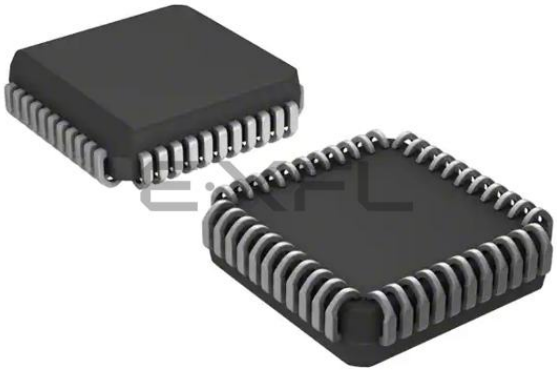


Welcome to E-XFL.COM

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	25MHz
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)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z86c9325vsc00tr

ZiLOG

PRODUCT SPECIFICATION

Z86C93

CMOS Z8® MULTIPLY/DIVIDE MICROCONTROLLER

FEATURES

- Complete microcontroller, up to 24 I/O lines, and up to 64 Kbytes of addressable external space each for program and data memory.
- 16-bit x 16-bit hardwired multiplier with 32-bit product in 17 clock cycles.
- 32-bit x 16-bit hardwired divider with 16-bit quotient and 16-bit remainder in 20 clock cycles.
- 256-byte register file, including 236 general-purpose registers, up to three I/O port registers and 16 status and control registers.
- 17-byte Expanded Register File, including two general-purpose registers and 15 status and control registers.
- Vectored, priority interrupts for I/O, counter/timers and UART.
- On-chip oscillator that accepts crystal or external clock drive.
- Two 16-bit counter timers with 6-bit prescalers.
- Third 16-bit counter/timer with 4-bit prescaler, one capture register and a fast decrement mode.
- Register Pointer for short, fast instructions that can access any one of the sixteen working register groups.
- Additional emulation signals SCLK, IACK, and /SYNC are made available.
- Two low power standby modes, STOP and HALT
- Full-duplex UART
- 3.3 ± 10% volt operation at 25 MHz
- 5.0 ± 10% volt operation at 20, 25 and 33 MHz

GENERAL DESCRIPTION

The Z86C93 is a CMOS ROMless Z8 microcontroller enhanced with a hardwired 16-bit x 16-bit multiplier and 32-bit/16-bit divider and three 16-bit counter timers (Figure 1). A capture register and a fast decrement mode is also provided. It is offered in 40-pin PDIP, 44-pin PLCC, 44-pin QFP and 48-pin VQFP (Figures 2, 3, 4, 5 and 6). Besides the four additional signals (SCLK, IACK, /SYNC and /WAIT), the Z86C93 is compatible with the Z86C91, yet it offers a much more powerful mathematical capability.

The Z86C93 provides up to 16 output address lines permitting an address space of up to 64 Kbytes of data and program memory each. Eight address outputs (AD7-AD0) are provided by a multiplexed, 8-bit, Address/Data bus. The remaining 8 bits can be provided by the software configuration of Port 0 to output address bits A15-A8.

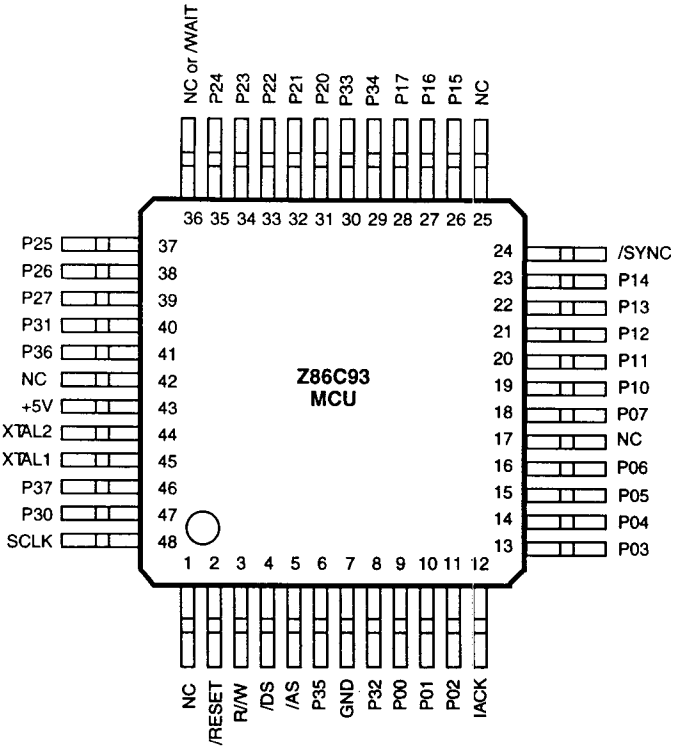


Figure 6. 48-Pin VQFP Package

Table 4. 48-Pin VQFP Pin Identification

No	Symbol	Function	Direction
1	N/C	Not Connected	Input
2	/RESET	Reset	Input
3	R/W	Read/Write	Output
4	/DS	Data Strobe	Output
5	/AS	Address Strobe	Output
6	P35	Port 3 pin 5	Input
7	GND	Ground GND	Input
8	P32	Port 3 pin 2	Input
9-11	P00-P02	Port 0 pin 3,4,5,6	In/Output
12	IACK	Int. Acknowledge	Output
13-16	P03-P06	Port 0 pin 3,4,5,6	In/Output
17	N/C	Not Connected	Input
18	P07	Port 0 pin 7	In/Output
19-23	P10-P14	Port 1 pin 0,1,2,3,4	In/Output
24	/SYNC	Synchronize Pin	Output

