = {*HRCSA, *LRCSA} + {*HRCSBC, *LRCSBC}

HRCSD- High Byte of RAM Check Sum result: FFEEh

Bit	Default	Description
7-0	8'h00, R/W	High byte of RAM Check Sum result

LRCSD - Low Byte of RAM Check Sum result: FFEDh

Bit	Default	Description
7-0	8'h00, R/W	Low byte of RAM Check Sum result

RCSCR - RAM Check SUM Control Register.: FFECh

Bit	Default	Description
7-3	00000, R/W	Reserved
2	0, R	ramcksumFail (Read only). 1: Abnormal completion of Check Sum.
1	0, R	ramcksumOK (Read only). 0: Check Sum in progress. 1: Finish Check Sum.
0	0, R	ramck_enable. 1: enable RAM Check Sum.

INSTAR - Interrupt Status Register: FFEBh

Bit	Default	Description
7-4	0000, R/W	Reserved
3	0, R/W	set /clear RAM check sum interrupt (only INT0 - priority selectable)
2	0, R/W	set/clearI2C2 interrupt (only INT0 - priority selectable)
1	0, R/W	set/clearI2C1 interrupt (only INT0 - priority selectable)
0	0, R/W	set / clear MAC interrupt (only INT1 - always highest priority)

PHYCTRL - Internal 10MHz PHY Control: FFEAh

Bit	Default	Description
7-2	000000, R/W	Reserved
1	0, R/W	0 = use external PHY, 1 = use internal PHY
0	1, R/W	0 = internal PHY enabled, 1 = internal PHY power down

B10CTRL-Current Fine Tune for INternal PHY :FFE9h

Bit	Default	Description
7-0	8'h00, R/W	Current fine tune for Internal 10MHz PHY

PAR - Port A Register: FFE8h

Bit	Default	Description
7-0	8'hFF, R/W	Read : p0in = PAOENR ? PAR: port A Write: PAR = PAOENR ? PAR: port A

PAOENR - Port A Output Enable Register: FFE7h

Bit	Default	Description
7-0	8'hFF, R/W	paiom=0: Port A = ADRL (P0-out latched by ALE falling) paiom=1: Port A = PAOENR ? PAR: peripheral inputs

I2C1SR - I2C1 Status Register: FFE3h

Bit	Default	Description
7	0, R	master/slave 1:slave mode 0:master mode
6	0, R	Reserved
5	0, R	SSTR slave start.
4	0, R	SSTP slave stop.
3	0, R	STPCMP This bit is set when stop has been transmitted.
2	0, R	SRW 1:savle read 0:slave write
1	0, R	RXNACK 1:received NACK. 0 : received ACK
0	0, R	I2C_RDY 1: Receive or transmit is ready

I2C1RBR - I2C1 Receive Buffer Register:FFE2h

Bit	Default	Description
7-0	8'hFF, R	Data received through the I^2C interface

I2C2SR - 2C2 Status Register: FFE1h

Bit	Default	Description
7	1, R	master/slave 1:slave mode 0:master mode
6	0, R	non
5	0, R	SSTR slave start.
4	0, R	SSTP slave stop.
3	0, R	STPCMP This bit is set when stop has been transmitted
2	0, R	SRW 1: slave read 0: slave write
1	0, R	RXNACK 1:received NACK. 0 : received ACK.
0	0, R	I2C_RDY 1: Received or transmitted is ready.

I2C2RBR - I2C2 Receive Buffer Register: FFE0h

Bit	Default	Description
7-0	8'hFF, R	Data received through the I^2C interface.

I2C1CR - I2C1 Control Register: FFDFh

Bit	Default	Description
7	0, W	Slave/Master Set this bit 1:slave mode 0:master mode.

Bit	Default	Description
6-5	00, W	MCLK Select ISCL clock frequency in master mode. 00 : 400 KHz 01 :100KHz 11 : 200KHz 10 : 50KHz
4	0, W	MSTR Once this bit is set, hardware set start signal in master mode.
3	0, W	MSTR Once this bit is set, hardware set stop signal in master mode.
2	0, W	RX/TX set this bit in master mode 0 : transmit 1: receive
1	1, W	TXNAK set this bit in Master/Slave mode 0: outputs NACK 1: outputs ACK
0	0, W	CLRRDY setting this bit will clear the I2CRDY bit in the I2C1 Status Register.

I2C1TBR - I2C1 Transmit Buffer Register: FFDEh

Bit	Default	Description
7-0	8'hFF, W	Data to be transmitted through the I2C interface

SI2C1CR - Slave I2C1 Control Register: FFDDh

Bit	Default	Description
7-6	00, W	Reserved.
5	0, W	CLRSSTR Setting this bit will clear the STAR bit in I2C1 Status Register[5].
4	0, W	CLRSSTP Setting this bit will clear the STOP bit in I2C1 Status Register[4].
3-0	00, W	Reserved

SI2C1AR - Slave I2C1 Address Register: FFDCh

Bit	Default	Description
7-1	1010000, W	These bits are device address in slave mode for I2C1.
0	1, W	Enable device address to be compared when this bit is set .

I2C2CR - I2C2 Control Register: FFDBh

Bit	Default	Description
7	1, W	Slave/Master Set this bit 1:slave mode 0:master mode
6-5	00, W	MCLK Select ISCL clock frequency in master mode. 00 : 400 KHz 01 :100KHz 11 : 200KHz 10 : 50KHz
4	0, W	MSTR Once this bit is set, hardware set start signal in master mode.
3	0, W	MSTR Once this bit is set, hardware set stop signal in master mode.

Bit	Default	Description
2	0, W	RX/TX set this bit in master mode 0 : transmit 1: receive
1	1, W	TXNAK set this bit in Master/Slave mode 0: outputs NACK 1:outputs ACK
0	0, W	CLRRDY setting this bit will clear the I2CRDY bit in the I2C2 Status Register.

I2C2TBR - I2C2 Transmit Buffer Register: FFDAh

Bit	Default	Description
7-0	8'hFF, W	Data to be transmitted through the I2C interface.

SI2C2CR - Slave I2C2 Control Register: FFD9h

Bit	Default	Description
7-6	00, W	Reserved.
5	0, W	CLRSSTR Setting this bit will clear the STAR bit in I2C2 Status Register[5].
4	0, W	CLRSSTP Setting this bit will clear the STOP bit in I2C2 Status Register[4]
3-0	0000, W	Reserved.

SI2C2AR - Slave I2C2 Address Register: FFD8h

Bit	Default	Description
7-1	1011000, W	These bits are device address in slave mode for I2C2
0	1, W	Enable device address to be compared when this bit is set.

MAC Register Address Map and Detailed Description

Address	Read	Write
FF80h	Command (CR)	Command (CR)
FF81h	-	Receive Page Start (RPSTART)
FF82h	-	Receive Page Stop (RPSTOP)
FF83h	Boundary Point (BNRY)	Boundary Point (BNRY)
FF84h	Transmit Status (TSR)	Transmit Page Start (TPSTART)
FF85h	-	Transmit Byte Count 0 (TBCNT0)
FF86h	-	Transmit Byte Count 1 (TBCNT1)
FF87h	Interrupt Status (ISR)	Interrupt Status (ISR)
FF88h	-	-
FF89h	-	-
FF8Ah	-	-
FF8Bh	-	-
FF8Ch	Receive Status (RSR)	Receive Configuration (RCR)

Address	Read	Write
FF8Dh	-	-
FF8Eh	-	-
FF8Fh	-	Interrupt Mask (IMR)
FF90h	Command (CR)	Command (CR)
FF91h	Physical Address Register 0 (PAR0)	PAR0
FF92h	Physical Address Register 1 (PAR1)	PAR1
FF93h	Physical Address Register 2 (PAR2)	PAR2
FF94h	Physical Address Register 3 (PAR3)	PAR3
FF95h	Physical Address Register 4 (PAR4)	PAR4
FF96h	Physical Address Register 5 (PAR5)	PAR5
FF97h	Current Page (CURR)	Current Page (CURR)
FF98h	Multicast Address Register 0 (MAR0)	MAR0
FF99h	Multicast Address Register 1 (MAR1)	MAR1
FF9Ah	Multicast Address Register 2 (MAR2)	MAR2
FF9Bh	Multicast Address Register 3 (MAR3)	MAR3
FF9Ch	Multicast Address Register 4 (MAR4)	MAR4
FF9Dh	Multicast Address Register 5 (MAR5)	MAR5
FF9Eh	Multicast Address Register 6 (MAR6)	MAR6
FF9Fh	Multicast Address Register 7 (MAR7)	MAR7

CR - Command, Read/Write: FF80h

Bit	Symbol	Description
7	-	-
6	-	-
5	-	-
4	-	-
3	-	-
2	XmtPkt	Transmit Packet
1	Start	NIC Start Command
0	Stop	NIC software reset command

ISR - Interrupt Status, Read/Write: FF87h

*Write 1'b1 will clear ISR bit status.

Bit	Symbol	Description
7	RstStatus	Reset Status or Receive Buffer OverFlow
6	-	-
5	-	-
4	RcvBufOverFlow	Receive Buffer OverFlow

Bit	Symbol	Description
3	XmtError	Transmit Error : Excessive Collision/TxFifo under-flow
2	RcvError	Receive Error: CRC Error/Frame Alignment Error/ RxFifo overFlow
1	XmtOk	Transmit Complete without error
0	RcvOk	Receive Packet without error

IMR - Interrupt Mask, WriteOnly: FF8Fh

Bit	Symbol	Description
7	-	-
6	-	-
5	-	-
4	RcvBufOverFlowEn	Receive Buffer OverFlow Interrupt Enable
3	XmtErrorEn	Transmit Error Interrupt Enable
2	RcvErrorEn	Receive Error Interrupt Enable
1	XmtOkEn	Transmit Complete Interrupt Enable
0	RcvOkEn	Receive Complete Interrupt Enable

TSR - Transmit Status, ReadOnly: FF84h

Bit	Symbol	Description
7	-	
6	-	
5	TxFifoUnderRun	Transmit Fifo UnderFlow
4	-	
3	XmtAbort	Transmit Abort
2	Col	Transmission collided at least once
1	-	
0	XmtOk	Transmit without error

RSR - Receive Status, ReadOnly: FF8Ch

Bit	Symbol	Description
7	Deferring	Deferring : CRS or COL(jabber) active
6	-	-
5	PhyAddr	Physical/Multicast Address - 0:PhyAddr 1:Multicast/Broadcast
4	RcvBufOverFlow	Receive Buffer OverFlow
3	RxFifoOverFlow	Receive Fifo OverFlow
2	-	-
1	CRCError	Receive Packet with CRC error

Bit	Symbol	Description
0	RcvOk	-

RCR - Receive Configuration, Read/Write: FFC8h

Bit	Symbol	Description
7	-	-
6	-	-
5	-	-
4	AP	Accept Physical Address
3	AM	Accept Multicast Address
2	AB	Accept Broadcast Address
1	AR	Accept Runt Packet
0	RcvOkEn	Receive Complete Interrupt Enable

Integrated Ethernet MAC (NIC Controller):

The CS6208 includes an integrated 10/100 ethernet MAC unit with 7-wire output allowing the system designer freedom to choose the specific physical layer best suited for their application.

Integrated Ethernet Physical Layer:

An on-chip 10Mbs ethernet PHY reduces the system package count for ethernet only applications. A few external passive components are al that is needed to complete a working ethernet interface.

Integrated Hardware Checksum:

A hardware checksum register is provided to speed processing of network protocols.

Please see the *CS6208 Programmers Guide - application note AN62,* for additional information on the use of integrated hardware checksum.

Integrated PLL Clock Generator:

The PLLFC (PLL Frequency Control) SFR at address 0xFFFE must be initialized by firmware for proper operation of the CS6208. This register is divided in upper and lower nibbles for programming purposes. Bits 7-4 are designated *high* and bits 3-0 are designated *low* (reset value of 0x89). The CS6208 system clock is set as follows:

System Clock Frequency (Mhz) = External Crystal Frequency (Mhz) * (1 + *low*) / (1 + *high*)

All on-chip hardware blobks are referenced to this system clock. Note that the external crystal frequency must be 20Mhz for the on-chip ethernet physical layer block to function.

32K byte On-chip XDATA SRAM:

This large block of SRAM may be utilized as XDATA space for user programs or partially allocated to the ethernet MAC for use as a transmit/receive packet buffer.

Memory Bank Selection:

The MBSR external SFR at 0xFFF3 (bits 3-4) are used to determine which CS6208 memory banking scheme is active. To use the memory banking the BANK.A51 assembly language file is needed. Please see the description of the MBSR external SFR for further details.

Analog/Digital Converter:

Please see the *CS6208 Programmers Guide* for additional information on the use of the on-chip Analog/Digital Converter. The ADC external SFRs used are: ADCCR, ADCCOR, and ADCSR at addresses 0xFFFD, 0xFFFC, and 0xFFFC respectively.

Here is the pseudo code for an ADC read operation:

```
xdata byte adcresult;
ADCSR = (ADCSR | 0x0F) & 0x01;          /* Select channel Ain0 for use. */
ADCCR = ADCCR | 0x20;                   /* Enable the ADC. */
while ( (ADCSR & 0x10) == 0x10)
    adcresult = ADCCOR;                 /* Read the ADC result. */
```

Register Map:

As mentioned previously the CS6208 has separate Program and Data Memory areas. The on-chip 256 byte scratch pad RAM is in addition to the external memory (refer to figure 3). There are also several Special Function Registers (SFRs) which can be accessed by software. The SFRs can be accessed only by direct addressing, while the on-chip RAM can be accessed by either direct or indirect addressing.

Since the scratchpad RAM is only 256 bytes it can be used only when data contents are small. In case larger data contents are present then the external Data Memory must be used. However, the on-chip RAM has the fastest access times. There are several other special purpose areas within the Scratchpad RAM. These are described as follows.

Working Registers:

There are four sets of working registers, each consisting of eight 8-bit registers. These are termed ads Banks 0, 1, 2, and 3. Individual registers within these banks can be directly accessed by separate instructions. These individual registers are named as R0, R1, R3, R4, R5, R6 and R7. However, at one time the CS6208 can work with only one particular bank. The bank selection is done by setting RS1-RS0 bits in the PSW. The R0 and R1 registers are used to store the address for indirect accessing.

Bit addressable Locations:

The Scratchpad RAM area from location 20h to 2Fh is byte as well as bit addressable. This means that a bit in this area can be individually addressed. In addition some of the SFRs are also bit addressable. The instruction decoder is able to distinguish a bit access from a byte access by the type of the instruction itself. In the SFR area, any existing SFR whose address ends in a 0 or 8 is bit addressable.

Stack:

The scratchpad RAM can be used for the stack. This area is selected by the Stack Pointer (SP), which stores the address of the top of the stack. Whenever a jump, call or interrupt is invoked the return address is placed on the stack. There is no restriction as to where the stack can begin in the RAM. By default however, the Stack Pointer contains 07h at reset. The user can then change this to any value desired. The SP will point to the last used value. Therefore, the SP will be incremented and then address saved onto the stack. Conversely, while popping from the stack the contents will be read first, then the SP decremented.

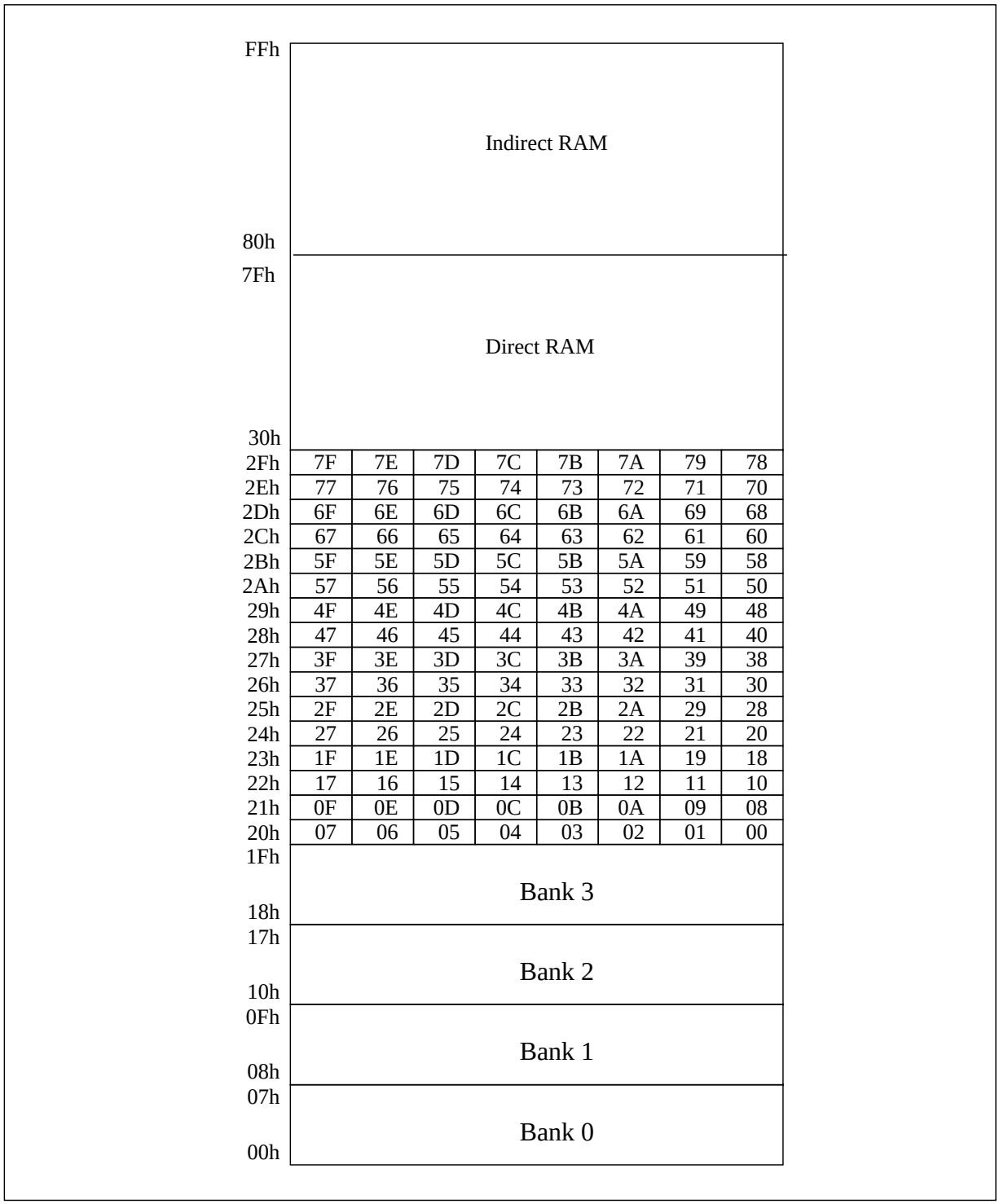


Figure-4 Scratchpad Register Addressing

8051 Style Special Function Registers:

The CS6208 uses Special Function Registers (SFRs) to control and monitor peripherals and their modes. There are two types of SFR within the CS6208: standard 8051 style SFRs and external SFRs residing in on-chip xdata memory (accessed using the movx instruction).

8051 style SFRs reside in the register locations 80-FFh and are accessed by direct addressing only. Some of these SFRs are bit addressable. This is very useful in cases where we wish to modify a particular bit without changing the others. The SFRs that are bit addressable are those whose addresses end in 0 or 8. The CS6208 contains all the SFRs present in the standard 8051. However some additional SFRs have been added, in these cases the unused bits in the original 8051 have been given new functions. The list of 8051 style SFRs is as follows.

Table-1 Standard 8051 SFR Location Table

F8	EIP							
F0	B							
E8	EIE							
E0	ACC							
D8	WDCON							
D0	PSW							
C8	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2		
C0								TA
B8	IP	SADEN						
B0	P3							
A8	IE	SADDR						
A0	P2							
98	SCON	SBUF						
90	P1							
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	
80		SP	DPL	DPH	DPL1	DPH1	DPS	PCON

a. The SFRs in the column with dark borders are bit-addressable.

The table is condensed with eight locations per row. Empty locations indicate that these are no registers at these addresses. When a bit or register is not implemented, it will read high.

A brief description of the standard 8051 SFRs now follows.

PORT 0

Since in the CS6208 the port 0 is used exclusively as the multiplexed address/data bus, it cannot be used by the user as a general purpose I/O port. Hence there is no Port 0 latch.

STACK POINTER

7	6	5	4	3	2	1	0
SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

Mnemonic: SP Address: 81h

The Stack Pointer stores the Scratchpad RAM address where the stack begins. In other words it always points to the top of the stack.
The SP is set to 07h on any reset.
There is unrestricted read/write access to this SFR.

DATA POINTER LOW

7	6	5	4	3	2	7	6
DPL.7	DPL.6	DPL5	DPL.4	DPL.3	DPL.2	DPL.1	DPL.0

Mnemonic: DPL Address: 82h

This is the low byte of the standard 8051 16-bit data pointer.
The DPL is reset to 00h by a reset.
There is unrestricted read/write access to this SFR.

DATA POINTER HIGH

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

Mnemonic: DPH Address: 83h

This is the high byte of the standard 8051 16-bit data pointer.
The DPH is reset to 00h by a reset.
There is unrestricted read/write access to this SFR.

DATA POINTER LOW1

	7	6	5	4	3	2	1	0
DPL1.7	DPL1.6	DPL15	DPL1.4	DPL1.3	DPL1.2	DPL1.1	DPL1.0	

Mnemonic: DPL1 Address: 84h

This is the low byte of the new additional 16-bit data pointer that has been added to the CS6208. The user can switch between DPL, DPH and DPL1, DPH1 simply by setting DPS = 1.The instructions that use DPTR will now access DPL1 and DPH1 in place of DPL and DPH. If they are not required they can be used an conventional register locations by the user.

The DPL1 is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

DATA POINTER HIGH1

7	6	5	4	3	2	1	0
DPH1.7	DPH1.6	DPH1.5	DPH1.4	DPH1.3	DPH1.2	DPH1.1	DPH1.0

Mnemonic: DPH1 Address: 85h

This is the high byte of the new additional 16-bit data pointer that has been added to the CS6208. The user can switch between DPL, DPH and DPL1, DPH1 simply by setting DPS = 1. The instructions that use DPTR will now access DPL1 and DPH1 in place of DPL and DPH. If they are not required they can be used an conventional register locations by the user.

The DPH1 is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

DATA POINTER SELECT

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

Mnemonic: DPS Address: 86h

DPS.0 This bit is used to select either the DPL,DPH pair or DPL1,DPH1 pair as the active Data Pointer. When set to 1, DPL1,DPH1 will be selected, else DPL,DPH will be selected.

DPS.1-7 These bits are reserved but will read 0.

The DPS is reset to 00h by a reset, and so will select standard DPTR.

There is unrestricted read/write access to this SFR

POWER CONTROL

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	-	GF1	GF0	PD	IDL

Mnemonic: PCON Address: 87h

SMOD This bit doubles the serial baud rate in mode 1, 2, and 3 when set to 1.

SMOD0 Framing Error Detection Enable: When SMOD0 is set to 1, then SCON.7 now indicates a Frame Error and acts as the FE flag. When SMOD0 is 0, then SCON.7 acts as per the standard 8051 function.

GF1-0 These two bits are general purpose user flags.

PD Setting this bit causes the CS6208 to go into the POWERDOWN mode. In this mode all the clocks are stopped and program execution is frozen.

IDL Setting this bit causes the CS6208 to go into the IDLE mode. In this mode the clocks to the CPU is stopped, so program execution is frozen. But the clock to the serial, timer and interrupt blocks is not stopped, and these blocks continue operating unhindered.

The PCON is reset to 00110000b by a reset.

There is unrestricted read/write access to this SFR.

TIMER CONTROL

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

Mnemonic: TCON Address: 88h

TF1 Timer 1 overflow flag: This bit is set when Timer 1 overflows. It is cleared automatically when the

program does a timer 1 interrupt service routine. Software can also set or clear this bit.

TR1 Timer 1 run control: This bit is set or cleared by software to turn timer/counter on or off.

TF0 Timer 0 overflow flag: This bit is set when Timer 0 overflows. It is cleared automatically when the program does a timer 0 interrupt service routine. Software can also set or clear this bit.

TR0 Timer 0 run control: This bit is set or cleared by software to turn timer/counter on or off.

IE1 Interrupt 1 edge detect: Set by hardware when an edge/level is detected on INT1. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the pin.

IT1 Interrupt 1 type control: Set/cleared by software to specify falling edge/ low level trig gered external inputs.

IE0 Interrupt 0 edge detect: Set by hardware when an edge/level is detected on INT0. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the pin.

IT0 Interrupt 0 type control: Set/cleared by software to specify falling edge/ low level triggered external inputs.

The TCON is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

TIMER MODE CONTROL

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

TIMER 1 TIMER 2

Mnemonic: TMOD Address: 89h

GATE Gating control: When this bit is set, Timer/counter x is enabled only while INTx pin is high and TRx control bit is set. When cleared, Timerx is enabled whenever TRx control bit is set.

C/T Timer or Counter Select: When cleared, the timer is incremented by internal clocks. When set, the timer counts high-to-low edges of the Tx pin.

M1 M0 Mode Select bits:

M1	M0	Mode
0	0	Mode 0: 8-bits with 5-bit pre-scaler.
0	1	Mode 1: 18-bits, no pre-scaler.
1	0	Mode 2: 8-bits with auto-reload from THx
1	1	Mode 3: (Timer 0) TL0 is an 8-bit timer/counter controlled by the standard Timer 0 control bits. TH0 is a 8-bit timer only controlled by Timer 1 control bits. (Timer 1) Timer/counter is stopped.

The TMOD is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

TIMER 0 LSB

7	6	5	4	3	2	1	0
TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0

Mnemonic: TL0 Address: 8Ah

TL0.7-0 Timer 0 LSB

The TL0 sfr is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

TIMER 1 LSB

7	6	5	4	3	2	1	0
TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0

Mnemonic: TL1 Address: 8Bh

TL1.7-0 Timer 1 LSB
The TL1 sfr is set to 00h on any reset.
There is unrestricted read/write access to this SFR.

TIMER 0 MSB

7	6	5	4	3	2	1	0
TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0

Mnemonic: TH0 Address: 8Ch

TH0.7-0 Timer 0 MSB
The TH0 sfr is set to 00h on any reset.
There is unrestricted read/write access to this SFR.

TIMER 1 MSB

7	6	5	4	3	2	1	0
TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0

Mnemonic: TH1 Address: 8Dh

TH1.7-0 Timer 1 MSB
The TH1 sfr is set to 00h on any reset.
There is unrestricted read/write access to this SFR.

