ZiLOG

ADVANCE INFORMATION SPECIFICATION

T-49-12-09

Z86C95

CMOS Z8® DIGITAL SIGNAL PROCESSOR

FEATURES

- Complete microcontroller, 16 I/O lines, and up to 64 Kbytes of addressable external space each for program and data memory.
- Embedded reduced instruction set DSP (Digital Signal Processor) for digital servo control, with 16-bit by 16-bit multiply and accumulate in one clock cycle.
- 8-channel, 8-bit A/D converter with track and hold and minimum single conversion time of 2 µsec.
- 8-bit D/A converter with 1/4 programmable gain stage and a maximum settling time of 3 µsec.
- Single channel 40/80 kHz pulse width modulator
- 256-byte register file, including 236 general-purpose registers, four I/O port registers and 16 status and control registers.
- 16-bit by 16-bit hardwired multiply and 32-bit by 16-bit divide, exclusive of DSP.
- Four external vectored priority interrupts for I/O, counter/timers and UART.
- On-chip oscillator that accepts crystal or external clock drive.
- Full-duplex UART
- 16-bit counter/timers with capture and compare registers.
- Register pointer for short, fast instructions to any one of the sixteen working register groups.
- Serial Peripheral Interface
- Multiplexed and demultiplexed address/data bus
- Single +5V power supply, all I/O pins TTL compatible
- 1.2 micron CMOS technology
- Clock speeds 20 and 24 MHz
- Three low-power standby modes; STOP, HALT and PAUSE
- Flash EPROM write support

GENERAL DESCRIPTION

The Z86C95 MCU (Microcontroller Unit) introduces a new level of sophistication for Superintegration™ ICS. The Z86C95 is a member of the Z8 single-chip microcontroller family incorporating a CMOS ROMless Z8 microcontroller with an embedded DSP processor for digital servo control. The DSP slave processor can perform 16-bit x 16-bit multiply and accumulate in one clock cycle. Additionally, the Z86C95 is further enhanced with a hardwired 16-bit x 16-bit multiplier and 32-bit/16-bit divider, three 16-bit counter timers with capture and compare registers, a half flash 8-bit A/D converter with a 2 µsec conversion time, an 8-bit DAC with 1/4 programmable gain stage, UART, serial peripheral interface, and a PWM output channel (Figure 1). It is fabricated using 1.2 micron CMOS technology and offered in an 80-pin QFP, 84-pin PLCC package, or a 100-pin VQFP (Figures 2 and 3).

The Z86C95 provides up to 16 output address lines. This permits 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 are provided via output address bits A15-A8.

GENERAL DESCRIPTION (Continued)

T-49-12-09

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. The Z86C95 contains 512 bytes of DSP Program RAM and 256 bytes of DSP data RAM.

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}

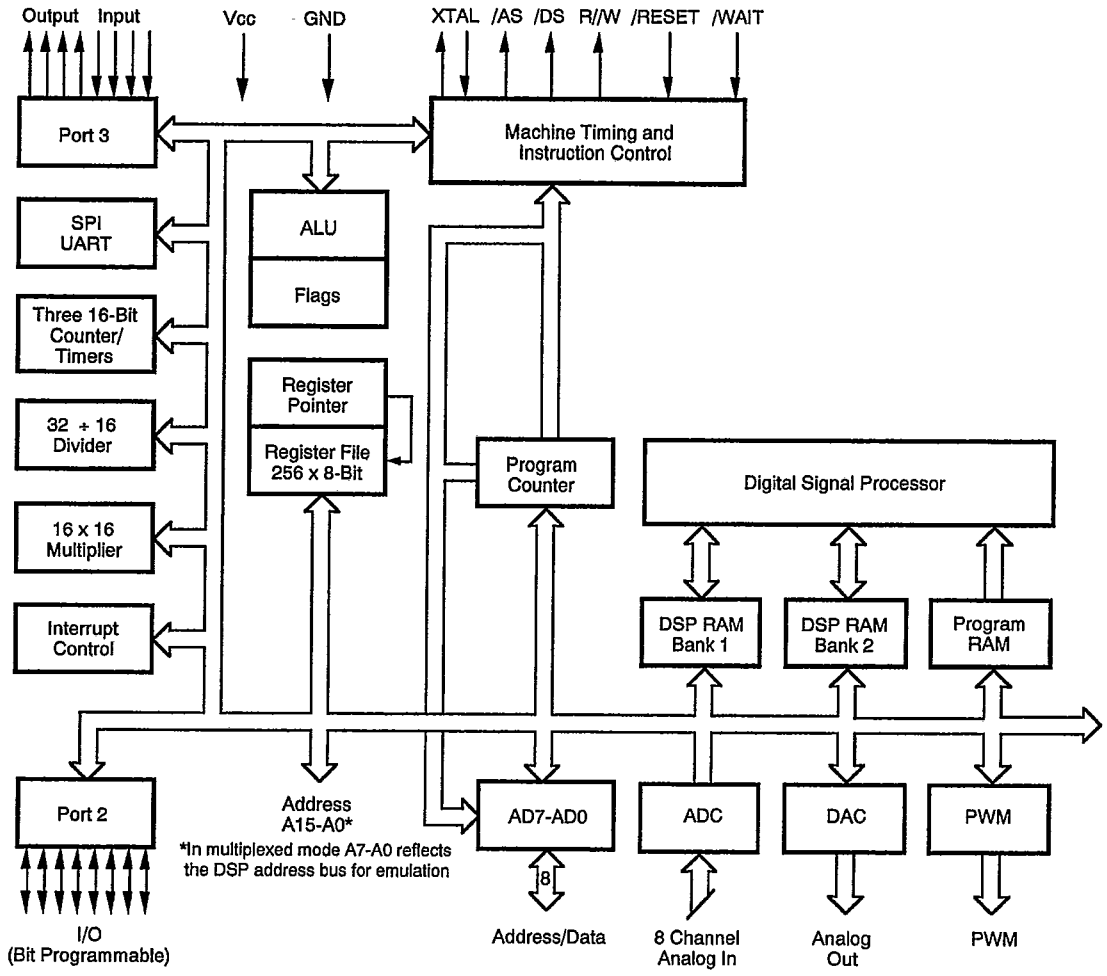


Figure 1. Functional Block Diagram

PIN DESCRIPTION

T-49-12-09

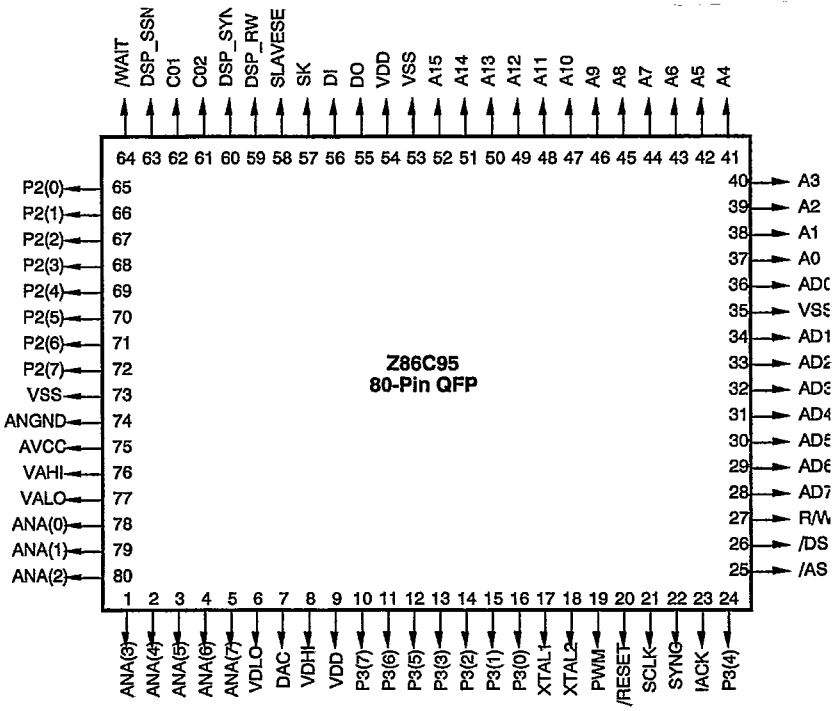


Figure 2. 80-Pin QFP Pin Assignments

PIN DESCRIPTION (Continued)

T-49-12-09

Table 1. 80-Pin QFP Pin Identification

No.	Symbol	Function	Direction
1-5	ANA(3)-ANA(7)	Input to A/D	Input
6	VD_{LO}	Low Ref Volt, DAC	Input
7	DAC	D/A Converter Output	Output
8	VD_{HI}	High Ref Volt, DAC	Input
9	V_{DD}	Digital Power Supply	Input
10-12	P3(7)-P3(5)	Port 3, Pins 7,6,5	Output
13-16	P3(3)-P3(0)	Port 3, Pins 3,2,1,0	Input
17	XTAL1	Crystal, OSC CLK	Input
18	XTAL2	Crystal, OSC CLK	Output
19	PWM	Pulse Width Modulator	Output
20	/RESET	Reset	Input
21	SCLK	System Clock	Output
22	SYNC	Synchronize Pin	Output
23	IACK	Interrupt Acknowledge	Output
24	P3(4)	Port 3, Pin 4	Output
25	/AS	Address Strobe	Output
26	/DS	Data Strobe	Output
27	R/W	Read/Write	Output
28-34	AD7-AD1	MUX ADD/DATA, Pins 7-1	Input/Output
35	V_{SS}	Digital Ground	Input
36	AD0	MUX ADD/DATA, Pin 0	Input/Output
37-52	A15-A0	External Address	Output
53	V_{SS}	Digital Ground	Input
54	V_{DD}	Digital Power Supply	Input
55	DO	SPI Data Out	Output
56	DI	SPI Data In	Input
57	SK	SPI Clock	Input/Output
58	SLAVESEL	Slave Select	Input
59	DSP_RW	DSP Emulation R/W Pin	Output
60	DSP_SYNC	DSP Emulation Sync Pin	Output
61-62	CO2-CO1	Compare Outputs for Timer 2	Output
63	DSP_SSN	DSP Emulation Single Step Pin	Input
64	/WAIT	Wait	Input
65-72	P2(0)-P2(7)	Port 2, Pins 0-7	Input, Output
73	V_{SS}	Digital Ground	Input
74	AN_{GND}	Analog Ground	Input
75	AV_{CC}	Analog Power Supply	Input
76	VA_{HI}	High Ref Volt, A/D	Input
77	VA_{LO}	Low Ref Volt, A/D	Input
78-80	ANA(0)-ANA(2)	Input to A/D	Input

T-49-12-09

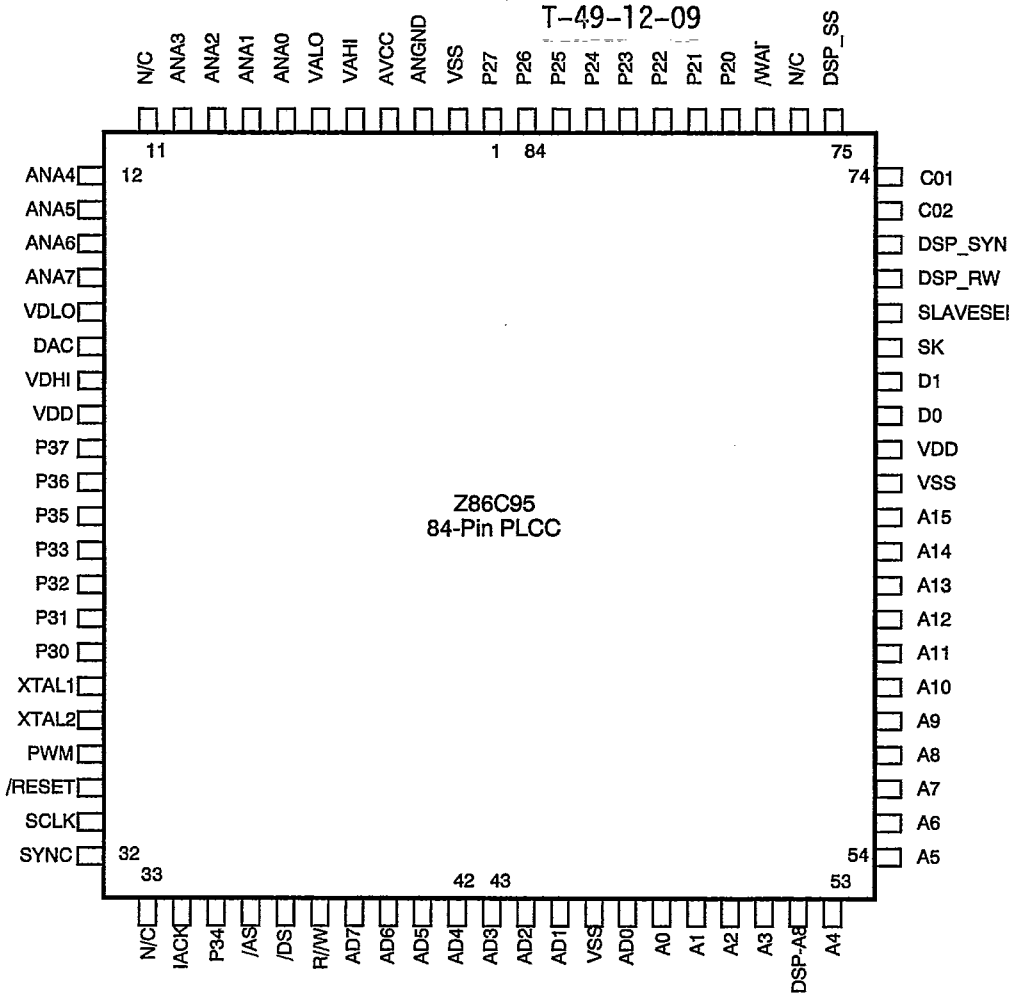


Figure 3. 84-Pin PLCC Pin Assignments

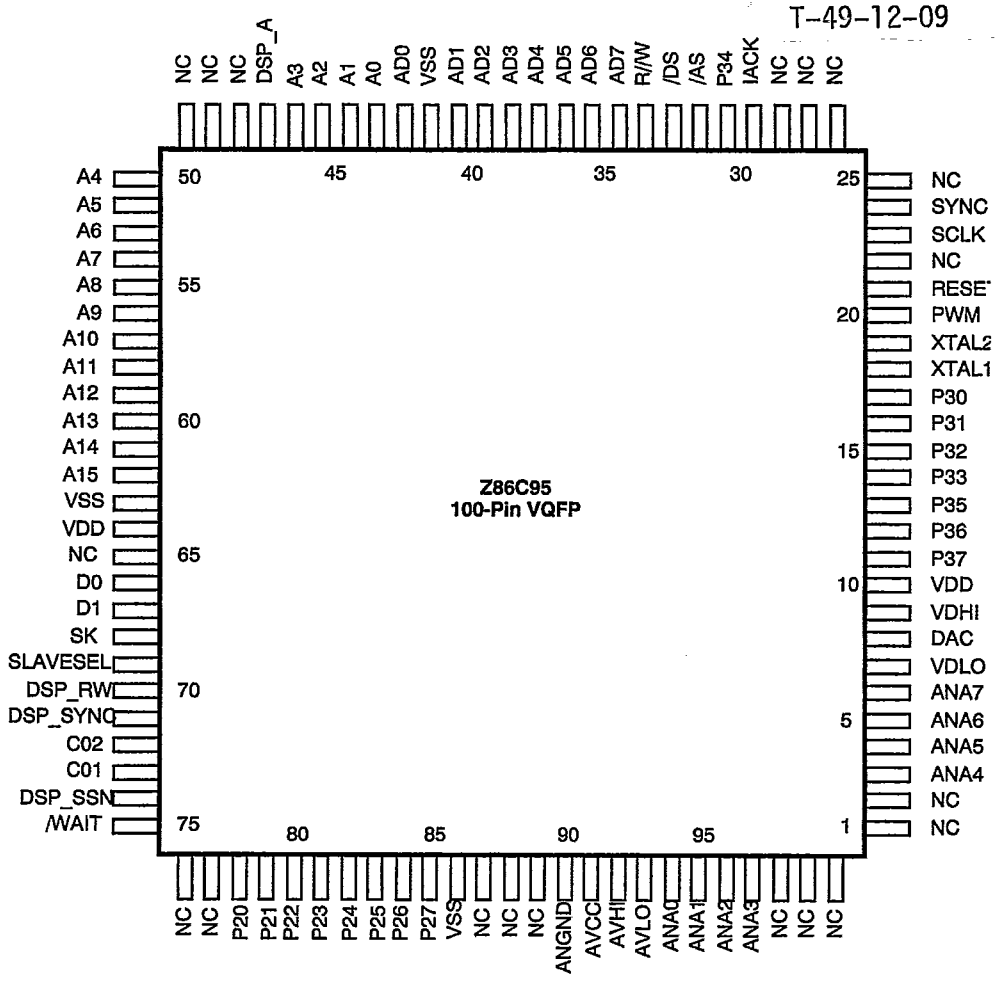
PIN DESCRIPTION (Continued)

T-49-12-09

Table 2. 84-Pin PLCC Pin Identification

No.	Symbol	Function	Direction
1	P27	Port 2 Pin 7	Input/Output
2	V_{SS}	Digital Ground	Input
3	AN_{GND}	Analog Ground	Input
4	AV_{CC}	Analog Power Supply	Input
5	VA_{HI}	High Ref Volt, A/D	Input
6	VA_{LO}	Low Ref Volt, A/D	Input
7-10	ANA0-ANA3	Input to A/D, Pins 0-3	Input
12-15	ANA4-ANA7	Input to AD, Pins 5-7	Input
16	VD_{LO}	Low Ref Volt, DAC	Input
17	DAC	D/A Converter Output	Output
18	VD_{HI}	High Ref Volt, DAC	Input
19	V_{DD}	Digital Power Supply	Input
20-22	P37-P35	Port 3, Pins 7-5	Output
23-26	P33-P30	Port 3, Pins 3-0	Input
27	XTAL1	Crystal, OSC CLK	Input
28	XTAL2	Crystal, OSC CLK	Output
29	PWM	Pulse Width Modulator	Output
30	/RESET	Reset	Input
31	SCLK	System Clock	Output
32	SYNC	Z8 Emulation Sync Pin	Output
33	NC	No connection	
34	IACK	Interrupt Acknowledge	Output
35	P34	Port 3, Pin 4	Output
36	/AS	Address Strobe	Output
37	/DS	Data Strobe	Output
38	R/W	Read/Write	Output
39-45	AD7-AD1	MUX ADD/DATA, Pins 7-1	Input/Output
46	V_{SS}	Digital Ground	Input
47	AD0	MUX AD0/DATA Pin 0	Input/Output
48-51	A3-A0	External Address	Output
52	DSP-A8	MSB of DSP PC	Output
53	A4	External Address	Output
54-64	A15-A5	External Address	Output
65	V_{SS}	Digital Ground	Input
66	V_{DD}	Digital Power Supply	Input
67	DO	SPI Data Out	Output
68	DI	SPI Data In	Input
69	SK	SPI Clock	Input/Output
70	SLAVESEL	Slave Select	Input
71	DSP_RW	DSP Emulation R/W Pin	Output
72	DSP_SYNC	DSP Emulation SYNC Pin	Output
73-74	C02-C01	Compare Outputs for Timer 2	Output
75	DSP_SSN	DSP Emulation Single Step Pin	Output
76	N/C	No Connection	
77	/WAIT	Wait	Input
78-84	P20-P26	Port 2, Pins 0-6	Input/Output

T-49-12-09



100-Pin VQFP Pin Assignments

PIN DESCRIPTION (Continued)

T-49-12-09

100-Pin VQFP Pin Identification

No.	Symbol	Function	Direction
1-2	NC	No connection	
3-6	ANA4-ANA7	Input to AD, Pins 5-7	Input
7	VD_{LO}	Low Ref Volt, DAC	Input
8	DAC	D/A Converter Output	Output
9	VD_{HI}	High Ref Volt, DAC	Input
10	V_{DD}	Digital Power Supply	Input
11-13	P37-P35	Port 3, Pins 7-5	Output
14-17	P33-P30	Port 3, Pins 3-0	Input
18	XTAL1	Crystal, OSC CLK	Input
19	XTAL2	Crystal, OSC CLK	Output
20	PWM	Pulse Width Modulator	Output
21	/RESET	Reset	Input
22	NC	No connection	
23	SCLK	System Clock	Output
24	SYNC	Synchronize Pin	Output
25-28	NC	No connection	
29	IACK	Interrupt Acknowledge	Output
30	P34	Port 3, Pin 4	Output
31	/AS	Address Strobe	Output
32	/DS	Data Strobe	Output
33	R/W	Read/Write	Output
34-40	AD7-AD1	MUX ADD/DATA, Pins 7-1	Input/Output
41	V_{SS}	Digital Ground	Input
42	AD0	MUX AD0/DATA Pin 0	Input/Output
43-46	A3-A0	External Address	Output
47	DSP-A8	MSB of DSP PC	Output
48-50	N/C	No Connection	
51-62	A15-A5	External Address	Output
63	V_{SS}	Digital Ground	Input
64	V_{DD}	Digital Power Supply	Input
65	N/C	No Connection	
66	D0	SPI Data Out	Output
67	D1	SPI Data In	Input
68	SK	SPI Clock	Input/Output
69	SLAVESEL	Slave Select	Input
70	DSP_RW	DSP Emulation R/W Pin	Output
71	DSP_SYNC	DSP Emulation SYNC Pin	Output
72-73	C02-C01	Compare Outputs for Timer 2	Output
74	DSP_SSN	DSP Emulation Single Step Pin	Output
75	/WAIT	Wait	Input
76-77	N/C	No Connection	
78-85	P20-P27	Port 2, Pins 0-6	Input/Output
86	V_{SS}	Digital Ground	Input
87-89	N/C	No Connection	
90	AN_{GND}	Analog Ground	Input
91	AV_{CC}	Analog Power Supply	Input
92	VA_{HI}	High Ref Volt, A/D	Input
93	VA_{LO}	Low Ref Volt, A/D	Input
94-97	ANA0-ANA3	Input to A/D, Pins 0-3	Input
98-100	N/C	No Connection	

PIN FUNCTIONS

T-49-12-09

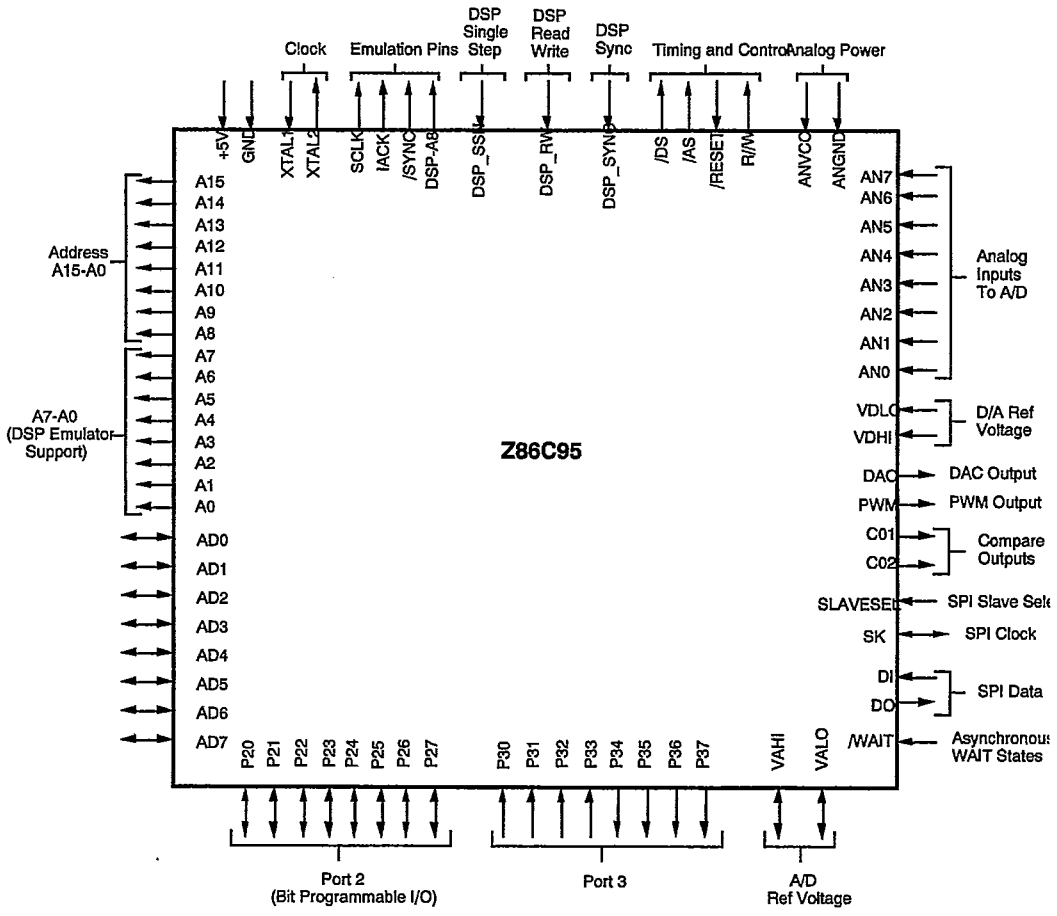


Figure 4. Pin Functions

/DS (output, active Low). Data Strobe is activated once for each external memory transfer. For a READ operation, data must be available prior to the trailing edge of /DS. For WRITE operations, the falling edge of /DS indicates that output data is valid. Data Strobe will tri-state in reset.