No	Symbol	Function	Direction
25	N/C	Not Connected	Input
26-28	P15-P17	Port 1 pin 5,6,7	In/Output
29	P34	Port 3 pin 4	Output
30	P33	Port 3 pin 33	Input
31-35	P20-P24	Port 2 pin 0,1,2,3,4	In/Output
36	N/C	Not Connected (20 MHz)	Input
	/WAIT	WAIT (25 or 33 MHz)	Input
37-39	P25-P27	Port 2 pin 5,6,7	In/Output
40	P31	Port 3 pin 1	Input
41	P36	Port 3 pin 6	Output
42	N/C	Not Connected	Input
43	V_{cc}	Power Supply	Input
44	XTAL2	Crystal, Osc. Clock	Output
45	XTAL1	Crystal, Osc. Clock	Input
46	P37	Port 3 pin 7	Output
47	P30	Port 3 pin 0	Input
48	SCLK	System Clock	Output

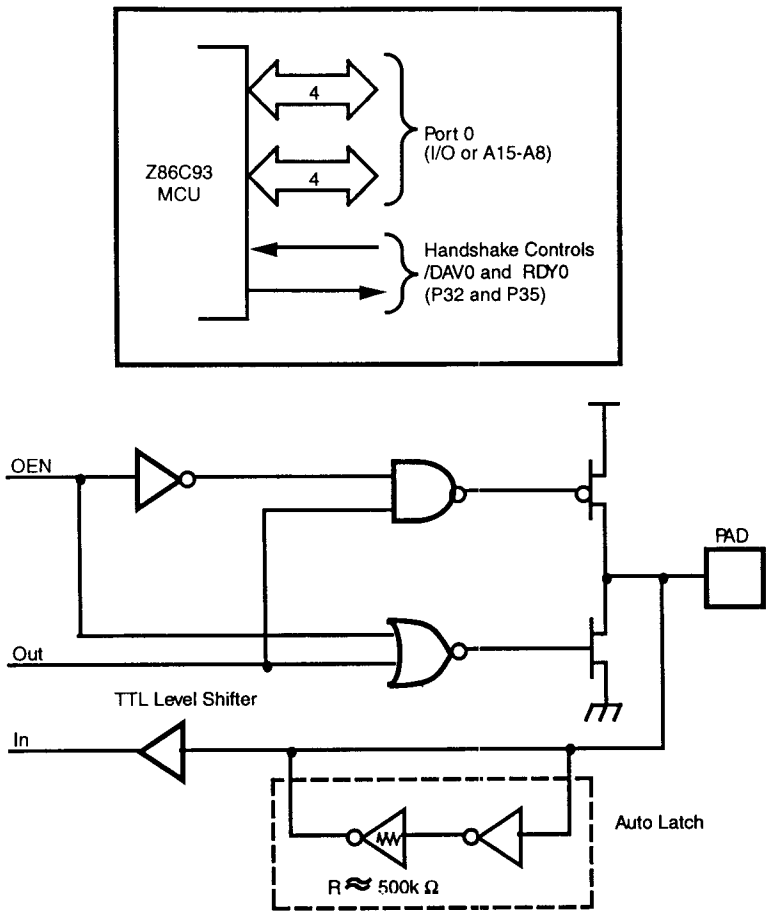


Figure 7. Port 0 Configuration

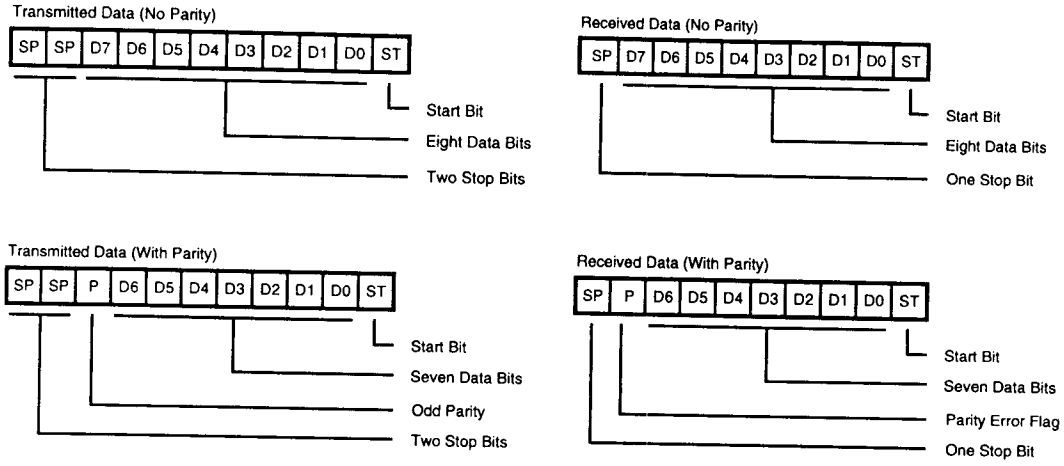


Figure 11. Serial Data Formats

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.