CLOCK CONTROL

7	6	5	4	3	2	1	0
WD1	WD0	T2M	T1M	T0M	MD2	MD1	MD0

Mnemonic: CKCONAddress: 8Eh

WD1-0 Watchdog timer mode select bits: These bits determine the time-out period for the watchdog timer. In all the four time-out options the reset time-out is 512 clock more than the interrupt time-out. period.

WD1	WD0	Interrupt time-out	Reset time-out
0	0	217	217 + 512
0	1	220	220 + 512
1	0	223	223 + 512
1	1	226	226 + 512

T2M Timer 2 clock select: When T2M is set to 1, timer 2 uses a divide by 4 clock, and when set to 0 uses a divide by 12 clock

T1M Timer 1 clock select: When T1M is set to 1, timer 1 uses a divide by 4 clock, and when set to 0 uses a divide by 12 clock.

T0M Timer 0 clock select: When T0M is set to 1, timer 0 uses a divide by 4 clock, and when set to 0 uses a divide by 12 clock.

MD2-0 Stretch MOVX select bits: These three bits are used to select the stretch value for the MOVX instruction. Using a variable MOVX length, enables the user to access slower memory devices or peripherals without the need for external circuits. The RD or WR strobe will be stretched by the selected interval. Al internal timing s are also stretched by the same amount. This operation is transparent to the user. By default, the stretch has value 1 cycle. If the user needs faster accessing, then stretch value of 0 should be selected.

MD2	MD1	MD0	Stretch value	MOVX duration
0	0	0	0	2 machine cycles
0	0	1	0	3 machine cycles
0	1	0	1	4 machine cycles
0	1	1	3	5 machine cycles
1	0	0	4	6 machine cycles
1	0	1	5	7 machine cycles
1	1	0	6	8 machine cycles
1	1	1	7	9 machine cycles

The CKCON sfr is set to 00000001b on any reset.

There is unrestricted read/write access to this SFR.

PORT 1

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

Mnemonic: P1 Address: 90h

P1.7-0 General purpose I/O port. Most instructions will read the port pins in case of a port read access, however in case of read-modify-write instructions, the port latch is read.

The P1 sfr is set to FFh by a reset.

There is unrestricted read/write access to this SFR.

SERIAL PORT CONTROL

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

Mnemonic: SCON Address: 98h

SM0/FE Serial port, Mode 0 bit or Framing Error Flag: The SMOD0 bit in PCON sfr determines wether this bit acts as SM0 or as FE. the operation of SM0 is described below.

When used as FE, this bit will be set to indicate an invalid stop bit. This bit must be manually cleared in software to clear the FE condition.

SM1 Serial port Mode bit 1:

SM0	SM1	Mode	Description	Length	Baud rate
0	0	0	Synchronous	8	4/12 tclk
0	1	1	Asynchronous	10	variable
1	0	2	Asynchronous	11	64/32 tclk
1	1	3	Asynchronous	11	variable

SM2 Multiple MCU communication: Setting this bit to 1 enables the multiprocessor communication feature in mode 2 and 3. In mode 2 or 3, if SM2 is set to 1, then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1, then RI will not be activated if a valid stop bit was not received. In mode 0, the SM2 bit controls the serial port clock. If set to 0, then the serial port runs at a divide by 12 clock of the oscillator. This gives compatibility with the standard 8051. When set to 1, the serial clock become divide by 4 of the oscillator clock. This results in faster synchronous serial communication.

REN Receive enable: When set to 1 serial reception is enabled, else reception is disabled.

TB8 This is the 9th bit to be transmitted in modes 2 and 3. this bit is set and cleared by soft ware as

desired.

RB8 In modes 2 and 3 this is the received 9th data bit. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0 it has no function.

TI Transmit interrupt flag: This flag is set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in all other modes during serial transmission. This bit must be cleared by software.

RI Receive interrupt flag: This flag is set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bits time in the other modes during serial reception. However the restrictions of SM2 applies on this bit. This bit can be cleared only by software

The SCON sfr is set to 00h by a reset.

There is unrestricted read/write access to this SFR

SERIAL DATA BUFFER

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

Mnemonic: SBUF Address: 99h

SBUF.7-0 Serial data is read from or written to this location. It actually consists of two separate 8-bit registers. On is the receive resister, and the other is the transmit buffer. Any read access gets data from the receive data buffer, while write accesses are to the transmit data buffer.

The SBUF sfr is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

PORT 2

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

Mnemonic: P2 Address: A0h

P2.7-0 Non-multiplexed address bus A15-A8: The port latch cannot be used for general I/O purposes but exists to support the MOVX instructions. Port 2 data will only be brought out on the P2.7-0 pins during indirect MOVX instructions.

The P2 sfr is set to FFh by a reset.

There is unrestricted read/write access to this SFR.

INTERRUPT ENABLE

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

Mnemonic: IE Address: A8h

EA Global enable: Enable/disable all interrupts except for PFI.
IE.6 Un-implemented bits, will read high.
ET2 Enable Timer 2 interrupt.
ES Enable Serial interrupt.
ET1 Enable Timer 1 interrupt
EX1 Enable external interrupt 1
ET0 Enable Timer 0 interrupt
EX0 Enable external interrupt 0

The IE sfr is set to 01000000b by a reset.

There is unrestricted read/write access to this SFR.

SLAVE ADDRESS

7	6	5	4	3	2	1	0
SADDR7	SADDR6	SADDR5	SADDR4	SADDR3	SADDR2	SADDR1	SADDR0

Mnemonic: SADDR Address: A9h

SADDR The SADDR should be programmed to the given or broadcast address for serial port to which the slave processor is designated.

The SADDR sfr is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

PORT 3

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

Mnemonic: P3 Address: B0h

P3.7-0 General purpose I/O port. Each pin also has a alternate input or output function. This alternate function is enabled if the corresponding port latch bit is set to 1, else the port pin will remain stuck at 0.

P3.7 $\overline{RD}$ Strobe for read from external RAM
P3.6 $\overline{WR}$ Strobe for write to external RAM
P3.5 T1 Timer/counter 1 external count input
P3.4 T0 Timer/counter 0 external count input
P3.3 $\overline{INT1}$ External interrupt 1
P3.2 $\overline{INT0}$ External interrupt 0
P3.1 TxD Serial port output
P3.0 RxD Serial port input

The P3 sfr is set to FFh by a reset.

There is unrestricted read/write access to this SFR.

INTERRUPT PRIORITY

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

Mnemonic: IP Address: B8h

IP.7-6 These bits are un-implemented and will read high.
PT2 This bit defines the Timer 2 interrupt priority. PT2 = 1 sets it to higher priority level.
PS This bit defines the Serial port interrupt priority. PS = 1 sets it to higher priority level.
PT1 This bit defines the Timer 1 interrupt priority. PT1 = 1 sets it to higher priority level.
PX1 This bit defines the External interrupt 1 priority. PX1 = 1 sets it to higher priority level.
PT0 This bit defines the Timer 0 interrupt priority. PT0 = 1 sets it to higher priority level.
PX0 This bit defines the External interrupt 0 priority. PX0 = 1 sets it to higher priority level.

The IP sfr is set to 11000000b by a reset.
There is unrestricted read/write access to this SFR.

SLAVE ADDRESS MASK ENABLE

7	6	5	4	3	2	1	0
SADEN7	SADEN6	SADEN5	SADEN4	SADEN3	SADEN2	SADEN1	SADEN0

Mnemonic: SADEN Address: B9h

SADEN This register enables the Automatic Address Recognition feature of the Serial port.When a bit in the SADEN is set to 1, the same bit location in SADDR will be compared with the incoming serial port data. When SADEN.n is 0, then the bit becomes a don't care in the comparison. This register enables the Automatic Address Recognition feature of the Serial port. When all the bits of SADEN are 0, interrupt will occur for any incoming address.

The SADEN sfr is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

TIMED ACCESS

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

Mnemonic: TA Address: C7h

TA The Timed Access register controls the access to protected bits in the CS6208. To access protected bits, the user must first write AAh to the TA sfr. This must be immediately followed by a write of 55h to TA. Now a window is opened in the protected bits for three machine cycles, during which the user can write to these bits.

The TA will read FFh but normally this read should not be required.

There is unrestricted write access to this SFR, a read should not be needed.

TIMER 2 CONTROL

7	6	5	4	3	2	1	0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Mnemonic: T2CON Address: C8h

TF2 Timer 2 overflow flag: This bit is set when Timer 2 overflows. It is also set when the count is equal to the capture register in down count mode. It can be set only if RCLK and TCLK are both 0. It is cleared only by software. Software can also set or clear this bit.

EXF2 Timer 2 External Flag: A negative transition on the T2EX pin (P1.1) or timer 2 underflow/overflow will cause this flag to set based on the CP/RL2, EXEN2 and DCEN bits. If set by a negative transition, this flag must be cleared by software. Setting this bit in software or detection of a negative transition on T2EX pin will force a timer interrupt if enabled.

RCLK Receive clock Flag: This bit determines the serial port timebase when receiving data in serial modes 1 or 3. If it is 0, then timer 1 overflow is used for baud rate generation, else timer 2 overflow is used. Setting this bit forces timer 2 in baud rate generator mode.

TCLK Transmit clock Flag: This bit determines the serial port timebase when transmitting data in mode 1 and 3. If it is set to 0, the timer 1 overflow is used to generate the baud rate clock, else timer 2 overflow is used.Setting this bit forces timer 2 in baud rate generator mode.

EXEN2 Timer 2 External Enable: This bit enables the capture/reload function on the T2EX pin if Timer 2 is not generating baud clocks for the serial port. If this bit is 0, then the T2EX pin will be ignored, else a negative transition detected on the T2EX pin will result in capture or reload.

TR2 Timer 2 Run Control: This bit enables/disables the operation of timer 2. halting this will preserve the current count in TH2, TL2.

C/T2 Counter/Timer select: This bit determines wether timer 2 will function as a timer or a counter. Independent of this bit, the timer will run at 2 clocks per tick when used in baud rate generator mode. If it is set to 0, then timer 2 operates as a timer at a speed depending on T2M bit (CKCON.5), else, it will count negative edges on T2 pin.

CP/RL2 Capture/Reload Select: This bit determines whether the capture or reload function will be used for timer 2. If either RCLK or TCLK is set, this bit will not function and the timer will function in an auto-reload mode following each overflow. If the bit is 0 then auto-reload will occur when timer 2 overflows or a falling edge is detected on T2EX if EXEN2 =1. If this bit is 1, then timer 2 captures will occur when a falling edge is detected on T2EX if EXEN2=1.

The T2CON is reset to 00h by a reset.

There is unrestricted read/write access to this SFR.

TIMER 2 MODE CONTROL

7	6	5	4	3	2	1	0

TIMER 2 MODE CONTROL

-	-	-	-	-	-	-	DCEN

Mnemonic: T2MOD Address: C9h

DCEN Down Count Enable: This bit, in conjunction with the T2EX pin, controls the direction that timer 2 counts in 16-bit auto-reload mode.

The T2MOD is reset to FEh by a reset.

There is unrestricted read/write access to this SFR.

TIMER 2 CAPTURE LSB

7	6	5	4	3	2	1	0
RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0

Mnemonic: RCAP2L Address: CBh

RCAP2L Timer 2 Capture LSB: This register is used to capture the TL2 value when a timer 2 is configured in capture mode. RCAP2L is also used as the LSB of a 16-bit reload value when timer 2 is configured in auto-reload mode.

The RCAP2L sfr is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

TIMER 2 CAPTURE MSB

7	6	5	4	3	2	1	0
RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0

Mnemonic: RCAP2H Address: CBh

RCAP2H Timer 2 Capture HSB: This register is used to capture the TH2 value when a timer 2 is configured in capture mode. RCAP2H is also used as the HSB of a 16-bit reload value when timer 2 is configured in auto-reload mode.

The RCAP2H sfr is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

TIMER 2 LSB

7	6	5	4	3	2	1	0
TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0

Mnemonic: TL2 Address:CCh

TL2 Timer 2 LSB

The TL2 sfr is set to 00h on any reset.

There is unrestricted read/write access to this SFR.

TIMER 2 MSB

7	6	5	4	3	2	1	0
TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0

Mnemonic: TL2 Address:CDh

TH2 Timer 2 MSB

The TH2 sfr is set to 00h on any reset.
There is unrestricted read/write access to this SFR.

PROGRAM STATUS WORD

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P

Mnemonic: PSW Address: D0h

CY Carry flag: Set for an arithmetic operation which results in a carry being generated from the ALU. It is also used as the accumulator for the bit operations.

AC Auxiliary carry: Set when the previous operation resulted in a carry (during addition0 or a borrow (during subtraction) from the high order nibble.

F0 User flag 0: General purpose flag that can be set or cleared by the user by software.

RS.1-0 Register bank select bits:

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

OV Overflow flag: Set when a carry was generated from the seventh bit but not from the 8th bit as a result of the previous operation or vice-versa.

F1 User Flag 1: General purpose flag that can be set or cleared by the user by software

P Parity flag: Set/cleared by hardware to indicate odd/even number of 1's in the accumulator.

The PSW sfr is set to 00h by a reset.

There is unrestricted read/write access to this SFR.

WATCHDOG CONTROL

7	6	5	4	3	2	1	0
POR	EPFI	PFI	WDIF	WTRF	EWT	RWT	

Mnemonic: WDCON Address: D8h

WDCON.7 This bit is un-implemented and is read high.

POR Power-on reset flag: Hardware will set this flag on a power up condition. This flag can be read or written by software. A write by software is the only way to clear this bit once it is set.

EPFI Enable Power Fail Interrupt: This bit when set to 1, enables the Power Fail Interrupt. If this bit is cleared, then there will be no interrupt in response to a Power Fail.

PFI Power Fail Interrupt flag: This flag is set by hardware whenever the device detects a power fail condition. Software must clear this bit manually. Writing a 1 to this bit will force an interrupt if the EPFI bit is also set.

WDIF Watchdog Timer Interrupt Flag: If the watchdog interrupt is enabled, hardware will set this bit to indicate that the watchdog interrupt has occurred. If the interrupt is not enabled, then this bit indicates that the time-out period has elapsed.

WTRF Watchdog Timer Reset Flag: Hard-ware will set this bit when the watchdog timer causes a reset. Software can read it but must clear it manually. A power-fail reset will also clear the bit. This bit helps software in determining the cause of a reset. If EWT =0, the watchdog timer will have no affect on this bit.

EWT Enable Watchdog timer Reset: Setting this bit will enable the Watchdog timer Reset function.

RWT Reset Watchdog Timer: This bit helps in putting the watchdog timer into a know state. It also helps in resetting the watchdog timer before a time-out occurs. Failing to set the EWT before time-out will cause an interrupt, if EWDI (EIE.4) is set, and 512 clocks after that a watchdog timer reset will be generated if EWT is set. This bit is self-clearing.

The WDCON sfr is set to a 1x0x0xx0b on an external reset. WTRF is set to a 1 on a Watchdog timer reset, but to a 0 on power on/down resets. WTRF is not altered by an external reset. POR is set to 1 by a power-on reset. EWT is set to 0 on a Power-on reset and unaffected by other resets.

All the bits in this SFR have unrestricted read access. POR, EWT, WDIF and RWT require timed access write. The remaining bits have unrestricted write accesses.

ACCUMULATOR

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

Mnemonic: ACC Address: E0h

ACC.7-0 The A or ACC register is the standard 8051 accumulator.
The ACC is reset to 00h on any reset
This sfr has unrestricted read/write access.

EXTENDED INTERRUPT ENABLE

7	6	5	4	3	2	1	0
-	-	-	EWDI	-	-	-	-

Mnemonic: EIE Address: E8h

EIE.7-5 Reserved bits, will read high
EWDI Enable Watchdog timer interrupt
EIE.3-0 Reserved bits, will read high
The EIE sfr will be set to 11101111b on any reset.
This sfr has unrestricted read/write access.

EXTENDED INTERRUPT PRIORITY

7	6	5	4	3	2	1	0
-	-	-	PWDI	-	-	-	-

Mnemonic: EIP Address: F8h

EIP.7-5 Reserved bits, will read high
PWDI Watchdog timer interrupt priority.
EIP3-0 Reserved bits, will read high
The EIP sfr will be set to 11101111b on any reset.
This sfr has unrestricted read/write access.

INSTRUCTION SET:

The CS6208 executes all the instructions of the standard 8051 family. The operation of these instructions, their effect on the flag bits the status bits is exactly the same. However, timing of these instructions is different. The reason for this is two fold. First in the CS6208, each machine cycle consists of 4 clock periods, while in the standard 8051 it consists of 12 clock periods. Also, in the CS6208 there is only one fetch per machine cycle i.e 4 clocks per fetch, while in the standard 8051 there can be two fetches per machine cycle, which works out to 6 clocks per fetch. The advantage the CS6208 has is that since there is only one fetch per machine cycle, the number of machine cycles in most cases is equal to the number of operands that the instruction has. In case of jumps and calls there will be an additional cycle that will be needed to calculate the new address. But overall the CS6208 reduces the number of dummy fetches and wasted cycles, thereby improving the efficiency as compared to the standard 8051.

INSTRUCTION SET

Table-2 Instructions that affect Flag settings*

Instruction	Carry	Overflow	Auxiliary Carry	Instruction	Carry	Overflow	Auxiliary Carry
ADD	X	X	X	CLR C	0		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C, bit	X		
MUL	0	X		ANL C, bit	X		
DIV	0	X		ORL C, bit	X		
DA A	X			ORL C, bit	X		
RRC A	X			MOV C, bit	X		
RLC A	X			CJNE	X		
SETB C	1						

Note: X indicates that the modification is as per the result of the instruction.

Table-3 Instruction Set for CS6208

Op-code	HEX Code	Bytes	CS6208 Machine Cycle	CS6208 Clock cycles	8051 Clock cycles	CS6208 vs.8051 Speed Ratio
NOP	00	1	1	4	12	3
ADD A, R0	28	1	1	4	12	3
ADD A, R1	29	1	1	4	12	3
ADD A, R2	2A	1	1	4	12	3
ADD A, R3	2B	1	1	4	12	3
ADD A, R4	2C	1	1	4	12	3
ADD A, R5	2D	1	1	4	12	3

Op-code	HEX Code	Bytes	CS6208 Machine Cycle	CS6208 Clock cycles	8051 Clock cycles	CS6208 vs.8051 Speed Ratio
ADD A, R6	2E	1	1	4	12	3
ADD A, R7	2F	1	1	4	12	3
ADD A, @R0	26	1	1	4	12	3
ADD A, @R1	27	1	1	4	12	3
ADD A, direct	25	2	2	8	12	1.5
ADD A, #data	24	2	2	8	12	1.5
ADDC A, R0	38	1	1	4	12	3
ADDC A, R1	39	1	1	4	12	3
ADDC A, R2	3A	1	1	4	12	3
ADDC A, R3	3B	1	1	4	12	3
ADDC A, R4	3C	1	1	4	12	3
ADDC A, R5	3D	1	1	4	12	3
ADDC A, R6	3E	1	1	4	12	3
ADDC A, R7	3F	1	1	4	12	3
ADDC A, @R0	36	1	1	4	12	3
ADDC A, @R1	37	1	1	4	12	3
ADDC A, direct	35	2	2	8	12	1.5
ADDC A, #data	34	2	2	8	12	1.5
SUBB A, R0	48	1	1	4	12	3
SUBB A, R1	49	1	1	4	12	3
SUBB A, R2	4A	1	1	4	12	3
SUBB A, R3	4B	1	1	4	12	3
SUBB A, R4	4C	1	1	4	12	3
SUBB A, R5	4D	1	1	4	12	3
SUBB A, R6	4E	1	1	4	12	3
SUBB A, R7	4F	1	1	4	12	3
SUBB A, @R0	46	1	1	4	12	3
SUBB A, @R1	47	1	1	4	12	3
SUBB A, direct	45	2	2	8	12	1.5
SUBB A, #data	44	2	2	8	12	1.5
INC A	04	1	1	4	12	3
INC R0	08	1	1	4	12	3
INC R1	09	1	1	4	12	3
INC R2	0A	1	1	4	12	3
INC R3	0B	1	1	4	12	3
INC R4	0C	1	1	4	12	3
INC R5	0D	1	1	4	12	3

Op-code	HEX Code	Bytes	CS6208 Machine Cycle	CS6208 Clock cycles	8051 Clock cycles	CS6208 vs.8051 Speed Ratio
INC R6	0E	1	1	4	12	3
INC R7	0F	1	1	4	12	3
INC @R0	06	1	1	4	12	3
INC @R1	07	1	1	4	12	3
INC direct	05	2	2	8	12	1.5
INC DPTR	A3	1	2	8	24	3
DEC A	14	1	1	4	12	3
DEC R0	18	1	1	4	12	3
DEC R1	19	1	1	4	12	3
DEC R2	1A	1	1	4	12	3
DEC R3	1B	1	1	4	12	3
DEC R4	1C	1	1	4	12	3
DEC R5	1D	1	1	4	12	3
DEC R6	1E	1	1	4	12	3
DEC R7	1F	1	1	4	12	3
DEC @R0	16	1	1	4	12	3
DEC @R1	17	1	1	4	12	3
DEC direct	15	2	2	8	12	1.5
DEC DPTR	A5	1	2	8	-	-
MUL AB	A4	1	5	20	48	2.4
DIV AB	84	1	5	20	48	2.4
DA A	D4	1	1	4	12	3
ANL A, R0	58	1	1	4	12	3
ANL A, R1	59	1	1	4	12	3
ANL A, R2	5A	1	1	4	12	3
ANL A, R3	5B	1	1	4	12	3
ANL A, R4	5C	1	1	4	12	3
ANL A, R5	5D	1	1	4	12	3
ANL A, R6	5E	1	1	4	12	3
ANL A, R7	5F	1	1	4	12	3
ANL A, @R0	56	1	1	4	12	3
ANL A, @R1	57	1	1	4	12	3
ANL A, direct	55	2	2	8	12	1.5
ANL A, #data	54	2	2	8	12	1.5
ANL direct, A	52	2	2	8	12	1.5
ANL direct, #data	53	3	3	12	24	2
ORL A, R0	48	1	1	4	12	3

Op-code	HEX Code	Bytes	CS6208 Machine Cycle	CS6208 Clock cycles	8051 Clock cycles	CS6208 vs.8051 Speed Ratio
ORL A, R1	49	1	1	4	12	3
ORL A, R2	4A	1	1	4	12	3
ORL A, R3	4B	1	1	4	12	3
ORL A, R4	4C	1	1	4	12	3
ORL A, R5	4D	1	1	4	12	3
ORL A, R6	4E	1	1	4	12	3
ORL A, R7	4F	1	1	4	12	3
ORL A, @R0	46	1	1	4	12	3
ORL A, @R1	47	1	1	4	12	3
ORL A, direct	45	2	2	8	12	1.5
ORL A, #data	44	2	2	8	12	1.5
ORL direct, A	42	2	2	8	12	1.5
ORL direct, #data	43	3	3	12	24	2
XRL A, R0	68	1	1	4	12	3
XRL A, R1	69	1	1	4	12	3
XRL A, R2	6A	1	1	4	12	3
XRL A, R3	6B	1	1	4	12	3
XRL A, R4	6C	1	1	4	12	3
XRL A, R5	6D	1	1	4	12	3
XRL A, R6	6E	1	1	4	12	3
XRL A, R7	6F	1	1	4	12	3
XRL A, @R0	66	1	1	4	12	3
XRL A, @R1	67	1	1	4	12	3
XRL A, direct	65	2	2	8	12	1.5
XRL A, #data	64	2	2	8	12	1.5
XRL direct, A	62	2	2	8	12	1.5
XRL direct, #data	63	3	3	12	24	2
CLR A	E4	1	1	4	12	3
CPL A	F4	1	1	4	12	3
RL A	23	1	1	4	12	3
RLC A	33	1	1	4	12	3
RR A	03	1	1	4	12	3
RRC A	13	1	1	4	12	3
SWAP A	C4	1	1	4	12	3
MOV A, R0	E8	1	1	4	12	3
MOV A, R1	E9	1	1	4	12	3
MOV A, R2	EA	1	1	4	12	3

Op-code	HEX Code	Bytes	CS6208 Machine Cycle	CS6208 Clock cycles	8051 Clock cycles	CS6208 vs.8051 Speed Ratio
MOV A, R3	EB	1	1	4	12	3
MOV A, R4	EC	1	1	4	12	3
MOV A, R5	ED	1	1	4	12	3
MOV A, R6	EE	1	1	4	12	3
MOV A, R7	EF	1	1	4	12	3
MOV A, @R0	E6	1	1	4	12	3
MOV A, @R1	E7	1	1	4	12	3
MOV A, direct	E5	2	2	8	12	1.5
MOV A, #data	74	2	2	8	12	1.5
MOV R0, A	F8	1	1	4	12	3
MOV R1, A	F9	1	1	4	12	3
MOV R2, A	FA	1	1	4	12	3
MOV R3, A	FB	1	1	4	12	3
MOV R4, A	FC	1	1	4	12	3
MOV R5, A	FD	1	1	4	12	3
MOV R6,A	FE	1	1	4	12	3
MOV R7, A	FF	1	1	4	12	3
MOV R0, direct	A8	2	2	8	12	1.5
MOV R1, direct	A9	2	2	8	12	1.5
MOV R2, direct	AA	2	2	8	12	1.5
MOV R3, direct	AB	2	2	8	12	1.5
MOV R4, direct	AC	2	2	8	12	1.5
MOV R5, direct	AD	2	2	8	12	1.5
MOV R6, direct	AE	2	2	8	12	1.5
MOV R7, direct	AF	2	2	8	12	1.5
MOV R0, #data	78	2	2	8	12	1.5
MOV R1, #data	79	2	2	8	12	1.5
MOV R2, #data	7A	2	2	8	12	1.5
MOV R3, #data	7B	2	2	8	12	1.5
MOV R4, #data	7C	2	2	8	12	1.5
MOV R5, #data	7D	2	2	8	12	1.5
MOV R6, #data	7E	2	2	8	12	1.5
MOV R7, #data	7F	2	2	8	12	1.5
MOV @R0, A	F6	1	1	4	12	3
MOV @R1, A	F7	1	1	4	12	3
MOV @R0, direct	A6	2	2	8	12	1.5
MOV @R1, direct	A7	2	2	8	12	1.5

Op-code	HEX Code	Bytes	CS6208 Machine Cycle	CS6208 Clock cycles	8051 Clock cycles	CS6208 vs.8051 Speed Ratio
MOV @R0, #data	76	2	2	8	12	1.5
MOV @R1, #data	77	2	2	8	12	1.5
MOV direct, A	F5	2	2	8	12	1.5
MOV direct, R0	88	2	2	8	12	1.5
MOV direct, R1	89	2	2	8	12	1.5
MOV direct, R2	8A	2	2	8	12	1.5
MOV direct, R3	8B	2	2	8	12	1.5
MOV direct, R4	8C	2	2	8	12	1.5
MOV direct, R5	8D	2	2	8	12	1.5
MOV direct, R6	8E	2	2	8	12	1.5
MOV direct, R7	8F	2	2	8	12	1.5
MOV direct, @R0	86	2	2	8	12	1.5
MOV direct, @R1	87	2	2	8	12	1.5
MOV direct, direct	85	3	3	12	24	2
MOV direct, #data	75	3	3	12	24	2
MOV DPTR, #data 16	90	3	3	12	24	2
MOVC A, @A+DPTR	93	1	2	8	24	3
MOVC A, @A+PC	83	1	2	8	24	3
MOVX A, @R0	E2	1	2 - 9	8 - 36	24	3 - 0.66
MOVX A, @R1	E3	1	2 - 9	8 - 36	24	3 - 0.66
MOVX A, @DPTR	E0	1	2 - 9	8 - 36	24	3 - 0.66
MOVX @R0, A	F2	1	2 - 9	8 - 36	24	3 - 0.66
MOVX @R1, A	F3	1	2 - 9	8 - 36	24	3 - 0.66
MOVX @DPTR, A	F0	1	2 - 9	8 - 36	24	3 - 0.66
PUSH direct	C0	2	2	8	24	3
POP direct	D0	2	2	8	24	3
XCH A, R0	C8	1	1	4	12	3
XCH A, R1	C9	1	1	4	12	3
XCH A, R2	CA	1	1	4	12	3
XCH A, R3	CB	1	1	4	12	3
XCH A, R4	CC	1	1	4	12	3
XCH A, R5	CD	1	1	4	12	3
XCH A, R6	CE	1	1	4	12	3
XCH A, R7	CF	1	1	4	12	3
XCH A, @R0	C6	1	1	4	12	3
XCH A, @R1	C7	1	1	4	12	3
XCHD A, @R0	D6	1	1	4	12	3

Op-code	HEX Code	Bytes	CS6208 Machine Cycle	CS6208 Clock cycles	8051 Clock cycles	CS6208 vs.8051 Speed Ratio
XCHD A, @R1	D7	1	1	4	12	3
XCH A, direct	C5	2	2	8	12	1.5
CLR C	C3	1	1	4	12	3
CLR bit	C2	2	2	8	12	1.5
SETB C	D3	1	1	4	12	3
SETB bit	D2	2	2	8	12	1.5
CPL C	B3	1	1	4	12	3
CPL bit	B2	2	2	8	12	1.5
ANL C, bit	82	2	2	8	24	3
ANL C, /bit	B0	2	2	8	24	3
ORL C, bit	72	2	2	8	24	3
ORL C, /bit	A0	2	2	8	24	3
MOV C, bit	A2	2	2	8	12	1.5
MOV bit, C	92	2	2	8	24	3
ACALL addr11	11,31,51, 71,91,B1, D1,F1	2	3	12	24	2
LCALL addr16	12	3	4	16	24	1.5
RET	22	1	2	8	24	3
RETI	32	1	2	8	24	3
AJMP ADDR11	01,21,41, 61,81, A1, C1,E1	2	3	12	24	2
LJMP addr16	02	3	4	16	24	1.5
JMP @A+DPTR	73	1	2	8	24	3
SJMP rel	80	2	3	12	24	2
JZ rel	60	2	3	12	24	2
JNZ rel	70	2	3	12	24	2
JC rel	40	2	3	12	24	2
JNC rel	50	2	3	12	24	2
JB bit, rel	20	3	4	16	24	1.5
JNB bit, rel	30	3	4	16	24	1.5
JBC bit, rel	10	3	4	16	24	1.5
CJNE A, direct, rel	B5	3	4	16	24	1.5
CJNE A, #data, rel	B4	3	4	16	24	1.5
CJNE @R0, #data, rel	B6	3	4	16	24	1.5
CJNE @R1, #data, rel	B7	3	4	16	24	1.5
CJNE R0, #data, rel	B8	3	4	16	24	1.5

Op-code	HEX Code	Bytes	CS6208 Machine Cycle	CS6208 Clock cycles	8051 Clock cycles	CS6208 vs.8051 Speed Ratio
CJNE R1, #data, rel	B9	3	4	16	24	1.5
CJNE R2, #data, rel	BA	3	4	16	24	1.5
CJNE R3, #data, rel	BB	3	4	16	24	1.5
CJNE R4, #data, rel	BC	3	4	16	24	1.5
CJNE R5, #data, rel	BD	3	4	16	24	1.5
CJNE R6, #data, rel	BE	3	4	16	24	1.5
CJNE R7, #data, rel	BF	3	4	16	24	1.5
DJNZ R0, rel	D8	2	3	12	24	2
DJNZ R1, rel	D9	2	3	12	24	2
DJNZ R2, rel	DA	2	3	12	24	2
DJNZ R3, rel	DB	2	3	12	24	2
DJNZ R4, rel	DC	2	3	12	24	2
DJNZ R5, rel	DD	2	3	12	24	2
DJNZ R6, rel	DE	2	3	12	24	2
DJNZ R7, rel	DF	2	3	12	24	2
DJNZ direct, rel	D5	3	4	16	24	1.5

ADDRESSING MODES

The CS6208 uses eight different addressing mode. These modes are as follows:

1) Register addressing
2) Direct addressing
3) Register Indirect addressing
4) Immediate addressing
5) Base register plus Index register Indirect addressing
6) Relative addressing
7) Absolute addressing
8) Long addressing

Register Addressing:

Register addressing uses the eight working registers (R0-R7) in the scratchpad RAM. The selected register is indicated by the last three bits in the instruction op-code. There are four register banks that reside in the scratchpad RAM. The two bits RS1-RS0 in the Program Status Word (PSW) are used to select the working bank. Thus the same instruction can access different registers simply by switching between banks. ACC, B, DPTR and CY can also be addressed as registers.

Example:

ADDC	A, R2	Add Accumulator to register R2 with carry.
DEC	R7	Decrement contents of register R7
CLR	A	Clear the Accumulator
SETB	C	Set the Carry flag (Boolean Processor Accumulator)

Direct Addressing:

Direct addressing is the only method of addressing the Special Function Registers (SFRs). This mode of addressing can also be used to access the entire lower 128 bytes of the Scratchpad RAM.

Direct addressing also applies to bit operations. The Scratchpad RAM area from 20h to 2Fh is bit-addressable. The bits in this area have their own unique bit addresses from 00h to 7Fh. These bits may be addressed by directly specifying the bit address. There are several SFRs which are also bit-addressable. The bits in these SFRs are addressed by adding the bit position to the SFR address.

Example:

MOV	20h, 21h	Move contents of RAM location 20h to RAM location 21h
MOV	P1, P3	Move data at Port 1 Pins to Port 3 latch
SETB	7Fh	Set bit 7Fh in Scratchpad RAM. This is actually MSB of the RAM location 2Fh.
MOV	TR1, C	Move carry flag contents to TR1 flag in TCON SFR

Register Indirect Addressing:

Register Indirect addressing mode uses the R0 and R1 registers to store the address of the selected Scratchpad RAM location. By changing the contents or R0 or R1, the same instruction will address different RAM locations. However, this mode of addressing is limited in the CS6208 to the 128 bytes of Scratchpad RAM only. Special Function Registers (SFRs) cannot be addressed by this method.

Example:

MOV	R0, #20h	Load register R0 with 20h
MOV	A, @R0	Move contents of RAM location 20h to the Accumulator

Stack operations are also examples of Register Indirect addressing. In this case the SFR Stack Pointer (SP) is used to store the address rather than the working registers R0 and R1. Instructions like PUSH and POP use this mode of addressing.

Example:

PUSH	TCON	Save the contents of SFR TCON onto the stack
POP	DPL	Loads the SFR DPL with the data at the top of the stack

Off-chip Data Memory may also be accessed using Register Indirect Addressing mode. The MOVX instructions are used for this purpose. The registers R0, R1, DPTR or DPTR1 can be used as pointers for this purpose. When using R0 and R1 as pointers, the lower 8-bits of the address are the contents of R0 or R1, while the upper 8-bits of the address are supplied by the Port 2 SFR. The Port 2 SFR must therefore be set to the correct value before using this instruction. When using DPTR or DPTR1 as the pointer, then the entire 16-bit address is taken from the selected Data Pointer.

Example:

MOVX	@R0, A	Moves the Accumulator contents to the external Data Memory locations pointed to by R0 in the page pointed to by P2.
MOVX	A, @DPTR	Move data from the Data Memory pointed to by the selected Data Pointer, into the Accumulator.

Immediate Addressing:

Immediate addressing is used when part of the op-code instruction is a constant. This is used most commonly to initialize registers and SFRs or to do mask operations.

Example:

MOV	A, #40h	Mov 40h to Accumulator
XRL	SCON, #FFh	Xor SCON SFR with FFh, thereby complementing its contents

Base Register plus Index Register Indirect Addressing:

This mode allows access to a byte from the Program Memory via an indirect move from a location whose address is the sum of the base register (PC or DPTR/DPTR1) and an index register (ACC).

The result is always written to the ACC, overwriting the index value that existed before.

Example:

MOV	DPTR, #1234h	Data Pointer is loaded with constant 1234h
MOV	A, #23h	Accumulator is loaded with index value 23h
MOVC	A, @A+DPTR	Accumulator is loaded with byte from the Program Memory at location 1234h + 23h = 1257h

Relative Addressing:

Relative Addressing is used to specify the destination address in some of the jump instructions. The destination address is given in the form of a two's complement address offset which is added to the PC. The relative address is one byte long, and so the offset will vary from -127 to +128.

Example:

SJMP	20h	Jump to a location 20h bytes after the current instruction. If the SJMP instruction is at 1010h, then the jump will be to location 1010h + 2h + 20h = 1032h
JZ	FEh	If the Accumulator is zero, then jump to the same instruction If the JZ instruction is at 2359h, then the jump will be to location

2359h + 2h + FEh = 2359h + 2h - 2h = 2359h

Absolute Addressing:

Absolute addressing is used to specify the destination address in ACALL and AJMP instructions. In this mode of addressing, the 16-bit address ids generated by taking the five highest order bits of the next instruction (PC + 2) and concatenating the lowest order 11 bit field contained in the current instruction. The addressed location will lie within the 2K page of the Program Memory as the first byte of the instruction following the current instruction.

Example:

ACALL	0200h	If the current instruction is at location 7FFFh, then a call will be made to the location 8200h

Long Addressing:

Long addressing is used to specify the destination address in LCALL and LJMP instructions. Here the complete 16-bit destination address is specified in the instruction itself. Therefore the jump can be to any location within the 64K of Program memory.

INSTRUCTION TIMING:

The instruction timing for the CS6208 is an important aspect, especially for those users who wish to use software instructions to generate timing delays. Also it provides the user an insight into the timing differences between the CS6208 and the standard 8051. In the CS6208 each machine cycle is four clock periods long. Each clock period is designated a state. Thus each machine cycle is made of four states C1, C2 C3 and C4 in that order. Due to the reduced time for each instruction execution, the CS6208 uses both the clock edged for internal timing. Hence it is important that the duty cycle of the clock be as close to 50% as possible to avoid timing conflicts.

As mentioned earlier, the CS6208 does one op-code fetch per machine cycle. Therefore, in most of the instructions the number of machine cycles needed to execute the instruction is equal to the number of bytes in the instruction.

Of the 256 available op-codes, 128 of them are single cycle instructions.Thus more than half of all op-codes in the CS6208 are executed in just four clock periods. Most of the two cycle instructions are those that have two byte instruction codes. However there are some instructions that have only one byte instructions, yet they are two cycle instructions. One instruction which is of importance is the MOVX instruction. In the standard 8051, the MOVX instruction is always two machine cycles long. However in the CS6208, the user has a facility to stretch the duration of this instruction from 2 machine cycles to 9 machine cycles. The RD and WR strobe lines are also proportionately elongated. This gives the user the flexibility in accessing both fast and slow peripherals without the use of external circuitry and with minimum software overhead. The reset of the instructions are either three, four or five machine cycle instructions. Note that in the CS6208, based on the number of machine cycles, there are five different types, while in the standard 8051 there are only three. However, in the CS6208 each machine cycle is made of only 4 clock periods compared to the 12 clock periods for the standard 8051. Therefore, even though the number of categories has increased, each instruction is at least 1.5 to 3 times faster than the standard 8051 in terms of clock periods.

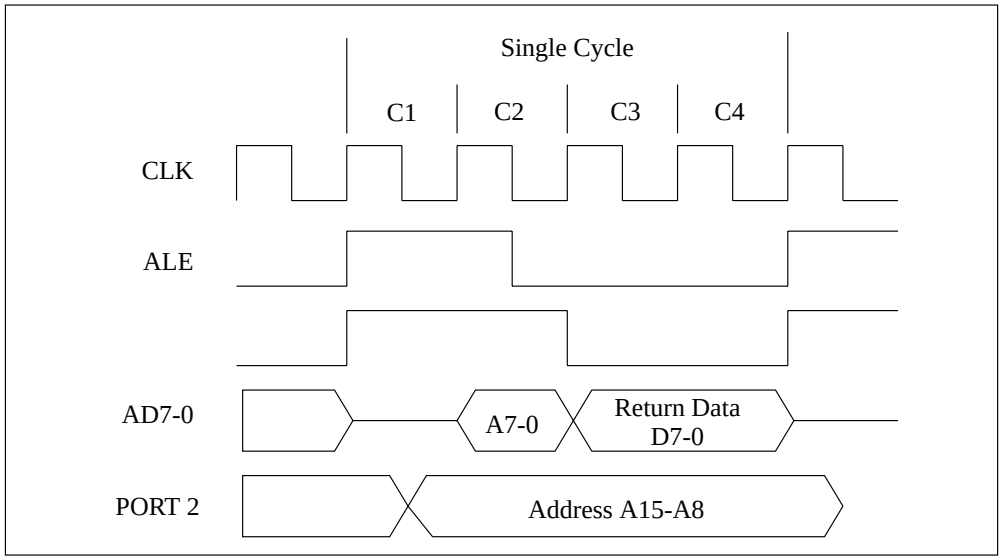


Figure-5 Single Cycle Instruction Timing

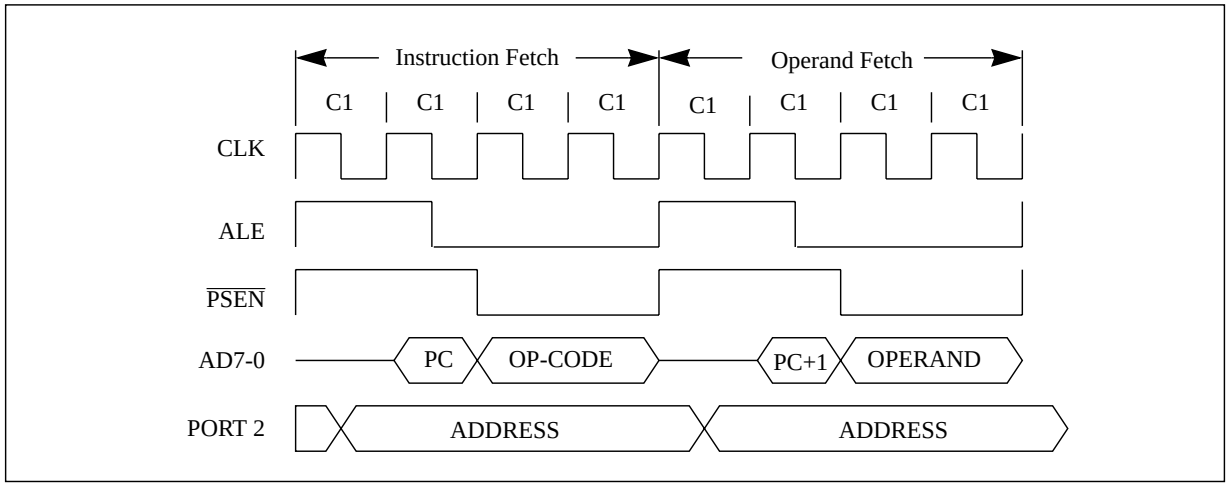


Figure-6 Two Cycle Instruction Timing

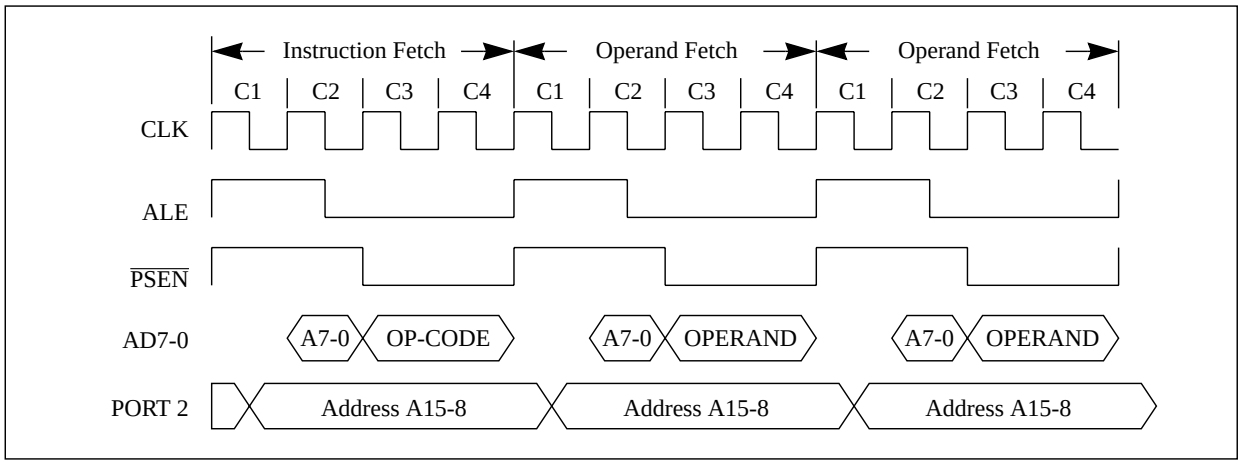


Figure-7 Three Cycle Instruction Timing

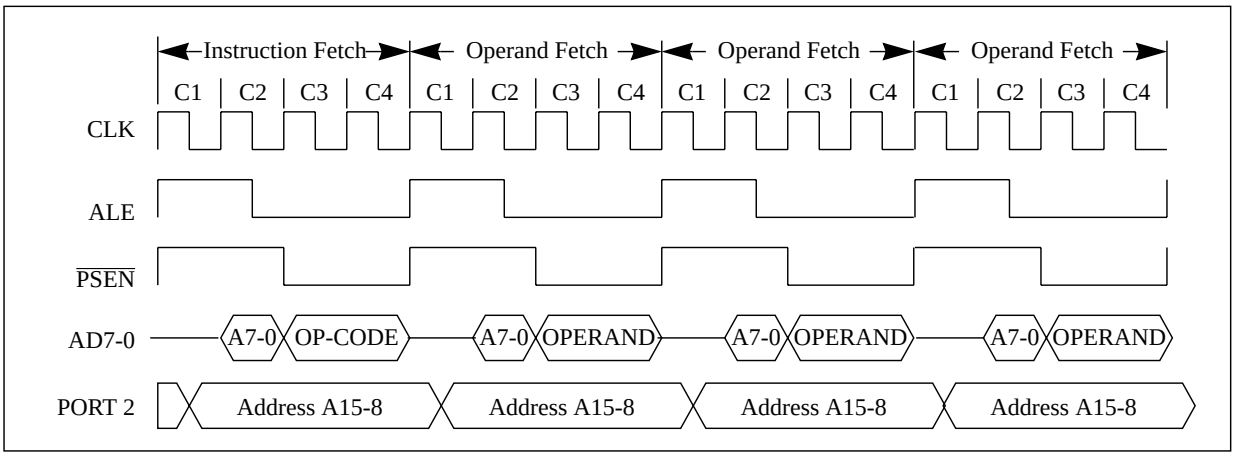


Figure-8 Four Cycle Instruction Timing

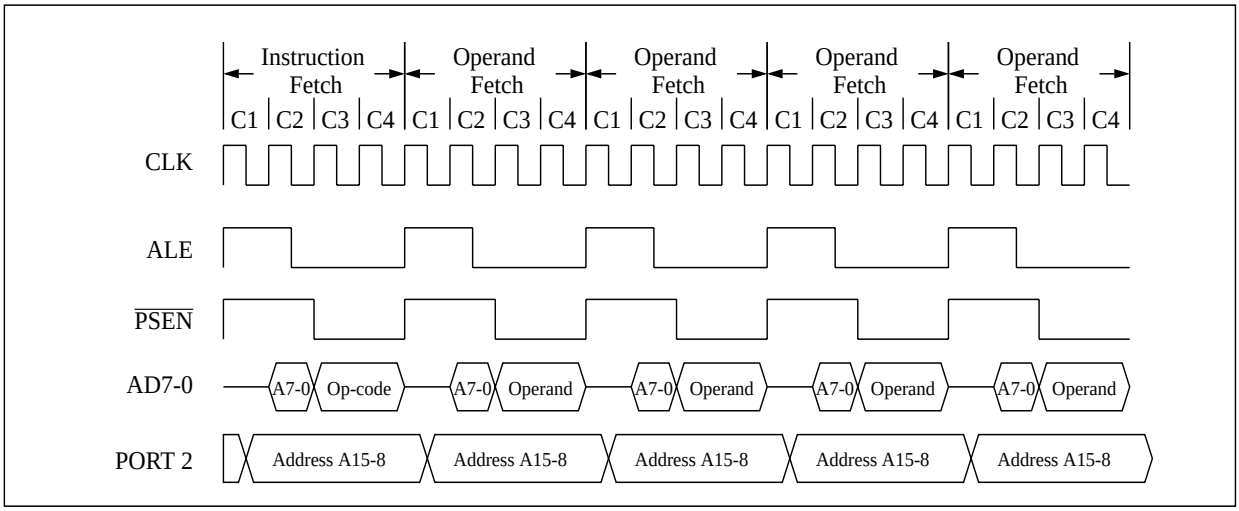


Figure-9 Five Cycle Instruction Timing

MOVX Instruction:

The CS6208, like the standard 8051 uses the MOVX instruction to access external Data Memory. This Data Memory include both off-chip memory as well as memory mapped peripherals. While the results of the MOVX instruction is the same as in the standard 8051, the operation and the timing of the strobe signals have been modified in order to give the user a mush greater flexibility.

The MOVX instruction is of two types, the MOVX @Ri and MOVX @DPTR. In the MOVX @Ri, the address of the external data comes from two sources. The lower 8-bits of the address are stored in the Ri register of the selected working register bank. The upper 8-bits of the address come from the port 2 SFR. In the MOVX @DPTR type, the full 16-bit address is supplied by the Data Pointer.

Since the CS6208 has two Data Pointers, DPTR and DPTR1, the user has to select between the two by setting or clearing the DPS bit. The Data Pointer Select bit (DPS) is the LSB of the DPS SFR which exists at location 86h. No other bits in this SFR have any effect and are set to 0. When DPS is 0, then DPTR is selected, and when set to 1, DPTR1 is slected.The user can switch between DPTR and DPTR1 by toggling the DPS bit. The quickest way to do this is by the INC or DEC instruction.

The advantage of having two Data Pointers is most obvious while performing block move operations. The accompanying code shows how the use of two separate Data Pointers speeds up the execution time for code performing the same task.

Block Move with single Data Pointer:

```
; SH and SL are the high and low bytes of Source Address
; DH and DL are the high and low bytes of Destination Address
; CNT is the number of bytes to be moved

MOV     R2, #CNT        Load R2 with the count value                    2   2
MOV     R3, #DL         Save low byte of Source Address in R3           2   2
MOV     R4, #DH         Save high byte of Source address in R4          2   2
MOV     R5, #DL         Save low byte of Destination Address in R5      2   2
MOV     R6, #DH         Save high byte of Destination address in R6     2   2
```

```
LOOP:
MOV     DPL, R3         Load DPL with low byte of Source address          2   2
MOV     DPH, R4         Load DPH with high byte of Source address         2   2
MOVX    A, @DPTR        Get byte from Source to Accumulator               2   2
INC     DPTR            Increment Source Address to next byte             2   2
MOV     R3, DPL         Save low byte of Source address in R3             2   2
MOV     R4, DPH         Save high byte of Source Address in R4            2   2
MOV     DPL, R5         Load low byte of Destination Address in DPL       2   2
MOV     DPH, R6         Load high byte of Destination Address in DPH      2   2
MOVX    @DPTR, A        Write data to destination                         2   2
INC     DPTR            Increment Destination Address                     2   2
MOV     DPL, R5         Save low byte of new destination address in R5    2   2
MOV     DPH, R6         Save high byte of new destination address in R6   2   2
DJNZ    R2, LOOP        Decrement count and do LOOP again if count <> 0   2   3
```

Machine cycles in standard 8051 = 10 + (26 * CNT)
Machine cycles in CS6208 = 10 + (26 * CNT)
CNT = 50d
Clock cycles in standard 8051= ((10 + (26 *50)) * 12 = (10 + 1300) * 12 = 15720
Clock cycles in CS6208 = ((10 + (26 * 50)) * 4 = (10 + 1300) * 4 = 5240

Block Move with Two Data Pointers in CS6208:

SH and SL are the high and low bytes of Source Address
DH and DL are the high and low bytes of Destination Address
CNT is the number of bytes to be moved

```
MOV     R2, #CNT        Load R2 with the count value                      2
MOV     DPS, #00h       Clear DPS to point to DPTR                        2
MOV     DPTR, #DHDL     Load DPTR with Destination address                3
INC     DPS             Set DPS to point to DPTR1                         2
MOV     DPTR, #SHSL     Load DPTR1 with Source address                    3
LOOP:
MOVX    A, @DPTR        Get data from Source block                        2
INC     DPTR            Increment source address                          2
DEC     DPS             Clear DPS to point to DPTR                        2
MOVX    @DPTR, A        Write data to Destination                         2
INC     DPTR            Increment destination address                     2
INC     DPS             Set DPS to point to DPTR1                         2
DJNZ    R2, LOOP        Check if all done                                 3
```

Machine cycles in CS6208 = 12 + (15 * CNT)
CNT = 50d
Clock cycles in CS6208 = (12 + (15 * 50)) * 4 = (12 + 750) * 4 = 3048

We can see that in the first program the standard 8051 takes 15720 cycles, while the CS6208 takes only 5240 cycles for the same code. In the second program, written for the CS6208, program execution requires only 3048 clock cycles. If the size of the block is increased then the saving is even greater.

Data Memory Timing:

The timing for the MOVX instruction is another feature of the CS6208. In the standard 8051, the MOVX instruction has a fixed execution time of 2 machine cycles. However in the CS6208, the duration of the access can be varied by the user.

The instruction starts off as a normal op-code fetch of 4 clocks. In the next machine cycle, the CS6208 puts out the address of the external Data Memory and the actual access occurs here. The user can change the duration of this access time by setting the STRETCH value. The Clock Control SFR (CKCON) has three bits that control the stretch value. These three bits are M2-0 (bits 2-0 of CKCON). These three bits give the user 8 different access time options. The stretch can be varied from 0 to 7, resulting in MOVX instructions that last from 2 to 9 machine cycles in length. Note that the stretching of the instruction only results in the elongation of the MOVX instruction, as if the state of the CPU was held for the desired period. There is no effect on any other instruction or its timing. By default, the Stretch value is set at 1 giving a MOVX instruction of 3 machine cycles.

If desired by the user the stretch value can be set to 0 to give the fastest MOVX instruction of only 2 machine cycles

Table-4 Data Memory Cycle Stretch Values

M2	M1	M0	Machine Cycles	$\overline{RD}$ or $\overline{WR}$ strobe width in Clocks	$\overline{RD}$ or $\overline{WR}$ strobe width in t @25MHz	$\overline{RD}$ or $\overline{WR}$ strobe width in t @40MHz
0	0	0	2	2	80 ns	50 ns
0	0	1	3(default)	4	160 ns	100 ns
0	1	0	4	8	320 ns	200 ns
0	1	1	5	12	480 ns	300 ns
1	0	0	6	16	640 ns	400 ns
1	0	1	7	20	800 ns	500 ns
1	1	0	8	24	960 ns	600 ns
1	1	1	9	28	1120 ns	700 ns

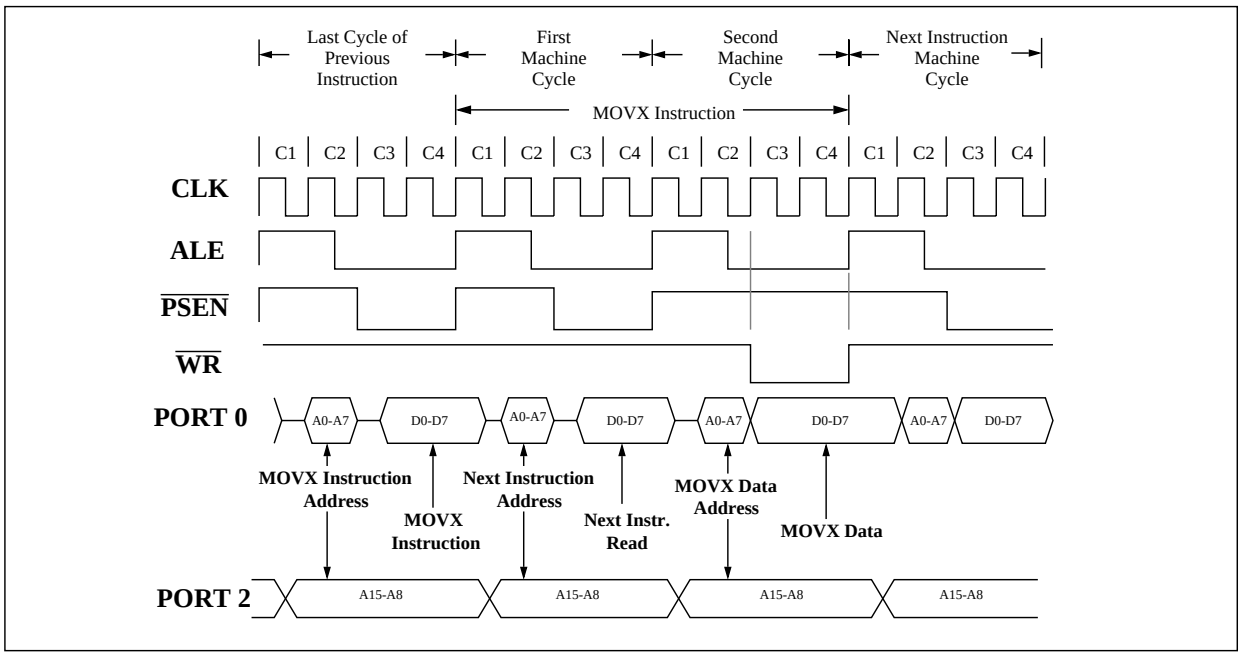


Figure-10 DATA MEMORY WRITE WITH STRETCH = 0

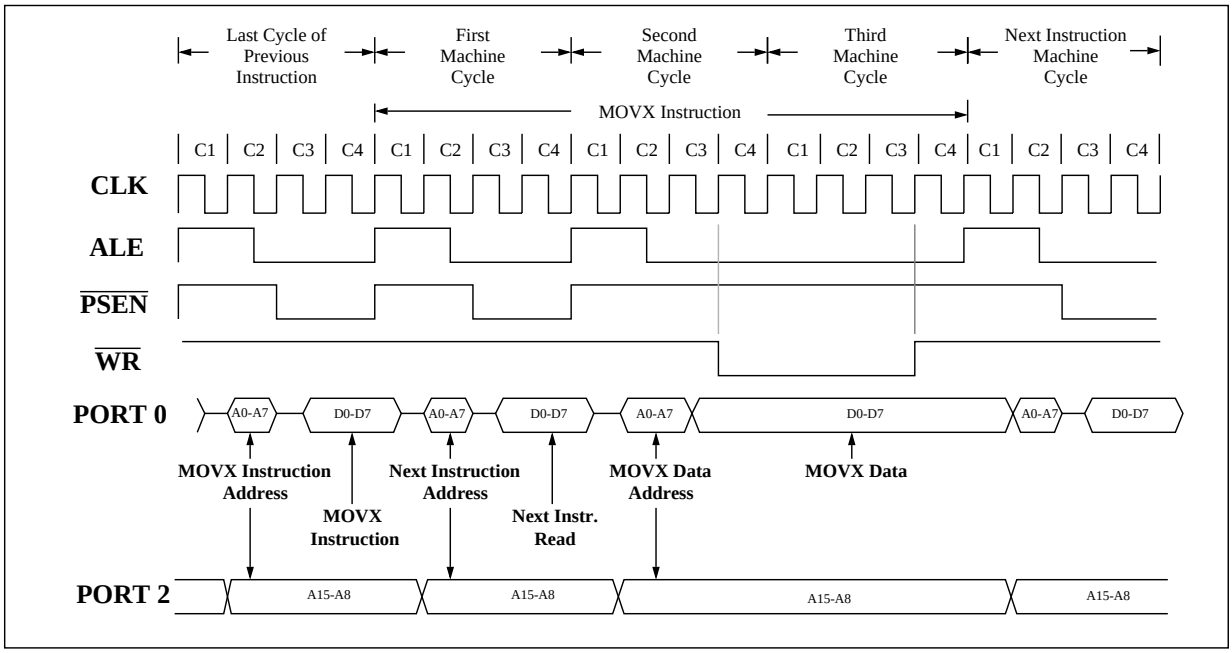


Figure-11 DATA MEMORY WRITE WITH STRETCH = 1

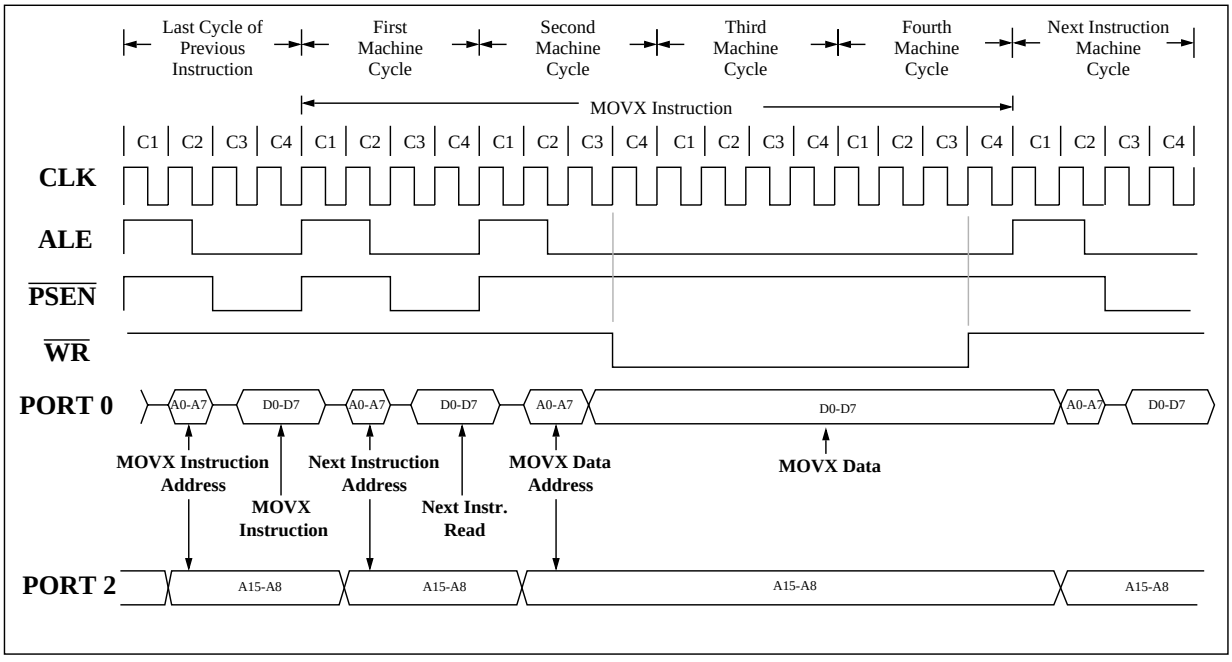


Figure-12 DATA MEMORY WRITE WITH STRETCH = 2

PORT STRUCTURES AND OPERATION

The four ports in the CS6208 are all bidirectional. Each port consists of two sections, the port SFR and the I/O pad.

The Ports 0 and 2 are involved in accesses to external memory. In this case, Port 0 outputs the lower byte of the external memory address while port 2 outputs the higher byte of the external address. The Port 0 bus is also used as a data bus for the data byte that is read or is to be written.

Therefore Port 0 is actually a time-multiplexed address/data bus. A point to note is that Port 2 outputs the upper 8 bits of the address only if the address is 16 bits wide, else it continues to emit the Port 2 SFR contents.

Ports 0 is used in the CS6208 as a multiplexed address/data bus. These pins have strong drivers. and will provide true drive capabilities for both logic levels. No pull-ups are needed. In fact pull-ups will degrade the CS6208 interface timing. In the CS6208, the port0 is permanently used for address/data bus purposes and cannot be used in any circumstances as an I/O port. Hence the Port0 SFR does not actually exist. Reading the port0 latch will return FFh.

Port 2 has the ability to serve as an I/O port or the MSB of an address bus. When serving as an address bus, port 2 will be driven with strong drivers at all times. When serving as an I/O port the drive will vary. For a logic 0, a strong pull-down is invoked. For logic 1, the port will invoke a strong pull-up for two machine cycles if the port pin has to make a transition from 0 to 1. After the transition time, a weak pull-up is invoked to maintain 1 on the pin. The weak 1 can easily be over-driven by an external logic source.

Port 1 has two pins which have alternate input functions and the other six pins are true input/output ports, having no alternate functions or multiplexed busses associated with it. The drive characteristics for port 1 are similar to port 2 in non-bus mode. That is logic 0 is created by a strong pull-down. The logic 1 is created by a strong transition pull-up followed by a weak pull-up. The alternate functions of port 1 are as follows.

Port Pin	Alternate Function
P1.0	T2 Timer/Counter external input
P1.1	T2EX Timer/Counter 2 Capture/Reload Trigger

All the port pins in Port 3 are multifunctional, and have alternate input or output functions. The drive characteristics are similar to port 1. The alternate functions of port 3 are listed as follows

Port Pin	Alternate Function
P3.0	RXD serial input
P3.1	TXD serial output
P3.2	INT0 external interrupt input
P3.3	INT1 external interrupt input
P3.4	T0 Timer/Counter external input
P3.5	T1 Timer/Counter external input
P3.6	WR external Data Memory write strobe
P3.7	RD external Data Memory read strobe

In order that the alternate functions are activated correctly, the corresponding bit latch must be held at value 1. If this is not done then the corresponding port pin is stuck at 0, and external or internal inputs will have no effect on the pin value.

As port 1, 2 and 3 have internal pull-ups, they will go high when configured as inputs and will source current. Hence they are also known as "quasi-bidirectional" ports. Port 0 on the other hand "floats" when configured as an input and hence is called a "true bidirectional" port.

Read-Modify-Write Feature

Each port is split into its SFR and its corresponding I/O pad. Therefore there are two options available for a port

read access. Either the SFR latch contents can be read or the input from the I/O pads can be read. The instructions that read the latch, modify the value and write it back to the latch are called "read-modify-write" instruction. In such instructions the latch and not the pin is read. The instruction of this category are listed as follows

ANL	logical AND
ORL	logical OR
XRL	logical XOR
JBC	jump if bit = 1 and then clear bit
CPL	complement bit
INC	increment
DEC	decrement
DJNZ	decrement and jump if not zero
MOV PX.Y, C	move carry bit to bit Y of Port X
CLR PX.Y	clear bit Y of port X
SETB PX.Y	set bit Y of X

ACCESSING EXTERNAL MEMORY

External Memory is accessed if either of the following two conditions is met

1) The signal $\overline{EA}$ is low

2) Whenever the program counter (PC) contains an address greater than 0FFFh.

Accesses to external memory are of two type: External Program Memory accesses and External Data Memory accesses. External Program Memory is accessed using the $\overline{PSEN}$ signal as the read strobe. External Data Memory is accessed using the $\overline{RD}$ or $\overline{WR}$ pins to strobe the memory.

Fetches from the external Program Memory always use a 16 bit address, while External Data Memory accesses can have addresses of 8 bit or 16 bit. Whenever a 16 bit address is used, Port 2 is used to emit the higher byte of the address. The point to note is that during external accesses, port 2 uses strong pull-ups while emitting 1s. During the time Port 2 does not emit the higher address byte, it continuously outputs the Port 2 SFR contents, which are not modified by the hard-ware (unless of course the user writes to the Port 2 SFR). If an 8 bit address is being used, then the Port 2 SFR contents will be outputted on the port pins throughout the external memory cycle.

The lower byte of the address is always multiplexed with the data byte on Port 0. The ADDR/DATA bus can drive both the active pull-up and pull-down transistors. Thus for external memory accesses, the port 0 pins are not open-drain outputs and do not need any external pull-ups. The falling edge of the ALE signal can be used to store the address on Ports 0 and 2 in an external address latch. During a write cycle, the data to be written to the external Data Memory is present on Port 0 pins and remains there until after $\overline{WR}$ is deactivated. In case of a read access, the data on Port 0 pins is read just before the deactivation of the $\overline{RD}$ signal.

During External Memory Accesses, both Ports 0 and 2 are used for Address/ Data transfer and therefore cannot be used for general I/O purposes. During external program fetches, Port 2 uses strong pull-ups to emit 1s.

POWER MANAGEMENT:

The CS6208 has several features that help the user to modify the power consumption of the device and to preclude loss of control due to power-failure. The power management features basically relate to power saving and power control. The power management features consist of POWER ON/POWER FAIL RESET and POWER FAIL WARNING INTERRUPT. The power saving features are basically the POWER DOWN mode and the IDLE mode of operation.

POWER MANAGMENT

The CS6208 has an in-built power-on reset system. This ensures that if the power levels are below the Vrst level, the device will be forced into the reset state. So when power is turned on, i.e. a cold reset, the device will go into reset condition automatically and stay there as long as Vcc is less that Vrst. Similarly, when the power falls below Vrst, the device will automatically go into reset state. This obviates the need for any external power monitor hardware.

In addition the CS6208 has an early power failure warning feature. If the power level falls below Vpfw the on-chip hardware will detect it and will set the PFI power fail interrupt flag. If the software has enabled this interrupt by setting the EPFI bit in WDCON.5, then the device will be interrupted and will go to vector address 33h. The user can write his own power fail routine to save critical data and set the port values to any user defined values. The PFI flag is set by hardware by can only be cleared by software. This flag will continue to be set as long as the power levels are below Vpfw. When the power returns to the proper level, the PFI flag, once cleared, will no longer be set. If however the power level continues to fall and goes below Vrst, then a Power fail reset will occur.

POWER SAVING

The CS6208 has two modes of operation the Power Down mode and the Idle mode. These modes are possible because the CS6208 is implemented in fully static CMOS logic. This permits the device to reduce the clock speed to DC.

Idle Mode

The user can put the device into idle mode by writing 1 to the bit PCON.0. The instruction that sets the idle bit is the last instruction that will be executed before the device goes into Idle Mode. In the Idle mode, the clock to the CPU is halted, but not to the Interrupt, Timer, Watchdog timer and Serial port blocks. This forces the CPU state to be frozen; the Program counter, the Stack Pointer, the Program Status Word, the Accumulator and the other registers hold their contents. The ALE and PSEN pins are held high during the Idle state. The port pins hold the logical states they had at the time Idle was activated.

The Idle mode can be terminated in two ways. Since the interrupt controller is still active, the activation of any enabled interrupt can wake up the processor. This will automatically clear the Idle bit, terminate the Idle mode, and the Interrupt service routine will be executed. After the ISR, execution of the program will continue from the instruction which put the device into Idle mode. The Idle mode can also be exited by activating the reset. The device can be put into reset either by applying a high on the external RST pin, a Power on/fail reset condition or a Watchdog timer reset. The external reset pin has to be held high for at least two machine cycles i.e. 8 clock periods to be recognized as a valid reset. In the reset condition the program counter is reset to 0000h and all the SFRs are set to the reset condition. Since the clock is already running there is no delay and execution starts immediately. In the Idle mode, the Watchdog timer continues to run, and if enabled, a time-out will cause a watchdog timer interrupt which will wake up the device. The soft-ware must reset the Watchdog timer in order to preempt the reset which will occur after 512 clock periods of the time-out. When the CS6208 when exiting from an Idle mode with a reset, the instruction following the one which put the device into Idle mode is not executed. So there is no danger of unexpected writes.

Power Down Mode

The device can be put into Power Down mode by writing 1 to bit PCON.1. The instruction that does this will be the last instruction to be executed before the device goes into Power Down mode. In the Power down mode, all the clocks are stopped and the device comes to a halt. All activity is completely stopped and the power consumption is reduced to the lowest possible value. In this state the ALE and PSEN pins are pulled low. The port pins output the values held by their respective SFRs.

The CS6208 will exit the Power Down mode with a reset or by an external interrupt pin enabled as low level detect. An external reset can be used to exit the Power down state. The high on RST pin terminates the Power Down mode, and restarts the clock. The on-chip hardware will now provide a delay of 65536 clock which is used to provide time for the oscillator to restart and stabilize. Once this delay is complete an internal reset is activated and the program execution will restart from 0000h. In the Power down mode, the clock is stopped, so the Watchdog timer cannot be used to provide the reset to exit Power down mode. The Power on/fail reset can however be used to provide a reset in case the power falls below the threshold level of Vrst.

The CS6208 can be woken from the Power Down mode by forcing an external interrupt pin low, provided the corresponding interrupt is enabled, the global enable is set and the external input has been set to a low level detect mode. If these conditions are met, then the low on the external pin re-enables the on-chip oscillator. The external pin should be held low long enough for the oscillator to start and stabilize. The Power down mode is finally exited when the external interrupt is released and returns to 1. The device now executes an internal interrupt for the corresponding external interrupt. After the interrupt service routine is completed, the program execution returns to the instruction after the one which put the device into Power Down mode and continues from there.

CS6208 power down and wake up procedure

To enter power down mode:

1) Enable external interrupt requests at high priority.

2) Set the power down bit to 1.

3) The CS6208 will disable the CPU core, internal RAM, internal ROM, MAC, PHY, and I^2C modules automatically.

4) Turn off the

Table-5 Status of external pins during Idle and Power Down

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

RESET CONDITIONS

The user has several hardware related options of placing the CS6208 into reset condition. In general, most register bits go to their reset condition irrespective of the reset condition, but there are a few flags whose state depends on the source of reset. The user can use these flags to determine the cause of reset using software.

There are three ways of putting the device into reset state. They are

External reset
Power on/fail reset
Watchdog reset

External Reset

The device continuously samples the RST pin at state C4 of every machine cycle. Therefore the RST pin must be held for at least 2 machine cycles to ensure detection of a valid RST high. the reset circuitry then synchronously applies the internal reset signal. Thus the reset is a synchronous operation and requires the clock to be running to cause an external reset.

Once the device is in reset condition, it will remain so as long as RST is 1. Even after RST is deactivated, the device will continue to be in reset state for up to two machine cycles, and then begin program execution from 0000h. There is no flag associated with the external reset condition. However since the other two reset sources have flags, the external reset can be considered as the default reset if those two flags are cleared.

Power on/fail Reset

The on-chip analog circuitry continuously monitors the Vcc levels and are used to detect a power on or power fail condition. As the Vcc level is below the threshold Vrst, the device will remain in the reset condition. Once the Vcc goes above the Vrst level, the analog circuits release the reset. However internal hardware will maintain this internal reset condition for 65536 clocks. This is the power-on reset delay period which is provided to allow the crystal oscillator time to start-up and stabilize.

After the delay of 655356 clocks, the hardware will automatically release the internal reset and program execution will begin from 0000h. The power on reset will set the POR flag in WDCON.6. This helps the software in determining the cause of reset.

The software must clear the POR flag after reading it, else it will not be possible to correctly determine future reset sources. If the power fails i.e. falls below Vrst, then the device will once again go into reset state. when the power return to the proper operating levels, the device will again perform a power on reset delay and set the POR flag.

Watchdog Timer Reset

The Watchdog timer is a free running timer with programmable time-out intervals. The user can clear the watchdog timer at any time causing it to restart the count. When the time-out interval is reached an interrupt flag is set. If the Watchdog reset is enabled and the watchdog timer is not reset, then 512 clocks from the flag being set the watchdog timer reset is activated. This places the device into the reset condition. The reset condition is maintained by hardware for two machine cycles. Once the reset is removed the device will begin execution from 0000h.

RESET STATE

Most of the SFRs and registers on the device will go to the same condition in the reset state. The Program Counter is forced to 0000h and is held there as long as the reset condition is applied. The reset state however does not affect the on-chip RAM. The data in the RAM will be preserved during the reset. However the stack pointer is reset to 07h, and therefore the stack contents will be lost. However, the RAM contents will be lost if the Vcc falls below approximately 2V, as this is the minimum voltage level required for the RAM to operate normally. Therefore after a first time power on reset the RAM contents will be indeterminate. During a power fail condition, if the power falls below 2V, the RAM contents are lost. Hence it should be assumed that after a power on/fail reset, POR = 1, the RAM contents are lost.

After a reset most SFRs are cleared. Interrupts and Timers are disabled. The Watchdog timer is disabled if the reset source was a POR. The port SFRs have FFh written into them which puts the port pins in a weak high state. Port 0 floats as it does not have on-chip pullups.

Table-6 SFR Reset Conditions

SFR Name	Reset Value	SFR Name	Reset Value
P0	11111111b	IE	01000000b
SP	00000111b	SADDR	00000000b
DPL	00000000b	P3	11111111b
DPH	00000000b	IP	11000000
DPL1	00000000b	SADEN	00000000b
DPH1	00000000b	T2CON	00000000b
DPS	00000000b	T2MOD	11111110b
PCON	00110000b	RCAP2L	00000000b
TCON	00000000b	RCAP2H	00000000b
TMOD	00000000b	TL2	00000000b
TL0	00000000b	TH2	00000000b
TL1	00000000b	TA	11111111b
TH0	00000000b	PSW	00000000b
TH1	00000000b	WDCON	1X0X0XX0b
CKCON	00000001b	ACC	00000000b
P1	11111111b	EIE	11101111b
SCON	00000000b	B	00000000b
SBUF	00000000b	EIP	11101111b
P2	11111111b	PC	0000h

The WDCON SFR bits are set/cleared in reset condition depending on the source of the reset.

	External reset	Watchdog reset	Power on reset
WDCON	1x0x0xx0b	1x0x01x0b	11000000b

The x indicates that the reset does not change the bit. The POR bit WDCON.6 is set only by the power on reset. The PFI bit WDCON.4 is set when the power fail condition occurs. However a power-on reset will clear this bit if

Vcc> Vpfw following the crystal start-up time. The WTRF bit WDCON.2 is set when the Watchdog timer causes a reset. A power on reset will also clear this bit. The EWT bit WDCON.1 is cleared by power on resets. This disables the Watchdog timer resets. A watchdog or external reset does not affect the EWT bit.

INTERRUPTS

The CS6208 has a three priority level interrupt structure with 8 interrupt sources. Each of the interrupt sources has an individual priority bit, flag, interrupt vector and enable bit. In addition, the interrupts a can be globally enabled or disabled.

Interrupt Sources

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can be either edge triggered or level triggered, depending on bits IT0 and IT1. The bits IE0 and IE1 in the TCON register are the flags which are checked to generate the interrupt. When an interrupt is generated by these external interrupt inputs, they will be cleared on entering the Interrupt Service routine, only if the interrupt type is edge triggered. In case of level triggered interrupt, the IE0 and IE1 flags are not cleared and will have to be cleared by the software. This is because in the level activated mode, it is the external requesting source that controls the interrupt flag bit rather than the on-chip hardware.

The Timer 0 and 1 Interrupts are generated by the TF0 and TF1 flags. These flags are set by the overflow in the Timer 0 and Timer 1. The TF0 and TF1 flags are automatically cleared by the hardware when the timer interrupt is serviced. The Timer 2 interrupt is generated by a logical OR of the TF2 and the EXF2 flags. These flags are set by overflow/underflow or capture/reload events in the timer 2 operation. The hardware does not clear these flags when a timer 2 interrupt is executed. Software has to resolve the cause of the interrupt between TF2 and EXF2 and clear the appropriate flag.

The Watchdog timer can be used as a system monitor or a simple timer. In either case, when the time-out count is reached, the Watchdog timer interrupt flag WDIF (WDCON.3) is set. If the interrupt is enabled by the enable bit EIE.4, then an interrupt will occur.

The Serial block can generated interrupts on reception or transmission. There is however only one interrupt source from the Serial block, which is obtained by ORing the RI and TI bits in the SCON SFR. These bits are not automatically cleared by the hardware and the user will have to clear these bits using software.

The CS6208 continuously monitors the Vcc levels, and if the Vcc level drops below the Vpfw level, then the Power Fail Interrupt flag PFI (WDCON.4) is set. If this interrupt is enabled then an interrupt will occur. Note that the Power fail interrupt has the highest priority of all the interrupts. Its priority cannot be altered by the user but it can be enabled or disabled by software. The EPFI bit enables the Power Fail interrupt. This bit is not controlled by the global interrupt enable EA.

All the bits that generate interrupts can be set or reset by hardware and thereby software initiated interrupts can be generated. Each of the individual interrupts can be enabled or disabled by setting or clearing a bit in the IE SFR. IE also has a global enable/disable bit EA, which can be cleared to disable all the interrupts, except PFI, at once.

Priority Level Structure

There are three priority levels for the interrupts, highest, high and low. The Power-fail interrupt PFI has the highest priority. No other interrupt can have this priority. The other interrupt sources can be individually set to either high or low levels. Naturally, a higher priority interrupt cannot be interrupted by a lower priority interrupt. However there exists a predefined hierarchy amongst the interrupts themselves. This hierarchy comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level. This hierarchy is defined as shown below, the interrupts are numbered starting from the highest priority to the lowest.

Table-7 Priority structure of interrupts

Source	Flag	Priority level
Power-fail	PFI	1 (highest)

Source	Flag	Priority level
External Interrupt 0	IE0	2
Timer 0 Overflow	TF0	3
External Interrupt 1	IE1	4
Timer 1 Overflow	TF1	5
Serial Port	RI + TI	6
Timer 2 Overflow	TF2 + EXF2	7
Watchdog Timer	WDIF	8 (lowest)

The interrupt flags are sampled in C2 of every machine cycle. In the same machine cycle, the sampled interrupts are polled and their priority is resolved. If certain conditions are met then the hardware will execute an internally generated LCALL instruction which will vector the process to the appropriate interrupt vector address. The conditions for generating the LCALL are

1. An interrupt of equal or higher priority is not currently being serviced.
2. The current polling cycle is the last machine cycle of the instruction currently being executed.
3. The current instruction does not involve a write to IP, IE, EIP or EIE registers and is not a RETI.

If any of these conditions are not met, then the LCALL will not be generated. The polling cycle is repeated every machine cycle, with the interrupts sampled in C2 in the same machine cycle. If an interrupt flag is active in one cycle but not responded to, and is not active when the above conditions are met, the denied interrupt will not be serviced. This means that active interrupts are not remembered, every polling cycle is new.

The processor responds to a valid interrupts by executing an LCALL instruction to the appropriate service routine. This may or may not clear the flag which caused the interrupt. In case of Timer interrupts, the TF0 or TF1 flags, are cleared by hardware whenever the processor vectors to the appropriate timer service routine. In case of External interrupt, $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, the flags are cleared only if they are edge triggered. In case of Serial interrupts, the flags are not cleared by hardware. In the case of Timer 2 interrupt, the flag are not cleared by hardware. The Power-fail interrupt PFI flag and Watchdog timer interrupt flag WDIF has to be cleared by software. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter contents onto the Stack, but does not save the Program Status Word PSW. The PC is reloaded with the vector address of that interrupt which caused the LCALL. These vector address for the different sources are as follows

Table 8: Vector locations for interrupt sources

Source	Vector Address
Power-fail	0033h
External Interrupt 0	0003h
Timer 0 Overflow	000Bh
External Interrupt 1	0013h
Timer 1 Overflow	001Bh
Serial Port	0023h
Timer 2 Interrupt	002Bh
Watchdog Timer	0063h

The vector table is not evenly spaced, this is to accommodate future expansions to the device family.

Execution continues from the vectored address till an RETI instruction is executed. On execution of the RETI instruction the processor pops the Stack and loads the PC with the contents at the top of the stack. The user must take care that the status of the stack is restored to what is was after the hardware LCALL, if the execution is to return to the interrupted program. The processor does not notice anything if the stack contents are modified and will proceed with execution from the address put back into PC. Note that a RET instruction would do exactly the same process as a RETI instruction, but it would not inform the Interrupt Controller, that the interrupt service routine is completed, an would leave the controller still thinking that the service routine is under progress.

External Interrupts

There are two external interrupt sources in this processor, $\overline{INT0}$ and $\overline{INT1}$. These interrupts can be programmed to be edge triggered or level activated, by setting bits IT0 and IT1 in TCON SFR. In the edge triggered mode, the INTx inputs are sampled at C4 in every machine cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag IEx in TCON is set. The flag bit the requests the interrupt. Since the external interrupts are sampled every machine cycle, they have to be held high or low for at least on complete machine cycle. The IEx flag is automatically cleared when the service routine is called.

If the level activated mode is selected, then the requesting source has to hold the pin low till the interrupt is serviced. The IEx flag will not be cleared by the hardware on entering the service routine. If the interrupt continues to be held low even after the service routine is completed, then the processor may acknowledge another interrupt request from the same source.

Response Time

The response time for each interrupt source depends on several factors like nature of the interrupt and the instruction under progress. In the case of external interrupt INT0 and INT1, they are sampled at C3 of every machine cycle and then their corresponding interrupt flags IE0 and IE1 will be set or reset. Similarly, the Serial port flags RI and TI are set in S5P2. The Timer 0 and 1 overflow flags are set at C3 of the machine cycle in which overflow has occurred. These flag values are polled only in the next machine cycle. If a request is active and all three conditions are met, then the hardware generated LCALL is executed. This call itself takes four machine cycles to be completed. Thus there is a minimum time of five machine cycles between the interrupt flag being set and the interrupt service routine being executed.

A longer response time should be anticipated if any of the three conditions are not met. If a higher or equal priority is being serviced, then the interrupt latency time obviously depends on the nature of the service routine currently being executed. If the polling cycle is not the last machine cycle of the instruction being executed, then an additional delay is introduced. The maximum response time (if no other interrupt is in service) occurs if the CS6208 is performing a write to IE, IP, EIE or EIP and then executes a MUL or DIV instruction. From the time an interrupt source is activated, the longest reaction time is 12 machine cycles. This includes 1 machine cycle to detect the interrupt, 2 machine cycles to complete the IE, IP, EIE or EIP access, 5 machine cycles to complete the MUL or DIV instruction and 4 machine cycles to complete the hardware LCALL to the interrupt vector location.

Thus in a single-interrupt system the interrupt response time will always be more than 5 machine cycles and not more than 12 machine cycles. The maximum latency of 12 machine cycle is 48 clock cycles. Note that in the standard 8051 the maximum latency is 8 machine cycles which equals 96 machine cycles. This is a 50% reduction in terms of clock periods.

PROGRAMMABLE TIMERS/COUNTERS

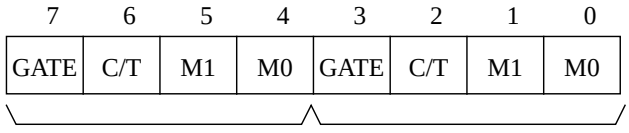
The CS6208 has three 16-bit programmable timer/counters and one programmable Watchdog timer. The

Watchdog timer is operationally quite different from the other two timers.

TIMER/COUNTERS 0 & 1

The CS6208 has two 16-bit Timer/Counters, TM0 and TM1. Each of these Timer/Counters has two 8 bit registers which form the 16 bit counting register. For Timer/Counter TM0 they are TH0, the upper 8 bits register and TL0, the lower 8 bit register. Similarly Timer/Counter TM1 has two 8 bit registers, TH1 and TL1. The two can be configured to operate either as timers, counting machine cycles or as counters counting external inputs.

When configured as a "Timer", the timer counts clock cycles. The timer clock can be programmed to be thought of as 1/12 of the system clock or ? of the system clock. In the "Counter" mode, the register is incremented on the falling edge of the external input pin, T0 in case of TM0 and T1 for TM1. The T0 and T1 inputs are sampled in every machine cycle at C4. If the sampled value is high in one machine cycle and low in the next, then a valid high to low transition on the pin is recognized and the count register is incremented. Since it takes two machine cycles to recognize a negative transition on the pin, the maximum rate at which counting will take place is 1/24 of the master clock frequency. In either the "Timer" or "Counter" mode, the count register will be updated at C3. Therefore, in the "Timer" mode, the recognized negative transition on pin T0 and T1 can cause the count register value to be updated only in the machine cycle following the one in which the negative edge was detected.



GATE — This bit controls the gating operations. When this bit is set, the Timer/Counter “x” is enabled only while “INTx” pin is high and “TRx” bit is set high. If this bit is cleared, then Timer/Counter “x” is enabled only if “TRx” is set.

C/T — This bit selects the Timer or Counter mode of operation. If this bit is set then the Counter mode is selected, else the Timer mode is selected.

M1, M0 — M1 and M0 selects the operating mode for the Timer/Counter

M1	M0	Operating Mode
0	0	“THx” acts as a 8 bit Timer/Counter, with “TLx” as the 5 bit pre-scaler.
0	1	“THx” and “TLx” are cascaded to form a single 16 bit Timer/Counter.
1	0	This is the Auto Reload mode.
1	1	Timer 0: TL0 is an 8 bit Timer/Counter controlled by the Timer 0 control bits. TH0 is a 8 bit timer controlled by the Timer 1 control bits. Timer 1: Timer/Counter is stopped.

Figure-13 TMOD: Timer/Counter Mode Control Register

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

TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to timer 1 interrupt routine.
TR1	Timer 1 Run control bit. Set/Cleared by software to turn Timer/Counter on/off.
TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to timer 0 interrupt routine.
TR0	Timer 0 Run control bit. Set/Cleared by software to turn Timer/Counter on/off.
IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt edge is detected. Cleared when interrupt is processed.
IT1	Interrupt 1 Type control bit. Set/ Cleared by software to specify falling edge/low level triggered external interrupt.
IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt edge id detected. Cleared when interrupt is processed.
IT0	Interrupt 0 Type control bit. Set/Cleared by software to specify falling edge /low level triggered external interrupt.

Figure-14 TCON: Timer/Counter Register.

The "Timer" or "Counter" function is selected by the "C/T" bit in the TMOD Special Function Register. Each Timer/ Counter has one selection bit for its own; bit 2 of TMOD selects the function for Timer/Counter 0 and bit 6 of TMOD selects the function for Timer/Counter 1. In addition each Timer/Counter can be set to operate in any one of four possible modes. The mode selection is done by bits M0 and M1 in the TMOD SFR. Modes 0, 1 and 2 are identical for both the Timer/Counters, but Mode 3 is different. The four modes of operation are described below.

Time -Base Selection

The CS6208 gives the user two modes of operation for the timer. The timers can be programmed to operate like the standard 8051 family, counting at the rate of 1/12 of the clock speed. This will ensure that timing loops on the CS6208 and the standard 8051 can be matched. This is the default mode of operation of the CS6208 timers. The user also has the option to count in the turbo mode, where the timers will increment at the rate of ? clock speed. This will straight away increase the counting speed three times. This selection is done by the T0M and T1M bits in CKCON SFR. A reset sets these bits to 0 and the timers operate in the standard 8051 mode. The user should set these bits to 1 if the timers are to operate in turbo mode.

MODE 0

In Mode 0, the timer/counters act as a 8 bit counter with a 5 bit, divide by 32 pre-scaler. In this mode we have a 13 bit timer/counter. The 13 bit counter consists of 8 bits of THx and 5 lower bits of TLx. The upper 3 bits of TLx

are ignored.

The negative edge of the clock increments the count in the TLx register. When the fifth bit in TLx moves from 1 to 0, then the count in the THx register is incremented. When the count in THx moves from FFh to 00h, then the overflow flag TFx in TCON SFR is set. The counted input is enabled only if TRx is set and either GATE = 0 or INTx = 1. When C/T is set to 1, then it will count clock cycles and if C/T is set to 0, then it will count 1 to 0 transitions on T0 (P3.4) for timer 0 and T1 (P3.5) for timer 1. When the 13 bit count reaches 1FFFh the next count will cause it to roll-over to 0000h. The timer overflow flag TFx of the relevant timer is set and if enabled an interrupts will occur.

Note that when used as a timer, the time base may be either clock cycles/12 or clock cycles/4 as selected by the bits TxM of the CKCON SFR.

MODE 1

Mode 1 is similar to Mode 0 except that the counting register form a 16 bit counter, rather than a 13 bit counter. This means that all the bits of THx and TLx are used. Roll-over occurs when the timer moves from a count of FFFFh to 0000h. The timer overflow flag TFx of the relevant timer is set and if enabled an interrupts will occur. The selection of the timebase in the timer mode is similar to that in Mode 0. The gate function operates similarly as in Mode 0.

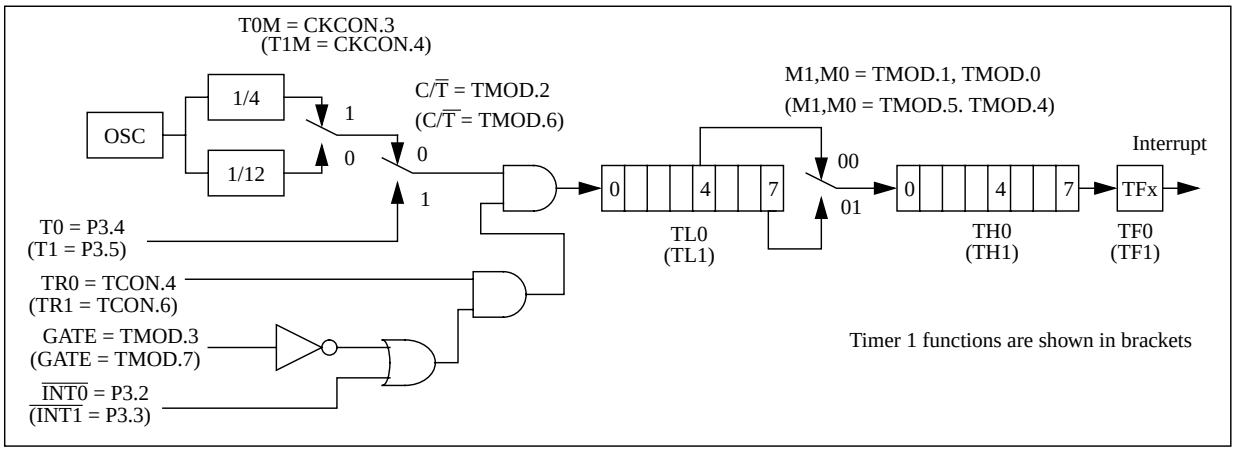


Figure-15 Mode 0 & Mode 1

MODE 2

In Mode 2, the timer/counter is in the Auto Reload Mode. In this mode, TLx acts as a 8 bit count register, while THx holds the reload value. When the TLx register overflows from FFh to 00h, the TFx bit in TCON is set and TLx is reloaded with the contents of THx and the counting process continues from here. The reload operation leaves the contents of the THx register unchanged.

Counting is enabled by the TRx bit and proper setting of GATE and INTn pins. As in the other two modes 0 and 1,

mode 2 allows counting of either clock cycles (clock/12 or clock/4) or pulses on pin Tn.

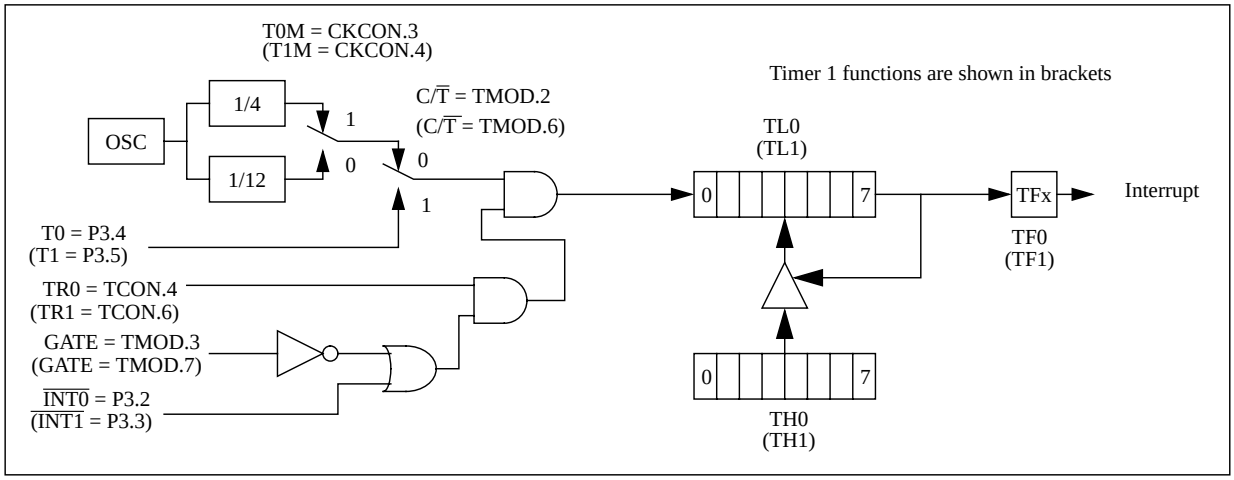


Figure-16 Mode 2: 8-bit Auto-reload mode

MODE 3

Mode 3 has different operating methods for the two timer/counters. For timer/counter TM1, mode 3 simply freezes the counter. This is the same as setting TR1 to 0.

Timer/Counter TM0 however configures TL0 and TH0 as two separate 8 bit count registers in this mode. The logic for this mode is shown in the figure. TL0 uses the Timer/Counter TM0 control bits C/T, GATE, TR0, INT0 and TF0. The TL0 can be used to count clock cycles (clock/12 or clock/4) or 1-to-0 transitions on pin T0 as determined by C/T (TMOD.2). TH0 is forced as a clock cycle counter (clock/12 or clock/4) and takes over the use of TR1 and TF1 from Timer/Counter TM1.

Mode 3 is used in cases where an extra 8 bit timer is needed. With Timer 0 in Mode 3, Timer 1 can still be used in Modes 0, 1 and 2., but its flexibility is somewhat limited. While its basic functionality is maintained it no longer has control over its overflow flag TF1 and the enable bit TR1.

Timer 1 can still be used as a timer/counter and retains the use of GATE and $\overline{INT1}$ pin. In this condition it can be turned on and off by switching it out of and into its own Mode 3. It can also be used as a baud rate generator for

the serial port.

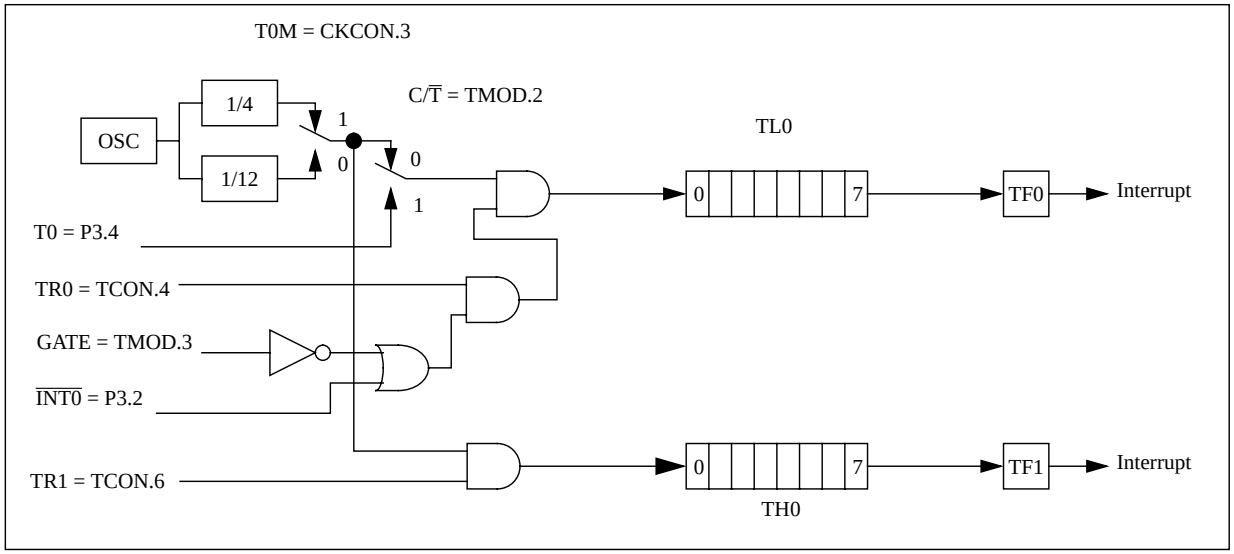


Figure-17 Mode 3

TIMER/COUNTER 2

As is seen in the register descriptions, Timer 2 has several abilities not found in Timers 0 and 1. However, it does not offer the 13-bit and dual 8-bit modes. Thus it runs 16-bit modes at all times. Also note that instead of offering an 8-bit auto-reload mode, Timer 2 has a16-bit auto-reload mode. This mode uses the Timer Capture registers to hold the reload values. The modes available on Timer 2 are described below.

16-BIT TIMER/COUNTER

In this mode, Timer 2 performs a simple timer or counter function where it behaves similarly to mode 1 of Timers 0 and 1, but uses 16 bits instead of 8 bits. This mode, along with the optional capture mode described below, is illustrated in Figure 11-4. The 16-bit count values are found in TL2 and TH2 Special Function Registers (addresses CCh and CDh respectively). The selection of whether a Timer or Counter function is performed is made using the bit C/T2 (T2CON.1). When C/T2 is set to a logic 1, Timer2 behaves as a counter where it counts 1 to 0 transitions at the T2 (P1.0) pin. When C/T2 is set to a logic 1, Timer2 functions as a timer where it counts the oscillator cycles divided by either 12 or 4 as determined by bit T2M(T2CON.5). Timing or counting is enabled by setting bit TR2 (T2CON.2) to 1, and disabled by setting it to 0. When the counter rolls over from FFFFh to 0000h, the TF2 flag (T2CON.7) is set and will cause an interrupt if Timer 2's interrupt.

16-BIT TIMER WITH CAPTURE

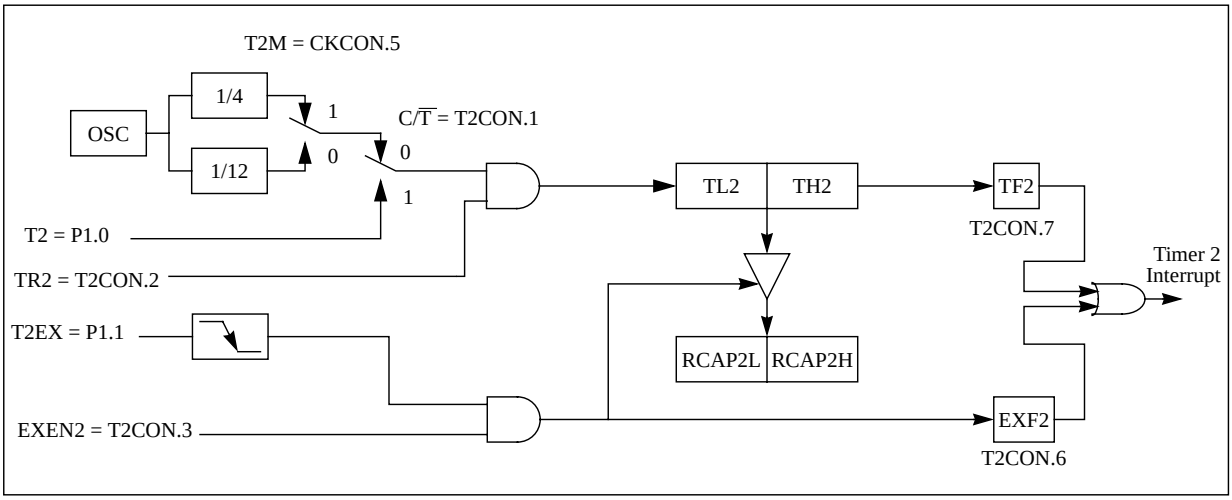


Figure-18 16-Bit Capture Mode

A diagram of Timer 2's Capture Mode is shown in Figure 19. In this mode, the timer performs basically the same 16-bit timer/counter function describe above. However, a 1 to 0 transition on T2EX (pin P1.1) causes the value in Timer 2 to be transferred into the capture registers if enabled by EXEN2 (T2CON.3). The capture registers, RCAP2L and RCAP2H, correspond to TL2 and TH2 respectively. The capture function is enabled by the CP/RL2 (T2CON.0) bit. When set to a logic 1, the timer is in capture mode as described, When set to a logic 0, the timer in auto-reload mode described later. As was possible with Timers 0 and 1, the timebase for Timer 2 may be selected to be oscillator cycles divided by either 12 or 4 when in this mode.

16-BIT AUTO-RELOAD TIMER/COUNTER

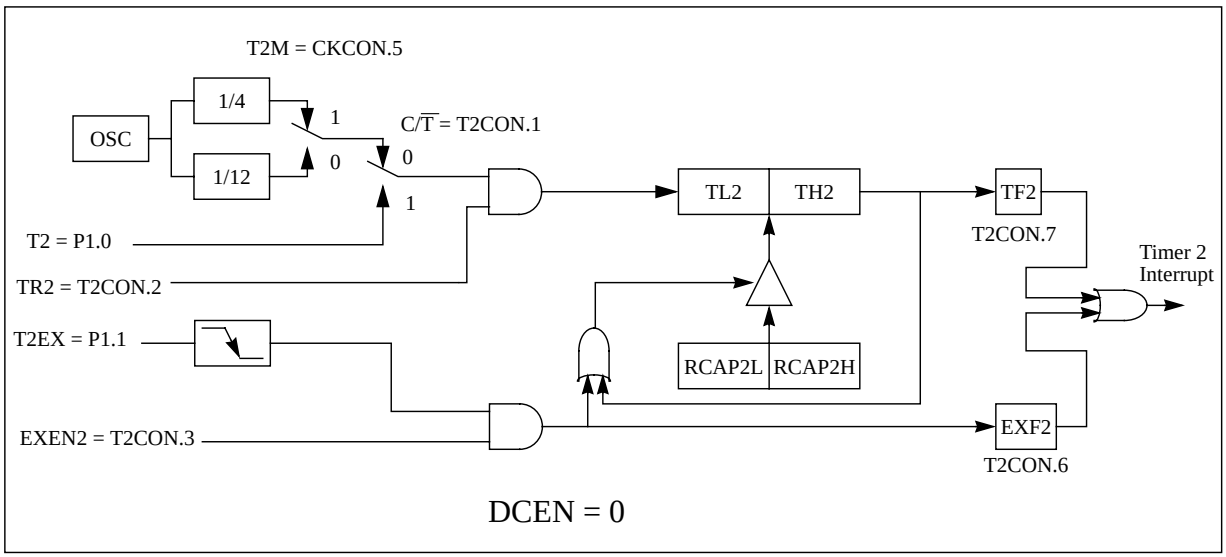


Figure-19 16-Bit Auto-reload Up Count Mode

This module is illustrated in Figure 20. When Timer 2 reaches an overflow state, i.e., rolls over from FFFFh to 0000, it will set the TF2 Flag. This flag can generate an interrupt if enabled. In addition, the timer will restore its starting value and begin timing (or counting again. The starting value is pre-loaded by software into the capture registers RCAP2L and RCAP2H. These registers cannot be used for capture functions while also performing auto-reload, so these modes are mutually exclusive. Auto-reload is invoked by the CP/$\overline{RL2}$ (T2CON.0) bit. When set to a logic 0, the timer is in auto-reload mode. When CP/$\overline{RL2}$ is set to a logic 1, the timer is in capture mode described above. If the oscillator timebase is (C/$\overline{T2}$ = T2CON.1 = 0), the timer's input may be selected to be oscillator cycles divided by either 12 or 4 as determined by T2M (CKCON.5). Otherwise pulses on pin T2 (P1.0) as counted when C/$\overline{T2}$ = 1. As in other modes Counting or timing is enabled or disabled with TR2 (T2CON.2).

When in auto-reload mode, Timer2 can also be forced to reload with T2EX (P1.1) pin. A 1 to 0 transition will force a reload if enabled bye the EXEN (T2CON.3) bit. If EXEN2 is set to a logic 1, then a 1 to 0 transition on T2EX will cause a reload. Otherwise, the T2EX pin will be ignored.

UP/DOWN COUNT AUTO-RELOAD TIMER/COUNTER

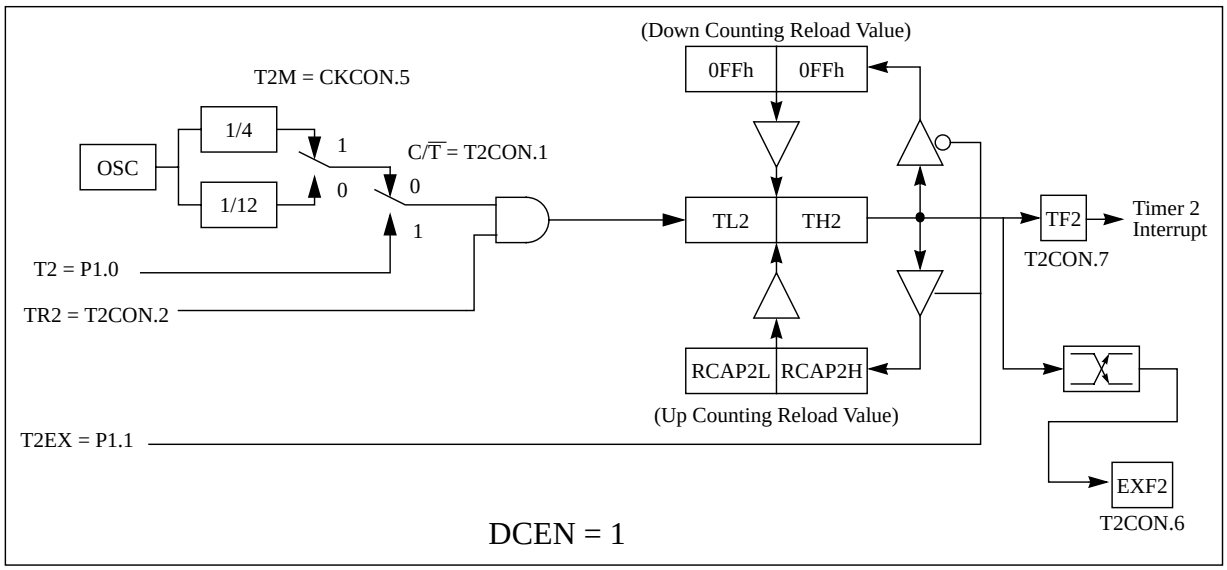


Figure-20 16-Bit Auto-reload Up/Down Count Mode

When operating in auto-reload mode, Timer2 can also function as an up/down auto-reload counter.This option is selected by the DCEN (T2MOD.0) bit, and is illustrated in Figure 21. When DCEN is set to a logic 1, Timer will count up or down as controlled by the state of pin T2EX(P1.1). T2EX will cause upward counting when a logic 0 is applied. When DCEN=0, timer2 only counts up.

When an overflow occurs in the upward counting direction, the value in RCAP2L and RCAP2H will be loaded into T2L and T2H respectively. In the down count direction, an underflow occurs when the values in T2L and T2H match the values in RCAP2L and RCAP2H respectively. When an underflow occurs, FFFFh is loaded into T2L and T2H and counting continues.

Note that in this mode, the overflow/underflow output of the timer is provided to an edge detection circuit as well as to the TF2 bit (T2CON.7). This edge detection circuit toggles the EXF2 bit (T2CON.6) on every overflow or underflow. Therefore, the EXF2 bit behaves as a seventeenth bit of the counter, and may be used as such.

BAUD RATE GENERATOR

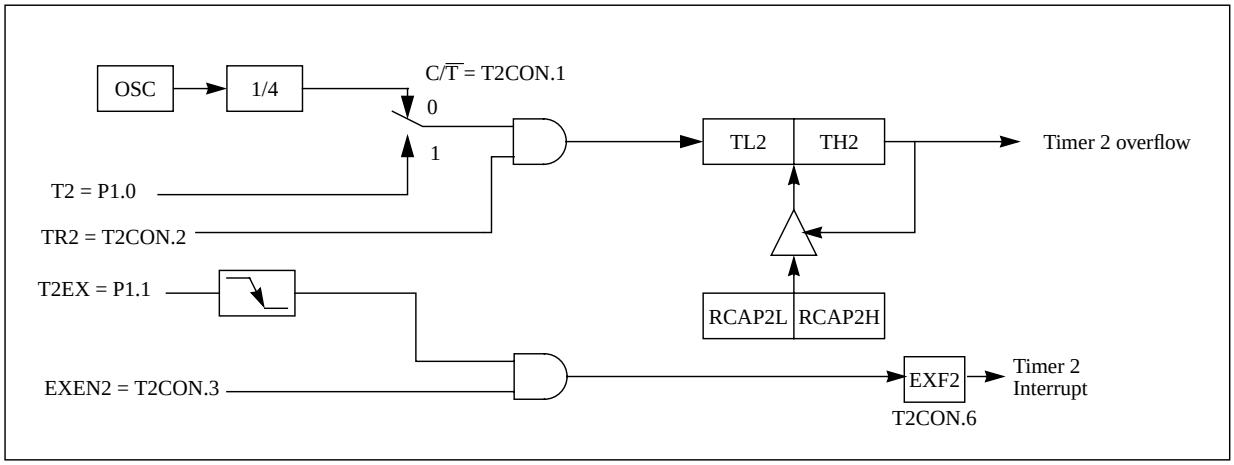


Figure-21 Baud Rate Generator Mode

Timer2 can be used to generate baud rates for Serial port in serial modes 1 or 3. Baud rate generator mode is invoked by setting either the RCLK or TCLK bit in the T2CON register to a logic 1, as illustrated in Figure 22. In this mode, the timer continues to function in auto-reload mode, but instead of setting the interrupt flag T2F (T2CON.7) and potentially causing an interrupt, the overflow is used to generate the shift clock for the serial port function. As in normal auto-reload mode, an overflow causes the values previously loaded into RCAP2L and RCAP2H to be transferred into T2L and T2H respectively. Note that when RCLK or TCLK is set to 1, the Timer 2 is forced into 16-bit auto-reload mode regardless of the CP/$\overline{RL2}$ bit.

As explained above, the timer itself cannot set the T2F interrupt flag and therefore cannot generate an interrupt. However if EXEN2 (T2CON.3) is set to 1, a 1 to 0 transition on the T2EX (P1.1) pin will cause a Timer 2 Interrupt to occur. Therefore in this mode, the T2EX pin may be used as an additional external interrupt if desired.

Another somewhat unique feature of the baud rate generator mode is that the crystal derived time-base for the timer is the crystal frequency divided by 2. No other crystal divider selection is possible. If a different timebase is desire, bit C/T2 (T2CON.1) may be set to a 1 allowing the timebase to be derived from an external clock source supplied by the user on pin T2 (P1.0).

WATCHDOG TIMER

The Watchdog timer is a free-running timer which can be programmed by the user to serve as a system monitor, a time-base generator or an event timer. It is basically a set of dividers that timer that divide the system clock. The divider output is selectable and determines the time-out interval.

When the time-out occurs a flag is set, which can cause an interrupt if so enabled, and a system reset can also be caused if it too is enabled. The interrupt will occur if the individual interrupt enable and the global enable is set. The interrupt and reset functions are independent of each other and may be used separately or together depending on the users software.

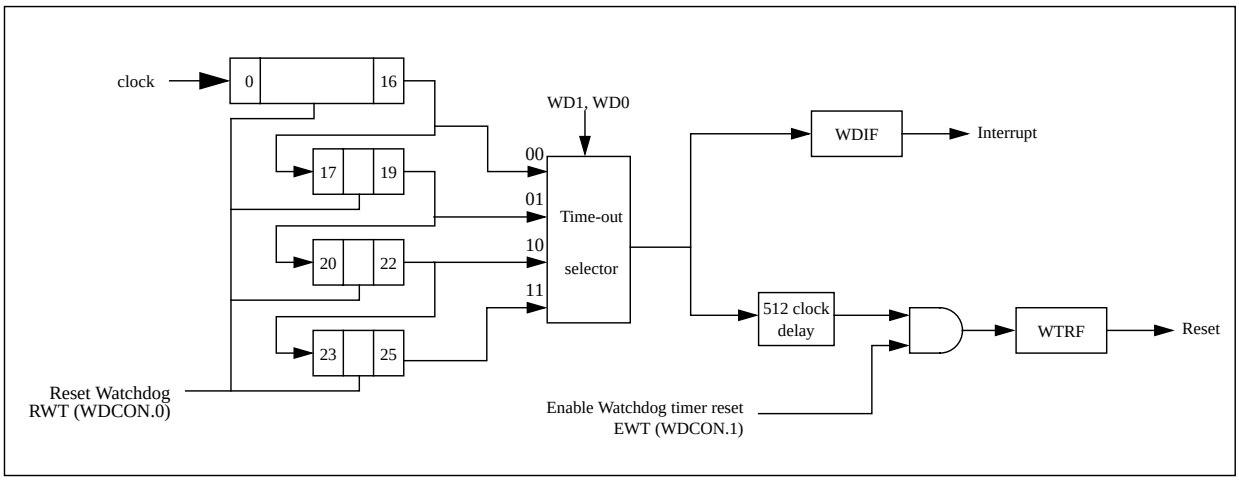


Figure-22 Watchdog Timer

The Watchdog timer should first be restarted by using RWT. This ensures that the timer starts from a known state. The RWT bit is used to restart the watchdog timer. This bit is self clearing, i.e. after writing a 1 to this bit the software will automatically clear it. The watchdog timer will now count clock cycles. The time-out interval is selected by the two bits WD1 and WD0 (CKCON.7 and CKCON.6). When the selected time-out occurs, the Watchdog interrupt flag WDIF (WDCON.3) is set After the time-out has occurred, the watchdog timer waits for an additional 512 clock cycles. The software must issue a RWT to reset, the watchdog before the 512 clock have elapsed. If the Watchdog reset EWT (WDCON.1) is enabled, then 512 clocks after the time-out if there is no RWT, then a system reset due to Watchdog timer will occur. This will last for two machine cycles and the Watchdog timer reset flag WTRF (WDCON.2) will be set. This indicates to the software that the watchdog was the cause of the reset.

When used as a simple timer, the reset and interrupt functions are disabled. The timer will set the WDIF flag each time the timer completes the selected time interval. The WDIF flag is polled to detect a time-out and the RWT allow software to restart the timer. The Watchdog timer can also be used as a very log timer. The interrupt feature is enabled in this case. Every time the time-out occurs an interrupt will occur if the global interrupt enable EA is set.

The main use of the Watchdog timer is as a system monitor. This is important in real time control applications. In case of some power glitches or electro-magnetic interference, the CS6208 may begin to execute errant code. If this is left unchecked the entire system may crash. Using the watchdog timer interrupt during software development will allow the user to select ideal watchdog reset locations. The code is first written without the watchdog interrupt or reset. Then the watchdog interrupt is enabled to identify code locations where interrupt

occurs. The user can now insert instructions to reset the watchdog timer which will allow the code to run without any watchdog timer interrupts. Now the watchdog timer reset is enabled and the watchdog interrupt may be disabled,. Now if any errant code is executed, then the reset watchdog timer instructions will not be executed at the required instants and watchdog reset will occur.

The watchdog time-out selection will result in different time-out values depending on the clock speed. The reset, when enabled, will occur 512 clocks after the time-out has occurred.

Table-8 Time-out values for the Watchdog timer

WD1	WD0	Watchdog Interval	Number of Clocks	Time @1.8432MHz	Time @10MHz	Time @20MHz
0	0	217	131072	71.11 ms	13.11 ms	5.24 ms
0	1	220	1048576	568.89 ms	104.86 ms	41.94 ms
1	0	223	8388608	4551.11 ms	838.86 ms	335.54 ms
1	1	226	67108864	36408.88 ms	6710.89 ms	2684.35 ms

The Watchdog timer will de disabled by a power-on/fail reset. The Watchdog timer reset does not disable the watchdog timer, but will restart it. In general, software should restart the timer to put it into a known state.

The control bits that support the Watchdog timer are discussed below.

WATCHDOG CONTROL

WDIF WDCON.3 - Watchdog Timer Interrupt flag. This bit is set whenever the time-out occurs in the watchdog timer. If the Watchdog interrupt is enables (EIE.4), then an interrupt will occur (if the global interrupt enable is set and other interrupt requirements are met). Software or any reset can clear this bit.

WTRF WDCON.2 - Watchdog Timer Reset flag. This bit is set whenever a watchdog reset occurs. This bit is useful for determined the cause of a reset. Software must read it, and clear it manually. A Power-fail reset will clear this bit. If EWT = 0, then this bit will not be affected by the watchdog timer.

EWT WDCON.1 - Enable Watchdog timer Reset. This bit when set to 1 will enable the Watchdog timer reset function. Setting this bit to 0 will disable the Watchdog timer reset function but will leave the timer running

RWT WDCON.0 - Reset Watchdog Timer. This bit is used to clear the Watchdog timer and to restart it. This bit is self-clearing, so after the software writes 1 to it the hardware will automatically clear it. If the Watchdog timer reset is enabled, then the RWT has to be set by the user within 512 clocks of the time-out. If this is not done then a Watchdog timer reset will occur.

CLOCK CONTROL

WD1, WD0 CKCON.7, CKCON.6 - WAtchdog timer Mode select bits. These two bits select the time-out interval for the watchdog timer. The reset time is 512 clock longer than the interrupt time-out value.

The default Watchdog time-out is 217 clocks which is the shortest time-out period. The EWT, WDIF and RWT bits are protected by the Timed Access procedure. This prevents software from accidently enabling or disabling the watchdog timer. Most importantly it makes it highly improbable that errant code can enable or disable the watchdog timer.

SERIAL PORT:

Serial port in the CS6208 is a full duplex port. The CS6208 provides the user with additional features such as the Frame Error Detection and the Automatic Address Recognition.

The serial ports are capable of synchronous as well as asynchronous communication. In Synchronous mode the CS6208 generates the clock and operates in a half duplex mode. In the asynchronous mode, full duplex

operation is available. This means that it can simultaneously transmit and receive data. In addition, the receive register is double buffered. This allows reception of the second data byte before the first byte is read from the receive register.

The transmit register and the receive buffer are both addressed as SBUF Special Function Register. However any write to SBUF will be to the transmit register, while a read from SBUF will be from the receive buffer register.

The serial port can operate in four different modes.

MODE 0

This mode provides the synchronous communication with external devices. In this mode serial data is transmitted and received on the RXD line. TXD is used to transmit the shift clock. The TxD clock is provided by the CS6208 wether the device is transmitting or receiving. This mode is therefore a half duplex mode of serial communication.

In this mode, 8 bits are transmitted or received per frame. The LSB is transmitted/received first. The baud rate is fixed at 1/12 or ? of the oscillator frequency. This baud rate is determined by the SM2 bit (SCON.5). When this bit is set to 0, then the serial port runs at 1/12 of the clock.

When set to 1, the serial port runs at ? of the clock. This additional facility of selectable baud rate in mode 0 is the only difference between the standard 8051 and the CS6208.

MODE 1

In Mode 1, the full duplex asynchronous mode is used. Serial communication frames are made up of 10 bits transmitted on TXD and received on RXD. The 10 bits consist of a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in the SFR SCON.

In mode 1, the baud rate in this mode is variable. The serial baud can be programmed to be 1/16 or 1/32 of the Timer 1 or timer 2 overflow. Since the Timer 1 and timer 2 can be set to different reload values, a wide variation is baud rates is possible.

MODE 2

This is an asynchronous mode in which 11 bits are transmitted on TXD and received on RXD. The 11 bits consist of a start bit (0), 8 data bits (LSB first), a programmable 9th data bit (TB8 in SCON) and a stop bit (1). The 9th bit can be assigned 1 or 0 by writing to TB8 (SCON.3). On transmission, this bit will be inserted into the data stream. On reception the received 9th bit goes into RB8 in SCON. The stop bit is ignored. The baud rate is programmable to 1/32 or 1/64 of the oscillator frequency. This baud rate is determined by the SMOD bit in PCON SFR.

MODE 3

This mode is similar to Mode 2 in all respects, except that the baud rate is programmable.

The user must first initialize the Serial related SFR SCON, before any communication can take place. This involves selection of the Mode and baud rate. The Timer 1 should also be initialized if modes 1 and 3 are used.

In all four modes, transmission is started by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. This will generate a clock on the TxD pin and shift in 8 bits on the RxD pin. Reception is initiated in the other modes by the incoming start bit if REN = 1. The external device will start the communication by transmitting the start bit.

Table-9 Serial Modes

The SFR bits that control the Serial port and the bits that act as status or flags related to the Serial port are as follows.

SM1	SM0	Mode	Type	Baud Clock	Frame size	Start bit	Stop bit	9th bit function
0	0	0	Synch	4 or 12 tclk	8 bits	No	No	None
0	1	1	Asynch	Timer 1 or 2	10 bits	1	1	None
1	0	2	Asynch	32 or 64 tclk	11 bits	1	1	0, 1
1	1	3	Asynch	Timer 1 or 2	11 bits	1	1	0, 1

SERIAL PORT CONTROL SFR

SM0/FE SCON.7 - Serial Port Mode bit 0 or Framing Error Flag. The SMOD0 bit (PCON.6) determines the function of this bit. There are actually two separate flags, one for FE and the other as SM0. If the SMOD0 bit is set, then the bit will show FE, else it will show SM0.

SM1 SCON.6 - Serial Port Mode bit 1.

SM2 SCON.5 - Serial Port Mode bit 2. Setting this bit to 1 enables the multi-processor communication in Mode 2 and 3. If the 9th bit is 0, then RI will not be set. In Mode 1, setting the SM2 bit to 1 causes the RI bit not to be set if a valid stop bit is not received. In the CS6208, SM2 has a new function in Mode0. When set to logic 0, the serial port runs at 1/12 clock speed in mode 0.

When set to 1, the serial port runs at ? clock speed in mode 0. This results in faster synchronous communication.

REN SCON.4 - Receive Enable. The receive shift register will be enabled only if this bit is 1.

TB8 SCON.3 - Set/Cleared to define the state of the 9th transmission data bit in mode 2 and 3.

RB8 SCON.2 - Indicates the state of the incoming 9th bit in modes 2 and 3. In mode1, when SM2=0, RB8 is the state of the stop bit. RB8 is not used in mode0.

TI SCON.1 - Transmit Interrupt Flag. This flag is set by hardware to indicate that a frame has been transmitted. In mode 0, Ti is set at the end of the 8th data bit. In all other modes, this bit is set at the end of the last data bit and not the end of the frame. This bit must be manually cleared by software.

RI SCON.0 - Receive Interrupt. This flag indicates that a serial word has been received. In mode 0, RI is set at the end of the 8th bit. In mode 1, it is set after the last sample of the incoming stop bit, subject to the state of SM2. In modes 2 and 3, it is set after the last sample of RB8. This bit must be manually cleared by software.

POWER CONTROL SFR

SMOD PCON.7 - This bit doubles the serial baud rate in modes 1, 2 and 3 when it is 1.

SMOD0 PCON.6 - Frame Error Detection Enable. When this bit is set to 1, then the SM0/FE bit (SCON.7) will show the FE bit. When SMOD0 is 0, then SM0/FE will show the SM0 function for the serial port.

Serial Port Description

Mode 0

Mode 0 is used to transmit data synchronously, in a half duplex format. The functional block diagram is shown below. Data enters and leaves the Serial port on the RxD line. The Txd line is used to output the shift clock. The shift clock is used to shift data into and out of the CS6208 and the device at the other end of the line.

In this mode 8 bits are transmitted per frame. The LSB is transmitted/ received first. The baud rate is selectable either 1/12 clock or 1/4 clock speed, depending on the SM2 bit. Any instruction that causes a write to SBUF will start the transmission. The shift clock will get activated and data will be shifted out on RxD pin till all 8 bits are transmitted. If SM2 = 1, then the data on RxD will appear 1 clock period before the falling edge of shift clock on TxD. The clock on TxD then remains low for 2 clock periods and then goes high again. If SM2 = 0, the data on RxD will appear 3 clock periods before the falling edge of shift clock on TxD. The clock on TxD then remains low for 6 clock periods and then goes high again. This ensures that at the receiving end the data on RxD line can be clocked in either on the rising edge of the shift clock on TxD or latched when the TxD clock is low. The TI flag is set high in C1 following the end of transmission of the last bit.

The Serial port will receive data when REN is 1 and RI is zero. The shift clock (TxD) will be activated and the serial port will latch data on the rising edge of shift clock. The external device should therefore present data on the falling edge on the shift clock. This process continues till all the 8 bits have been received. The RI flag is set in C1 following the last rising edge of the shift clock on TxD. This will stop reception, till the RI is cleared by software.

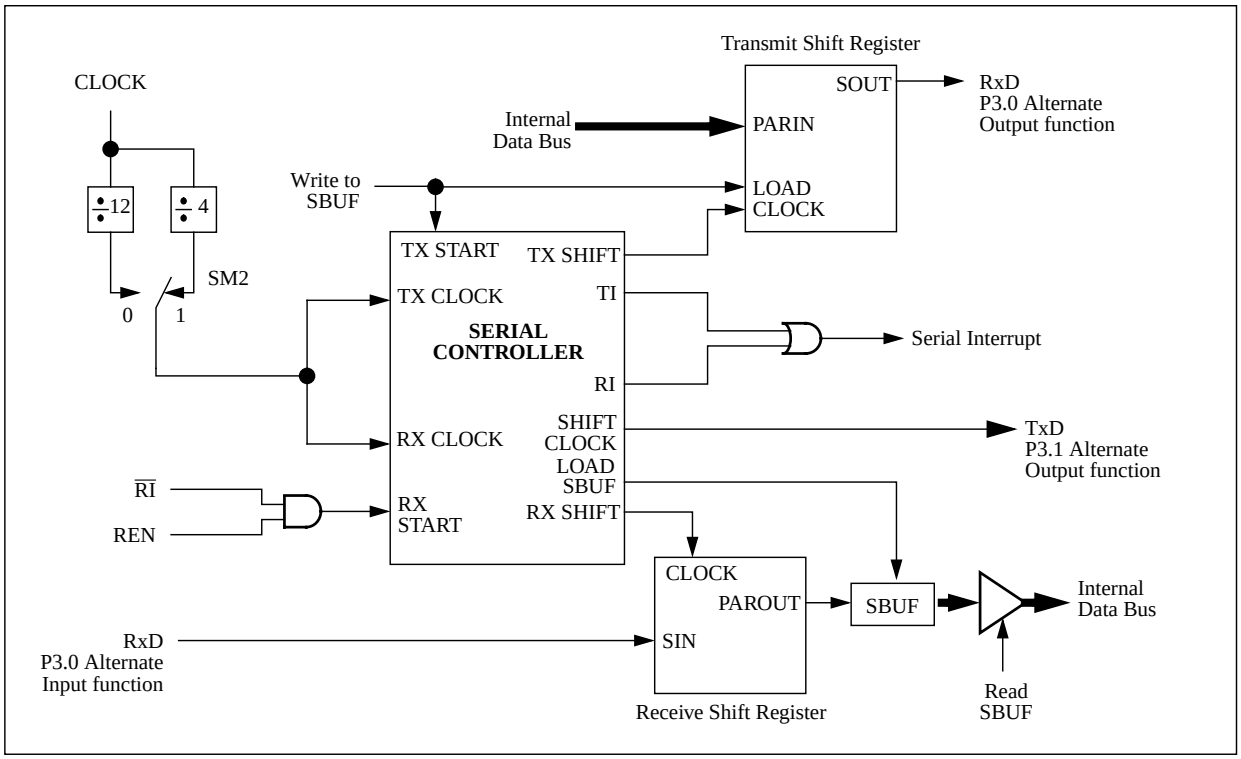


Figure-23 Serial Port Mode 0

Mode 1

In this mode 10 bits are transmitted (on TxD) or received (on RxD). The frame consists of a start bit (0), 8 data

bits (LSB first), and a stop bit (1). On reception, the stop bit is placed into RB8. In the CS6208, the baud rate is determined by the Timer 1 overflow rate, and so it can be controlled by the user. The figure below gives the simplified functional block for Mode 1.

Transmission begins with a write to SBUF. The serial data is brought out on to TxD pin at C1 following the first roll-over of divide by 16 counter. The next bit is placed on TxD pin at C1 following the next rollover of the divide by 16 counter. Thus the transmission is synchronized to the divide by 16 counter and not directly to the write to SBUF signal. After all 8 bits of data are transmitted, the stop bit is transmitted. The TI flag is set in the C1 state after the stop bit has been put out on TxD pin. This will be at the 10th rollover of the divide by 16 counter after a write to SBUF.

Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. The 1-to-0 detector continuously monitors the RxD line sampling it at the rate of 16 times the selected baud rate. When a falling edge is detected, the divide by 16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide by 16 counter.

The 16 states of the counter effectively divide the bit time into 16 slices. The bit detection is done on a best of three basis. The bit detector samples the RxD pin, at the 8th, 9th and 10th counter states. By using a majority 2 of 3 voting system, the bit value is selected. This is done to improve the noise rejection feature of the serial port. If the first bit detected after the falling edge of RxD pin, is not 0, then it indicates an invalid start bit, and the reception is immediately aborted. the serial port again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the SBUF.

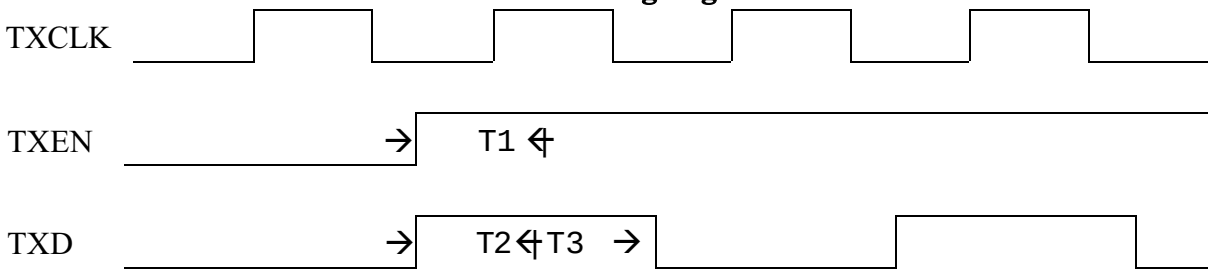
After shifting in 8 data bits, there is one more shift to do, after which the SBUF and RB8 are loaded and RI is set. However certain conditions must be met before, The loading and setting of RI can be done.

1. RI must be 0 and
2. Either SM2 = 0, or the received stop bit = 1.

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Else the received frame may be lost. After the middle of the stop bit, the receiver goes back to looking for a 1-to-0

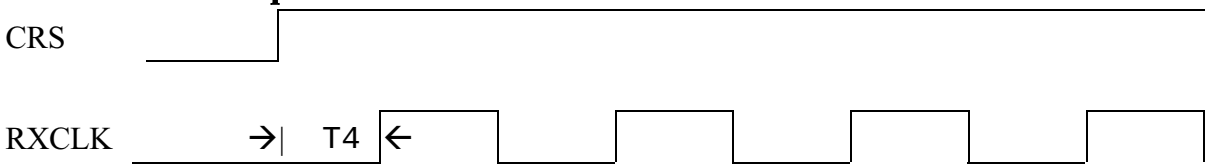
Transmit timing (start of packet)

1. **T1: Transmit enable setup time from rising edge of TXCLK =40ns**
2. **T2: Transmit data setup time from rising edge of TXCLK=40ns**
3. **T3: Transmit data hold time from rising edge of TXCLK=60ns**



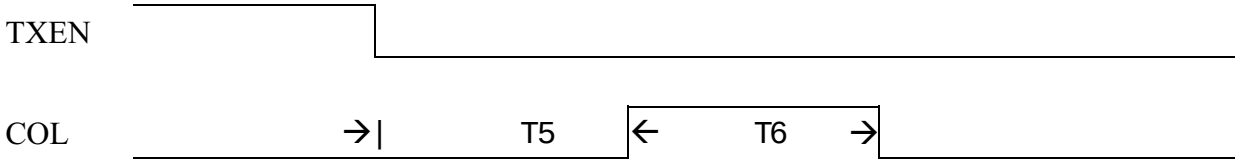
Receive timing (start of packet)

1. T4: Decoder acquisition time=529ns



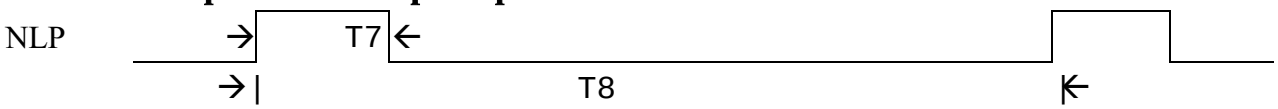
Heartbeat timing

1. **T5: CD heartbeat delay=1.41us**
2. **T6: CD heartbeat duration=892ns**



Normal link pulse timing

1. **T7: Clock , Data pulse width=83ns~100ns**
2. **T8: Clock pulse to clock pulse period=16.4ms**



detected, then the rest of the bits are also detected and shifted into the SBUF.

After shifting in 8 data bits, there is one more shift to do, after which the SBUF and RB8 are loaded and RI is set. However certain conditions must be met before, The loading and setting of RI can be done.

1. RI must be 0 and

2. Either SM2 = 0, or the received stop bit = 1.

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Else the received frame may be lost. After the middle of the stop bit, the receiver goes back to looking for a 1-to-0 transition on the RxD pin.

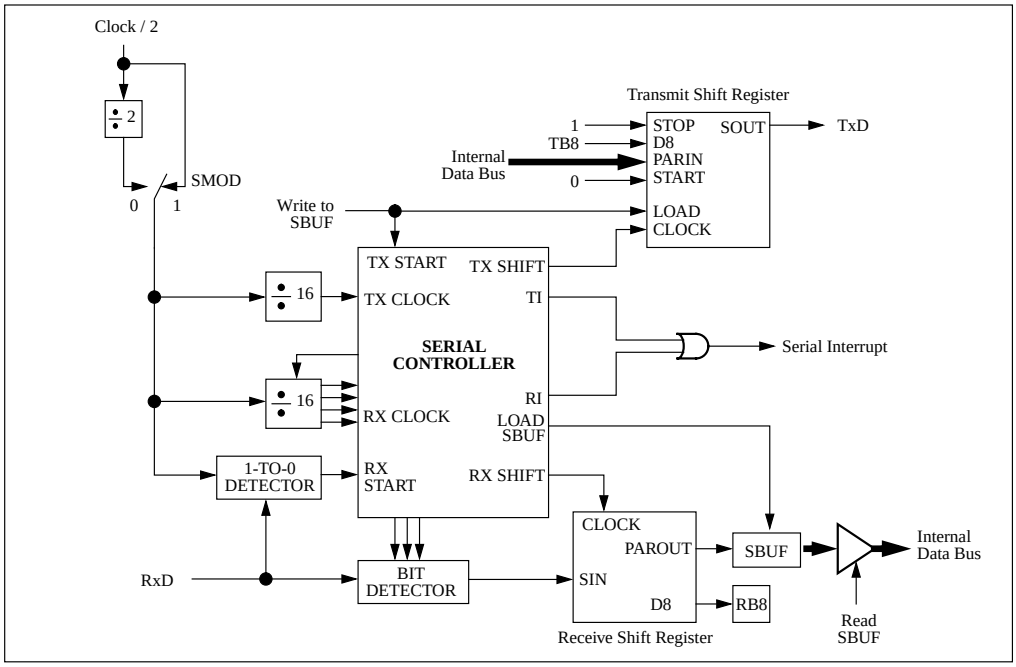


Figure-25 Serial Port Mode 2

Mode 3

Mode 3 is exactly similar in operation to Mode 2, except for the baud rate. In Mode 3 the baud rate is determined by the Timer 1 overflow. The figure shows the functional description of serial port in Mode 3.

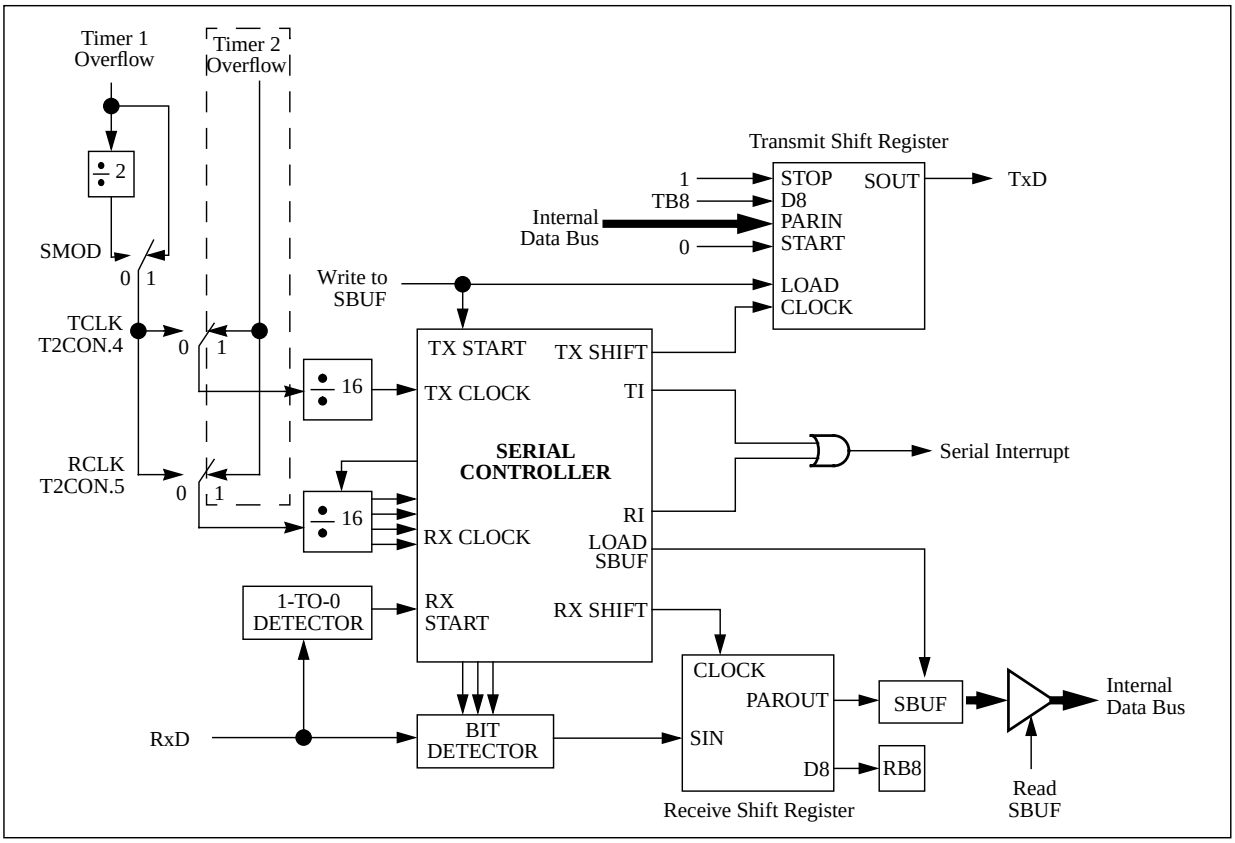


Figure-26 Serial Port Mode 3

Framing Error Detection

A Frame Error occurs when a valid stop bit is not detected. This could indicate incorrect serial data communication. Typically the frame error is caused due to noise and contention on the serial communication line. The CS6208 has the facility to detect such framing errors and set a flag which can be checked by software.

The Frame Error FE bit is located in SCON.0. This bit is normally used as SM0 in the standard 8051 family. However in the CS6208 it serves a dual function and is called SM0/FE. There are actually two separate flags, on for SM2 and the other for FE. The flag that is actually accessed as SCON.0 is determined by SMOD0 (PCON.6) bit. When SMOD0 is set to 1, then the FE flag is indicated in SM0/FE. When SMOD0 is set to 0, then the SM0 flag is indicated in SM0/FE.

The FE bit is set to 1 by hardware but must be cleared by software. Note that SMOD0 must be 1 while reading or writing to FE. If FE is set, then any following frames received without any error will not clear the FE flag. The clearing has to be done by software.

Multiprocessor Communications

Multiprocessor communications makes use of the 9th data bit in modes 2 and 3. In the CS6208, the RI flag is set only if the received byte corresponds to the Given or Broadcast address. This hardware feature eliminates the software overhead required in checking every received address, and greatly simplifies the programmers task.

In the multiprocessor communication mode, the address bytes are distinguished from the data bytes by transmitting the address with the 9th bit set high. When the master processor wants to transmit a block of data to one of the slaves, it first sends out the address of the targeted slave (or slaves). All the slave processors should have their SM2 bit set high when waiting for an address byte. This ensures that they will be interrupted only by the reception of a address byte. The Automatic address recognition feature ensures that only the addressed slave will be interrupted. The address comparison is done in hardware not software.

The addressed slave the clears it SM2 bit, thereby clearing the way to receive data bytes. With SM2 = 0, the slave will be interrupted on the reception of every single complete frame of data. The unaddressed slave will be unaffected, as they will be still waiting for their address.

In Mode 1, the 9th bit is the stop bit which is 1 in case of a valid frame. If SM2 is 1, then RI is set only if a valid frame is received and the received byte matches the Given or Broadcast address. In Mode 0, the SM2 bit has relevance for multiprocessor communications mode.

The Master processor can selectively communicate with groups of slaves by using the Given Address. All the slaves can be addressed together using the Broadcast Address. The addresses for each slave are defined by the SADDR and SADEN SFRs. The slaves address is an 8-bit value specified in the SADDR SFR. The SADEN SFR is actually a mask for the byte value in SADDR.

If a bit position in SADEN is 0, then the corresponding bit position in SADDR is don't care. Only those bit positions in SADDR, whose corresponding bits in SADEN are 1, are used to obtain the Given Address. This gives the user flexibility to address multiple slaves without changing the salve address in SADDR.

The following example shows how the user can define the Given Address to address different slaves

Slave 1:

SADDR	1010 0100
SADEN	1111 1010
Given	1010 0x0x

Slave 2:

SADDR	1010 0111
SADEN	1111 1001
Given	1010 0xx1

The Given address for slave 1 and 2 differ in the in the LSB. For slave 1, it is a don't care, while for slave 2 it is 1. Thus to communicate only with slave 1, the master must send an address with LSB = 0 (1010 0000). Similarly the bit 1 position is 0 for slave 1 and don't care for slave 2. Hence to communicate only with slave 2 the master has to transmit an address with bit 1 = 1 (1010 0011). If the master wishes to communicate to both slaves simultaneously, then the address must have bit 0 = 1 and bit 1 = 0. The bit 3 position is don't care for both the slaves. This allows two different address to select both slaves (1010 0001 and 1010 0101).

The master can communicate with all the slaves simultaneously with the Broadcast Address. This address is formed from the logical ORing of the SADDR and SADEN SFRs. The zeros in the result are defined as don't cares In most cases the Broadcast Address is FFh.

The SADDR and SADEN SFRs are located at address A9h and B9h respectively. On reset, these two SFRs are initialized to 00h. This results in Given Address and Broadcast Address being set as XXXX XXXX. This effectively removes the multiprocessor communications feature, since any selectivity is disabled.

Baud Rates

Mode 0:

In Mode 0 the baud clock is derived from the system clock. There are two options for baud clock. It can be set to ? of clock speed or 1/12 of clock speed. The selection of the baud rate is done by the SM2 bit. If SM2 is 0, then the baud rate is fixed at 1/12 of the clock frequency. If SM2 is 0, then he baud rate is fixed at ? of the clock frequency. After a reset SM2 is initialized to 0, therefore unless the user sets SM2 with software, the baud rate in Mode 0 will be similar to the standard 8051 family.

Mode 2:

In Mode 0 the baud clock is derived from the system clock. There are two options for baud clock. It can be set to 1/32 of clock speed or 1/64 of clock speed. The selection of the baud rate is done by the SMOD bit in PCON SFR. If SMOD is 0 then the baud rate is 1/64 of the oscillator frequency. If the bit is set to 1, then the baud rate is 1/32 of the oscillator frequency.

$$\mathbf{Mod2BaudRate} = \frac{2^{\mathbf{SMOD}}}{64} \times \mathbf{OscillatorFrequency}$$

Mode 1 and 3:

The baud rates in Mode 1 and 3 are determined by the overflow rates of Timer 1 or Timer 2. The Timer 1 overflow clock is divided by 16 to generate the actual transmit and receive baud clock. In the CS6208, the user can select wether the Timer 1, when operating as a timer, counts at the rate of 1/4 of clock or 1/12 of clock. If the 1/12 clock option is used then compatibility with the standard 8051 family can be maintained.

Using Timer 1 to generate baud rates.

When Timer 1 is used as a baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD.

$$\mathbf{Mode}1,3\mathbf{BaudRate} = \frac{2^{\mathbf{SMOD}}}{32} \times \mathbf{Timer}1\mathbf{OverflowRate}$$

The Timer 1 interrupt should be disabled in this mode. The timer itself can be configured as either "timer" or "counter", in any of its 3 operating modes. Commonly, Timer 1 is configured as a timer in auto-reload mode. In such a case the baud rate is given by

$$\mathbf{Mode}1,3\mathbf{BaudRate} = \frac{2^{\mathbf{SMOD}}}{32} \times \frac{\mathbf{ClockFrequency}}{12 \times (256 - \mathbf{TH}1)}$$

Note that the 12 in the denominator is if Timer 1 is counting once every 12 clocks. If T1M (CKCON.4) is set to 1, the timer counts once every 4 machine cycles. In this case the 12 in the denominator is replaced with 4.

It is also possible to achieve very low baud rates by putting Timer 1 in Mode1 as a 16 bit timer and enabling its interrupt. The Timer interrupt is used to reload the Timer 1 with a 16 bit value using software methods.

To use Timer 2 as baud rate generator for the serial port, the timer is configured in auto-reload mode. Then either TCLK or RCLK bits or both are set to logic 1. With TCLK = 1, thee timer 2 acts as baud rate generator for the transmitter. and RCLK = 1 selects timer 2 as the baud rate generator for the receiver. Thus the serial port can have the transmit and receive operating at different baud rates.

When using the timer 2 to generate the baud rates, the formula will be as follows.

$$\textbf{Mode}1,3\textbf{BaudRate} = \frac{\textbf{ClockFrequency}}{32 \times (65536 - \textbf{RCAP}2\textbf{H}, \textbf{RCAP}2\textbf{L})}$$

TIMED ACCESS PROTECTION

The CS6208 has several new features like the Watchdog timer and the Power on-fail reset flag which are crucial to proper operation of the system. If left unprotected, errant code may write to the Watchdog control bits resulting in incorrect operation and loss of control. In order to prevent this, the CS6208 has a protection scheme which controls the write access to critical bits. This protection scheme is done using a timed access.

In this method, the bits which are to be protected have timed write enable window. A write is successful only if this window is active, else the write will be unsuccessful. This write enable window is open for 3 machine cycles if certain conditions are met. After 3 machine cycles, this window automatically closes. The window is opened by writing AAh and immediately 55h to the Timed Access TA SFR. This SFR is located at address C7h. The suggested code for opening the timed access window is

```
MOV     TA, #0AAh
MOV     TA, #055h
```

When the software writes AAh to the TA SFR, a counter is started. This counter waits for 3 machine cycles looking for a write of 55h to TA. If the second write (55h) occurs within 3 machine cycles of the first write (AAh), then the timed access window is opened. It remains open foe 3 machine cycles, during which the user may write to the protected bits. Once the window closes the procedure must be repeated to access the other protected bits.

Examples of Timed Assessing are shown below

Example 1: Valid access

```
MOV     TA, #0AAh       3 M/C
MOV     TA, #055h       3 M/C
MOV     WDCON, #00h     3 M/C
```

Example 2: Valid assess

```
MOV     TA, #0AAh       3 M/C
MOV     TA, #055h       3 M/C
NOP                     1 M/C
SETB    EWT             2 M/C
```

Example 3: Invalid access

```
MOV     TA, #0AAh       3 M/C
MOV     TA, #055h       3 M/C
NOP                     1 M/C
NOP                     1 M/C
CLR     POR             2 M/C
```

Example 4: Invalid Access

```
MOV     TA, #0AAh       3 M/C
NOP                     1 M/C
MOV TA, #055h           3 M/C
SETB EWT                2 M/C
```

In the first two examples, the writing to the protected bits is done before the 3 machine cycle window closes. In the third case however, the writing to the protected bit occurs after the window has closed and so there is effectively no change in the status of the protected bit. In the third example, the second write to TA occurs 4 machine cycles after the first write, therefore the timed access window in not opened at all, and the write to the protected bit fails.

AC ELECTRICAL CHARACTERISTICS

PARAMETER	SYMBOL	VARIABLE CLOCK MIN	VARIABLE CLOCK MAX	UNITS
Oscillator Frequency	$1/t_{CLCL}$	0	40	MHz
ALE Pulse Width	t_{LHLL}	$1.5t_{CLCL}$ - 5		ns
Address Valid to ALE Low	t_{AVLL}	$.5t_{CLCL}$ - 5		ns
Address Hold After ALE Low	t_{LLAX1}	$.5t_{CLCL}$ - 5		ns
Address Hold After ALE Low for MOVX WR	t_{LLAX2}	$.5t_{CLCL}$ - 5		ns
ALE Low to Valid Instruction In	t_{LLIV}		$2.5t_{CLCL}$ - 20	ns
ALE Low to $\overline{PSEN}$ Low	t_{LLPL}	$.5t_{CLCL}$ - 5		ns
$\overline{PSEN}$ Pulse Width	t_{PLPH}	$2.0t_{CLCL}$ - 5		ns
$\overline{PSEN}$ Low to Valid Instruction In	t_{PLIV}		$2.0t_{CLCL}$ - 20	ns
Input Instruction Hold After $\overline{PSEN}$	t_{PXIX}	0		ns
Input Instruction Float After $\overline{PSEN}$	t_{PXIZ}		t_{CLCL} - 5	ns
Port 0 Address to Valid Instr. In	t_{AVIV1}		$3.0t_{CLCL}$ - 20	ns
Port 2 Address to Valid Instr. In	t_{AVIV2}		$3.5t_{CLCL}$ - 20	ns
$\overline{PSEN}$ Low to Address Float	t_{PLAZ}		0	ns
Data Hold After Read	t_{RHDX}	0		ns
Data Float After Read	t_{RHDZ}		t_{CLCL} - 5	ns
$\overline{RD}$ Low to Address Float	t_{RLAZ}		$.5t_{CLCL}$ - 5	ns
$\overline{RD}$ or $\overline{WR}$ High to ALE High	t_{WHLH}	0	see table	ns
$\overline{RD}$ Pulse Width	t_{RLRH}	see table		ns
$\overline{WR}$ Pulse Width	t_{WLWH}	see table		ns
$\overline{RD}$ Low to Data Valid In	t_{RLDV}		see table	ns
ALE Low to Data Valid In	t_{LLDV}		see table	ns
Address to Valid Data In	t_{AVDV}		see table	ns
ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	t_{LLWL}	see table	see table	ns
Port 0 Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	t_{AVWL1}	see table		ns
Port 2 Valid to $\overline{RD}$ or $\overline{WR}$ Low	t_{AVWL2}	see table		ns
Data Valid to $\overline{WR}$ Transition	t_{QVWX}	see table		ns
Data Hold After Write	t_{WHQX}	see table		ns

MOVX CHARACTERISTICS USING STRETCH MEMORY CYCLES

PARAMETER	SYMBOL	VARIABLE CLOCK MIN	VARIABLE CLOCK MAX	UNITS	STRETCH
Data Access ALE Pulse Width	t_{LLHL2}	$1.5t_{CLCL} - 5$ $2.0t_{CLCL} - 5$		ns	$t_{MCS}=0$ $t_{MCS}>0$
Address Hold After ALE Low for MOVX write	t_{LLAX2}	$0.5t_{CLCL} - 5$		ns	
RD Pulse Width	t_{RLRH}	$2.0t_{CLCL} - 5$ $t_{MCS} - 10$		ns	$t_{MCS}=0$ $t_{MCS}>0$
WR Pulse Width	t_{WLWH}	$2.0t_{CLCL} - 5$ $t_{MCS} - 10$		ns	$t_{MCS}=0$ $t_{MCS}>0$
RD Low to Valid Data In	t_{RLDV}		$2.0t_{CLCL} - 20$ $t_{MCS} - 20$	ns	$t_{MCS}=0$ $t_{MCS}>0$
Data Hold after Read	t_{RHDX}	0		ns	
Data Float after Read	t_{RHDZ}		$t_{CLCL} - 5$ $2.0t_{CLCL} - 5$	ns	$t_{MCS}=0$ $t_{MCS}>0$
ALE Low to Valid Data In	t_{LLDV}		$2.5t_{CLCL} - 5$ $t_{MCS} + 2t_{CLCL} - 40$	ns	$t_{MCS}=0$ $t_{MCS}>0$
Port 0 Address to Valid Data In	t_{AVDV1}		$3.0t_{CLCL} - 20$ $2.0t_{CLCL} - 5$	ns	$t_{MCS}=0$ $t_{MCS}>0$
Port 2 Address to Valid Data In	t_{AVDV2}		$3.5t_{CLCL} - 20$ $2.5t_{CLCL} - 5$	ns	$t_{MCS}=0$ $t_{MCS}>0$
ALE Low to RD or WR Low	t_{LLWL}	$0.5t_{CLCL} - 5$ $1.5t_{CLCL} - 5$	$0.5t_{CLCL} + 5$ $1.5t_{CLCL} + 5$	ns	$t_{MCS}=0$ $t_{MCS}>0$
Port 0 Address to RD or WR Low	t_{AVWL}	$t_{CLCL} - 5$ $2.0t_{CLCL} - 5$		ns	$t_{MCS}=0$ $t_{MCS}>0$
Port 2 Address to RD or WR Low	t_{AVWL2}	$1.5t_{CLCL} - 5$ $2.5t_{CLCL} - 5$		ns	$t_{MCS}=0$ $t_{MCS}>0$
Data Valid to WR Transition	t_{QVWX}	-5 $1.0t_{CLCL} - 5$		ns	$t_{MCS}=0$ $t_{MCS}>0$
Data Hold after Write	t_{WHQX}	$t_{CLCL} - 5$ $2.0t_{CLCL} - 5$		ns	$t_{MCS}=0$ $t_{MCS}>0$
RD Low to Address Float	t_{RLAZ}		$0.5t_{CLCL} - 5$	ns	
RD or WR high to ALE high	t_{WHLH}	0 $2.0t_{CLCL} - 5$	10 $2.0t_{CLCL} - 5$	ns	$t_{MCS}=0$ $t_{MCS}>0$

Note: t_{MCS} is a time period related to the Stretch memory cycle selection. The following table shows the time period of t_{MCS} for each selection of the Stretch value.

M2	M1	M0	MOVX Cycles	t_{MCS}
0	0	0	2 machine cycles	0
0	0	1	3 machine cycles	2 t_{CLCL}
0	1	0	4 machine cycles	6 t_{CLCL}
0	1	1	5 machine cycles	10 t_{CLCL}
1	0	0	6 machine cycles	14 t_{CLCL}
1	0	1	7 machine cycles	18 t_{CLCL}
1	1	0	8 machine cycles	22 t_{CLCL}
1	1	1	9 machine cycles	26 t_{CLCL}

EXTERNAL CLOCK CHARACTERISTICS

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Clock High Time	t_{CHCX}	12.5			ns	
Clock Low Time	t_{CLCX}	12.5			ns	
Clock Rise Time	t_{CLCH}			10	ns	
Clock Fall Time	t_{CLCL}			10	ns	

SERIAL PORT MODE 0 TIMING CHARACTERISTICS

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Serial Port Clock Cycle Time SM2=0 12 clocks per cycle SM2=1 4 clocks per cycle	t_{XLXL}		$12t_{CLCL}$ $4t_{CLCL}$		ns	
Output Data Setup to Clock Rising Edge SM2=0 12 clocks per cycle SM2=1 4 clocks per cycle	t_{QVXH}		$10t_{CLCL}$ $3t_{CLCL}$		ns	
Output Data Hold to Clock Rising Edge SM2=0 12 clocks per cycle SM2=1 4 clocks per cycle	t_{XHQX}		$2t_{CLCL}$ t_{CLCL}		ns	
Input Data Hold after Clock Rising SM2=0 12 clocks per cycle SM2=1 4 clocks per cycle	t_{XHDX}		t_{CLCL} t_{CLCL}		ns	

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Clock Rising Edge to Input Data Valid SM2=0 12 clocks per cycle SM2=1 4 clocks per cycle	t_{XHDV}		$11t_{CLCL}$ $3t_{CLCL}$		ns	

EXPLANATION OF LOGIC SYMBOLS

In order to maintain compatibility with the original 8051 family, this device specifies the same parameter as such device, using the same symbols. The explanation of the symbols is as follows.

t	Time
A	Address
C	Clock
D	Input Data
H	Logic level high
L	Logic level low
I	Instruction
P	PSEN
Q	Output Data
R	RD signal
V	Valid
W	WR signal
X	No longer a valid state
Z	Tri-state

PROGRAM MEMORY READ CYCLE

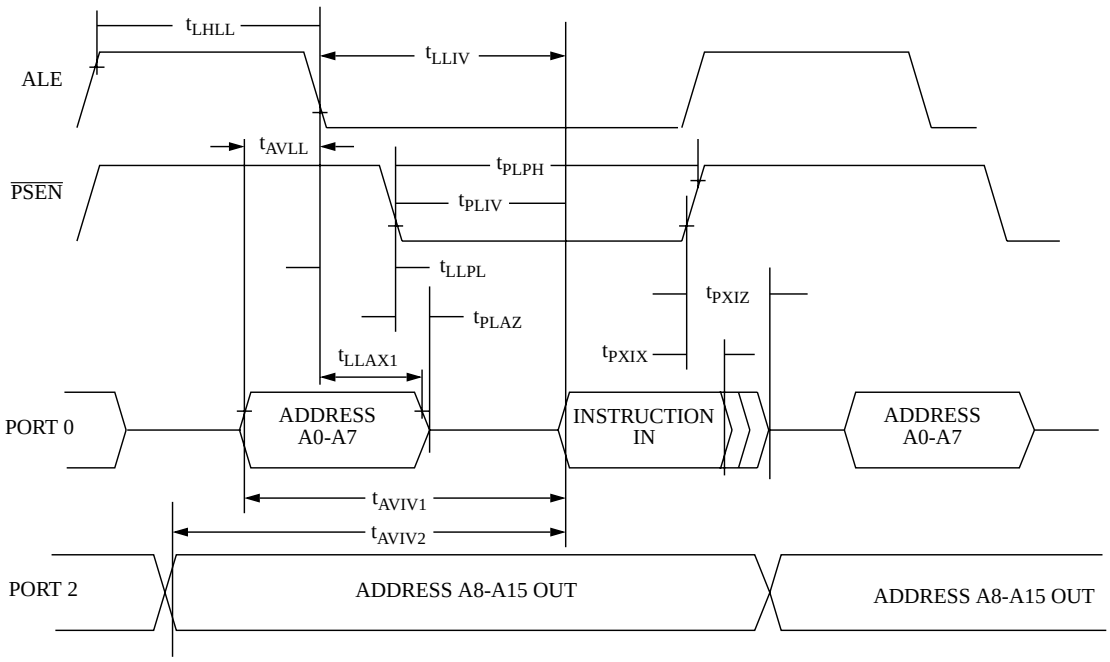


Figure-27 Program Memory Read Cycle

AC ELECTRICAL CHARACTERISTICS

PARAMETER	SYMBOL	PLL=22MHz		PLL=60MHz		UNITS
		min	max	min	max	
ALE Pulse Width	t_{LHLL}	67.6				ns
ALE Low to Valid Instruction In	t_{LLIV}		44			ns
Address Valid to ALE Low	t_{AVLL}	16				ns
/PSEN Pulse Width	t_{PLPH}	88				ns
/PSEN Low to Valid Instruction In	t_{PLIV}	18				ns
ALE Low to /PSEN Low	t_{LLPL}	24				ns
/PSEN Low to Address Float	t_{PLAZ}		11.8			ns
Address Hold After ALE Low	t_{LLAX1}	34				ns
Input Instruction Float After /PSEN	t_{PXIZ}					ns
Input Instruction Hold After /PSEN	t_{PXIX}	47.8				ns
Port 0 Address to Valid Instr. In	t_{AVIV1}	51.6				ns
Port 2 Address to Valid Instr. In	t_{AVIV2}	51.6				ns
						ns

DATA MEMORY READ CYCLE

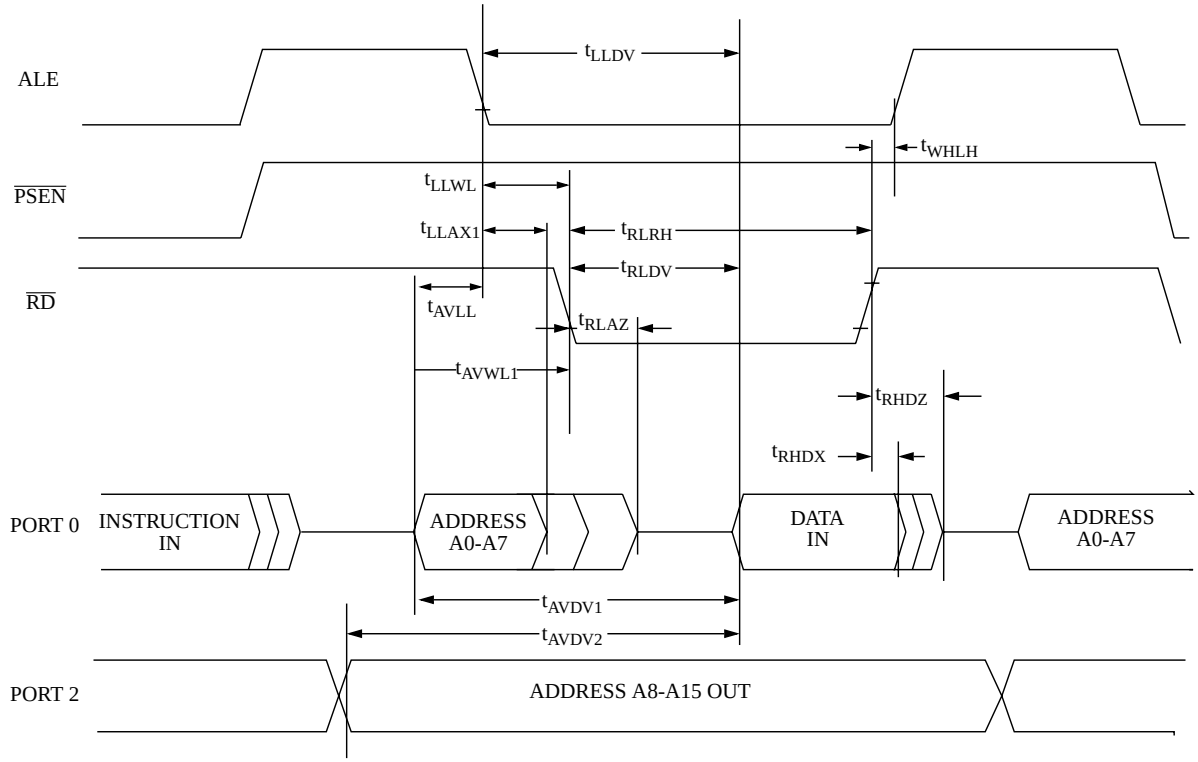


Figure-28 Data Memory Read Cycle

AC ELECTRICAL CHARACTERISTICS

PARAMETER	SYMBOL	PLL=22MHz		PLL=60MHz		UNITS
		min	max	min	max	
ALE Low to Data Valid In	t_{LLDV}	240				ns
ALE Low to /RD or /WR Low	t_{LLWL}	72				ns
/RD or /WR High to ALE High	t_{WHLH}		40			ns
Address Hold After ALE Low	t_{LLAX1}	239				ns
/RD Pulse Width	t_{RLRH}	180				ns
/RD Low to Data Valid In	t_{RLDV}	167.6				ns
Address Valid to ALE Low	t_{AVLL}	19				ns
/RD Low to Address Float	t_{RLAZ}	167				ns
Port 0 Address to /RD or /WR In	t_{AVWL1}	84				ns
Data Folat After Read	t_{RHDZ}					ns
Data Hold After Read	t_{RHDX}	19				ns
Port 0 Address to Valid Data In	t_{AVDV1}	255				ns
Port 2 Address to Valid Data In	t_{AVDV2}	255				ns

DATA MEMORY WRITE CYCLE

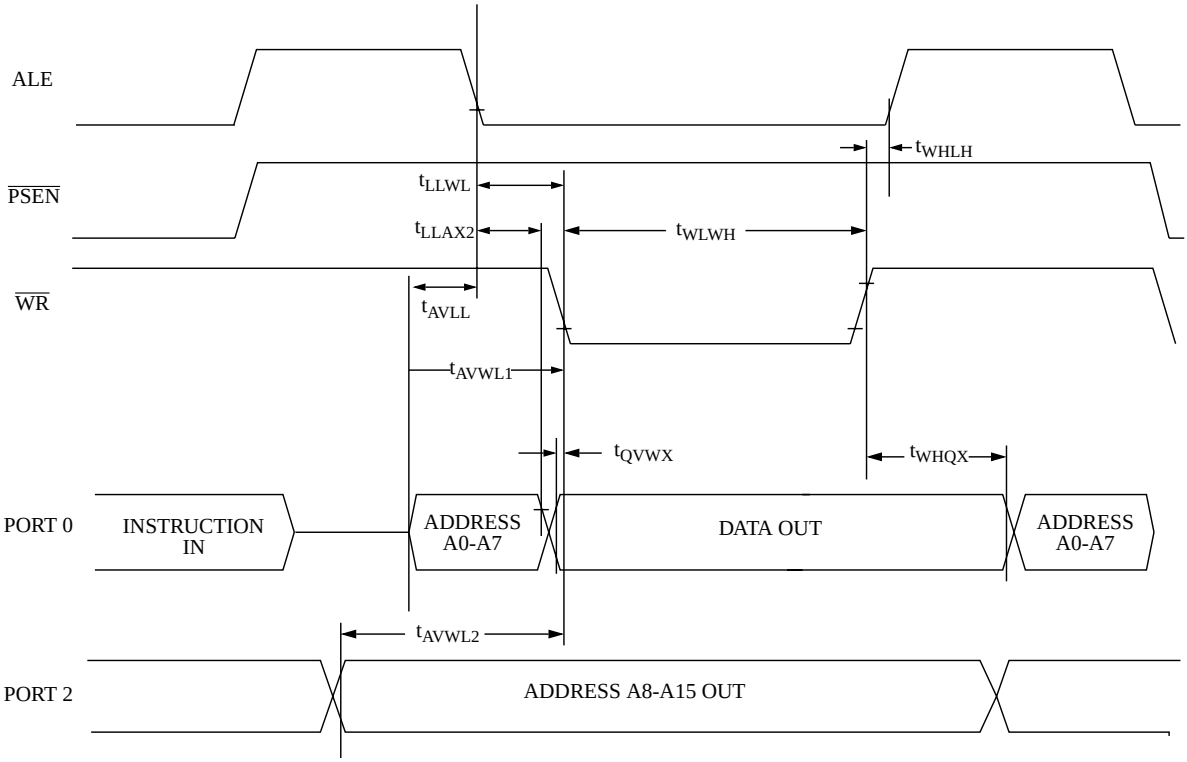


Figure-29 Data Memory Write Cycle

AC ELECTRICAL CHARACTERISTICS

PARAMETER	SYMBOL	PLL=22MHz		PLL=60MHz		UNITS
		min	max	min	max	
/RD or /WR High to ALE High	t_{WHLH}		42			ns
ALE Low to /RD or /WR Low	t_{LLWL}	66				ns
Address Hold After ALE Low for MOVX /WR	t_{LLAX2}	36				ns
/WR Pulse Width	t_{WLWH}	180				ns
Address Valid to ALE Low	t_{AVLL}	8				ns
Port 0 Address to /RD or /WR In	t_{AVWL1}	67.5				ns
Data Valid to /WR Transition	t_{QVWX}	32				ns
Data Hold After /WR	t_{WHQX}	96				ns
Port 2 Address to /RD or /WR Low	t_{AVWL2}	75				ns

DATA MEMORY WRITE WITH STRETCH = 1

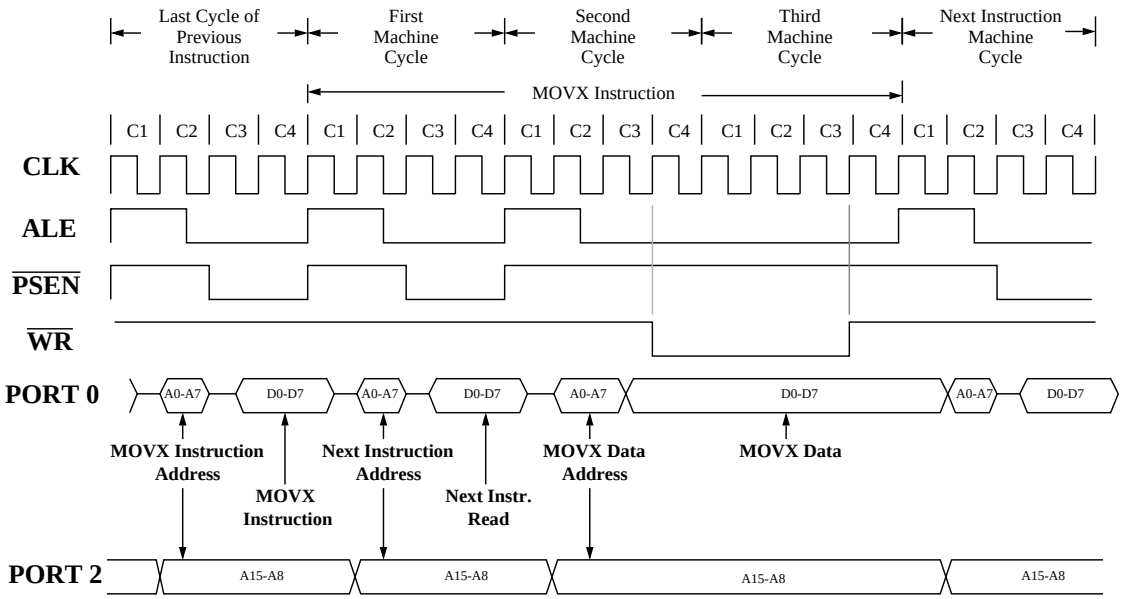


Figure-30 Data Memory Write with stretch = 1

Figure-31

DATA MEMORY WRITE WITH STRETCH = 2

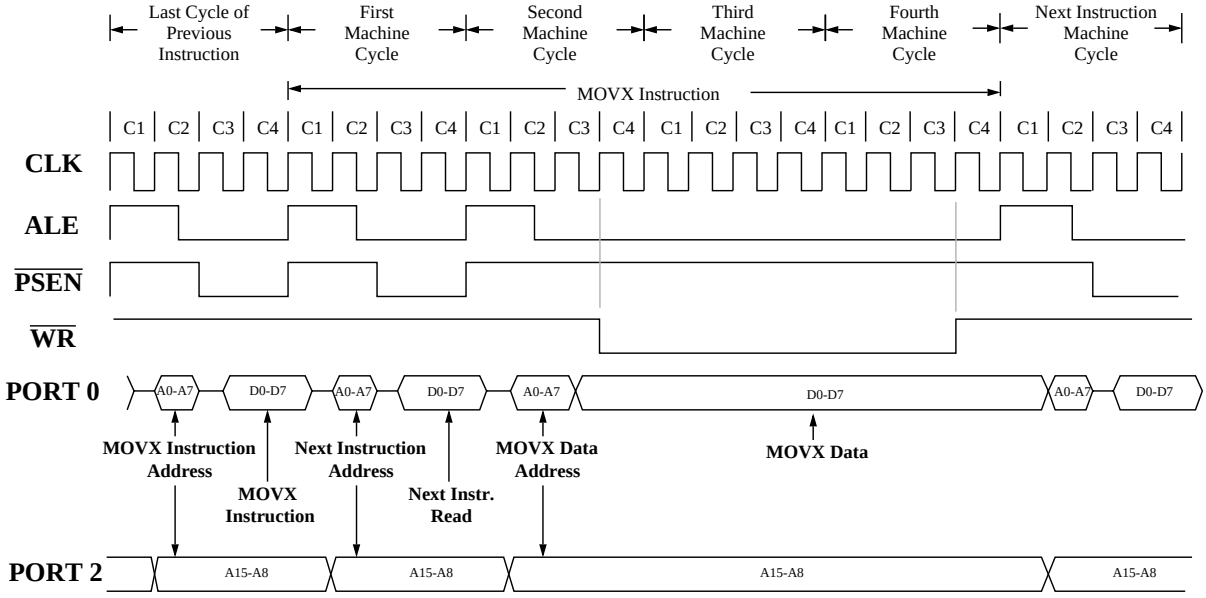


Figure-32 Data Memory Write with STRETCH = 2

EXTERNAL CLOCK DRIVE

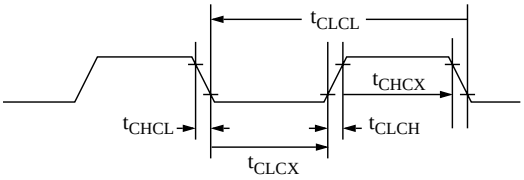


Figure-33 External Clock Drive

SERIAL PORT MODE 0 TIMING

SERIAL PORT 0 (SYNCHRONOUS MODE)

HIGH SPEED OPERATION SM2 = 1 => TXD CLOCK = XTAL/4

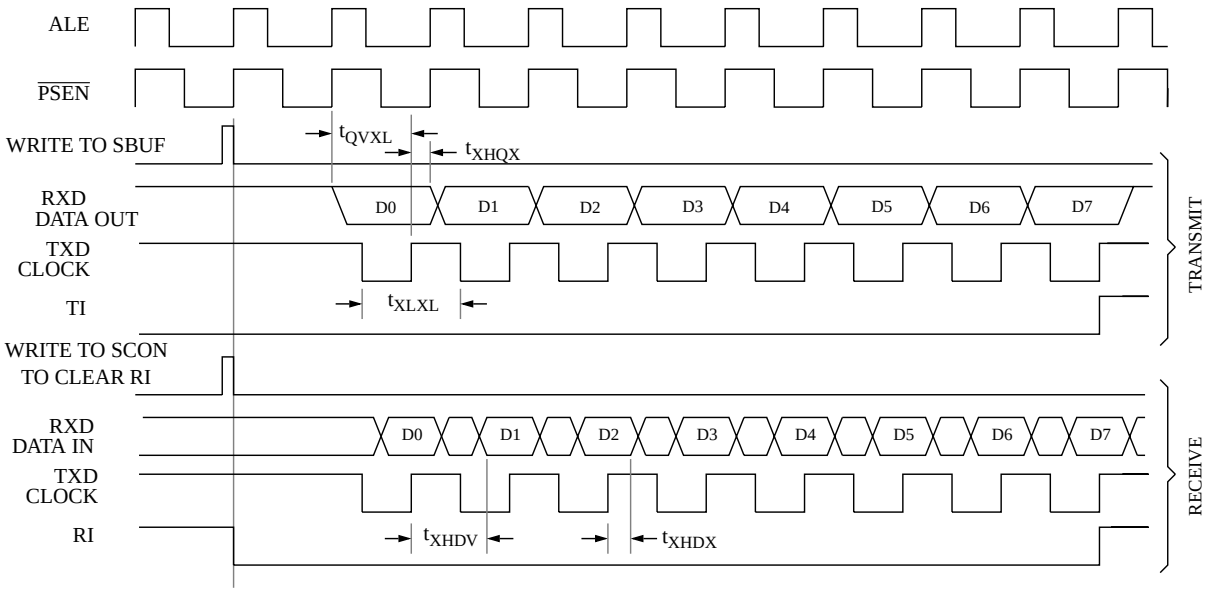


Figure-34 Serial Port Mode 0 Timing

SERIAL PORT 0 (SYNCHRONOUS MODE)

SM2 =0 => TXD CLOCK = XTAL/12

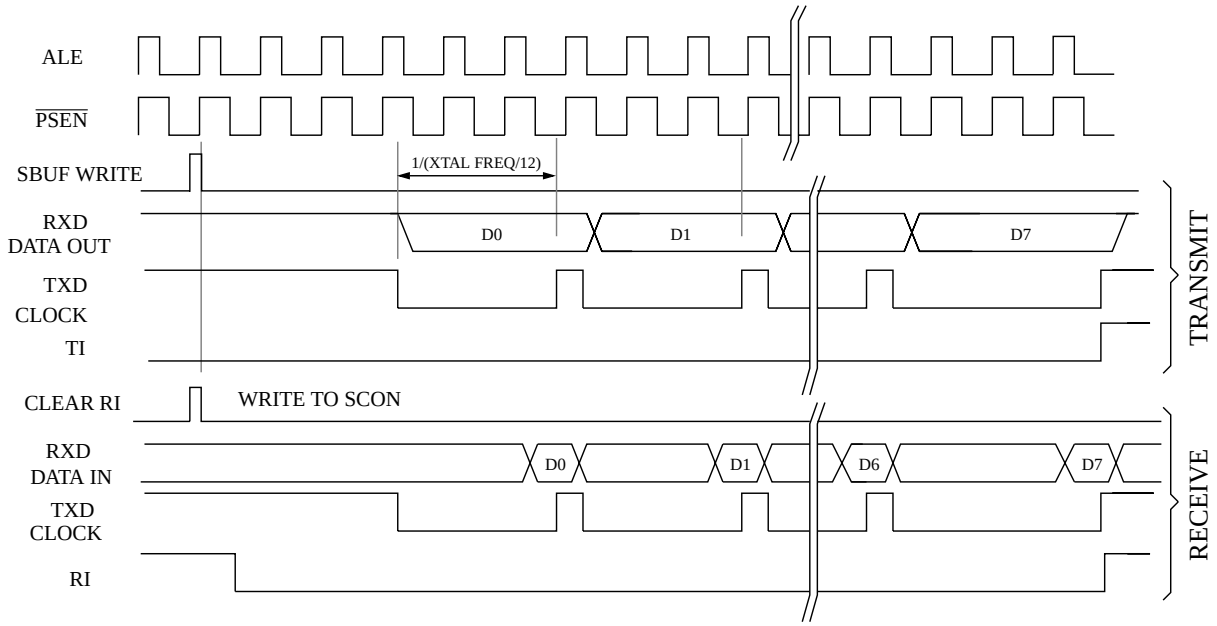


Figure-35 Serial Port 0

	MIN		TYPICAL		MAX				
Frequency	**Voltage**	**Current**	**Voltage**	**Current**	**Voltage**	**Current**	**Idle Mode**	**Power Down**	**PHY Off**
22Mhz	2.78V	55mA	3.3V	69mA	5V	125mA	61mA	17mA	64mA
33Mhz	2.78V	97mA	3.3V	97mA	5V	164mA	81mA	17mA	92mA
44Mhz	2.97V	115mA	3.3V	115mA	5V	202mA	100mA	17mA	110mA
60Mhz	2.91V	126mA	3.3V	149mA	5V	258mA	128mA	17mA	144mA

Idle Mode: PCON = 0x01

Power Down Mode: PCON = 0x02

Ethernet PHY Off: PHYCTRL = 0x03

Figure 37: **Voltage and Power Consumption of the CS6208**

DC ELECTRICAL CHARACTERISTICS

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
High level output voltage	VoH	I_{oH} = 8ma	2.4			V
Low level output voltage	VoL	I_{oL} = 8 ma			0.4	V

Note1: V_{IH} min = 2.0V and V_{IL} max = 0.8V.

Note2: DC specification for all digital I/O signals, including: I/O port, address, data, PSEN, COL, CRS, RXC, and RXD.

Figure 38: **Digital DC Electrical Characteristics**

Appendix A: Interrupt Pseudocode for Integrated Peripherals (RAM Checksum, I2C1, I2C2, and MAC).

```
// Interrupt Vector Function for INT0 (0x0003)
void INT0 (void) interrupt 0
{
  if (INSTAR & 0x08)              // Ram Checksum Interrupt.
    {
      if (RCSCR & 0x02)
        checksum = {HRCSD, LRCSD};
      else if (RCSCR & 0x04)
        checksum failure.       // Set the Checksum error flag and poll from the main loop.
    }

    else if (INSTAR & 0x04)     // I2C2 Interrupt.
      I2C2 ISR here.
    else if (INSTAR & 0x02)     // I2C1 Interrupt.
      I2C1 ISR here.
    else                        // Enternal Interrupt 0.
      External Interrupt 0 Service Routine here.
}
```

Lit#: DSCS6208-01

Sales and Technical Support Contact Information.

For the latest contact and support information on Myson Century Semiconductor devices, please visit the Myson Century website at www.myson.com.tw or www.century-semi.com. These sites contain technical literature, local sales contacts, technical support and many other features.

Number 2, Industry East Road III
Hsinchu Taiwan, R.O.C.
Tel: 886 3 578 4866
Fax: 886 3 578 5002/4349
E-Mail: sales@myson.com.tw
Web Site: www.myson.com.tw

1485 Saratoga Ave. Suite 200
San Jose, California 95129
Tel: (408) 973 8388
Fax: (408) 973 9388
E-Mail: sales@century-semi.com
Web Site: www.century-semi.com