Z89C00

16-BIT DIGITAL SIGNAL PROCESSOR

FEATURES

- 16-Bit Single Cycle Instructions
- Zero Overhead Hardware Looping
- 16-Bit Data
- Ready Control for Slow Peripherals
- Single Cycle Multiply/Accumulate (100 ns)
- Six-Level Stack
- 512 Words of On-Chip RAM
- Static Single-Cycle Operation
- 16-Bit I/O Port
- 4K Words of On-Chip Masked ROM
- Three Vectored Interrupts
- 64K Words of External Program Address Space
- Two Conditional Branch Inputs/Two User Outputs
- 24-Bit ALU, Accumulator and Shifter
- IBM® PC Development Tools

GENERAL DESCRIPTION

The Z89C00 is a second generation, 16-bit, fractional, two's complement CMOS Digital Signal Processor (DSP). Most instructions, including multiply and accumulate, are accomplished in a single clock cycle. The processor contains 1 Kbyte of on-chip data RAM (two blocks of 256 16-bit words), 4K words of program ROM and 64K words of program memory addressing capability. Also, the processor features a 24-bit ALU, a 16 x 16 multiplier, a 24-bit Accumulator and a shifter. Additionally, the processor contains a six-level stack, three vectored interrupts and two inputs for conditional program jumps. Each RAM block contains a set of three pointers which may be incremented or decremented automatically to affect hardware looping without software overhead. The data RAMs can be simultaneously addressed and loaded to the multiplier for a true single cycle multiply.

There is a 16-bit address and a 16-bit data bus for external program memory and data, and a 16-bit I/O bus for transferring data. Additionally, there are two general purpose user inputs and two user outputs. Operation with slow peripherals is accomplished with a ready input pin. The clock may be stopped to conserve power.

Development tools for the IBM PC include a relocatable assembler, a linker loader, and an ANSI-C compiler. Also, the development tools include a simulator/debugger, a cross assembler for the TMS320 family assembly code and a hardware emulator.

To assist the user in understanding the Z89C00 DSP Q15 two's complement fractional multiplication, an application note has been included in this product specification as an appendix.

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}

GENERAL DESCRIPTION (Continued)

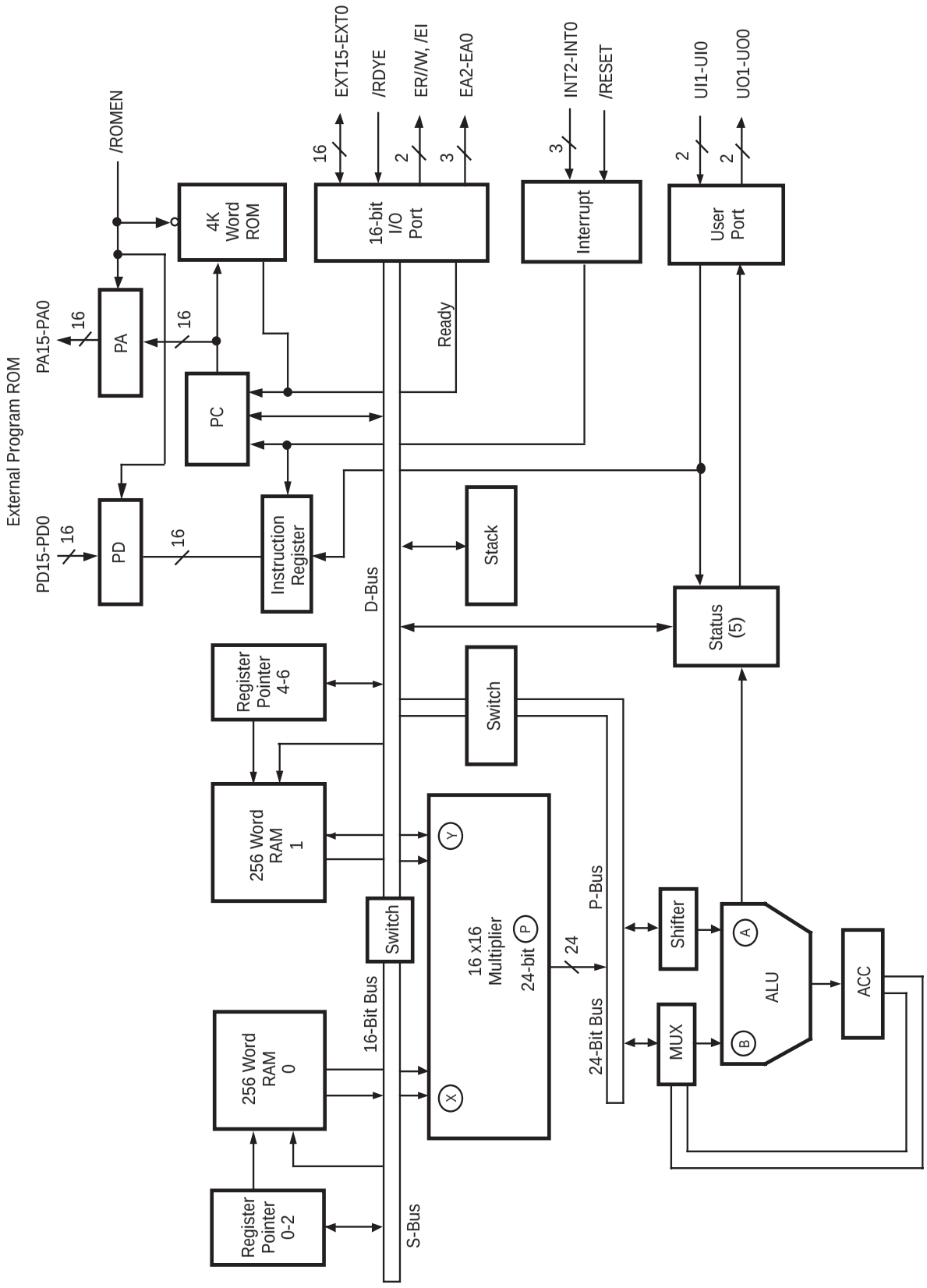


Figure 1. Functional Block Diagram

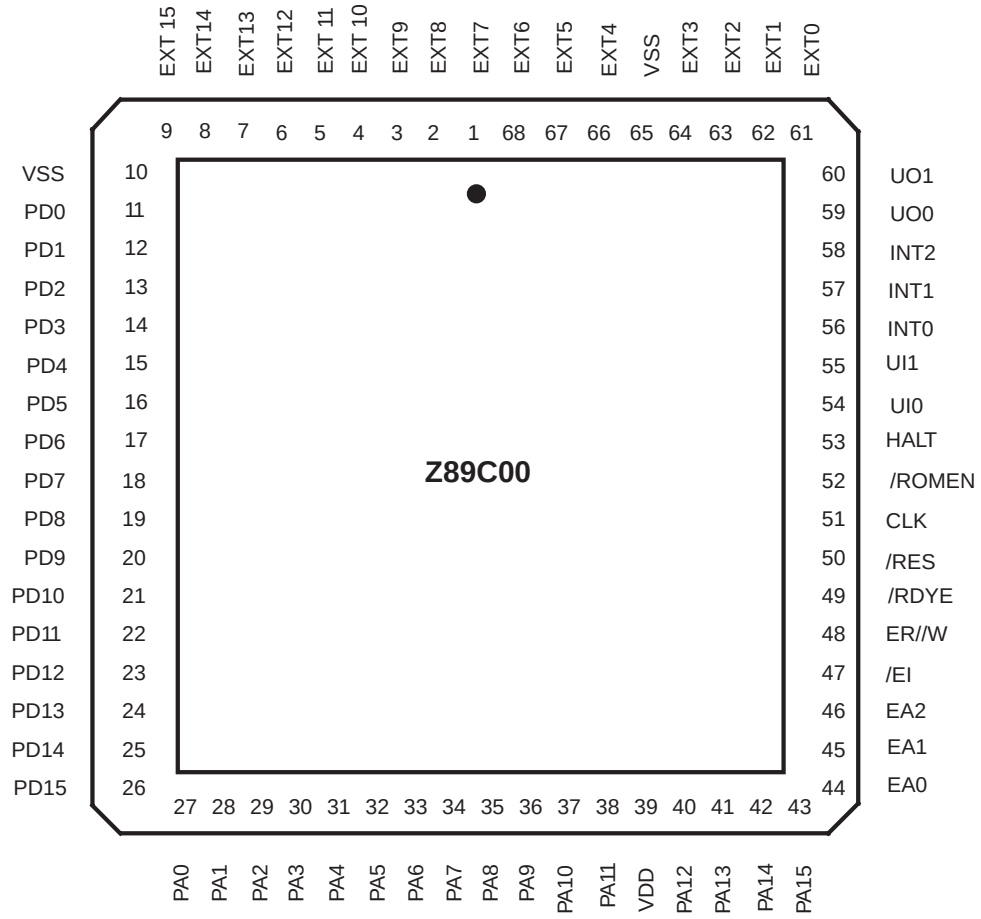


Figure 2. 68-Pin PLCC Pin Assignments

Table 1. 68-Pin PLCC Pin Identification

No.	Symbol	Function	Direction
1-9	EXT15-EXT7	External data bus	Input/Output
10	V_{SS}	Ground	Input
11-26	PD15-PD0	Program data bus	Input
27-38	PA11-PA0	Program address bus	Output
39	V_{DD}	Power Supply	Input
40-43	PA15-PA12	Program address bus	Output
44-46	EA2-EA0	External address bus	Output
47	/EI	R/W for external bus	Output
48	ER//W	External bus direction	Output
49	/RDYE	Data ready	Input
50	/RES	Reset	Input
51	CLK	Clock	Input
52	/ROMEN	Enable ROM	Input
53	HALT	Stop execution	Input
54-55	UI1-UI0	User inputs	Input
56-58	INT2-INT1	Interrupts	Input
59-60	UO1-UO0	User outputs	Output
61-64	EXT3-EXT0	External data bus	Input/Output
65	V_{SS}	Ground	Input
66-68	EXT6-EXT4	External data bus	Input/Output

PIN FUNCTIONS

CLK *Clock* (input). External clock. The clock may be stopped to reduce power.

EXT15-EXT0 *External Data Bus* (input/output). Data bus for user defined outside registers such as an ADC or DAC. The pins are normally in output mode except when the outside registers are specified as source registers in the instructions. All the control signals exist to allow a read or a write through this bus.

ER/W *External Bus Direction* (output, active Low). Data direction signal for EXT-Bus. Data is available from the CPU on EXT15-EXT0 when this signal is Low. EXT-Bus is in input mode (high-impedance) when this signal is High.

EA2-EA0 *External Address* (output). User-defined register address output. One of eight user-defined external registers is selected by the processor with these address pins for read or write operations. Since the addresses are part of the processor memory map, the processor is simply executing internal reads and writes.

/EI *Enable Input* (output). Write timing signal for EXT-Bus. Data is read by the external peripheral on the rising edge of /EI. Data is read by the processor on the rising edge of CLK, not /EI.

HALT *Halt State* (input). Stop Execution Control. The CPU continuously executes NOPs and the program counter remains at the same value when this pin is held High. This signal must be synchronized with CLK.

INT2-INT0 *Three Interrupts* (rising edge triggered). Interrupt request 2-0. Interrupts are generated on the rising edge of the input signal. Interrupt vectors for the interrupt service starting address are stored in the program memory locations 0FFFH for INT0, 0FFEH for INT1 and 0FFDH for INT2. Priority is: 2 = lowest, 0 = highest.

PA15-PA0 *Program memory address bus* (output). For up to 64K x 16 external program memory. These lines are tri-stated during Reset Low.

PD15-PD0 *Program Memory Data Input* (input). Instructions or data are read from the address specified by PD15-PD0, through these pins and are executed or stored.

/RES *Reset* (input, active Low). Asynchronous reset signal. A Low level on this pin generates an internal reset signal. The /RES signal must be kept Low for at least one clock cycle. The CPU pushes the contents of the PC onto the stack and then fetches a new Program Counter (PC) value from program memory address 0FFCH after the Reset signal is released. RES Low tri-states the PA and PD bases.

/ROMEN *ROM Enable* (input). An active Low signal enables the internal ROM. Program execution begins at 0000H from the ROM. An active High input disables the ROM and external fetches occur from address 0000H.

/RDYE *Data Ready* (input). User-supplied Data Ready signal for data to and from external data bus. This pin stretches the /EI and ER//W lines and maintains data on the address bus and data bus. The ready signal is sampled from the rising edge of the clock with appropriate setup and hold times. The normal write cycle will continue from the next rising clock only if ready is active.

UI1-UI0 *Two Input Pins* (input). General purpose input pins. These input pins are directly tested by the conditional branch instructions. These are asynchronous input signals that have no special clock synchronization requirements.

UO1-UO0 *Two Output Pins* (output). General purpose output pins. These pins reflect the inverted value of status register bits S5 and S6. These bits may be used to output data by writing to the status register.

ADDRESS SPACE

Program Memory. Programs of up to 4K words can be masked into internal ROM. Four locations are dedicated to the vector address for the three interrupts (0FFDH-0FFFH) and the starting address following a Reset (0FFCH). Internal ROM is mapped from 0000H to 0FFFH, and the highest location for program is 0FFBH. If the /ROMEN pin is held High, the internal ROM is inactive and the processor executes external fetches from 0000H to FFFFH. In this case, locations FFFC-FFFF are used for vector addresses.

Internal Data RAM. The Z89C00 has an internal 512 x 16-bit word data RAM organized as two banks of 256 x 16-bit words each, referred to as RAM0 and RAM1. Each data RAM bank is addressed by three pointers, referred to as Pn:0 (n = 0-2) for RAM0 and Pn:1 (n = 0-2) for RAM1. The RAM addresses for RAM0 and RAM1 are arranged from 0-255 and 256-511, respectively. The address pointers, which may be written to or read from, are 8-bit registers connected to the lower byte of the internal 16-bit D-Bus and are used to perform no overhead looping. Three addressing modes are available to access the Data RAM: register indirect, direct addressing, and short form direct. These modes are discussed in detail later. The contents of the RAM can be read or written in one machine cycle per word without disturbing any internal registers or status other than the RAM address pointer used for each RAM. The contents of each RAM can be loaded simultaneously into the X and Y inputs of the multiplier.

Registers. The Z89C00 has 12 internal registers and up to an additional eight external registers. The external registers are user definable for peripherals such as A/D or D/A or to DMA or other addressing peripherals. External registers are accessed in one machine cycle the same as internal registers.

FUNCTIONAL DESCRIPTION

General. The Z89C00 is a high-performance Digital Signal Processor with a modified Harvard-type architecture with separate program and data memory. The design has been optimized for processing power and minimizing silicon space.

Instruction Timing. Many instructions are executed in one machine cycle. Long immediate instructions and Jump or Call instructions are executed in two machine cycles. When the program memory is referenced in internal RAM indirect mode, it takes three machine cycles. In addition, one more machine cycle is required if the PC is selected as the destination of a data transfer instruction. This only happens in the case of a register indirect branch instruction.

An Acc + P => Acc; a(i) * b(j) → P calculation and modification of the RAM pointers, is done in one machine cycle. Both operands, a(i) and b(j), can be located in two independent RAM (0 and 1) addresses.

Multiply/Accumulate. The multiplier can perform a 16-bit x 16-bit multiply or multiply accumulate in one machine cycle using the Accumulator and/or both the X and Y inputs. The multiplier produces a 32-bit result, however, only the 24 most significant bits are saved for the next instruction or accumulation. The multiplier provides a flow through operation whenever the X or Y register is updated, an automatic multiply operation is performed and the P register is updated. For operations on very small numbers where the least significant bits are important, the data should first be scaled by eight bits (or the multiplier and multiplicand by four bits each) to avoid truncation errors. Note that all inputs to the multiplier should be fractional two's complement 16-bit binary numbers. This puts them in the range [–1 to 0.9999695], and the result is in 24-bits so that the range is [–1 to 0.9999999]. In addition, if 8000H is loaded into both X and Y registers, the resulting multiplication is considered an illegal operation as an overflow would result. Positive one cannot be represented in fractional notation, and the multiplier will actually yield the result 8000H x 8000H = 8000H (–1 x –1 = –1).

ALU. The 24-bit ALU has two input ports, one of which is connected to the output of the 24-bit Accumulator. The other input is connected to the 24-bit P-Bus, the upper 16 bits of which are connected to the 16-bit D-Bus. A shifter between the P-Bus and the ALU input port can shift the data by three bits right, one bit right, one bit left or no shift.

Hardware Stack. A six-level hardware stack is connected to the D-Bus to hold subroutine return addresses or data. The CALL instruction pushes PC+2 onto the stack. The RET instruction pops the contents of the stack to the PC.

User Inputs. The Z89C00 has two inputs, UI0 and UI1, which may be used by jump and call instructions. The jump or call tests one of these pins and if appropriate, jumps to a new location. Otherwise, the instruction behaves like a NOP. These inputs are also connected to the status register bits S10 and S11 which may be read by the appropriate instruction (Figure 3).

User Outputs. The status register bits S5 and S6 connect through an inverter to UO0 and UO1 pins and may be written to by the appropriate instruction.

Interrupts. The Z89C00 has three positive edge triggered interrupt inputs. An interrupt is acknowledged at the end of any instruction execution. It takes two machine cycles to enter an interrupt instruction sequence. The PC is pushed onto the stack. A RET instruction transfers the contents of the stack to the PC and decrements the stack pointer by one word. The priority of the interrupts is 0 = highest, 2 = lowest.

Registers. The Z89C00 has 12 physical internal registers and up to eight user-defined external registers. The EA2-EA0 determines the address of the external registers. The /EI, /RDYE, and ER//W signals are used to read or write from the external registers.

REGISTERS

There are 12 internal registers which are defined below:

Register	Register Definition
P	Output of Multiplier, 24-bit, Read Only
X	X Multiplier Input, 16-bit
Y	Y Multiplier Input, 16-bit
A	Accumulator, 24-bit
SR	Status Register, 16-bit
Pn:b	Six Ram Address Pointers, 8-bit Each
PC	Program Counter, 16-bit

The following are virtual registers as physical RAM does not exist on the chip.

Register	Register Definition
EXTn	External registers, 16-bit
BUS	D-Bus
Dn:b	Eight Data Pointers

P holds the result of multiplications and is read only.

X and **Y** are two 16-bit input registers for the multiplier. These registers can be utilized as temporary registers when the multiplier is not being used. The contents of the P register will change if X or Y is changed.

A is a 24-bit Accumulator. The output of the ALU is sent to this register. When 16-bit data is transferred into this register, it goes into the 16 MSB's and the least significant eight bits are set to zero. Only the upper 16 bits are transferred to the destination register when the Accumulator is selected as a source register in transfer instructions.

Pn:b are the pointer registers for accessing data RAM. (n = 0,1,2 refer to the pointer number) (b = 0,1 refers to RAM bank 0 or 1). They can be directly read from or written to, and can point to locations in data RAM or indirectly to Program Memory.

EXT(n) are external registers (n = 0 to 7). There are eight 16-bit registers here for accessing External data, peripherals, or memory. Note that the actual register RAM does not exist on the chip, but would exist as part of the external device such as an ADC result latch.

BUS is a read-only register which, when accessed, returns the contents of the D-Bus.

Dn:b refer to possible locations in RAM that can be used as a pointer to locations in program memory. The programmer decides which location to choose from two bits in the status register and two bits in the operand. Thus, only the lower 16 possible locations in RAM can be specified. At any one time there are eight usable pointers, four per bank, and the four pointers are in consecutive locations in RAM. For example, if S3/S4 = 01 in the status register, then D0:0/D1:0/D2:0/D3:0 refer to locations 4/5/6/7 in RAM bank 0. Note that when the data pointers are being written to, a number is actually being loaded to Data RAM, so they can be used as a limited method for writing to RAM.

REGISTERS (Continued)

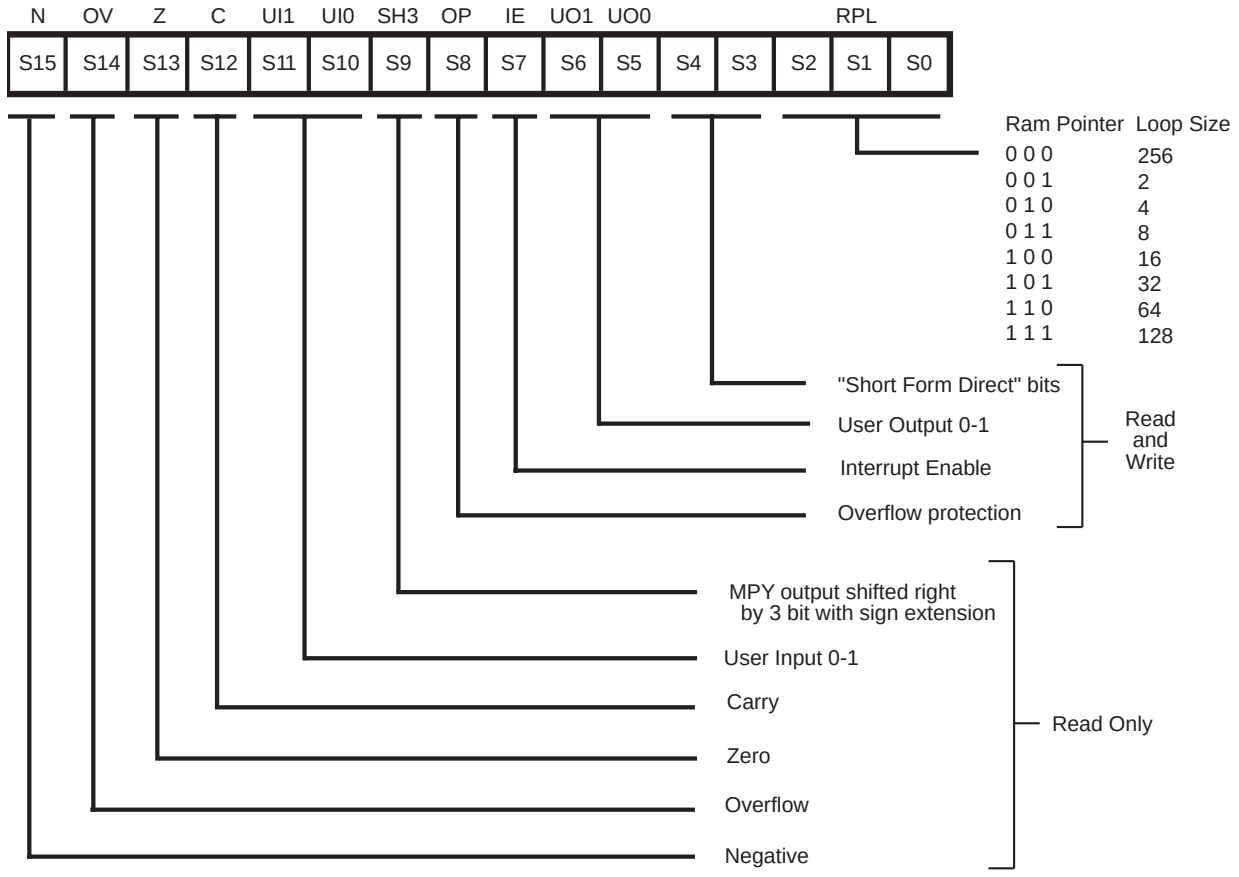


Figure 3. Status Register

SR is the status register (Figure 3) which contains the ALU status and certain control bits as shown in the following table.

Status Register Bit	Function
S15 (N)	ALU Negative
S14 (OV)	ALU Overflow
S13 (Z)	ALU Zero
S12 (L)	Carry
S11 (UI1)	User Input 1
S10 (UI0)	User Input 0
S9 (SH3)	MPY Output Shifted Right by Three Bits
S8 (OP)	Overflow Protection
S7 (IE)	Interrupt Enable
S6 (UO1)	User Output 1
S5 (UO0)	User Output 0
S4-3	"Short Form Direct" Bits
S2-0 (RPL)	RAM Pointer Loop Size

RPL Description

S2	S1	S0	Loop Size
0	0	0	256
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The status register may always be read in its entirety. S15-S10 are set/reset by the hardware and can only be read by software. S9-S0 can be written by software.

S15-S12 are set/reset by the ALU after an operation. S11-S10 are set/reset by the user inputs. S6-S0 are control bits described elsewhere. S7 enables interrupts. S8, if 0 (reset), allows the hardware to overflow. If S8 is set, the hardware clamps at maximum positive or negative values instead of overflowing. If S9 is set and a multiply instruction is used, the shifter shifts the result three bits right with sign extension.

PC is the Program Counter. When this register is assigned as a destination register, one NOP machine cycle is added automatically to adjust the pipeline timing.

RAM ADDRESSING

The address of the RAM is specified in one of three ways (Figure 4):

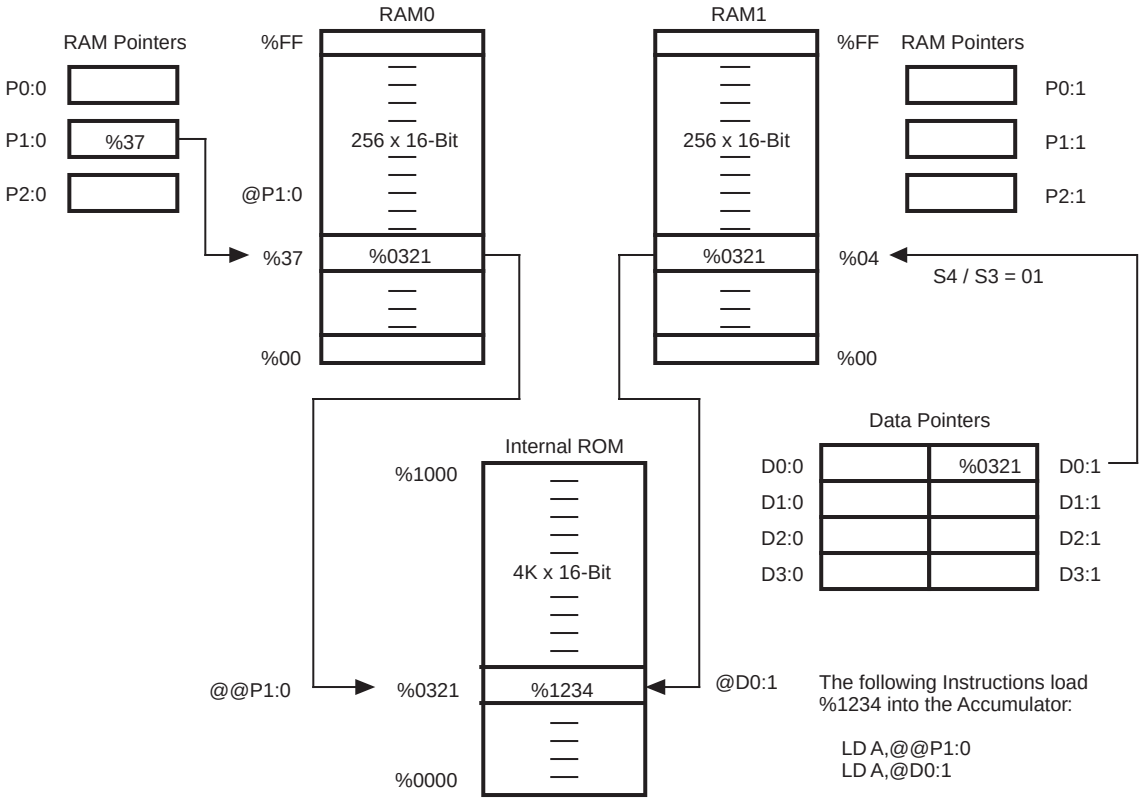


Figure 4. RAM, ROM, and Pointer Architecture

1. Register Indirect

Pn:b n = 0-2, b = 0-1

The most commonly used method is a register indirect addressing method, where the RAM address is specified by one of the three RAM address pointers (n) for each bank (b). Each source/destination field in Figures 5 and 8 may be used by an indirect instruction to specify a register pointer and its modification after execution of the instruction.

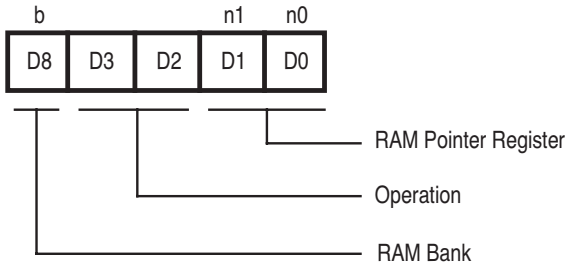


Figure 5. Indirect Register

RAM ADDRESSING (Continued)

The register pointer is specified by the first and second bits in the source/destination field and the modification is specified by the third and fourth bits according to the following table:

D3-D0		Meaning
00xx	NOP	No Operation
01xx	+1	Simple Increment
10xx	–1/LOOP	Decrement Modulo the Loop Count
11xx	+1/LOOP	Increment Modulo the Loop Count
xx00	P0:0 or P0:1	See Note a.
xx01	P1:0 or P1:1	See Note a.
xx10	P2:0 or P2:1	See Note a.
xx11		See Short Form Direct

Note:
a. If bit 8 is zero, P0:0 to P2:0 are selected; if bit 8 is one, P0:1 to P2:1 are selected.

When Loop mode is selected, the pointer to which the loop is referring will cycle up or down, depending on whether a –LOOP or +LOOP is specified. The size of the loop is obtained from the least significant three bits of the Status Register. The increment or decrement of the register is accomplished modulo the loop size. As an example, if the loop size is specified as 32 by entering the value 101 into bits 2-0 of the Status Register (S2-S0) and an increment +LOOP is specified in the address field of the instruction, i.e., the RPi field is 11xx, then the register specified by RPi will increment, but only the least significant five bits will be affected. This means the actual value of the pointer will cycle round in a length 32 loop, and the lowest or highest value of the loop, depending on whether the loop is up or down, is set by the three most significant bits. This allows repeated access to a set of data in RAM without software intervention. To clarify, if the pointer value is 10101001 and if the LOOP = 32, the pointer increments up to 10111111, then drops down to 10100000 and starts again. The upper three bits remaining unchanged. Note that the original value of the pointer is not retained.

2. Direct Register

The second method is a direct addressing method. The address of the RAM is directly specified by the address field of the instruction. Because this addressing method consumes nine bits (0-511) of the instruction field, some instructions cannot use this mode (Figure 6).

Figures 8 to 13 show the different register instruction formats along with the two tables below Figure 8.

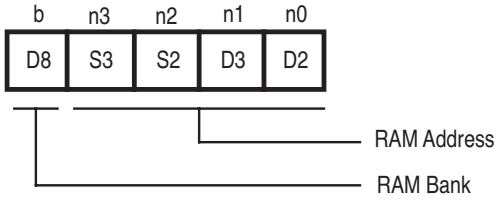


Figure 7. Short Form Direct Address

3. Short Form Direct

Dn:b n = 0-3, b = 0-1

The last method is called Short Form Direct Addressing, where one out of 32 addresses in internal RAM can be specified. The 32 addresses are the 16 lower addresses in RAM Bank 0 and the 16 lower addresses in RAM Bank 1. Bit 8 of the instruction field determines RAM Bank 0 or 1. The 16 addresses are determined by a 4-bit code comprised of bits S3 and S4 of the status register and the third and fourth bits of the Source/ Destination field. Because this mode can specify a direct address in a short form, all of the instructions using the register indirect mode can use this mode (Figure 7). This method can access only the lower 16 addresses in the both RAM banks and as such has limited use. The main purpose is to specify a data register, located in the RAM bank, which can then be used to point to a program memory location. This facilitates down-loading look-up tables, etc. from program memory to RAM.

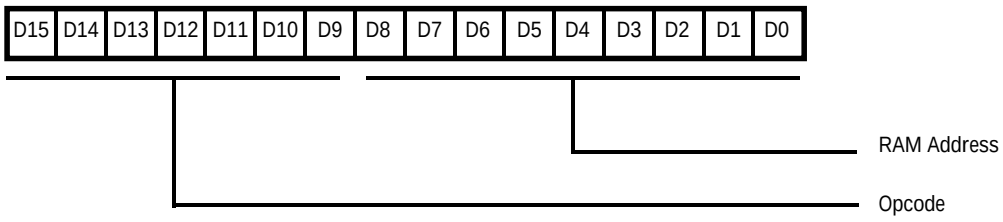
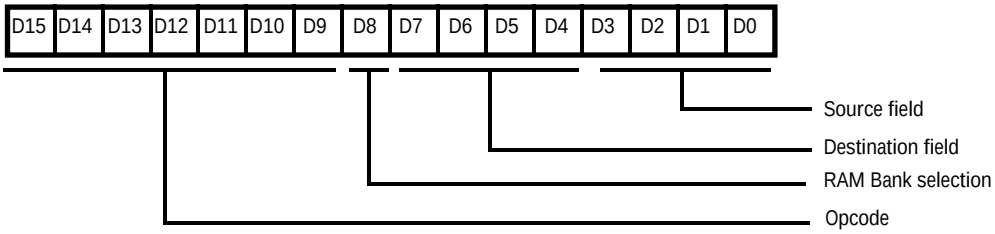


Figure 6. Direct Internal RAM Address Format

INSTRUCTION FORMAT



Note:
Source/Destination fields can specify either register or RAM addresses in RAM pointer indirect mode.

Figure 8. General Instruction Format

A. Registers

Source/Destination	Register
0000	BUS**
0001	X
0010	Y
0011	A
0100	SR
0101	STACK
0110	PC
0111	P**
1000	EXT0
1001	EXT1
1010	EXT2
1011	EXT3
1100	EXT4
1101	EXT5
1110	EXT6
1111	EXT7

B. Register Pointers Field

Source/Destination	Meaning
00xx	NOP
01xx	+1
10xx	–1/LOOP
11xx	+1/LOOP
xx00	P0:0 or P0:1*
xx01	P1:0 or P1:1*
xx10	P2:0 or P2:1*
xx11	Short Form Direct Mode

Notes:
* If RAM Bank bit is 0, then Pn:0 are selected.
If RAM Bank bit is 1, then Pn:1 are selected.
** Read only.

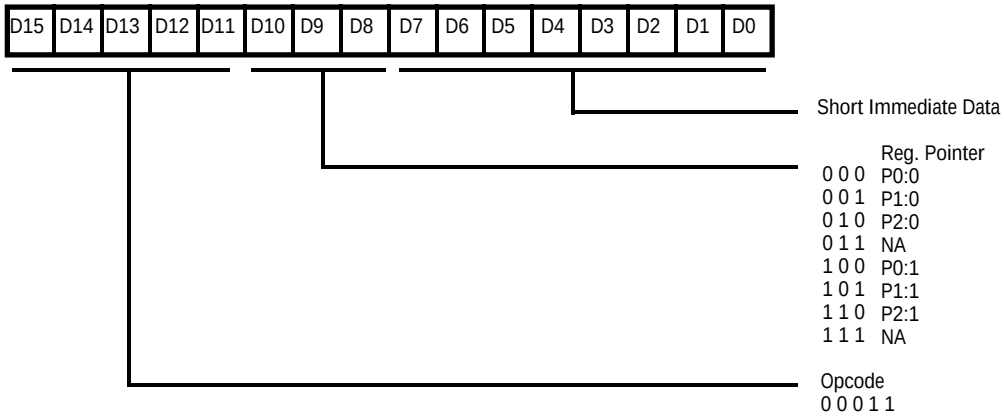


Figure 9. Short Immediate Data Load Format

INSTRUCTION FORMAT (Continued)