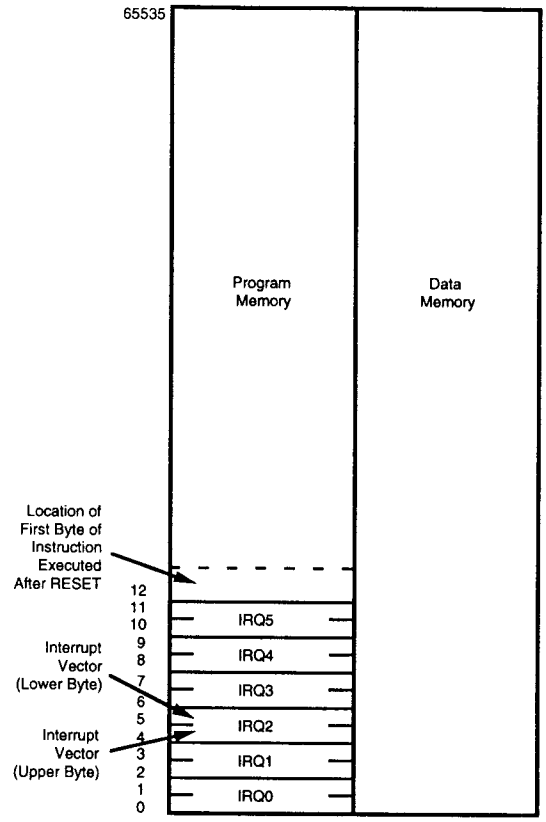


Figure 12. Program and Data Memory Configuration

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.

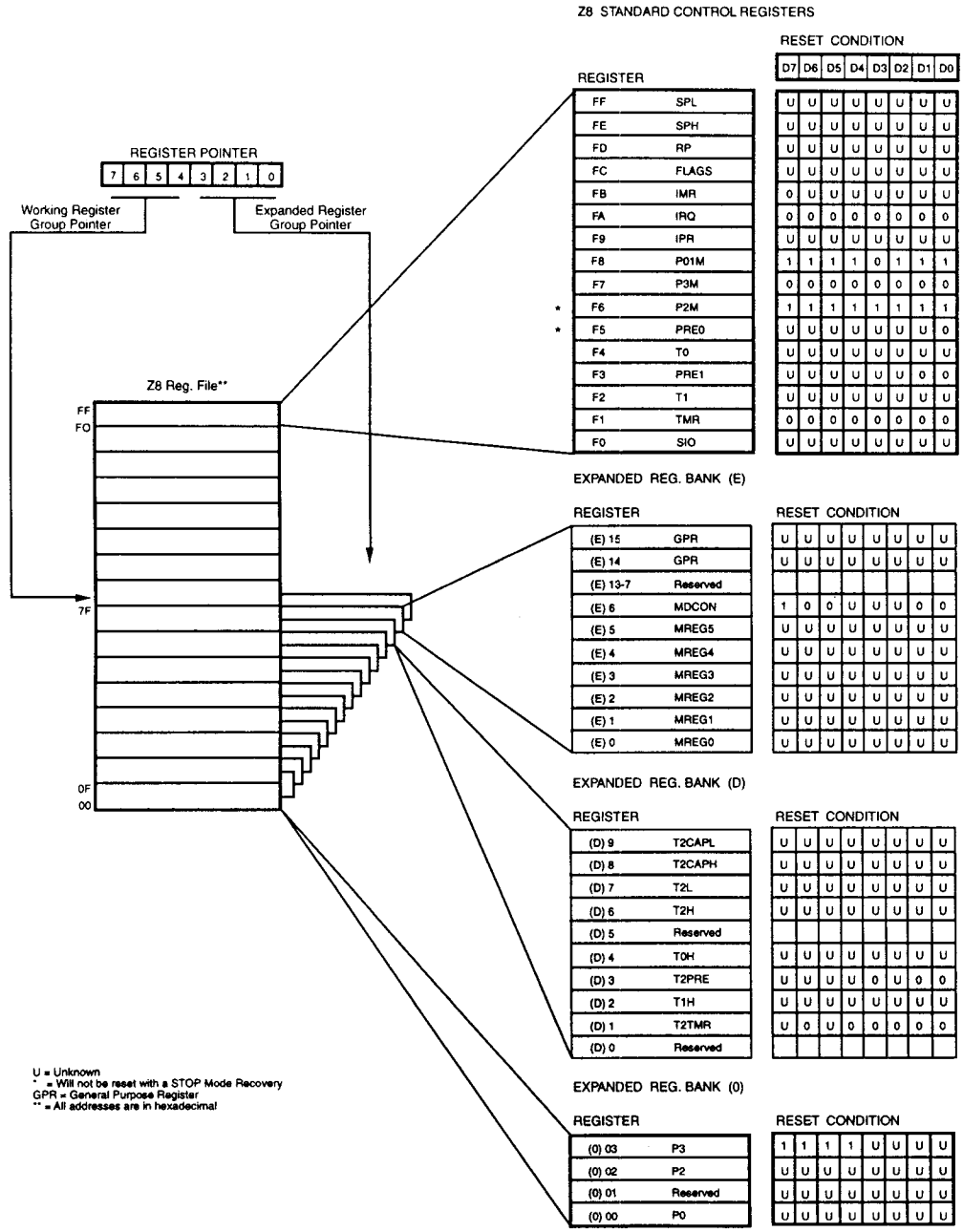


Figure 13. Register File

FUNCTIONAL DESCRIPTION

This section breaks down the Z86C93 into its main functional parts.

Multiply/Divide Unit

The Multiply/Divide unit describes the basic features, implementation details of the interface between the Z8 and the multiply/divide unit.

