



# Z89323/373/393

## 16-BIT DIGITAL SIGNAL PROCESSORS

### FEATURES

| Device | DSP ROM<br>(K Words) | OTP<br>(K Words) | DSP RAM<br>(Words) | Max Core<br>MIPS |
|--------|----------------------|------------------|--------------------|------------------|
| Z89323 | 8                    |                  | 512                | 20               |
| Z89373 |                      | 8                | 512                | 16               |
| Z89393 | 64*                  |                  | 512                | 20               |

\* External

- Operating Temperature Ranges:  
0°C to +70°C (Standard)  
-40°C to +85°C (Extended)

- 4.5- to 5.5-Volt Operating Range

### DSP Core

- 20 MIPS @ 20 MHz, 16-Bit Fixed Point DSP
- 50 ns Instruction Cycle Time
- Single-Cycle Multiply and ALU Operations
- Two Internal Data Buses and Address Generators
- Six Register Address Pointers
- Optimized Instruction Set (30 Instructions)

| Package<br>Device | 44-Pin<br>PLCC | 68-Pin<br>PLCC | 44-Pin<br>QFP | 80-Pin<br>QFP | 100-Pin<br>QFP |
|-------------------|----------------|----------------|---------------|---------------|----------------|
| Z89323            | ✓              | ✓              | ✓             | ✓             |                |
| Z89373            | ✓              | ✓              | ✓             | ✓             |                |
| Z89393            |                |                |               |               | ✓              |

### On-Board Peripherals

- 4-Channel, 8-Bit Analog to Digital Converter (A/D)
- On-Board Serial Peripheral Interface (SPI)
- Up to 40 Bits of Programmable I/O
- Two Channels of Programmable Pulse Width Modulators (PWM)
- Three General-Purpose Timer/Counters
- Two Watch-Dog Timers (WDT)
- Programmable PLL
- Three Vectored Interrupts Servicing Eight Interrupt Sources
- Power-Down and Power-On Reset

### GENERAL DESCRIPTION

The Z89323/373/393 DSP family of products builds on Zilog's first generation Z893XX DSP core, integrating several peripherals especially well suited for cost-effective voice, telephony, and control applications.

These DSP devices feature a modified Harvard architecture supported by one program bus and two on-chip data buses. This bus structure is supported by two address generators and six register pointers to ensure that the 20 MIPS DSP CPU is continually active.

The Z893X3 DSP family is designed to provide a complete DSP and control system on a single chip. By integrating

various peripherals, such as a high-speed 4-channel, 8-bit A/D, an SPI, three timers with PWM and WDT support, the Z893X3 family provides a compact system solution and reduces overall system cost.

To support a wide variety of development needs, the Z893X3 DSP product family features the cost-effective Z89323 with 8 Kwords of on-chip ROM, and the Z89373, a 16-MIPS OTP version of the Z89323, ideal for prototypes and early production builds. For systems requiring more than 8 Kwords of program memory, the Z89393 device can address up to 64 Kwords of external program memory.

**GENERAL DESCRIPTION** (Continued)

The Z893X3 DSP family is 100 percent source and object-code compatible with the existing Z89321/371/391 devices, providing users, who can benefit from increased integration and reduced system cost, an easy migration path from one DSP product to the next.

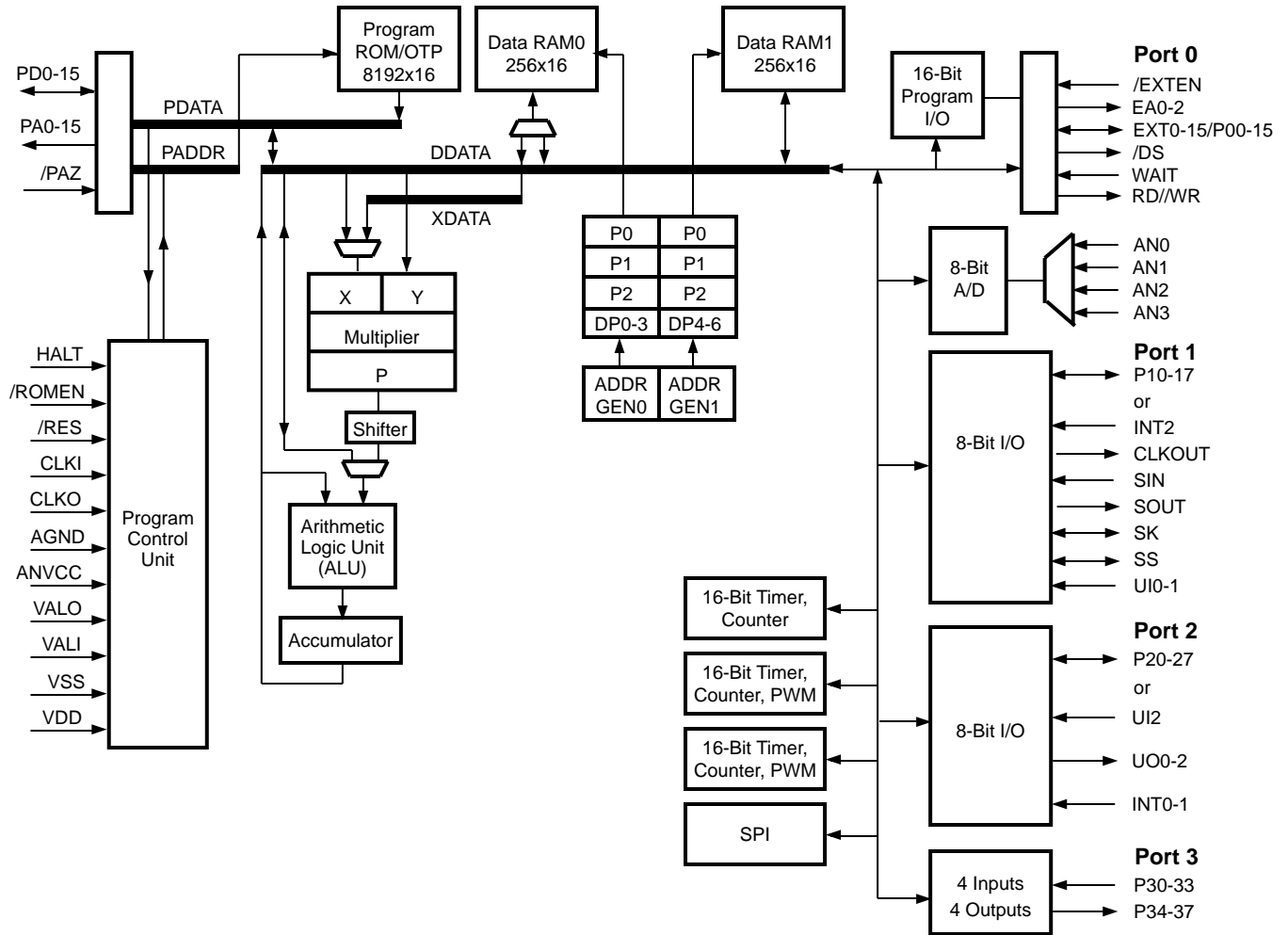
Throughout this specification, references to the Z89323 device applies equally to the Z89373 and Z89393, unless otherwise specified.

**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 <sub>CC</sub> | V <sub>DD</sub> |
| Ground     | GND             | V <sub>SS</sub> |



**Figure 1. Z893X3 Functional Block Diagram**

PIN DESCRIPTION

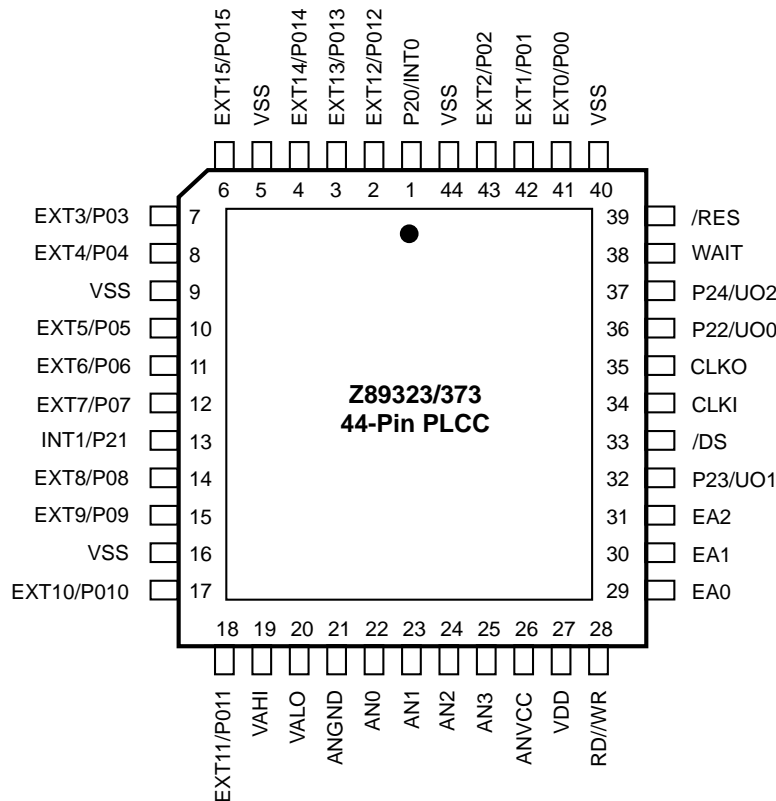


Figure 2. 44-Pin PLCC Z89323/373 Pin Configuration

Table 1. 44-Pin PLCC Z89323/373 Pin Description

| No. | Symbol          | Function          | Direction | No. | Symbol          | Function           | Direction |
|-----|-----------------|-------------------|-----------|-----|-----------------|--------------------|-----------|
| 1   | P20/INT0        | Port20/Interrupt0 | In/Output | 23  | AN1             | A/DInput1          | Input     |
| 2   | EXT12/P012      | ExtData12/Port012 | In/Output | 24  | AN2             | A/DInput2          | Input     |
| 3   | EXT13/P013      | ExtData13/Port013 | In/Output | 25  | AN3             | A/DInput3          | Input     |
| 4   | EXT14/P014      | ExtData14/Port014 | In/Output | 26  | ANVCC           | AnalogPower        | Input     |
| 5   | V <sub>SS</sub> | Ground            |           | 27  | V <sub>DD</sub> | Power              |           |
| 6   | EXT15/P015      | ExtData15/Port015 | In/Output | 28  | RD/WR           | R/WExternalBus     | Output    |
| 7   | EXT3/P03        | ExtData3/Port03   | In/Output | 29  | EA0             | ExtAddress0        | Output    |
| 8   | EXT4/P04        | ExtData4/Port04   | In/Output | 30  | EA1             | ExtAddress1        | Output    |
| 9   | V <sub>SS</sub> | Ground            |           | 31  | EA2             | ExtAddress2        | Output    |
| 10  | EXT5/P05        | ExtData5/Port05   | In/Output | 32  | P23/UO1         | Port23/UserOutput1 | In/Output |
| 11  | EXT6/P06        | ExtData6/Port06   | In/Output | 33  | /DS             | ExtDataStrobe      | Output    |
| 12  | EXT7/P07        | ExtData7/Port07   | In/Output | 34  | CLKI            | Clock/CrystalIn    | Input     |
| 13  | P21/INT1        | Port21/Interrupt1 | In/Output | 35  | CLKO            | Clock/CrystalOut   | Input     |
| 14  | EXT8/P08        | ExtData8/Port08   | In/Output | 36  | P22/UO0         | Port22/UserOutput0 | In/Output |
| 15  | EXT9/P09        | ExtData9/Port09   | In/Output | 37  | P24/UO2         | Port24/UserOutput2 | In/Output |
| 16  | V <sub>SS</sub> | Ground            |           | 38  | WAIT            | WaitforExt         | Input     |
| 17  | EXT10/P010      | ExtData10/Port010 | In/Output | 39  | /RES            | Reset              | Input     |
| 18  | EXT11/P011      | ExtData11/Port011 | In/Output | 40  | V <sub>SS</sub> | Ground             |           |
| 19  | VAHI            | AnalogHighRef.    | Input     | 41  | EXT0/P00        | ExtData0/Port00    | In/Output |
| 20  | VALO            | AnalogLowRef.     | Input     | 42  | EXT1/P01        | ExtData1/Port01    | In/Output |
| 21  | ANGND           | AnalogGround      | Input     | 43  | EXT2/P02        | ExtData2/Port02    | In/Output |
| 22  | AN0             | A/DInput0         | Input     | 44  | V <sub>SS</sub> | Ground             |           |

PIN DESCRIPTION (Continued)

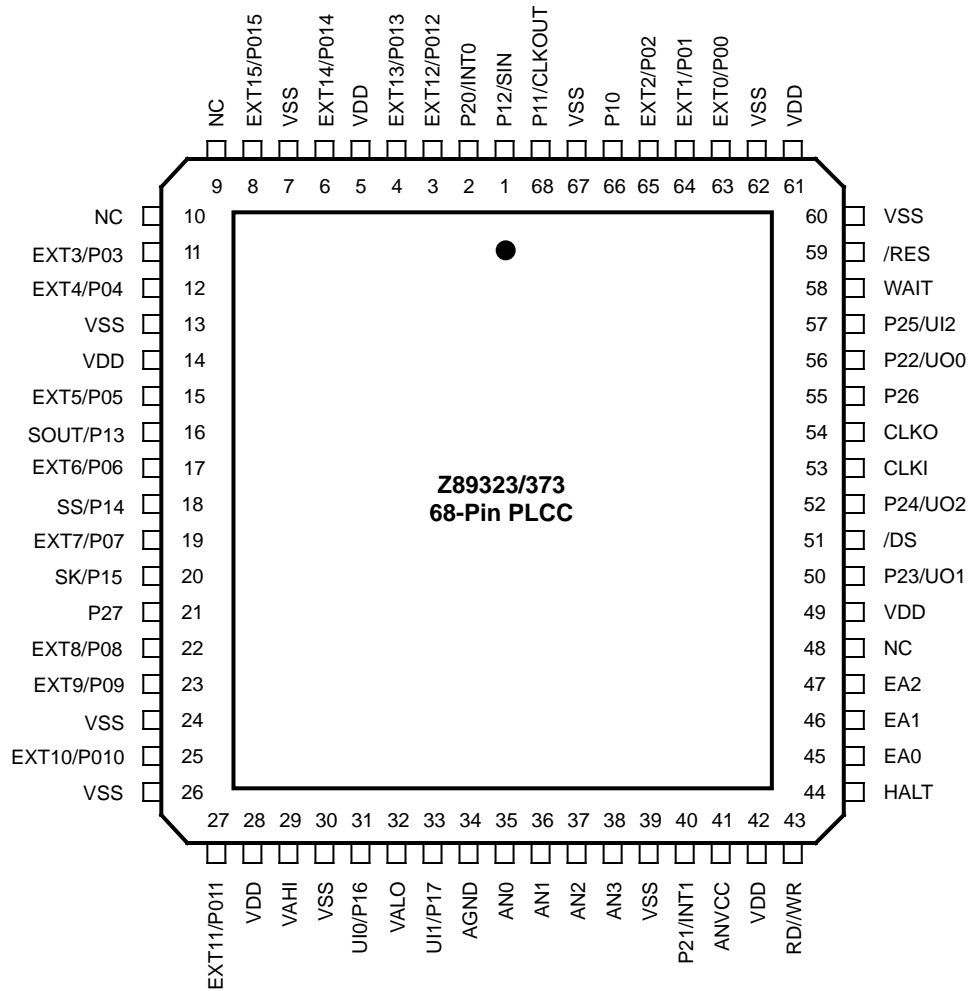


Figure 3. 68-Pin PLCC Z89323/373 Pin Configuration

Table 2. 68-Pin PLCC Z89323/373 Pin Description

| No. | Symbol          | Function            | Direction | No. | Symbol          | Function           | Direction |
|-----|-----------------|---------------------|-----------|-----|-----------------|--------------------|-----------|
| 1   | P12/SIN         | Port12/SerialInput  | In/Output | 35  | AN0             | A/DInput0          | Input     |
| 2   | P20/INT0        | Port20/Interrupt0   | In/Output | 36  | AN1             | A/DInput1          | Input     |
| 3   | EXT12/P012      | ExtData12/Port012   | In/Output | 37  | AN2             | A/DInput2          | Input     |
| 4   | EXT13/P013      | ExtData13/Port013   | In/Output | 38  | AN3             | A/DInput3          | Input     |
| 5   | VDD             | Power               |           | 39  | V <sub>SS</sub> | Ground             |           |
| 6   | EXT14/P014      | ExtData14/Port014   | In/Output | 40  | P21/INT1        | Port21/Interrupt1  | In/Output |
| 7   | V <sub>SS</sub> | Ground              |           | 41  | ANMCC           | AnalogPower        | Input     |
| 8   | EXT15/P015      | ExtData15/Port015   | In/Output | 42  | V <sub>DD</sub> | Power              | Input     |
| 9   | NC              | NoConnection        |           | 43  | RD/WR           | R/WExternalBus     | Output    |
| 10  | NC              | NoConnection        |           | 44  | HALT            | HaltExecution      | Input     |
| 11  | EXT3/P03        | ExtData3/Port03     | In/Output | 45  | EA0             | ExtAddress0        | Output    |
| 12  | EXT4/P04        | ExtData4/Port04     | In/Output | 46  | EA1             | ExtAddress1        | Output    |
| 13  | V <sub>SS</sub> | Ground              |           | 47  | EA2             | ExtAddress2        | Output    |
| 14  | V <sub>DD</sub> | Power               |           | 48  | NC              | NoConnection       |           |
| 15  | EXT5/P05        | ExtData5/Port05     | In/Output | 49  | V <sub>DD</sub> | Power              |           |
| 16  | P13/SOUT        | Port13/SerialOutput | In/Output | 50  | P23/UO1         | Port23/UserOutput1 | In/Output |
| 17  | EXT6/P06        | ExtData6/Port06     | In/Output | 51  | /DS             | ExtDataStrobe      | Output    |
| 18  | P14/SS          | Port14/SerialSelect | In/Output | 52  | P24/UO2         | Port24/UserOutput2 | In/Output |
| 19  | EXT7/P07        | ExtData7/Port07     | In/Output | 53  | CLKI            | Clock/CrystalIn    | Input     |
| 20  | P15/SK          | Port15/SerialClock  | In/Output | 54  | CLKO            | Clock/CrystalOut   | Input     |
| 21  | P27             | Port27              | In/Output | 55  | P26             | Port26             | In/Output |
| 22  | EXT8/P08        | ExtData8/Port08     | In/Output | 56  | P22/UO0         | Port22/UserOutput0 | In/Output |
| 23  | EXT9/P09        | ExtData9/Port09     | In/Output | 57  | P25/UI2         | Port25/UserInput2  | In/Output |
| 24  | V <sub>SS</sub> | Ground              |           | 58  | WAIT            | WaitforExt         | Input     |
| 25  | EXT10/P010      | ExtData10/Port010   | In/Output | 59  | RES             | Reset              | Input     |
| 26  | V <sub>SS</sub> | Ground              |           | 60  | V <sub>SS</sub> | Ground             |           |
| 27  | EXT11/P011      | ExtData11/Port011   | In/Output | 61  | V <sub>DD</sub> | Power              |           |
| 28  | V <sub>DD</sub> | Power               |           | 62  | V <sub>SS</sub> | Ground             |           |
| 29  | VAHI            | AnalogHighRef.      | Input     | 63  | EXT0/P00        | ExtData0/Port00    | In/Output |
| 30  | V <sub>SS</sub> | Ground              |           | 64  | EXT1/P01        | ExtData1/Port01    | In/Output |
| 31  | P16/UI0         | Port16/UserInput0   | In/Output | 65  | EXT2/P02        | ExtData2/Port02    | In/Output |
| 32  | VALO            | AnalogLowRef.       | Input     | 66  | P10/INT2        | Port10/Interrupt2  | In/Output |
| 33  | P17/UI1         | Port17/UserInput1   | In/Output | 67  | V <sub>SS</sub> | Ground             |           |
| 34  | ANGND           | AnalogGround        | Input     | 68  | P11/CLKOUT      | Port11/ClockOutput | In/Output |

PIN DESCRIPTION (Continued)

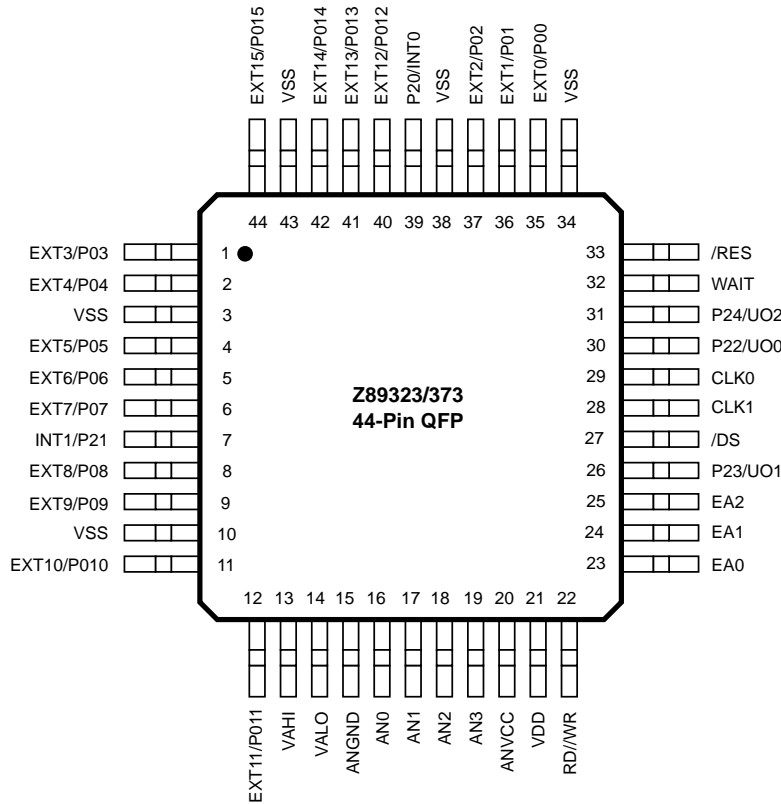


Figure 4. 44-Pin QFP Z89323/373 Pin Configuration

Table 3. 44-Pin QFP Z89323/373 Pin Description

| No. | Symbol          | Function          | Direction | No. | Symbol          | Function           | Direction |
|-----|-----------------|-------------------|-----------|-----|-----------------|--------------------|-----------|
| 1   | EXT3/P03        | ExtData3/Port03   | In/Output | 23  | EA0             | ExtAddress0        | Output    |
| 2   | EXT4/P04        | ExtData4/Port04   | In/Output | 24  | EA1             | ExtAddress1        | Output    |
| 3   | V <sub>SS</sub> | Ground            |           | 25  | EA2             | ExtAddress2        | Output    |
| 4   | EXT5/P05        | ExtData5/Port05   | In/Output | 26  | P23/UO1         | Port23/UserOutput1 | In/Output |
| 5   | EXT6/P06        | ExtData6/Port06   | In/Output | 27  | /DS             | ExtDataStrobe      | Output    |
| 6   | EXT7/P07        | ExtData7/Port07   | In/Output | 28  | CLK1            | Clock/CrystalIn    | Input     |
| 7   | P21/INT1        | Port21/Interrupt1 | In/Output | 29  | CLK0            | Clock/CrystalOut   | Input     |
| 8   | EXT8/P08        | ExtData8/Port08   | In/Output | 30  | P22/UO0         | Port22/UserOutput0 | In/Output |
| 9   | EXT9/P09        | ExtData9/Port09   | In/Output | 31  | P24/UO2         | Port24/UserOutput2 | In/Output |
| 10  | V <sub>SS</sub> | Ground            |           | 32  | WAIT            | WaitforExt         | Input     |
| 11  | EXT10/P010      | ExtData10/Port010 | In/Output | 33  | /RES            | Reset              | Input     |
| 12  | EXT11/P011      | ExtData11/Port011 | In/Output | 34  | V <sub>SS</sub> | Ground             |           |
| 13  | VAH             | AnalogHighRef.    | Input     | 35  | EXT0/P00        | ExtData0/Port00    | In/Output |
| 14  | VALO            | AnalogLowRef.     | Input     | 36  | EXT1/P01        | ExtData1/Port01    | In/Output |
| 15  | ANGND           | AnalogGround      | Input     | 37  | EXT2/P02        | ExtData2/Port02    | In/Output |
| 16  | AN0             | A/DInput0         | Input     | 38  | V <sub>SS</sub> | Ground             |           |
| 17  | AN1             | A/DInput1         | Input     | 39  | P20/INT0        | Port20/Interrupt0  | In/Output |
| 18  | AN2             | A/DInput2         | Input     | 40  | EXT12/P012      | ExtData12/Port012  | In/Output |
| 19  | AN3             | A/DInput3         | Input     | 41  | EXT13/P013      | ExtData13/Port013  | In/Output |
| 20  | ANVCC           | AnalogPower       | Input     | 42  | EXT14/P014      | ExtData14/Port014  | In/Output |
| 21  | V <sub>DD</sub> | Power             | Input     | 43  | V <sub>SS</sub> | Ground             |           |
| 22  | RD/WR           | R/WExternalBus    | Output    | 44  | EXT15/P015      | ExtData15/Port015  | In/Output |

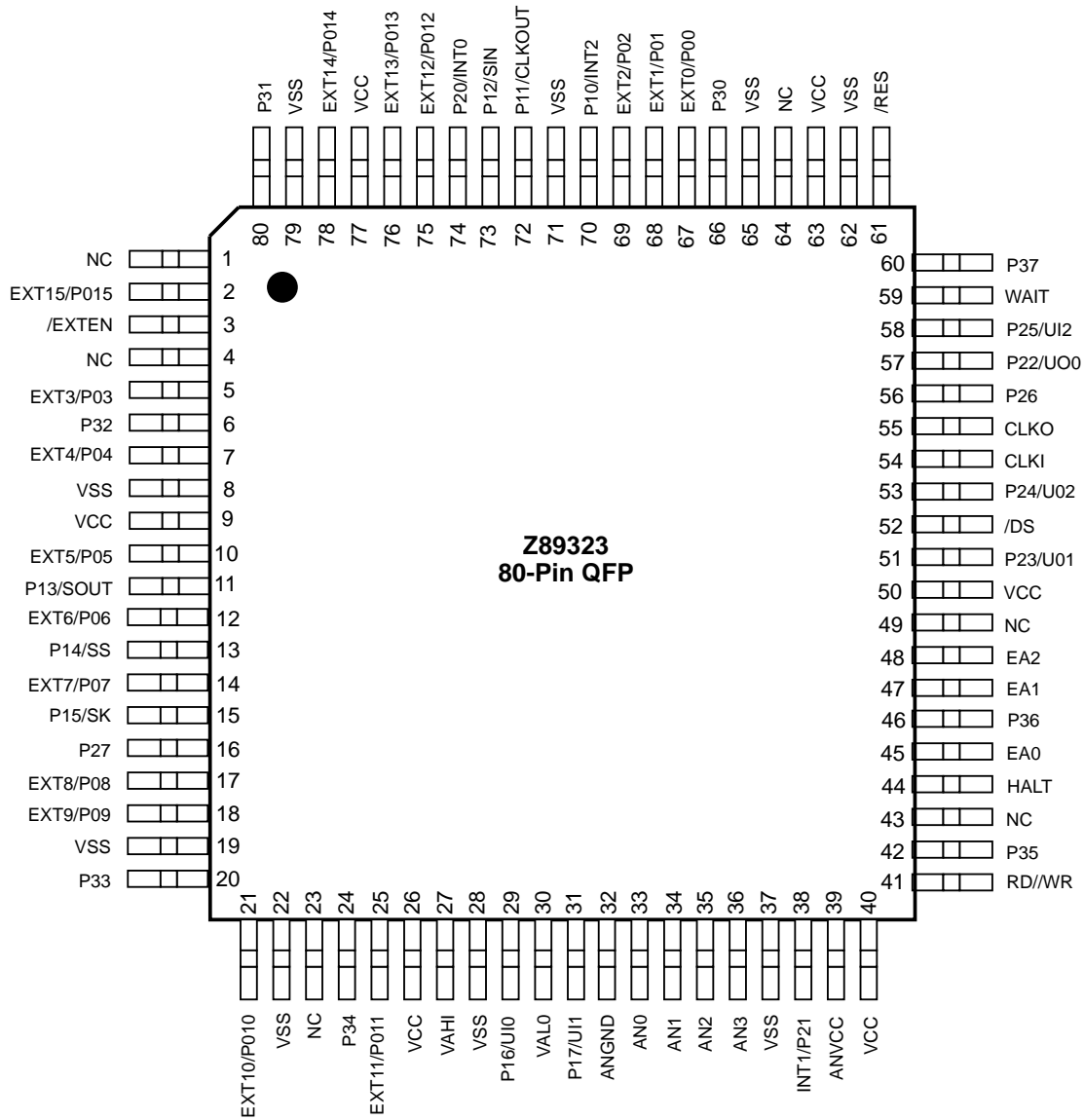


Figure 4a. 80-Pin QFP Z89323/373 Pin Configuration

**PIN DESCRIPTION** (Continued)

**Table 4a. 80-Pin QFP Z89323/373 Pin Description**

| No. | Symbol          | Function            | Direction | No. | Symbol          | Function           | Direction |
|-----|-----------------|---------------------|-----------|-----|-----------------|--------------------|-----------|
| 1   | NC              | NoConnection        |           | 41  | RD/MR           | R/WExternalBus     | Output    |
| 2   | EXT15/P015      | ExtData15/Port015   | In/Output | 42  | P35             | Port35             | Output    |
| 3   | /EXTEN          | ExtEnable           | Input     | 43  | NC              | NoConnection       |           |
| 4   | NC              | NoConnection        |           | 44  | HALT            | HaltExecution      | Input     |
| 5   | EXT3/P03        | ExtData3/Port03     | In/Output | 45  | EA0             | ExtAddress0        | Output    |
| 6   | P32             | Port32              | Input     | 46  | P36             | Port36             | Output    |
| 7   | EXT4/P04        | ExtData4/Port04     | In/Output | 47  | EA1             | ExtAddress1        | Output    |
| 8   | V <sub>SS</sub> | Ground              |           | 48  | EA2             | ExtAddress2        | Output    |
| 9   | V <sub>DD</sub> | Power               |           | 49  | NC              | NoConnection       |           |
| 10  | EXT5/P05        | ExtData5/Port05     | In/Output | 50  | V <sub>DD</sub> | Power              |           |
| 11  | P13/SOUT        | Port13/SerialOutput | In/Output | 51  | P23/UO1         | Port23/UserOutput1 | In/Output |
| 12  | EXT6/P06        | ExtData6/Port06     | In/Output | 52  | /DS             | ExtDataStrobe      | Output    |
| 13  | P14/SS          | Port14/SerialSelect | In/Output | 53  | P24/UO2         | Port24/UserOutput2 | In/Output |
| 14  | EXT7/P07        | ExtData7/Port07     | In/Output | 54  | CLKI            | Clock/CrystalIn    | Input     |
| 15  | P15/SK          | Port15/SerialClock  | In/Output | 55  | CLKO            | Clock/CrystalOut   | Input     |
| 16  | P27             | Port27              | In/Output | 56  | P26             | Port26             | In/Output |
| 17  | EXT8/P08        | ExtData8/Port08     | In/Output | 57  | P22/UO0         | Port22/UserOutput0 | In/Output |
| 18  | EXT9/P09        | ExtData9/Port09     | In/Output | 58  | P25/UI2         | Port25/UserInput2  | In/Output |
| 19  | V <sub>SS</sub> | Ground              |           | 59  | WAIT            | WaitforExt         | Input     |
| 20  | P33             | Port33              | Input     | 60  | P37             | Port37             | Output    |
| 21  | EXT10/P010      | ExtData10/Port010   | In/Output | 61  | /RES            | Reset              | Input     |
| 22  | V <sub>SS</sub> | Ground              |           | 62  | V <sub>SS</sub> | Ground             |           |
| 23  | NC              | NoConnection        |           | 63  | V <sub>DD</sub> | Power              |           |
| 24  | P34             | Port34              | Output    | 64  | NC              | NoConnection       |           |
| 25  | EXT11/P011      | ExtData11/Port011   | In/Output | 65  | V <sub>SS</sub> | Ground             |           |
| 26  | V <sub>DD</sub> | Power               |           | 66  | P30             | Port30             | Input     |
| 27  | VAHI            | AnalogHighRef.      | Input     | 67  | EXT0/P00        | ExtData0/Port00    | In/Output |
| 28  | V <sub>SS</sub> | Ground              |           | 68  | EXT1/P01        | ExtData1/Port01    | In/Output |
| 29  | P16/UI0         | Port16/UserInput0   | In/Output | 69  | EXT2/P02        | ExtData2/Port02    | In/Output |
| 30  | VAL0            | AnalogLowRef.       | Input     | 70  | P10/INT2        | Port10/Interrupt2  | In/Output |
| 31  | P17/UI1         | Port17/UserInput1   | In/Output | 71  | V <sub>SS</sub> | Ground             |           |
| 32  | ANGND           | AnalogGround        | Input     | 72  | P11/CLKOUT      | Port11/ClockOutput | In/Output |
| 33  | AN0             | A/DInput0           | Input     | 73  | P12/SIN         | Port12/SerialInput | In/Output |
| 34  | AN1             | A/DInput1           | Input     | 74  | P20/INT0        | Port20/Interrupt0  | In/Output |
| 35  | AN2             | A/DInput2           | Input     | 75  | EXT12/P012      | ExtData12/Port012  | In/Output |
| 36  | AN3             | A/DInput3           | Input     | 76  | EXT13/P013      | ExtData13/Port013  | In/Output |
| 37  | V <sub>SS</sub> | Ground              |           | 77  | V <sub>DD</sub> | Power              |           |
| 38  | P21/INT1        | Port21/Interrupt1   | In/Output | 78  | EXT14/P014      | ExtData14/Port014  | In/Output |
| 39  | ANVCC           | AnalogPower         | Input     | 79  | V <sub>SS</sub> | Ground             |           |
| 40  | V <sub>DD</sub> | Power               | Input     | 80  | P31             | Port31             | Input     |



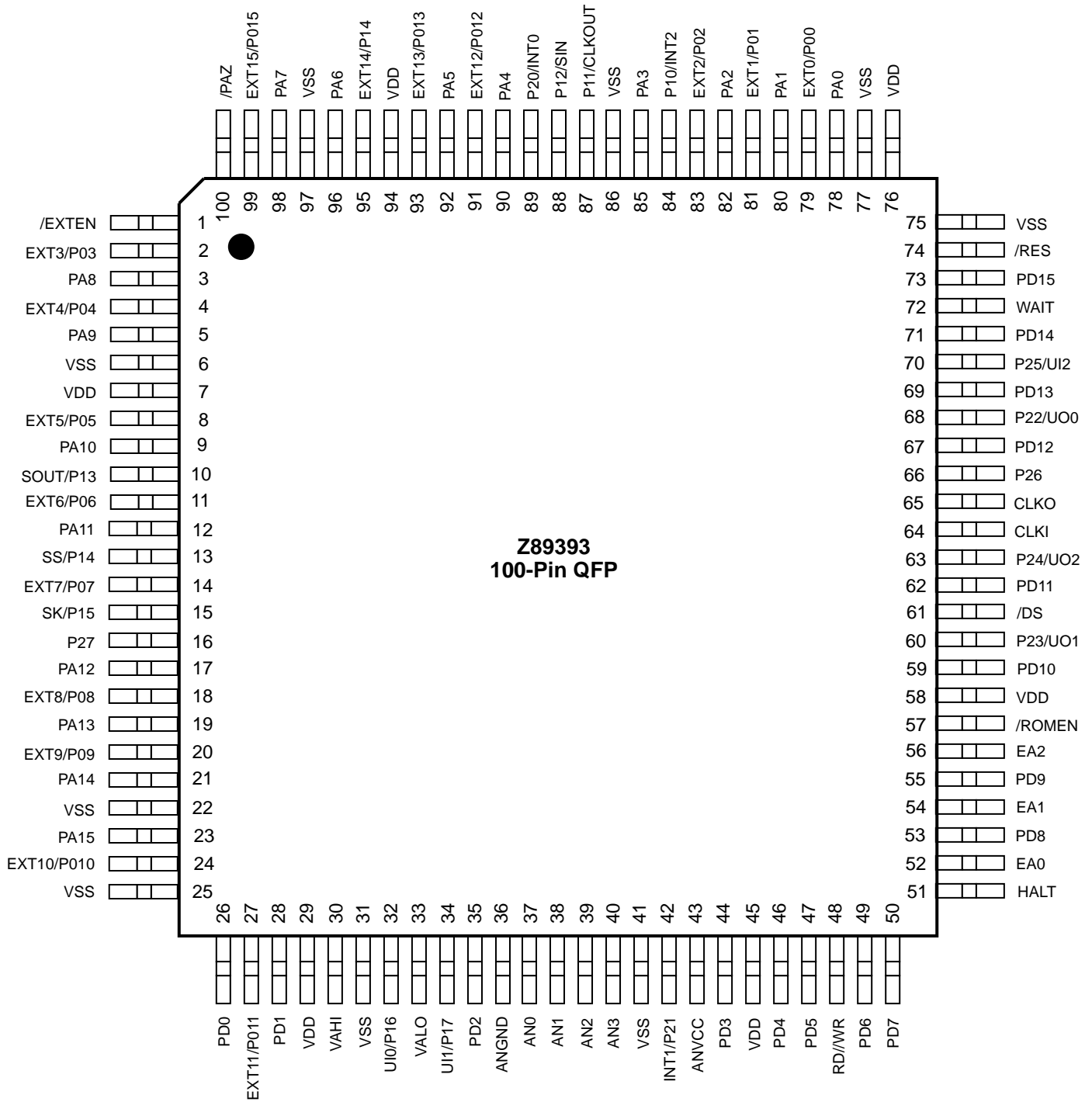


Figure 5. 100-Pin QFP Z89393 Pin Configuration

**PIN DESCRIPTION** (Continued)

**Table 4. 100-Pin QFP Z89393 Pin Description**

| No. | Symbol          | Function            | Direction | No. | Symbol          | Function            | Direction |
|-----|-----------------|---------------------|-----------|-----|-----------------|---------------------|-----------|
| 1   | EXTEN           | EXTEnable           | Input     | 51  | HALT            | HaltExecution       | Input     |
| 2   | EXT3P03         | ExtData3/Port03     | In/Output | 52  | EA0             | ExtAddress0         | Output    |
| 3   | PA8             | ProgramAddress8     | Output    | 53  | FD8             | ProgramData8        | Input     |
| 4   | EXT4P04         | ExtData4/Port04     | In/Output | 54  | EA1             | ExtAddress1         | Output    |
| 5   | PA9             | ProgramAddress9     | Output    | 55  | FD9             | ProgramData9        | Input     |
| 6   | V <sub>SS</sub> | Ground              |           | 56  | EA2             | ExtAddress2         | Output    |
| 7   | V <sub>DD</sub> | Power               |           | 57  | /ROMEN          | ROMEnable           | Input     |
| 8   | EXT5P05         | ExtData5/Port05     | In/Output | 58  | V <sub>DD</sub> | Power               |           |
| 9   | PA10            | ProgramAddress10    | Output    | 59  | FD10            | ProgramData10       | Input     |
| 10  | P13SCOUT        | Port13/SerialOutput | In/Output | 60  | P23UO1          | Port23/UserOutput1  | In/Output |
| 11  | EXT6P06         | ExtData6/Port06     | In/Output | 61  | /DS             | ExtDataStrobe       | Output    |
| 12  | PA11            | ProgramAddress11    | Output    | 62  | FD11            | ProgramData11       | Input     |
| 13  | P14SS           | Port14/SerialSelect | In/Output | 63  | P24UO2          | Port24/UserOutput2  | In/Output |
| 14  | EXT7P07         | ExtData7/Port07     | In/Output | 64  | CLKI            | Clock/CrystalIn     | Input     |
| 15  | P15SK           | Port15/SerialClock  | In/Output | 65  | CLKO            | Clock/CrystalOut    | Input     |
| 16  | P27             | Port27              | In/Output | 66  | P26             | Port26              | In/Output |
| 17  | PA12            | ProgramAddress12    | Output    | 67  | FD12            | ProgramData12       | Input     |
| 18  | EXT8P08         | ExtData8/Port08     | In/Output | 68  | P22UO0          | Port22/UserOutput0  | In/Output |
| 19  | PA13            | ProgramAddress13    | Output    | 69  | FD13            | ProgramData13       | Input     |
| 20  | EXT9P09         | ExtData9/Port09     | In/Output | 70  | P25UI2          | Port25/UserInput2   | In/Output |
| 21  | PA14            | ProgramAddress14    | Output    | 71  | FD14            | ProgramData14       | Input     |
| 22  | V <sub>SS</sub> | Ground              |           | 72  | WAIT            | WaitforExt          | Input     |
| 23  | PA15            | ProgramAddress15    | Output    | 73  | FD15            | ProgramData15       | Input     |
| 24  | EXT10P010       | ExtData10/Port010   | In/Output | 74  | /RES            | Reset               | Input     |
| 25  | V <sub>SS</sub> | Ground              |           | 75  | V <sub>SS</sub> | Ground              |           |
| 26  | FD0             | ProgramData0        | Input     | 76  | V <sub>DD</sub> | Power               |           |
| 27  | EXT11P011       | ExtData11/Port011   | In/Output | 77  | V <sub>SS</sub> | Ground              |           |
| 28  | FD1             | ProgramData1        | Input     | 78  | PA0             | ProgramAddress0     | Output    |
| 29  | V <sub>DD</sub> | Power               |           | 79  | EXT0P00         | ExtData0/Port00     | In/Output |
| 30  | VAH             | AnalogHighRef.      | Input     | 80  | PA1             | ProgramAddress1     | Output    |
| 31  | V <sub>SS</sub> | Ground              |           | 81  | EXT1P01         | ExtData1/Port01     | In/Output |
| 32  | P16UI0          | Port16/UserInput0   | In/Output | 82  | PA2             | ProgramAddress2     | Output    |
| 33  | VALO            | AnalogLowRef.       | Input     | 83  | EXT2P02         | ExtData2/Port02     | In/Output |
| 34  | P17UI1          | Port17/UserInput1   | In/Output | 84  | P10/INT2        | Port10/Interrupt2   | In/Output |
| 35  | FD2             | ProgramData2        | Input     | 85  | PA3             | ProgramAddress3     | Output    |
| 36  | ANGND           | AnalogGround        | Input     | 86  | V <sub>SS</sub> | Ground              |           |
| 37  | AN0             | A/DInput0           | Input     | 87  | P11/CLKOUT      | Port11/ClockOutput  | In/Output |
| 38  | AN1             | A/DInput1           | Input     | 88  | P12/SIN         | Port12/SerialInput  | In/Output |
| 39  | AN2             | A/DInput2           | Input     | 89  | P20/INT0        | Port20/Interrupt0   | In/Output |
| 40  | AN3             | A/DInput3           | Input     | 90  | PA4             | ProgramAddress4     | Output    |
| 41  | V <sub>SS</sub> | Ground              |           | 91  | EXT12P012       | ExtData12/Port012   | In/Output |
| 42  | P21/INT1        | Port21/Interrupt1   | In/Output | 92  | PA5             | ProgramAddress5     | Output    |
| 43  | ANVCC           | AnalogPower         | Input     | 93  | EXT13P013       | ExtData13/Port013   | In/Output |
| 44  | FD3             | ProgramData3        | Input     | 94  | V <sub>DD</sub> | Power               |           |
| 45  | V <sub>DD</sub> | Power               |           | 95  | EXT14P014       | ExtData14/Port014   | In/Output |
| 46  | FD4             | ProgramData4        | Input     | 96  | PA6             | ProgramAddress6     | Output    |
| 47  | FD5             | ProgramData5        | Input     | 97  | V <sub>SS</sub> | Ground              |           |
| 48  | RDMWR           | R/WExternalBus      | Output    | 98  | PA7             | ProgramAddress7     | Output    |
| 49  | FD6             | ProgramData6        | Input     | 99  | EXT15P015       | ExtData15/Port015   | In/Output |
| 50  | FD7             | ProgramData7        | Input     | 100 | /PAZ            | Tri-stateProgramBus | Input     |

## PIN FUNCTIONS

**CLKO-CLKI** *Clock* (output/input). These pins act as the clock circuit input and output.

**EXT15-EXT0** *External Data Bus* (input/output). These pins act as the data bus for user-defined outside registers, such as an ADC or DAC. The pins are normally tri-stated, except when the outside registers are specified as destination registers in the instructions. All the control signals exist to allow a read or a write through this bus. If user I/O Port 0 is enabled, these signals function as user Programmable I/O.

**RD//WR** *Read/Write Strobe* (output). This pin controls the data direction signal for the 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). These pins control the user-defined register address output (latched). 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. External Addresses are used internally by the processor if the ADC, bit I/O (Port 0- 2), or SPI are enabled. (See the banks allocation of the EXT registers in Tables 6 and 7.)

**/DS** *Data Strobe* (output). This pin control the data strobe signal for EXT-Bus. Data is read by the external peripheral on the rising edge of /DS. Data is also read by the processor on the rising edge of CK.

**HALT** *Halt State* (input). This pin controls Stop Execution. The CPU continuously executes NOPs and the program counter remains at the same value when this pin is held High. An interrupt request must be executed (enabled) to exit HALT mode. After the interrupt service routine, the program continues from the instruction after the HALT (active high).

**/INT0-/INT2** *Three Interrupts* (input, active on rising edge). These pins control interrupt requests 0-2. Interrupts are generated on the rising edge of the input signal. Interrupt vectors for the interrupt service starting address are stored in the following program memory locations:

| Device     | /INT0 | /INT1 | /INT2 |
|------------|-------|-------|-------|
| Z89323/373 | 1FFFH | 1FFEH | 1FFDH |
| Z89393     | FFFFH | FFFEH | FFFDH |

Priority is: INT2 = lowest, INT0 = highest. (**Note:** INT2 pin is not bonded out on the 44-pin QFP or PLCC packages.)

**/RES** *Reset* (input, active Low). This pin controls the asynchronous reset signal. The /RES signal must be kept Low for at least one clock cycle (clock output of the PLL block). The CPU pushes the contents of the Program Counter (PC) onto the stack and then fetches a new PC value from program memory address 0FFCH (or FFFCH for the Z89393) after the reset signal is released.

**WAIT** *WAIT State* (input). The wait signal is sampled at the rising edge of the clock with appropriate setup and hold times. The normal write cycle will continue when wait is inactive on a rising clock. A single wait-state can be generated internally by setting the appropriate bits in the wait state register (Bank 15/Ext 3) (active high).

**P00-P015** *Port 0* (input/output). These pins control Port 0 input and output when EXT I/F is not in use.

**P10-P17** *Port 1* (input/output). These pins are used for Port 1 programmable bit I/O when INT2, CLKOUT, SPI, or UI0-1 are not being used.

**P20-P27** *Port 2* (input/output). These pins control Port 2 input or output when UI2, UO0-2 or INT0-INT1 are not being used.

**P30-P37** *Port 3* Port3 (3:0) are four inputs and P3 (7:0) are four outputs.

**UI1-UI0** *Two Input Pins* (input). These general-purpose 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). These general-purpose output pins reflect the value of two bits in the status register S5 and S6. These bits have no special significance and may be used to output data by writing to the status register. **Note:** The user output value is the opposite of the status register content.

**SIN/SOUT.** When enabled, these pins control SPI input and output.

**AN0-AN3.** These pins are used for Analog-to-Digital converter input.

**ANGND and ANVCC.** Analog to Digital ground and power supply.

**PIN FUNCTIONS** (Continued)

**VAHI and VALO.** Analog to Digital reference voltages.

**/PAZ** *Tri-state Program Bus.* This pin enables the Program Address bus for emulation purposes.

**/EXTEN** *Ext Enable.* This pin enables Ext output continuously for emulation purposes.

**/ROMEN** *ROM Enable.* This pin selects internal or external Program Memory.

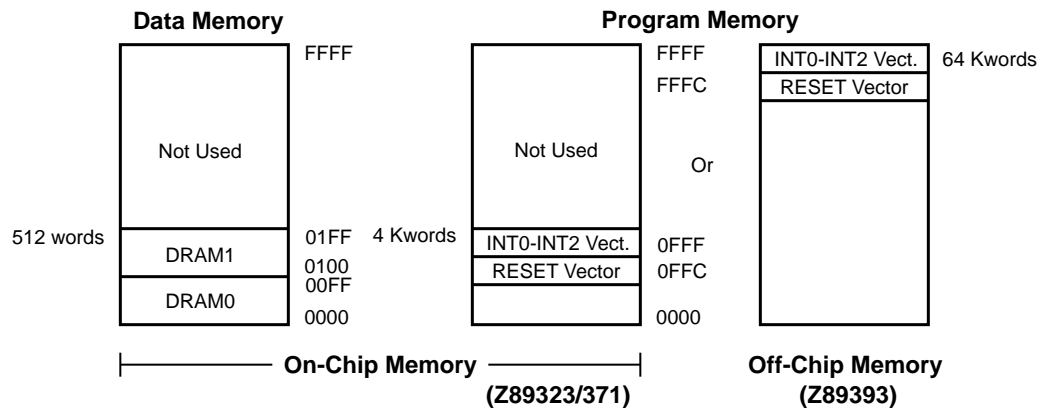
**ADDRESS SPACE**

**Program Memory.** Programs of up to 8 Kwords can be masked into internal ROM (OTP for Z89373). Four locations are dedicated to the vector address for the three interrupts (IFFDH-IFFFH) and the starting address following a Reset (IFFCH). Internal ROM is mapped from 0000H to IFFFFH, and the highest location for program is IFFBH.

**Internal Data RAM.** The Z89323 has an internal 512 x 16-bit word data RAM organized as two banks of 256 x 16-bit words each: RAM0 and RAM1. Each data RAM bank is addressed by three pointers: 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 modulo

addressing. Three addressing modes are available to access the Data RAM: register indirect, direct addressing, and short form direct. The contents of the RAM can be read to, or written from, 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 Z89323 has 19 internal registers and eight external registers and a secondary set of 15 peripheral control registers. Both external and internal registers are accessed in one machine cycle. The external registers are used to access the on-chip peripherals when they are enabled.



**Figure 6. Memory Map**

## REGISTERS

The internal registers of the Z89323/373/393 are defined below:

| Register | Register Definition                  |
|----------|--------------------------------------|
| P        | Output of Multiplier, 24-bit         |
| 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              |
| EXT0     |                                      |
| EXT1     |                                      |
| EXT2     |                                      |
| EXT3     |                                      |
| EXT4     |                                      |
| EXT5     |                                      |
| EXT6     |                                      |
| EXT7     |                                      |

See Table 6 and Table 7 for the different assignments of EXT7-EXT0 in the different banks.

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

**Note:**

\* These data pointers occupy the first four locations in RAM bank.

**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.

**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 is placed into the 16 MSBs 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 Program Memory.

**EXTn** are external registers (n = 0 to 7). There are eight 16-bit registers provided here for mapping external devices into the address space of the processor. Note that the actual register RAM does not exist on the chip, but would exist as part of the internal or external device, such as an ADC.

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

**Dn:b** refers to locations in RAM that can be used as a pointer to locations in program memory which is efficient for coefficient addressing. 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 register 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.

**SR** is the status register (Figure 8) which contains the ALU status and certain control bits (Table 5).

**Table 5. Status Register Bit Functions**

| 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 Arithmetically Shifted Right by three bits |
| S8 (OP)             | Overflow Protection                                   |
| S7 (IE)             | Interrupt Enable                                      |
| S6 (UO1)            | User Output 1   |
| S5 (UO0)            | User Output 0   |
| S4-S3               | “Short Form Direct” bits                              |
| S2-S0 (RPL)         | RAM Pointer Loop Size                                 |

REGISTERS (Continued)

The Status Register

The status register can always be read in its entirety. S15-S10 are set/reset by hardware and can only be read by software. S9-S0 control hardware looping and can be written by software (Table 8).

Table 8. 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       |

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 in Table 5. S7 enables interrupts. If S8 is set, the hardware clamps at maximum positive or negative values instead of overflowing. If S9 is set and a multiple/shift option is used, then the shifter shifts the result three bits right. This feature allows the data to be scaled and prevents overflows.

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.

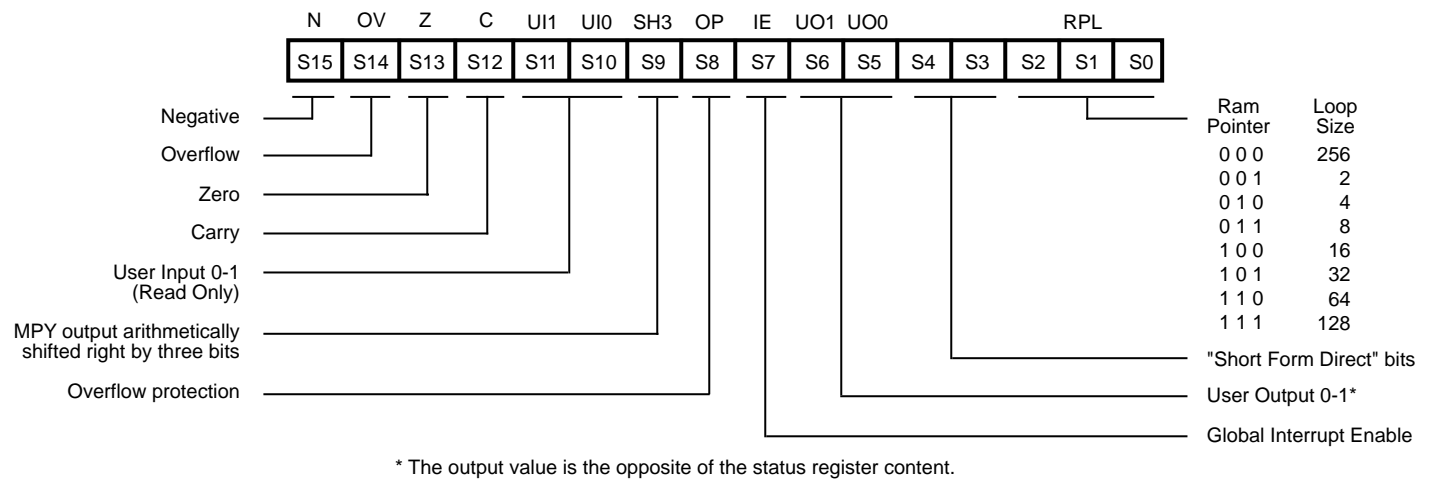


Figure 7. Status Register

## EXT Register Assignments

The EXT registers support is extended in the Z893X3 family: In addition to up to seven external registers, there are 28 internal registers on the EXT bus. There are 16 different pages of EXT registers. The same EXT7 register exist in all the pages and control of the bank switching is done via EXT7 register.

Banks 0 to 5 support different combinations of external registers and internal data registers. The user should use the bank that has the internal data registers and the number of external registers to support his application and to use this bank as a working bank to minimize the number of bank switching. Bank 5 has all the A/D registers. Banks 13 to 15 are control registers bank. These control registers are usually used only in the initialization routines.

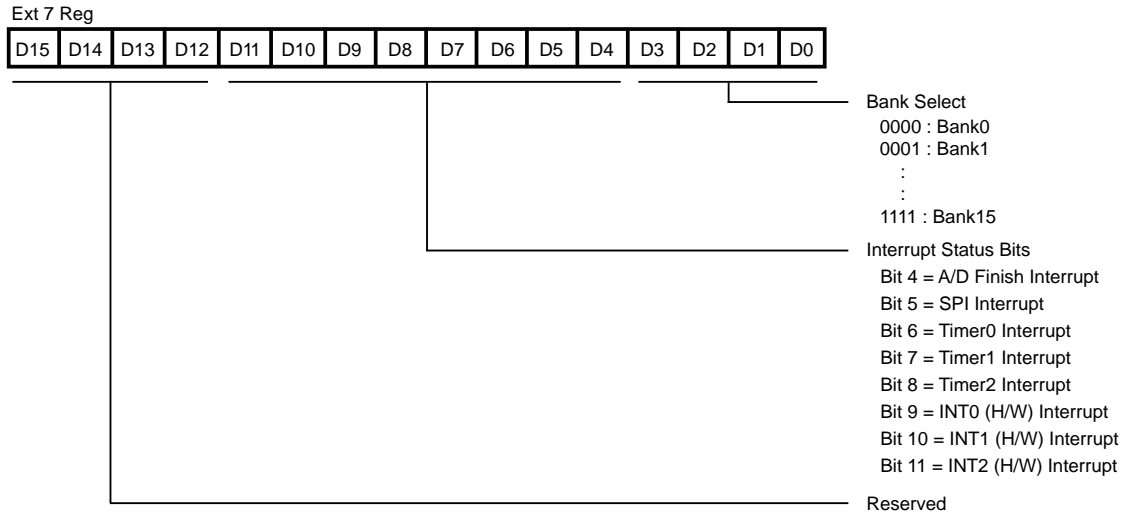
**Table 6. EXT Register Assignments Banks 0–4**

| EXT\Bank | 0               | 1               | 2               | 3               | 4               |
|----------|-----------------|-----------------|-----------------|-----------------|-----------------|
| EXT0     | Ext0-user       | Ext0-user       | Ext0-user       | Ext0-user       | Ext0-user       |
| EXT1     | Ext1-user       | Ext1-user       | Ext1-user       | Ext1-user       | Ext1-user       |
| EXT2     | Ext2-user       | Ext2-user       | Ext2-user       | Ext2-user       | Ext2-user       |
| EXT3     | SPI data        | Ext3-user       | Ext3-user       | SPI data        | Ext3-user       |
| EXT4     | Port0           | Port0           | Ext4-user       | Ext4-user       | Ext4-user       |
| EXT5     | Port1/Port2     | Port1/Port2     | Port3           | Ext5-user       | Ext5-user       |
| EXT6     | A/D_ch0         | A/D_ch1         | A/D_ch2         | A/D_ch3         | Ext6-user       |
| EXT7     | Bank/Int_status | Bank/Int_status | Bank/Int_status | Bank/Int_status | Bank/Int_status |

**Table 7. EXT Register Assignments Banks 6–15**

| EXT\Bank | 5               | 6-12            | 13               | 14               | 15              |
|----------|-----------------|-----------------|------------------|------------------|-----------------|
| EXT0     | A/D_ch1         |                 | A/D control      | Timer2 load      | P0 control      |
| EXT1     | A/D_ch2         |                 | Timer0 control   | Timer1 control   | P1 control      |
| EXT2     | A/D_ch3         |                 | Timer0 load      | Timer1 load      | P2 control      |
| EXT3     | SPI data        |                 | Timer0           | Timer1           | Wait State      |
| EXT4     | Port0           |                 | Timer0 pr. load  | Timer1 pr. load  | SPI control     |
| EXT5     | Port1/Port2     |                 | Timer0 prescaler | Timer1 prescaler | PLL control     |
| EXT6     | A/D_ch0         | A/D_ch0         | A/D_ch0          | A/D_ch0          | Int. Allocation |
| EXT7     | Bank/Int_status | Bank/Int_status | Bank/Int_status  | Bank/Int_status  | Bank/Int_status |

**EXT Register Assignments** (Continued)



**Figure 8. EXT7 Register Bit Assignment**

**Interrupt Status Bits**

When read, these bits provide interrupt information to identify the source for INT2, or when the DSP works in Pending Interrupt mode, to warn the DSP of pending interrupts. These bits also clear the interrupt status bits. Writing 1 will clear these bits.

**Wait-State Register**

The Wait-State Control Register enables insertion of Wait States when the DSP needs to access slow, inexpensive peripherals. This software-controlled register enables insertion of one Wait State when accessing EXT bus. (One Wait State gives 100 nsec access time instead of 50 nsec

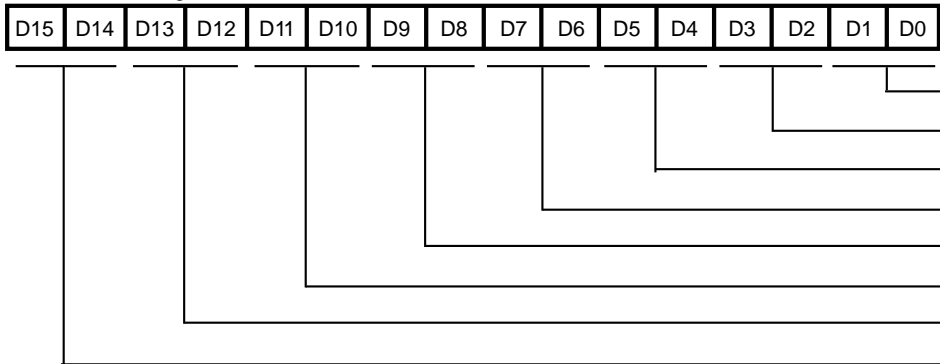
access time with a 20 MHz oscillator.) When more than one Wait State is needed, an input pin (WAIT) coupled with external logic can support more than one Wait State. The Wait-State Control Register enables mapping specific EXT register (from EXT0 to EXT6) and specific operation (read or write) to include insertion of one Wait State. EXT7 is always internal register, therefore no Wait State is needed for EXT7.

**Note:**

When the programmer switches banks it is important to change the Wait State mapping of the EXT registers to match the desired Wait State mapping of the new bank.



Bank15/EXT3 Reg



Bits 1 - 0 = Wait-State EXT0  
 Bits 3 - 2 = Wait-State EXT1  
 Bits 5 - 4 = Wait-State EXT2  
 Bits 7 - 6 = Wait-State EXT3  
 Bits 9 - 8 = Wait-State EXT4  
 Bits 11 - 10 = Wait-State EXT5  
 Bits 13 - 12 = Wait-State EXT6  
 Bit14 = Reserved  
 Bit 15 = Test Mode

0 Normal Operation (default)  
 1 Test Mode: Bits 6-5 of the Status Register drives, P23 and P22, respectively (VO0 and VO1).

Figure 8a. Bank 15/EXT3 Register

## FUNCTIONAL DESCRIPTION

### Analog to Digital Converter (ADC)

The ADC is an 8-bit half flash converter that uses two reference resistor ladders for its upper 4 bits (Most Significant Bits) and lower 4 bits (Least Significant Bits) conversion. Two reference voltage pins, VA (High) and VA (Low), are provided for external reference voltage supplies. During the sampling period from one of the four 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  $\mu$ s. (See Figure 9, ADC Architecture.)

The ADC control register is Bank 13/Ext 0. A conversion can be initiated in one of four ways: by writing to the A/D control register, INT1 input pin, Timer 2 or Timer 0 equal 0. These four are programmable 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 3 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  $\mu$ s conversion time for clock inputs from 4 MHz to 20 MHz. For example, with a 20 MHz crystal clock the prescaler should be programmed for divide by 40, which then gives a 2  $\mu$ s 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 during 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.

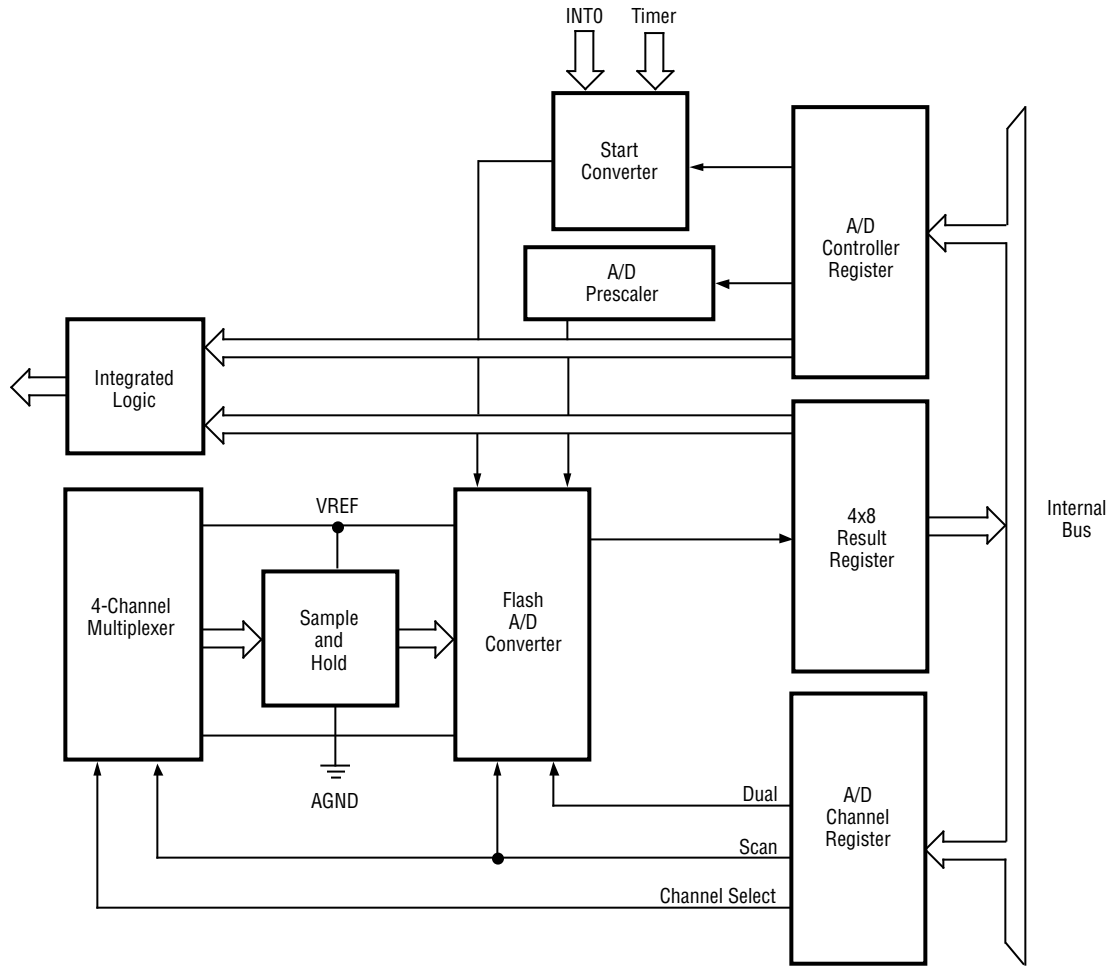
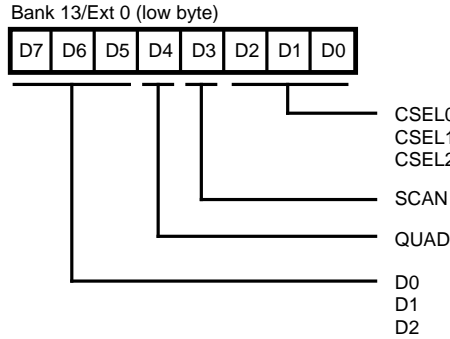


Figure 9. ADC Architecture

**FUNCTIONAL DESCRIPTION** (Continued)



**Figure 10. ADCTL Register (Low Byte)**

**Prescaler Values (bits 7, 6, 5)**

| D2 | D1 | D0 | Prescaler<br>(Crystal 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                                |
| 1  | 1  | 1  | 64                                |

**Note:**

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

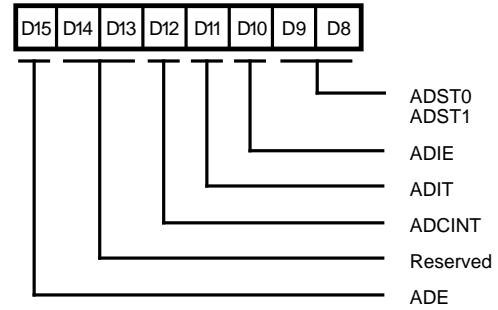
**Modes (bits 4, 3)**

| 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       |

Bank 13/Ext 0 (high byte)



**Figure 11. ADCTL Register (High Byte)**

**ADE** (bit 15). A 0 disables any A/D conversions or access-ing any ADC registers except writing to ADE bit. A 1 Enables all ADC accesses.

**Reserved** (bits 14, 13). Reserved for future use.

**ADCINT** (bit 12). This is the ADC Interrupt bit and is Read Only. The ADCINT will be reset any time this register is written.

**ADIT** (bit 11). This bit selects when to set the ADC Interrupt if ADIE=1. A value of 0 sets the Interrupt after the first A/D conversion is complete. A value of 1 sets the Interrupt after the fourth A/D conversion is complete.

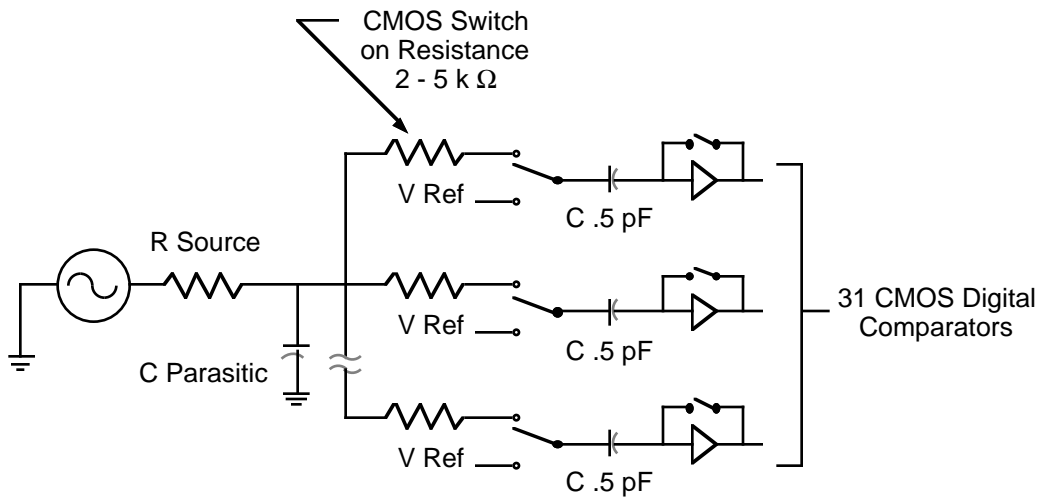
**ADIE** (bit 10). This is the ADC Interrupt Enable. A value of 0 disables setting the ADC Interrupt. A value of 1 enables setting the ADC Interrupt.

**START (bits 9, 8)**

| ADST1 | ADST0 | Mode   |
|-------|-------|--|
| 0     | 0     | Conversion starts when this register is written.   |
| 0     | 1     | Conversion starts on a rising edge INT1 input pin. |
| 1     | 0     | Conversion starts when Timer 2 times out.          |
| 1     | 1     | Conversion starts when Timer 0 times out.          |

There are four ADC result registers. For their location in the different banks, see EXT Register Assignments.

Figure 12 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. This effectively shunts 31 parallel internal resistance of the analog switches and simultaneously charges 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 conditions 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 factor.



**Figure 12. Input Impedance of ADC**

FUNCTIONAL DESCRIPTION (Continued)

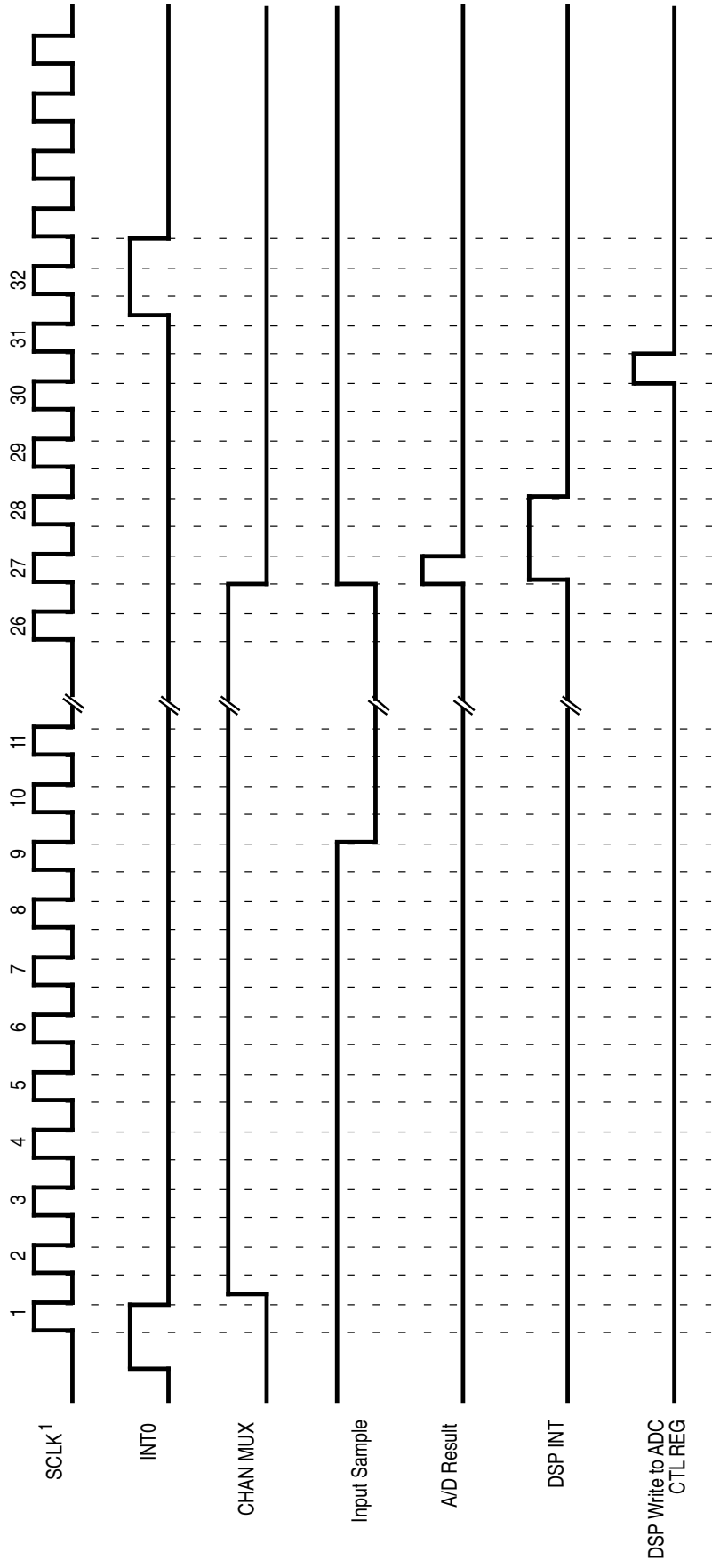


Figure 13. ADC Timing Diagram

## TIMER/COUNTERS

The Z89323/373/393 has two 16-bit Timer/Counters that can be independently configured to operate in various modes. Each is implemented as a 16-bit Load Register (TMLR) and a 16-bit down counter (TMR). Timer/Counter inputs can be selected from among UI0 or UI1 pins and outputs from among UO0 or UO1 pins. The Timer/Counter clock is a scaled version of system clock. Each counter has an 8-bit clock prescaler with divide count controlled by the 16-bit Prescaler Load Register (TPLR). The clock rates of the two timer/counters are independent of each other. External input events occur optionally on the rising edge,

the falling edge, or both rising and falling edges of the input. Output actions on external pins can be programmed to occur with either polarity. The Timer/Counter operational modes are selected through the 16-bit Control Register (TCTL). This register defines the operational modes of its companion Timer (Figure 14).

Each Timer contains a set of five 16-bit Registers. The Ext Register Assignment specifies the location of each Timer Registers. All accesses to Timer Registers occur with zero Wait States.

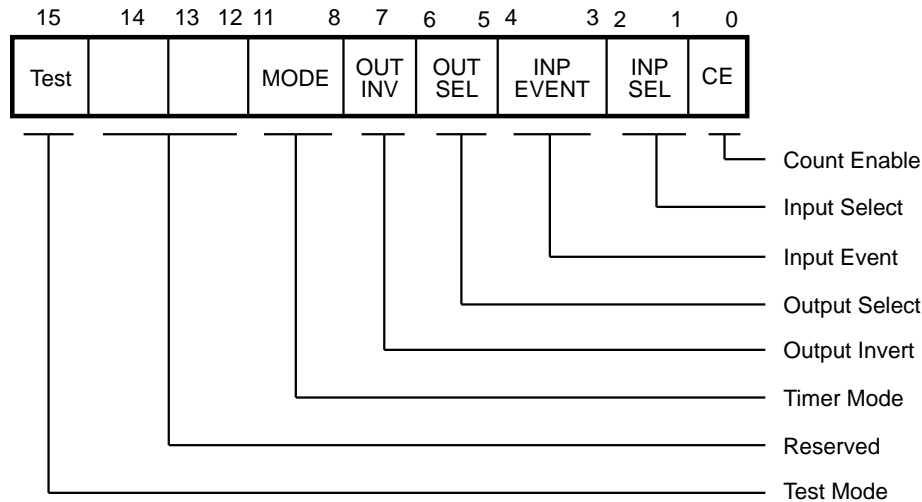


Figure 14. TCTL Register

## TIMER/COUNTERS (Continued)

### Timer Modes

The Timer modes can be categorized as input modes and output modes. In input modes, the Timer/Counter is used for input signals only. In output modes, a selected output pin is driven. If a Timer/Counter is enabled (CE=1) and an output pin, UO0 or UO1, is selected to be driven, the DSP Processor's Status Register bits 5 or 6 does not affect the state of that pin.

### Output Modes

**MODE 0.** The Timer/Counter is configured to generate a continuous square wave of 50% duty cycle. Writing a new value to the TMLR Register takes effect at the end of current cycle unless TMR is written.

**MODE 1.** The Timer/Counter is configured to generate a single pulse of programmable duration. The asserted state may be either logic high or logic low. Retriggering the one-shot before the end of the pulse causes it to continue for the new duration.

**MODE 2.** The Timer/Counter is configured to generate a pulse-width modulated repeating waveform. The duty cycle ranges from 0–100% (0/256 to 255/256) of a cycle in steps of 1/256 of a cycle. The asserted state of the waveform may be either logic high or logic low. Writing a new pulse-width value to the TMLR Register takes effect at the end of current cycle unless TMR is written.

**MODE 3.** The Timer/Counter is configured to generate a pulse-width modulated repeating waveform. The duty cycle ranges from 0–100% (0/65,536 to 65,535/65,536) of a cycle in steps of 1/65,536 of a cycle. The asserted state of the waveform may be either logic high or logic low. Writing a new pulse-width value to the TMLR Register takes effect at the end of current cycle unless TMR is written.

**MODE 4.** The Timer/Counter is configured to generate a series of pulses ranging from 0 to 65,535. The pulses are actually the Timer Clock (TMCLK), which is gated to the output until the counter under flows.

**MODE 5.** The Timer/Counter is configured to generate an output pulse that is asserted under program control, and de-asserted when a programmable number of input edges (up to 65,535) have been counted on an input pin (UI0 or UI1). Assertion may be either logic high or logic low.

**MODE 6.** The Timer/Counter is configured to generate a Hardware Reset on time-out unless retrigged by software.

**MODE 7.** The Timer/Counter is configured to generate a Hardware Reset on time-out unless retrigged by an event on one of the input pins UI0 or UI1.

### Input Modes

The input modes use one of the input pins UI0 or UI1. The signals on these pins are synchronized with the internal Timer Clock, TMCLK, before being applied to the Timer. The input signal frequency must be no higher than 1/4th of TMCLK frequency.

**MODE 8.** The Timer/Counter is configured to measure the time for which its input is asserted.

**MODE 9.** The Timer/Counter is configured to measure the period from one rising (falling) edge to the next rising (falling) edge on the input.

**MODE 10.** The Timer/Counter is configured to count the number of input edges (up to 65,535). Input edges may be selected as rising or falling or both.

**MODE 11.** The Timer/Counter is configured to count the number of input edges (up to 65,535) in a time window set by the second timer. Edges are counted until the second timer under flows. Input edges may be selected as rising or falling or both.



Bank 13/EXT1 (Timer0) or Bank 14/EXT1 (Timer1)  
Timer Control Register (TCTL)

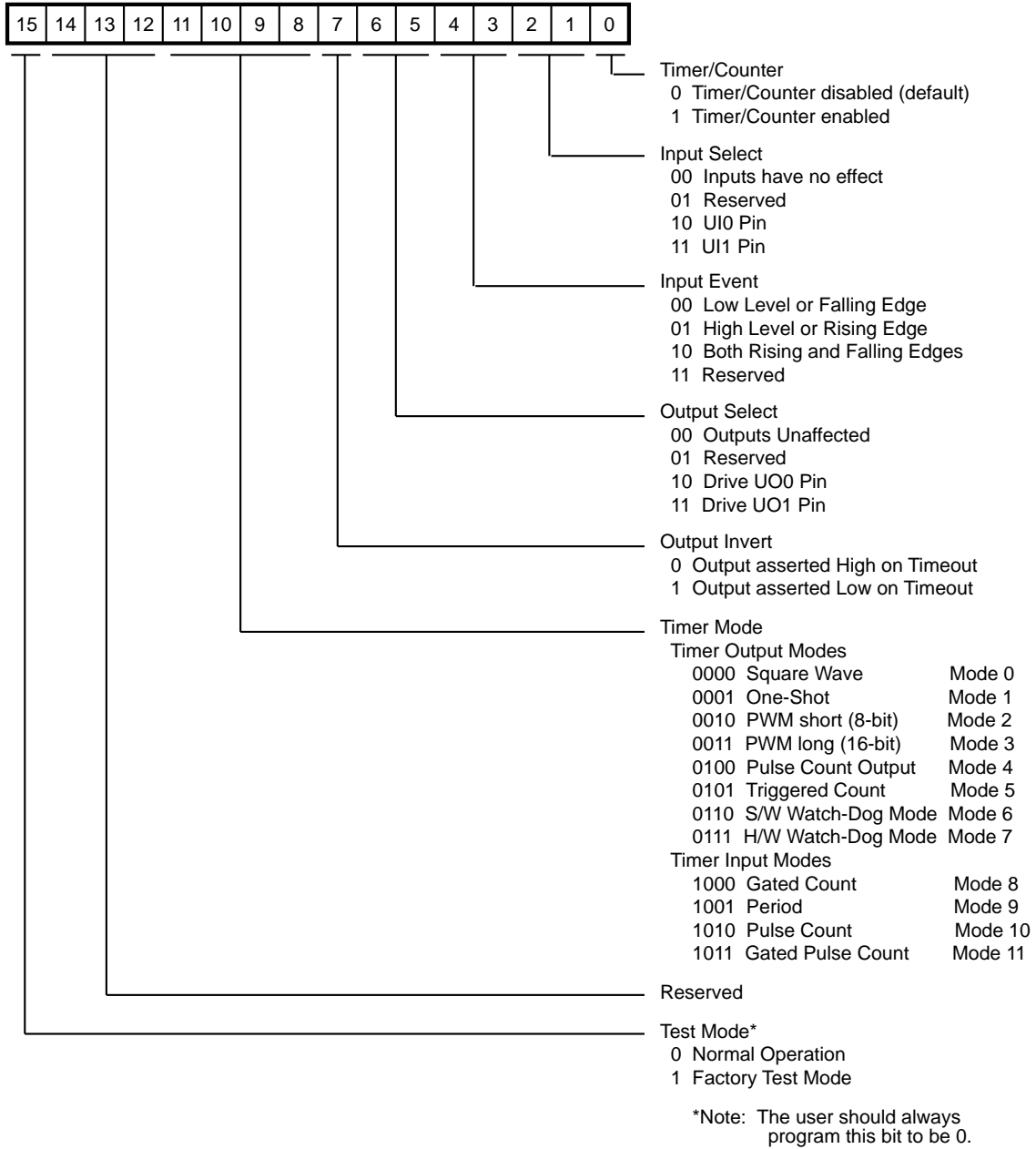
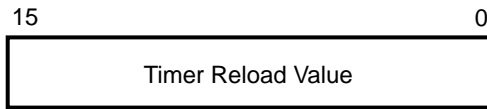


Figure 15. Register Bit Fields

**TIMER/COUNTERS** (Continued)

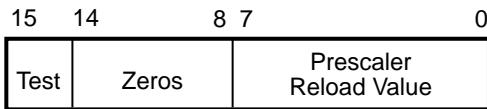
**Timer Load Register (TMLR)**

This 16-bit Register holds a value that is reloaded into timer upon timer under flow.



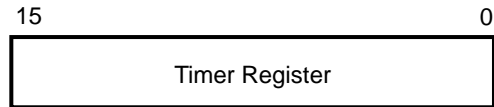
**Timer Prescaler Load Register (TPLR)**

The 16-bit TPLR Register holds the prescaler reload value in its lower 8 bits. Bit 15 is the Timer's Interrupt Pending bit. When set, it signifies an interrupt event in its companion timer. The IP bit can only be set by the Timer. It can be cleared only by software when it writes a value to this register with a "1" in bit position 15; a "0" in bit position 15 will have no effect on the state of IP bit. Bits [14:8] must always be written with 0s for future compatibility.



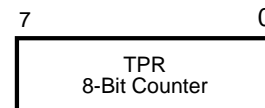
**Timer Register (TMR)**

TMR is a 16-bit down counter that holds the current Timer/Counter value. It can be read as any ordinary register. However, writing to TMR is different than writing to an ordinary register. A write to TMR Register causes the contents of TMLR Register to be written into it, causing the Timer to be retrigged. Any data on DSP's Memory Data (MD) Bus is ignored during a write to TMR.



**Timer Prescale Register (TPR)**

TPR is an 8-bit down counter that holds the current Prescaler count value. It can be read as any ordinary register. However, writing to TPR is different than writing to an ordinary register. A write to TPR Register causes the lower 8-bit contents of TPLR Register to be written into it, causing the Prescaler to be retrigged. Any data on DSP's Memory Data (MD) Bus is ignored during a write to TPR.



### Prescaler Operation

The Timer/Counter Clock (TMCLK) is generated by the output of the prescaler. The Prescaler is an 8-bit down counter, TPR, followed by a divide-by-two flip-flop that generates a 50 percent duty cycle output clock TMCLK. The Prescaler's input clock is the system clock, CLKIN, divided by two. Thus, the maximum prescaler output frequency is 1/4 of the system clock frequency.

Once the prescaler counter is loaded, it decrements at its clocked frequency and generates an output to the divide-by-two flip-flop. When the count reaches 0, the counter is reloaded from the lower 8 bits of TPLR Register.

The 8-bit prescaler counter is loaded with value in TPLR Register field [7:0] in one of three ways:

1. When 8-bit prescaler counter, TPR, decrements to zero.
2. By writing to TPR Register.
3. When companion Timer/Counter TMR is reloaded upon under flow from its TMLR Register, or retrigged by writing directly to TMR Register.

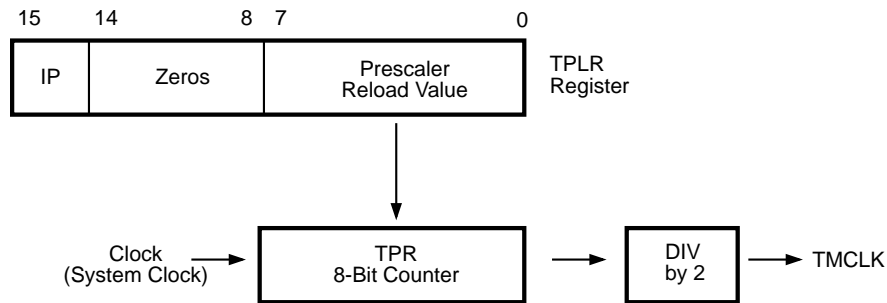


Figure 16. Prescaler Block Diagram

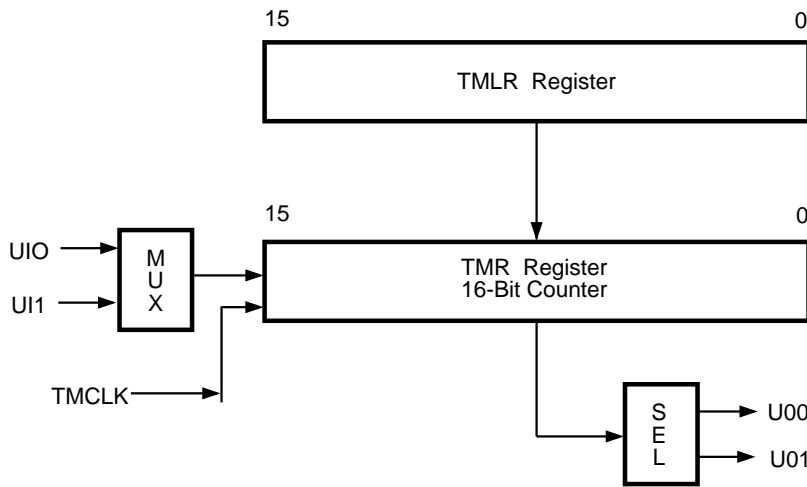


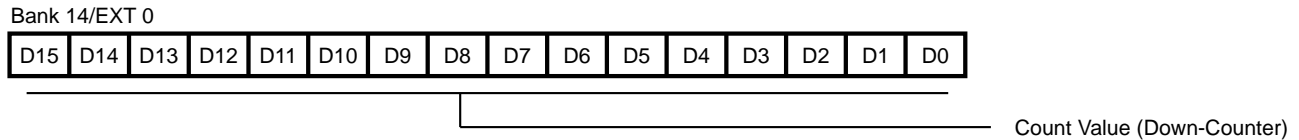
Figure 17. Counter/Timer Block Diagram

**TIMER/COUNTERS** (Continued)

**16-Bit General-Purpose Timer/Counter T2**

The 16-bit timer/counter is available for general-purpose use. When the counter counts down to the zero state, the timer 2 load register loads into timer 2, and if timer 2 interrupt is enabled, an interrupt is received. The counting operation of the counter can be disabled. The timer/counter clock source can be selected to be system clock/2 or UI2.

The counter is defaulted to the Enable state. If the system designer does not choose to use the timer, the counter can be disabled.



**Figure 18. Timer/Counter 2 Load Register**

**I/O Ports**

I/O pin allocation for ports in the different package types is designed to provide increased flexibility and support for various modes of operation. The 44-pin package features the special signals, as well as all packages supporting the EXT 16-bit bus. In cases where the application does not

require an external EXT bus, these I/O pins can be allocated to 16-bit general-purpose I/O port (P0), the special signals port (P1) or additional port (P3). The 80-pin PQFP package supports up to 40 I/O pins.

**Table 9. Various Package I/O Port Allocation**

| Pin Count Package | 44-Pin PLCC/PQFP | 68-Pin PLCC | 80-Pin PQFP | 100-Pin PQFP† |
|-------------------|------------------|-------------|-------------|---------------|
| P0[15:8]          | EXT,P0,P1*       | EXT,P0      | EXT,P0      | EXT,P0        |
| P0[7:0]           | EXT,P0           | EXT,P0      | EXT,P0      | EXT,P0        |
| P1[7:0]           |                  | P1*         | P1*         | P1*           |
| P2[7:0]           | P2[4:0]*         | P2*         | P2*         | P2*           |
| P3[7:0]           |                  |             | P3          |               |

**Note:**

\* Ports with special signals: Interrupts inputs, Serial Peripheral Interface (SPI), CLKOUT and Timers inputs and outputs.

† (ICE chip)



### 8-Bit Programmable I/O (Port 1)

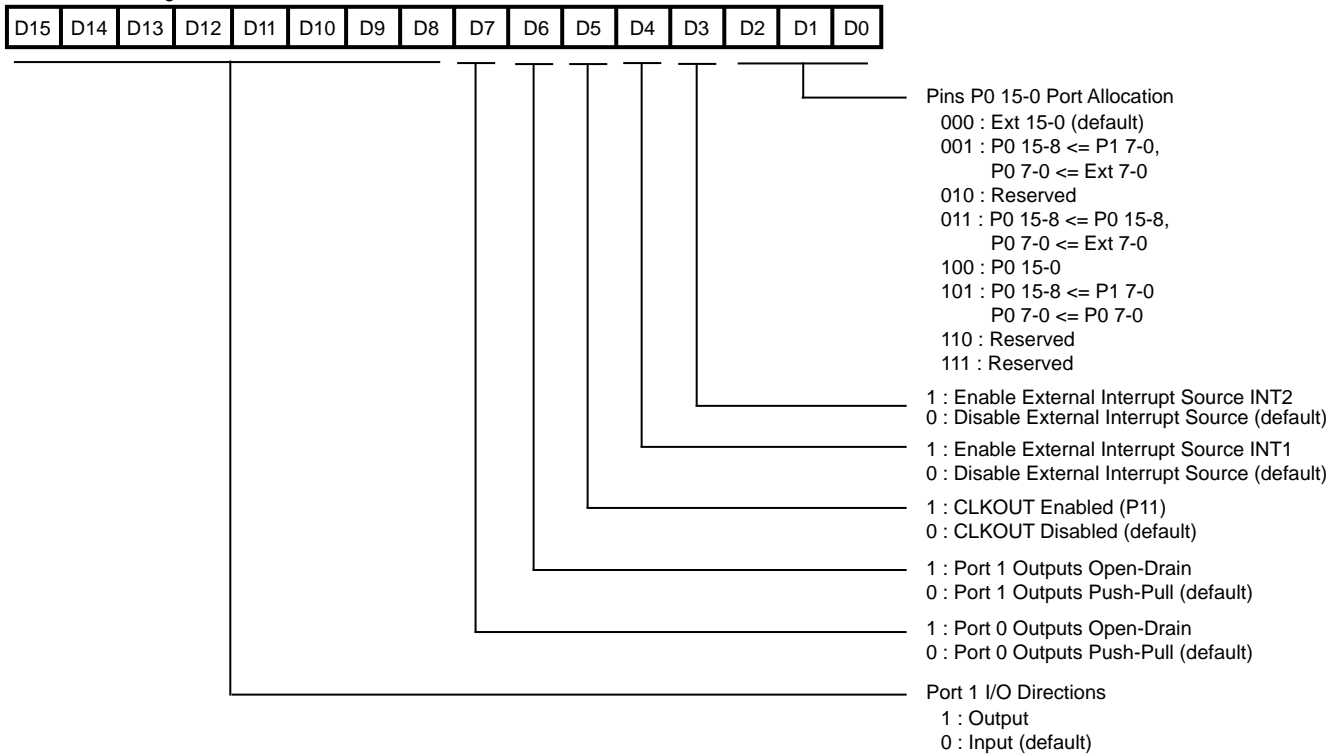
When the appropriate bit is set in the Port 1 control register, Port 1 acts as an 8-bit programmable, bi-directional, CMOS-compatible port. Each of the eight lines can be independently programmed as an input or an output or

globally as an open-drain output. When enabled, Bank0/EXT5 (Least Significant Bit) acts as the data I/O register. Bank15/EXT1 serves as the Port1 direction control register. Port 1 can also be programmed to provide special I/O functions.

**Table 10. Port 1 Bit Function Selection**

| Port.Bit | IF (Condition Explanation)                               | Then   | Else |
|----------|--|--------|------|
| P1.0     | Bank15/Ext1(3)=1 (Enable External Interrupt Source INT2) | INT2   | P10  |
| P1.1     | Bank15/Ext1(5)=1 (CLKOUT Enable)                         | CLKOUT | P11  |
| P1.2     | Bank15/Ext4(0)=1 (SPI Enable)                            | SIN    | P12  |
| P1.3     | Bank15/Ext4(0)=1 (SPI Enable)                            | SOUT   | P13  |
| P1.4     | Bank15/Ext4(0)=1 (SPI Enable)                            | SS     | P14  |
| P1.5     | Bank15/Ext4(0)=1 (SPI Enable)                            | SK     | P15  |
| P1.6     | Bank13/Ext1(2-1)=10 or Bank14/Ext1(2-1)=10 (UI0 Enable)  | UI0    | P16  |
| P1.7     | Bank13/Ext1(2-1)=11 or Bank14/Ext1(2-1)=11 (UI0 Enable)  | UI1    | P17  |

Bank 15/Ext 1 Reg



**Figure 21. Bank15/EXT1 Register**

### 8-Bit Programmable I/O (Port 2)

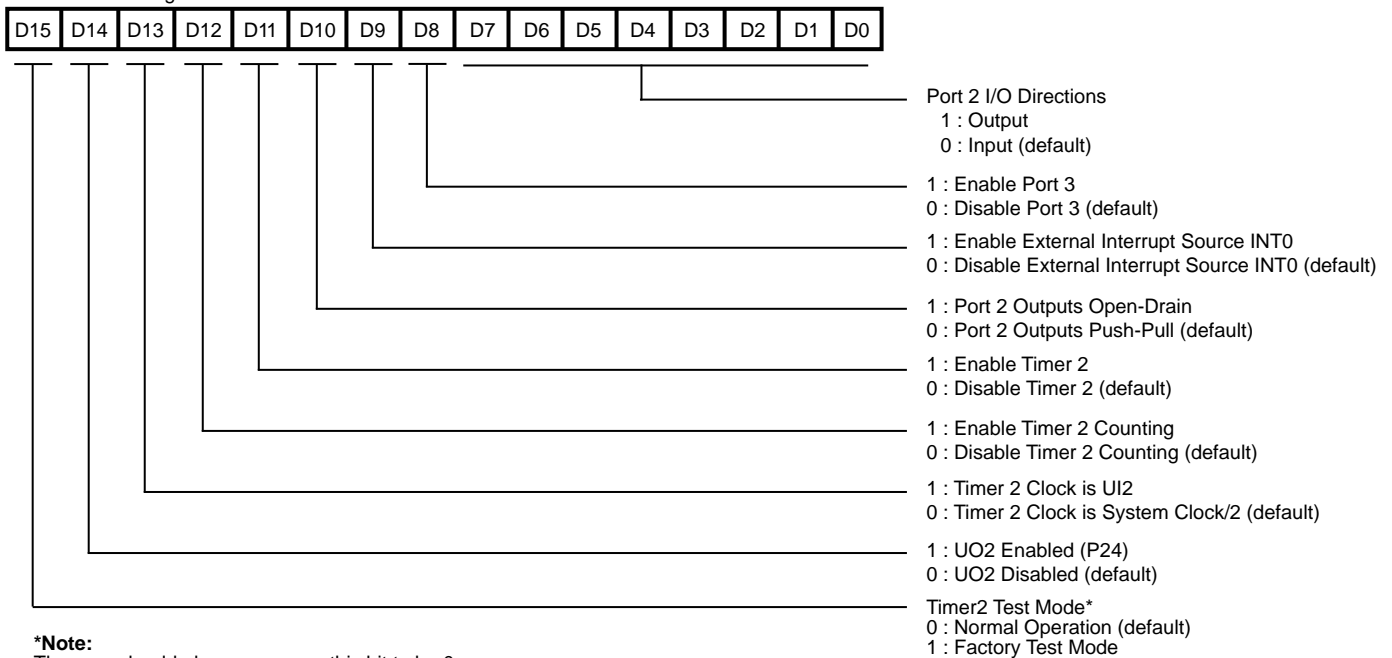
Port 2 is an 8-bit programmable, bidirectional, CMOS-compatible port. Each of the eight lines can be independently programmed as an input or an output or globally as an open-drain output. Port 2 can also be

programmed to provide special I/O functions. When Port 2 acts as programmable I/O, Bank0/EXT5 (MSB) acts as the data I/O register. Bank15/EXT2 serves as Port 2 control register.

**Table 11. Port 2 Bit Function Selection**

| Port.Bit | IF (Condition Explanation)                               | Then | Else |
|----------|--|------|------|
| P2.0     | Bank15/Ext2(9)=1 (Enable External Interrupt Source INT0) | INT0 | P20  |
| P2.1     | Bank15/Ext1(4)=1 (Enable External Interrupt Source INT1) | INT1 | P21  |
| P2.2     | Bank13/Ext1(6-5)=10 or Bank14/Ext1(6-5)=10 (UO0 Enable)  | UO0  | P22  |
| P2.3     | Bank13/Ext1(6-5)=11 or Bank14/Ext1(6-5)=11 (UO0 Enable)  | UO1  | P23  |
| P2.4     | Bank15/Ext2(14)=1 (UO2 Enable)                           | UO2  | P24  |
| P2.5     | Bank15/Ext2(13)=1 (Timer2 Clock is UI2)                  | UI2  | P25  |
| P2.6     |  | P26  | P26  |
| P2.7     |  | P27  | P27  |

Bank 15/Ext 2 Reg



**Figure 22. Bank15/EXT2 Register**

### 8-Bit Programmable I/O (Port 3)

Port 3 is an additional I/O port featured only in the 80-pin PQFP package. P3[3:0] are inputs and P3[7:4] are outputs. The purpose of this additional port is to serve applications that need more than 32 I/O pins. **Port 3 enables the user to support up to 40 I/O pins. Port 3 is not**

**supported in the 100-pin ICE chip PQFP package, therefore this port is not supported in the Z893x3 emulator, and use of this port is not recommended in cases when the other I/O ports can support the I/O requirements.**

## Serial Peripheral Interface

**Serial Peripheral Interface (SPI).** The Z893X3 incorporates a serial peripheral interface for communication with other microcontrollers and peripherals. The SPI includes features such as Master/Slave selection. The SPI consists of two registers; SPI Control Register (SCON), SPI Receive/Buffer Register (RxBUF), and SPI Shift Register (Figure 23). **Note:** The SPI shift register and Receive/Buffer register are one in the same and are shown in Figure 41. SCON is located in bank 15/Ext4 (LSB). This register is a read/write register that controls; Master/Slave selection, SS polarity, 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 tri-stated. 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 is the SS polarity bit. A 0 selects active Low (default) polarity on SS, and a 1 selects active High. Bit 4 signals that a receive character is available in the RxBUF Register. If the associated interrupt enable bit 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 a division of the internal system clock for clocking the SPI. 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 Low. Data is transferred into the

slave's SPI Shift Register, through the SIN 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 SPI 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.

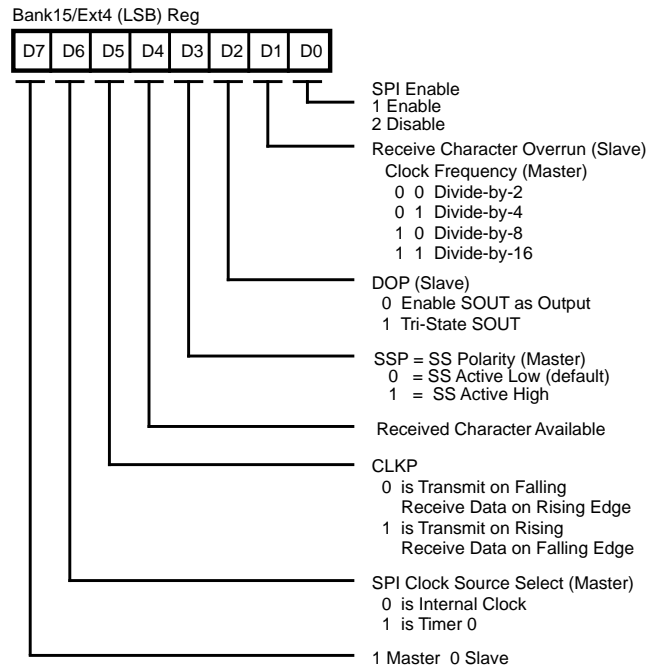


Figure 23. SPI Control Register (SCON)

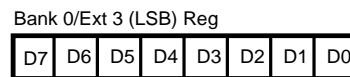


Figure 24. SPI TXRXDATA Register



When the communication between the master and slave is complete, the SS goes High. Unless disconnected, for every bit that is transferred into the slave through the SIN pin, a bit is transferred out through the SOUT pin on the opposite clock edge. During slave operation, the SPI clock pin (SK) is an input (Figure 25). In master mode, the DSP must first activate a SS through one of it's I/O ports. Next, data is transferred through the master's SOUT 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 SPI interrupt and flag is generated. Before data is transferred through the SOUT pin, the SPI Enable bit in the SCON Register must be enabled. The MSB bit 7 is shifted out first.

**SPI Clock.** The SPI clock can be driven from three sources; with T0, 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. Divided 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. The SPI interrupt can be enabled or disabled (default) in the Interrupt Allocation Register (Bank 15/Ext 6). 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.

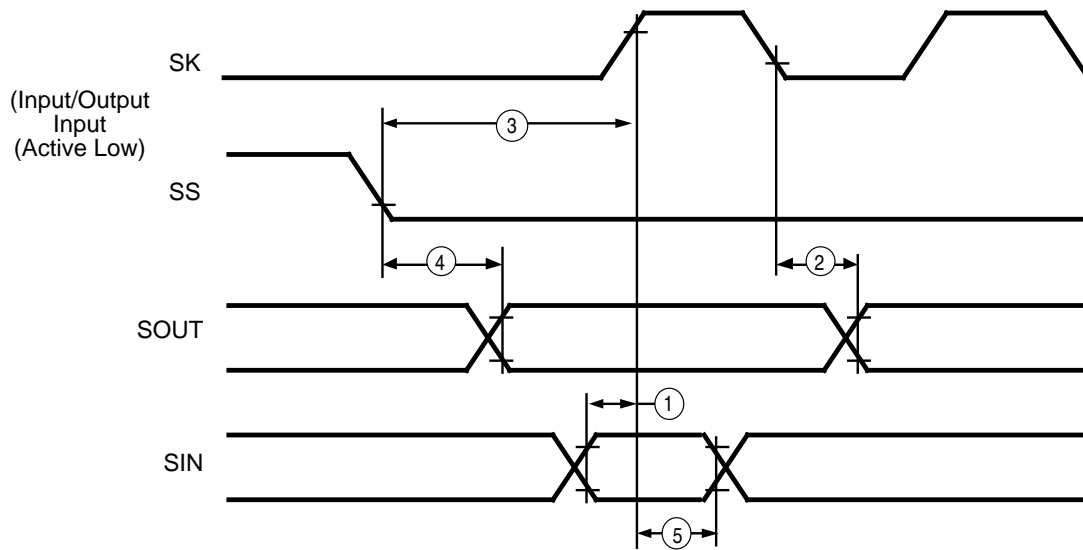


Figure 25. SPI Timing

### CLOCK Circuits

The clock generator includes Phase-Locked Loop (PLL) circuit to enable use of low frequency crystal. The benefits of using low frequency crystal are low system cost, low power consumption and low EMI. The PLL circuit can be bypass (s/w controlled).

The clock generated by the PLL circuit (VCO clock) is programmable and controlled by the PLL Divider register.

DSP (System) clock source is programmable and can be one of the 4 options: VCO clock, VCO clock divided by 2, VCO clock divided by 4 or twice the crystal frequency.

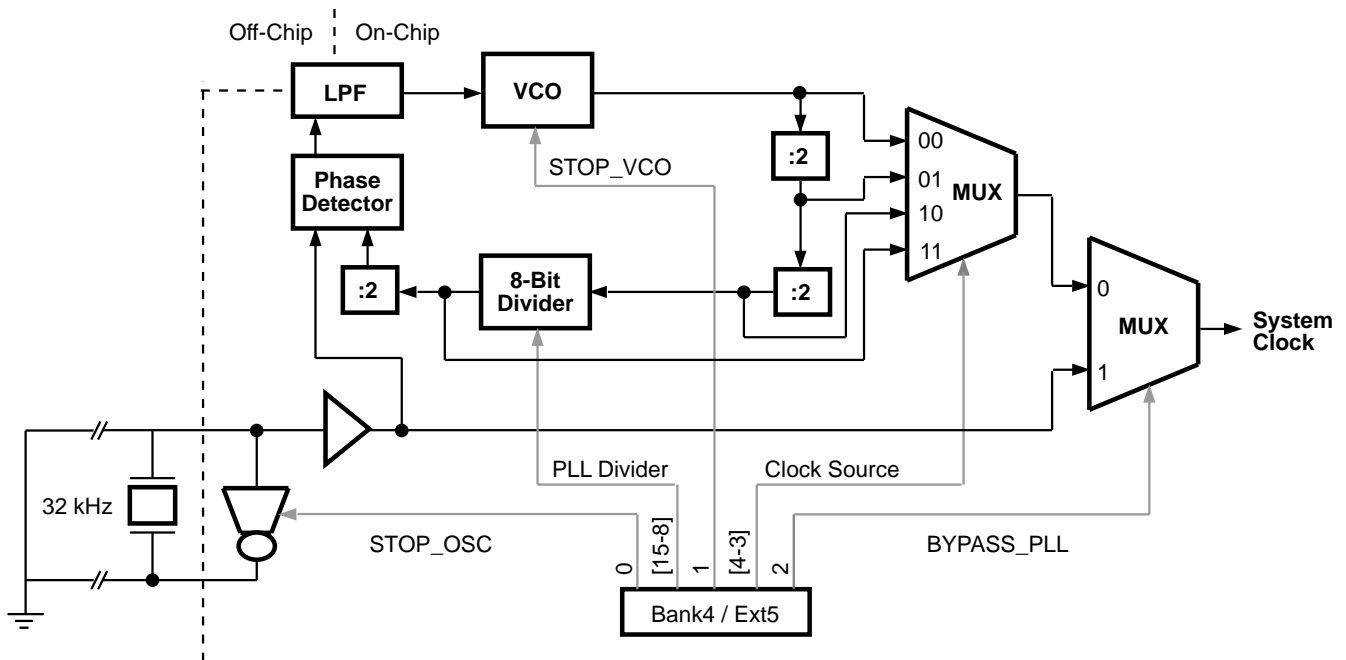
Whenever the PLL circuit is switched from Stop VCO to Enable VCO, a software delay of 10 msec must be used before switching the system clock from the oscillator to the PLL, in order to give the PLL time to be stable.

**Table 12. CLOCK Modes**

| STOP_OSC | STOP_VCO | BYPASS_PLL | Mode  |
|----------|----------|------------|---|
| 0        | 0        | 0          | 0) Normal - High frequency clock                |
| 0        | 0        | 1          | 1) 32 Khz - VCO running (fast switching time)** |
| 0        | 1        | 0          | 2) STOP CLOCK - Oscillator running              |
| 0        | 1        | 1          | 3) 32 Khz                                       |
| 1        | 1        | 0          | 4) STOP CLOCK                                   |
| 1        | 1        | 1          | 5) EXTERNAL CLOCK source *                      |

**Notes:**

- \* In this clock mode, it is possible to use external clock source instead of the internal oscillator source.
- \*\* Default (power-up) mode of operation.



**Figure 26. PLL Functional Block Diagram**

### Power Down

The Z893X3 supports different levels of power-down modes to minimize device power consumption. The lowest power consumption is at STOP Clock Mode when the Oscillator is turned off (clock modes 2 and 4 when there is no external clock.) The highest power consumption is when the Z893X3 in Normal mode (Clock Mode 0) and there is medium power consumption mode .The SLOW Clock Mode is when the DSP is running with 32 kHz clock (Crystal Clock Modes 1 and 3) and disabling all the peripherals which are not needed in this mode.

### Slow Mode

The SLOW mode reduce the chip power consumption by using the 32 kHz clock (Clock Mode 3) of the crystal as a DSP clock and disabling in software all the unnecessary peripherals.

Clock Mode 1 also uses the 32 kHz clock, but in this mode the VCO is still running to enable fast switching (wake up) to the high frequency.

### Stop Mode

The STOP mode provides the lowest possible device standby current. In this mode of operation the on chip oscillator and internal system clock are turned off.

In Clock Mode 2 the Oscillator is running while the system clock is turned off to enable fast switching (wake up) to the high frequency.

STOP mode is exited when the recovery source as defined in Bank4/EXT5[6:5] is toggled to the recovery defined level. In case of Clock Mode 2 the program resumes operation starting from the next instruction after the stop instruction. In case of Clock Mode 4, the program resumes operation starting from the reset vector address after executing operations similar to the Power-On Reset sequence of operations.

Bank 15/Ext 5 Reg

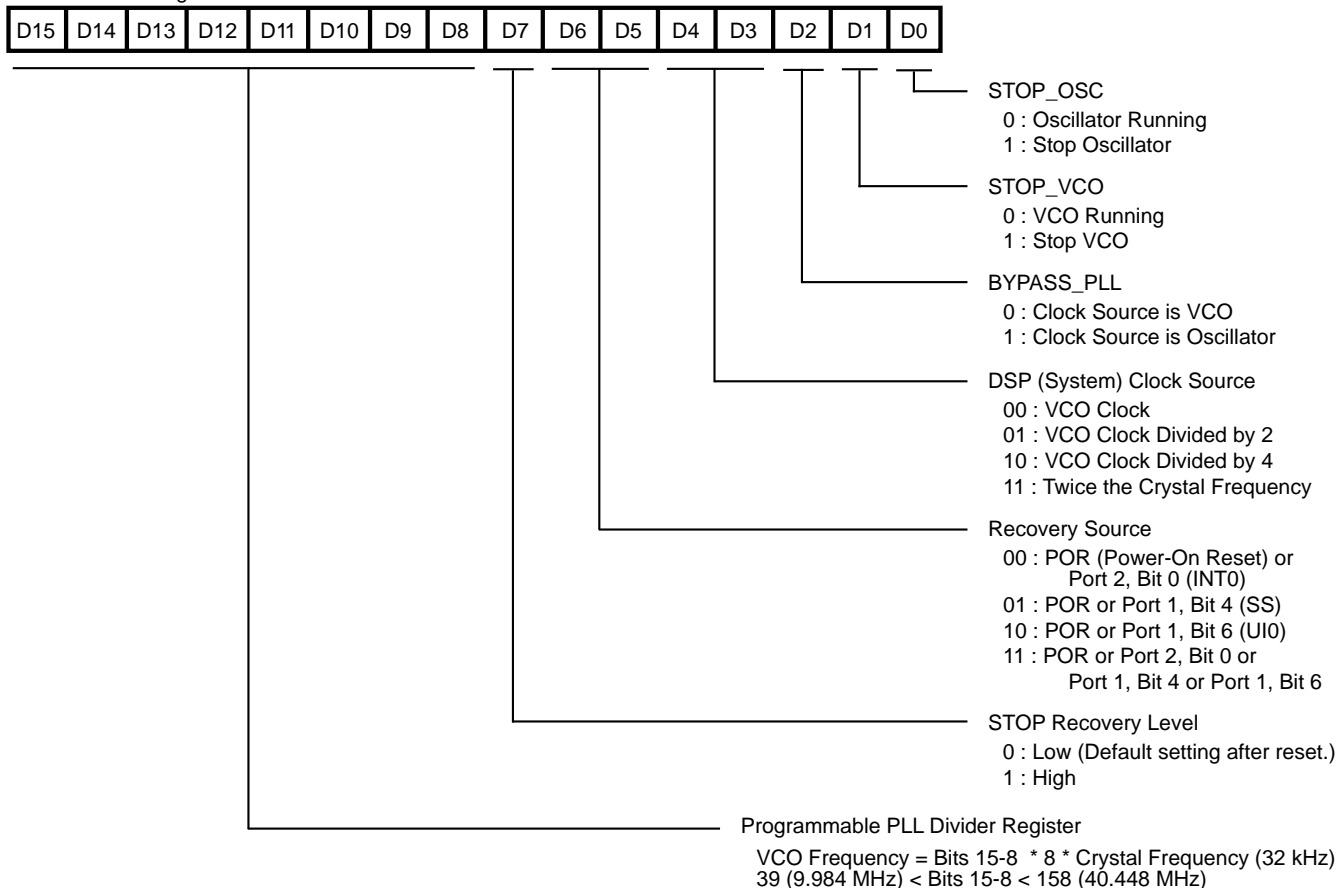


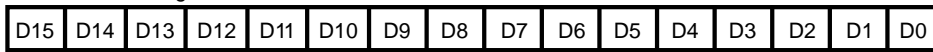
Figure 27. PLL Register

## Interrupt Controller

There are eight different interrupt sources (when all of them are enabled). Bits [3:0] of the Interrupt Allocation Register defines which interrupt source will have the highest priority and will be allocated into IINT0 (Internal INT0). Bits[7:0] of the Interrupt Allocation Register defines which interrupt source will have the second highest priority and will be allocated into IINT1 (Internal INT1). Bits[15:8] are enable

bits for specific interrupt sources. All the enabled interrupts which are not already allocated into IINT0 or IINT1 are allocated into IINT2. When interrupt happen on IINT2 then IINT2 interrupt routine is reading the Interrupt Status Register (EXT7 in all the Banks) to determine which interrupt occurred and decides on the relative priority. The Interrupt Status Register can be used for polling interrupts mode.

Bank 15/Ext 6 Reg



IINT0 Source  
 0000 : A/D Finish      0001 : SPI  
 0010 : Timer0          0011 : Timer1  
 0100 : Timer2          0101 : INT0 H/W  
 0110 : INT1 H/W        0111 : INT2 H/W  
 1000 – 1111 : IINT0 Disabled

IINT1 Source  
 0000 : A/D Finish      0001 : SPI  
 0010 : Timer0          0011 : Timer1  
 0100 : Timer2          0101 : INT0 H/W  
 0110 : INT1 H/W        0111 : INT2 H/W  
 1000 – 1111 : IINT1 Disabled

IINT2 Interrupt Sources

|                    | Interrupt Enable | Interrupt Disable |
|--------------------|------------------|-------------------|
| Bit 8 = A/D Finish | 1                | 0                 |
| Bit 9 = SPI        | 1                | 0                 |
| Bit 10 = Timer0    | 1                | 0                 |
| Bit 11 = Timer1    | 1                | 0                 |
| Bit 12 = Timer2    | 1                | 0                 |
| Bit 13 = INT0 H/W  | 1                | 0                 |
| Bit 14 = INT1 H/W  | 1                | 0                 |
| Bit 15 = INT2 H/W  | 1                | 0                 |

**Note:** An Interrupt that is not selected as a source to IINT0, IINT1 or IINT2 is disabled.

**Figure 28. Interrupt Allocation Register**

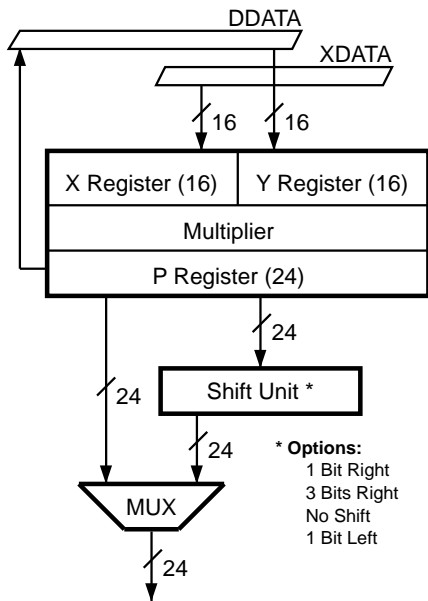
**FUNCTIONAL DESCRIPTION**

**Instruction Timing.** Most instructions are executed in one machine cycle. Long immediate instructions and Jump or Call instructions are executed in two machine cycles. A multiplication or multiplication/accumulate instruction requires a single cycle. Specific instruction cycle times are described in the Condition Code section.

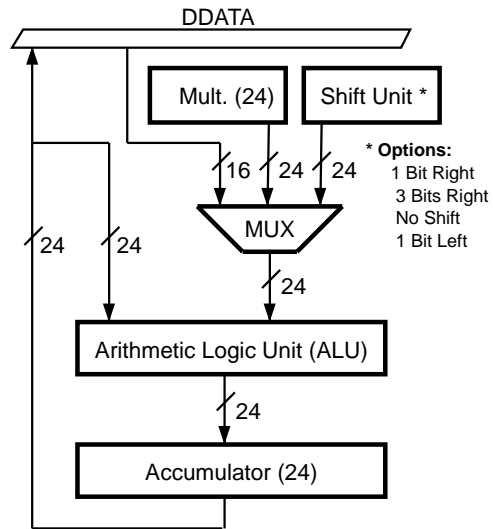
**Multiply/Accumulate.** The multiplier can perform a 16-bit x 16-bit multiply, or multiply accumulate, in one machine cycle using the Accumulator and/or both the X and Y inputs. The multiplier produces a 32-bit result, however, only the 24 most significant bits are saved for the next instruction or accumulation. 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 (Figure 29). 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  $8000H \times 8000H = 8000H$  ( $-1 \times -1 = -1$ ).

**ALU.** The 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 (Figure 30).



**Figure 29. Multiplier Block Diagram**



**Figure 30. ALU Block Diagram**

## FUNCTIONAL DESCRIPTION (Continued)

**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, and the RET instruction pops the contents of the stack to the PC.

**User Inputs.** The Z89323 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 8).

**User Outputs.** The status register bits S5 and S6 connect directly to UO0 and UO1 pins and may be written to by the appropriate instruction. **Note:** The user output value is the opposite of the status register content.

**Interrupts.** The Z89323 has three positive edge-triggered interrupt inputs serving up to eight interrupt sources. An interrupt is acknowledged at the end of an instruction execution. It takes two machine cycles to enter an interrupt instruction sequence. The PC is pushed onto the stack and Interrupts are globally disabled. 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 IINT0 = highest, IINT2 = lowest. **Note:** The SIEF instruction globally enables the interrupts. The SIEF instruction must be used before exiting an interrupt routine since the interrupts are automatically disabled when entering the routine. (See Interrupt Controller section for more details.)

**Registers.** The Z89323 has 28 physical internal registers, eight external registers and 15 peripheral control registers. The EA2-EA0 determines the address of the external registers. The signals are used to read from or write to the external registers /DS, WAIT, RD//WR.

**I/O Bus.** The processor provides a 16-bit, CMOS-compatible bus. I/O Control pins provide convenient communication capabilities with external peripherals, and single-cycle access is possible. For slower communications, an on-board hardware wait-state generator can be used to accommodate timing conflicts. Three latched I/O address pins are used to access external registers. Disabling a peripheral allows access to these addresses for general-purpose use.

**Wait-State Generator.** An internal Wait-State generator is provided to accommodate slow external peripherals. A single Wait-State can be implemented through a control register. For additional states, a dedicated pin (WAIT) can be held High. The WAIT pin is monitored only during execution of a read or write instruction to external peripherals (EXT bus).

**Analog to Digital Converter.** The Z89323 has a 4-channel, 8-bit half-flash analog to digital converter. Two external reference voltages are available externally. The ADC prescales to the system clock and can drive an interrupt at the end of a conversion. There are four channels of input with the ADC which can be programmed to convert values either continuously or on an event (timer or interrupt).

**Timer/Counter/PWMs (T0, T1).** Timer0 and Timer1 are 16-bit timer-counters with 8-bit prescalers. They also offer the option of being used as PWM generators and have both hardware and software Watch-Dog capabilities. Both timers are identical and can be externally or internally clocked and can drive any of the three hardware interrupts.

**Timer/Counter (T2).** Timer 2 is a general-purpose 16-bit timer/counter. It can be externally or internally clocked and drive either IINT0 and IINT1.

**Port 0.** Port 0 is a 16-bit user I/O port. Bits can be configured as input or output or globally as open-drain output. When enabled, Port 0 consumes the 16 data lines used by the EXT bus. Port 0 function and EXT use can be dynamically changed by enabling and disabling Port 0.

**Port 1.** Port 1 is an 8-bit user I/O port. Bits can be configured as input or output or globally as open-drain output.

**Port 2.** Port 2 has multiple functions. It can be used as an 8-bit user I/O port when the other functions within the port are not in use. As an I/O port, these bits can be configured as input or output or globally as open-drain output. Port 2 also supports the SPI, CLKOUT, all three external hardware interrupt signals and all three timer input and output signals.

RAM ADDRESSING

The address of the RAM is specified in one of three ways (Figure 22):

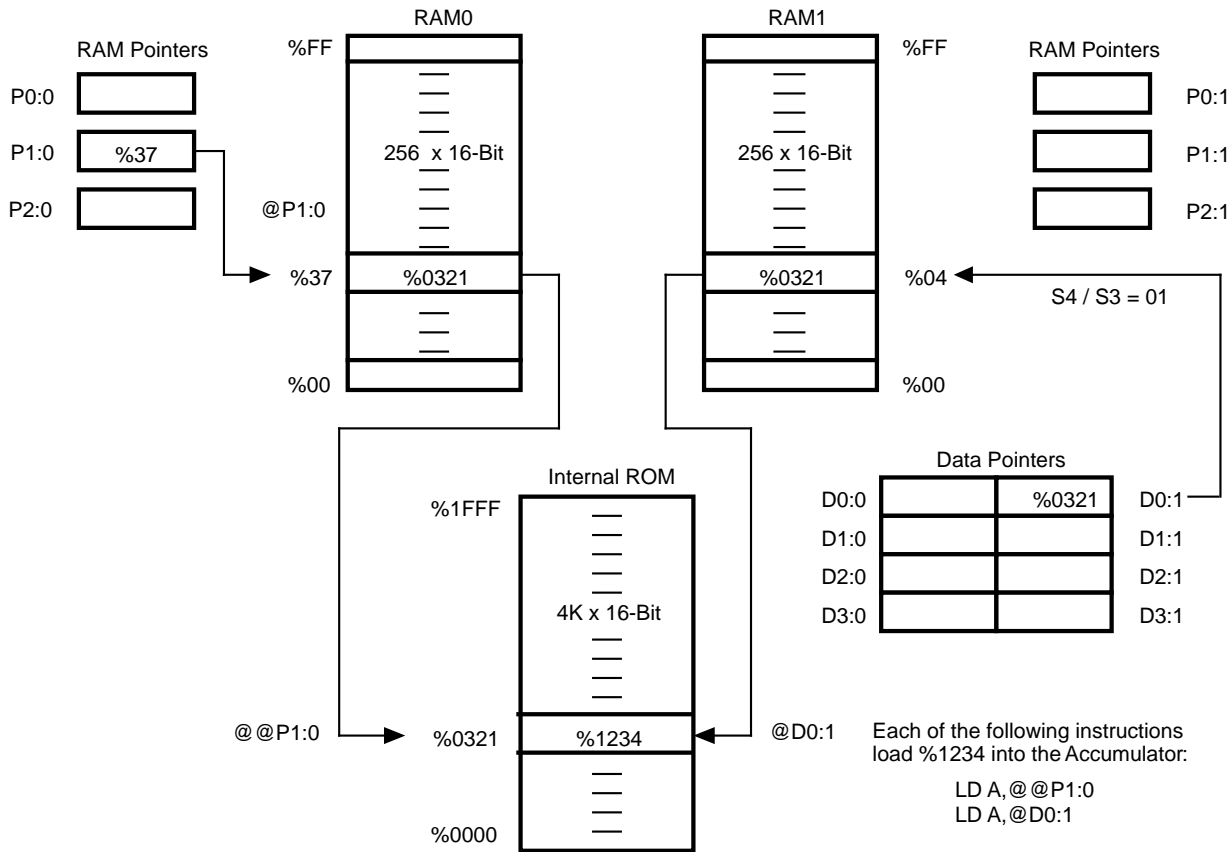


Figure 31. RAM, ROM, and Pointer Architecture

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 6 and 9 may be used by an indirect instruction to specify a register pointer and its modification after execution of the instruction.

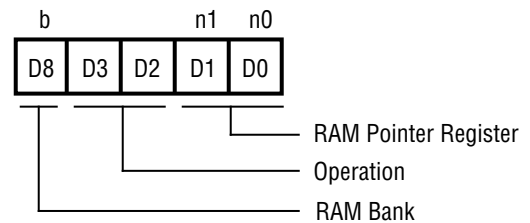


Figure 32. Indirect Register

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 | *                               |
| xx01  | P1:0 or P1:1 | *                               |
| xx10  | P2:0 or P2:1 | *                               |
| xx11  |              | See Short Form Direct           |

**Note:**

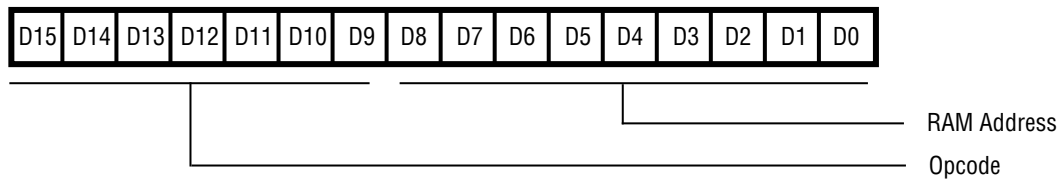
\* If bit 8 is zero, P0:0 to P2:0 are selected;  
if bit 8 is one, P0:1 to P2:1 are selected.

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, for example, 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.

**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 (see Figure 33).



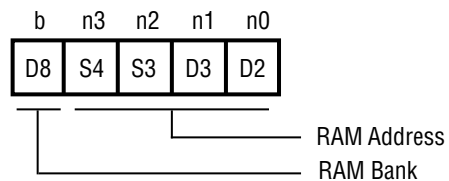
**Figure 33. Direct Internal RAM Address Format**

**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 30). 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 downloading lookup tables and other instructions from program memory to RAM.



**Figure 34. Short Form Direct Address**



INSTRUCTION FORMAT

Table 13. Registers

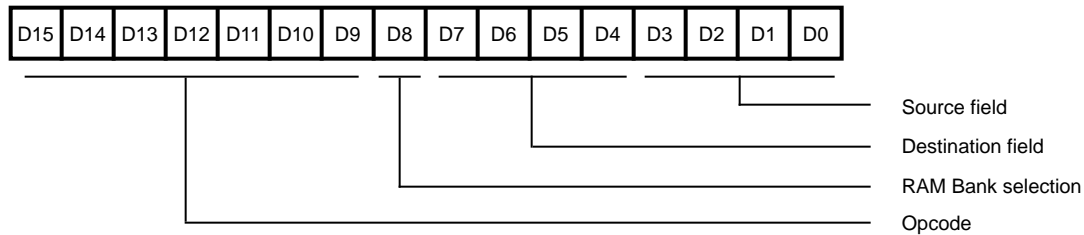
| Source/Destination | Register           |
|--------------------|--------------------|
| 0000               | BUS <sup>[1]</sup> |
| 0001               | X                  |
| 0010               | Y                  |
| 0011               | A                  |
| <hr/>              |                    |
| 0100               | SR                 |
| 0101               | STACK              |
| 0110               | PC                 |
| 0111               | P <sup>[1]</sup>   |
| <hr/>              |                    |
| 1000               | EXT0               |
| 1001               | EXT1               |
| 1010               | EXT2               |
| 1011               | EXT3               |
| <hr/>              |                    |
| 1100               | EXT4               |
| 1101               | EXT5               |
| 1110               | EXT6               |
| 1111               | EXT7               |

Table 14. Register Pointers Field

| Source/Destination | Meaning                               |
|--------------------|---------------------------------------|
| 00xx               | NOP                                   |
| 01xx               | + 1                                   |
| 10xx               | -1/LOOP                               |
| 11xx               | +1/LOOP                               |
| <hr/>              |                                       |
| xx00               | P0:0 or P0:1 <sup>[2]</sup>           |
| xx01               | P1:0 or P1:1 <sup>[2]</sup>           |
| xx10               | P2:0 or P2:1 <sup>[2]</sup>           |
| xx11               | Short Form Direct Mode <sup>[3]</sup> |

Notes:

- [1] If RAM Bank bit is 0, then Pn:0 are selected.  
If RAM Bank bit is 1, then Pn:1 are selected.
- [2] Read only.
- [3] When the short form direct mode is selected, 00000-01111 or 10000-11111 are used as RAM addresses.



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

Figure 35. General Instruction Format

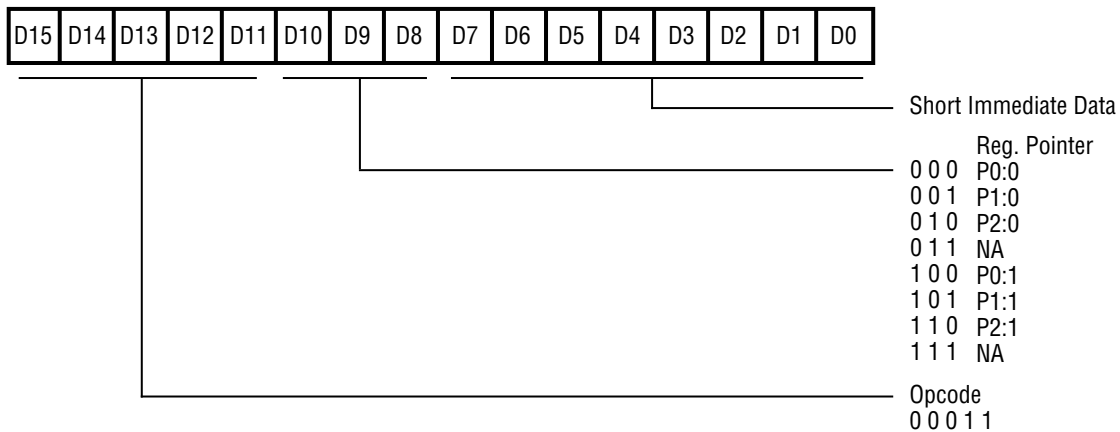


Figure 36. Short Immediate Data Load Format

INSTRUCTION FORMAT (Continued)

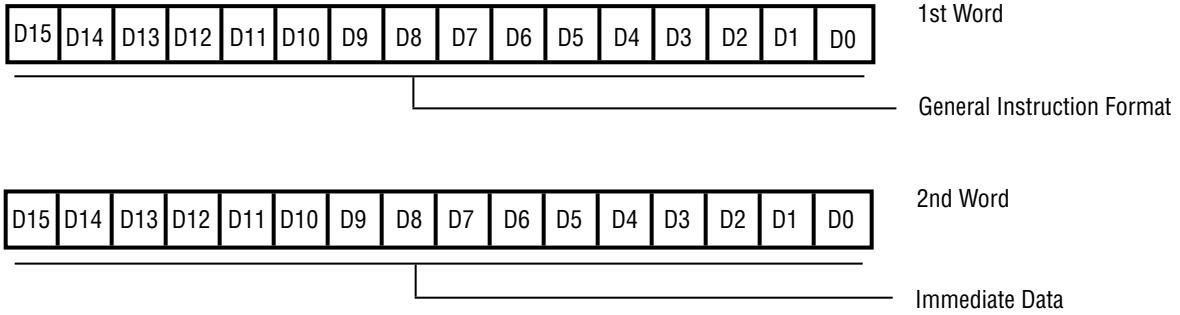


Figure 37. Immediate Data Load Format

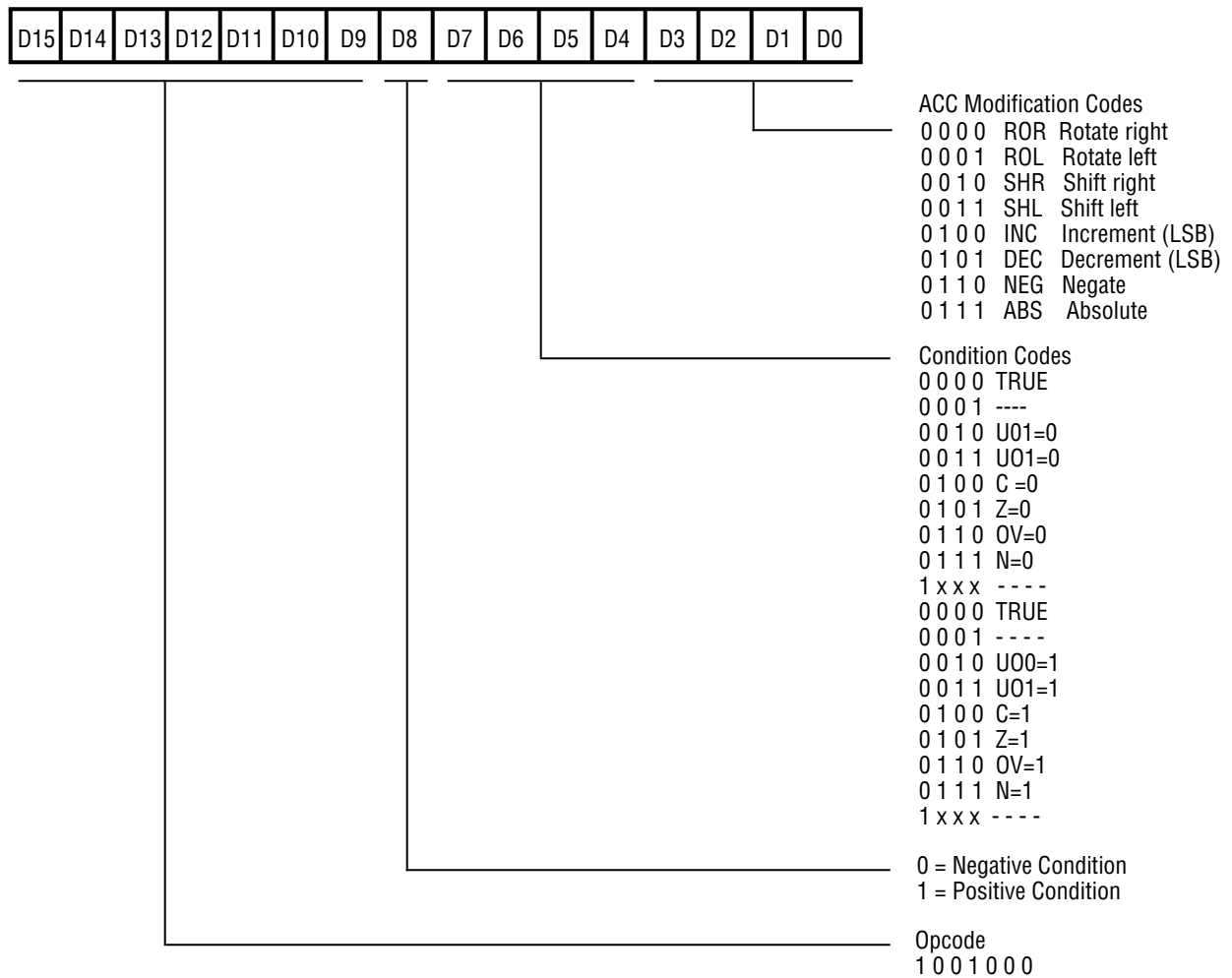


Figure 38. Accumulator Modification Format



## ADDRESSING MODES

This section discusses the syntax of the addressing modes supported by the DSP assembler.

**Table 15. Addressing Modes**

| Symbolic Name                          | Syntax   | Description  |
|--|--|--|
| <pregs>                                | Pn:b   | Pointer Register   |
| <dregs><br>(Points to RAM)             | Dn:b   | Data Register  |
| <hwregs>                               | X,Y,PC,SR,P<br>EXTn,A,BUS                                | Hardware Registers   |
| <accind><br>(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><br>(Points to RAM)            | @Pn:b<br>@Pn:b+<br>@Pn:b-LOOP<br>@Pn:b+LOOP              | Pointer Register Indirect<br>Pointer Register Indirect with Increment<br>Pointer Register Indirect with Loop Decrement<br>Pointer register Indirect with Loop Increment  |
| <memind><br>(Points to Program Memory) | @@Pn:b<br>@Dn:b<br>@@Pn:b-LOOP<br>@@Pn:b+LOOP<br>@@Pn:b+ | Pointer Register Memory Indirect<br>Data Register Memory Indirect<br>Pointer Register Memory Indirect with Loop Decrement<br>Pointer Register Memory Indirect with Loop Increment<br>Pointer Register Memory Indirect with Increment |

There are eight distinct addressing modes for data transfer.

**<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, therefore @P1:1 instructs 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 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. Therefore, @@P1:1 instructs 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 address mode indicates a long immediate load. A 16-bit word can be copied directly from the operand into the specified register or memory.

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

## CONDITION CODES

The following Instruction Description 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.

| Code | 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          |

| Code | 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      | ABSNC,A        |
|       |                  |                      | A             | 1     | 1      | ABSA           |
| ADD   | Addition         | ADD<dest>,<src>      | A,<pregs>     | 1     | 1      | ADDA,P0:0      |
|       |                  |                      | A,<dregs>     | 1     | 1      | ADDA,D0:0      |
|       |                  |                      | A,<imm>       | 2     | 2      | ADDA,#%1234    |
|       |                  |                      | A,<memind>    | 1     | 3      | ADDA,@@P0:0    |
|       |                  |                      | A,<direct>    | 1     | 1      | ADDA,%F2       |
|       |                  |                      | A,<regind>    | 1     | 1      | ADDA,@P1:1     |
|       |                  |                      | A,<hwregs>    | 1     | 1      | ADDAX          |
|       |                  |                      | A,<simmm>     |       |        | ADDA,#%12      |
| AND   | Bitwise AND      | AND<dest>,<src>      | A,<pregs>     | 1     | 1      | ANDA,P2:0      |
|       |                  |                      | A,<dregs>     | 1     | 1      | ANDA,D0:1      |
|       |                  |                      | A,<imm>       | 2     | 2      | ANDA,#%1234    |
|       |                  |                      | A,<memind>    | 1     | 3      | ANDA,@@P1:0    |
|       |                  |                      | A,<direct>    | 1     | 1      | ANDA,%2C       |
|       |                  |                      | A,<regind>    | 1     | 1      | ANDA,@P12+LOOP |
|       |                  |                      | A,<hwregs>    | 1     | 1      | ANDA,EXT3      |
|       |                  |                      | A,<simmm>     |       |        | ANDA,#%12      |
| CALL  | Subroutine call  | CALL[<cc>,<address>] | <cc>,<direct> | 2     | 2      | CALLZ,sub2     |
|       |                  |                      | <direct>      | 2     | 2      | CALLsub1       |
| CCF   | Clear carry flag | CCF                  | None          | 1     | 1      | CCF            |
| CEF   | Clear Carry Flag | CEF                  | None          | 1     | 1      | CEF            |
| COFF  | Clear OP flag    | COFF                 | None          | 1     | 1      | COFF           |
| CP    | Comparison       | CP<src1>,<src2>      | A,<pregs>     | 1     | 1      | CPA,P0:0       |
|       |                  |                      | A,<dregs>     | 1     | 1      | CPA,D3:1       |
|       |                  |                      | A,<memind>    | 1     | 3      | CPA,@@P0:1     |
|       |                  |                      | A,<direct>    | 1     | 1      | CPA,%FF        |
|       |                  |                      | A,<regind>    | 1     | 1      | CPA,@P2:1+     |
|       |                  |                      | A,<hwregs>    | 1     | 1      | CPA,STACK      |
|       |                  |                      | A,<imm>       | 2     | 2      | CPA,#%FFCF     |
|       |                  |                      | A,<simmm>     |       |        | CPA,#%12       |
| DEC   | Decrement        | DEC[<cc>,<dest>]     | <cc>A,        | 1     | 1      | DECNZA         |
|       |                  |                      | A             | 1     | 1      | DECA           |
| INC   | Increment        | INC[<cc>,<dest>]     | <cc>A,        | 1     | 1      | INCPLA         |
|       |                  |                      | A             | 1     | 1      | INCA           |
| JP    | Jump             | JP[<cc>,<address>]   | <cc>,<direct> | 2     | 2      | JPNE,Label     |
|       |                  |                      | <direct>      | 2     | 2      | JPLLabel       |

| Inst.  | Description                   | Synopsis                        | Operands                       | Words | Cycles | Examples           |
|--|-------------------------------|---------------------------------|--------------------------------|-------|--------|--------------------|
| LD   | Loaddestination<br>withsource | LD<dest>,<src>                  | A,<hwregs>                     | 1     | 1      | LDA,X              |
|  |                               |                                 | A,<dregs>                      | 1     | 1      | LDA,D0:0           |
|  |                               |                                 | A,<pregs>                      | 1     | 1      | LDA,P0:1           |
|  |                               |                                 | A,<regind>                     | 1     | 1      | LDA,@P1:1          |
|  |                               |                                 | A,<memind>                     | 1     | 3      | LDA,@D0:0          |
|  |                               |                                 | A,<direct>                     | 1     | 1      | LDA,124            |
|  |                               |                                 | <direct>,A                     | 1     | 1      | LD124,A            |
|  |                               |                                 | <dregs>,<hwregs>               | 1     | 1      | LDD0:0,EXT7        |
|  |                               |                                 | <pregs>,<simn>                 | 1     | 1      | LDP1:1,#%FA        |
|  |                               |                                 | <pregs>,<hwregs>               | 1     | 1      | LDP1:1,EXT1        |
|  |                               |                                 | <regind>,<limn>                | 1     | 1      | LD@P1:1,#1234      |
|  |                               |                                 | <regind>,<hwregs>              | 1     | 1      | LD@P1:1+,X         |
|  |                               |                                 | <hwregs>,<pregs>               | 1     | 1      | LDY,P0:0           |
|  |                               |                                 | <hwregs>,<dregs>               | 1     | 1      | LDSR,D0:0          |
|  |                               |                                 | <hwregs>,<limn>                | 2     | 2      | LDPC,#%1234        |
|  |                               |                                 | <hwregs>,<accind>              | 1     | 3      | LDX,@A             |
|  |                               |                                 | <hwregs>,<memind>              | 1     | 3      | LDY,@D0:0          |
|  |                               |                                 | <hwregs>,<regind>              | 1     | 1      | LDA,@P0:0-LOOP     |
| <hwregs>,<hwregs>  | 1                             | 1                               | LDX,EXT6                       |       |        |                    |
| <p><b>Note:</b> When &lt;dest&gt; is &lt;hwregs&gt;, &lt;dest&gt; cannot be P.</p> <p><b>Note:</b> When &lt;dest&gt; is &lt;hwregs&gt; and &lt;src&gt; is &lt;hwregs&gt;, &lt;dest&gt; cannot be EXTn if &lt;src&gt; is EXTn, &lt;dest&gt; cannot be X if &lt;src&gt; is X, &lt;dest&gt; cannot be SR if &lt;src&gt; is SR.</p> <p><b>Note:</b> When &lt;src&gt; is &lt;accind&gt; &lt;dest&gt; cannot be A.</p> |                               |                                 |                                |       |        |                    |
| MLD  | Multiply                      | MLD<src>,<src>[,<bankswitch>]   | <hwregs>,<regind>              | 1     | 1      | MLDA,@P0:0+LOOP    |
|  |                               |                                 | <hwregs>,<regind>,<bankswitch> | 1     | 1      | MLDA,@P1:0,OFF     |
|  |                               |                                 | <regind>,<regind>              | 1     | 1      | MLD@P1:1,@P2:0     |
|  |                               |                                 | <regind>,<regind>,<bankswitch> | 1     | 1      | MLD@P0:1,@P1:0,ON  |
| <p><b>Note:</b> If src1 is &lt;regind&gt; it must be a bank 1 register. Src2's &lt;regind&gt; must be a bank 0 register.</p> <p><b>Note:</b> &lt;hwregs&gt; for src1 cannot be X.</p> <p><b>Note:</b> For the operands &lt;hwregs&gt;, &lt;regind&gt; the &lt;bankswitch&gt; defaults to OFF. For the operands &lt;regind&gt;, the &lt;bankswitch&gt; defaults to ON.</p>  |                               |                                 |                                |       |        |                    |
| MPYA   | Multiply and add              | MPYA<src>,<src2>[,<bankswitch>] | <hwregs>,<regind>              | 1     | 1      | MPYAA,@P0:0        |
|  |                               |                                 | <hwregs>,<regind>,<bankswitch> | 1     | 1      | MPYAA,@P1:0,OFF    |
|  |                               |                                 | <regind>,<regind>              | 1     | 1      | MPYA@P1:1,@P2:0    |
|  |                               |                                 | <regind>,<regind>,<bankswitch> | 1     | 1      | MPYA@P0:1,@P1:0,ON |
| <p><b>Note:</b> If src1 is &lt;regind&gt; it must be a bank 1 register. Src2's &lt;regind&gt; must be a bank 0 register.</p> <p><b>Note:</b> &lt;hwregs&gt; for src1 cannot be X.</p> <p><b>Note:</b> For the operands &lt;hwregs&gt;, &lt;regind&gt; the &lt;bankswitch&gt; defaults to OFF. For the operands &lt;regind&gt;, the &lt;bankswitch&gt; defaults to ON.</p>  |                               |                                 |                                |       |        |                    |

**INSTRUCTION DESCRIPTIONS** (Continued)

| Inst.   | Description           | Synopsis                         | Operands                       | Words    | Cycles | Examples           |
|---|-----------------------|----------------------------------|--------------------------------|----------|--------|--------------------|
| MPYS  | Multiply and subtract | MPYS<src1>,<src2>[,<bankswitch>] | <hwregs>,<regind>              | 1        | 1      | MPYSA,@P00         |
|   |                       |                                  | <hwregs>,<regind>,<bankswitch> | 1        | 1      | MPYSA,@P1.0,OFF    |
|   |                       |                                  | <regind>,<regind>              | 1        | 1      | MPYS@P1:1,@P20     |
|   |                       |                                  | <regind>,<regind>,<bankswitch> | 1        | 1      | MPYS@P0:1,@P1:0,ON |
| <p><b>Note:</b> Ifsrc1 is&lt;regind&gt;it must be a bank 1 register. Src2's&lt;regind&gt; must be a bank 0 register.</p> <p><b>Note:</b> &lt;hwregs&gt; forsrc1 cannot be X.</p> <p><b>Note:</b> For the operands &lt;hwregs&gt;,&lt;regind&gt; the &lt;bankswitch&gt; defaultsto OFF. For the operands &lt;regind&gt;,&lt;regind&gt; the &lt;bankswitch&gt; defaultsto ON.</p> |                       |                                  |                                |          |        |                    |
| NEG   | Negate                | NEG<cc>,A                        | <cc>,A<br>A                    | 1<br>1   | 1<br>1 | NEGMI,A<br>NEGA    |
| NOP   | Nooperation           | NOP                              | Nbre                           | 1        | 1      | NOP                |
| OR  | BitwiseOR             | OR<dest>,<src>                   | A,<pregs>                      | 1        | 1      | ORA,P0:1           |
|   |                       |                                  | A,<dregs>                      | 1        | 1      | ORA,D0:1           |
|   |                       |                                  | A,<imm>                        | 2        | 2      | ORA,#%2C21         |
|   |                       |                                  | A,<memind>                     | 1        | 3      | ORA,@@P21+         |
|   |                       |                                  | A,<direct>                     | 1        | 1      | ORA,%2C            |
|   |                       |                                  | A,<regind>                     | 1        | 1      | ORA,@P1:0-LOOP     |
|   |                       |                                  | A,<hwregs>                     | 1        | 1      | ORA,EXT6           |
| A,<simmm>   |                       |                                  |                                | ORA,#%12 |        |                    |
| POP   | Popvalue fromstack    | POP<dest>                        | <pregs>                        | 1        | 1      | POP,P0:0           |
|   |                       |                                  | <dregs>                        | 1        | 1      | POP,D0:1           |
|   |                       |                                  | <regind>                       | 1        | 1      | POP@P00            |
|   |                       |                                  | <hwregs>                       | 1        | 1      | POP,A              |
| PUSH  | Pushvalue ontostack   | PUSH<src>                        | <pregs>                        | 1        | 1      | PUSH,P0:0          |
|   |                       |                                  | <dregs>                        | 1        | 1      | PUSH,D0:1          |
|   |                       |                                  | <regind>                       | 1        | 1      | PUSH@P00           |
|   |                       |                                  | <hwregs>                       | 1        | 1      | PUSH,BUS           |
|   |                       |                                  | <imm>                          | 2        | 2      | PUSH,#12345        |
|   |                       |                                  | <accind>                       | 1        | 3      | PUSH@A             |
| <memind>  | 1                     | 3                                | PUSH@@@P00                     |          |        |                    |
| RET   | Returnfromsubroutine  | RET                              | Nbre                           | 1        | 2      | RET                |
| RL  | RotateLeft            | RL<cc>,A                         | <cc>,A                         | 1        | 1      | RLNZA              |
|   |                       |                                  | A                              | 1        | 1      | RLA                |
| RR  | RotateRight           | RR<cc>,A                         | <cc>,A                         | 1        | 1      | RRC,A              |
|   |                       |                                  | A                              | 1        | 1      | RRA                |



| Inst. | Description              | Synopsis        | Operands   | Words | Cycles | Examples       |
|-------|--------------------------|-----------------|------------|-------|--------|----------------|
| SCF   | SetCflag                 | SCF             | Nbre       | 1     | 1      | SCF            |
| SEF   | SetEflag                 | SEF             | Nbre       | 1     | 1      | SEF            |
| SLL   | Shiftleft<br>logical     | SLL             | [<cc>],A   | 1     | 1      | SLLNZ,A        |
|       |                          |                 | A          | 1     | 1      | SLLA           |
| SCFF  | SetOPflag                | SCFF            | Nbre       | 1     | 1      | SCFF           |
| SRA   | Shiftright<br>arithmetic | SRA<cc>,A       | <cc>,A     | 1     | 1      | SRANZ,A        |
|       |                          |                 | A          | 1     | 1      | SRAA           |
| SUB   | Subtract                 | SUB<dest>,<src> | A,<pregs>  | 1     | 1      | SUBA,P1:1      |
|       |                          |                 | A,<dregs>  | 1     | 1      | SUBA,D0:1      |
|       |                          |                 | A,<imm>    | 2     | 2      | SUBA,#%2C2C    |
|       |                          |                 | A,<memind> | 1     | 3      | SUBA,@D0:1     |
|       |                          |                 | A,<direct> | 1     | 1      | SUBA,%15       |
|       |                          |                 | A,<regind> | 1     | 1      | SUBA,@P20-LOOP |
|       |                          |                 | A,<hwregs> | 1     | 1      | SUBA,STACK     |
|       |                          |                 | A,<simmm>  |       |        | SUBA,#%12      |
| XOR   | BitwiseexclusiveOR       | XOR<dest>,<src> | A,<pregs>  | 1     | 1      | XORA,P20       |
|       |                          |                 | A,<dregs>  | 1     | 1      | XORA,D0:1      |
|       |                          |                 | A,<imm>    | 2     | 2      | XORA,#13933    |
|       |                          |                 | A,<memind> | 1     | 3      | XORA,@@P21+    |
|       |                          |                 | A,<direct> | 1     | 1      | XORA,%2F       |
|       |                          |                 | A,<regind> | 1     | 1      | XORA,@P20      |
|       |                          |                 | A,<hwregs> | 1     | 1      | XORA,BUS       |
|       |                          |                 | A,<simmm>  |       |        | XORA,#%12      |

**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, the keywords ON and OFF are used to state

the direction of the switch. These keywords are referenced in the instruction descriptions through the <bank switch> symbol. The most notable capability this provides is that a source operand can be multiplied by itself (squared).

**ABSOLUTE MAXIMUM RATINGS**

| Symbol    | Description        | Min. | Max. | Units |
|-----------|--------------------|------|------|-------|
| $V_{CC}$  | Supply Voltage (*) | -0.3 | +7.0 | V     |
| $T_{STG}$ | Storage Temp       | -65° | +150 | °C    |
| $T_A$     | Oper Ambient Temp  |      | †    | °C    |

**Notes:**

- \* Voltage on all pins with respect to GND.
- † 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 (Figure 41).

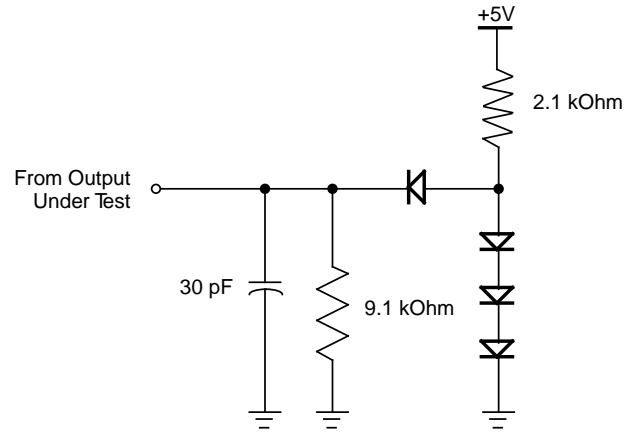


Figure 41. Test Load Diagram

**DC ELECTRICAL CHARACTERISTICS (20 MHZ)**

( $V_{DD} = 5V \pm 10\%$ ,  $T_A = 0^\circ C$  to  $+70^\circ C$ , unless otherwise noted.)

| Symbol   | Parameter                       | Condition            | Standard Temp<br>$T_A = 0^\circ$ to $+70^\circ C$ |                | Extended Temp<br>$T_A = -40^\circ$ to $+85^\circ C$ |                | Units   |
|----------|---------------------------------|----------------------|---|----------------|---|----------------|---------|
|          |                                 |                      | Min.  | Max.           | Min.  | Max.           |         |
| $I_{DD}$ | Supply Current                  | $V_{DD} = 5.5V$      |   | 60             |   | 55             | nA      |
| $I_{DC}$ | DC Power Consumption            |                      |   | 5              |   | 5              | nA      |
| $V_H$    | Input High Level                |                      | 27  |                | 27  |                | V       |
| $V_L$    | Input Low Level                 |                      |   | 8              |   | 8              | V       |
| $I_L$    | Input Leakage                   |                      |   | 10             |   | 10             | $\mu A$ |
| $V_{OH}$ | Output High Voltage             | $I_{OH} = -100\mu A$ |   | $V_{DD} - 0.2$ |   | $V_{DD} - 0.2$ | V       |
| $V_{OL}$ | Input Low Voltage               | $I_{OL} = 2.0mA$     |   | 5              |   | 5              | V       |
| $I_{FL}$ | Output Floating Leakage Current |                      |   | 10             |   | 10             | $\mu A$ |

### AC ELECTRICAL CHARACTERISTICS (20 MHZ)

( $V_{DD} = 5V \pm 10\%$ ,  $T_A = 0^\circ C$  to  $+70^\circ C$ , unless otherwise noted.)

| Symbol            | Parameter                           | Standard Temp<br>$T_A = 0^\circ$ to $+70^\circ C$ |      | Units |
|-------------------|-------------------------------------|---|------|-------|
|                   |                                     | Min.  | Max. |       |
| <b>Clock</b>      |                                     |   |      |       |
| TCY               | Clock Cycle Time                    | 50  |      | ns    |
| Tr                | Clock Rise Time                     |   | 2    | ns    |
| Tf                | Clock Fall Time                     |   | 2    | ns    |
| CPW               | Clock Pulse Width                   | 23  |      | ns    |
| <b>I/O</b>        |                                     |   |      |       |
| DSVALID           | /DS Valid Time from CLOCK Fall      | 0   | 15   | ns    |
| DSHOLD            | /DS Valid Time from CLOCK Rise      | 4   | 15   | ns    |
| EASET             | EA Setup Time to /DS Fall           | 12  |      | ns    |
| EAHOLD            | EA Hold Time from /DS Rise          | 4   |      | ns    |
| RDSET             | Data Read Setup Time to /DS Rise    | 14  |      | ns    |
| RDHOLD            | Data Read Hold Time from /DS Fall   | 6   |      | ns    |
| WRVALID           | Data Write Valid Time from /DS Fall |   | 18   | ns    |
| WRHOLD            | Data Write Hold Time from /DS Rise  | 5   |      | ns    |
| <b>Interrupt</b>  |                                     |   |      |       |
| INTSET            | Interrupt Setup Time to CLOCK Fall  | 7   |      | ns    |
| INTWIDTH          | Interrupt Low Pulse Width           | 1 TCY   |      | ns    |
| <b>Reset</b>      |                                     |   |      |       |
| RRise             | Reset Rise Time                     |   | 1000 | ns    |
| RSET              | Reset Setup Time to CLOCK Rise      | 15  |      | ns    |
| RWIDTH            | Interrupt Low Pulse Width           | 2 TCY   |      | ns    |
| <b>Wait State</b> |                                     |   |      |       |
| WSET              | Wait Setup Time to CLOCK Rise       | 23  |      | ns    |
| WHOLD             | Wait Hold Time from CLOCK Rise      | 1   |      | ns    |
| <b>Halt</b>       |                                     |   |      |       |
| HSET              | Halt Setup Time to CLOCK Rise       | 3   |      | ns    |
| HHOLD             | Halt Hold Time from CLOCK Rise      | 10  |      | ns    |

**AC ELECTRICAL CHARACTERISTICS (20 MHZ)** (Continued)

( $V_{DD} = 5V \pm 10\%$ ,  $T_A = 0^\circ C$  to  $+70^\circ C$ , unless otherwise noted.)

| Analog to Digital          | Min.      | Typical | Max        | Units     |
|----------------------------|-----------|---------|------------|-----------|
| Resolution                 |           | 8       |            | Bits      |
| Integral Non-Linearity     |           | 0.5     | 1          | LSB       |
| Differential Non-Linearity |           | 0.5     | 1          | LSB       |
| Zero Error at 25°C         |           |         | 45         | mV        |
| Supply Range               | 4.5       | 5.0     | 5.5        | Volts     |
| Power Dissipation, No Load |           | 50      | 85         | mW        |
| Clock Frequency            |           |         | 33         | MHz       |
| Input Voltage Range        | VALO      |         | VAHI       | Volts     |
| Conversion Time            |           |         | 2          | $\mu sec$ |
| Input Capacitance on ANA   | 25        |         | 40         | pF        |
| VAHI Range                 | VALO +2.5 |         | ANVCC      | Volts     |
| VALO Range                 | ANGND     |         | ANVCC -2.5 | Volts     |
| VAHI-VALO                  | 2.5       |         | ANVCC      | Volts     |

**DC ELECTRICAL CHARACTERISTICS (10 MHZ)**

( $V_{DD} = 5V \pm 10\%$ ,  $T_A = 0^\circ C$  to  $+70^\circ C$ , unless otherwise noted.)

**fclock = 10 MHz**

| Symbol   | Parameter                       | Condition            | Standard Temp<br>$T_A = 0^\circ C$ to $+70^\circ C$ |      | Extended Temp<br>$T_A = -40^\circ C$ to $+85^\circ C$ |      | Units   |
|----------|---------------------------------|----------------------|---|------|---|------|---------|
|          |                                 |                      | Min.  | Max. | Min.  | Max. |         |
| $I_{DD}$ | Supply Current                  | $V_{DD} = 5.5V$      |   | 30   |   | 55   | nA      |
| $I_{DC}$ | DC Power Consumption            |                      |   | 5    |   | 5    | nA      |
| $V_H$    | Input High Level                |                      | 27  |      | 27  |      | V       |
| $V_L$    | Input Low Level                 |                      |   | 8    |   | 8    | V       |
| $I_L$    | Input Leakage                   |                      |   | 10   |   | 10   | $\mu A$ |
| $V_{OH}$ | Output High Voltage             | $I_{OH} = -100\mu A$ | $V_{DD} - 0.2$                                      |      | $V_{DD} - 0.2$  |      | V       |
| $V_{OL}$ | Input Low Voltage               | $I_{OL} = 2.0mA$     |   | 5    |   | 5    | V       |
| $I_{FL}$ | Output Floating Leakage Current |                      |   | 10   |   | 10   | $\mu A$ |

**AC ELECTRICAL CHARACTERISTICS (10 MHZ)**

 ( $V_{DD} = 5V \pm 10\%$ ,  $T_A = 0^\circ C$  to  $+70^\circ C$ , unless otherwise noted.)

| Symbol            | Parameter                           | Standard Temp<br>$T_A = 0^\circ$ to $+70^\circ C$ |      | Units |
|-------------------|-------------------------------------|---|------|-------|
|                   |                                     | Min.  | Max. |       |
| <b>Clock</b>      |                                     |   |      |       |
| TCY               | Clock Cycle Time                    | 100   |      | ns    |
| Tr                | Clock Rise Time                     |   | 2    | ns    |
| Tf                | Clock Fall Time                     |   | 2    | ns    |
| CPW               | Clock Pulse Width                   | 48  |      | ns    |
| <b>I/O</b>        |                                     |   |      |       |
| DSVALID           | /DS Valid Time from CLOCK Fall      | 0   | 25   | ns    |
| DSHOLD            | /DS Valid Time from CLOCK Rise      | 6   | 25   | ns    |
| EASET             | EA Setup Time to /DS Fall           | 18  |      | ns    |
| EAHOLD            | EA Hold Time from /DS Rise          | 6   |      | ns    |
| RDSET             | Data Read Setup Time to /DS Rise    | 21  |      | ns    |
| RDHOLD            | Data Read Hold Time from /DS Fall   | 9   |      | ns    |
| WRVALID           | Data Write Valid Time from /DS Fall |   | 30   | ns    |
| WRHOLD            | Data Write Hold Time from /DS Rise  | 8   |      | ns    |
| <b>Interrupt</b>  |                                     |   |      |       |
| INTSET            | Interrupt Setup Time to CLOCK Fall  | 11  |      | ns    |
| INTWIDTH          | Interrupt Low Pulse Width           | 1 TCY   |      | ns    |
| <b>Reset</b>      |                                     |   |      |       |
| RRise             | Reset Rise Time                     |   | 1500 | ns    |
| RSET              | Reset Setup Time to CLOCK Rise      | 22  |      | ns    |
| RWIDTH            | Interrupt Low Pulse Width           | 2 TCY   |      | ns    |
| <b>Wait State</b> |                                     |   |      |       |
| WSET              | Wait Setup Time to CLOCK Rise       | 35  |      | ns    |
| WHOLD             | Wait Hold Time from CLOCK Rise      | 2   |      | ns    |
| <b>Halt</b>       |                                     |   |      |       |
| HSET              | Halt Setup Time to CLOCK Rise       | 5   |      | ns    |
| HHOLD             | Halt Hold Time from CLOCK Rise      | 15  |      | ns    |

| Analog to Digital          | Min.      | Typical | Max        | Units |
|----------------------------|-----------|---------|------------|-------|
| Resolution                 |           | 8       |            | Bits  |
| Integral Non-Linearity     |           | 0.5     | 1          | LSB   |
| Differential Non-Linearity |           | 0.5     | 1          | LSB   |
| Zero Error at 25°C         |           |         | 45         | mV    |
| Supply Range               | 4.5       | 5.0     | 5.5        | Volts |
| Power Dissipation, No Load |           | 50      | 85         | mW    |
| Clock Frequency            |           |         | 33         | MHz   |
| Input Voltage Range        | VALO      |         | VAHI       | Volts |
| Conversion Time            |           |         | 2          | µsec  |
| Input Capacitance on ANA   | 25        |         | 40         | pF    |
| VAHI Range                 | VALO +2.5 |         | ANVCC      | Volts |
| VALO Range                 | ANGND     |         | ANVCC -2.5 | Volts |
| VAHI-VALO                  | 2.5       |         | ANVCC      | Volts |

TIMING DIAGRAMS

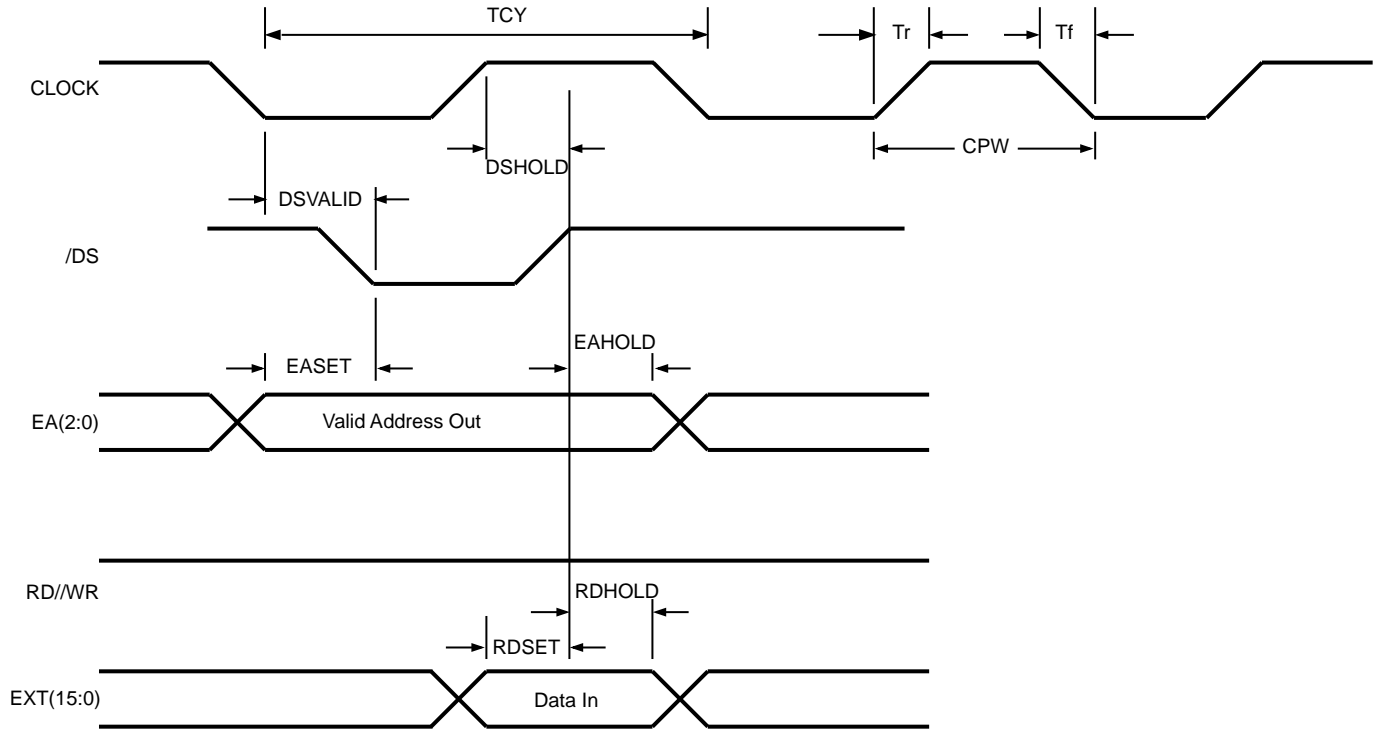


Figure 42. Read Timing

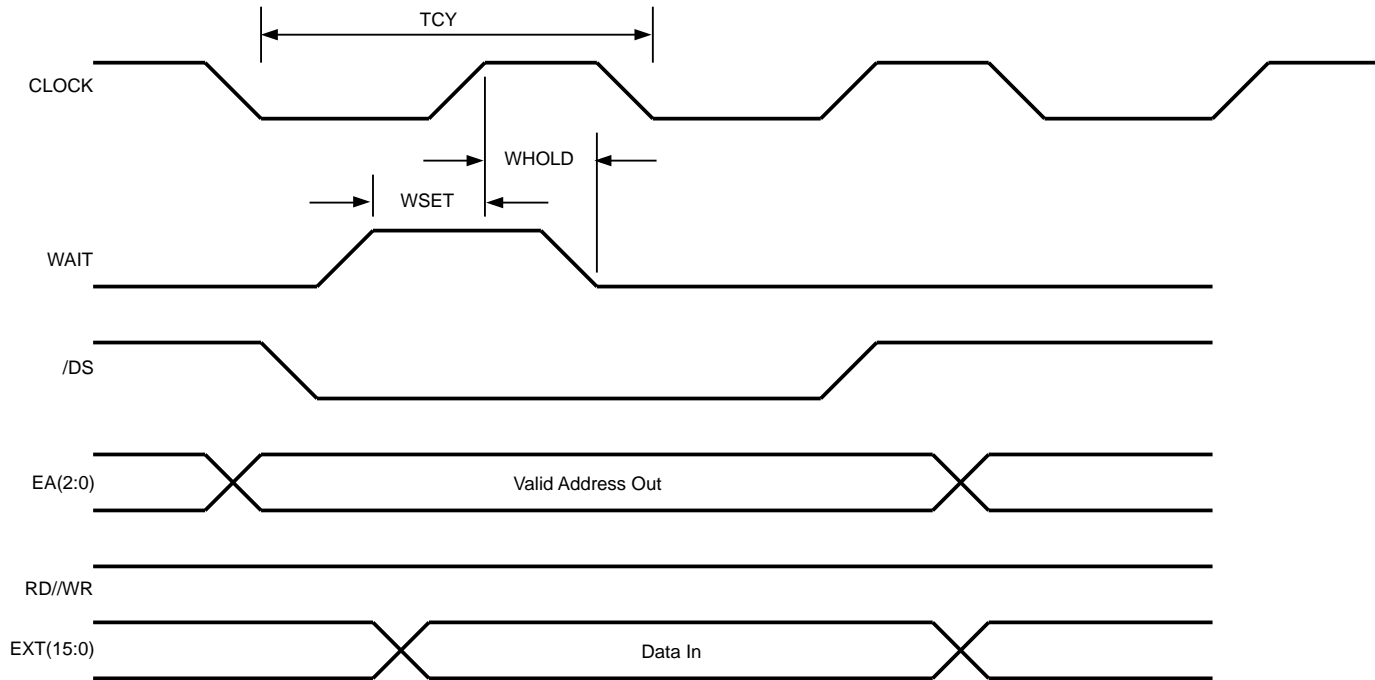


Figure 43. Read Timing Using WAIT Pin

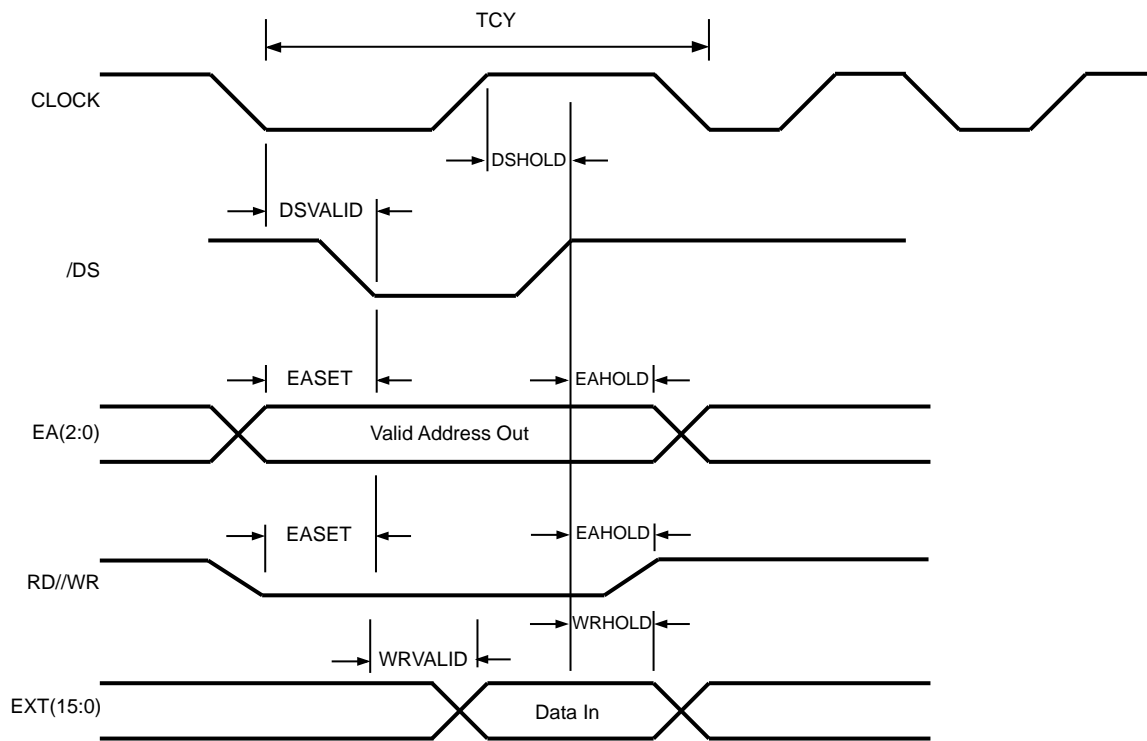


Figure 44. Write Timing

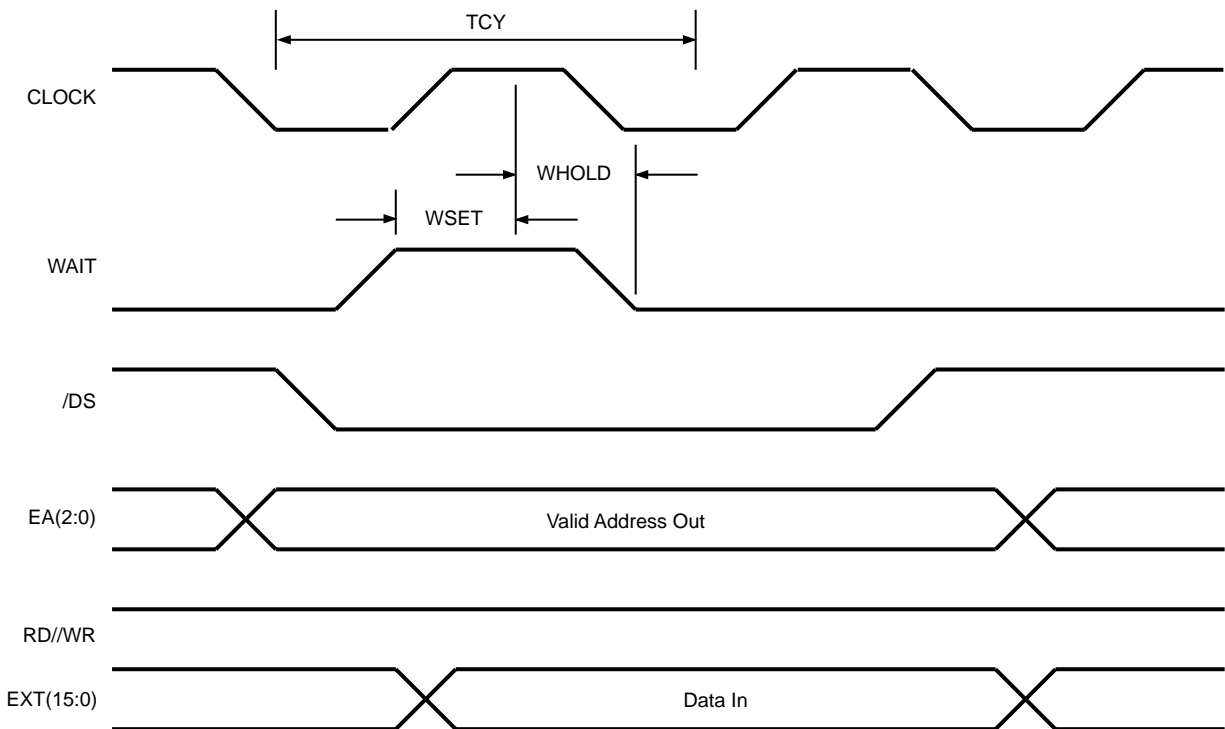


Figure 45. Write Timing Using WAIT Pin

TIMING DIAGRAMS (Continued)

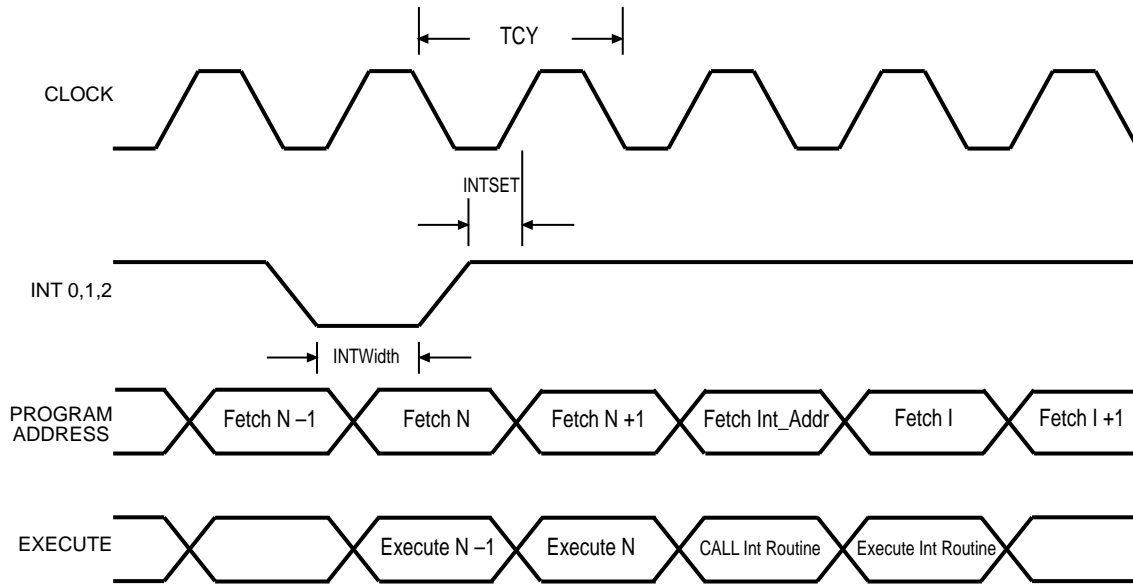


Figure 46. Interrupt Timing

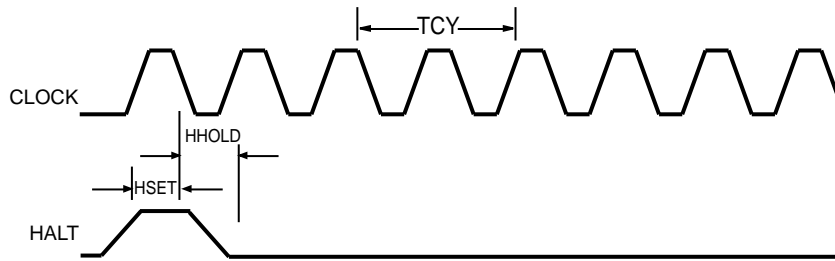
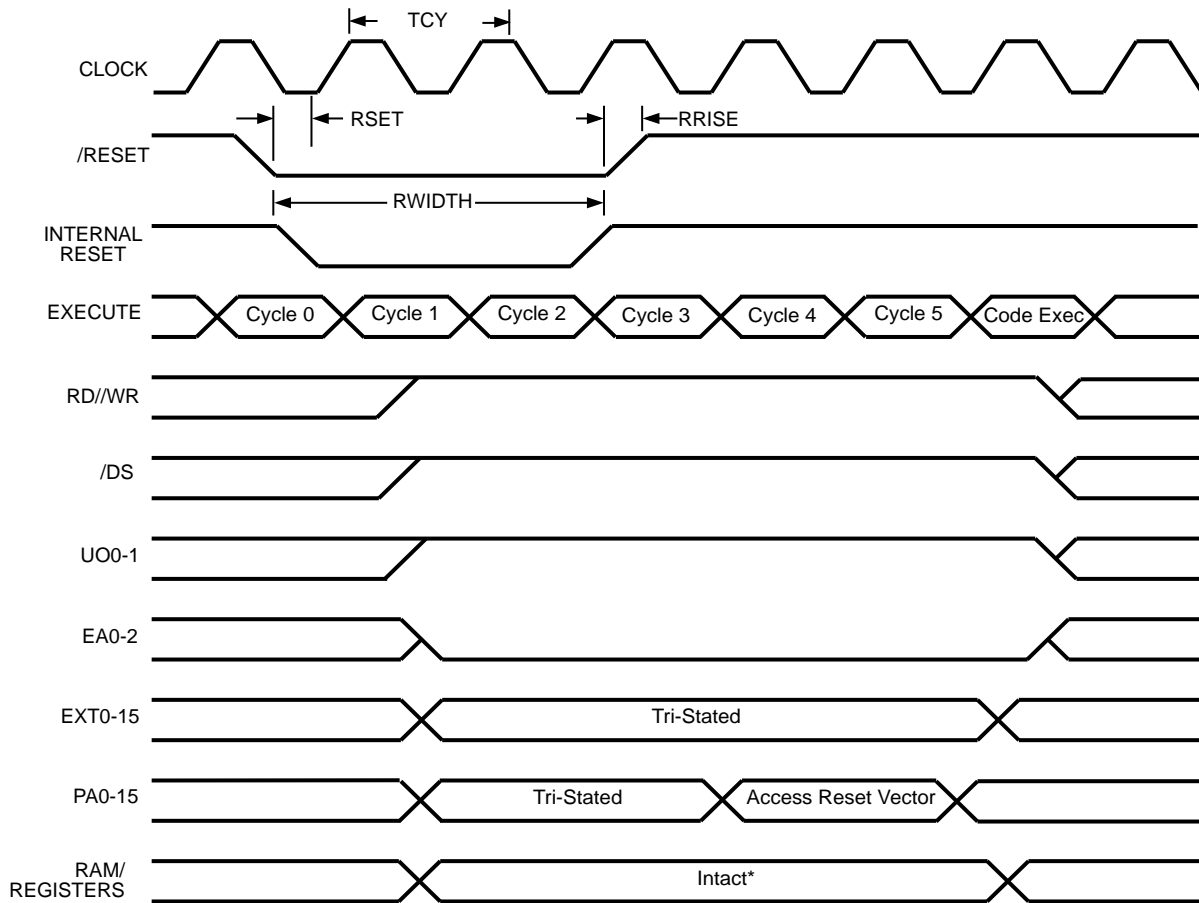


Figure 47. HALT Timing





\* The RAM and hardware registers are left intact during a warm reset. A cold reset will produce random data in these locations. The status register is set to zeroes in both cases.

Figure 48. RESET Timing

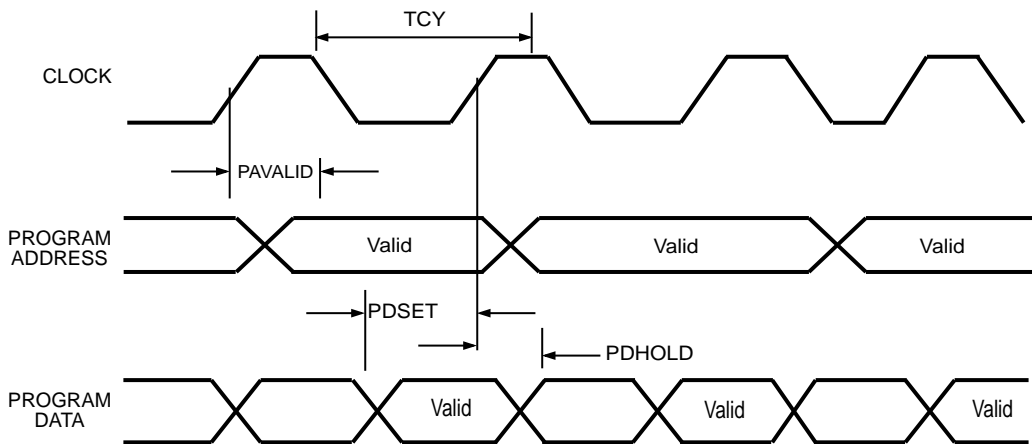
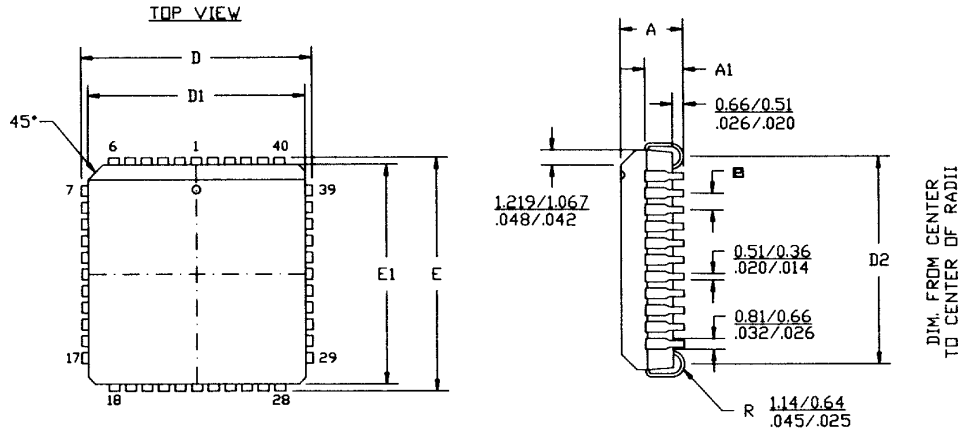


Figure 49. External Program Memory Port Timing

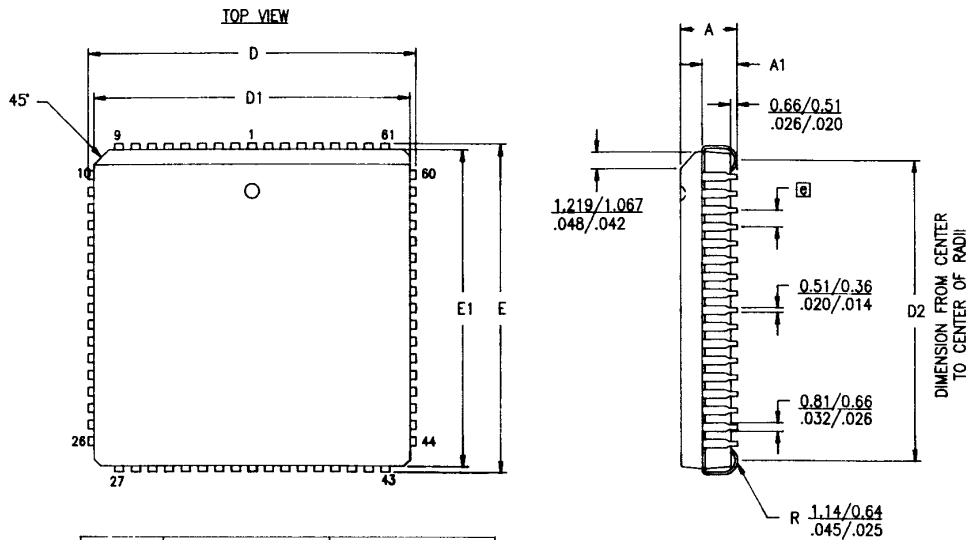
PACKAGE INFORMATION



- NOTES:  
 1. CONTROLLING DIMENSIONS : INCH  
 2. LEADS ARE COPLANAR WITHIN .004 IN.  
 3. DIMENSION :  $\frac{MM}{INCH}$

| SYMBOL | MILLIMETER |       | INCH     |      |
|--------|------------|-------|----------|------|
|        | MIN        | MAX   | MIN      | MAX  |
| A      | 4.27       | 4.57  | .168     | .180 |
| A1     | 2.41       | 2.92  | .095     | .115 |
| D/E    | 17.40      | 17.65 | .685     | .695 |
| D1/E1  | 16.51      | 16.66 | .650     | .656 |
| D2     | 15.24      | 16.00 | .600     | .630 |
| a      | 1.27 TYP   |       | .050 TYP |      |

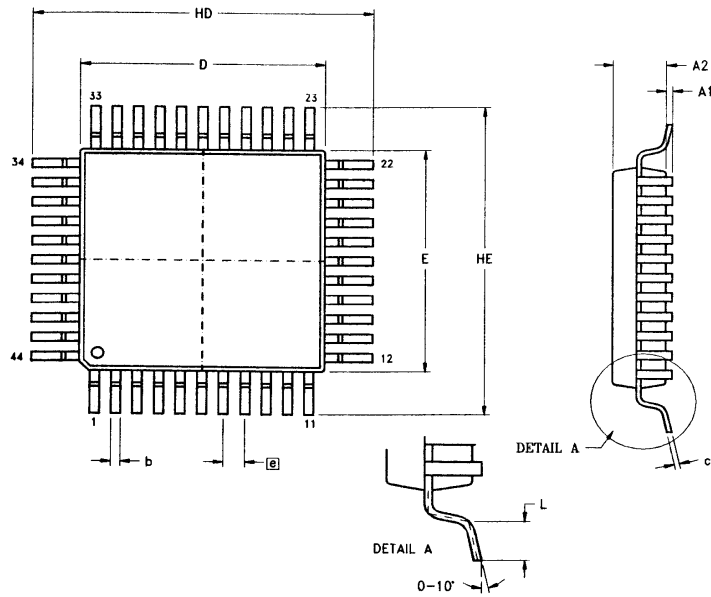
44-Pin PLCC Package Diagram



| 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  |
| a      | 1.27 TYP   |       | .050 TYP |       |

- NOTES:  
 1. CONTROLLING DIMENSIONS : INCH  
 2. LEADS ARE COPLANAR WITHIN .004 IN RANGE.  
 3. DIMENSION :  $\frac{MM}{INCH}$

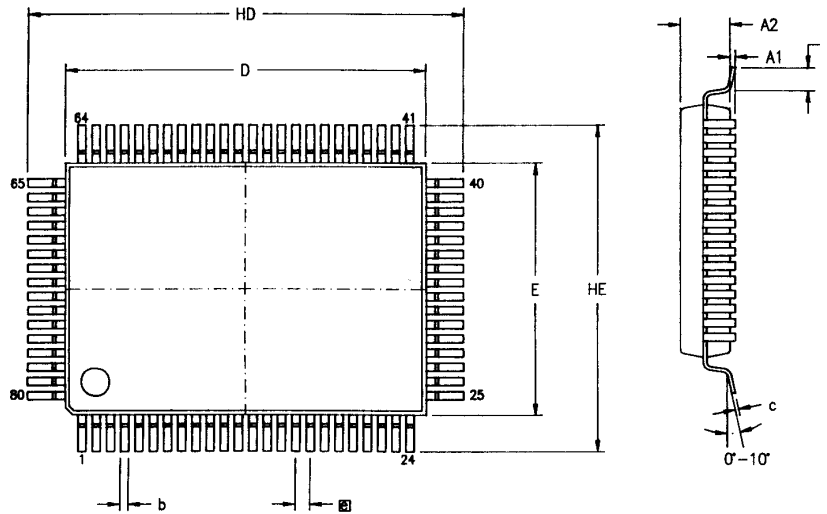
68-Pin PLCC Package Diagram



| SYMBOL | MILLIMETER |       | INCH      |      |
|--------|------------|-------|-----------|------|
|        | MIN        | MAX   | MIN       | MAX  |
| A1     | 0.05       | 0.25  | .002      | .010 |
| A2     | 2.00       | 2.25  | .078      | .089 |
| b      | 0.25       | 0.45  | .010      | .018 |
| c      | 0.13       | 0.20  | .005      | .008 |
| HD     | 13.70      | 14.15 | .539      | .557 |
| D      | 9.90       | 10.10 | .390      | .398 |
| HE     | 13.70      | 14.15 | .539      | .557 |
| E      | 9.90       | 10.10 | .390      | .398 |
| a      | 0.80 TYP   |       | .0315 TYP |      |
| L      | 0.60       | 1.20  | .024      | .047 |

NOTES:  
1. CONTROLLING DIMENSIONS : MILLIMETER  
2. LEAD COPLANARITY : MAX  $\frac{.10}{.004}$

44-Pin QFP Package Diagram

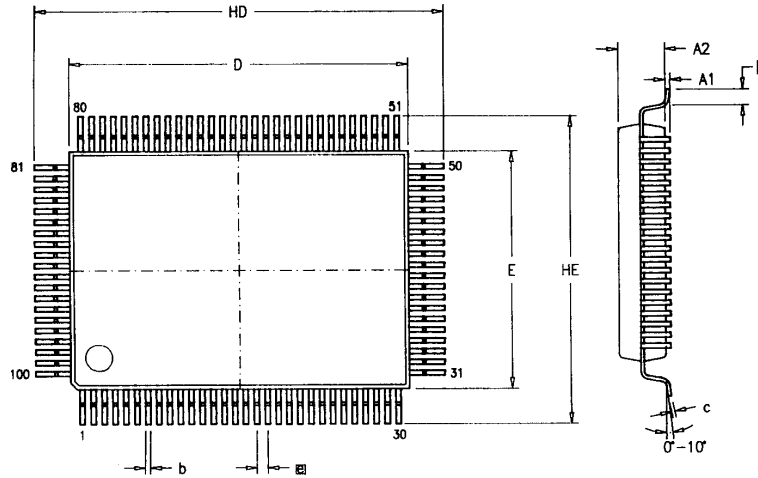


NOTES:  
1. CONTROLLING DIMENSIONS : MILLIMETER  
2. MAX COPLANARITY :  $\frac{.10}{.004}$

| SYMBOL | MILLIMETER |       | INCH     |      |
|--------|------------|-------|----------|------|
|        | MIN        | MAX   | MIN      | MAX  |
| A1     | 0.10       | 0.30  | .004     | .012 |
| A2     | 2.60       | 2.80  | .102     | .110 |
| b      | 0.30       | 0.45  | .012     | .018 |
| c      | 0.13       | 0.20  | .005     | .008 |
| HD     | 23.70      | 24.40 | .933     | .961 |
| D      | 19.90      | 20.10 | .783     | .791 |
| HE     | 17.70      | 18.40 | .697     | .724 |
| E      | 13.90      | 14.10 | .547     | .555 |
| a      | 0.80 TYP   |       | .031 TYP |      |
| L      | 0.70       | 1.20  | .028     | .047 |

80-Pin QFP Package Diagram

PACKAGE INFORMATION (Continued)



NOTES:  
1. CONTROLLING DIMENSIONS : MILLIMETER  
2. MAX COPLANARITY :  $\frac{.10}{.004}$

| SYMBOL | MILLIMETER |       | INCH     |      |
|--------|------------|-------|----------|------|
|        | MIN        | MAX   | MIN      | MAX  |
| A1     | 0.10       | 0.30  | .004     | .012 |
| A2     | 2.60       | 2.80  | .102     | .110 |
| b      | 0.25       | 0.40  | .010     | .016 |
| c      | 0.13       | 0.20  | .005     | .008 |
| HD     | 23.70      | 24.40 | .933     | .961 |
| D      | 19.90      | 20.10 | .783     | .791 |
| HE     | 17.70      | 18.40 | .697     | .724 |
| E      | 13.90      | 14.10 | .547     | .555 |
| ⓐ      | 0.65 TYP   |       | .026 TYP |      |
| L      | 0.70       | 1.20  | .028     | .047 |

100-Pin QFP Package Diagram

**ORDERING INFORMATION**

| <b>Z89323</b>              | <b>Z89373</b>      | <b>Z89393</b>       |
|----------------------------|--------------------|---------------------|
| <b>44-Pin PLCC</b>         | <b>44-Pin PLCC</b> | <b>100-Pin PQFP</b> |
| Z8932320VSC<br>Z8932320VEC | Z8937316VSC        | Z8939320FSC         |
| <b>68-Pin PLCC</b>         | <b>68-Pin PLCC</b> |                     |
| Z893232XVSC<br>Z893232XVEC | Z893731XVSC        |                     |
| <b>44-Pin PQFP</b>         | <b>44-Pin PQFP</b> |                     |
| Z8932320FSC<br>Z8932320FEC | Z8937316FSC        |                     |
| <b>80-Pin PQFP</b>         | <b>80-Pin PQFP</b> |                     |
| Z893232YFSC<br>Z893232YFEC | Z893731YFSC        |                     |

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

**Package**

V = Plastic PLCC  
F = Plastic QFP

**Temperature**

S = 0°C to +70°C  
E = -40°C to +85°C

**Speed**

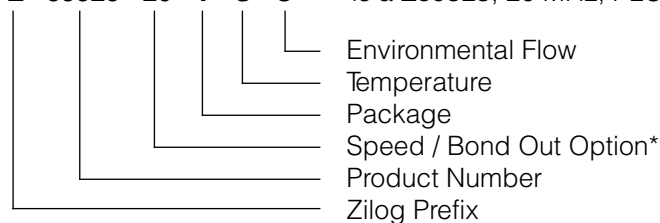
16 = 16 MHz  
20 = 20 MHz

**Environmental**

C = Plastic Standard

**Example:**

**Z 89323 20 V S C** is a Z89323, 20 MHz, PLCC, 0°C to +70°C, Plastic Standard Flow



- \* 2X = 20 MHz, 68-pin PLCC style package
- 20 = 20 MHz, 44-pin package
- 2Y = 20 MHz, 80-Pin PQFP style package
- 1X = 16 MHz, 68-Pin PLCC style package
- 16 = 16 MHz, 44-Pin package
- 1Y = 16 MHz 80-Pin PQFP style package