/AS (output, active Low). Address Strobe is pulsed once at the beginning of each machine cycle. Memory address transfers are valid at the trailing edge of /AS. Under program control, /AS can be placed in the high-impedance state along with Port 1, Data Strobe, and Read/Write.

PIN FUNCTIONS (Continued)

T-49-12-09

/RESET (input, active Low). To avoid asynchronous and noisy reset problems, the Z86C95 is equipped with a reset filter of four external clocks (4TpC). If the external /RESET signal is less than 4TpC in duration, no reset occurs.

On the 5th clock after the /RESET is detected, an internal RST signal is latched and held for an internal register count of 18 external clocks, or for the duration of the external /RESET, whichever is longer. During the reset cycle, /DS is held active Low while /AS cycles at a rate of TpC/2. When /RESET is deactivated, program execution begins at location 000CH. Reset time must be held Low for 50 ms or until V_{DD} is stable, whichever is longer.

XTAL1, XTAL2. *Crystal 1, Crystal 2* (time-based input and output, respectively). These pins connect a parallel-resonant crystal, ceramic resonator, LC, or any external single-phase clock to the on-chip oscillator and buffer.

R/W (output, read High/write Low). The Read/Write signal is low when the MCU is writing to the external program or data memory. Will tri-state in reset.

A15-A8 (output). Demultiplexed high byte of external address bus. Auto latch when in reset.

A7-A0 (output). Demultiplexed low byte of external address bus.

AD7-AD0 (input, output). Multiplexed Z8 address/data bus. Auto latch when in reset.

AN7-AN0 (analog input). Analog inputs to the A/D converter.

DAC (output). Analog output of the D/A converter.

PWM (output). Pulse Width Modulator output. Open Drain.

CO1 (output). Compare output1 for timer T2.

CO2 (output). Compare output2 for timer T2.

SLAVESEL (input, active Low). SPI Slave Select is used in Slave mode to mark the beginning and end of a transaction.

SK (input, output). SPI clock.

DI (input, active High). SPI serial data input in both master and slave mode.

DO (output, active High). SPI serial data output.

/WAIT (input, active Low). Introduces asynchronous wait states into the external memory fetch cycle. When this input goes Low during an external memory access, the Z8 freezes the fetch cycle until this pin goes High. This pin is sampled after /DS goes Low; should be pulled up if not used.

VA_{HI} (input). Reference voltage (High) for the A/D converter.

VA_{LO} (input). Reference voltage (Low) for the A/D converter.

ANV_{CC} (input). Analog power supply for A/D and D/A.

AN_{GND} (input). Analog ground for A/D and D/A.

VD_{HI} (input). Reference voltage (High) for D/A converter.

VD_{LO} (input). Reference voltage (Low) for D/A converter.

SSTEP (input, active High). DSP single-step control pin. The DSP processor will execute a NOP instruction and hold the program counter value when this pin is High. /SSTEP is synchronized with the system clock; should be pulled Low if not used.

SCLK System Clock (output). The internal system clock is available at this pin.

IACK Interrupt Acknowledge (output, active High). This output, when High, indicates that the Z86C95 is in an interrupt cycle.

/SYNC (output, active Low). This signal indicates the last clock cycle of the currently executing instruction.

Port 2 P20-P27. Port 2 is an 8-bit, bit programmable, bidirectional, CMOS compatible port. Each of these eight I/O lines can be independently programmed as an input or output or globally as an open-drain output. Port 2 is always available for I/O operation. When used as an I/O port, Port 2 may be placed under handshake control. In this configuration, Port 3 lines P31 (Port 3, bit 1) and P36 are used as the handshake controls lines /DAV2 and RDY2. The handshake signal assignment for Port 3 lines P31 and P36 is dictated by the direction (input or output) assigned to P27 (Figure 5).

T-49-12-09

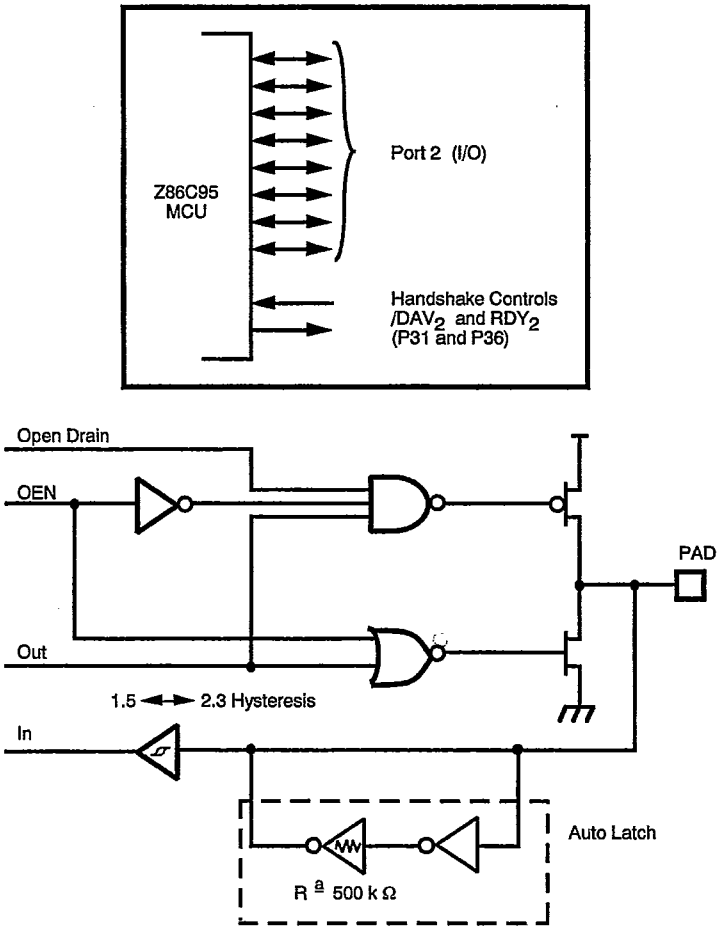


Figure 5. Port 2 Configuration

PIN FUNCTIONS (Continued)

T-49-12-09

Port 3 P30-P37. Port 3 is an 8-bit, CMOS compatible four-fixed input and four-fixed output port. These 8 I/O lines have four-fixed (P33-P30) input and four-fixed (P37-P34) output ports (Table 3). Port 3 pins P30 and P37, when used as serial I/O, are programmed as serial in and serial out, respectively (Figure 6).

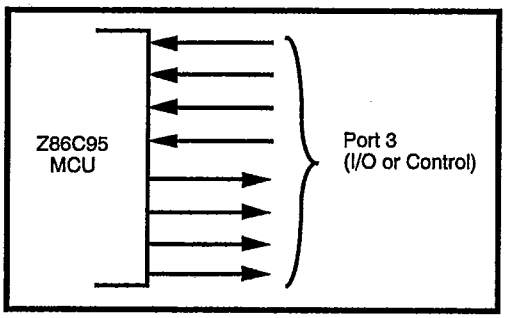


Figure 6. Port 3 Configuration

Port 3 is configured under software control to provide the following control functions: Handshakes for Port 2 (/DAV and RDY); four external interrupt request signals (IRQ0-IRQ3); timer input and output signals (T_{IN} and T_{OUT}), and Data Memory Select (/DM).

Port 3 lines P30 and P37, can be programmed as serial I/O lines for full-duplex serial asynchronous receiver/transmitter operation. The bit rate is controlled by the Counter/Timer 0.

Table 3. Port 3 Pin Assignments

Pin #	I/O	CTC1	Int.	P2HS	UART	Ext.
P30	In		IRQ3		Serial In	
P31	In	T_{IN}	IRQ2	D/R		
P32	In		IRQ0			
P33	In		IRQ1			
P34	Out					DM
P35	Out					
P36	Out	T_{OUT}		R/D		
P37	Out				Serial Out	

The Z86C95 automatically adds a start bit and two stop bits to transmitted data (Figure 7). Odd parity is also available as an option. Eight data bits are always transmitted, regardless of parity selection. If parity is enabled, the eighth bit is the odd parity bit. An interrupt request (IRQ4) is generated on all transmitted characters.

Received data must have a start bit, eight data bits and at least one stop bit. If parity is On, bit 7 of the received data is replaced by a parity error flag. Received characters generate the IRQ3 interrupt request.

Auto Latch. The Auto Latch puts valid CMOS levels on all CMOS inputs that are not externally driven. Whether this level is 0 or 1, cannot be determined. A valid CMOS level rather, than a floating node, reduces excessive supply current flow in the input buffer.

T-49-12-09

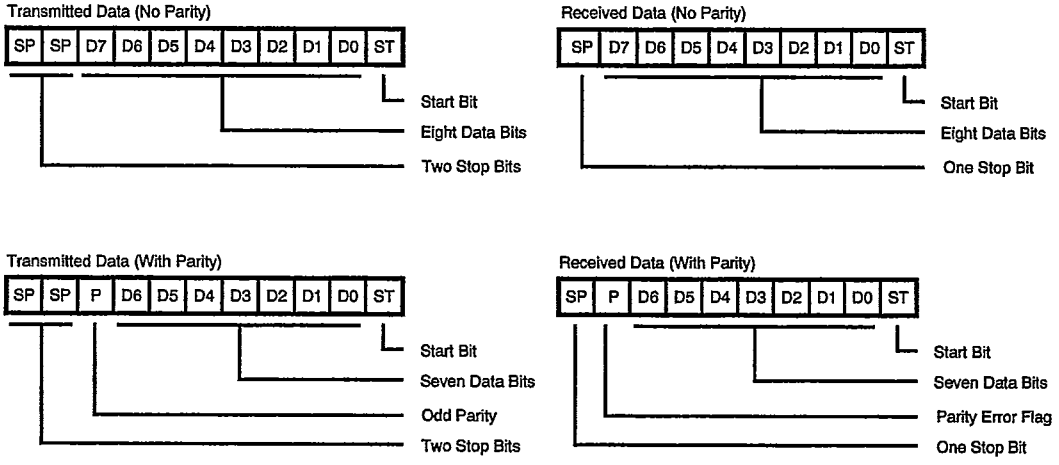


Figure 7. Serial Data Formats

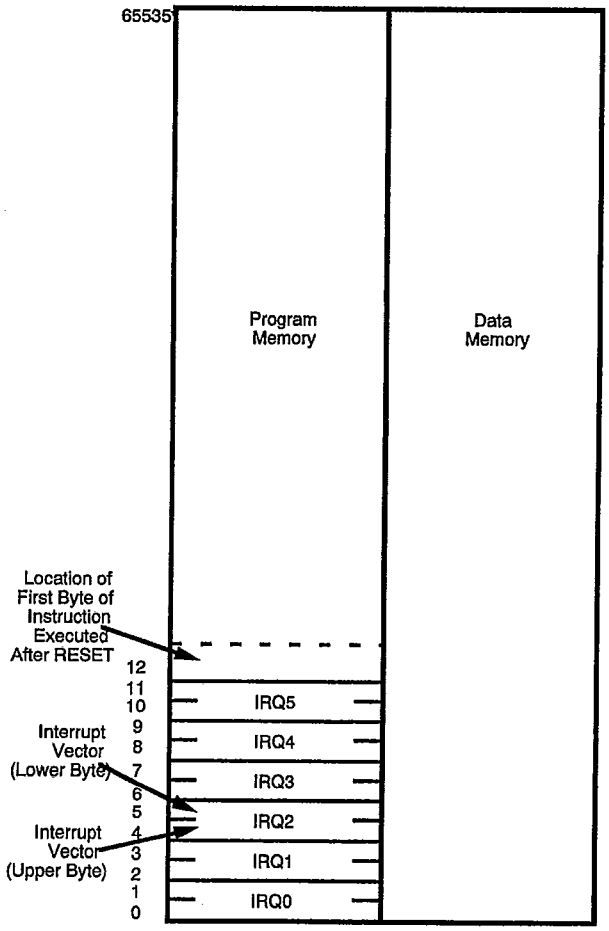
ADDRESS SPACE

T-49-12-09

Program Memory. The Z86C95 can address up to 64 Kbytes of external program memory (Figure 8). Program execution begins at external location 000CH after a reset.

Data Memory (/DM). The Z86C95 can address up to 64 Kbytes of external data memory (Figure 9). External data memory may be 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 (Port 3, bit 4), is used to distinguish between data and program memory space. 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. Data Memory will tri-state in reset.

Register Memory Map. The Z86C95 register memory space is split into five register files; the original Z8 Register File, Expanded Register File A (ERF-A), Expanded Register File B (ERF-B), Expanded Register File C (ERF-C) and Expanded Register File D (ERF-D). (Figure 9).



*All addresses in Hexadecimal

Figure 8. Program and Data Memory Configuration

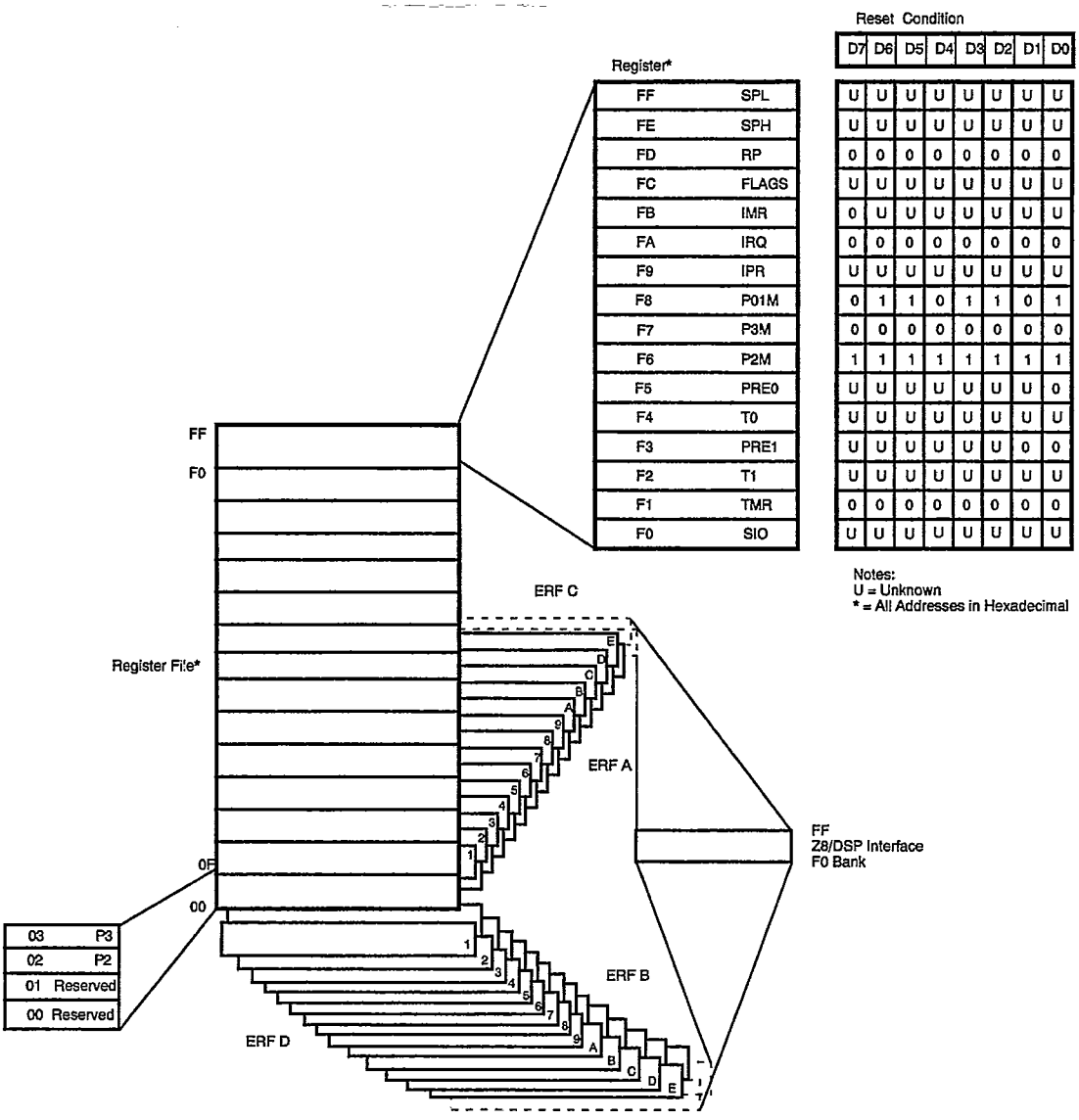


Figure 9. Register File

MEMORY INTERFACE (Continued)

T-49-12-09

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 Z86C95 also allows short 4-bit register addressing using the Register Pointer (Figures 10-11). In the 4-bit mode, the Register File is divided into 16 working register groups, each occupying 16 contiguous locations. The Register Pointer addresses the starting location of the active working-register group.

Note: Expanded Register Banks can only be accessed through working register and indirect addressing modes.

Expanded Register File. The register memory has been further expanded into four additional register files known as Expanded Register Files A thru D. Each of these register files contain 15 banks of 16 registers per bank. ERF-A stores data for the DSP processor in nine banks of its register space as well as system control registers and peripheral device registers in the remaining six banks. ERF-B contains the remaining four banks of DSP data memory (total DSP data memory is 208 bytes [accessible by the Z8]) as well as ten banks of DSP program memory. ERF-C contains fourteen banks of DSP program memory, and ERF-D contains eight banks of DSP program memory making a total of 512 bytes. Bank F is common to all four Expanded Register Files. Register (F8H) in bank F is the Z8/DSP control register. This register allows a quick means of switching between register files while in the DSP. To do this, bits 5 and 6 of the Z8/DSP control register are used as follows: D6/5 - 00 for ERF-A, D6/5 - 01 for ERF-B,D6/5 - 10 for ERF-C,D6/5 - 11 for ERF-D. On power-up, bits 5 and 6 are reset to 0 thereby enabling access to ERF-A. Bits 7-4 of the register pointer, RP, select the working register bank of the register file while bits 3-0 of the register pointer, RP, selects the working register bank of the Expanded Register File. Once an expanded register bank is selected it is effectively overlayed onto Bank 0 of the Z8's working register file. When an expanded register bank is selected, access to the Z8's ports is turned off.

Stack. The Z86C95 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 (SPI-Bits 15-8) can be used as a general purpose register when using internal stack only.

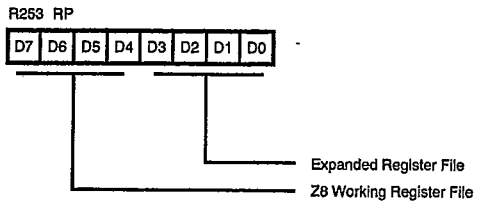


Figure 10. Register Pointer Register

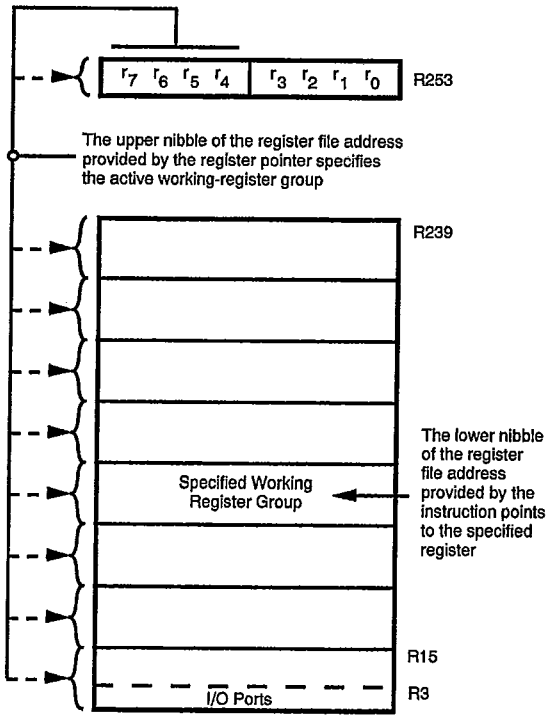


Figure 11. Register Pointer

Z8/DSP MEMORY INTERFACE

T-49-12-09

There are three types of memory spaces residing in the Z8/DSP interface:

1. DSP Program Memory. The size of this memory is 512 bytes. This memory space is mapped into ERF-B, C, and D of the Z8®. It occupies bank 1 through bank A in ERF-B, bank 1 through bank E in ERF-C, and bank 1 through bank 8 in ERF-D. (Figure 12).