Basic features:

- 16-bit by 16-bit multiply with 32-bit product
- 32-bit by 16-bit divide with 16-bit quotient and 16-bit remainder
- Unsigned integer data format
- Simple interface to Z8

Interface to Z8. The following is a brief description of the register mapping in the multiply/divide unit and its interface to the Z8 (Figure 16).

The multiply/divide unit is interfaced like a peripheral. The only addressing mode available with the peripheral interface is register addressing. In other words, all of the operands are in the respective registers before a multiplication/division can start.

Register mapping. The registers used in the multiply/divide unit are mapped onto the expanded register file in Bank E. The exact register locations used are shown below.

REGISTER	ADDRESS
MREG0	(E) 00
MREG1	(E) 01
MREG2	(E) 02
MREG3	(E) 03
MREG4	(E) 04
MREG5	(E) 05
MDCON	(E) 06
GPR	(E) 14
GPR	(E) 15

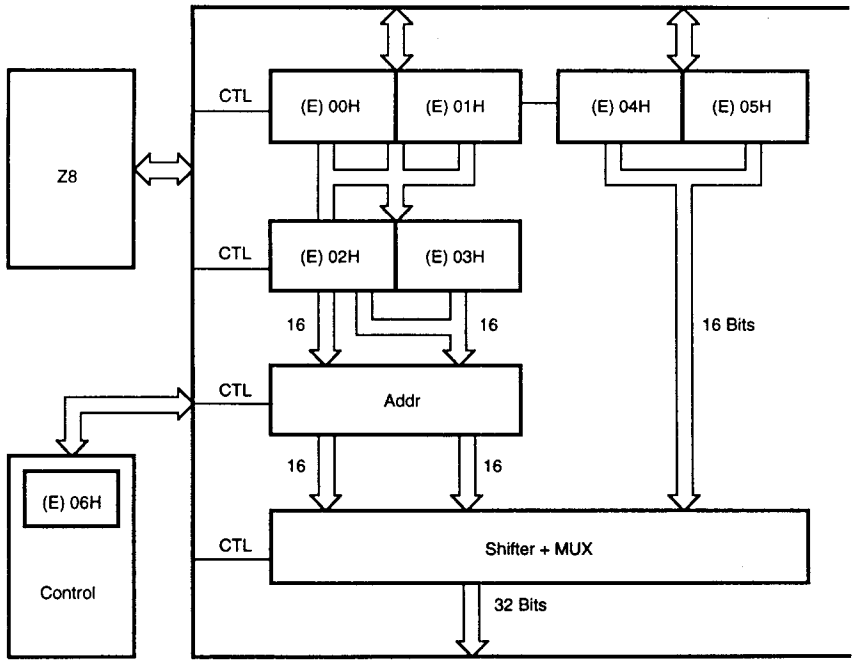


Figure 16. Multiply/Divide Unit Block Diagram

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

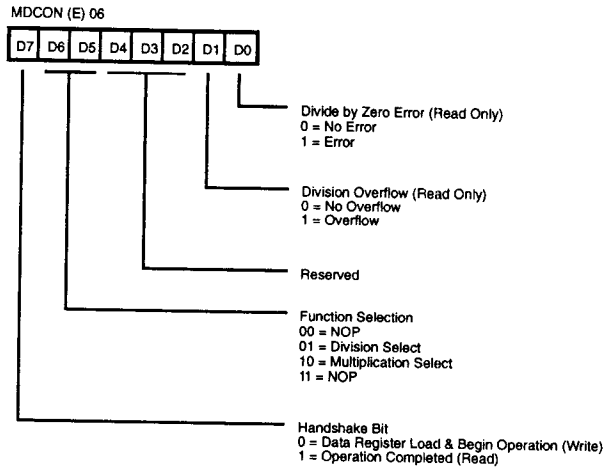


Figure 17. Multiply/Divide Control Register (MDCON)

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.

DIVZR. *Division by Zero* (D0). When set to 1, this indicates an error of division by 0. This bit is read only.

Example:

Upon reset, the status of the MDCON register is 100uuu00b (D7 to D0).

u = Undefined
x = Irrelevant
b = Binary

If multiplication operation is desired, the MDCON register is set to 010xxxxxb.

If the MDCON register is READ during multiplication, it would have a value of 000uuu00b.

Upon completion of multiplication, the result of the MDCON register is 100uuu00b.

If division operation is desired, the MDCON register is set to 001xxxxxb.

During division operation, the register would contain 000uu??b (? - value depends on the DIVIDEND, DIVISOR).

Upon completion of division operation, the MDCON register contains 100uuu??b.

Note that once the multiplication/division operation starts, all data registers (MREG5 through MREG0) are write-protected and so are the writable bits of the MDCON register. The write protection is released once the math unit operation is complete. However, the registers may be read at any time.

A multiplication sequence would look like:

1. Load multiplier and multiplicand.
2. Load MDCON register to start multiply operation.
3. Wait for the DONE bit of the MDCON register to be set to 1 and then read results.

Note that while the multiply/divide operation is in progress, the programmer can use the Z8 to do other things. Also, since the multiplication/division takes a fixed number of cycles, he can start reading the results before the DONE bit is set.

During a division operation, the error flag bits are set at the beginning of the division operation which means the flag bits can be checked by the Z8 while the division operation is being done.

The two general purpose registers can be used as scratch pad registers or as external data memory address pointers during an LDE instruction. MREG0 through MREG5, if not used for multiplication or division, can be used as general purpose registers.