Figure 10. Immediate Data Load Format

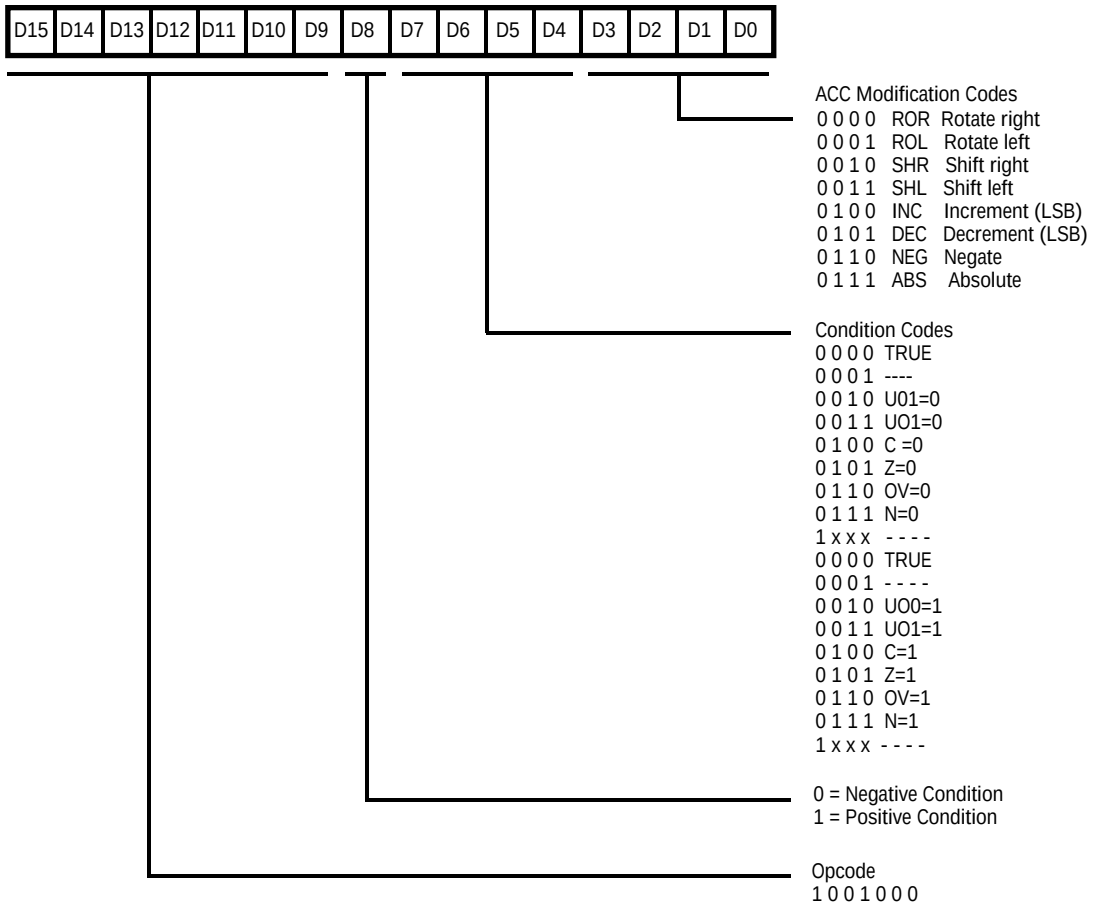


Figure 11. Accumulator Modification Format

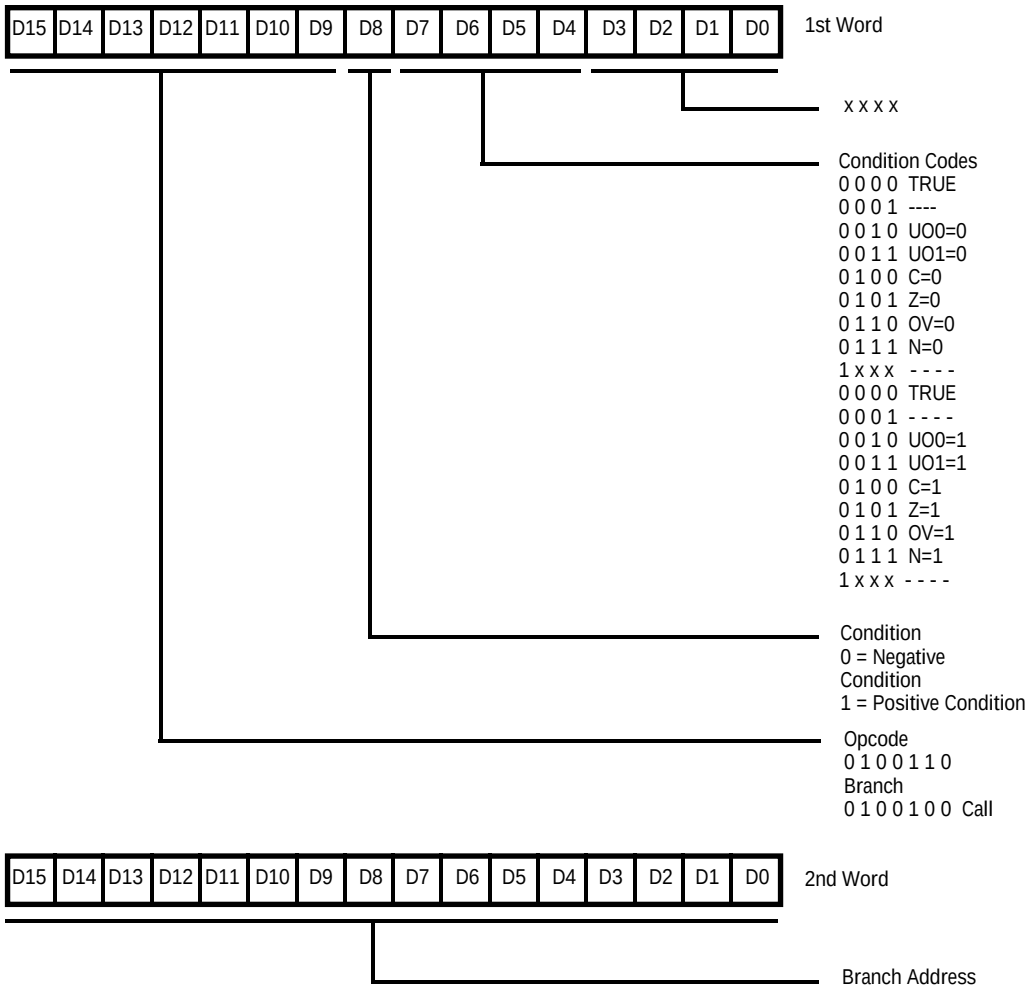


Figure 12. Branching Format

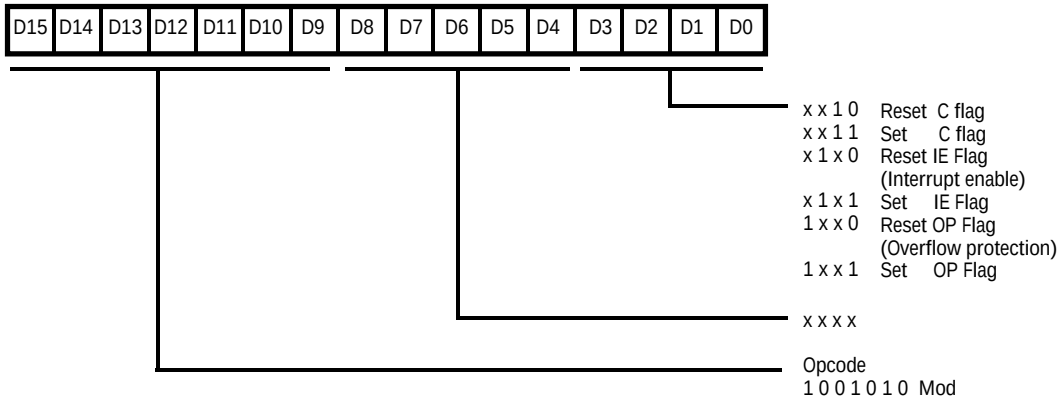


Figure 13. Flag Modification Format

ADDRESSING MODES

This section discusses the syntax of the addressing modes supported by the DSP assembler. The symbolic name is used in the discussion of instruction syntax in the instruction descriptions.

Symbolic Name	Syntax	Description
<pregs>	Pn:b	Pointer Register
<dregs> (Points to RAM)	Dn:b	Data Register
<hwregs>	X,Y,PC,SR,P EXTn,A,BUS	Hardware Registers
<accind> (Points to Program Memory)	@A	Accumulator Memory Indirect
<direct>	<expression>	Direct Address Expression
<limm>	#<const exp>	Long (16-bit) Immediate Value
<simm>	#<const exp>	Short (8-bit) Immediate Value
<regind> (Points to RAM)	@Pn:b @Pn:b+ @Pn:b–LOOP @Pn:b+LOOP	Pointer Register Indirect Pointer Register Indirect with Increment Pointer Register Indirect with Loop Decrement Pointer register Indirect with Loop Increment
<memind> (Points to Program Memory)	@@Pn:b @Dn:b @@Pn:b–LOOP @@Pn:b+LOOP @@Pn:b+	Pointer Register Memory Indirect Data Register Memory Indirect Pointer Register Memory Indirect with Loop Decrement Pointer Register Memory Indirect with Loop Increment Pointer Register Memory Indirect with Increment

There are eight distinct addressing modes for transfer of data (Figure 4 and the table above).

<pregs>, <hwregs> These two modes are used for simple loads to and from registers within the chip such as loading to the Accumulator, or loading from a pointer register. The names of the registers need only be specified in the operand field. (Destination first then source)

<regind> This mode is used for indirect accesses to the data RAM. The address of the RAM location is stored in the pointer. The "@" symbol indicates "indirect" and precedes the pointer, so @P1:1 tells the processor to read or write to a location in RAM1, which is specified by the value in the pointer.

<dregs> This mode is also used for accesses to the data RAM but only the lower 16 addresses in either bank. The 4-bit address comes from the status register and the operand field of the data pointer. Note that data registers are typically used not for addressing RAM, but loading data from program memory space.

<memind> This mode is used for indirect, indirect accesses to the program memory. The address of the memory is located in a RAM location, which is specified by the value in a pointer. So @@P1:1 tells the processor to read (write is not possible) from a location in memory, which is specified by a value in RAM, and the location of the RAM is in turn specified by the value in the pointer. Note that the data pointer can also be used for a memory access in this manner, but only one "@" precedes the pointer. In both cases the memory address stored in RAM is incremented by one each time the addressing mode is used to allow easy transfer of sequential data from program memory.

<accind> Similar to the previous mode, the address for the program memory read is stored in the Accumulator. @A in the second operand field loads the number in memory specified by the address in A.

<direct> The direct mode allows read or write to data RAM from the Accumulator by specifying the absolute address of the RAM in the operand of the instruction. A number between 0 and 255 indicates a location in RAM0, and a number between 256 and 511 indicates a location in RAM1.

<limm> This indicates a long immediate load. A 16-bit word can be copied directly from the operand into the specified register or memory.

<simm> This can only be used for immediate transfer of 8-bit data in the operand to the specified RAM pointer.

CONDITION CODES

The following table defines the condition codes supported by the DSP assembler. If the instruction description refers to the <cc> (condition code) symbol in one of its addressing modes, the instruction will only execute if the condition is true.

Name	Description
C	Carry
EQ	Equal (same as Z)
F	False
IE	Interrupts Enabled
MI	Minus
NC	No Carry
NE	Not Equal (same as NZ)
NIE	Not Interrupts Enabled
NOV	Not Overflow
NU0	Not User Zero

Name	Description
NU1	Not User One
NZ	Not zero
OV	Overflow
PL	Plus (Positive)
U0	User Zero
U1	User One
UGE	Unsigned Greater Than or Equal (Same as NC)
ULT	Unsigned Less Than (Same as C)
Z	Zero

INSTRUCTION DESCRIPTIONS

Inst.	Description	Synopsis	Operands	Words	Cycles	Examples
ABS	Absolute Value	ABS[<cc>,]<src>	<cc>,A	1	1	ABS NC,A
			A	1	1	ABS A
ADD	Addition	ADD<dest>,<src>	A,<pregs>	1	1	ADD A,P0:0
			A,<dregs>	1	1	ADD A,D0:0
			A,<limm>	2	2	ADD A,#%1234
			A,<memind>	1	3	ADD A,@@P0:0
			A,<direct>	1	1	ADD A,%F2
			A,<regind>	1	1	ADD A,@P1:1
			A,<hwregs>	1	1	ADD A,X
AND	Bitwise AND	AND<dest>,<src>	A,<pregs>	1	1	AND A,P2:0
			A,<dregs>	1	1	AND A,D0:1
			A,<limm>	2	2	AND A,#%1234
			A,<memind>	1	3	AND A,@@P1:0
			A,<direct>	1	1	AND A,%2C
			A,<regind>	1	1	AND A,@P1:2+LOOP
			A,<hwregs>	1	1	AND A,EXT3
CALL	Subroutine call	CALL [<cc>,]<address>	<cc>,<direct>	2	2	CALL Z,sub2
			<direct>	2	2	CALL sub1
CCF	Clear carry flag	CCF	None	1	1	CCF
CIEF	Clear Carry Flag	CIEF	None	1	1	CIEF
COPF	Clear OP flag	COPF	None	1	1	COPF
CP	Comparison	CP<src1>,<src2>	A,<pregs>	1	1	CP A,P0:0
			A,<dregs>	1	1	CP A,D3:1
			A,<memind>	1	3	CP A,@@P0:1
			A,<direct>	1	1	CP A,%FF
			A,<regind>	1	1	CP A,@P2:1+
			A,<hwregs>	1	1	CP A,STACK
			A<limm>	2	2	CP A,#%FFCF
DEC	Decrement	DEC [<cc>,]<dest>	<cc>A,	1	1	DEC NZ,A
			A	1	1	DEC A
INC	Increment	INC [<cc>,] <dest>	<cc>,A	1	1	INC PL,A
			A	1	1	INC A
JP	Jump	JP [<cc>,]<address>	<cc>,<direct>	2	2	JP NIE,Label
			<direct>	2	2	JP Label

Inst.	Description	Synopsis	Operands	Words	Cycles	Examples
LD	Load destination with source	LD<dest>,<src>	A,<hwregs>	1	1	LD A,X
			A,<dregs>	1	1	LD A,D0:0
			A,<pregs>	1	1	LD A,P0:1
			A,<regind>	1	1	LD A,@P1:1
			A,<memind>	1	3	LD A,@D0:0
			A,<direct>	1	1	LD A,124
			<direct>,A	1	1	LD 124,A
			<dregs>,<hwregs>	1	1	LD D0:0,EXT7
			<pregs>,<simm>	1	1	LD P1:1,#%FA
			<pregs>,<hwregs>	1	1	LD P1:1,EXT1
			<regind>,<limm>	1	1	LD@P1:1,#1234
			<regind>,<hwregs>	1	1	LD @P1:1+,X
			<hwregs>,<pregs>	1	1	LD Y,P0:0
			<hwregs>,<dregs>	1	1	LD SR,D0:0
			<hwregs>,<limm>	2	2	LD PC,#%1234
			<hwregs>,<accind>	1	3	LD X,@A
			<hwregs>,<memind>	1	3	LD Y,@D0:0
			<hwregs>,<regind>	1	1	LD A,@P0:0–LOOP
			<hwregs>,<hwregs>	1	1	LD X,EXT6
			Note: If X or Y register is the destination, an automatic multiply operation is performed.			
			Note: The P register is Read Only and cannot be destination.			
			Note: LD EXT_N, EXT_N is not allowed.			
			Note: LD A, @A is not allowed.			
MLD	Multiply	MLD<src1>,<src2>[,<bank switch>]	<hwregs>,<regind>	1	1	MLD A,@P0:0+LOOP
			<hwregs>,<regind>,<bank switch>	1	1	MLD A,@P1:0,OFF
			<regind>,<regind>	1	1	MLD @P1:1,@P2:0
			<regind>,<regind>,<bank switch>	1	1	MLD @P0:1,@P1:0,ON
			Note: If src1 is <regind> it must be a bank 1 register. Src2's <regind must be a bank 0 register.			
			Note: <hwregs> for src1 cannot be X.			
			Note: For the operands <hwregs>, <regind> the <band switch> defaults to OFF. For the operands <regind>, the <bank switch> defaults to ON.			
MPYA	Multiply and add	MPYA <src1>,<src2>[,<bank switch>]	<hwregs>,<regind>	1	1	MPYA A,@P0:0
			<hwregs>,<regind>,<bank switch>	1	1	MPYA A,@P1:0,OFF
			<regind>,<regind>	1	1	MPYA @P1:1,@P2:0
			<regind>,<regind>,<bank switch>	1	1	MPYA@P0:1,@P1:0,ON
			Note: If src1 is <regind> it must be a bank 1 register. Src2's <regind> must be a bank 0 register.			
			Note: <hwregs> for src1 cannot be X or A.			
			Note: For the operands <hwregs>, <regind> the <bank switch> defaults to OFF. For the operands <regind>, the <bank switch> defaults to ON.			

INSTRUCTION DESCRIPTIONS (Continued)

Inst.	Description	Synopsis	Operands	Words	Cycles	Examples
MPYS	Multiply and subtract	MPYS<src1>,<src2>[,<bank switch>]	<hwregs>,<regind>	1	1	MPYS A,@P0:0
			<hwregs>,<regind>,<bank switch>	1	1	MPYS A,@P1:0,OFF
			<regind>,<regind>	1	1	MPYS @P1:1,@P2:0
			<regind>,<regind>,<bank switch>	1	1	MPYS @P0:1,@P1:0,ON
			Note: If src1 is <regind> it must be a bank 1 register. Src2's <regind> must be a bank 0 register.			
			Note: <hwregs> for src1 cannot be X or A.			
			Note: For the operands <hwregs>, <regind> the <bank switch> defaults to OFF. For the operands <regind>, <regind> the <bank switch> defaults to ON.			
NEG	Negate	NEG <cc>,A	<cc>, A	1	1	NEG MI,A
			A	1	1	NEG A
NOP	No operation	NOP	None	1	1	NOP
OR	Bitwise OR	OR <dest>,<src>	A, <pregs>	1	1	OR A,P0:1
			A, <dregs>	1	1	OR A, D0:1
			A, <limm>	2	2	OR A,#%2C21
			A, <memind>	1	3	OR A,@@P2:1+
			A, <direct>	1	1	OR A, %2C
			A, <regind>	1	1	OR A,@P1:0–LOOP
			A, <hwregs>	1	1	OR A,EXT6
POP	Pop value from stack	POP <dest>	<pregs>	1	1	POP P0:0
			<dregs>	1	1	POP D0:1
			<regind>	1	1	POP @P0:0
			<hwregs>	1	1	POP A
PUSH	Push value onto stack	PUSH <src>	<pregs>	1	1	PUSH P0:0
			<dregs>	1	1	PUSH D0:1
			<regind>	1	1	PUSH @P0:0
			<hwregs>	1	1	PUSH BUS
			<limm>	2	2	PUSH #12345
			<accind>	1	3	PUSH @A
			<memind>	1	3	PUSH @@P0:0
RET	Return from subroutine	RET	None	1	2	RET
RL	Rotate Left	RL <cc>,A	<cc>,A	1	1	RL NZ,A
			A	1	1	RL A
RR	Rotate Right	RR <cc>,A	<cc>,A	1	1	RR C,A
			A	1	1	RR A

Inst.	Description	Synopsis	Operands	Words	Cycles	Examples
SCF	Set C flag	SCF	None	1	1	SCF
SIEF	Set IE flag	SIEF	None	1	1	SIEF
SLL	Shift left logical	SLL	[<cc>,]A	1	1	SLL NZ,A
			A	1	1	SLL A
SOPF	Set OP flag	SOPF	None	1	1	SOPF
SRA	Shift right arithmetic	SRA<cc>,A	<cc>,A	1	1	SRA NZ,A
			A	1	1	SRA A
SUB	Subtract	SUB<dest>,<src>	A,<pregs>	1	1	SUB A,P1:1
			A,<dregs>	1	1	SUB A,D0:1
			A,<limm>	2	2	SUB A,#%2C2C
			A, <memind>	1	3	SUB A,@D0:1
			A, <direct>	1	1	SUB A,%15
			A, <regind>	1	1	SUB A,@P2:0-LOOP
			A, <hwregs>	1	1	SUB A,STACK
XOR	Bitwise exclusive OR	XOR <dest>,<src>	A, <pregs>	1	1	XOR A,P2:0
			A, <dregs>	1	1	XOR A,D0:1
			A, <limm>	2	2	XOR A,#13933
			A, <memind>	1	3	XOR A,@@P2:1+
			A, <direct>	1	1	XOR A,%2F
			A, <regind>	1	1	XOR A,@P2:0
			A, <hwregs>	1	1	XOR A,BUS

Bank Switch Enumerations. The third (optional) operand of the MLD, MPYA and MPYS instructions represents whether a bank switch is set on or off. To more clearly represent this, two keywords are used (ON and OFF) which state the direction of the switch. These keywords are referred to in the instruction descriptions through the <bank switch> symbol.

ABSOLUTE MAXIMUM RATINGS

Symbol	Description	Min	Max	Units
V_{CC}	Supply Voltage(*)	–0.5	7.0	V
T_{STG}	Storage Temp.	–65°	+150°	C
T_A	Oper. Ambient Temp.	†		C

Notes:
* Voltages on all pins with respect to ground.
† See Ordering Information

Stresses 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 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 ground. Positive current flows into the referenced pin (Test Load Diagram, Figure 14).

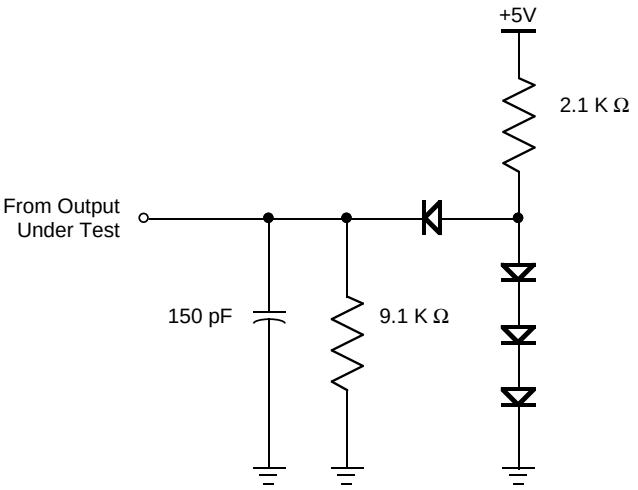


Figure 14. Test Load Diagram

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V ± 5%, T_A = 0°C to +70°C unless otherwise specified)

Symbol	Parameter	Condition	Min.	Max.	Units
I_{CC}	Supply Current	V_{CC} = 5.25V fclock = 10 MHz		60	mA
I_{CC1}	Halt Mode	V_{CC} = 5.25V fclock = 0 MHz (stopped)	1	5	mA
V_{IH}	Input High Level		0.9 V_{CC}		V
V_{IL}	Input Low Level			0.1 V_{CC}	V
I_{IL}	Input Leakage			1	µA
V_{OH}	Output High Voltage	I_{OH} = –100 µA	V_{CC} – 0.2		V
V_{OL}	Output Low Voltage	I_{OL} = 0.5 mA		0.5	V
I_{FL}	Output Floating Leakage Current			5	µA

AC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V ± 5%, T_A = 0°C to +70°C unless otherwise specified)

No.	Symbol	Parameter	Min.	Max.	Units
1	TCY	Clock Cycle Time	100	1000	ns
2	PWW	Clock Pulse Width	45		ns
3	Tr	Clock Rise Time	2	4	ns
4	Tf	Clock Fall Time	2	4	ns
5	TEAD	EA,ER//W Delay from CK	9	33	ns
6	TXVD	EXT Data Output Valid from CLK	5	27	ns
7	TXWH	EXT Data Output Hold from CLK	6	22	ns
8	TXRS	EXT Data Input Setup Time	15		ns
9	TXRH	EXT Data Input Hold from CLK	5	15	ns
10	TIEDR	/EI Delay Time from Rising CLK Edge	3	15	ns
11	TIEDF	/EI Delay Time from Falling CLK Edge	0	23	ns
12	TINS	Interrupt Setup Time	5		ns
13	TINL	Interrupt Hold Time	15		ns
14	TPAD	PA Delay from CLK	5	22	ns
15	TPDS	PD Input Setup Time	20		ns
16	TPDH	PD Input Hold Time	20	28	ns
17	TCTLS	Halt Setup Time	5		ns
18	TCTLH	Halt Hold Time	20		ns
19	RDYS	Ready Setup Time	10		ns
20	RDYH	Ready Hold Time	7		ns

AC TIMING DIAGRAM

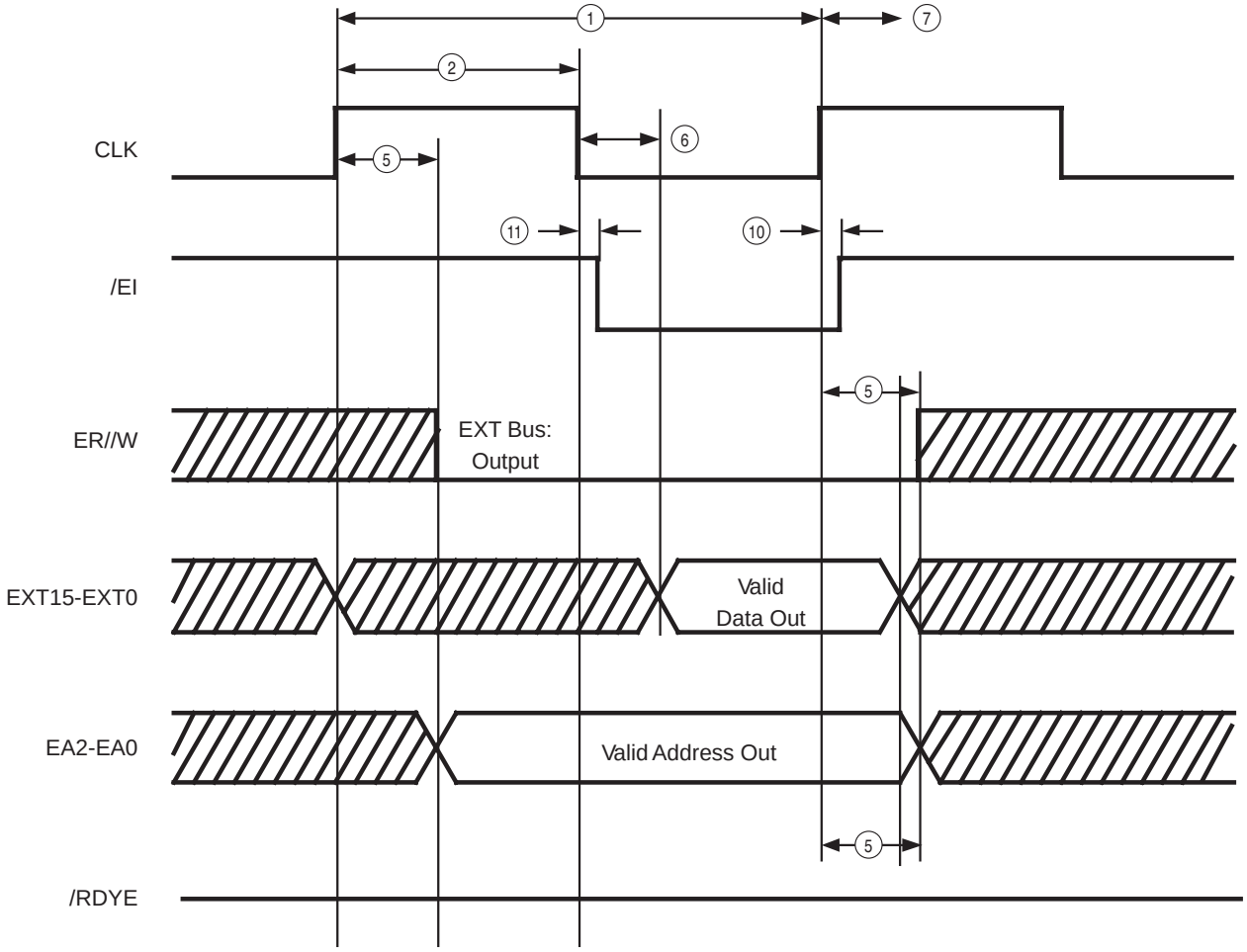


Figure 15. Write To External Device Timing

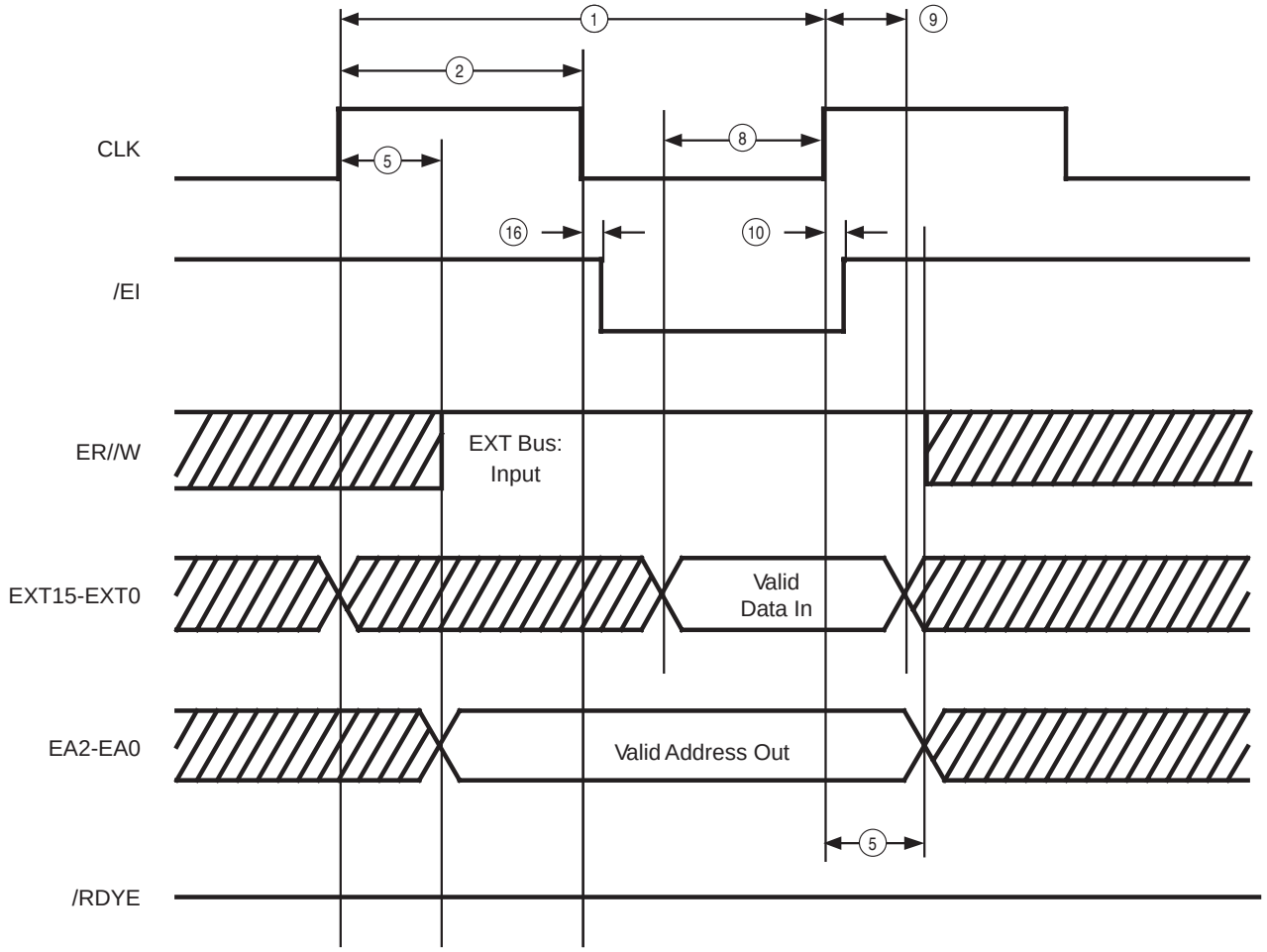


Figure 16. Read From External Device Timing

AC TIMING DIAGRAM

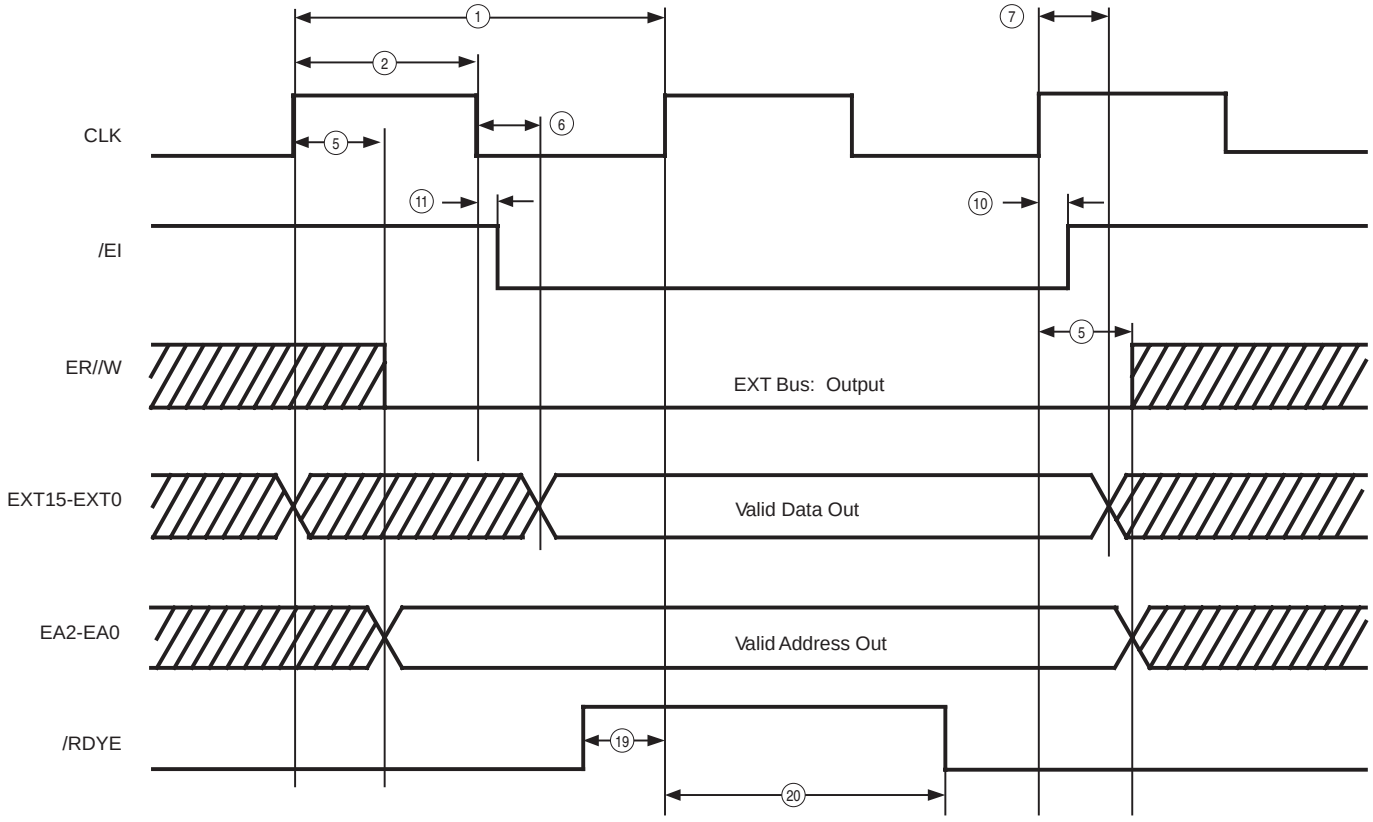


Figure 17. Write To External Device Timing (/RDYE used to hold data one clock cycle)*

Note: * /RDYE is checked during rising edge of clock.

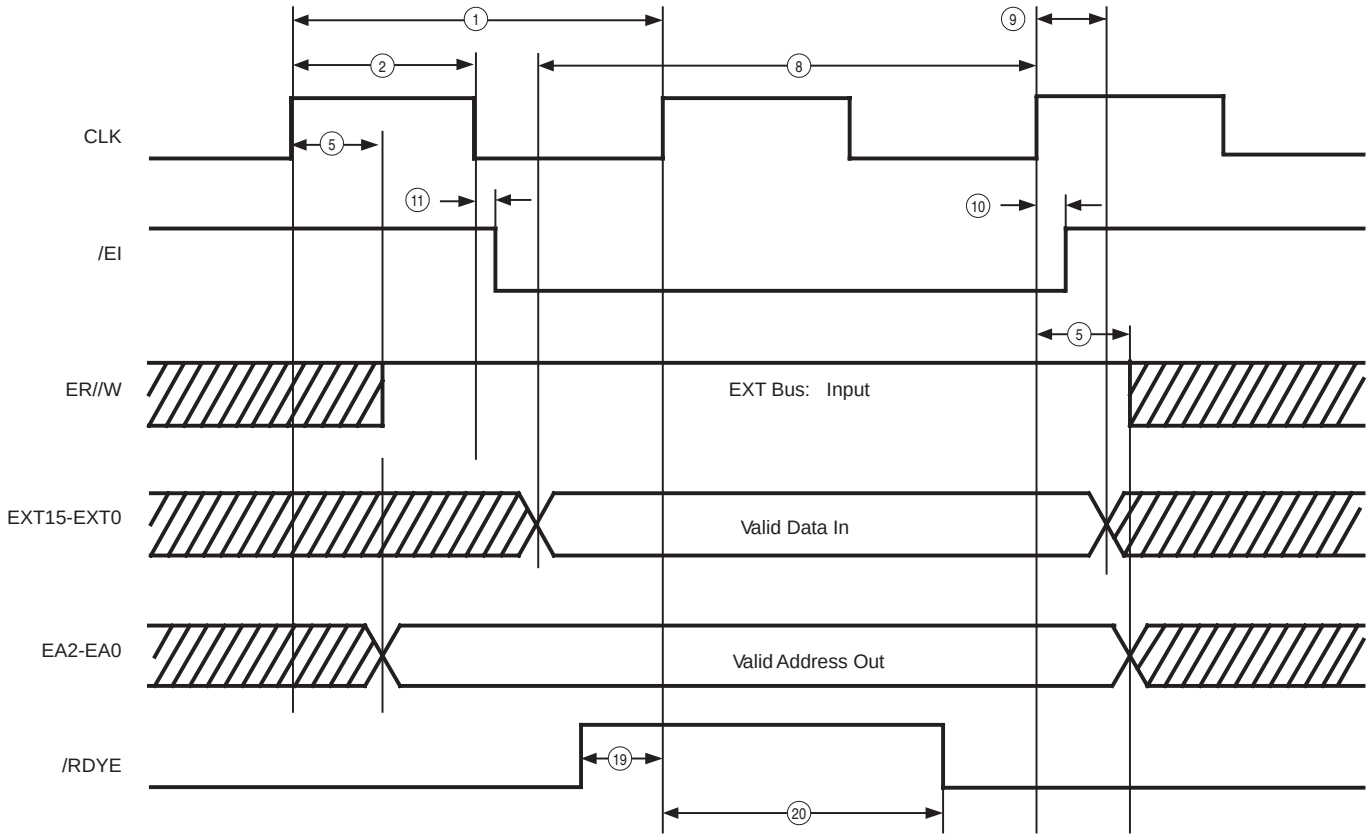


Figure 18. Read From External Device Timing (/RDYE used to hold data one clock cycle)*

Note: * /RDYE is checked during rising edge of clock.

AC TIMING DIAGRAM

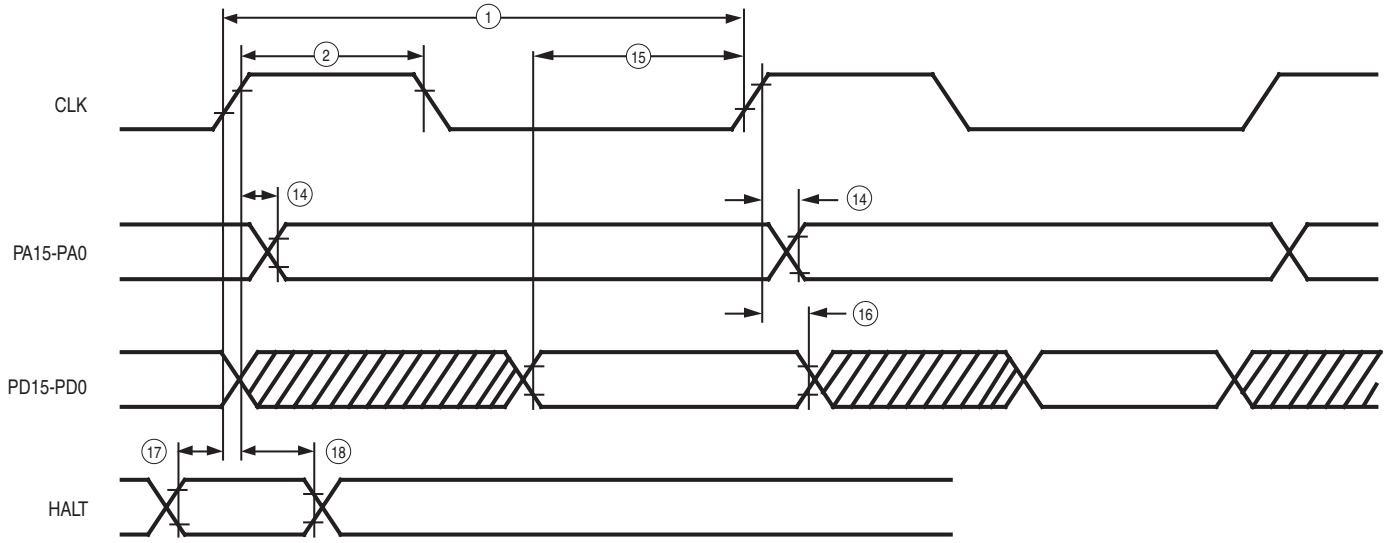


Figure 19. Memory Port Timing

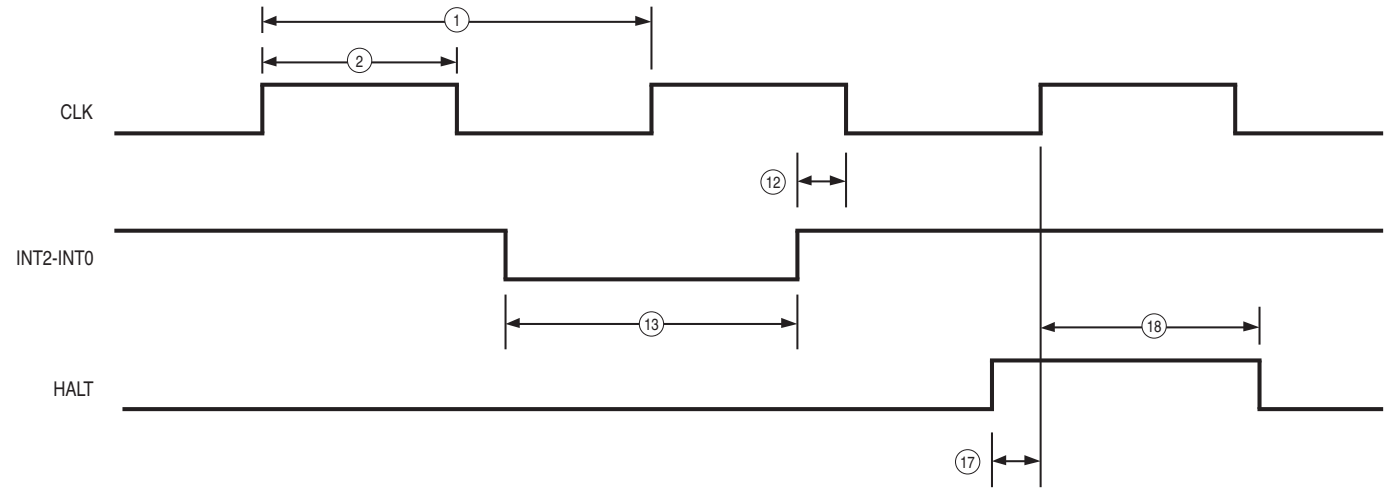
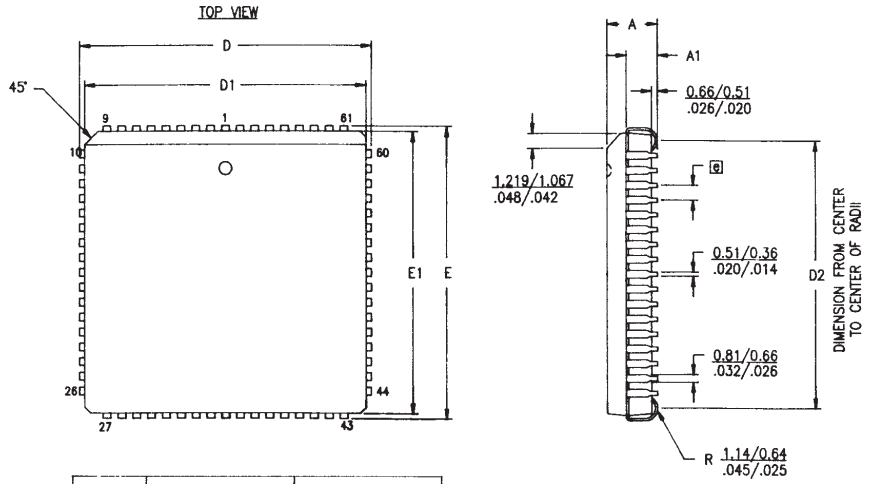


Figure 20. Interrupt and HALT Timing

PACKAGE INFORMATION



SYMBOL	MILLIMETER		INCH	
	MIN	MAX	MIN	MAX
A	4.32	4.57	.170	.180
A1	2.43	2.92	.095	.115
D/E	25.02	25.40	.985	1.000
D1/E1	24.13	24.33	.950	.958
D2	22.86	23.62	.900	.930
e	1.27 TYP		.050 TYP	

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

68-Pin PLCC Package Diagram

ORDERING INFORMATION

Z89C00

10 MHz
68-pin PLCC
Z89C0010VSC

15 MHz
68-pin PLCC
Z89C0015VSC

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

Package
V = Plastic Leaded Chip Carrier

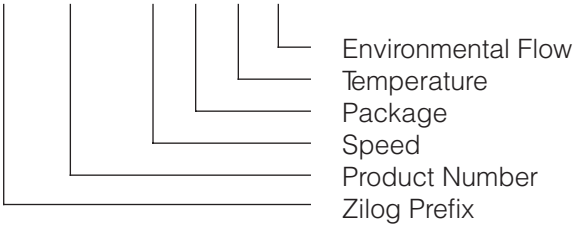
Temperature
S = 0°C to +70°C

Speeds
10 = 10 MHz
15 = 15 MHz

Environmental
C = Plastic Standard

Example:

Z 89C00 10 V S C is a Z89C00, 10 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow



© 1997 by Zilog, Inc. All rights reserved. No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Zilog, Inc. The information in this document is subject to change without notice. Devices sold by Zilog, Inc. are covered by warranty and patent indemnification provisions appearing in Zilog, Inc. Terms and Conditions of Sale only. Zilog, Inc. makes no warranty, express, statutory, implied or by description, regarding the information set forth herein or regarding the freedom of the described devices from intellectual property infringement. Zilog, Inc. makes no warranty of merchantability or fitness for any purpose. Zilog, Inc. shall not be responsible for any errors that may appear in this document. Zilog, Inc. makes no commitment to update or keep current the information contained in this document.

Zilog's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the customer and Zilog prior to use. Life support devices or systems are those which are intended for surgical implantation into the body, or which sustains life whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

Zilog, Inc. 210 East Hacienda Ave.
Campbell, CA 95008-6600
Telephone (408) 370-8000
Telex 910-338-7621
FAX 408 370-8056
Internet: http://www.zilog.com