2. DSP Data Memory. There are two data memory banks each 64 x 16 in size called DSP RAM0 and DSP RAM1. This translates to 256 bytes. However, only 208 bytes out of 256 are shared between the Z8 and the DSP. Out of this 208 bytes, 144 bytes are mapped to Bank 1 through Bank 9 of ERF-A. The remaining bytes are mapped into Bank B through Bank E of ERF-B (Figure 12).

3. Z8/DSP Interface Registers. The register mapping of the various registers which are part of the Z8/DSP interface are shown in Figure 13.

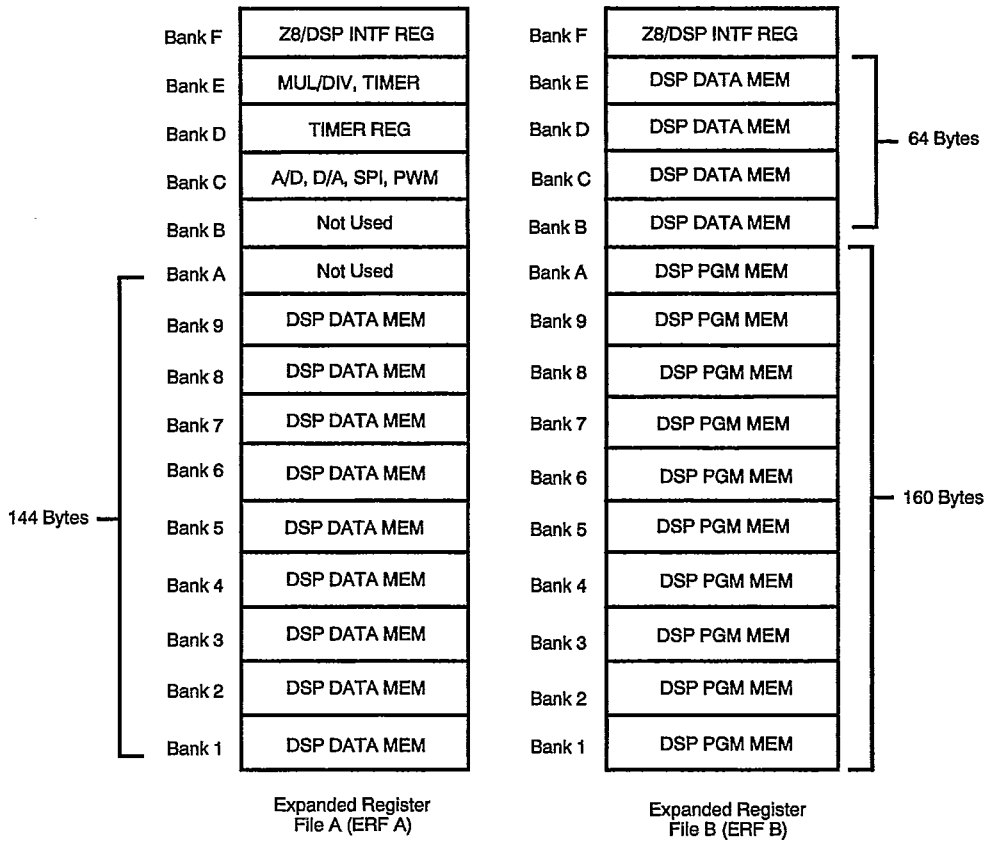


Figure 12. DSP Program and Data Memory

Z8/DSP MEMORY INTERFACE (Continued)

T-49-12-09

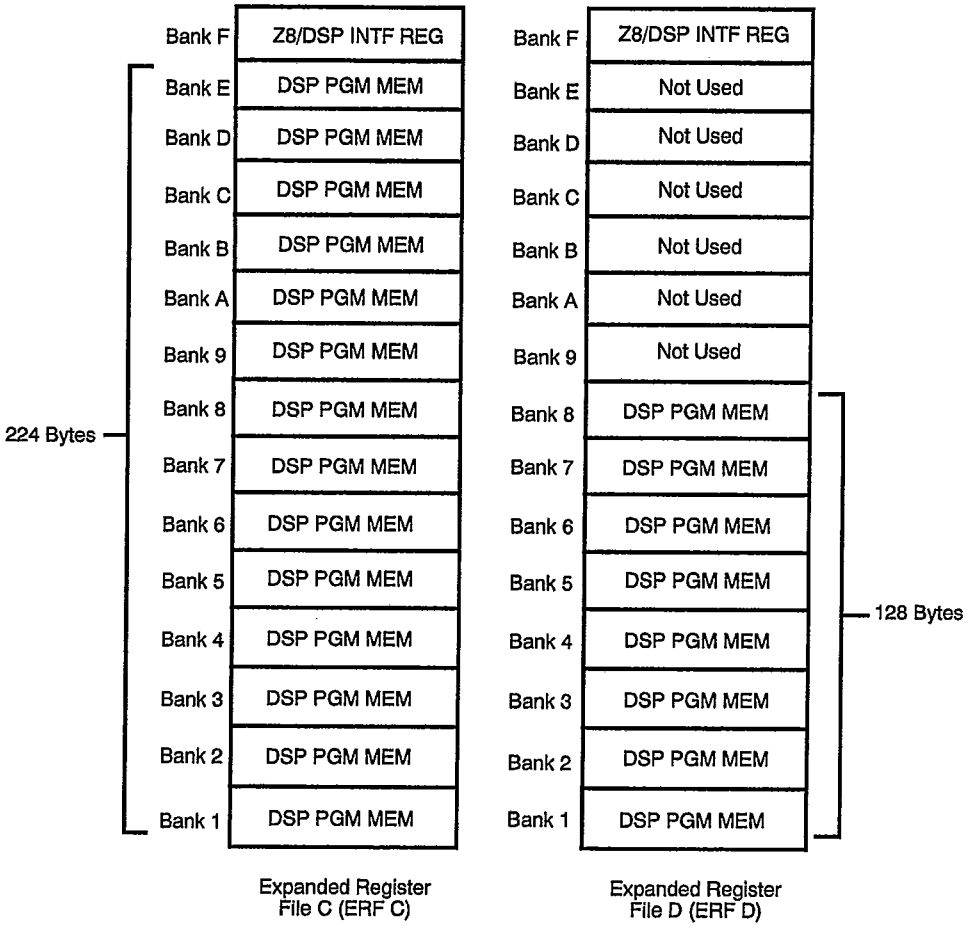


Figure 12. DSP Program and Data Memory (Continued)

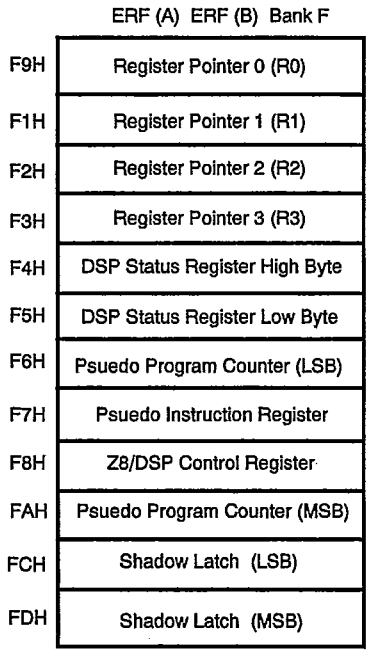


Figure 13. Z8/DSP Interface Register Mapping

The details of the data memory mapping between the Z8 and the DSP are shown in Figures 14 and 15. For example, Bank 1 of ERF-A is split between DSP RAM1 and RAM0. Bytes 15 through 8 are mapped to DSP RAM1 and bytes 7 through 0 are mapped to DSP RAM0. Similarly, Banks 2, 3 and so on are all split between RAM0 for the first 8 bytes and RAM1 for the last 8 bytes. Also, notice that the higher order bits (15 through 8 of the DSP word) are mapped to an even number byte of the Z8 and the lower order bits of the DSP (7 through 0) are mapped to the odd numbered bytes of the Z8. The size of DSP RAM1 and RAM0 is 64 16-bit words each. These occupy hex addresses 00 through 3F. The following is the bank mapping of Z8 ERF-A and ERF-B to the DSP RAM1 and RAM0.

DSP RAM1/RAM0	Z8 Bank
00 - 03	Bank 1 of ERF-A
04 - 07	Bank 2 of ERF-A
08 - 0B	Bank 3 of ERF-A
0C - 0F	Bank 4 of ERF-A
10 - 13	Bank 5 of ERF-A
14 - 17	Bank 6 of ERF-A
18 - 1B	Bank 7 of ERF-A
1C - 1F	Bank 8 of ERF-A
20 - 23	Bank 9 of ERF-A
24 - 27	Bank B of ERF-B
28 - 2B	Bank C of ERF-B
2C - 2F	Bank D of ERF-B
30 - 33	Bank E of ERF-B
34 - 3F	Not mapped to Z8

Z8/DSP MEMORY INTERFACE (Continued)

T-49-12-09

Access to a working bank in ERF-A is achieved by selecting the appropriate lower four bits, 3-0 of the Register Pointer, RP located within the Z8's Standard Register Bank. Bits 5 and 6 of the Z8/DSP control register are used to access the remaining register files as follows: D6/5 - 00 for ERF-A, D6/5 - 01 for ERF-B,D6/5 - 10 for ERF-C,D6/5 - 11 for ERF-D. Notice that bank F in ERF-B or C or D is the same as that of ERF-A. This provides common access to the Z8/DSP control register which allows movement from any register file to any other file. In other words, all the registers in bank F can be accessed from any of the four ERFs.

The interface registers (except the Z8/DSP control register) program memory and data memory of the DSP can not be accessed while the DSP is executing from the internal program memory.

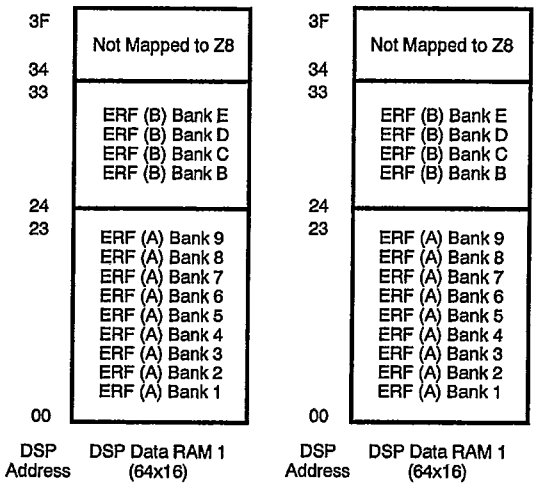


Figure 14. Data Memory Mapping of Z8 and DSP

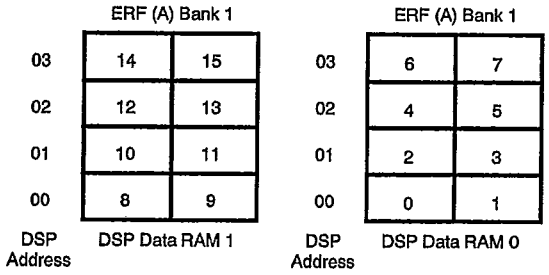


Figure 15. Close-Up of ERF (A) Bank 1 Byte Addressing

T-49-12-09

Z8/DSP Interface. The block diagram of the Z8/DSP interface logic and shared RAM between the Z8 and DSP is shown in Figure 16a and 16b.

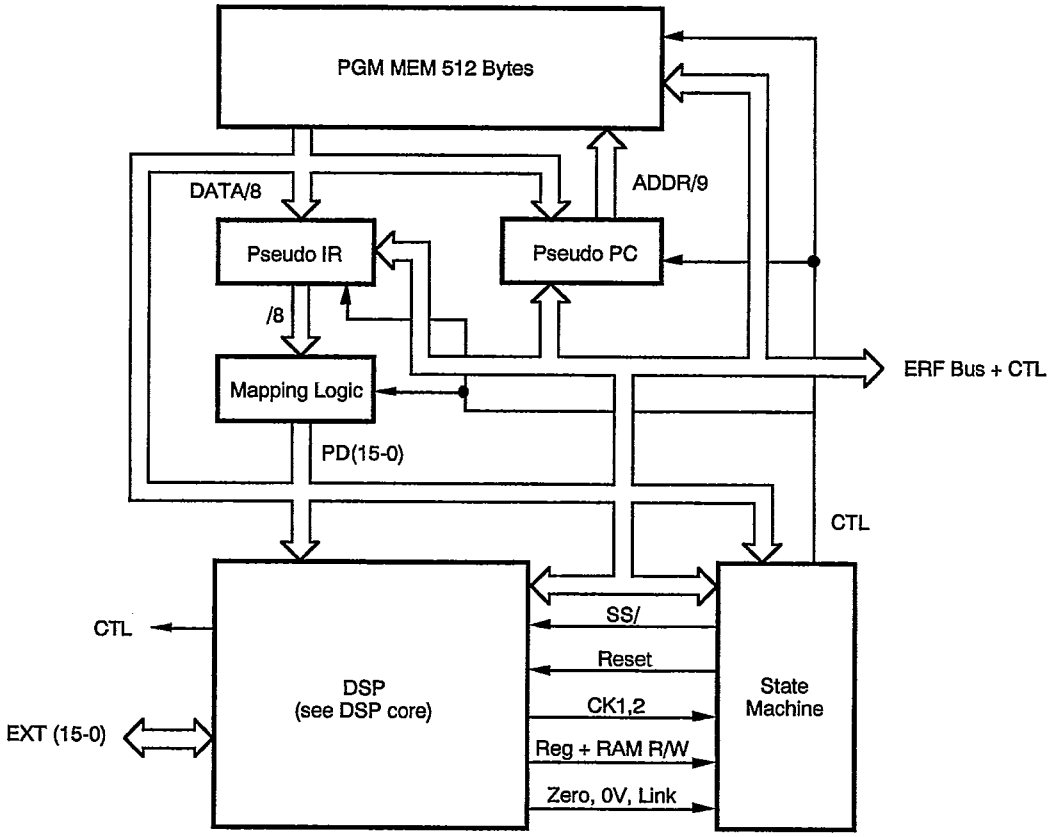


Figure 16a. Block Diagram of Z8/DSP Interface

Z8/DSP MEMORY INTERFACE (Continued) T-49-12-09

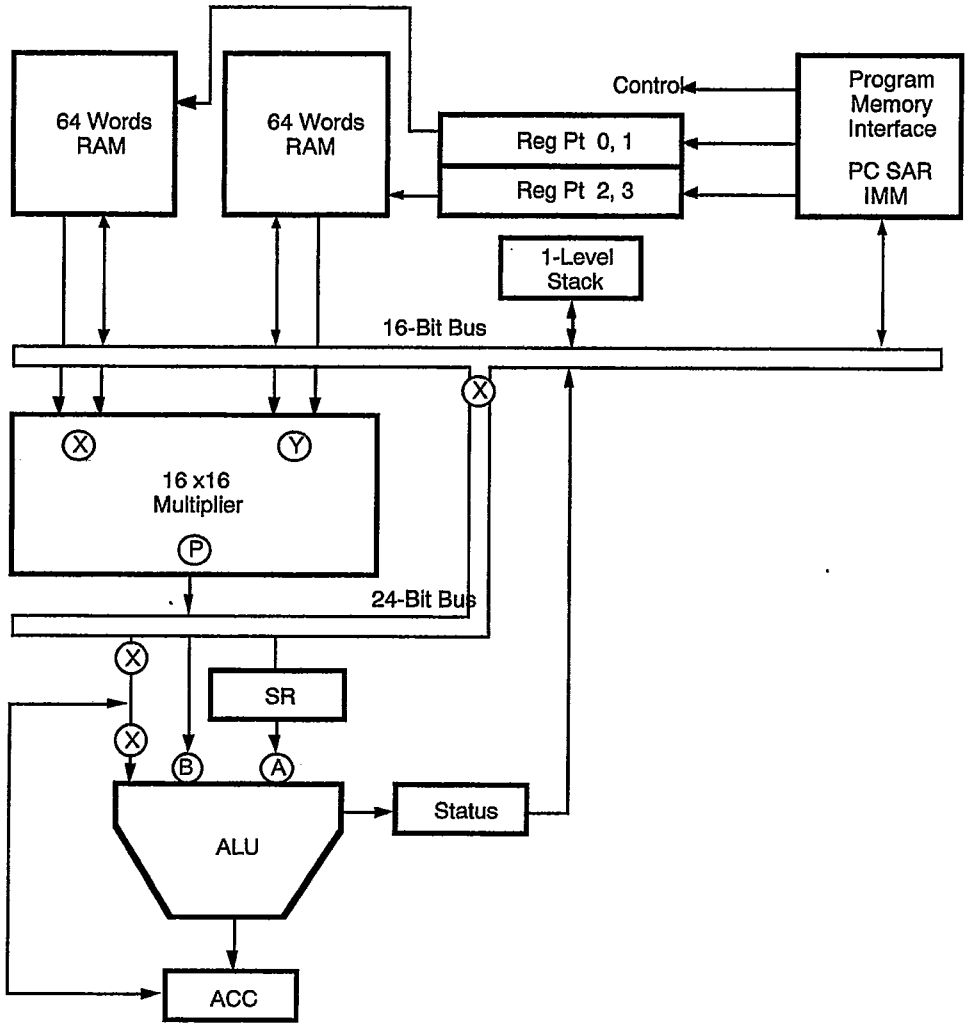


Figure 16b. DSP Core

T-49-12-09

Access to DSP Processor

There are three ways to instruct the DSP to execute instructions.

1. Via the internal program memory (512 Bytes):

   The program memory can be loaded by the Z8 (series of load immediate instructions). After loading the program memory, the Pseudo-PC can be loaded with the start address for program execution. Loading of the pseudo-PC will start the DSP. The DSP keeps executing until STOP DSP instruction which puts the DSP in the low power mode. As shown in the DSP instruction set, branching is allowed within the program memory space (512 bytes). The instruction execution time in this mode is one state time in the pipeline mode except for Branch and Load Immediate to the register pointers. This takes three and two state times, respectively, in the pipeline mode.

2. Another way to start the DSP execution from the internal program memory is to load the Shadow latch register with the start address and set bit 3 of the Z8/DSP control register to 1. When the A/D converter finishes conversion, it generates an interrupt to the DSP which then loads the pseudo-PC from the shadow latch and start the execution from that location. Notice that this enables a very fast LOOP execution by avoiding a Z8 interrupt wait time delay.

3. Via loading the Psuedo-IR with the appropriate instruction:

   The DSP instruction (8 bits) is loaded as a Load Immediate data value into the Pseudo-IR. The DSP then wakes up from the power-down mode, executes the instruction and goes back to the power down mode. Since a Load Immediate operation takes six clocks, the instruction execution time in this mode is six clocks. Notice, that branching in this mode can be done by examining the Status register of the DSP which is mapped to the Z8 space (Figure 17).

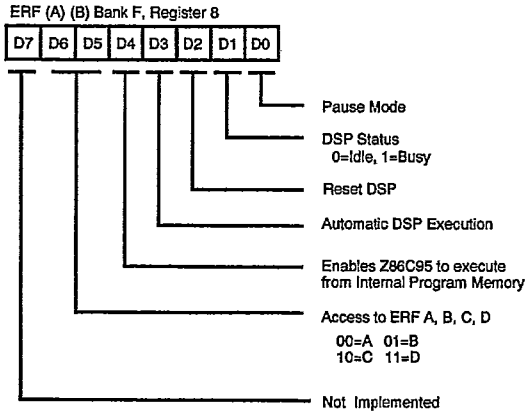


Figure 17. Z8/DSP Control Register

Bits 7 not implemented.

Bits 5, 6 Access to ERF A/B/C/D.

Bit 4 enables the Z8 to execute from internal memory (Flash EPROM Support) when set to 1 but note only the lower 256 byte of DSP program memory can be used (ERFB and ERFC up to bank 6). This bit is automatically reset to 0 on power-up.

Bit 3 enables automatic DSP execution when the A/D completes conversion (when set to 1).

Bit 2 allows reset of the DSP.

Bit 1 indicates the status of the DSP. When bit 1 is set to 1 it indicates the DSP is busy executing from internal program memory. Bit 1 is reset to 0 on power-up.

Bit 0 enables PAUSE mode when set to 1. Bit 0 is reset to 0 on power-up (see power-down mode).

FUNCTIONAL DESCRIPTION

T-49-12-09

Z8 Multiply/Divide Unit

This section describes the basic features, implementation details and the interface between the Z8 and the multiply/divide unit (Figure 18).

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 Z8.

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 the operands are in the respective registers before a multiplication/division can start.

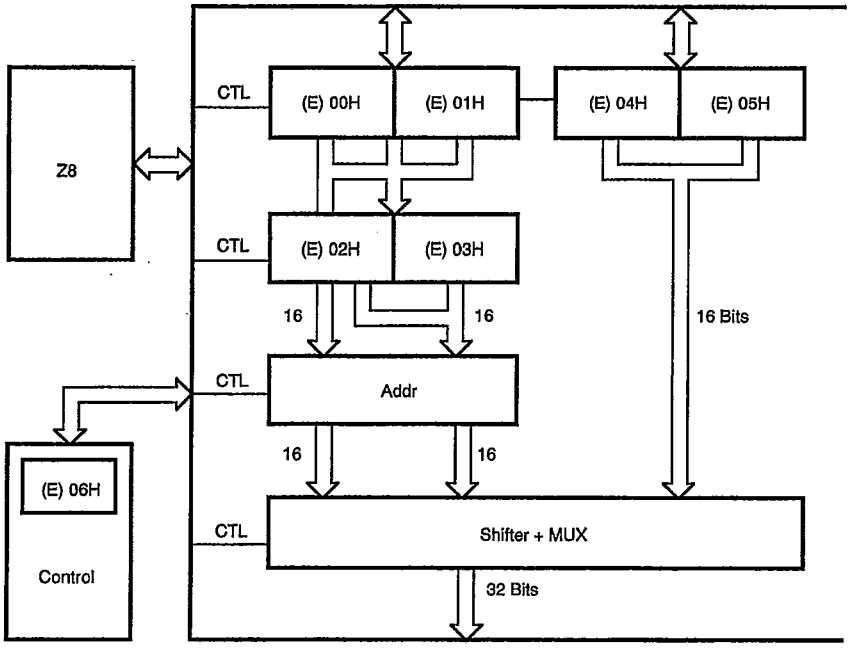


Figure 18. Z8 Multiply/Divide Unit Block Diagram

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

Register	Address
REG0	(E) 00H
REG1	(E) 01H
REG2	(E) 02H
REG3	(E) 03H
REG4	(E) 04H
REG5	(E) 05H
REG6	(E) 06H
REG7	(E) 14H
REG8	(E) 15H

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

Allocation	Register
Multiplier high byte	REG2
Multiplier low byte	REG3
Multiplicand high byte	REG4
Multiplicand low byte	REG5
Result high byte of high word	REG0
Result low byte of high word	REG1
Result high byte of low word	REG2
Result low byte of low word	REG3
Control register	REG6

The following is the register allocation during division.