Performance of multiplication. The actual multiplication takes 17 internal clock cycles. It is expected that the chip would run at a 10 MHz internal clock frequency (external clock divided by two). This results in an actual multiplication time (16-bit x 16-bit) of 1.7 µs. If the time to load operands and read results is included:

Number of internal clock cycles to load 5 registers: 30
Number of internal clock cycles to read 4 registers: 24

The total internal clock cycles to perform a multiplication is 71. This results in a net multiplication time of 7.1 µs. Note that this would be the worst case. This assumes that all of the operands are loaded from the external world as opposed to some of the operands being already in place as a result of a previous operation whose destination register is one of the math unit registers.

Performance of division. The actual division needs 20 internal clock cycles. This translates to 2.0 µs for the actual division at 10 MHz (internal clock speed). If the time to load operands and read results is included:

Number of internal clock cycles to load operands: 42
Number of internal clock cycles to read results: 24

The total internal clock cycles to perform a division is 86. This translates to 8.6 µs at 10 MHz.

Counter/Timers

This section describes the enhanced features of the counter/timers (CTC) on the Z86C93. It contains the register mapping of CTC registers and the bit functions of the newly added Timer2 control register.

In a standard Z8, there are two 8-bit programmable counter/timers (T0 and T1), each driven by its own 6-bit programmable prescaler. The T1 prescaler is driven by internal or external clock sources; however, the T0 prescaler is driven by the internal clock only.

The 6-bit prescalers divide the input frequency of the clock source by any integer number from 1 to 64. Each prescaler drives its counter, which decrements the value (1 to 256) that has been loaded into the counter. When the counter reaches the end of the count, a timer interrupt request IRQ4 (T0) or IRQ5 (T1), is generated.

Interrupts

The Z86C93 has six different interrupts from nine different sources. The interrupts are maskable and prioritized. The nine sources are divided as follow: four sources are claimed by Port 3 lines P30-P33, one in Serial Out, one in Serial In, and three in the counter/timers. The Interrupt Mask Register globally or individually enables or disables the six interrupt requests. When more than one interrupt is pending, priorities are resolved by a programmable priority encoder that is controlled by the Interrupt Priority register. All Z86C93 interrupts are vectored through locations in the program memory. When an interrupt machine cycle is activated an interrupt request is granted. Thus, this disables all of the subsequent interrupts, save the Program Counter and Status Flags, and then branches to the program memory vector location reserved for that interrupt. This memory location and the next byte contain the 16-bit address of the interrupt service routine for that particular interrupt request.

To accommodate polled interrupt systems, interrupt inputs are masked and the Interrupt Request register is polled to determine which of the interrupt requests need service. Software initiated interrupts are supported by setting the appropriate bit in the Interrupt Request Register (IRQ).

Internal interrupt requests are sampled on the falling edge of the last cycle of every instruction. The interrupt request must be valid 5TpC before the falling edge of the last clock cycle of the currently executing instruction.

When the device samples a valid interrupt request, the next 48 (external) clock cycles are used to prioritize the interrupt, and push the two PC bytes and the FLAG register on the stack. The following nine cycles are used to fetch the interrupt vector from external memory. The first byte of the interrupt service routine is fetched beginning on the 58th TpC cycle following the internal sample point, which corresponds to the 63rd TpC cycle following the external interrupt sample point.

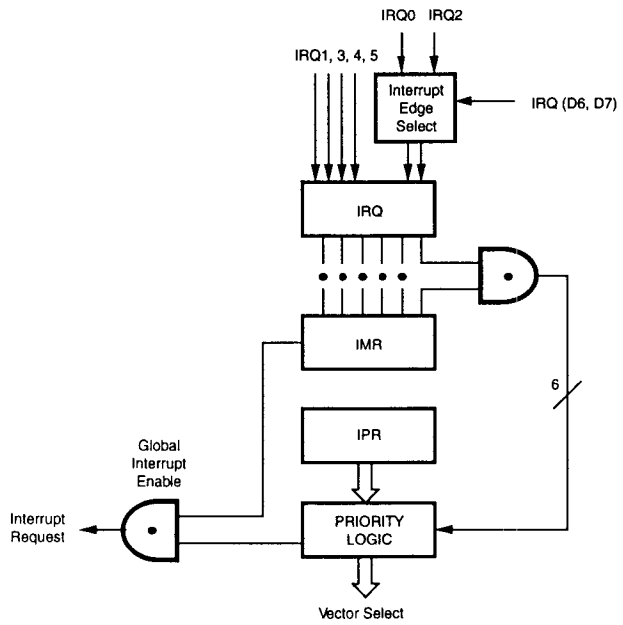


Figure 21. Interrupt Block Diagram

Table 7. Interrupt Types, Sources, and Vectors

Name	Source	Vector Location	Comments
IRQ 0	/DAV 0, P32, T2	0, 1	External (P32), Programmable Rise or Fall Edge Triggered
IRQ 1,	P33	2, 3	External (P33), Fall Edge Triggered
IRQ 2	/DAV 2, P31, T_{IN}	4, 5	External (P31), Programmable Rise or Fall Edge Triggered
IRQ 3	P30, Serial In	6, 7	External (P30), Fall Edge Triggered
IRQ 4	T0, Serial Out	8, 9	Internal
IRQ 5	TI	10, 11	Internal

Clock

The Z86C93 on-chip oscillator has a high-gain, parallel-resonant amplifier for connection to a crystal, LC, ceramic resonator, or any suitable external clock source (XTAL1=Input, XTAL2=Output). The external clock levels are not TTL. The crystal should be AT cut, 1 MHz to 25 MHz max, and series resistance (RS) is less than or equal to 100 Ohms. The crystal should be connected across XTAL1 and XTAL2 using the recommended capacitors (10 pF<CL<100 pF) from each pin to ground (Figure 20).

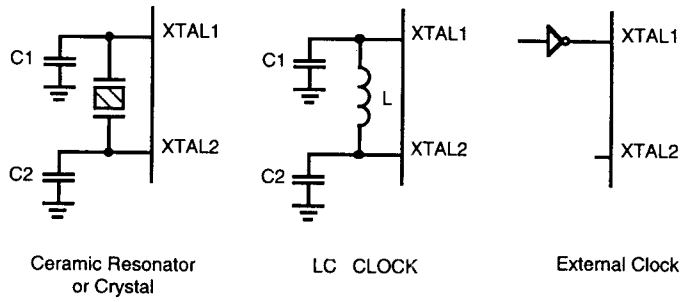


Figure 22. Oscillator Configuration

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	0.1xV_{CC}		V	Driven by External Clock Generator
V_{IH}	Input High Voltage	0.7xV_{CC}	V_{CC}		V	
V_{IL}	Input Low Voltage	-0.3	0.1xV_{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	0.8xV_{CC}	V_{CC}		V	
V_{RI}	Reset Input Low Voltage	-0.03	0.1xV_{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 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.

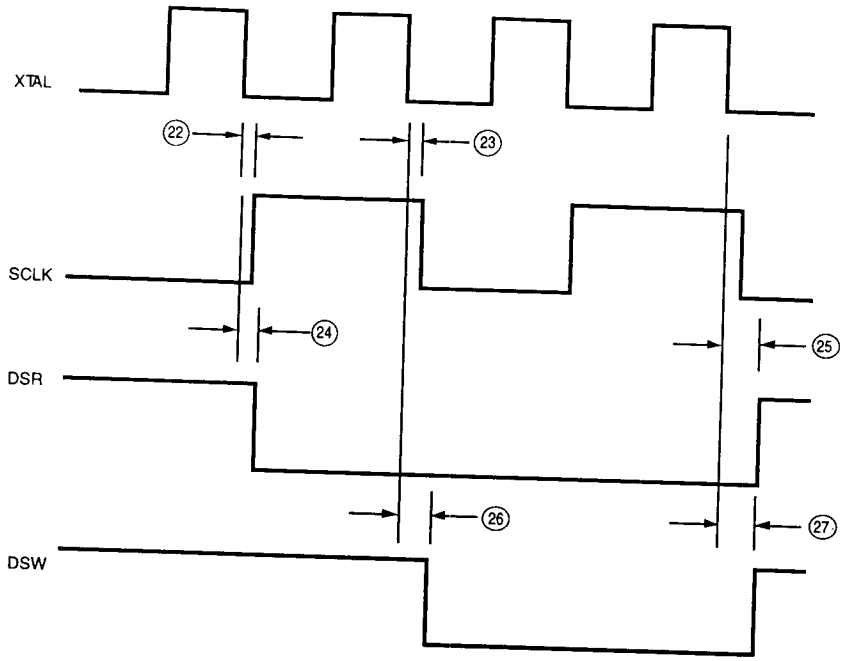


Figure 25. XTAL/SCLK To DSR and DSW Timing

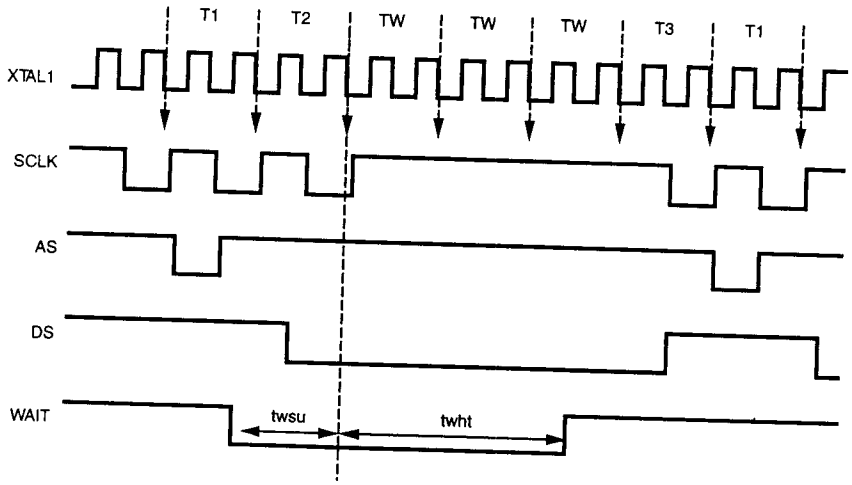


Figure 26. XTAL/SCLK To WAIT Timing (25 MHz Device Only)

AC CHARACTERISTICS
Additional Timing Diagram

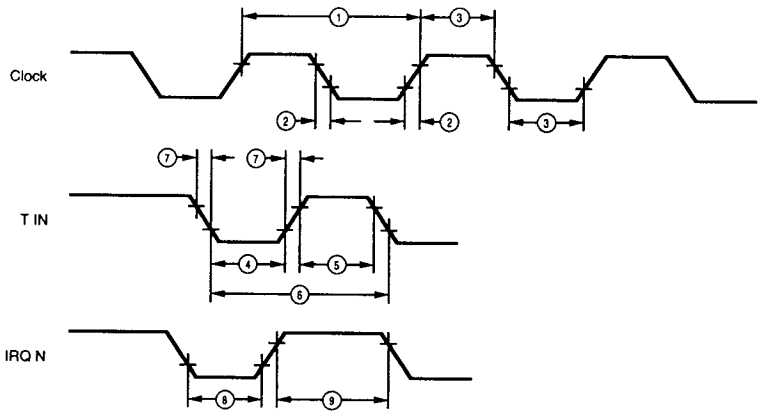


Figure 27. Additional Timing

AC CHARACTERISTICS
Additional Timing Table

No	Symbol	Parameter	T_A = 0°C to +70° C						Units	Notes
			33 MHz		24 MHz		20 MHz			
			Min	Max	Min	Max	Min	Max		
1	TpC	Input Clock Period	30	1000	42	1000	50	1000	ns	[1]
2	TrC,TfC	Clock Imput Rise & Fall Times		5		10		10	ns	[1]
3	TwC	Input Clock Width	10		11		15		ns	[1]
4	TwTinL	Timer Input Low Width	75		75		75		ns	[2]
5	TwTinH	Timer Input High Width	3TpC		3TpC		3TpC			[2]
6	TpTin	Timer Input Period	8TpC		8TpC		8TpC			[2]
7	TrTin,TfTin	Timer Input Rise & Fall Times	100		100		100		ns	[2]
8A	TwIL	Interrupt Request Input Low Times	70		70		70		ns	[2,4]
8B	TwIL	Interrupt Request Input Low Times	5TpC		5TpC		5TpC			[2,5]
9	TwIH	Interrupt Request Input High Times	3TpC		3TpC		3TpC			[2,3]

Notes:
[1] Clock timing references use 3.8V for a logic 1 and 0.8V for a logic 0.
[2] Timing references use 2.0V for a logic 1 and 0.8V for a logic 0.
[3] Interrupt references request via Port 3.
[4] Interrupt request via Port 3 (P31-P33)`.
[5] Interrupt request via Port 30.

AC CHARACTERISTICS

Handshake Timing Table

No	Symbol	Parameter	T_A = 0°C to +70°C Min	Max	Units	Data Direction
1	TsDI(DAV)	Data In Setup Time to /DAV	0		ns	In
2	ThDI(DAV)	RDY to Data In Hold Time	0		ns	In
3	TwDAV	/DAV Width	40		ns	In
4	TdDAVIf(RDYf)	/DAV to RDY Delay		70	ns	In
5	TdDAVIr(RDYr)	DAV Rise to RDY Wait Time		40	ns	In
6	TdRDYOr(DAVIf)	RDY Rise to DAV Delay	0		ns	In
7	TdDO(DAV)	Data Out to DAV Delay		TpC	ns	Out
8	TdDAVOf(RDYIf)	/DAV to RDY Delay	0		ns	Out
9	TdRDYIf(DAVOr)	RDY to /DAV Rise Delay		70	ns	Out
10	TwRDY	RDY Width	40		ns	Out
11	TdRDYIr(DAVOf)	RDY Rise to DAV Wait Time		40	ns	Out

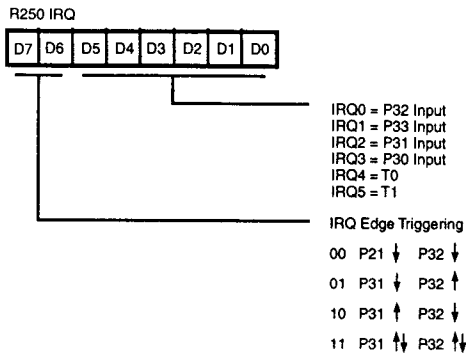


Figure 47. Interrupt Request Register (FAH: Read/Write)

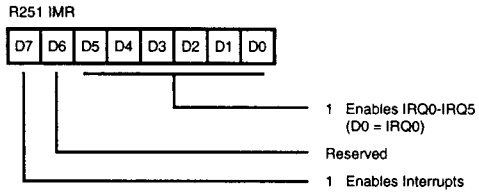


Figure 48. Interrupt Mask Register (FBH: Read/Write)

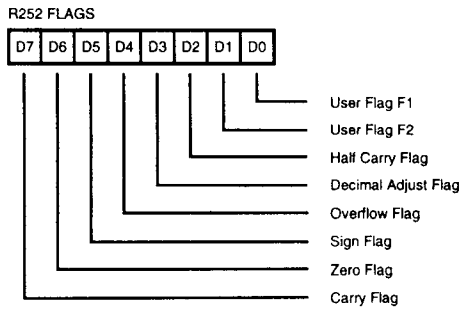


Figure 49. Flag Register (FCH: Read/Write)

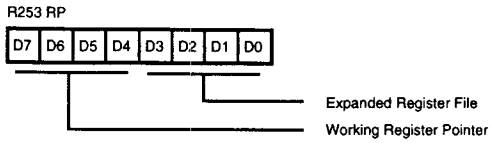


Figure 50. Register Pointer (FDH: Read/Write)

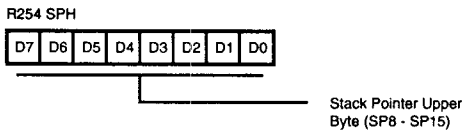


Figure 51. Stack Pointer High (FEH: Read/Write)

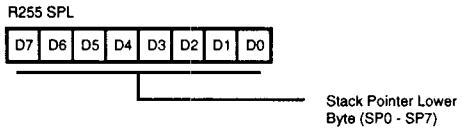
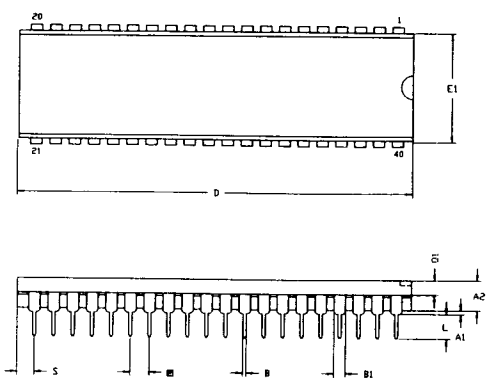


Figure 52. Stack Pointer Low (FFH: Read/Write)

INSTRUCTION SUMMARY (Continued)

Instruction and Operation	Address Mode dst	Address Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
ADC dst, src dst←dst + src +C	†		1[ ]	*	*	*	*	0	*
ADD dst, src dst←dst + src	†		0[ ]	*	*	*	*	0	*
AND dst, src dst←dst AND src	†		5[ ]	-	*	*	0	-	-
CALL dst SP←SP - 2 @SP←PC, PC←dst	DA IRR		D6 D4	-	-	-	-	-	-
CCF C←NOT C			EF	*	-	-	-	-	-
CLR dst dst←0	R IR		B0 B1	-	-	-	-	-	-
COM dst dst←NOT dst	R IR		60 61	-	*	*	0	-	-
CP dst, src dst - src	†		A[ ]	*	*	*	*	-	-
DA dst dst←DA dst	R IR		40 41	*	*	*	X	-	-
DEC dst dst←dst - 1	R IR		00 01	-	*	*	*	-	-
DECW dst dst←dst - 1	RR IR		80 81	-	*	*	*	-	-
DI IMR(7)←0			8F	-	-	-	-	-	-
DJNZr, dst r←r - 1 if r ≠ 0 PC←PC + dst Range: +127, -128	RA		rA r = 0 - F	-	-	-	-	-	-
EI IMR(7)←1			9F	-	-	-	-	-	-
HALT			7F	-	-	-	-	-	-

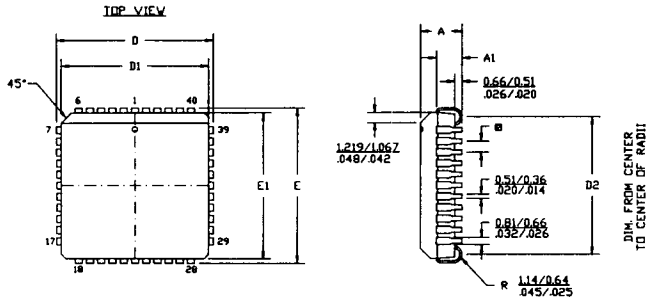
Instruction and Operation	Address Mode dst	Address Mode src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
INC dst dst←dst + 1	r		rE r = 0 - F	-	*	*	*	-	-
	R		20						
	IR		21						
INCW dst dst←dst + 1	RR IR		A0 A1	-	*	*	*	-	-
IRET FLAGS←@SP; SP←SP + 1 PC←@SP; SP←SP + 2; IMR(7)←1			BF	*	*	*	*	*	*
JP cc, dst if cc is true PC←dst	DA IRR		cD c = 0 - F 30	-	-	-	-	-	-
JR cc, dst if cc is true, PC←PC + dst Range: +127, -128	RA		cB c = 0 - F	-	-	-	-	-	-
LD dst, src dst←src	r	Im	rC	-	-	-	-	-	-
	r	R	r8						
	R	r	r9						
			r = 0 - F						
	r	X	C7						
	X	r	D7						
	r	Ir	E3						
	Ir	r	F3						
	R	R	E4						
	R	IR	E5						
	R	IM	E6						
	IR	IM	E7						
	IR	R	F5						
LDC dst, src	r	Irr	C2	-	-	-	-	-	-
LDCI dst, src dst←src r←r +1; rr←rr + 1	Ir	Irr	C3	-	-	-	-	-	-



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A1	0.51	0.81	020	.032
A2	3.25	3.43	128	.135
B	0.38	0.53	015	021
B1	1.02	1.52	040	060
C	0.23	0.38	009	015
D	52.07	52.58	2.050	2.070
E	15.24	15.75	.600	.620
E1	13.59	14.22	.535	.560
eB	2.54 TYP		100 TYP	
eA	15.49	16.51	.610	.650
L	3.18	3.81	.125	150
D1	1.52	1.91	060	075
S	1.52	2.29	.060	.090

CONTROLLING DIMENSIONS : INCH

40-Pin DIP Package Diagram



NOTES:
1. CONTROLLING DIMENSIONS : INCH
2. LEADS ARE COPLANAR WITHIN .004 IN.
3. DIMENSION : MM
INCH

SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.27	4.57	.168	.180
A1	2.67	2.92	.105	.115
D/E	17.40	17.65	.685	.695
D1/E1	16.51	16.66	.650	.656
D2	15.24	16.00	.600	.630
eB	1.27 TYP		.050 TYP	

44-Pin PLCC Package Diagram

Notes: