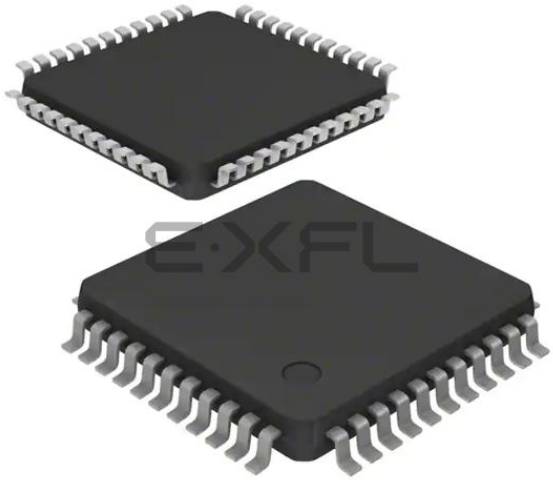


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | Z8 |
| Core Size | 8-Bit |
| Speed | 33MHz |
| Connectivity | EBI/EMI, UART/USART |
| Peripherals | - |
| Number of I/O | 24 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| EEPROM Size | - |
| RAM Size | 236 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-LQFP |
| Supplier Device Package | 44-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/zilog/z86c9333fsg |

## GENERAL DESCRIPTION (Continued)

There are 256 registers located on-chip and organized as 236 general-purpose registers, 16 control and status registers, and four I/O port registers. The register file can be divided into sixteen groups of 16 working registers each. Configuration of the registers in this manner allows the use of short format instructions; in addition, any of the individual registers can be accessed directly. There are an additional 17 registers implemented in the Expanded Register File in Banks D and E. Two of the registers may be used as general-purpose registers, while 15 registers supply the data and control functions for the Multiply/Divide Unit and Counter/Timer blocks.

Notes:
All Signals with a preceding front slash, "/", are active Low, e.g.: B/W (WORD is active Low); /B/W (BYTE is active Low, only).

Power connections follow conventional descriptions below:

| Connection | Circuit | Device |
|---|---|---|
| Power | $V_{CC}$ | $V_{DD}$ |
| Ground | GND | $V_{SS}$ |

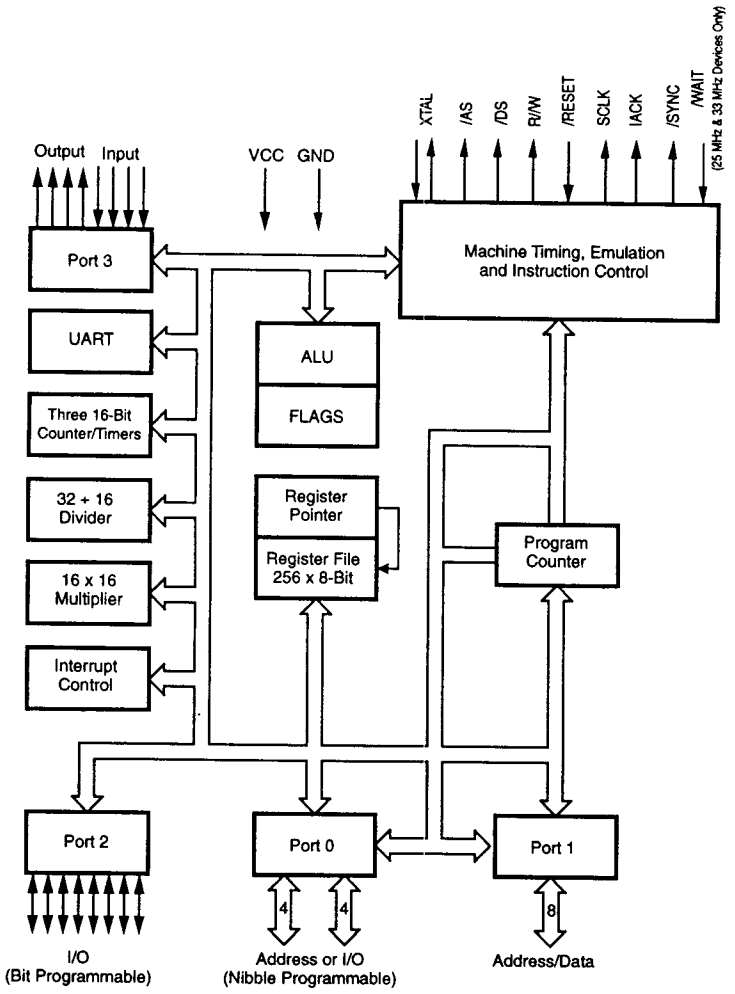



**Figure 1. Functional Block Diagram**

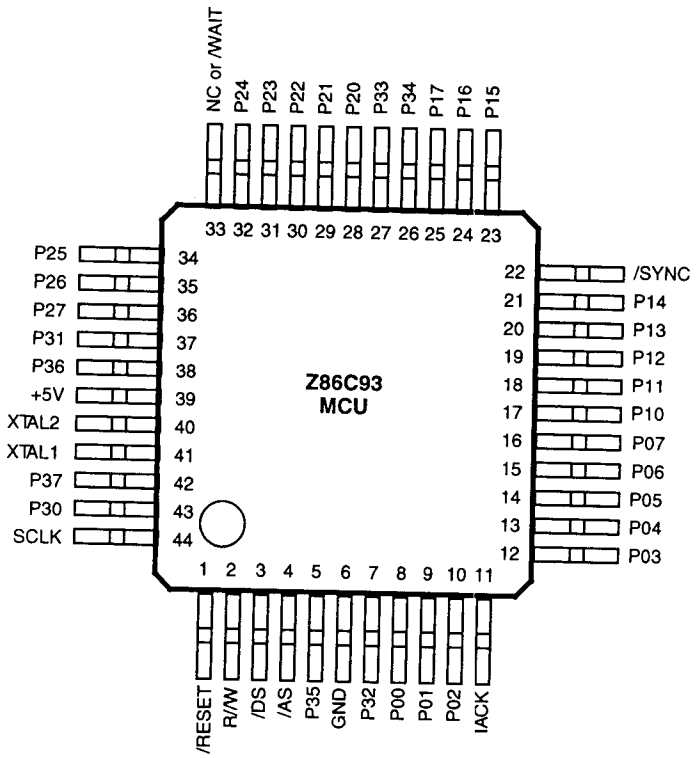



**Figure 5. 44-Pin QFP**

**Table 3. 44-Pin QFP Pin Identification**

| No | Symbol | Function | Direction |
|---|---|---|---|
| 1 | /RESET | Reset | Input |
| 2 | R/W | Read/Write | Output |
| 3 | /DS | Data Strobe | Output |
| 4 | /AS | Address Strobe | Output |
| 5 | P35 | Port 3 pin 5 | Input |
| 6 | GND | Ground GND | Input |
| 7 | P32 | Port 3 pin 2 | Input |
| 8-10 | P00-P02 | Port 0 pin 0,1,2 | In/Output |
| 11 | IACK | Int. Acknowledge | Output |
| 12-16 | P03-P07 | Port 0 pin 3,4,5,6,7 | In/Output |
| 17-21 | P10-P14 | Port 1 pin 0,1,2,3,4 | In/Output |
| 22 | /SYNC | Synchronize Pin | Output |
| 23-25 | P15-P17 | Port 1 pin 5,6,7 | In/Output |
| 26 | P34 | Port 3 pin 4 | Output |
| 27 | P33 | Port 3 pin 3 | Input |
| 28-32 | P20-P24 | Port 2 pin 0,1,2,3,4 | In/Output |
| 33 | N/C | Not Connected (20 MHz) | Input |
| | /WAIT | WAIT (25 or 33 MHz) | Input |
| 34-36 | P25-P27 | Port 2 pin 5,6,7 | In/Output |
| 37 | P31 | Port 3 pin 1 | Input |
| 38 | P36 | Port 3 pin 6 | Output |
| 39 | $V_{CC}$ | Power Supply | Input |
| 40 | XTAL2 | Crystal, Osc. Clock | Output |
| 41 | XTAL1 | Crystal, Osc. Clock | Input |
| 42 | P37 | Port 3 pin 7 | Output |
| 43 | P30 | Port 3 pin 0 | Input |
| 44 | SCLK | System Clock | Output |

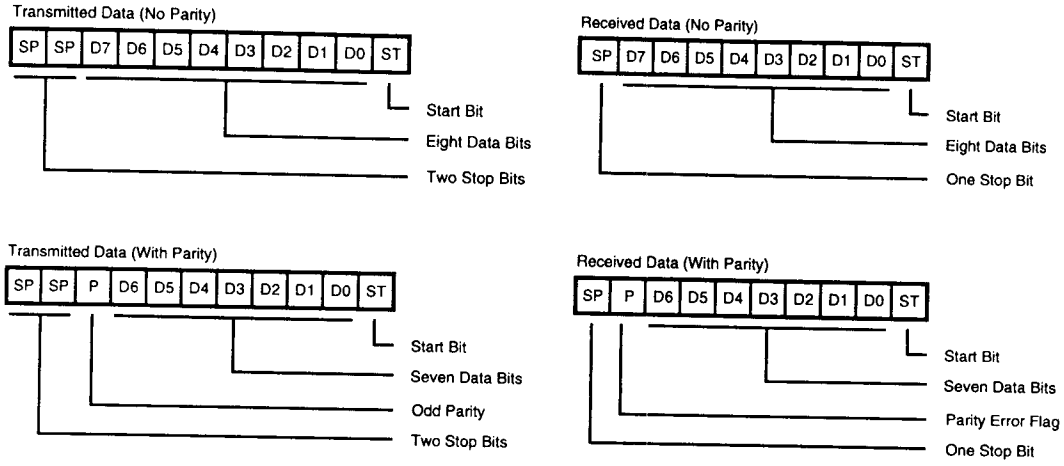



**Figure 11. Serial Data Formats**

## ADDRESS SPACE

Program Memory. The Z86C93 can address up to 64 Kbytes of external program memory. Program execution begins at external location 000CH after a reset.

Data Memory. The Z96C93 can address up to 64 Kbytes of external data memory. External data memory is included with, or separated from, the external program memory space. /DM, an optional I/O function that can be programmed to appear on pin P34 is used to distinguish between data and program memory space (Figure 12). The state of the /DM signal is controlled by the type instruction being executed. An LDC opcode references PROGRAM (/DM inactive) memory, and an LDE instruction references DATA (/DM active Low) memory.

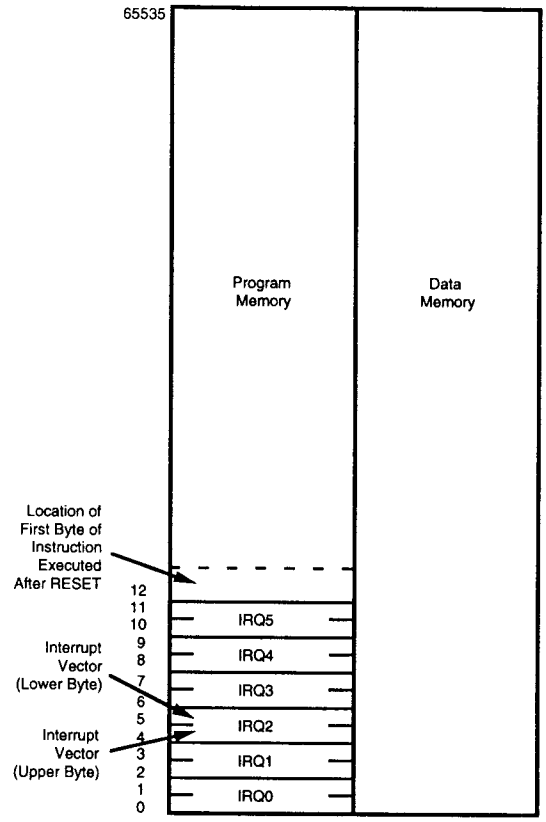



**Figure 12. Program and Data Memory Configuration**

**Expanded Register File.** The register file has been expanded to allow for additional system control registers, and for mapping of additional peripheral devices along with I/O ports into the register address area (Figure 13). The Z8 register address space R0 through R15 has now been implemented as 16 groups of 16 registers per group. These register groups are known as the ERF (Expanded Register File). Bits 7-4 of register RP select the working register group. Bits 3-0 of register RP select the expanded register group (Figure 14). The registers that are used in the multiply/divide unit reside in the Expanded Register File at Bank E and those for the additional timer control words reside in Bank D. The rest of the Expanded Register is not physically implemented and is open for future expansion.

**Register File.** The Register File consists of four I/O port registers, 236 general-purpose registers and 16 control and status registers. The instructions can access registers directly or indirectly via an 8-bit address field. The Z86C93 also allows short 4-bit register addressing using the Register Pointer (Figure 15). In the 4-bit mode, the Register File is divided into 16 working register groups, each occupying 16 continuous locations. The Register Pointer addresses the starting location of the active working-register group.

**Note:** Register Group E0-EF can only be accessed through working registers and indirect addressing modes.

**Stack.** The Z86C93 has a 16-bit Stack Pointer (R254-R255), used for external stack, that resides anywhere in the data memory. An 8-bit Stack Pointer (R255) is used for the internal stack that resides within the 236 general-purpose registers (R4-R239). The high byte of the Stack Pointer (SPH, Bits 8-15) can be used as a general-purpose register when using internal stack only.

Register allocation. The following is the register allocation during multiplication.

| | |
|---|---|
| Multiplier high byte | MREG2 |
| Multiplier low byte | MREG3 |
| Multiplicand high byte | MREG4 |
| Multiplicand low byte | MREG5 |
| Result high byte of high word | MREG0 |
| Result low byte of high word | MREG1 |
| Result high byte of low word | MREG2 |
| Result low byte of low word | MREG3 |
| Multiply/Divide Control register | MDCON |

The following is the register allocation during division.

| | |
|---|---|
| High byte of high word of dividend | MREG0 |
| Low byte of high word of dividend | MREG1 |
| High byte of low word of dividend | MREG2 |
| Low byte of low word of dividend | MREG3 |
| High byte of divisor | MREG4 |
| Low byte of divisor | MREG5 |
| High byte of remainder | MREG0 |
| Low byte of remainder | MREG1 |
| High byte of quotient | MREG2 |
| Low byte of quotient | MREG3 |
| Multiply/Divide Control register | MDCON |

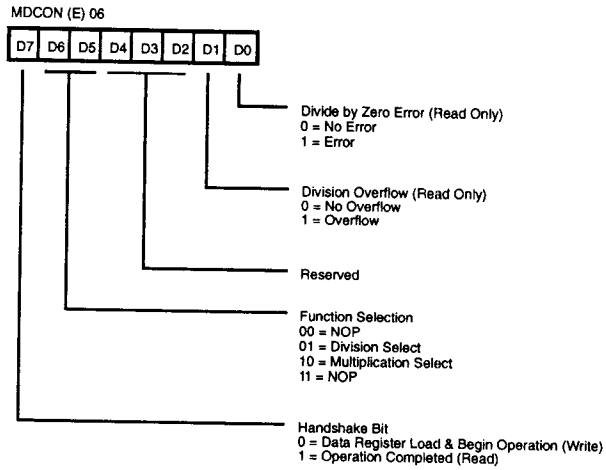



**Figure 17. Multiply/Divide Control Register (MDCON)**

Control register. The MDCON (Multiply/Divide Control Register) is used to interface with the multiply/divide unit (Figure 16). Specific functions of various bits in the control register are given below.

DONE bit (D7). This bit is a handshake bit between the math unit and the external world. On power up, this bit is set to 1 to indicate that the math unit has completed the previous operation and is ready to perform the next operation.

Before starting a new multiply/divide operation, this bit should be reset to 0 by the processor/programmer. This indicates that all the data registers have been loaded and the math unit can now begin a multiply/divide operation. During the process of multiplication or division, this bit is write-protected. Once the math unit completes its operation it sets this bit to indicate the completion of operation. The processor/programmer can then read the result.

MULSL. *Multiply Select* (D6). If this bit is set to 1, it indicates a multiply operation directive. Like the DONE bit, this bit is also write-protected during math unit operation and is reset to 0 by the math unit upon starting of the multiply/divide operation.

DIVSL. *Division Select* (D5). Similar to D6, D5 starts a division operation.

D4-D2. Reserved.

DIVOVF. *Division Overflow* (D1). This bit indicates an overflow during the division process. Division overflow occurs when the high word of the dividend is greater than or equal to the divisor. This bit is read only. When set to 1, it indicates overflow error.

The counters are programmed to start, stop, restart to continue, or restart from the initial value. The counters can also be programmed to stop upon reaching zero (single pass mode) or to automatically reload the initial value and continue counting (modulo-n continuous mode).

The counters, but not the prescalers, are read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and is either the internal microprocessor clock divided by four, or an external signal input via Port 3. The Timer Mode register configures the external timer input (P31) as an external clock, a trigger input that is retriggerable or non-retriggerable, or as a gate input for the internal clock. The counter/timers are cascaded by connecting the T0 output to the input of T1. Either T0 or T1 can be outputted via P36.

The following are the enhancements made to the counter/timer block on the Z86C93 (Figure 18):

- T0 counter length is extended to 16 bits. For example, T0 now has a 6-bit prescaler and 16-bit down counter.
- T1 counter length is extended to 16 bits. For example, T1 now has a 6-bit prescaler and 16-bit down counter.
- A new counter/timer T2 is added. T2 has a 4-bit prescaler and a 16-bit down counter with capture register.

These three counters are cascadable as shown in Table 6. The result is that T2 may be extendable to 32 bits and T1 extendable to 24 bits. Bits 1 and 0 (CAS1 AND CAS0) of the T2 Prescaler Register (PRE2) determine the counter length.

**Table 6. Counter Length Configurations**

| CAS 1 | CAS0 | T0 | T1 | T2 |
|---|---|---|---|---|
| 0 | 0 | 8 | 8 | 32 |
| 0 | 1 | 16 | 16 | 16 |
| 1 | 0 | 8 | 24 | 16 |
| 1 | 1 | 8 | 16 | 24 |

The controlling clock input to T2 is programmed to XTAL/2 or XTAL/8 (only when T2 counter length is 16 bits), which results in a resolution of 100 ns at an external XTAL clock speed of 20 MHz.

Capture Register. T2 has a 16-bit capture register associated with T2 HIGH BYTE and T2 LOW BYTE registers. The negative going transition on pin P33 enables the latching of the current T2 value (16 bits) into the capture register. The register mapping of the capture register is in Bank D (Figure 13). Note that the negative transition on P33 is capable of generating an interrupt. Also, the negative transition on P33 always latches the current T2 value into the capture register. There is no need for a control bit to enable/disable the latching; the capture register is read only.

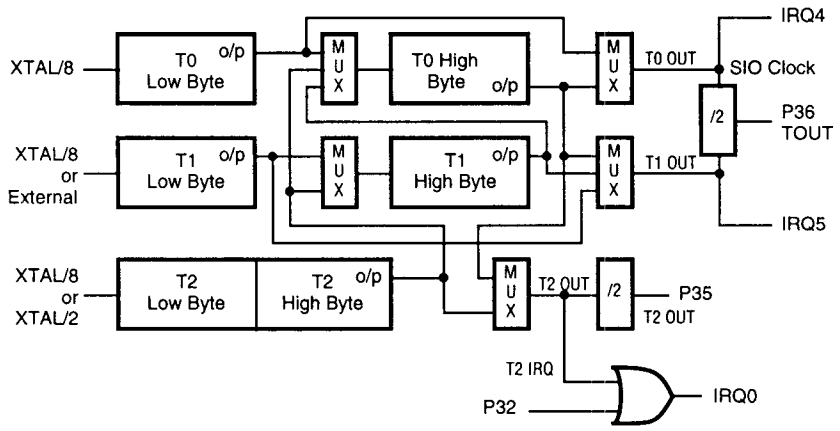



**Figure 18. Counter/Timer Block Diagram**

## Operation

Except for the programmable down counter length and clock input, T2 is identical to T0.

T0 and T1 retain all their features except that now they are extendable interims of the down-counter length.

The output of T2, under program control, goes to an output pin (P35). Also, the interrupt generated by T2 is ORed with the interrupt request generated by P32. Note that the service routine then has to poll the T2 flag bit and also clear it (Bit 7 of T2 Timer Mode Register).

On power up, T0 and T1 are configured in the 8-bit down counter length mode (to be compatible with Z86C91) and T2 is in the 32-bit mode with its output disabled (no interrupt is generated and T2 output DOES NOT go to port pin P35).

The UART uses T0 for generating the bit clock. This means, while using UART, T0 should be in 8-bit mode. So, while using the UART there are only two independent timer/counters.

The counters are configured in the following manner:

| Timer | Mode | Byte |
|---|---|---|
| T0 | 8-bit | Low Byte (T0) |
| T0 | 16-bit | High Byte (T0) + Low Byte (T0) |
| T1 | 8-bit | Low Byte (T1) |
| T1 | 16-bit | High Byte (T1)+ Low Byte (T1) |
| T1 | 24-bit | High Byte (T0) + High Byte (T1) + Low Byte (T1) |
| T2 | 16-bit | High Byte (T2) + Low Byte (T2) |
| T2 | 24-bit | High Byte (T0) + High Byte (T2) + Low Byte (T2) |
| T2 | 32-bit | High Byte (T0) + High Byte (T1) + High Byte (T2) + Low Byte (T2) |

Note that the T2 interrupt is logically ORed with P32 to generate IRQ0.

The T2 Timer Mode register is shown in Figure 19. Upon reaching end of count, bit 7 of this register is set to one. This bit IS NOT reset in hardware and it has to be cleared by the interrupt service routine.

T2 interrogates the state of the Count Mode Bit (D2) once it has counted down to it's zero value. T2 then makes the decision to continue counting (Module N Mode) or stop (Single Pass Mode). Observe this functionality if attempting to modify the count mode prior to the end of count bit (D7) being set.

The register map of the new CTC registers is shown in Figure 13. T0 high byte and T1 high byte are at the same relative locations as their respective low bytes, but in a different register bank.

The T2 prescaler register is shown in Figure 19. Bits 1 and 0 of this register control the various cascade modes of the counters.

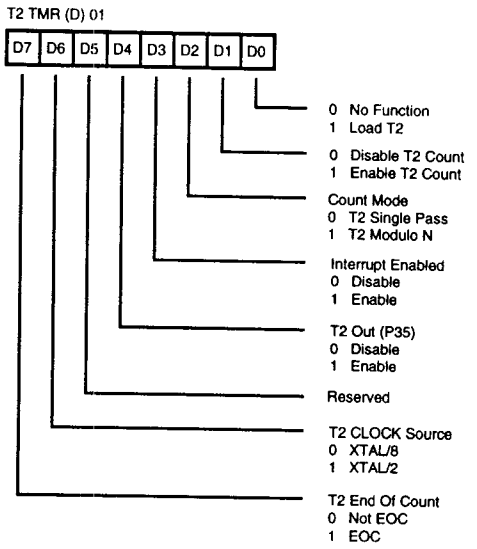



**Figure 19. T2 Timer Mode Register (T2)**

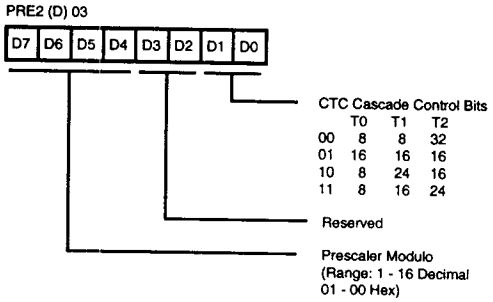



**Figure 20. T2 Prescaler Register (PRE2)**

## DC ELECTRICAL CHARACTERISTICS

$V_{CC}$ = 3.3V ± 10%

| Sym | Parameter | $T_A$ = 0°C to +70°C Min | Max | Typical at 25°C | Units | Conditions |
|---|---|---|---|---|---|---|
| | Max Input Voltage | | 7 | | V | $I_{IN}$ 250 µA |
| $V_{CH}$ | Clock Input High Voltage | 0.8 $V_{CC}$ | $V_{CC}$ | | V | Driven by External Clock Generator |
| $V_{CL}$ | Clock Input Low Voltage | -0.03 | 0.1x$V_{CC}$ | | V | Driven by External Clock Generator |
| $V_{IH}$ | Input High Voltage | 0.7x$V_{CC}$ | $V_{CC}$ | | V | |
| $V_{IL}$ | Input Low Voltage | -0.3 | 0.1x$V_{CC}$ | | V | |
| $V_{OH}$ | Output High Voltge | 1.8 | | | V | $I_{OH}$ = -1.0 mA |
| $V_{OH}$ | Output High Voltge | $V_{CC}$ - 100mV | | | V | $I_{OH}$ = -100 µA |
| $V_{OL}$ | Output Low Voltage | | 0.4 | | V | $I_{OL}$ = +1.0 mA |
| $V_{RH}$ | Reset Input High Voltage | 0.8x$V_{CC}$ | $V_{CC}$ | | V | |
| $V_{RI}$ | Reset Input Low Voltage | -0.03 | 0.1x$V_{CC}$ | | V | |
| $I_{IL}$ | Input Leakage | -2 | 2 | | µA | Test at 0V, $V_{CC}$ |
| $I_{OL}$ | Output Leakage | -2 | 2 | | µA | Test at 0V, $V_{CC}$ |
| $I_{IR}$ | Reset Input Current | | -80 | | µA | $V_{RL}$ = 0V |
| $I_{CC}$ | Supply Current | | 30 | 20 | mA | @ 25 MHz [1] |
| $I_{CC1}$ | Stand By Current (HALT Mode) | | 12 | 8 | mA | HALT Mode $V_{IN}$=0V, $V_{CC}$ @ 25 MHz [1] |
| $I_{CC2}$ | Stand By Current (HALT Mode) | | 8 | 1 | µA | STOP Mode $V_{IN}$=0V, $V_{CC}$ [1] |
| $I_{ALL}$ | Auto Latch Low Current | -10 | 10 | 5 | µA | |

**Note:**

[1] All inputs driven to 0V, $V_{CC}$ and outputs floating.

## AC CHARACTERISTICS

External I/O or Memory Read/Write Timing Diagram

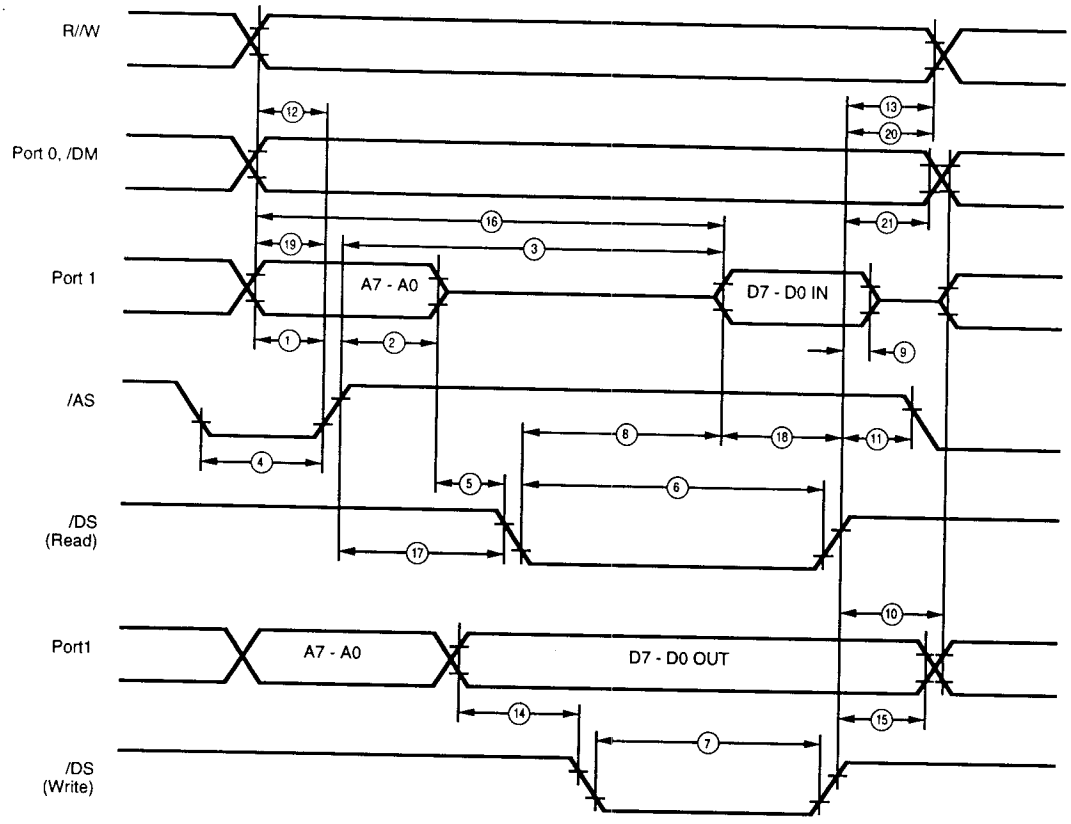



**Figure 24. External I/O or Memory Read/Write Timing**

## AC CHARACTERISTICS

External I/O or Memory Read and Write; DSR/DSW; WAIT Timing Table

| | | | $T_A$ = 0°C to +70°C | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 33 MHz | | 25 MHz | | 20 MHz | | Typical | |
| No | Sym | Parameter | Min | Max | Min | Max | Min | Max | $V_{cc}$=5.0V @ 25°C | Units |
| 1 | TdA(AS) | Address Valid To /AS Rise Delay | 13 | | 22 | | 26 | | | ns |
| 2 | TdAS(A) | /AS Rise To Address Hold Time | 20 | | 25 | | 28 | | | ns |
| 3 | TdAS(DI) | /AS Rise Data In Req'd Valid Delay | | 90 | | 130 | | 160 | | ns |
| 4 | TwAS | /AS Low Width | 20 | | 28 | | 36 | | | ns |
| 5 | TdAZ(DSR) | Address Float To /DS (Read) | 0 | | 0 | | 0 | | | ns |
| 6 | TwDSR | /DS (Read) Low Width | 65 | | 100 | | 130 | | | ns |
| 7 | TwDSW | /DS (Write) Low Width | 40 | | 65 | | 75 | | | ns |
| 8 | TdDSR(DI) | /DS (Read) To Data in Req'd Valid Delay | | 30 | | 78 | | 100 | | ns |
| 9 | ThDSR(DI) | /DS Rise (Read) to Data In Hold Time | 0 | | 0 | | 0 | | | ns |
| 10 | TdDS(A) | /DS Rise To Address Active Delay | 25 | | 34 | | 40 | | | ns |
| 11 | TdDS(AS) | /DS Rise To /AS Delay | 16 | | 30 | | 36 | | | ns |
| 12 | TdR/W(AS) | R/W To /AS Rise Delay | 12 | | 26 | | 32 | | | ns |
| 13 | TdDS(R/W) | /DS Rise To R/W Valid Delay | 12 | | 30 | | 36 | | | ns |
| 14 | TdDO(DSW) | Data Out To /DS (Write) Delay | 12 | | 34 | | 40 | | | ns |
| 15 | ThDSW(DO) | /DS Rise (Write) To Data Out Hold Time | 12 | | 34 | | 40 | | | ns |
| 16 | TdA(DI) | Address To Data In Req'd Valid Delay | | 110 | | 160 | | 200 | | ns |
| 17 | TdAS(DSR) | /AS Rise To /DS (Read) Delay | 20 | | 40 | | 48 | | | ns |
| 18 | TaDI(DSR) | Data In Set-up Time To /DS Rise Read | 16 | | 30 | | 36 | | | ns |
| 19 | TdDM(AS) | /DM To /AS Rise Delay | 10 | | 22 | | 26 | | | ns |
| 20 | TdDS(DM) | /DS Rise To /DM Valid Delay | | | | | | | 34* | ns |
| 21 | ThDS(A) | /DS Rise To Address Valid Hold Time | | | | | | | 34* | ns |
| 22 | TdXT(SCR) | XTAL Falling to SCLK Rising | | | | | | | 20* | ns |
| 23 | TdXT(SCF) | XTAL Falling to SCLK Falling | | | | | | | 23* | ns |
| 24 | TdXT(DSRF) | XTAL Falling to/DS Read Falling | | | | | | | 29* | ns |
| 25 | TdXT(DSRR) | XTAL Falling to /DS Read Rising | | | | | | | 29* | ns |
| 26 | TdXT(DSWF) | XTAL Falling to /DS Write Falling | | | | | | | 29* | ns |
| 27 | TdXT(DSWF) | XTAL Falling to /DS Write Rising | | | | | | | 29* | ns |
| 28 | TsW(XT) | Wait Set-up Time | | | | | | | 10* | ns |
| 29 | ThW(XT) | Wait Hold Time | | | | | | | 15* | ns |
| 30 | TwW | Wait Width (One Wait Time) | | | | | | | 25* | ns |

**Notes:**
When using extended memory timing add 2 TpC.
Timing numbers given are for minimum TpC.
* Preliminary value to be characterized.

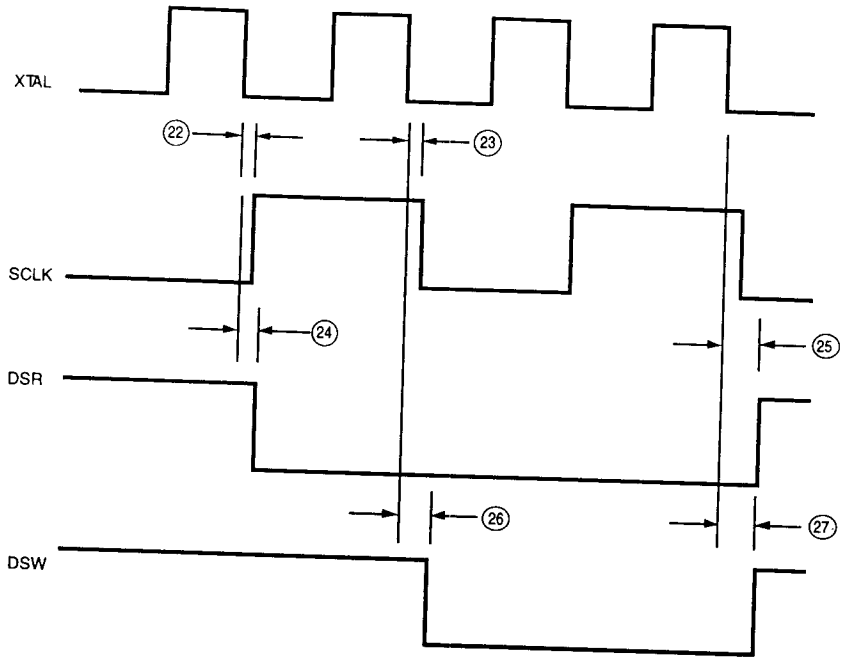



**Figure 25. XTAL/SCLK To DSR and DSW Timing**

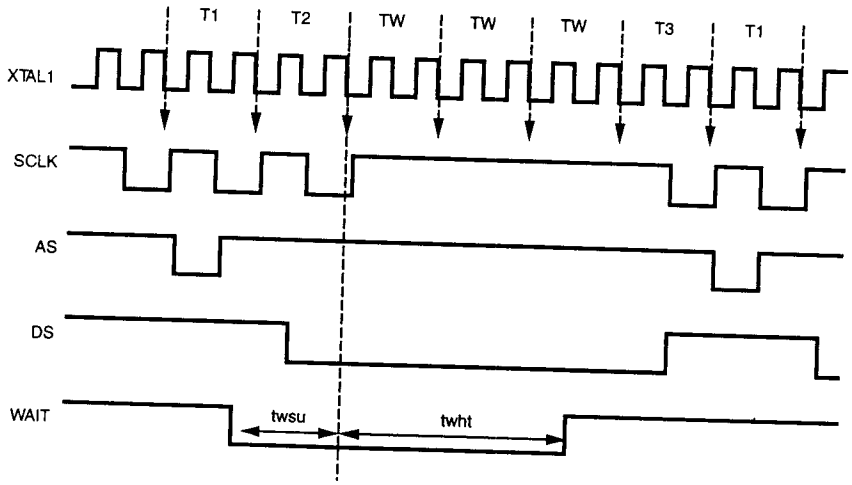



**Figure 26. XTAL/SCLK To WAIT Timing (25 MHz Device Only)**

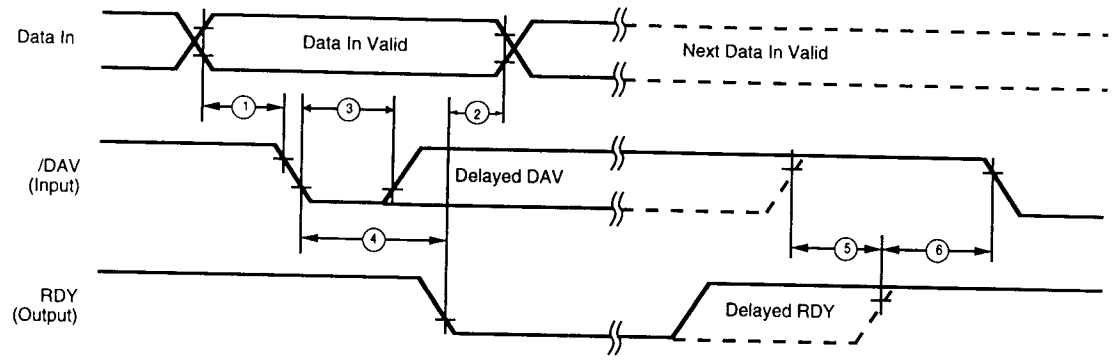



**Figure 28. Input Handshake Timing**

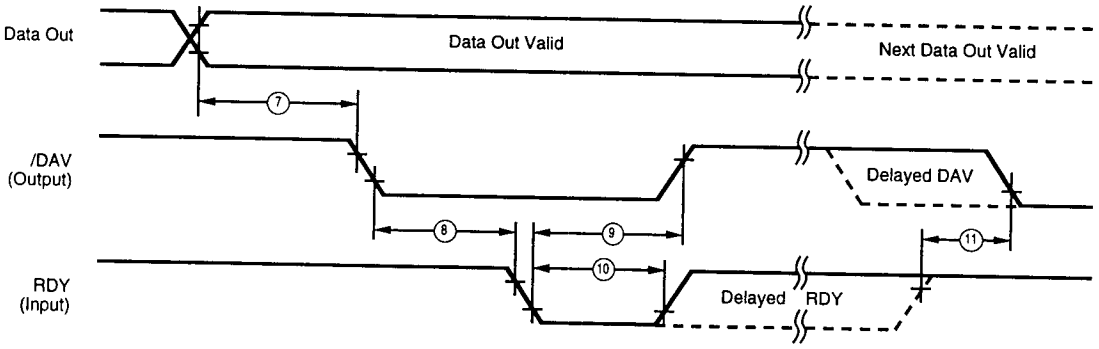



**Figure 29. Output Handshake Timing**

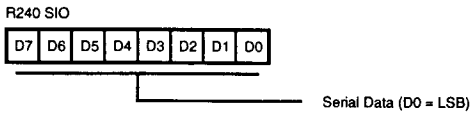



**Figure 37. Serial I/O Register (F0H: Read/Write)**

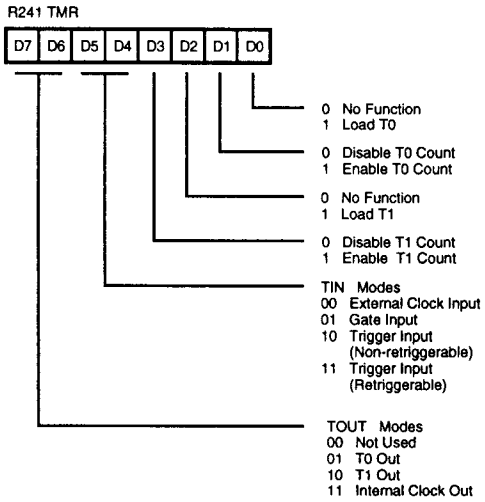



**Figure 38. Timer Mode Register (F1H: Read/Write)**

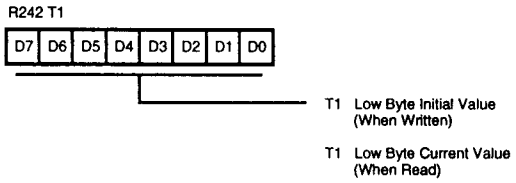



**Figure 39. Counter/Timer 1 Register (F2H: Read/Write)**

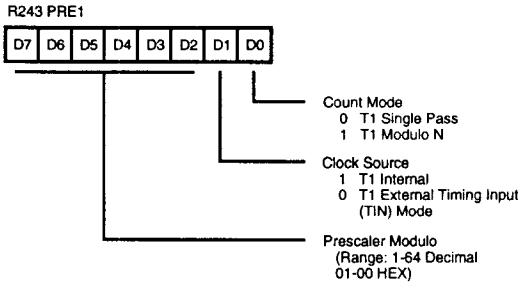



**Figure 40. Prescaler 1 Register (F3H: Write Only)**

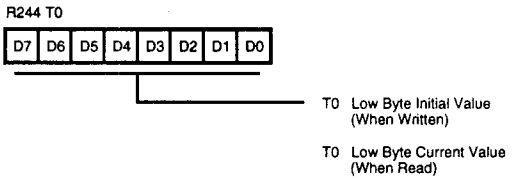



**Figure 41. Counter/Timer 0 Register (F4H: Read/Write)**

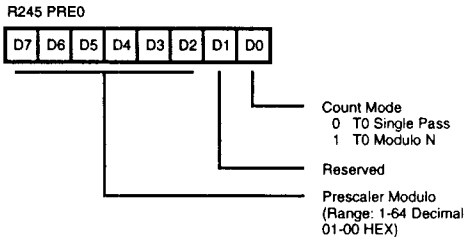



**Figure 42. Prescaler 0 Register (F5H: Write Only)**

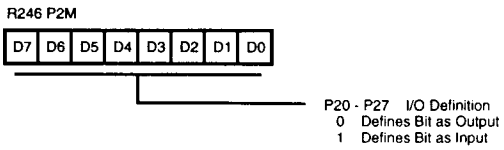



**Figure 43. Port 2 Mode Register (F6H: Write Only)**

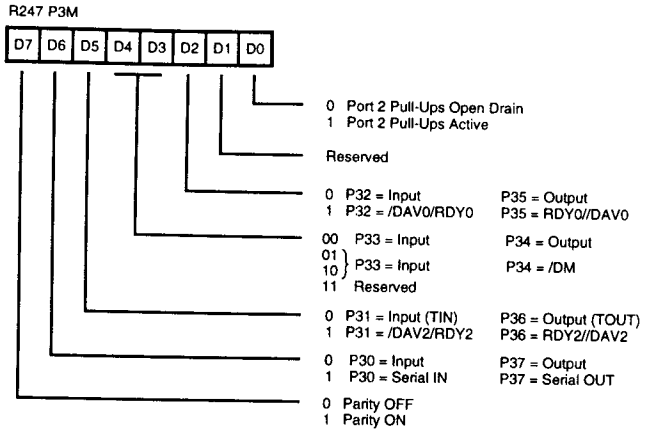



**Figure 44. Port 3 Mode Register (F7H: Write Only)**

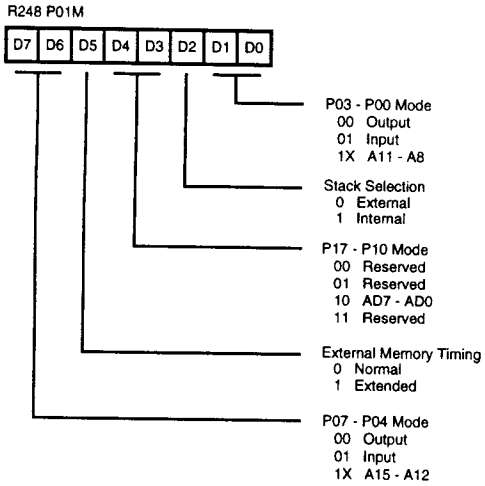



**Figure 45. Ports 0 and 1 Mode Registers (F8H: Write Only)**

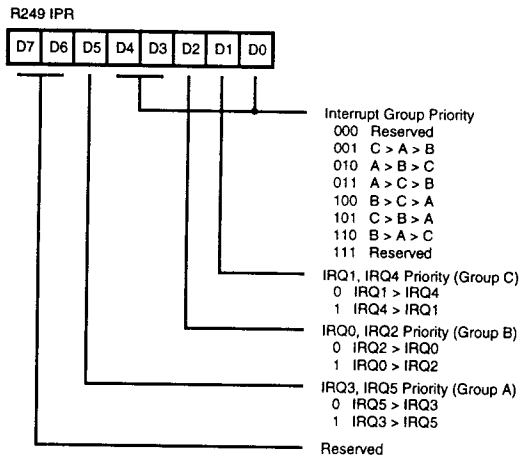



**Figure 46. Interrupt Priority Register (F9H: Write Only)**

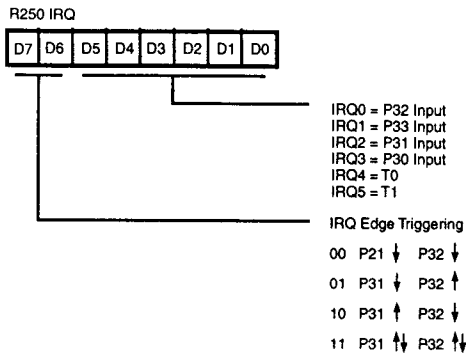



**Figure 47. Interrupt Request Register (FAH: Read/Write)**

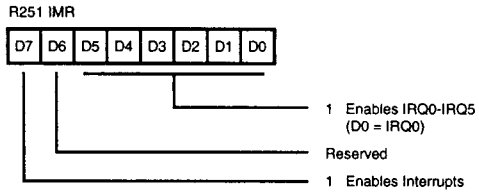



**Figure 48. Interrupt Mask Register (FBH: Read/Write)**

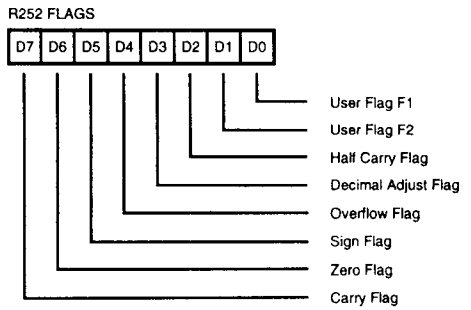



**Figure 49. Flag Register (FCH: Read/Write)**

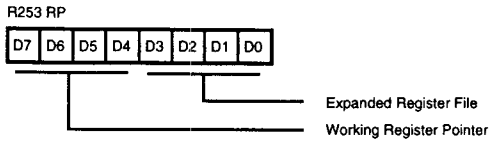



**Figure 50. Register Pointer (FDH: Read/Write)**

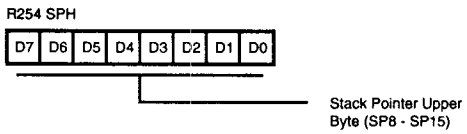



**Figure 51. Stack Pointer High (FEH: Read/Write)**

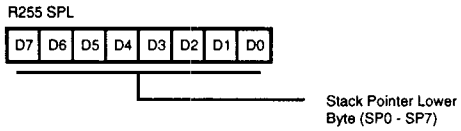
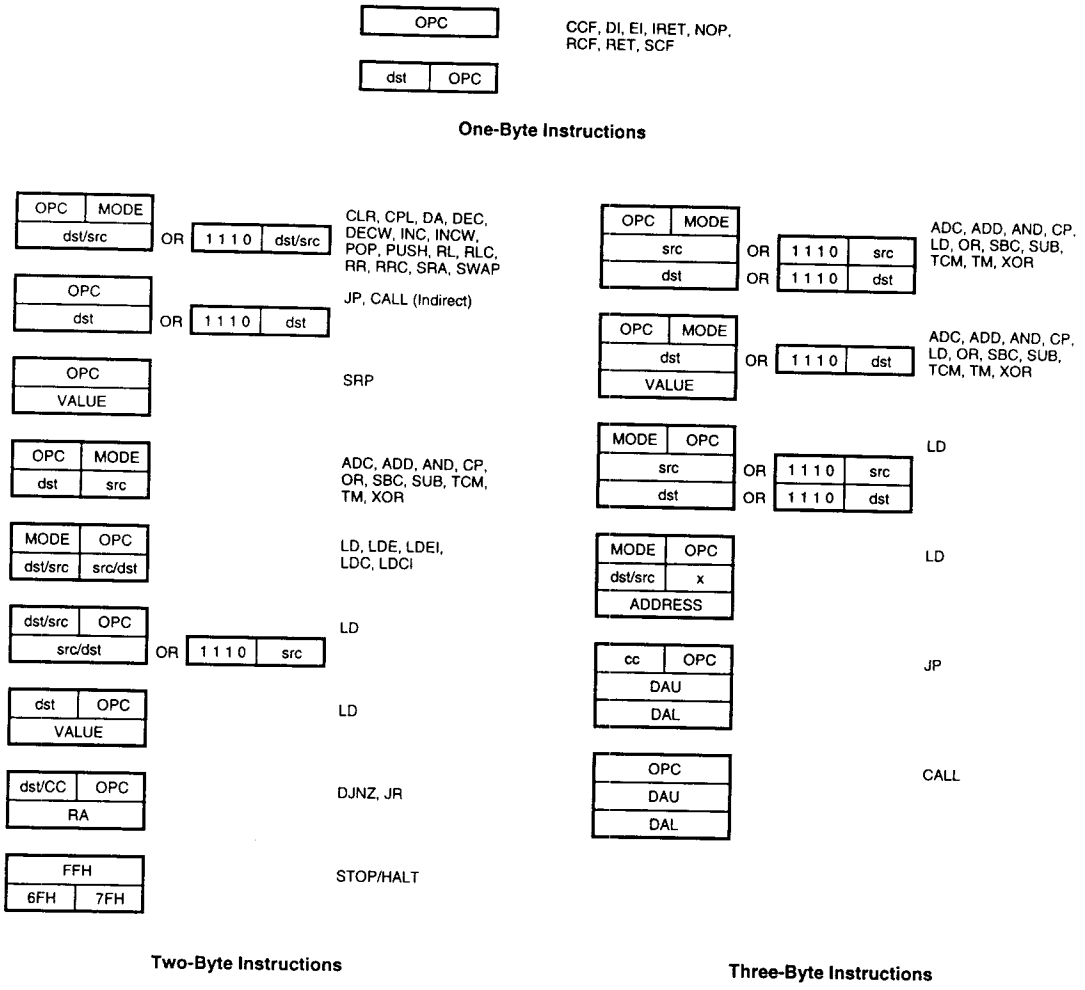



**Figure 52. Stack Pointer Low (FFH: Read/Write)**

## INSTRUCTION FORMATS




## INSTRUCTION SUMMARY

Note: Assignment of a value is indicated by the symbol " ← ". For example:

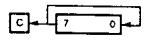
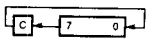
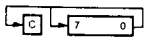
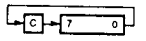
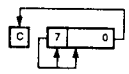
dst ← dst + src

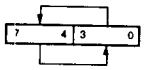
indicates that the source data is added to the destination data and the result is stored in the destination location. The notation "addr (n)" is used to refer to bit (n) of a given operand location. For example:

dst (7)

refers to bit 7 of the destination operand.

## INSTRUCTION SUMMARY (Continued)

| Instruction and Operation | Address Mode dst | Address Mode src | Opcode Byte (Hex) | Flags Affected C | Z | S | V | D | H |
|---|---|---|---|---|---|---|---|---|---|
| **NOP** | | | FF | - | - | - | - | - | - |
| **OR** dst, src<br>dst←dst OR src | † | | 4[ ] | - | * | * | 0 | - | - |
| **POP** dst<br>dst←@SP;<br>SP←SP + 1 | R<br>IR | | 50<br>51 | - | - | - | - | - | - |
| **PUSH** src<br>SP←SP - 1;<br>@SP←src | | R<br>IR | 70<br>71 | - | - | - | - | - | - |
| **RCF**<br>C←0 | | | CF | 0 | - | - | - | - | - |
| **RET**<br>PC←@SP;<br>SP←SP + 2 | | | AF | - | - | - | - | - | - |
| **RL** dst<br>  | R<br>IR | | 90<br>91 | * | * | * | * | - | - |
| **RLC** dst<br>  | R<br>IR | | 10<br>11 | * | * | * | * | - | - |
| **RR** dst<br>  | R<br>IR | | E0<br>E1 | * | * | * | * | - | - |
| **RRC** dst<br>  | R<br>IR | | C0<br>C1 | * | * | * | * | - | - |
| **SBC** dst, src<br>dst←dst←src←C | † | | 3[ ] | * | * | * | * | 1 | * |
| **SCF**<br>C←1 | | | DF | 1 | - | - | - | - | - |
| **SRA** dst<br>  | R<br>IR | | D0<br>D1 | * | * | * | 0 | - | - |
| **SRP** src<br>RP←src | | Im | 31 | - | - | - | - | - | - |

| Instruction and Operation | Address Mode dst | Address Mode src | Opcode Byte (Hex) | Flags Affected C | Z | S | V | D | H |
|---|---|---|---|---|---|---|---|---|---|
| **STOP** | | | 6F | - | - | - | - | - | - |
| **SUB** dst, src<br>dst←dst←src | † | | 2[ ] | * | * | * | * | 1 | * |
| **SWAP** dst<br>  | R<br>IR | | F0<br>F1 | X | * | * | X | - | - |
| **TCM** dst, src<br>(NOT dst)<br>AND src | † | | 6[ ] | - | * | * | 0 | - | - |
| **TM** dst, src<br>dst AND src | † | | 7[ ] | - | * | * | 0 | - | - |
| **XOR** dst, src<br>dst←dst<br>XOR src | † | | B[ ] | - | * | * | 0 | - | - |

† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a '[ ]' in this table, and its value is found in the following table to the left of the applicable addressing mode pair.

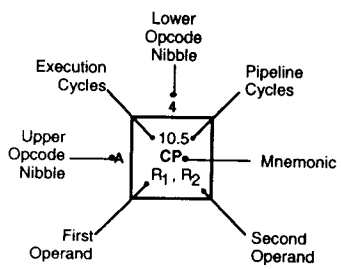
For example, the opcode of an ADC instruction using the addressing modes r (destination) and Ir (source) is 13.

| Address Mode dst | Address Mode src | Lower Opcode Nibble |
|---|---|---|
| r | r | [2] |
| r | Ir | [3] |
| R | R | [4] |
| R | IR | [5] |
| R | IM | [6] |
| IR | IM | [7] |

## OPCODE MAP

Lower Nibble (Hex)

| Upper Nibble (Hex) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 6.5 **DEC** R1 | 6.5 **DEC** IR1 | 6.5 **ADD** r1, r2 | 6.5 **ADD** r1, Ir2 | 10.5 **ADD** R2, R1 | 10.5 **ADD** IR2, R1 | 10.5 **ADD** R1, IM | 10.5 **ADD** IR1, IM | 6.5 **LD** r1, R2 | 6.5 **LD** r2, R1 | 12/10.5 **DJNZ** r1, RA | 12/10.0 **JR** cc, RA | 6.5 **LD** r1, IM | 12.10.0 **JP** cc, DA | 6.5 **INC** r1 | |
| 1 | 6.5 **RLC** R1 | 6.5 **RLC** IR1 | 6.5 **ADC** r1, r2 | 6.5 **ADC** r1, Ir2 | 10.5 **ADC** R2, R1 | 10.5 **ADC** IR2, R1 | 10.5 **ADC** R1, IM | 10.5 **ADC** IR1, IM | | | | | | | | |
| 2 | 6.5 **INC** R1 | 6.5 **INC** IR1 | 6.5 **SUB** r1, r2 | 6.5 **SUB** r1, Ir2 | 10.5 **SUB** R2, R1 | 10.5 **SUB** IR2, R1 | 10.5 **SUB** R1, IM | 10.5 **SUB** IR1, IM | | | | | | | | |
| 3 | 8.0 **JP** IRR1 | 6.1 **SRP** IM | 6.5 **SBC** r1, r2 | 6.5 **SBC** r1, Ir2 | 10.5 **SBC** R2, R1 | 10.5 **SBC** IR2, R1 | 10.5 **SBC** R1, IM | 10.5 **SBC** IR1, IM | | | | | | | | |
| 4 | 8.5 **DA** R1 | 8.5 **DA** IR1 | 6.5 **OR** r1, r2 | 6.5 **OR** r1, Ir2 | 10.5 **OR** R2, R1 | 10.5 **OR** IR2, R1 | 10.5 **OR** R1, IM | 10.5 **OR** IR1, IM | | | | | | | | |
| 5 | 10.5 **POP** R1 | 10.5 **POP** IR1 | 6.5 **AND** r1, r2 | 6.5 **AND** r1, Ir2 | 10.5 **AND** R2, R1 | 10.5 **AND** IR2, R1 | 10.5 **AND** R1, IM | 10.5 **AND** IR1, IM | | | | | | | | |
| 6 | 6.5 **COM** R1 | 6.5 **COM** IR1 | 6.5 **TCM** r1, r2 | 6.5 **TCM** r1, Ir2 | 10.5 **TCM** R2, R1 | 10.5 **TCM** IR2, R1 | 10.5 **TCM** R1, IM | 10.5 **TCM** IR1, IM | | | | | | | | 6.0 **STOP** |
| 7 | 10/12.1 **PUSH** R2 | 12/14.1 **PUSH** IR2 | 6.5 **TM** r1, r2 | 6.5 **TM** r1, Ir2 | 10.5 **TM** R2, R1 | 10.5 **TM** IR2, R1 | 10.5 **TM** R1, IM | 10.5 **TM** IR1, IM | | | | | | | | 7.0 **HALT** |
| 8 | 10.5 **DECW** RR1 | 10.5 **DECW** IR1 | 12.0 **LDE** r1, Irr2 | 18.0 **LDEI** Ir1, Irr2 | | | | | | | | | | | | 6.1 **DI** |
| 9 | 6.5 **RL** R1 | 6.5 **RL** IR1 | 12.0 **LDE** r2, Irr1 | 18.0 **LDEI** Ir2, Irr1 | | | | | | | | | | | | 6.1 **EI** |
| A | 10.5 **INCW** RR1 | 10.5 **INCW** IR1 | 6.5 **CP** r1, r2 | 6.5 **CP** r1, Ir2 | 10.5 **CP** R2, R1 | 10.5 **CP** IR2, R1 | 10.5 **CP** R1, IM | 10.5 **CP** IR1, IM | | | | | | | | 14.0 **RET** |
| B | 6.5 **CLR** R1 | 6.5 **CLR** IR1 | 6.5 **XOR** r1, r2 | 6.5 **XOR** r1, Ir2 | 10.5 **XOR** R2, R1 | 10.5 **XOR** IR2, R1 | 10.5 **XOR** R1, IM | 10.5 **XOR** IR1, IM | | | | | | | | 16.0 **IRET** |
| C | 6.5 **RRC** R1 | 6.5 **RRC** IR1 | 12.0 **LDC** r1, Irr2 | 18.0 **LDCI** Ir1, Irr2 | | | | 10.5 **LD** r1,x,R2 | | | | | | | | 6.5 **RCF** |
| D | 6.5 **SRA** R1 | 6.5 **SRA** IR1 | 12.0 **LDC** r2, Irr1 | 18.0 **LDCI** Ir2, Irr1 | 20.0 **CALL*** IRR1 | | 20.0 **CALL** DA | 10.5 **LD** r2,x,R1 | | | | | | | | 6.5 **SCF** |
| E | 6.5 **RR** R1 | 6.5 **RR** IR1 | | 6.5 **LD** r1, IR2 | 10.5 **LD** R2, R1 | 10.5 **LD** IR2, R1 | 10.5 **LD** R1, IM | 10.5 **LD** IR1, IM | | | | | | | | 6.5 **CCF** |
| F | 8.5 **SWAP** R1 | 8.5 **SWAP** IR1 | | 6.5 **LD** Ir1, r2 | | 10.5 **LD** R2, IR1 | | | | | | | | | | 6.0 **NOP** |
| Bytes per Instruction | 2 | | 3 | | 2 | | 3 | | 1 | | | | | | | |




**Legend:**
R = 8-bit address
r = 4-bit address
$R_1$ or $r_2$ = Dst address
$R_1$ or $r_2$ = Src address

**Sequence:**
Opcode, First Operand, Second Operand

**Note:** The blank areas are not defined.

* 2-byte instruction appears as a 3-byte instruction

Notes:

## ZILOG DOMESTIC SALES OFFICES AND TECHNICAL CENTERS

CALIFORNIA
Agoura ..... 818-707-2160
Campbell ..... 408-370-8120
Irvine ..... 714-453-9701

COLORADO
Boulder ..... 303-494-2905

FLORIDA
Largo ..... 813-585-2533

GEORGIA
Norcross ..... 404-448-9370

ILLINOIS
Schaumburg ..... 708-517-8080

MINNESOTA
Minneapolis ..... 612-944-0737

NEW HAMPSHIRE
Nashua ..... 603-888-8590

OHIO
Independence ..... 216-447-1480

OREGON
Portland ..... 503-274-6250

PENNSYLVANIA
Ambler ..... 215-653-0230

TEXAS
Dallas ..... 214-987-9987

WASHINGTON
Seattle ..... 206-523-3591

## INTERNATIONAL SALES OFFICES

CANADA
Toronto ..... 416-673-0634

GERMANY
Munich ..... 49-8967-2045
Sömmerda ..... 49-3634-23906

JAPAN
Tokyo ..... 81-3-3587-0528

HONG KONG
Kowloon ..... 852-7238979

KOREA
Seoul ..... 82-2-577-3272

SINGAPORE
Singapore ..... 65-2357155

TAIWAN
Taipei ..... 886-2-741-3125

UNITED KINGDOM
Maidenhead ..... 44-628-392-00



**Zilog, Inc. 210 East Hacienda Ave.**
**Campbell, CA 95008-6600**
**Telephone (408) 370-8000**
**Telex 910-338-7621**
**FAX 408 370-8056**