T-49-12-09

Allocation	Register
High byte of high word of dividend	REG0
Low byte of high word of dividend	REG1
High byte of low word of dividend	REG2
Low byte of low word of dividend	REG3
High byte of divisor	REG4
Low byte of divisor	REG5
High byte of remainder	REG0
Low byte of remainder	REG1
High byte of quotient	REG2
Low byte of quotient	REG3
Control register	REG6

Control register. The MDCON control register is used to interface with the multiply/divide unit (Figure 19). Specific functions of various bits in the control register are shown.

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 will indicate 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 will set this bit to indicate the completion of operation. The processor/programmer can then read the result.

FUNCTIONAL DESCRIPTION (Continued)

T-49-12-09

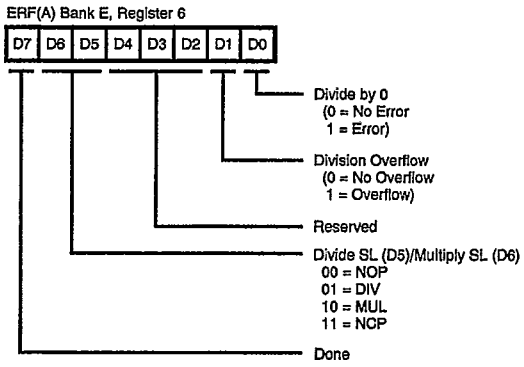


Figure 19a. Multiply/Divide Control Register (MDCON)

General Purpose Register
General Purpose Register
Compare Register 1 Low Byte
Compare Register 1 High Byte
Not Used
Not Used
Not Used
Not Used
Not Used
MUL/DIV Control Register
MUL/DIV Register 5
MUL/DIV Register 4
MUL/DIV Register 3
MUL/DIV Register 2
MUL/DIV Register 1
MUL/DIV Register 0

Figure 19b. ERF (A) Bank E

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

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

D4-D2. *Reserved.*

DIVOVF. *Division Overflow* (D1). This bit indicated 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 should be set to 010xxxxxb.

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

On completion of multiplication, the result of the MDCON register will be 100uuu00b.

If division operation is desired, the MDCON register should be 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 would contain 100uuu??b.

Note that once the multiplication/division operation starts, all data registers (REG5 thru REG0) 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 can be read 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 fixed numbers of cycles, the results can be read 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.

REG7 and REG8 can be used as scratch pad registers or as external data memory address pointers during an LDE instruction. REG0 thru REG5 and REG7 and REG8, if not used for multiplication or division, can be used as general-purpose registers.

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

T-49-12-09

number of clock cycles to load 5 registers = 30,
number of clock cycles to read 4 registers = 24

then, the total number of clock cycles 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 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 clock cycles to load operands = 42,
number of clock cycles to read results = 24.

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

FUNCTIONAL DESCRIPTION (Continued)

T-49-12-09

Counter/Timers

This section describes the enhanced features of the counter/timers (CTC) on the Z86C95 (Figure 20). 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 can be driven by internal or external clock sources; however, the T0 prescaler is driven by the internal clock only.

The 6-bit prescalers can 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.

The counters can be programmed to start, stop, restart to continue, or restart from the initial value. Also, the counters can 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, can be read at any time without disturbing their value or count mode. The clock source for T1 is user-definable and can be 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 can be retriggerable or non-retriggerable, or as a gate input for the internal clock. The counter/timers can be 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 Z86C95:

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 three capture registers and two compare registers.

These three counters are cascadable as shown in Table 4. 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 4. Z86C95 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 can be programmed to XTAL/2 or XTAL/8 which results in a resolution of 100 ns at external XTAL clock speed of 20 MHz.

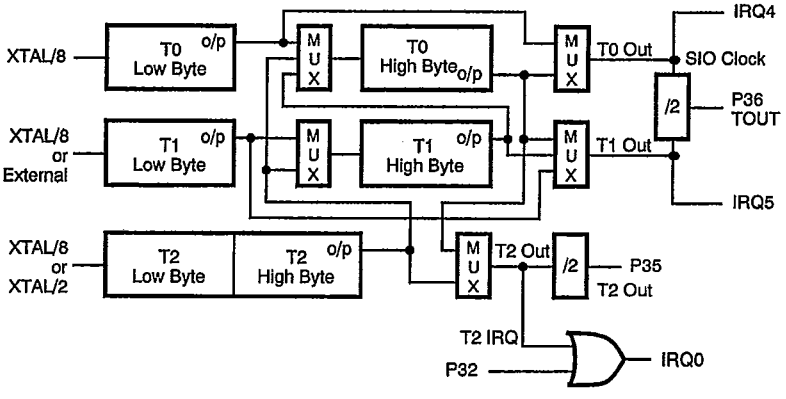


Figure 20. Counter/Timer Block Diagram

Capture and Compare

T-49-12-09

There are three capture registers associated with T2 HIGH BYTE and T2 LOW BYTE registers and two compare registers on timer T2 (Figure 21). At the falling edge of the appropriate Port 3 input, the current value of Timer 2 (T2) is "captured" into a read only register. For example, the negative going transition on pin P33 will enable the latching of the current T2 value (16-bits) into the Capture Register 1 (CAP1). The register mapping and the appropriate inputs are shown below (Table 5). Note that the negative transition on P33, P32, and P30 is capable of generating an interrupt. Also, the negative transition on Port 3 will always latch the current T2 value into the capture register. There in no need for a control bit to enable/disable the latching.

ERF (A), Bank D

DFH	Compare Register 2 Low Byte
DEH	Compare Register 2 High Byte
DDH	Capture Register 3 Low Byte
DCH	Capture Register 3 High Byte
DBH	Capture Register 2 Low Byte
DAH	Capture Register 2 High Byte
D9H	Capture Register 1 Low Byte
D8H	Capture Register 1 High Byte
D7H	Timer 2 Low Byte
D6H	Timer 2 High Byte
D5H	Capture/Compare Control Register
D4H	Timer 0 High Byte
D3H	Timer 2 Prescaler
D2H	Timer 1 High Byte
D1H	Timer 2 Mode Register

Figure 21. Capture and Compare Registers

Table 5. Capture Register Mapping

Capture Register	Port 3	Input	Addr. High	Addr. Low
CAP1	P33	Falling	(D) 8	(D) 9
CAP2	P32	Falling	(D) A	(D) B
CAP3	P30	Falling	(D) C	(D) D

FUNCTIONAL DESCRIPTION (Continued)

T-49-12-09

Compare Registers. Whenever the current value of T2 equals the contents of the compare register, some action is taken depending on the contents of the T2 Compare

Control Register (COMCON). Also, a successful comparison can generate an interrupt (if enabled) and also set a bit in the control register that can be polled at a later date (Figure 22).

COM2. *Compare 2* (D7). This bit is set to 1 when the contents of Compare Register 2 (COM2) match the current value of T2. This bit will have to be cleared by the interrupt polling routine.

Interrupt Enable 2 (D6). This bit, when set to 1, enables the interrupt for COM2.

CO2 Output (D5,D4). Controls the value outputted on CO2 according to Table 6.

COM1. *Compare 1* (D3). This bit is set to 1 when the contents of Compare Register 1 (COM1) match the current value of T2. This bit must be cleared by the interrupt polling routine.

Interrupt Enable 1 (D2). This bit when set to 1 enables the interrupt for COM1. When either D6 or D2 is set and the corresponding compare register contents match the current value of T2, an interrupt is generated on IRQ5, which is configured as an OR of T1IRQ, COM1, or COM2 interrupts.

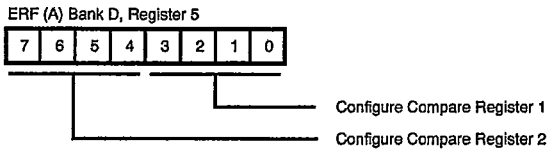


Figure 22. T2 Compare Control Register (COMCON)

CO1 Output (D1,D0). Controls the value outputted on CO1 according to the following table:

Table 6. Compare Output Status

Bit5 Bit1	Bit4 Bit0	Output on Compare Output
0	0	NOP (CO1/CO2 retain previous value)
0	1	Reset to "0"
1	0	Set to "1"
1	1	Toggle status

Table 7. Compare Register Mapping

Compare Register	Addr. High	Addr. Low
COM1	(E) C	(E) D
COM2	(D) E	(D) F

Observations:
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, can go to an output pin (P35). Also, the interrupt generated by T2 can be 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:

T0 in 8-bit mode	T0 Low byte
T0 in 16-bit mode	T0 High byte, T0 Low byte
T1 in 8-bit mode	T1 Low byte
T1 in 16-bit mode	T1 High byte, T1 Low byte
T1 in 24-bit mode	T0 High byte, T1 High byte T1 Low byte
T2 in 16-bit mode	T2 High byte, T2 Low byte
T2 in 24-bit mode	T0 High byte, T2 High byte T2 Low byte
T2 in 32-bit mode	T0 High byte, T1 High byte, T2 High byte, T2 Low byte

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

The T2 Timer Mode register is shown in Figure 23. Upon reaching end of count, bit 7 of this register is set to 1. This bit IS NOT reset in hardware and it has to be cleared by the interrupt service routine.

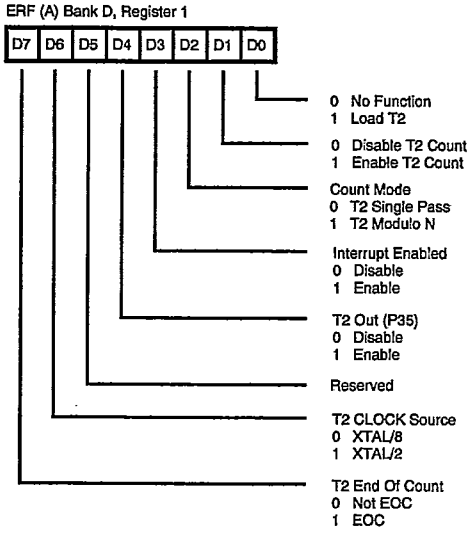


Figure 23. T2 Timer Mode Register (T2)

The register map of the new CTC registers is shown in Figure 11. T0 High byte, 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 24. Bit 1 and Bit 0 of this register controls the various cascade modes of the counters as shown in Table 1.

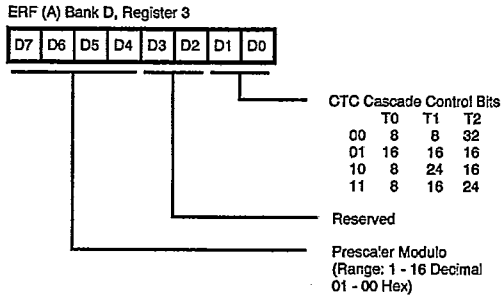


Figure 24. T2 Prescaler Register (PRE2)

Analog to Digital Converter (ADC)

T-49-12-09

The ADC is an 8-bit half flash converter which uses two reference resistor ladders for its upper 4 bits (MSBs) and lower 4 bits (LSBs) conversion. Two reference voltage pins, VA_{HI} (High) and VA_{LO} (Low), are provided for external reference voltage supplies. During the sampling period from one of the eight channel inputs, the converter is also being auto-zeroed before starting the conversion. The conversion time is dependent on the external clock frequency and the selection of the prescaler value for the internal ADC clock source. The minimum conversion time is 2.0 μsec. (See Figure 25, ADC Architecture.)

The ADC is controlled by the Z8 and its six registers (two Control and four Result) are mapped into the Extended Register File. The first Result register is also readable by the DSP. The DSP can access the ADC control register 0, and this allows the DSP to change Input Channel selections.

A conversion can be initiated in one of four ways: by writing to the Command register, from a rising or falling edge on Port 32 pin or Timer 0 equal 0. These four are programmably selectable. There are four modes of operation that can be selected: one channel converted four times with the results written to each Result register, one channel continuously converted and one Result channel updated for each conversion, four channels converted once each and the four results written to the Result registers, and four channels repeatedly converted and the Result registers kept updated. The channel to be converted is programmable and if one of the four-channel modes is selected then the programmed channel will be the first channel converted and the other three will be in sequence following with wraparound from Channel 7 to Channel 0.

The start commands are implemented in such a way as to begin a conversion at any time, if a conversion is in progress and a new start command is received, then the conversion in progress will be aborted and a new conversion will be initiated. This allows the programmed values to be changed without affecting a conversion-in-progress. The new values will take effect only after a new start command is received.

The clock prescaler can be programmed to derive a minimum 2 μsec conversion time for XTAL clock inputs from 4 MHz to 20 MHz. For example, with a 20 MHz XTAL clock the prescaler should be programmed for divide by 40 which then gives a 2 μsec conversion rate.

The ADC can generate an interrupt after either the first or fourth conversion is complete depending on the programmable selection.

The ADC can be disabled (for low power) or enabled by a Control Register bit.

Though the ADC will function for a smaller input voltage and voltage reference, the noise and offsets remain constant over the specified electrical range. The errors of the converter will increase and the conversion time may also take slightly longer due to smaller input signals.

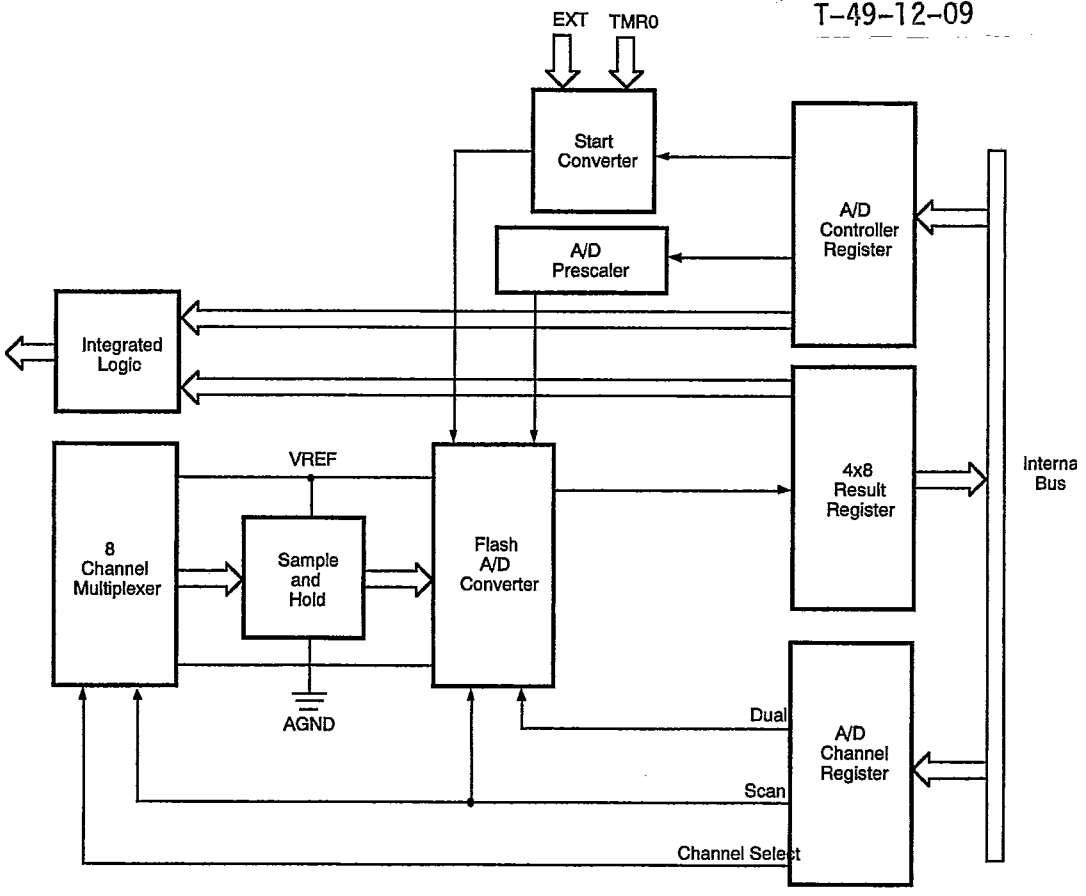


Figure 25. ADC Architecure

FUNCTIONAL DESCRIPTION (Continued)

T-49-12-09

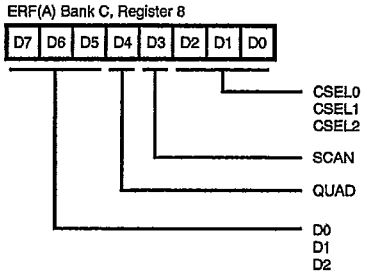


Figure 26. ADC Control Register 0

Prescaler Values (bits 7, 6, 5).

D2	D1	D0	Prescaler (XTAL divided by)
0	0	0	8
0	0	1	16
0	1	0	24
0	1	1	32
1	0	0	40
1	0	1	48
1	1	0	56

Note:
The ADC is being characterized as of this date. The errors of the converter are estimated to increase to 2LSBs (Integral non-linearity), 1 LSB (Differential non-linearity) and 10mV (Zero error at 25°C) if the voltage swing on the reference ladder is decreased to -3V.

Modes (bits 4, 3).

QUAD	SCAN	
0	0	Convert selected channel 4 times then stop
0	1	Convert selected channel then stop
1	0	Convert 4 channels then stop
1	1	Convert 4 channels continuously

Channel Select (bits 2, 1, 0).

CSEL2	CSEL1	CSEL0	Channel
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

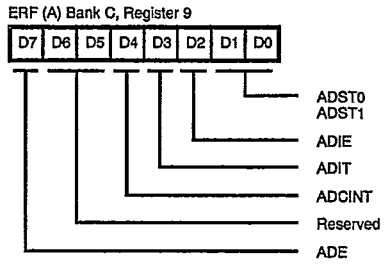


Figure 27. ADC Control Register 1

ADE (bit 7). A0 disables any A-D conversions or accessing any ADC registers except writing to ADE bit. A1 Enables all ADC accesses.

Reserved (bits 6, 5). Reserved for future use.

ADCINT (bit 4). This is the ADC Interrupt bit and is read only by the Z8, the ADCINT will be reset any time this register is written.

ADIT (bit 3). This bit selects when to set the ADC Interrupt if ADIE=1. A 0 sets the Interrupt after the first A-D conversion is complete. A 1 sets the Interrupt after the fourth A-D conversion is complete.

ADIE (bit 2). This is the ADC Interrupt Enable. A 0 disables setting the ADC Interrupt. A 1 enables setting the ADC Interrupt.

T-49-12-09

START (bits 1, 0).

ADST1	ADST0	Mode
0	0	Conversion starts when this register is written.
0	1	Conversion starts on a rising edge at Port 3-2.
1	0	Conversion starts on a falling edge at Port 3-2.
1	1	Conversion starts when Timer0 times out.

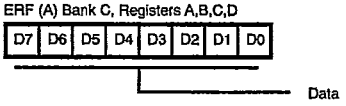
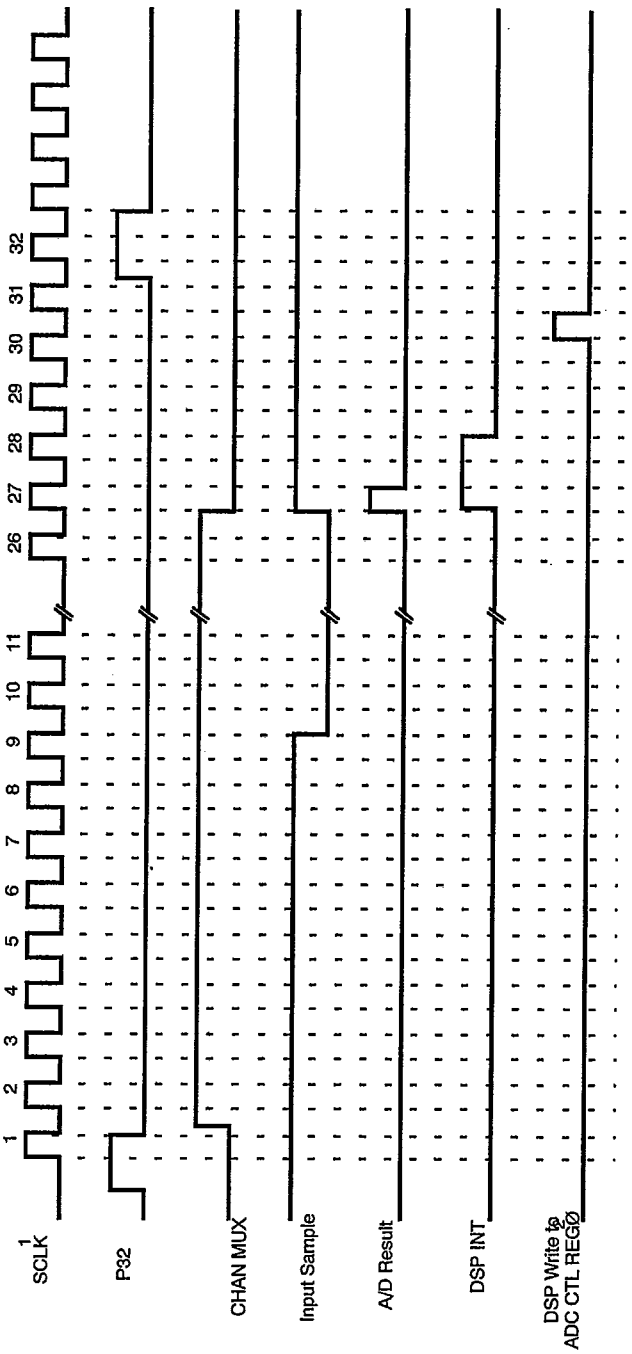


Figure 28. Result Registers

These are the four ADC result registers, Reg-A holds the first result and Reg-D the fourth result. These registers are R/W by the Z8 (Writable for test purposes) and Reg-A is Read Only by the DSP and is mapped to Reg 1 for the DSP. Figure 29 shows the timing diagram for the ADC.

FUNCTIONAL DESCRIPTION (Continued)

T-49-12-09



Notes:
1. SCLK = 12 MHz (XTAL = 24 MHz)
2. ADC CTL REGØ = 85
   ADC CTL REG1 = 85

Figure 29. ADC Timing Diagram

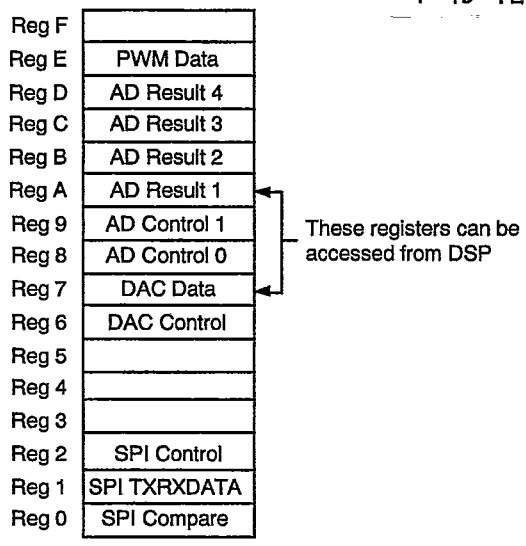


Figure 30. ERF(A) Bank C

Figure 31 shows the input circuit of the ADC. When conversion starts, the analog input voltage from one of the eight channel inputs is connected to the MSB and LSB flash converter inputs as shown in the Input Impedance CKT diagram. Effectively shunting 31 parallel internal resistance of the analog switches and simultaneously charging 31 parallel 0.5 pF capacitors, which is equivalent to seeing a 400 Ohms input impedance in parallel with a 16 pF capacitor. Other input stray capacitance adds about 10 pF to the input load. For input source resistances up to 2 kOhms can be used under normal operating condition without any degradation of the input settling time. For larger input source resistance, longer conversion cycle time may be required to compensate the input settling time problem.

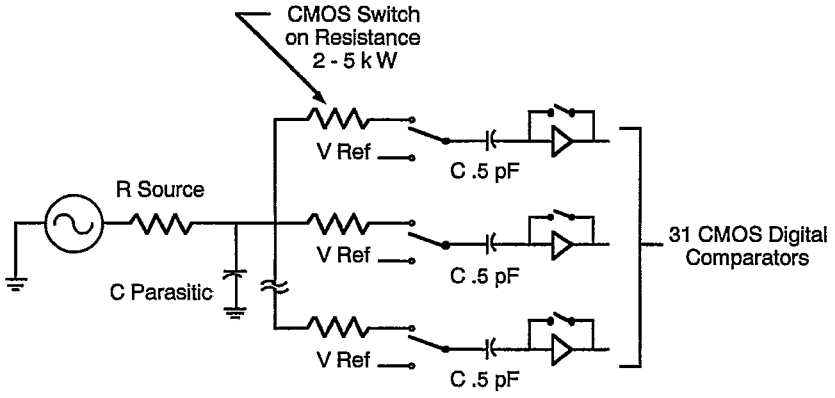


Figure 31. Input Impedance of ADC

Digital to Analog Converter (DAC)

T-49-12-09

The DAC (Digital to Analog Converter) is an 8-bit resistor string, with a programmable 0.25X and 0.5X gain output buffer. The DAC output voltage is settled after the internal digital data is latched. Two pins are provided externally for the DAC reference voltage supplies, VD_{HI} and VD_{LO}, these should not exceed the supply voltages. The DAC output is latch-up protected and can drive output loads (Figure 31).

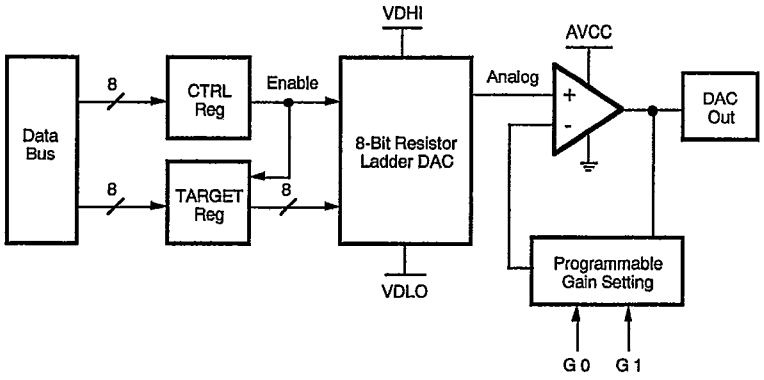


Figure 32. DAC Block Diagram

The DAC is controlled by the Z8. Its two registers (Control 1 and Data 1) are mapped into the ERF (Figures 33 and 34). The Data 1 register is writeable by the DSP.

The DAC can be enabled or disabled by programming the Control 1 register or it can be programmed to output an analog voltage when the Data 1 register is loaded. The Control 1 register is used to program for the Gain factor of the DAC output.

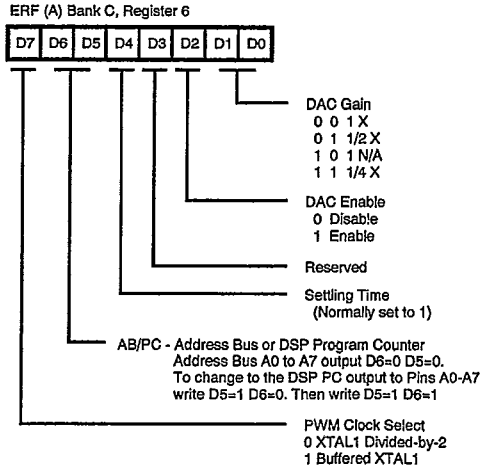


Figure 33. DAC Control Register

The DAC Data Register is initialized to 80H on power-up (Figure 34). Also the DAC gain control pivots about a mid-point rather than ground. (Figure 34). When the gain control is at 1.0X or 0.5X or 0.25X the DAC output remains constant when the DAC data register equals 80H.

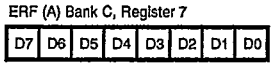


Figure 34. DAC Data Register

T-49-12-09

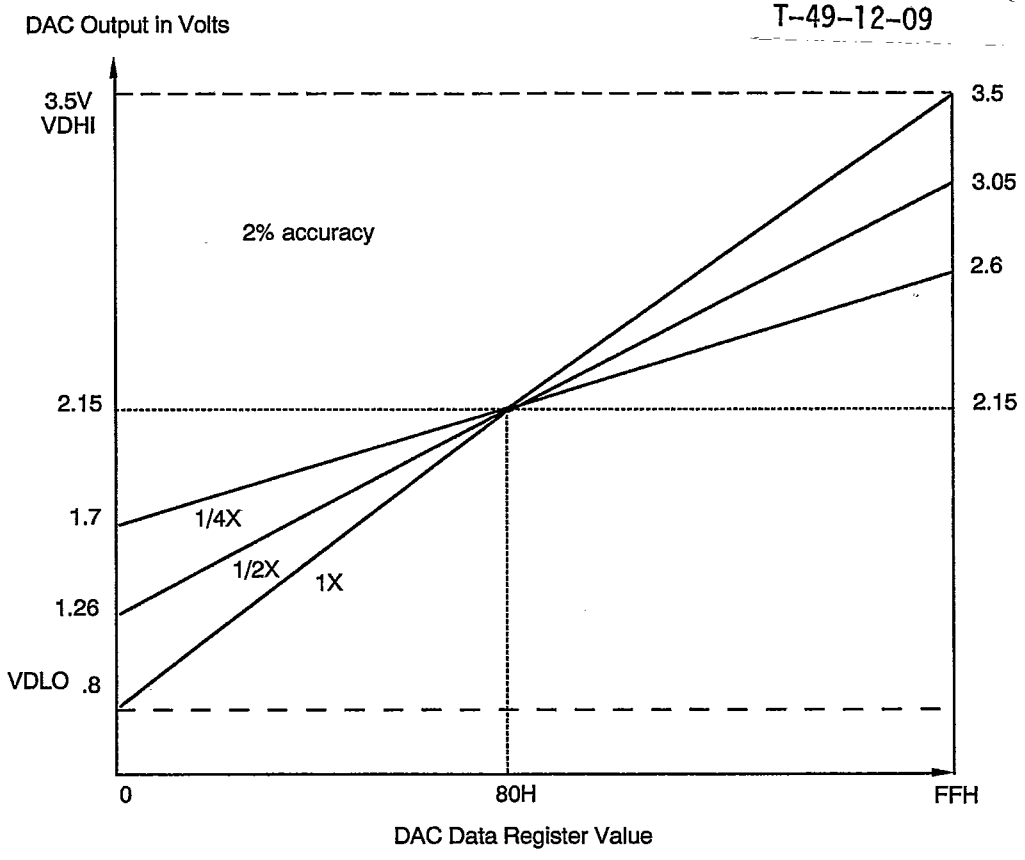


Figure 35. Gain Control on DAC

FUNCTIONAL DESCRIPTION (Continued)

T-49-12-09

Serial Peripheral Interface

Serial Peripheral Interface (SPI). The Z86C95 incorporates a serial peripheral interface for communication with other microcontrollers and peripherals. The SPI includes features such as Master/Slave selection and Compare mode. The SPI consists of four registers; SPI Control Register (SCON), SPI Compare Register (SCOMP), SPI Receive/Buffer Register (RxBUF), and SPI Shift Register. SCON is located in bank C of the Expanded Register Group at Address 02 (Figure 36). This register is a read/write register that controls; Master/Slave selection, interrupts, clock source and phase selection, and error flag. Bit 0 enables/disables the SPI with the default being SPI disabled. A 1 in this location enables the SPI, and a 0 disables the SPI.

Bits 1 and 2 of the SCON register in Master Mode selects the clock rate. The user may choose whether internal clock is divide by 2, 4, 8 or 16. In Slave Mode, Bit 1 of this register flags the user if an overrun of the RxBUF Register has occurred.

The RxCharOverrun flag can only be reset by writing a 0 to this bit. In slave mode, bit 2 of the Control Register can disable the data-out I/O function. If a 1 is written to this bit, the data-out pin is tristated. If a 0 is written to this bit, the SPI will shift out one bit for each bit received. Bit 3 of the SCON Register enables the interrupt of the SPI, with the default being disabled. Bit 4 signals that a receive character is available in the RxBUF Register. If the associated IRQ0 is enabled, an interrupt is generated. Bit 5 controls the clock phase of the SPI. A 1 in Bit 5 allows for receiving data on the clock's falling edge and transmitting data on the clock's rising edge. A 0 allows receiving data on the clock's rising edge and transmitting on the clock's falling edge.

The SPI clock source is defined in bit 6 for Master mode. A 1 uses Timer0 output for the SPI clock, and a 0 uses TCLK for clocking the SPI. In Slave mode, bit 6 will enable or disable the address compare feature. Finally, bit 7 determines whether the SPI is used as a Master or a Slave. A 1 puts the SPI into Master mode and a 0 puts the SPI into Slave mode.

SPI Operation. The SPI can be used in one of two modes; either as system slave, or a system master. In the slave mode, data transfer starts when the slave select (SLAVESEL) pin goes active. Data is transferred into the slave's SPI Shift Register, through the DI pin, which has the same address as the RxBUF Register. After a byte of data has been received by the SPI Shift Register a Receive Character Available (RCA/IRQ0) interrupt and flag is generated. The next byte of data may be received at this time, but the RxBUF Register must be cleared, or a Receive Character Overrun (RxCharOverrun) flag is set in the SCON Register and the data in the RxBUF Register is overwritten.

T-49-12-09

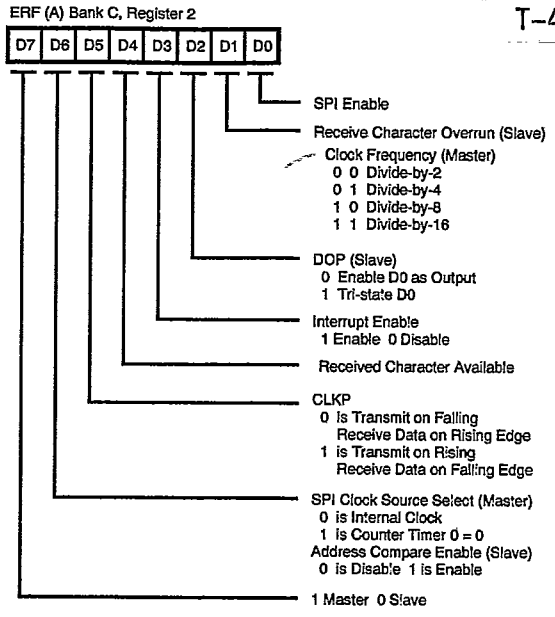


Figure 36. SPI Control Register (SCON)

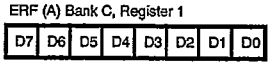


Figure 37. SPI TXRXDATA Register

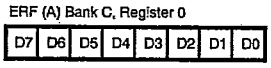


Figure 38. SPI Compare Register

Serial Peripheral Interface (Continued)

T-49-12-09

When the communication between the master and slave is complete, the SS goes inactive. Unless disconnected, for every bit that is transferred into the slave through the DI pin, a bit is transferred out through the DO pin on the opposite clock edge. During slave operation, the SPI clock pin (SK) is an input. In master mode, the CPU must first activate a SS through one of it's I/O ports. Next, data is transferred through the master's DO pin one bit per master clock cycle. Loading data into the shift register initiates the transfer. In master mode, the master's clock drives the slave's clock. At the conclusion of a transfer, a Receive Character Available (RCA/IRQ0) interrupt and flag is generated. Before data is transferred via the DO pin, the SPI Enable bit in the SCON Register must be enabled.

SPI Compare. When the SPI Compare Enable bit, D6 of the SCON Register is set to 1, the SPI Compare feature is enabled. The compare feature is only valid for slave mode. A compare transaction begins when the SS line goes active. Data is received as if it were a normal transaction, but there is no data transmitted to avoid bus contention with other slave devices. When the compare byte is received, IRQ0 is not generated. Instead, the data is compared with the contents of the SCOMP Register. If the data does not match, DO will remain inactive and the slave will ignore all data until the SS signal is reset.

SPI Clock. The SPI clock can be driven from three sources; with Timer0, a division of the internal system clock, or an external master when in slave mode. Bit D6 of the SCON Register controls what source drives the SPI clock. Divide-by-2, 4, 8 or 16 can be chosen as the scaler with bits D2, D1 in master mode.

Receive Character Available and Overrun. When a complete data stream is received an interrupt is generated and the RxCharAvail bit in the SCON Register is set. Bit 4 in the SCON Register is for enabling or disabling the RxCharAvail interrupt. The RxCharAvail bit is available for interrupt polling purposes and is reset when the RxBUF Register is read. RxCharAvail is generated in both master and slave modes. While in slave mode, if the RxBUF is not read before the next data stream is received and loaded into the RxBUF Register, Receive Character Overrun (RxCharOverrun) occurs. Since there is no need for clock control in slave mode, bit D1 in the SPI Control Register is used to log any RxCharOverrun.

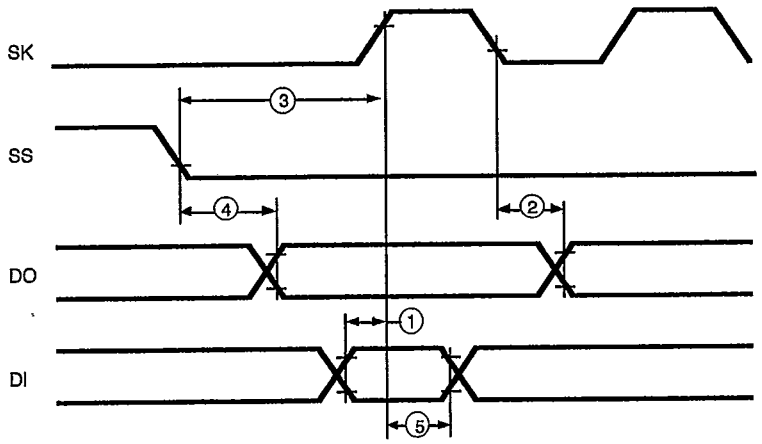


Figure 39. SPI Timing

Interrupts

T-49-12-09

The Z86C95 has six different interrupts from ten different sources. The interrupts are maskable and prioritized. The eight sources are divided as follow: four sources are claimed by Port 3 lines P30-P33, one is Serial Out, one is Serial In, and two 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 Z86C95 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, and 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 interrupts on the Z86C95.

Table 8. Z86C95 Interrupts

Name	Source	Vector	Comments
IRQ0	P3.2, SPI, A/D Start	0,1	Falling Edge Triggered
IRQ1	P3.3, A/D Finish	2,3	Falling Edge Triggered
IRQ2	P3.2, T_{IN}	4,5	Falling Edge Triggered
IRQ3	P3.0, Serial In	6,7	Falling Edge Triggered
IRQ4	T0, Serial Out	8,9	Internal
IRQ5	T1	10,11	Internal

FUNCTIONAL DESCRIPTION (Continued)

T-49-12-09

Clock

The Z86C95 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 crystal should be AT cut, 1 MHz to 24 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 40).

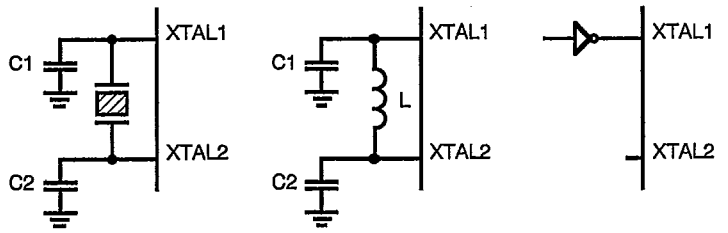


Figure 40. Oscillator Configuration

Power Down Modes

HALT. Will turn off the internal CPU clock but not the XTAL oscillation. The counter/timers and the external interrupts IRQ0, IRQ1, IRQ2 and IRQ3 remains active. The devices may be recovered by interrupts, either externally or internally generated.

STOP. This instruction turns off the internal clock and external crystal oscillation and reduces the standby current to 10 µA or less. The STOP mode is terminated by a /RESET, which causes the processor to restart the application program at address 000CH.

In order to enter STOP (or HALT) mode, it is necessary to first flush the instruction pipeline to avoid suspending execution in mid-instruction. To do this, the user must execute a NOP (opcode=OFFH) immediately before the appropriate sleep instruction. i.e.:

```
FF    NOP    ; clear the pipeline
6F    STOP   ; enter STOP mode
               or
FF    NOP    ; clear the pipeline
7F    HALT   ; enter HALT mode
```

PAUSE. This is similar to the STOP mode, except in the recovery method, and the fact that the program counter simply continues from where it paused instead of resetting to 000CH. PAUSE mode is entered by setting bit 0 of DSP control register to 1 and executing a HALT instruction. All the internal clocks are stopped during the PAUSE mode thus resulting in very low power. To recover from the PAUSE mode, the Z86C95 needs to see a negative going transition on Port 32. This generates an interrupt and operation can resume by simply doing an IRET in the interrupt execution routine. The recovery time from PAUSE mode is equal to the XTAL oscillator stabilization time +1.3 msec (XTAL frequency of 20 MHz).

Pulse Width Modulator (PWM)

This block provides a Pulse Width Modulated output at a constant period based on the input clock.

The PWM provides an output waveform whose period is either the internal system clock or the buffered XTAL input divided by 256. The duty cycle of this waveform is programmable by a register in the Extended Register File of the Z8 and can have values from 0 to 99.6% (Reg = 0 to 255). A programmed value of 0 will disable the counter and place the PWM in a low power mode. Any non-zero value programmed in this register will enable the PWM divider and generate the selected output waveform.

The clock source for the PWM is programmable providing the user access to a higher frequency clock versus using the internal clock. The clock source is selected using Bit 7 of the DAC control register (ERF(A) Bank C, Register 6). D7=1 selects a buffered XTAL1 clock, D7=0 selects XTAL divided-by-2 (Figure 33).

T-49-12-09

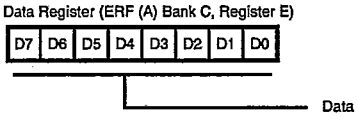


Figure 41. Pulse Width Modulation Register Assignment

The PWM register is used to program the duty cycle of the PWM. If the programmed value is 0, then the PWM is disabled and the PWM output is OFF. For any non-zero value the PWM output is a periodic waveform which is High for (value/256)X100 % of the period.

The DSP slave processor is a 16-bit fixed point, two's complement high-speed digital signal processor. The basic concept behind the DSP megacell is to simplify the architecture and instructions as much as possible, providing a user-friendly programming environment for various DSP algorithms. Additionally, a convenient mapping architecture was designed to allow the Z8 to map the DSP memory into the shared expanded register file architecture of the Z8.

The Z86C95's DSP has two sets of high-speed on-chip RAMs for data storage. The RAM data specified by two different RAM address registers or instruction address fields are read out in one machine cycle. Multiplication, addition and register loading can be accomplished in one clock cycle. The instructions are one cycle pipelined, which are transparent to the users.

Architectural Overview

The Z86C95's DSP employs a 16-bit fixed point, two's complement number system. The binary point is assumed to be placed right next to the sign bit. DSP algorithms are accomplished by single-cycle multiply/accumulate instructions, two on-chip RAM banks, dedicated arithmetic logic unit, user-definable I/O for signal processing and other functions. (See DSP Commands Section below.)

Cycles Per Instruction

Most instructions are one machine cycle instructions which are executed in one cycle time. Load register pointer immediate and Branch instructions need two machine cycles to execute. Besides these execution machine cycles, one more cycle is required if the PC (Program Counter) is selected as the destination of a data transfer instruction. This happens when register indirect branch is executed. An $a_1*b_1+ACC \rightarrow ACC$ calculation is done in one machine clock cycle modifying the RAM pointer contents. Both a_1 and b_1 can be RAM contents located in two independent addresses. Since each instruction is fetched into the instruction register one cycle earlier and the pre-fetched instruction is decoded at the next machine cycle, one additional machine cycle is required to modify the PC content. For instance, consider the example of a simple branch instruction, BRA NZ, 135. At $t=T_n$, the pre-fetched content of the pseudo instruction register, BRA NZ, 135, starts to decode and execute while the pseudo PC is automatically increased to 105. Since the instruction is to change the PC to 135 if the condition is NZ, the next fetched instruction would be treated as a NOP.

Indirect Addressing Mode

Register Indirect addressing is the method of addressing within the Z86C95's DSP. This is accomplished by means of four register pointers, two for each bank. These pointers are R0 and R1 for DSP RAM(0) and R2 and R3 for DSP RAM (1). The register pointers are located within the Z8/DSP interface register bank (Bank F in both ERF (A) and ERF (B)). For example, LDI R0, 14 will load 14 into the register pointer R0. If followed by an instruction LD (R0) for example the contents of the register in DSP RAM0 whose address is 14 will be loaded into the accumulator.

Arithmetic Logic Unit

Upon loading the DSP data RAM the Z86C95's DSP can multiply two 16-bit integers and accumulate a 24-bit result in one clock cycle. For example, an MPYA (R0), (R1) will load the contents of the DSP RAM(0) registers pointed to by R0 and R1, respectively, into the multiplier, multiply the RAM(0) registers and add the result to the accumulator. The result of the multiplication is available at the next machine cycle.

DSP Single Step Timing

The occurrence of pre-selected DSP_PC stop address and DSP_SYNC needs to be detected. DSP_SSN is pulled high which stops the DSP until DSP_SSN is pulled low at which time the DSP will execute until the next time DSP_SSN is high. DSP_SSN should not be pulled high for 2nd or 3rd bytes of multi-byte instructions, only for 1st byte of multi-byte instructions or for single byte instructions (Figure 42).

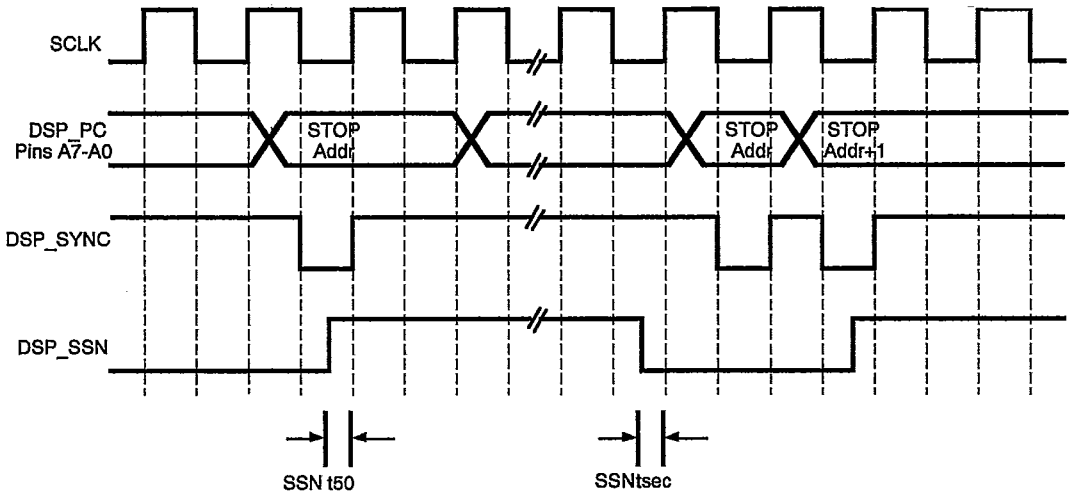


Figure 42. DSP Single Step Timing

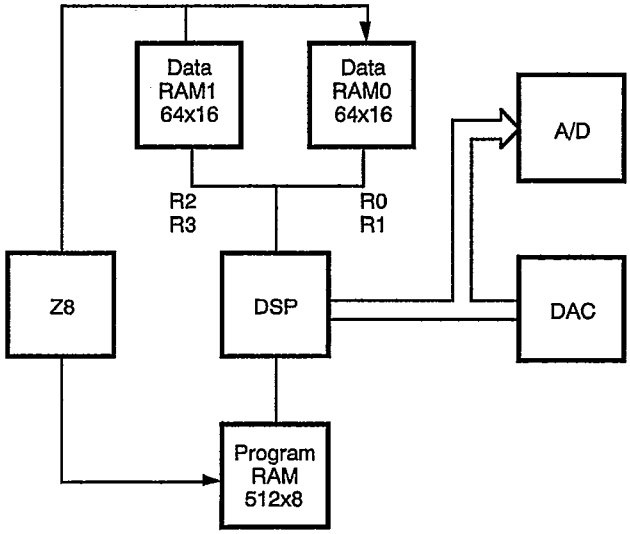


Figure 43. Block Diagram of DSP

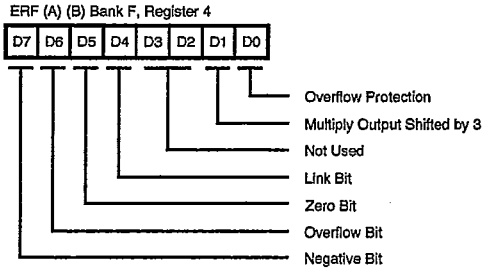


Figure 44. DSP Status Register 1

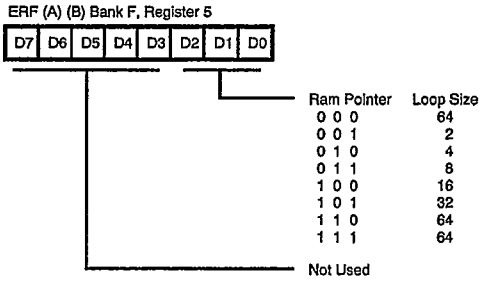


Figure 45. DSP Status Register 0

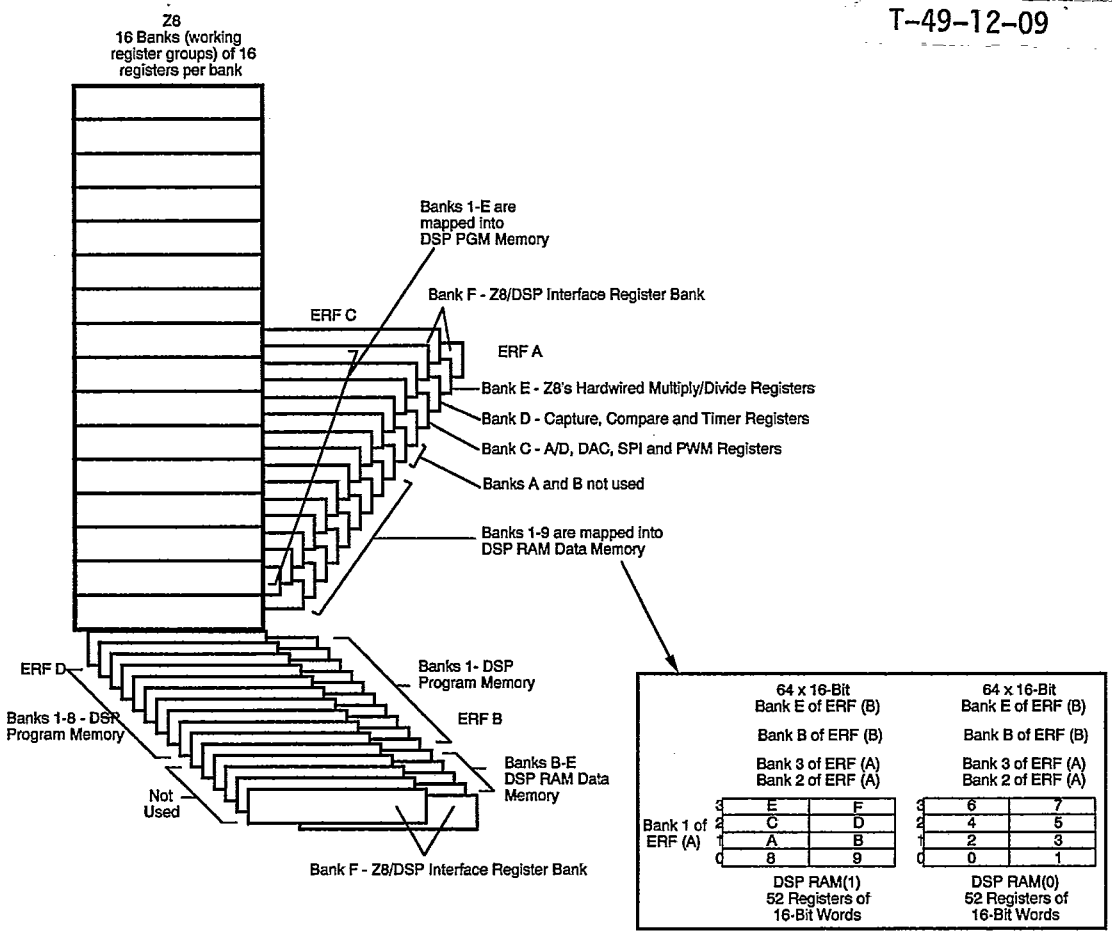


Figure 46. Z86C95 Memory Architecture

ABSOLUTE MAXIMUM RATINGS

T-49-12-09

Symbol	Description	Min	Max	Unit
V_{DD}	Supply Voltage*	-0.3	+7.0	V
T_{STG}	Storage Temp	-65	+150	C
T_A	Oper Ambient Temp	†	†	C

* Voltages on all pins with respect to GND.
† See Ordering Information

Stress greater than those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; operation of the device at any condition above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for an extended period may affect device reliability.

STANDARD TEST CONDITIONS

The characteristics listed below apply for standard test conditions as noted. All voltages are referenced to V_{SS}. Positive current flows into the referenced pin (Figure 47).

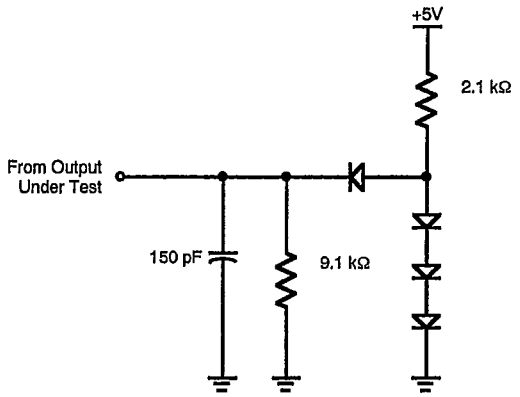


Figure 47. Test Load Diagram

DC ELECTRICAL CHARACTERISTICS

T-49-12-09

Sym	Parameter	T_A = 0°C to +70°C Min	Max	T_A = -40°C to +105°C Min	Max	Typical at 25°C	Units	Conditions
	Max Input Voltage		7		7		V	I_{CI} 250 µA
V_{CH}	Clock Input High Voltage	3.8	V_{CC}	3.8	V_{CC}		V	Driven by External Clock Generator
V_{CL}	Clock Input Low Voltage	-0.03	0.8	-0.03	0.8		V	Driven by External Clock Generator
V_{IH}	Input High Voltage	2.0	V_{CC}	2.0	V_{CC}		V	
V_{IL}	Input Low Voltage	-0.3	0.8	-0.3	0.8		V	
V_{OH}	Output High Voltge	2.4		2.4			V	I_{OH} = -2.0 mA
V_{OH}	Output High Voltge	V_{CC} - 100mV		V_{CC} - 100mV			V	I_{OH} = -100 µA
V_{OL}	Output Low Voltage		0.4		0.4		V	I_{OH} = +2.0 mA
V_{RH}	Reset Input High Voltage	3.8	V_{CC}	3.8	V_{CC}		V	
V_{RI}	Reset Input Low Voltage	-0.03	0.8	-0.03	0.8		V	
I_{IL}	Input Leakage	-2	2	-2	2		µA	Test at 0V, V_{CC}
I_{OL}	Output Leakage	-2	2	-2	2		µA	Test at 0V, V_{CC}
I_{IR}	Reset Input Current		-80		-80		µA	V_{RL} = 0V
I_{CC}	Supply Current		82		82	50	mA	@ 24 MHz [1]
			65		65	40	mA	@ 20 MHz [1]
I_{CC1}	HALT Mode		16		16	10	mA	HALT Mode V_{IN}=0V, V_{CC} @ 20 MHz [1]
			20		20	13	mA	HALT Mode V_{IN}=0V, V_{CC} @ 24 MHz [1]
I_{CC2}	PAUSE and STOP Mode		20		20	6	µA	STOP Mode V_{IN}=0V, V_{CC} [1]
I_{ALL}	Auto Latch Low Current	-10	10	-14	14	5	µA	

Note:
[1] All inputs driven to OV, VCC and outputs floating.

AC CHARACTERISTICS

T-49-12-09

External I/O or Memory Read and Write Timing Diagram

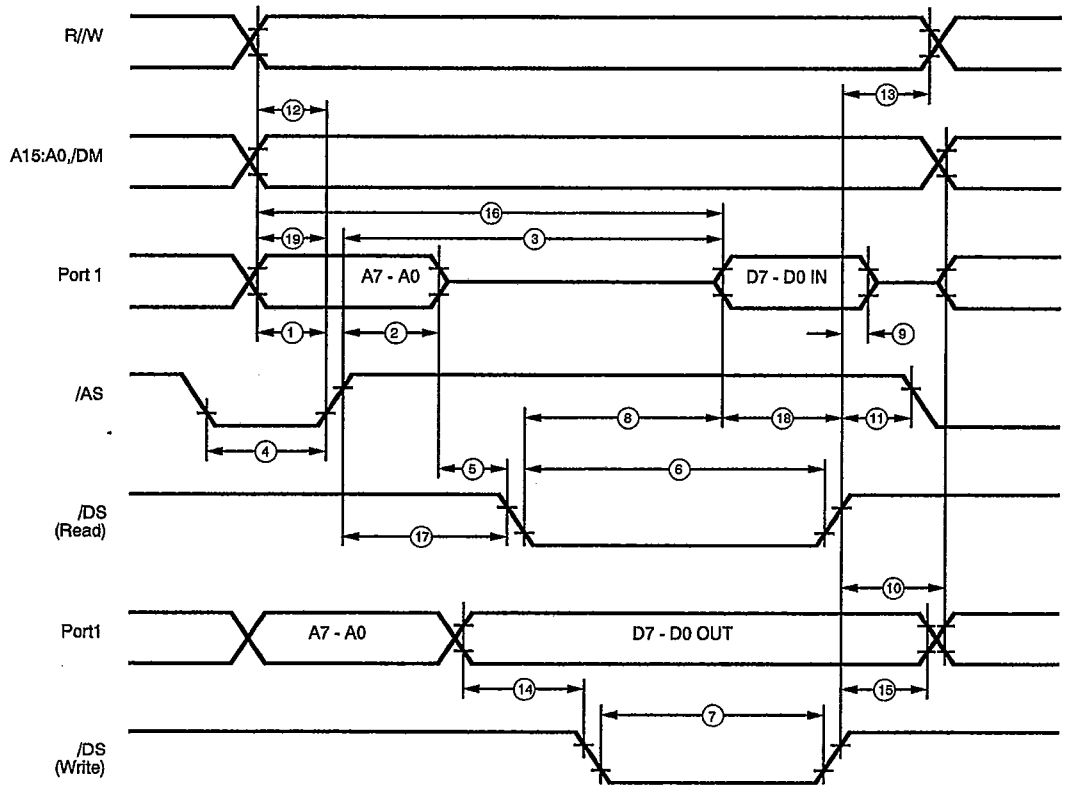


Figure 48. External I/O or Memory Read/Write

AC CHARACTERISTICS

T-49-12-09

External I/O or Memory Read and Write Timing Table

No	Symbol	Parameter	T_A = 0°C to +70°C				T_A = -40°C to +105°C				Units	Notes
			24 MHz		20 MHz		24 MHz		20 MHz			
			Min	Max	Min	Max	Min	Max	Min	Max		
1	TdA(AS)	Address Valid To /AS Rise Delay	17		20		17		20		ns	[2,3]
2	TdAS(A)	/AS Rise To Address Float Delay	25		25		25		25		ns	[2,3]
3	TdAS(DR)	/AS Rise To Read Data Req'd Valid		126		150		126		150	ns	[1,2,3]
4	TwAS	/AS Low Width	26		30		26		30		ns	[2,3]
5	TdAZ(DS)	Address Float To /DS Fall	0		0		0		0		ns	
6	TwDSR	/DS (Read) Low Width	87		105		87		105		ns	[1,2,3]
7	TwDSW	/DS (Write) Low Width	55		65		55		65		ns	[1,2,3]
8	TdDSR(DR)	/DS Fall To Read Data Req'd Valid		40		55		40		55	ns	[1,2,3]
9	ThDR(DS)	Read Data To /DS Rise Hold Time	0		0		0		0		ns	[2,3]
10	TdDS(A)	/DS Rise To Address Active Delay	34		40		34		40		ns	[2,3]
11	TdDS(AS)	/DS Rise To /AS Fall Delay	22		25		22		25		ns	[2,3]
12	TdR/W(AS)	R/W Valid To /AS Rise Delay	17		20		17		20		ns	[2,3]
13	TdDS(R/W)	/DS Rise To R/W Not Valid	19		25		19		25		ns	[2,3]
14	TdDW(DSW)	Write Data Valid To /DS Fall (Write) Delay	17		20		17		20		ns	[2,3]
15	TdDS(DW)	/DS Rise To Write Data Not Valid Delay	18		25		18		25		ns	[2,3]
16	TdA(DR)	Address Valid To Read Data Req'd Valid		148		180		148		180	ns	[1,2,3]
17	TdAS(DS)	/AS Rise To /DS Fall Delay	28		35		28		35		ns	[2,3]
18	TdDI(DS)	Data Input Setup To /DS Rise	24		50		24		50		ns	[1,2,3]
19	TdDM(AS)	/DM Valid To /AS Rise Delay	12		20		12		20		ns	[2,3]

Notes:

[1] When using extended memory timing add 2 TpC.

[2] Timing numbers given are for minimum TpC.

[3] See clock cycle dependent characteristics Table 5.

Standard Test Load

All timing references use 2.0V for a logic 1 and 0.8V for a logic 0.

Clock Dependent Equations

No	Symbol	Equation	Units
1	TdA(AS)	0.40TpC + 0.32	ns
2	TdAS(A)	0.59TpC - 3.25	ns
3	TdAS(DR)	2.83TpC + 6.14	ns
4	TwAS	0.66TpC - 1.65	ns
6	TwDSR	2.33TpC - 10.56	ns
7	TwDSW	1.27TpC + 1.67	ns
8	TdDSR(DR)	1.97TpC - 42.5	ns
10	TdDS(A)	0.8TpC	ns
11	TdDS(AS)	0.59TpC - 3.14	ns

No	Symbol	Equation	Units
12	TdR/W(AS)	0.4TpC	ns
13	TdDS(R/W)	0.8TpC - 15	ns
14	TdDW(DSW)	0.4TpC	ns
15	TdDS(DW)	0.88TpC - 19	ns
16	TdA(DR)	4TpC - 20	ns
17	TdAS(DS)	0.91TpC - 10.7	ns
18	TsDI(DS)	0.8TpC - 10	ns
19	TdDM(AS)	0.9TpC - 26.3	ns

AC CHARACTERISTICS
Additional Timing Diagram

T-49-12-09

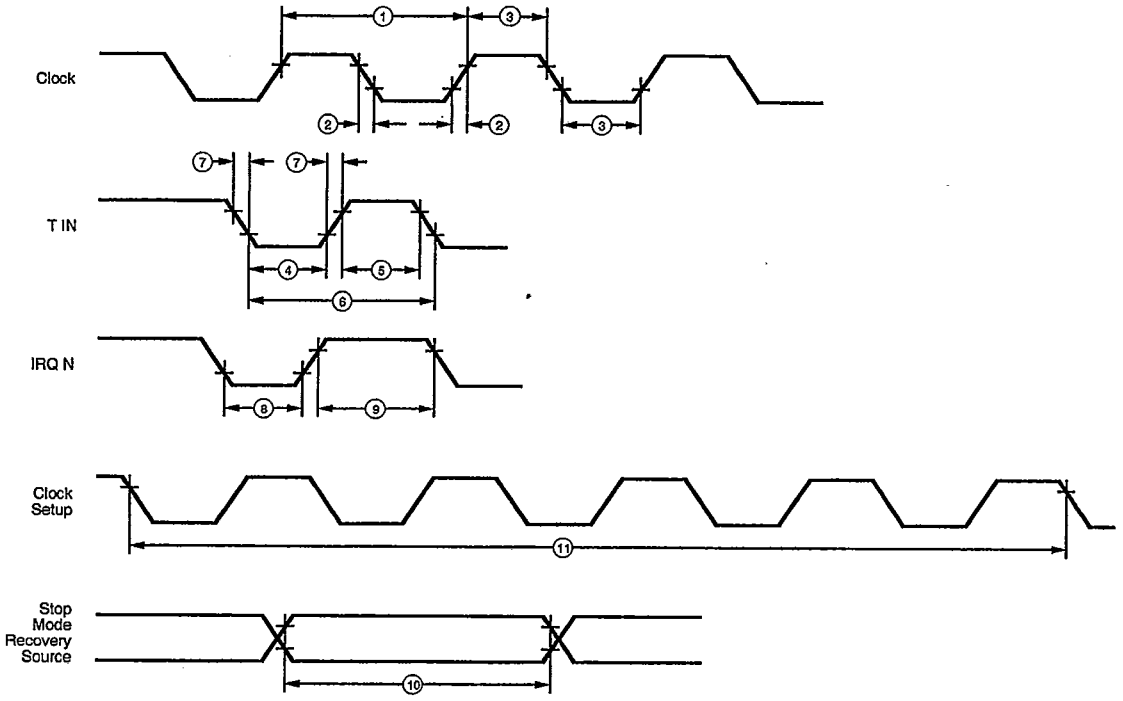


Figure 49. Additional Timing

AC CHARACTERISTICS
Additional Timing Table

T-49-12-09

No	Symbol	Parameter	T_A = 0°C to +70°C				T_A = -40°C to +105°C				Units	Notes
			24 MHz		20 MHz		24 MHz		20 MHz			
			Min	Max	Min	Max	Min	Max	Min	Max		
1	TpC	Input Clock Period	42	1000	50	1000	42	1000	50	1000	ns	[1]
2	TrC,TfC	Clock Imput Rise & Fall Times		10		10		10		10	ns	[1]
3	TwC	Input Clock Width	11		15		11		15		ns	[1]
4	TwTinL	Timer Input Low Width	75		75		75		75		ns	[2]
5	TwTinH	Timer Input High Width	3TpC		3TpC		3TpC		3TpC			[2]
6	TpTin	Timer Input Period	8TpC		8TpC		8TpC		8TpC			[2]
7	TrTin,TfTin	Timer Input Rise & Fall Times	100		100		100		100		ns	[2]
8a	TwIL	Interrupt Request Input Low Times	70		70		70		70		ns	[2,4]
8b	TwIL	Interrupt Request Input Low Times	5TpC		5TpC		5TpC		5TpC			[2,5]
9	TwIH	Interrupt Request Input High Times	3TpC		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 Diagrams

T-49-12-09

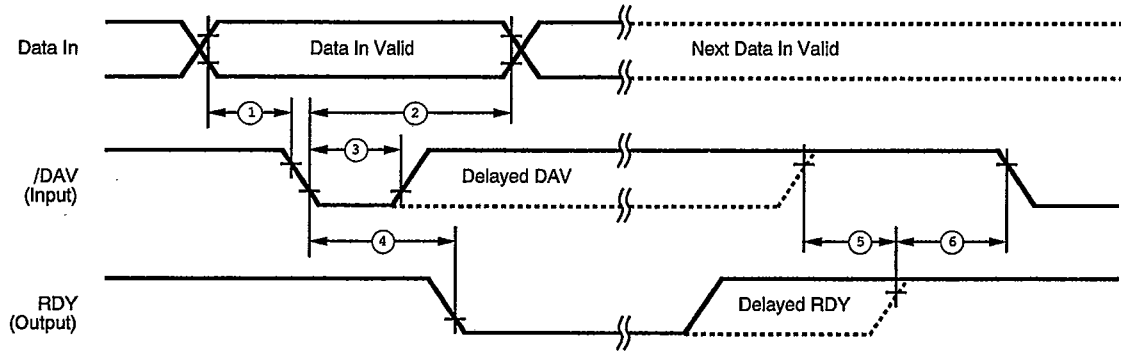


Figure 50. Input Handshake Timing

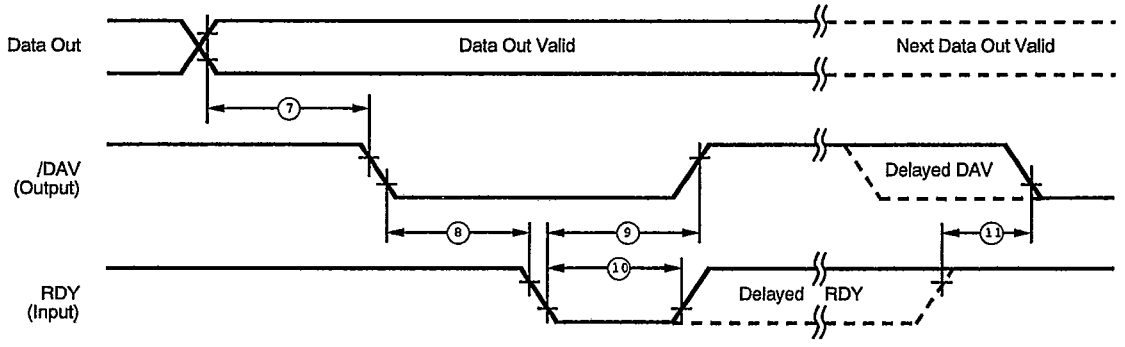


Figure 51. Output Handshake Timing

AC CHARACTERISTICS
Handshake Timing Table

T-49-12-09

No	Symbol	Parameter	T_A = 0°C to +70°C				T_A = -40°C to +105°C				Units	Notes
			24 MHz		20 MHz		24 MHz		20 MHz			
			Min	Max	Min	Max	Min	Max	Min	Max		
1	TsDI(DAV)	Data In Setup Time	0		0		0		0		ns	IN
2	ThDI(DAV)	Data In Hold Time	145		145		145		145		ns	IN
3	TwDAV	Data Available Width	110		110		110		110		ns	IN
4	TdDAVlf(RDYf)	DAV Fall to RDY Fall Delay		115		115		115		115	ns	IN
5	TdDAVlr(RDYr)	DAV Rise to RDY Rise Delay		115		115		115		115	ns	IN
6	TdRDYOr(DAVlf)	RDY Rise to DAV Fall Delay	0		0		0		0		ns	IN
7	TdDO(DAV)	Data Out to DAV Fall Delay		TpC		TpC		TpC		TpC		OUT
8	TdDAVOf(RDYlf)	DAV Fall to RDY Fall Delay	0		0		0		0		ns	OUT
9	TdRDYlf(DAVOr)	RDY Fall to DAV Rise Delay		115		115		115		115	ns	OUT
10	TwRDY	RDY Width	110		110		110		110		ns	OUT
11	TdRDYlr(DAVOf)	RDY Rise to DAV Fall Delay		115		115		115		115	ns	OUT

AC CHARACTERISTICS (Continued)

T-49-12-09

A/D Converter Electrical Characteristics

Parameter	Minimum	Typical	Maximum	Units
Resolution		8		Bits
Integral non-linearity		0.5	1	LSB
Differential non-linearity			0.5	LSB
Zero Error at 25°C			5.0	mV
Supply Range	4.5	5.0	5.5	Volts
Power dissapation, no load		35	75	mW
Clock frequency			24	MHz
Input voltage range	VA_{LO}		VA_{HI}	Volts
Conversion time			2	µsec
Input capacitance on ANA	25		40	pF
VA_{HI} range	AN_{GND}		AV_{CC}	Volts
VA_{LO} range	AN_{GND}		AV_{CC}	Volts

Notes:
Voltage 4.5 -5.5V
Temp 0-86°C

D/A Converter Electrical Characteristics

Parameter	Minimum	Typical	Maximum	Units
Resolution		8		Bits
Integral non-linearity		0.25	1	LSB
Differential non-linearity		0.25	0.5	LSB
Setting time, 1/2 LSB		1.5	3.0	µsec
Zero Error at 25°C		10	20	mV
Full Scale error at 25°C		0.25	0.5	LSB
Supply Range	4.5	5.0	5.5	Volts
Power dissapation, no load		10		mW
Ref Input resistance	TBS	25K		Ohms
Output noise voltage		50		µVp-p
VD_{HI} range at 5 volts	2.6		3.5	Volts
VD_{LO} range at 5 volts	0.8		1.7	Volts
VD_{HI}-VD_{LO}, at 5 volts	0.9		2.7	Volts
Capacitive output load, CL			30	pF
Resistive output load, RL	100K			Ohms
Output slew rate	TBS	3.0		V/µsec

Notes:
Voltage 4.5 -5.5V
Temp 0-86°C

EXPANDED REGISTER FILE CONTROL REGISTERS

T-49-12-09

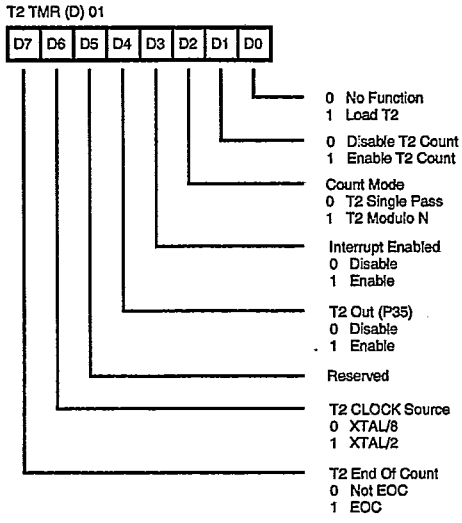


Figure 52. Timer 2 Mode Register (DH 01: Read/Write)

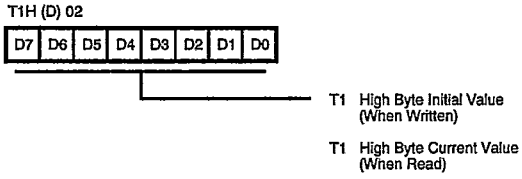


Figure 53. Counter Timer 1 Register High Byte (DH 02: Read/Write)

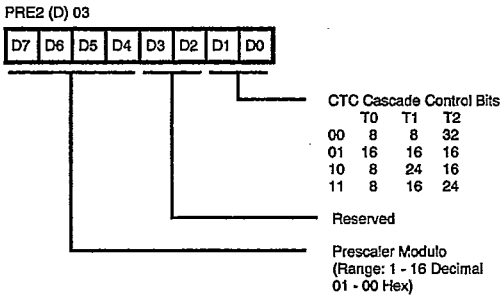


Figure 54. Prescaler 2 Register High Byte (DH 03: Write Only)

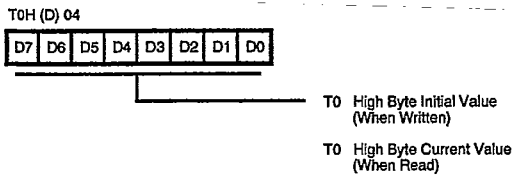


Figure 55. Counter Timer 0 Register High Byte (DH 04: Read/Write)

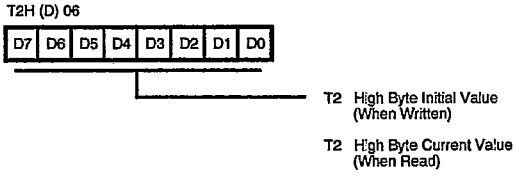


Figure 56. Counter Timer 2 Register High Byte (DH 06: Read/Write)

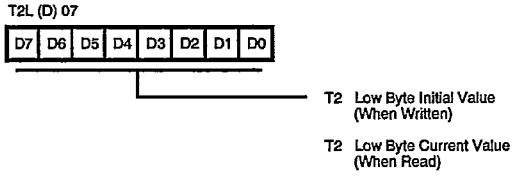


Figure 57. Counter Timer 2 Register Low Byte (DH 07: Read/Write)

Z8 CONTROL REGISTER DIAGRAMS

T-49-12-09

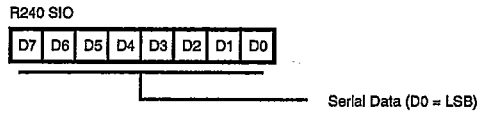


Figure 58. Serial I/O Register (F0H: Read/Write)

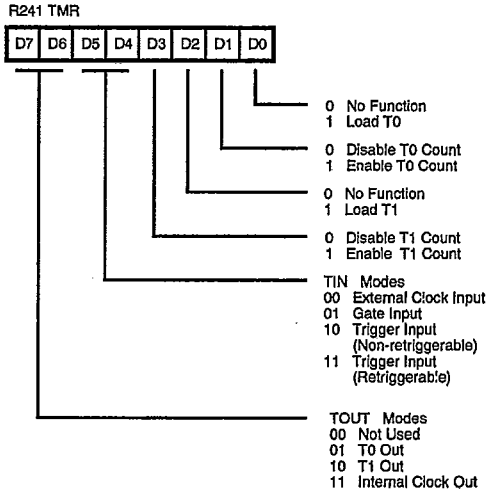


Figure 59. Timer Mode Register (F1H: Read/Write)

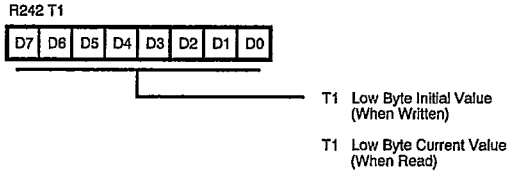


Figure 60. Counter/Timer 1 Register (F2H: Read/Write)

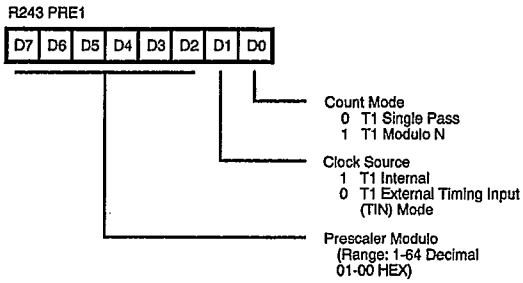


Figure 61. Prescaler 1 Register (F3H: Write Only)

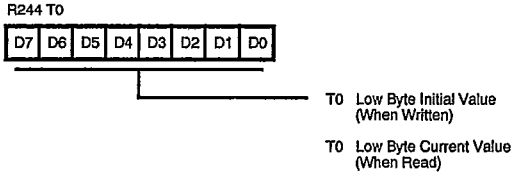


Figure 62. Counter/Timer 0 Register (F4H: Read/Write)

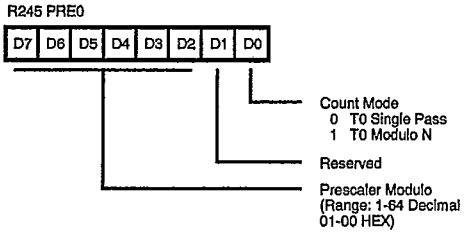


Figure 63. Prescaler 0 Register (F5H: Write Only)

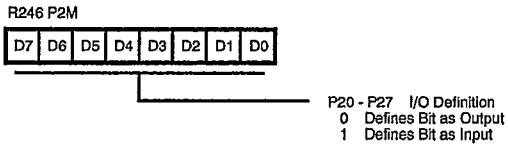


Figure 64. Port 2 Mode Register (F6H: Write Only)

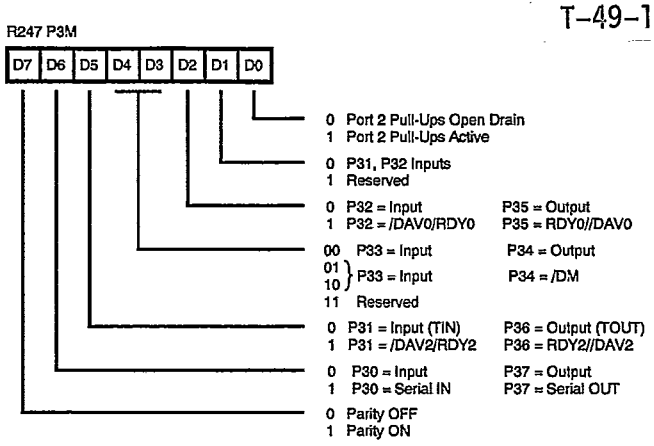


Figure 65. Port 3 Mode Register (F7H: Write Only)

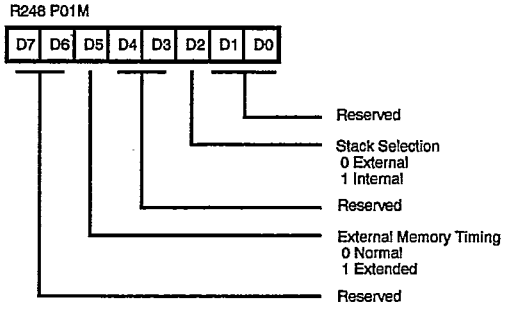


Figure 66. Port 0 and 1 Mode Register (F8H: Write Only)

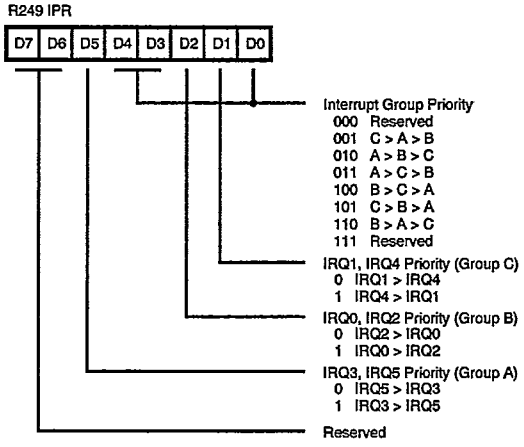


Figure 67. Interrupt Priority Register (F9H: Write Only)

Z8 CONTROL REGISTER DIAGRAMS (Continued)

T-49-12-09

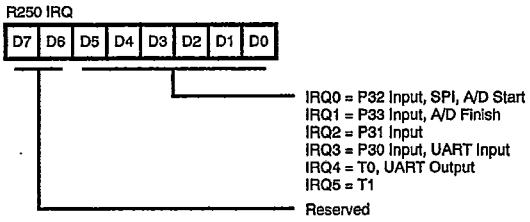


Figure 68. Interrupt Request Register (FAH: Read/Write)

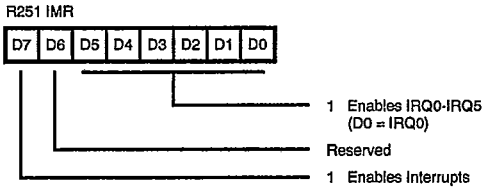


Figure 69. Interrupt Mask Register (FBH: Read/Write)

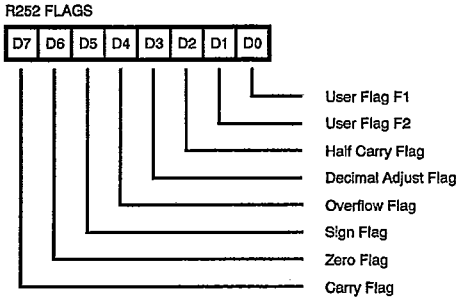


Figure 70. Flag Register (FCH: Read/Write)

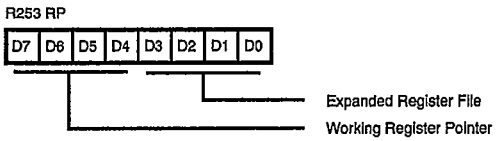


Figure 71. Register Pointer (FDH: Read/Write)

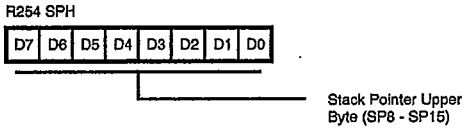


Figure 72. Stack Pointer High (FEH: Read/Write)

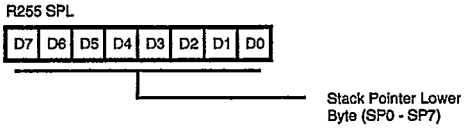


Figure 73. Stack Pointer Low (FFH: Read/Write)

INSTRUCTION SET NOTATION

T-49-12-09

Addressing Modes. The following notation is used to describe the addressing modes and instruction operations as shown in the instruction summary.

Symbol	Meaning
IRR	Indirect register pair or indirect working-register pair address
Irr	Indirect working-register pair only
X	Indexed address
DA	Direct address
RA	Relative address
IM	Immediate
R	Register or working-register address
r	Working-register address only
IR	Indirect-register or indirect working-register address
Ir	Indirect working-register address only
RR	Register pair or working register pair address

Symbols. The following symbols are used in describing the instruction set.

Symbol	Meaning
dst	Destination location or contents
src	Source location or contents
cc	Condition code
@	Indirect address prefix
SP	Stack Pointer
PC	Program Counter
FLAGS	Flag register (Control Register 252)
RP	Register Pointer (R253)
IMR	Interrupt mask register (R251)

Flags. Control register (R252) contains the following six flags:

Symbol	Meaning
C	Carry flag
Z	Zero flag
S	Sign flag
V	Overflow flag
D	Decimal-adjust flag
H	Half-carry flag

Affected flags are indicated by:

0	Clear to zero
1	Set to one
*	Set to clear according to operation
-	Unaffected
x	Undefined

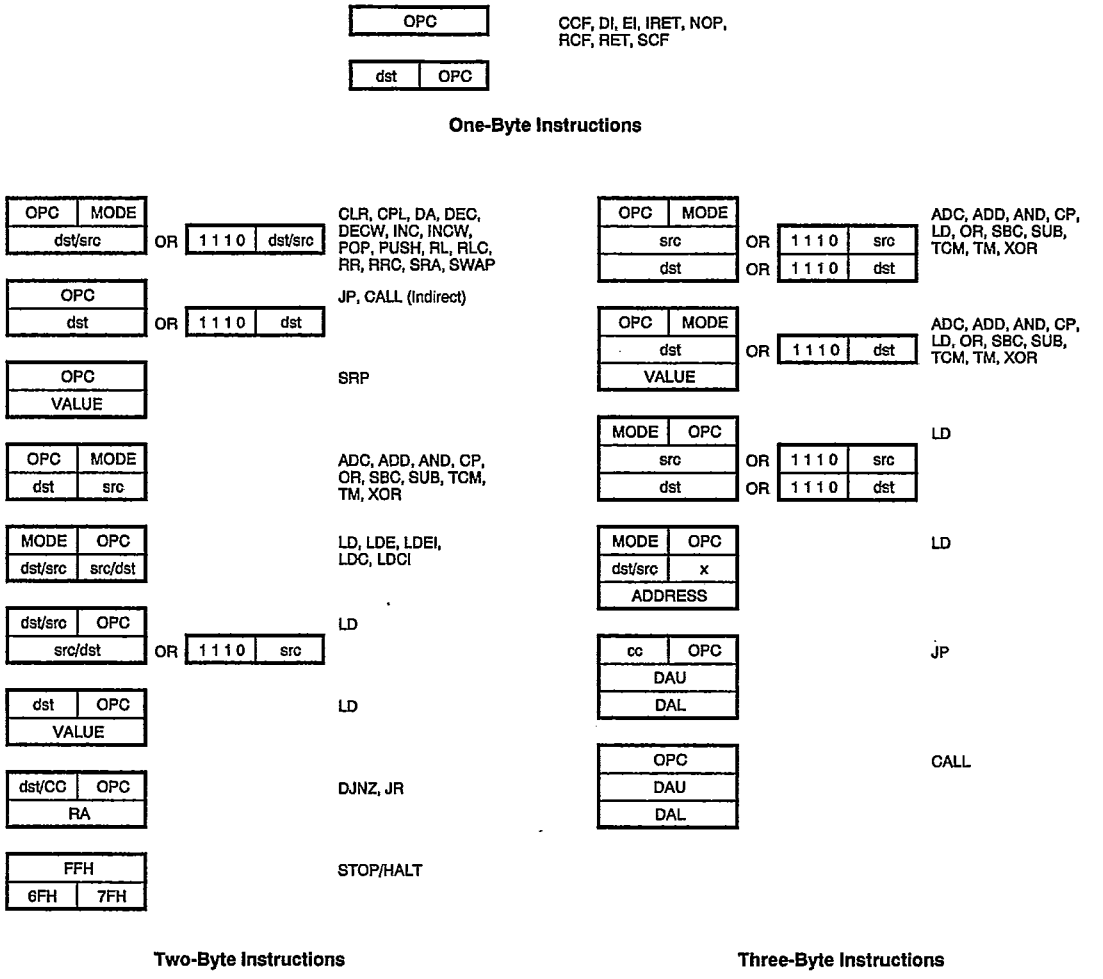
CONDITION CODES

T-49-12-09

Value	Mnemonic	Meaning	Flags Set
1000		Always True	
0111	C	Carry	C = 1
1111	NC	No Carry	C = 0
0110	Z	Zero	Z = 1
1110	NZ	Not Zero	Z = 0
1101	PL	Plus	S = 0
0101	MI	Minus	S = 1
0100	OV	Overflow	V = 1
1100	NOV	No Overflow	V = 0
0110	EQ	Equal	Z = 1
1110	NE	Not Equal	Z = 0
1001	GE	Greater Than or Equal	(S XOR V) = 0
0001	LT	Less Than	(S XOR V) = 1
1010	GT	Greater Than	[Z OR (S XOR V)] = 0
0010	LE	Less Than or Equal	[Z OR (S XOR V)] = 1
1111	UGE	Unsigned Greater Than or Equal	C = 0
0111	ULT	Unsigned Less Than	C = 1
1011	UGT	Unsigned Greater Than	(C = 0 AND Z = 0) = 1
0011	ULE	Unsigned Less Than or Equal	(C OR Z) = 1
0000		Never True	

INSTRUCTION FORMATS

T-49-12-09



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)

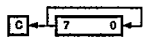
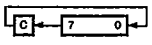
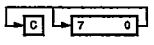
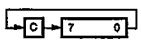
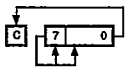
refers to bit 7 of the destination operand.

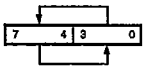
INSTRUCTION SUMMARY (Continued)

T-49-12-09

Instruction and Operation	Address Mode dst	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	src	Opcode Byte (Hex)	Flags Affected C	Z	S	V	D	H
INC dst dst←dst + 1	r R IR		rE r = 0 - F 20 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 r R r X r Ir R R R IR IR	Im R r X r Ir r R IR IM IM R	rC r8 r9 r = 0 - F C7 D7 E3 F3 E4 E5 E6 E7 F5	-	-	-	-	-	-
LDC dst, src	r	Irr	C2	-	-	-	-	-	-
LDCI dst, src dst←src r←r +1; rr←rr + 1	Ir	Irr	C3	-	-	-	-	-	-

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 dst←dst OR src	†		4[ ]	-	*	*	0	-	-
POP dst dst←@SP; SP←SP + 1	R IR		50 51	-	-	-	-	-	-
PUSH src SP←SP - 1; @SP←src		R IR	70 71	-	-	-	-	-	-
RCF C←0			CF	0	-	-	-	-	-
RET PC←@SP; SP←SP + 2			AF	-	-	-	-	-	-
RL dst 	R IR		90 91	*	*	*	*	-	-
RLC dst 	R IR		10 11	*	*	*	*	-	-
RR dst 	R IR		E0 E1	*	*	*	*	-	-
RRC dst 	R IR		C0 C1	*	*	*	*	-	-
SBC dst, src dst←dst←src←C	†		3[ ]	*	*	*	*	1	*
SCF C←1			DF	1	-	-	-	-	-
SRA dst 	R IR		D0 D1	*	*	*	0	-	-
SRP src 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 dst←dst←src	†		2[ ]	*	*	*	*	1	*
SWAP dst 	R IR		F0 F1	X	*	*	X	-	-
TCM dst, src (NOT dst) AND src	†		6[ ]	-	*	*	0	-	-
TM dst, src dst AND src	†		7[ ]	-	*	*	0	-	-
XOR dst, src dst←dst XOR src	†		B[ ]	-	*	*	0	-	-

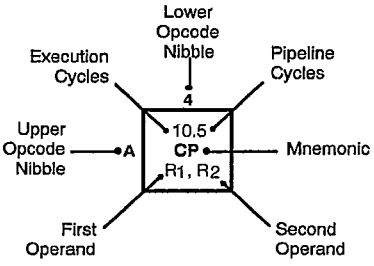
† These instructions have an identical set of addressing modes, which are encoded for brevity. The first opcode nibble is found in the instruction set table above. The second nibble is expressed symbolically by a '[ ]' in this table, and its value is found in the following table to the left of the applicable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

Upper Nibble (Hex) \ Lower 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	



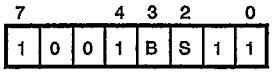
Legend:
R = 8-bit address
r = 4-bit address
R_1 or r_2 = Dst address
R_1 or r_2 = Src address

Sequence:
Opcode, First Operand, Second Operand

Note: Blank areas not defined.

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

DSP COMMANDS



LD (Ri),ADC

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The contents of the ADC register are copied to the register specified by the register pointer.

ADC → (Ri)

Flag change: No

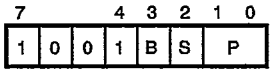
The ADC register is a Read - Only register as far as the DSP is concerned. It may be used to transfer the current A/D conversion result into DSP memory space.

Addressing:
(Ri) is specified by both the bank bit (Bit 3) and bit 2 as follows.

B	S	
0	0	R0
0	1	R1
1	0	R2
1	1	R3

R0 and R1 are register pointers associated with DSP Data Memory RAM(0).

R2 and R3 are register pointers associated with DSP Data Memory RAM(1)



ST (Ri)

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The contents of the accumulator are stored into the register specified by the register pointer.

Accumulator → (Ri)

Flag change: No

Addressing:
(Ri) is specified by both the bank bit (bit 3) and bit 2 as follows:

B	S	
0	0	R0
0	1	R1
1	0	R2
1	1	R3

R0 and R1 are register pointers associated with DSP Data Memory RAM(0).

R2 and R3 are register pointers associated with DSP Data Memory RAM(1).

P field (modification field for register pointers):

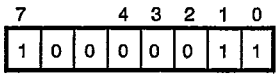
0	0	NOP
0	1	+1
1	0	-1 Loop
1	1	ILLEGAL

Example:
The instruction ST (R2-) will store the accumulator into the register whose address is specified by R2 and then it will decrement R2.

Quick reference:

Accumulator → (Ri)
Ri + 1 or Ri - 1 or Ri → Ri

DSP COMMANDS (Continued)



ST DAC

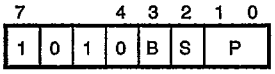
Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The contents of the accumulator are copied to the DAC register.

Accumulator → DAC

Flag change: No

The DAC register is a write-only register as far as the DSP is concerned. It may be used to transfer the DSP results into the D/A converter register.



ADD (Ri)

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The contents of the register specified by the register pointer are added to the accumulator. Ri register is modified according to the P field after execution.

Accumulator ← Accumulator + (Ri)

Flag change: Yes (OV), (L), (Z), (N)

Addressing:
(Ri) is specified by both the bank bit (bit 3) and bit 2 as follows:

B	S	
0	0	R0
0	1	R1
1	0	R2
1	1	R3

R0 and R1 are register pointers associated with DSP Data Memory RAM(0).

R2 and R3 are register pointers associated with DSP Data Memory RAM(1).

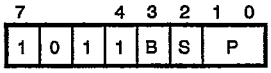
P field (modification field for register pointers):

0	0	NOP
0	1	+1
1	0	-1 Loop
1	1	+1 Loop

Example:
The instruction ADD (R0+) will add the accumulator with the register whose address is specified by R0 and then it will increment R0.

Quick reference:

Accumulator ← Accumulator + (Ri)
Ri + 1 or Ri - 1 or Ri → Ri



SUB (Ri)

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The contents of the register specified by the register pointer are subtracted from the accumulator. Ri register is modified according to the P field after execution.

Accumulator ← Accumulator - (Ri)

Flag change: Yes (OV), (L), (Z), (N)

Addressing:
(Ri) is specified by both the bank bit (bit 3) and bit 2 as follows:

B	S	
0	0	R0
0	1	R1
1	0	R2
1	1	R3

R0 and R1 are register pointers associated with DSP Data Memory RAM(0).

R2 and R3 are register pointers associated with DSP Data Memory RAM(1).

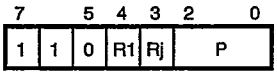
P field (modification field for register pointers):

0	0	NOP
0	1	+1
1	0	-1 Loop
1	1	+1 Loop

Example:
The instruction SUB (R0+) will subtract the register whose address is specified by R0 form the accumulator and then it will increment R0.

Quick reference:

Accumulator ← Accumulator - (Ri)
Ri + 1 or Ri - 1 or Ri → Ri



MLD (Ri),(Rj)

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The contents of the RAM registers whose address is specified by the register pointers Ri and Rj are copied to the multiplier temporary registers X and Y respectively. Ri specifies the addresss for DSP Data RAM(0) and Rj specifies the address for DSP Data RAM(1). Ri,Rj are modified according to the modification field P after accumulator is cleared.

Quick reference:

X ← (Ri)
Y ← (Rj)
Accumulator ← 0

Flag change: No

Addressing:
(Ri) is specified by bit 4 and (Rj) by bit 3 as follows:

If bit 4 = 0, R0 is selected else R1 is selected.
If bit 3 = 0, R2 is selected else R3 is selected.

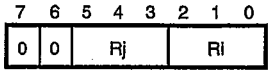
R0 and R1 are register pointers associated with DSP Data Memory RAM(0).

R2 and R3 are register pointers associated with DSP Data Memory RAM(1).

P field (modification field for register pointers):

P(2:0)	Ri	Rj
000	NOP	+1
001	NOP	-1 Loop
010	+1	NOP
011	+1	+1
100	+1	-1 Loop
101	-1 Loop	NOP
110	-1 Loop	+1
111	-1 Loop	-1 Loop

DSP COMMANDS (Continued)



MPYA (Rj),(Ri)

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The multipler output is added to the accumulator. Then, the contents of the RAM registers whose address is specified by register pointers Ri and Rj are copied to the multiplier temporary registers X and Y respectively. Ri specifies the address for DSP Data RAM(0) and Rj specifies the address for DSP Data RAM(1). Ri,Rj are modified according to the modification field P after the copy execution.

Quick reference:

X ← (Ri)
Y ← (Rj)
Accumulator ← Accumlator + P

Flag change: Yes (OV), (L), (Z), (N)

Addressing:
(Ri) is specified by bits 2:0 and (Rj) by bit 5:3 as follows:

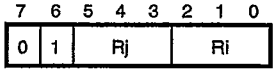
If bit 2 = 0, R0 is selected else R1 is selected.
If bit 5 = 0, R2 is selected else R3 is selected.

R0 and R1 are register pointers associated with DSP Data Memory RAM(0).

R2 and R3 are register pointers associated with DSP Data Memory RAM(1).

Modification field for register pointers:

Bit1/Bit4	Bit0/Bit3	Ri/Rj
0	0	NOP
0	1	+1
1	0	-1 Loop
1	1	+1 Loop



MPYS (Rj),(Ri)

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The multipler output is subtracted from the accumulator. Then, the contents of the RAM registers whose address is specified by register pointers Ri and Rj are copied to the multiplier temporary registers X and Y respectively. Ri specifies the address for DSP Data RAM(0) and Rj specifies the address for DSP Data RAM(1). Ri,Rj are modified according to the modification field P after the copy execution.

Quick reference:

X ← (Ri)
Y ← (Rj)
Accumulator ← Accumlator - P
Ri + 1 or Ri - 1 or Ri → Ri
Rj + 1 or Rj - 1 or Rj → Rj

Flag change: Yes (OV), (L), (Z), (N)

Addressing:
(Ri) is specified by bits 2:0 and (Rj) by bit 5:3 as follows:

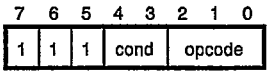
If bit 2 = 0, R0 is selected else R1 is selected.
If bit 5 = 0, R2 is selected else R3 is selected.

R0 and R1 are register pointers associated with DSP Data Memory RAM(0).

R2 and R3 are register pointers associated with DSP Data Memory RAM(1).

Modification field for register pointers:

Bit1/Bit4	Bit0/Bit3	Ri/Rj
0	0	NOP
0	1	+1
1	0	-1 Loop
1	1	+1 Loop



MOD cond, OP

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The contents of the accumulator are modified if the condition is met. Otherwise a NOP is executed. The exact nature of the accumulator modification is specified by the OPCODE field.

OPCODE	Mnemonic	Operation	Flags
000	ROR	Rotate Right	(OV) (Z) (N)
001	ROL	Rotate Left	(OV) (Z) (N)
010	SHR	Arithmetic Right Shift	(OV) (Z) (N)
011	SHL	Arithmetic Left Shift	(OV) (Z) (N)
100	INC	Increment A	(OV) (Z) (N)
101	DEC	Decrement A	(OV) (Z) (N)
110	NEG	Negate A	(OV) (Z) (N)
111	ABS	Absolute A	(OV) (Z) (N)

Condition Field:

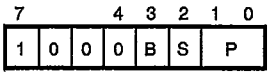
Bit 4	Bit 3	
0	0	ILLEGAL
0	1	Always True (MOD, always, OP)
1	0	IF OV = 1 (MOD, OV = 1, OP)
1	1	IF N = 1 (MOD, N = 1, OP)

If the condition is met then

000 (L) → a15, a15 → a14, ..., a1 → a0, a0 → (L)
001 (L) ← a15, a15 ← a14, ..., a1 ← a0, a0 ← (L)
010 Accumulator/2 → Accumulator
011 Accumulator*2 → Accumulator
100 Accumulator+1 → Accumulator
101 Accumulator-1 → Accumulator
110 -Accumulator → Accumulator
111 |Accumulator| → Accumulator

Flag change: Yes (OV), (L), (Z), (N)

DSP COMMANDS (Continued)



LD (Ri)

Instruction: 1 Byte
Cycle: 1 Cycle

Operation:
The contents of the register specified by the register pointer are copied to the accumulator.

(Ri) → Accumulator

Flag change: No

Addressing:
(Ri) is specified by both the bank bit (bit 3) and bit 2 as follows:

B	S	
0	0	R0
0	1	R1
1	0	R2
1	1	R3

R0 and R1 are register pointers associated with DSP Data Memory RAM(0).

R2 and R3 are register pointers associated with DSP Data Memory RAM(1).

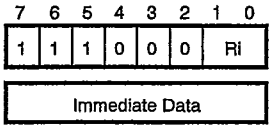
P field (modification field for register pointers):

0	0	NOP
0	1	+1
1	0	-1 Loop
1	1	ILLEGAL

Example:
The instruction LD (R0+) will load the accumulator with the register whose address is specified by R0 and then it will increment R0.

Quick reference:

(Ri) → Accumulator
Ri + 1 or Ri - 1 or Ri → Ri



LDI Ri

Instruction: 2 Bytes
Cycle: 2 Cycles

Operation:
The register pointers (R0, R1, R2, R3) are loaded with the immediate value specified

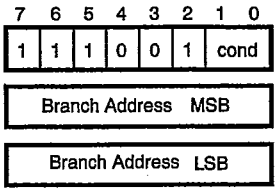
Flag change: No

Addressing:
(Ri) is specified by bits (1:0) as follows.

Bit 1	Bit 0	
0	0	R0
0	1	R1
1	0	R2
1	1	R3

Quick reference:

Ri ← Immediate data



Quick reference:

If condition = True, then Pseudo PC = Branch address
Else, Pseudo PC = Pseudo PC + 1

BRA cond,addr

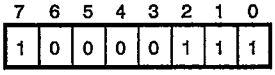
Instruction: 3 Bytes
Cycle: 3 Cycles

Operation:
The condition is tested and the branch is taken if the condition is true. The branch address has to be within the internal program memory space (512 bytes).

Flag change: No

Branch conditions are specified as follows:

Bit 1	Bit 0	
0	0	Branch on NOT ZERO
0	1	Branch on OVERFLOW
1	0	Branch ALWAYS
1	1	Branch on LINK bit SET



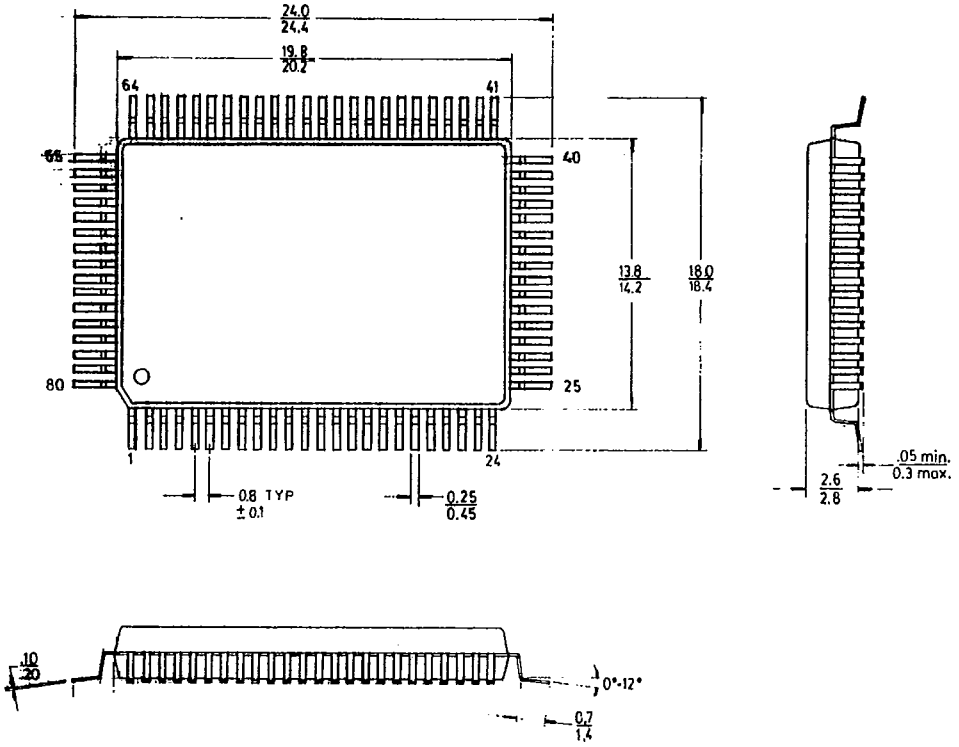
DSP STOP

Instruction: 1 Byte
Cycle: 1 Cycle

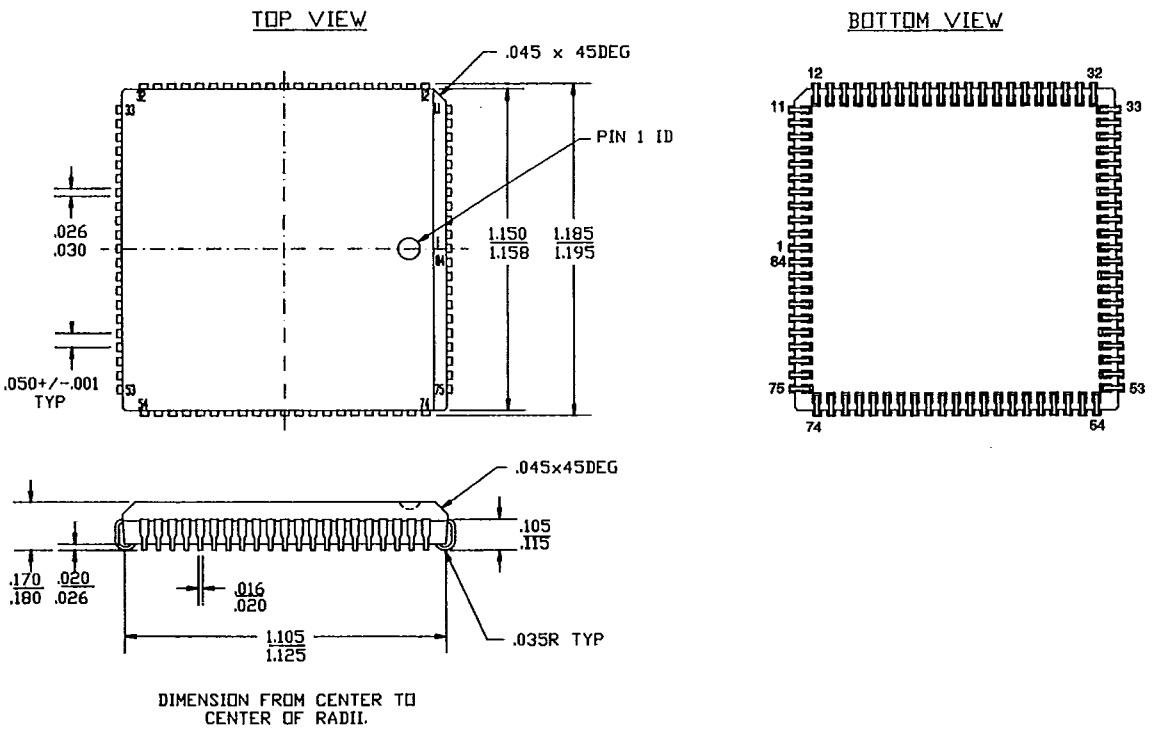
Operation:
This instruction will stop the DSP.

Flag change: No

PACKAGE INFORMATION

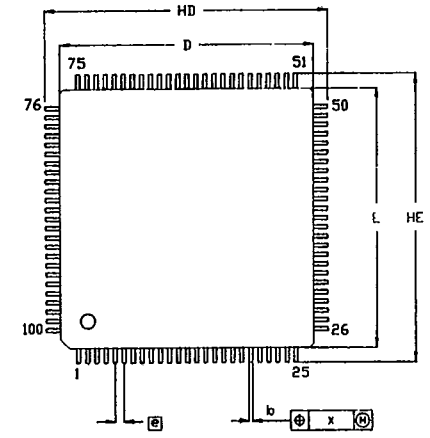


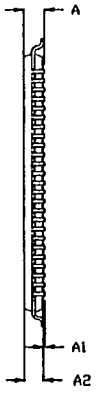
80-Pin QFP Package Diagram



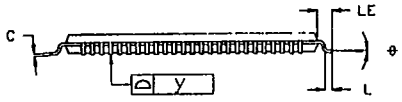
84-Pin PLCC Package Diagram

PACKAGE INFORMATION (Continued)





SYMBOL	DIMENSIONS IN MM.		
	MIN.	NOM.	MAX.
A	–	–	1.60
A1	–	–	0.20
A2	–	1.40	–
b	–	0.20	–
c	0.10	0.15	0.20
D	–	14.0	–
E	–	14.0	–
e	–	0.50	–
L	–	0.60	–
LE	–	1.00	–
HD	–	16.00	–
HE	–	16.00	–
x	–	–	0.15
y	–	–	0.10
θ	0°	–	5°



100-Pin VQFP Package Diagram

ORDERING INFORMATION

Z86C95

20 MHz		
84-pin PLCC	80-pin QFP	100-pin VQFP
Z86C9520VSC	Z86C9520FSC	Z86C9520ASC
Z86C9520VEC	Z86C9520FEC	Z86C9520AEC

24 MHz		
84-pin PLCC	80-pin QFP	100-pin VQFP
Z86C9524VSC	Z86C9524FSC	Z86C9524ASC
Z86C9524VEC	Z86C9524FEC	Z86C9524AEC

For fast results, contact your local Zilog sales office for assistance in ordering the part desired.

Package
V = Plastic Chip Carrier

Longer Lead Time
F = Plastic Quad Flat Pack
A = Very Small QFP

Temperature
S = 0° C to +70° C
E = -40° C to +105° C

Speed
20 = 20 MHz
24 = 24 MHz

Environmental
C = Plastic Standard

Example:
Z 86C95 24 V S C is an 86C95 24 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow