ZORAN ZR38001

PRELIMINARY

PROGRAMMABLE DIGITAL SIGNAL PROCESSOR

FEATURES

- High Performance
  - 33 MIPs execution of multiple operation 32-bit instruction words
  - Single-cycle execution of three-data-operand instructions
  - 66 MHz internal operation enable two data memory transfers per instruction cycle
  - Zero overhead looping and repeat instructions with four levels of loop nesting
  - 16-word instruction cache
  - Four cycle execution of the radix-2 butterfly inner loop
  - 1024-point radix-2 complex FFT in 658 microseconds
- Low-Cost Single-Chip Solution
  - Large on-chip memories: 4K x 20 RAM for data and 1K x 32 RAM and 12K x 32 ROM for program/data.
  - Internal oscillator with x2 PLL for use with 33 MHz crystal
  - No host microcontroller needed
  - Single, optional external memory for program and/or data
  - Wait-state generation for low-cost external memory
  - 128-pin Plastic Quad Flat Pack (PQFP) packaging
- General-Purpose Architecture and Instruction Set
  - 20 x 20 bit multiplier
  - 48-bit ALU
  - 8-word data register file
  - Separate bi-directional barrel shifter
  - Division primitive instructions
- Powerful Address Generation
  - Large 1-Mword unified program/data address space
  - Dual address generators
  - Modulo addressing
  - FFT bit-reversed addressing
- High-Precision and Wide Dynamic Range
  - 20-bit data precision for 120 dB dynamic range
  - 48-bit fixed-point ALU and dual 48-bit accumulator registers
  - Block floating-point support extends dynamic range
  - Normalization of up to 20 bits in two instruction cycles
  - Support for multiple-precision arithmetic
- Flexible Input/Output
  - 6 half-duplex serial ports with programmable data rates
  - 2 single-bit general purpose interfaces
  - Serial and 32-bit parallel host interface
- Software and Hardware PC Development Environment
  - On-chip ICE support
  - Assembler/Linker
  - Software simulator with Microsoft Windows user interface
  - PC AT bus hardware development board for real-time algorithm execution and debugging

APPLICATIONS

- For low-cost, yet computationally demanding applications in:
  - Multichannel digital audio encoding/decoding
  - Adaptive equalization and cancellation
  - Voice processing
  - Music synthesis
  - High speed facsimile/modem

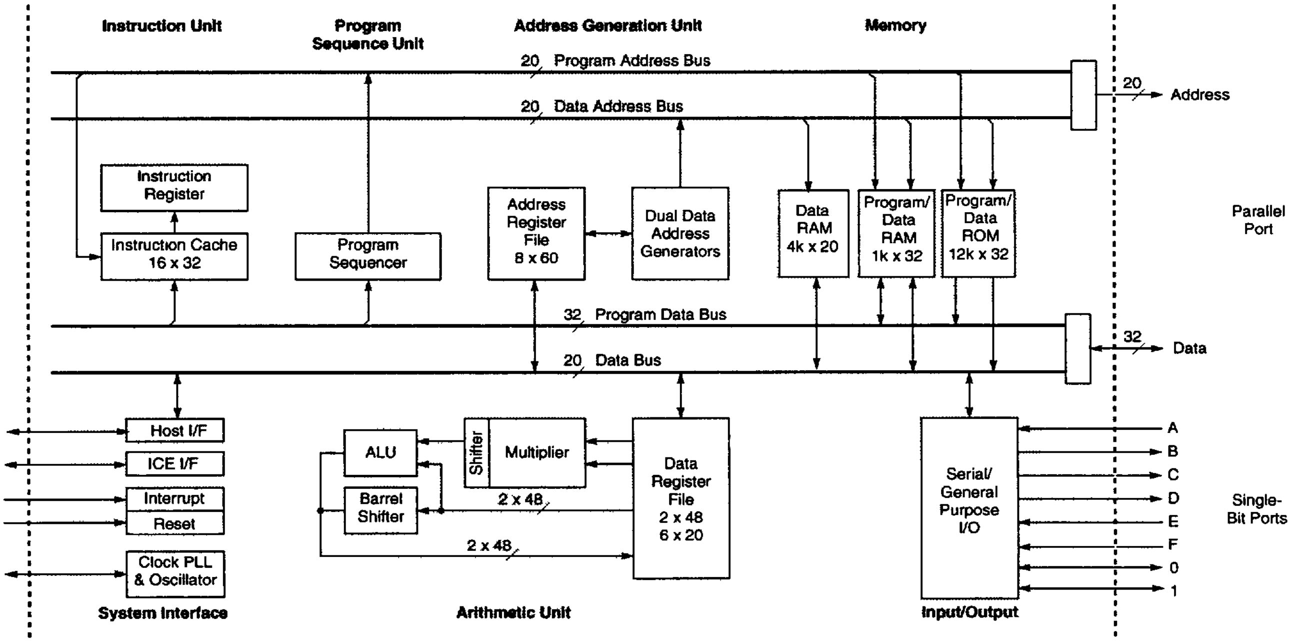


Figure 1. ZR38001 Simplified Block Diagram

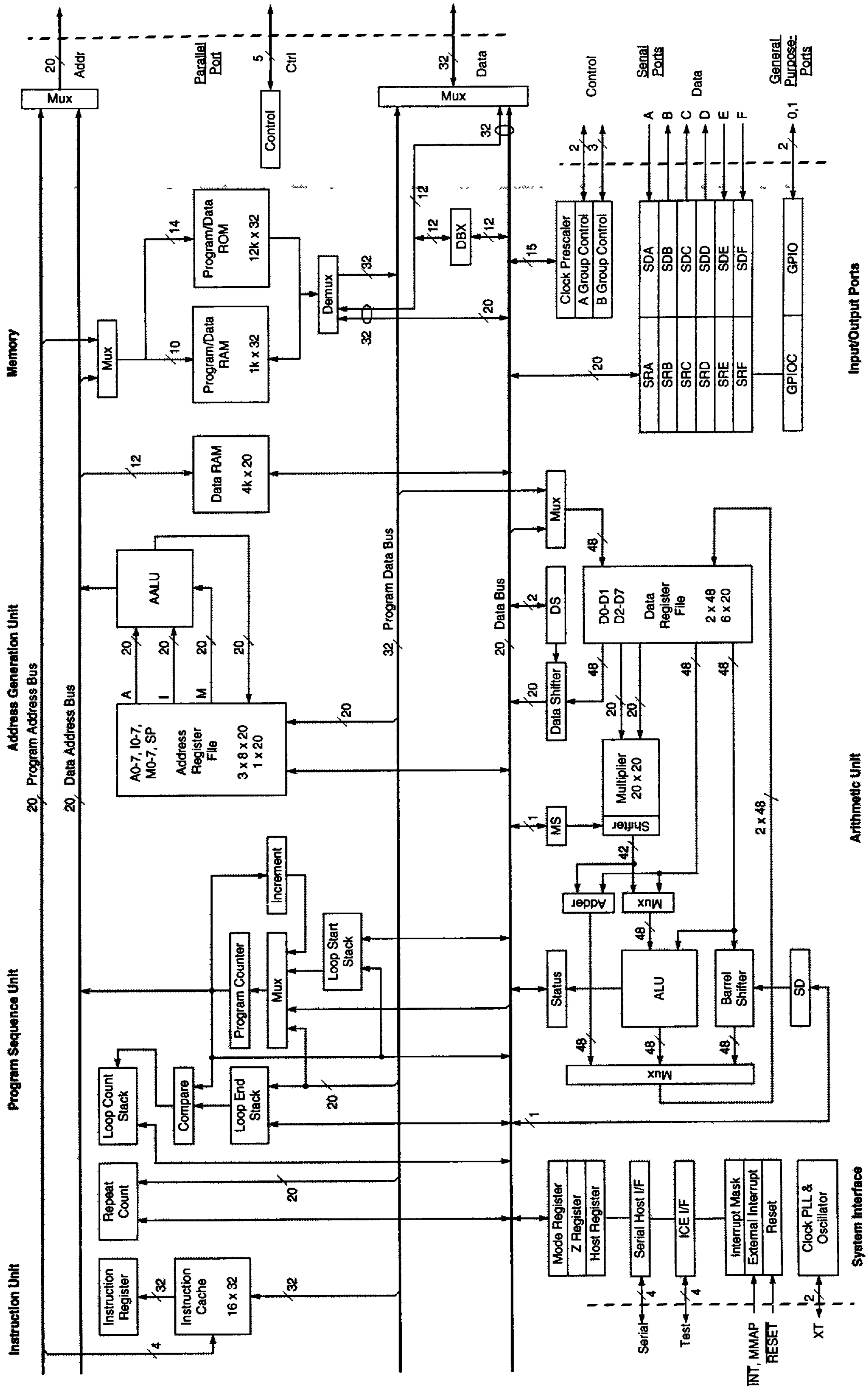


Figure 2. ZR38001 Detailed Block Diagram

GENERAL DESCRIPTION

The ZR38001 fixed-point digital signal processor (DSP) is a high performance, programmable, single-chip microprocessor optimized for audio and speech compression applications. Due to its highly parallel architecture, the ZR38001 is the first processor capable of real-time single-chip decoding of the Dolby Laboratories AC-3 six-channel digital surround sound algorithm. It is also capable of single-chip real-time encoding and decoding of the MPEG-1 audio compression standard.

With its versatile internal architecture (see Figure 2), general purpose instruction set and inherent high speed, the ZR38001 is also capable of executing many other types of algorithms in a wide variety of DSP applications, including low bit-rate, high-quality speech coding (e.g., low delay CELP), high-performance echo cancellation and high-speed modem functions. In addition, the ZR38001 delivers breakthrough performance when executing the Fast Fourier Transform (FFT) algorithm: it is the first fixed-point DSP to execute the 1024-point radix-2 complex FFT benchmark in less than one millisecond (0.658 ms when operating at 33.3 MHz). The radix-2 butterfly is supported by a special FFT primitive instruction, allowing this inner loop to be executed in four cycles (the theoretical minimum for a processor with a single multiplier).

This high level of performance is made possible by the 32-bit wide instruction set which allows the device to perform a large number of concurrent operations. For example, in a single instruction cycle the following operations can be performed:

- Fetch two source operands from registers, execute an arithmetic operation and store the result in a register. In the special case of an FFT primitive instruction, fetch four source operands from registers, execute the arithmetic, and store two results in registers.
- Update two data address pointers
- Perform two parallel data move operations
- Generate the next program address
- Fetch the next program instruction.

Individual bit and immediate data instructions along with the ZR38001's four-level zero-overhead loop and repeat instructions, produce very compact code. Most instructions execute in a single cycle.

The ZR38001 uses an internal clock rate of 66.6 MHz to achieve 33.3 million instructions per second (33.3 MIPS) performance. This allows accessing internal data memory twice per instruction cycle. An internal phase-locked loop (PLL) 2x clock multiplier circuit allows a 33.3 MHz external crystal or input clock to be used.

The ZR38001's optimized 20-bit (120 dB) data precision make it particularly well suited for compact disk-quality audio applications including audio compression, audio mixing, and music synthesis where the 16-bit data precision of conventional fixed-point DSPs is insufficient. Furthermore, by providing high performance support for block floating-point operations to extend dynamic range (including one cycle exponent detection and two cycle normalization), ZR38001-based systems are inherently more cost effective to implement than 24-bit precision fixed-point DSPs which expand dynamic range solely via extended data precision. High performance block floating-point is due to the ZR38001's bi-directional barrel shifter, a feature unavailable on most conventional 16- and 24-bit fixed-point DSPs.

To ease programming and increase speed, the ZR38001 architecture provides a general purpose data register file which can provide up to four source registers and two destination registers per instruction. A total of eight 20-bit data registers are provided, with two registers extended to 48-bits for use as accumulator registers with 8-bit overflow protection.

The ZR38001 also provides a dual address generator and register file capable of generating two independent addresses per instruction cycle. The address generator supports modulo and bit-reversed addressing, in addition to a complete set of pre- and post-modify addressing modes.

Three input and three output serial ports support a wide variety of stereo A/D and D/A converters as master or slave. Clock scalers and frame sizes are fully programmable as are format variations. This large number of on-chip serial ports allows either compression or decompression of up to six channels of audio information.

The ZR38001 is a complete, stand-alone single-chip microprocessor which may be interfaced to external memory and peripherals and/or be a peripheral itself to a host. The only external components that are required for operation are a crystal and tuning capacitors, assuming the on-chip 12k instruction word ROM has been mask programmed with the application code. A large 1k x 32 bit program/data RAM and 4k x 20 bit data RAM are available on-chip. Optionally, the ZR38001's on-chip memory resources can be extended off-chip via its 32-bit multiplexed program/data bus and 20-bit external memory address bus, allowing it to address up to 1 M program/data words in a unified address space. Programmable wait-states accommodate lower-cost slow external memories if desired. If only additional data memory is required, external memory can be limited to a width of 20 bits or less to reduce system cost.

The ZR38001 makes a good system component as well. It can connect to a host processor through a low-cost bit-serial interface (SPI compatible) or a 32-bit parallel interface. External memory can be shared with a host and two single-bit general-purpose registers, along with the external interrupt, make for easy signaling between the ZR38001 and its host.

A sixteen word on-chip instruction cache allows the ZR38001 to efficiently execute looped code stored in off-chip memory. After the first external access of program loops of up to sixteen instructions, subsequent instruction fetches are obtained from the cache. This frees the external bus for a simultaneous external data memory access, if needed, and boosts performance.

FUNCTIONAL DESCRIPTION

ARCHITECTURAL OVERVIEW

Figure 2 shows the detailed functional units of the ZR38001. The data path consists of the Arithmetic Unit, the portions of Memory used for data, and its associated Address Generation Unit. The control path is the Instruction Unit, the portions of Memory used for program, and its associated Program Sequence Unit. The remainder are the Input/Output Ports and the System Interface.

Data flow between data path units is over the single 20-bit Data Bus with a corresponding 20-bit Data Address Bus. Control flow is over the single 32-bit Program Data Bus with a corresponding 20-bit Program Address Bus. These dual data and address buses are multiplexed to single external buses for external memory and parallel I/O. This simple space-efficient bus structure maintains high performance as each internal bus makes two transfers per instruction cycle and each unit is self-contained with its own local memory.

The high performance of the ZR38001 is apparent from the power of the data functional units with their attendant instructions and their being matched by the power of the control functional units and their instructions. Both are described in turn. Data and control paths are assured of working together in parallel because of the fast interconnecting bus structure and the wide-word instruction set controlling both. This view of the operation by function and instruction can confirm basic benchmark performance. In actual designs, the powerful assembler and simulator show the details of the pipelined operations and intermeshing of functions and transfers to assure balanced operation.

ARITHMETIC UNIT

The arithmetic unit performs all data path operations in the processor, using a full-function ALU, a bi-directional barrel shifter and a 20 x 20-bit multiplier, all operating out of the multiport register file. The seven ports allow two transfers in or out of the register file from memory in parallel with a three operand multiplier and ALU operation, including storing the result, every 30 nanoseconds.

In addition to the basic two's-complement arithmetic and logical operations, the 48-bit ALU also can find minimums and maximums, normalize, determine exponents for block floating-point, support multiple precisions and perform division primitives. A further refinement is a butterfly primitive that computes both a product sum and difference using an auxiliary adder. This fetching of four operands, doing a multiply, addition and substraction and storing two results facilitates a very fast 4-cycle radix-2 FFT butterfly. ALU results set appropriate Status register bits in the System Interface, which has sticky bits for multiple precision and array computations. A large class of immediate data logical and arithmetic instructions free register space and reduce instruction count in the bit operations so common in communications coding applications.

The multiplier provides both signed and unsigned operations with an optional one-bit left shift on the output determined by the MS bit in the Mode register. This shift for fractional number alignment preserves the maximum 42 bits of shifted products. The 48-bit barrel shifter does both logical and arithmetic shifts; the SD bit in the Mode register allows a positive shift operator to be interpreted as either a left or a right direction shift. A third Data Shifter provides arithmetic shifts, rounding and limiting when transferring data from the register file onto the Data Bus. The shifting range of 1 bit to the right through to 2 bits to the left is determined by the DS bits in the Mode register.

Two of the eight registers of the register file (D0 & D1) are 48 bits, the remaining six are 20 bits and align as shown in Figure 3. In general all arithmetic unit operations are for implicit 20-bit operands with data being overflowed, limited, rounded or truncated accordingly for registers D2-D7. However when D0 or D1 are the source or destination, then the operations are such as to preserve the full 48-bit precision results in these registers. Likewise, transfers in and out of D0 & D1 with the data buses are extended or reduced based on their being 48-bit operands. These two registers usually serve as the high precision accumulators which are central to most signal processing algorithms. Any of the three fields can be explicitly addressed if the implicit operands are not the desired ones.

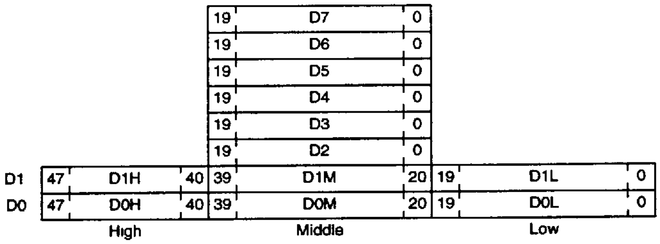


Figure 3. Data Register File

ADDRESS GENERATION UNIT

Data operated on by the Arithmetic Unit is read from and restored to the Data Register File. Register file locations are directly addressed by register fields within the operate field of the instructions. For data transfers with the larger internal and external memories and registers, direct addressing can also be used, but indirect addressing by the Address Generation Unit is often faster and more program memory efficient. The Address Generator can sequentially produce two 20-bit addresses for the two bus transfers possible per cycle and post-modify the same two addresses in the same 30 nanoseconds.

The indirect addresses generated can be linearly incremented or decremented, indexed, bit-reverse indexed or circular with an arbitrary modulus M. This is done in the Address Generation Unit by the Address ALU (AALU) and the Address Register File which is organized as in Figure 4. The next address is produced in a postmodify operation using the appropriate sum of the address register Ax with index register Ix and a compare with modulus register Mx. The five addressing modes in their assembler notation are:

(ax) At the address in address register Ax with no postmodify operation

(ax)+ With a postincrement by one

(ax)- With a postdecrement by one

(ax)+i With a postincrement by the value in index register Ix

(ax)-i With a postdecrement by the value in index register Ix

Note there is no indexing or circular addressing for the stack pointer SP. For M = Hex FFFFF the corresponding A register is incremented in a bit-reverse manner for doing the radix-2 FFT. For an N-point FFT the incrementing index register must be loaded with N/2.

The Address Register File is accessible on the Data Bus and can be used for general purpose registers. Further, they can be loaded with immediate data from the Program Data Bus.

	Stack Pointer							
19	A7	0	19	I7	0	19	M7	0
19	A6	0	19	I6	0	19	M6	0
19	A5	0	19	I5	0	19	M5	0
19	A4	0	19	I4	0	19	M4	0
19	A3	0	19	I3	0	19	M3	0
19	A2	0	19	I2	0	19	M2	0
19	A1	0	19	I1	0	19	M1	0
19	A0	0	19	I0	0	19	M0	0
	Address			Index			Modulus	

Figure 4. Address Register File

MEMORY

Internal

There are three internal on-chip memories, a 4k x 20-bit RAM, a 1k x 32-bit RAM and a 12k x 32-bit mask-programmable ROM. The 20-bit wide RAM is used exclusively as data memory, it transfers on the Data Bus and is addressed only from the Data Address Bus. It is always located in lowest memory address space starting at Hex 00000 up to 00FFF. The other RAM and the ROM are 32-bits wide and can be used for both data and program memory. They are addressable by both the Program and Data Address Buses and are sources, and the RAM a destination, for transfers on both Data and Program Data Buses. The ROM is always at locations Hex E0000 to E2FFF in memory space on both Address Buses. The standard product has the ROM coded with a bootstrap program for loading the main operating program from a host or a byte-wide external ROM when there is no host processor. The upper 11 Kwords of the internal mask-programmable ROM may be user specified for production quantities to minimize cost. The Program/Data RAM is always at locations Hex D0000 to D03FF in memory space on both Address Buses. It provides fast internal memory *without the cost* of a programmed ROM when the ZR38001 is used with a host.

All internal memories have a single port, but consistent with the buses, all can perform two complete operations per instruction cycle. The memories can operate in parallel provided buses are available. Each internal address bus has its own address space, but since the internal memories do not overlap and external memories share a common address bus, all memories can be considered to be in one address space as shown in Figure 5.

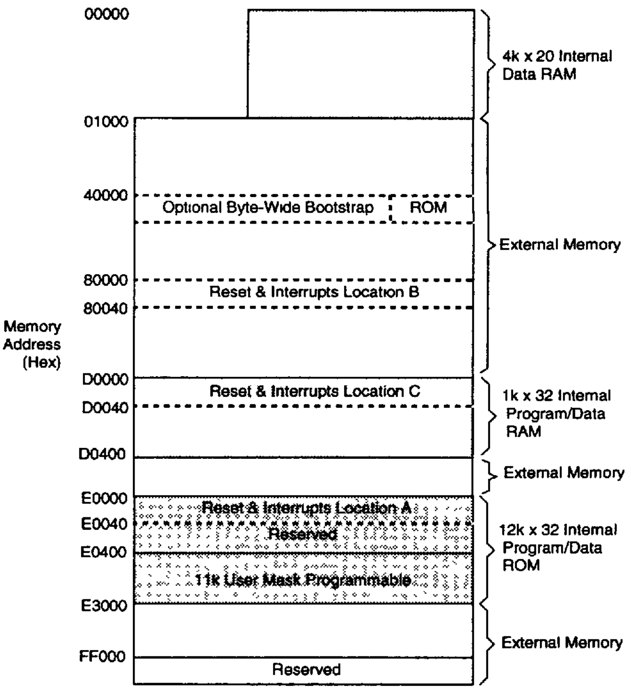

00000
01000
40000
80000
80040
D0000
D0040
D0400
E0000
E0040
E0400
E3000
FF000
Memory Address (Hex)
4k x 20 Internal Data RAM
Optional Byte-Wide Bootstrap ROM
External Memory
Reset & Interrupts Location B
Reset & Interrupts Location C
1k x 32 Internal Program/Data RAM
External Memory
Reset & Interrupts Location A
Reserved
11k User Mask Programmable
12k x 32 Internal Program/Data ROM
External Memory
Reserved

Figure 5. Program/Data Memory Map

External

Program and data memory can be extended externally on the Parallel Port to fill the complete 1024k memory address space shown in Figure 5. Internal address buses and data buses are multiplexed into single buses for external memory so that only one data or program transfer can take place at a time. Also, only a single transfer can be made in each instruction cycle due to the slower cycle-time of external memories. This memory cycle-time can be lengthened by inserting wait-states to allow the use of lower-cost slow memories. The number of wait-states is determined in the Mode register so that external memory operations take one, two or eight 30-ns instruction-cycle-times.

External memory may be 32-bits wide to accommodate the program instruction word. Non-program data is then stored externally in the least significant 20 bits of the memory location. The most significant 12 bits are loaded at the same time from the Data Bus Extension (DBX) register. This register is loaded with the most significant 12 bits of data whenever 32-bit internal or external memory is read. The DBX register can also be loaded or read as a general register with data in the least significant 12 bits.

External memory on the Parallel Port can be a mixture of RAM and the various types of ROM, provided they all operate within the selected wait-state cycle-time. Using a single byte-wide bootstrap ROM for transfer into program RAM is a typical use for such a memory combination. Hex 4000 is the starting address for these ROM bootstrap programs as shown in Figure 5.

Reset and Interrupt Memory Locations

The reset and interrupt vectors occupy a reserved block of memory of 64 (Hex 40) locations. As shown in Figure 5 these can be located at the lowest portion of the on-chip 12k x 32-bit ROM or 1k x 32-bit RAM, or in the external memory locations starting at Hex 80000. This is selected by the MM and PM bits in the Mode Register and the MMAP signal pin as follows:

Table 1. Reset and Interrupt Start Locations

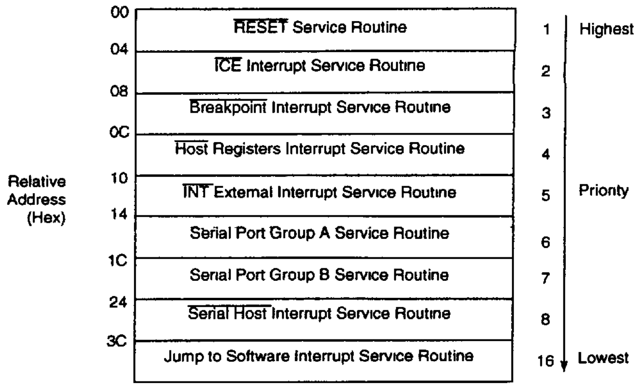
MMAP pin	MM bit	PM bit	Reset & Interrupts Location	Start Address (Hex)
0	0	0	A - Internal ROM	E0000
0	1	0	B - External memory	80000
1	0	0	B - External memory	80000
1	1	0	B - External memory	80000
X	X	1	C - Internal RAM	D0000

PROGRAM SEQUENCE UNIT

All processor operation is governed by the decoded instruction in the Instruction Register (IR). The control flow of the processor is the sequence of instructions that are presented to the IR. The Program Sequence Unit determines this flow by generating the program address to fetch instructions from program memory. This unit in the ZR38001 is a powerful address generator also, often producing a long sequence of operations with a minimum of program memory transfers. Examples of this are the RePeaT and LOOP instructions which allow repeated single and multiple instructions respectively with no instruction overhead. In addition to these instructions, major changes in the control flow are determined by the reset operation, interrupts, branches and subroutines to which the Program Sequence Unit responds.

Reset and Interrupt Operation

Operation of the processor starts with the system asserting the $\overline{\text{RESET}}$ pin. When $\overline{\text{RESET}}$ is asserted, the Mode register is set to Hex 00038 and the Status register is set to Hex 00000. The serial port data registers are all cleared, as are the shift registers for the output serial ports, and the serial port shift register pointers are reset. The modulus registers and the loop end registers are cleared. The cache is invalidated and the program counter is set to Hex 80000 or E0000, before unconditionally jumping to the beginning of the Reset and Interrupt block (shown in Figure 6), to start executing the reset service routine. The complete service routine may be read from an external bootstrap device and in turn, executed.



Relative Address (Hex)		Priority
00	$\overline{\text{RESET}}$ Service Routine	1 Highest
04	$\overline{\text{ICE}}$ Interrupt Service Routine	2
08	$\overline{\text{Breakpoint}}$ Interrupt Service Routine	3
0C	$\overline{\text{Host}}$ Registers Interrupt Service Routine	4
10	$\overline{\text{INT}}$ External Interrupt Service Routine	5
14	Serial Port Group A Service Routine	6
1C	Serial Port Group B Service Routine	7
24	$\overline{\text{Serial Host}}$ Interrupt Service Routine	8
3C	Jump to Software Interrupt Service Routine	16 Lowest

Figure 6. Reset and Interrupt Block Memory Map

Assertion of $\overline{\text{RESET}}$ does not affect the stack pointer, loop start register, loop and repeat count registers, address and index registers, internal RAM and the data registers.

The eight hardware interrupts and the software Interrupt, have their corresponding vector addresses and the priority shown in Figure 6. The priority reflects only the order of servicing when more than one request is pending, and does not determine whether or not a currently executing interrupt service routine will itself be interrupted. All interrupts are collectively disabled with the IE bit and individually enabled with their own mask bits in the auxiliary Interrupt Mask Register (IMR). Before an interrupt service routine is executed, the processor clears the IE bit to disable further interrupts and then pushes the return address and Status register contents on to the stack.

RePeaT and LOOP Instructions

The RePeaT instruction allows the single instruction that follows it to be repeated with no instruction overhead beyond the initial single RePeaT instruction. The Repeat Count (RC) register in the Program Sequence Unit allows up to 2^{20} repeated operations. Likewise, the LOOP instruction allows zero overhead for repeating multiple instruction sequences. The Loop Count (LC), the Loop Start (LS) and the Loop End (LE) registers implement this instruction. Loops may be nested up to four deep with these registers automatically being pushed on their individual stacks. The RC, LC, LS and LE can be a source or destination for general register data transfers, with each transfer in or out of the LE register being the appropriate push or pop operation respectively for their stacks.

Subroutines and Stack Operations

An operational stack is maintained in data memory to service context switches caused by changes in control flow. Interrupts as well as the PUSH, POP and Jump SubRoutine instruction macros use the stack. The Stack Pointer (SP) in the Address Generation Unit determines the stack location, usually in the internal Data or Program/Data RAM for highest speed.

INSTRUCTION UNIT

Pipeline and Cache

Each instruction is fetched from program memory (either cache, internal RAM or ROM, or external memory), decoded in the Instruction register and finally executed. This three stage instruction pipeline takes a minimum of three instruction cycles, but is generally transparent to the user. The delayed branch instructions, however clearly exhibit this pipeline's delay. The pipeline is extended, in effect, whenever there is a requirement for multiple simultaneous accesses to a particular memory resource that cannot be resolved in a single cycle. This occurs, for example, when the next instruction is fetched from external memory and the currently executed instruction specifies an external memory data access, or when an instruction fetch and a dual data move all require access to the internal Program/Data RAM or ROM.

A simple instruction cache is used to minimize the internal and external memory conflicts. The cache is a sixteen location circular buffer with the most recently executed sixteen instructions. Provided these were contiguous in-line instructions then any jump or return to the beginning of a loop within these instructions can continue to execute out of the cache without new fetches from other program memory. This is particularly time saving when the program is in external wait-stated memory. Cache operation is transparent to the user, with its contents being used only when it is known to be valid.

Instruction Set

Each of the instructions of the ZR38001 is a single word in length and except for program flow control instructions, all generally execute in a single cycle unless multiple external memory accesses are required. Much of the power of the processor lies in the parallel operations that go on within one instruction. Instructions are named for the dominant operation that executes, usually an Arithmetic Unit operation or a Program Sequence Unit operation. The instruction set names are summarized in Table 2 by the functional unit. Also listed are the instruction macros which the assembler generates from the basic instructions.

The instructions divide into seven classes or bit-pattern formats summarized in Table 3. It is here that the full power of the ZR38001 is most evident. The first three classes provide the full function of the Arithmetic Unit with its operate fields (Opcode and Operand), but also simultaneous parallel operations. The parallel operate fields (Parallel Opcode and Parallel Operand) specify single and double, direct and indirect transfers with the sources and destinations listed along with address generation modify operations. The last four classes of instructions are for the less used large-field direct data transfers and program control.

There are six sub-classes for the parallel transfers, the most powerful being the last which can do the following four types of sequential dual transfers:

Table 2. Instruction Set Summary

INSTRUCTIONS				
	Arithmetic Unit			
Arithmetic	Logic	Multiplier	Address Generation Unit	Program Sequence Unit
ABS	AND, ANDI	BFY	MOVE	Delayed Branch
ADD, ADDI	DEC	MADD		Conditional DB
ASHift, ASHI	INC	MNEG		Jump to SW Interrupt
CMP, CMPI	LSHift, LSHI	MSUB		LOOP
CMPA	OR, ORI	MUL, MULI		RePeaT
CMPZ	XOR, XORI	MULSU		
DIVS	NOP	MULUU		
DIVU	CLRBit			
MOVEMAX	SETBit			
MOVEMIN	TSTBit			
NEG				
NORM				
NORMMAX				
SUB				
MACROS				
CLeaR			POP	DO
			PUSH	Jump Conditional
				JuMP unconditional
				Jump SubRoutine
				ReTurn Interrupt
				ReTurn Subroutine

First transfer	Second transfer
Data register to memory	Data register to memory
Data register to memory	Memory to data register
Memory to data register	Data register to memory
Memory to data register	Memory to data register

All memory references in this subclass are indirect and with possible address modification.

Table 3. Instruction Class Summary Table

I Single operand ALU operations with parallel transfer operations

Class Code	Opcode	Operand	Parallel Opcode	Parallel Operands

II Two operand ALU operations with parallel transfer operations

Class	Opcode	Operands	Parallel Opcode	Parallel Operands

III Three operand ALU operations with parallel transfer operations

Opcode	Operands	Parallel Opcode	Parallel Operands

IV Load/Store direct

Class Code	Register	Address

V Load immediate

Class Code	Register	Data

VI Conditional delayed branch

Class Code	Condition Code	Address

VII Repeat immediate

Class Code	Data

where for classes IV, V and the single transfers of classes I, II and III, the possible source and destination registers are:

D0-D7	MS	A0-A7	RC	STATUS	PC	SRA-SRF
D0L,M,H	SD	I0-I7	LC	MODE	SP	GPIO
D1L,M,H	DS	M0-M7	LS	IE	Z	HREG
		DBX	LE	WAIT		HREGI

and where for parallel transfer operations there are the following sub-classes:

i Register-to-register transfers (including auxiliary), single

ii Load register immediate (6-bits), single

iii Register-to-memory transfers, single

iv Memory-to-register transfers, single

v Address modify, single and dual

vi Single and dual transfers including memory-to-memory and with optional address modify

INPUT/OUTPUT PORTS

External memory, peripherals and any parallel connection to a host processor are made through the input/output ports. There is a single 32-bit parallel port and eight single-bit ports; six transfer data in various bit-serial formats and two as programmable bits in an internal register.

Parallel Port

The external memory and parallel host interface consists of the 20-bit address bus A(19-0), the 32-bit bi-directional data bus D(31-0), and the control signals $\overline{\text{CS}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{HREQ}}$, and $\overline{\text{HACK}}$.

In normal operation when the ZR38001 is a master, $\overline{\text{CS}}$ is asserted whenever there is an access to external memory or registers for the duration of the access cycle. $\overline{\text{RD}}$ is asserted during an external read cycle, and can be used as an output enable for memory or a read strobe for registers. $\overline{\text{WR}}$ is asserted during an external write cycle, and can be used as a write enable for memory or a write strobe for registers.

The ZR38001 can generate wait-states for use with slow external memory using the WAIT field of the Mode register. In access cycles with wait-states, the timing relationship of the transitions of the memory interface signals remain the same as in a zero-wait cycle, but all are stretched by the specified number of instruction clock periods (1 or 7).

During an instruction cycle in which there is no external data access or external instruction fetch, the $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals are not active. However, the address bus continues to be driven with the internal instruction fetch address, and if there is an access to a cached external memory word, $\overline{\text{CS}}$ and $\overline{\text{RD}}$ are also active.

When $\overline{\text{RESET}}$ is asserted, the address and data buses and control signals $\overline{\text{CS}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ are all set to a high-impedance state.

The ZR38001 can release being master allowing a host to have access to the external memory or to make parallel transfers with its internal Host Registers (HREG and HREGI). Asserting $\overline{\text{HREQ}}$ floats the address output, makes the data and control pins inputs and asserts $\overline{\text{HACK}}$ after the next instruction cycle. Multiple read or write operations with HREG and HREGI can then take place. Each will cause an interrupt and the HW bit in the Status Register indicates when it has been a write operation. Releasing $\overline{\text{HREQ}}$ will return mastership to the ZR38001.

Serial Ports

The serial ports are flexible on the ZR38001 to serve a wide variety of applications and peripheral device conventions. The three inputs and three outputs may be variously grouped to share two sets of common control signals, each being a source or a slave. Other selections are word or frame synchronization, frame size and either 16- or 20-bit word transfers. A master clock output has a programmable rate as do the two group clocks. The time-division-multiplex (TDM) format, the I^{2}S format and the 20-

bit word in an LSB justified 32-bit frame of the Japanese DAC format are all supported.

Ports A, E, and F are always data inputs while B, C and D are always data outputs. They may be configured in two groups with shared clocking: all inputs and all outputs, or as two groups with both inputs and outputs in each group. This selection is made by the AB-bit in the Mode register. The B group, which always has a predominance of outputs, is unique in that when operating as a source to D/As, its clock outputs can also be derived from an externally supplied x256 master clock input (SCKIN).

Transfers are on the appropriate edge of the bit-rate clocks (SCKA and SCKB) with the most significant bit being shifted first into or out of the double buffered shift registers. Word boundaries are signaled by a single-bit-duration frame signal (FSA and FSB) for each word or an alternating word signal (WSA and WSB) indicating left or right channel, even or odd word. The signal type is selected independently for each group as is the word length of 16 or 20 bits and the frame size of 16 to 256 bits per frame. The Word Select bits in the Status register reflect when the left or right channel is being transferred for each group. The WS/FS signals maybe delayed by one bit interval for the non-I^{2}S format. Completed frame transfers for each group are indicated to the processor by a vectored interrupt when individually enabled. An exception is for TDM where there is an interrupt for each word within a frame.

Each group can be a source or a slave as selected in the auxiliary Serial Port Mode register. When a source, the clock rates are independently programmable sub-multiples of the internally generated master clock. The B group clocks can come from the external master clock input (SCKIN) as well. If this input is not used the pin may be selected as an output for the internally generated master clock. A programmable pre-scaler determines its rate as a sub-multiple of the processor clock.

General Purpose Ports

The two single-bit general purpose ports may be individually selected as an input or output in the GPIOC auxiliary register. If configured as an input, its sampled state may be read in the GPIO general register, or if an output, its state may be set by writing to the GPIO register.

SYSTEM INTERFACE

The system interface consists of all external signal functions other than I/O plus the general and auxiliary registers which are associated with more than one functional unit's operation.

General Registers

In addition to the primary data flow and control flow of instructions between functional units on the two data buses, there is the secondary control flow between the general and auxiliary registers for initialization and maintenance of operation. The following general registers are directly addressable on the Data Bus for register-to-register, memory-to-register or register-to-memory parallel transfers.

Mode Register

The Mode register is a source or destination register containing 17 bit-fields that define the basic processor configuration. They tend to be set once at initialization and not change. The interrupt enable (IE), the wait-state selection (WAIT), the multiplier shifter (MS), the register file data shifter (DS) and the shift direction (SD) bits that may change during processing are also

Table 1. Serial Ports Function Summary

Function	A Group	B Group
Grouping: AB = 0	3 Inputs (A, E, F)	3 Outputs (B, C, D)
Grouping: AB = 1	2 Inputs (A, E), 1 Output (D)	1 Input (F), 2 Outputs (B,C)
Word size	16 or 20 bits	16 or 20 bits
Frame size (bits/frame)	16, 32, 64, 128, 192, 193, 256	16, 32, 64, 128, 192, 193, 256
Synchronization	Word or Frame	Word or Frame
Source and slave clocking modes	Yes	Yes
External master clock input	No	Yes
Internal master clock output	No	Yes
Pre-scaler for internal master clock	5-bit Counter	
Internal clock scaler	11-bit Counter	11-bit Counter
I^{2}S format	Yes	Yes
TDM format	Yes	Yes
EIAJ DAC format	Yes	Yes

individually addressable as registers. The IE, IM, AM and BM bits are also accessible in the Interrupt Mask Register (IMR). The Mode register is defined as follows:

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE	IM	AM	BM	0	0	PM	0	WFA	WFB	AB	AW	BW	MM	WAIT		MS	DS		SD

IE **Interrupt Enable** when set enables all unmasked interrupts. When cleared, disables all interrupts.

IM **INT Mask** when set enables the external interrupt input.

AM **A Mask** when set enables the A Group serial ports interrupt.

BM **B Mask** when set enables the B Group serial ports interrupt.

PM **Program Memory** selection. When set the Reset and Interrupt Block is located in internal RAM. When cleared its location is determined by the MM bit and the MMAP pin.

WFA **Word/Frame A** group serial port synchronization mode bit. Word synchronization when set, Frame synchronization when cleared.

WFB **Word/Frame B** group serial port synchronization mode bit. Word synchronization when set, Frame synchronization when cleared.

AB **A/B** groupings of serial ports. When set, A Group is ports A, D & E; B Group is ports B, C & F. When cleared, A group is ports A, E & F; B Group is ports B, C & D.

AW **A Word** precision. When set, A group serial port transfers are 16-bit words, when cleared are 20-bit.

BW **B Word** precision. When set, B group serial port transfers are 16-bit words, when cleared are 20-bit.

MM **Memory Map** selection when the PM bit is cleared. When MM is set the Reset and Interrupt Block is located in external memory. When cleared and MMAP pin is not asserted, the block is in internal ROM. When cleared and MMAP is asserted the block is in external memory.

WAIT **Wait**-state selection for external memory. Wait = 00 for no wait-states, = 01 for one wait-state and = 11 for seven wait-states (a total of eight instruction cycles for an external memory operation).

MS **Multiplier Shifter.** When set specifies 1-bit left arithmetic shift on multiplier output, when cleared there is no shifting.

DS **Data Shifter** on transfers out of the Arithmetic Unit from the Data Register File.

DS	Arithmetic Shift
00	No shift
01	Left shift by one
10	Left shift by two
11	Right shift by one

SD **Shift Direction** on the barrel shifter. When cleared a positive shift code corresponds to a left shift, when set a positive shift code corresponds to a right shift.

Status Register

The Status register is a source or destination register containing 11 bits that reflect the state of the processor following each instruction cycle. They affect the conditional program control of the processor. The least significant 8 bits reflect arithmetic and logical operation results from the ALU, multiplier, barrel shifter or on transfers that involve scaling or limiting. The remaining three involve word identification on the serial and host ports. The Status Register is defined as follows:

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	HW	WSA	WSB	Q	SS	SL	SV	V	C	N	Z

HW **Host Write** indicated the host interrupt is due to a write operation to the Host register.

WSA **Word Select A** bit indicates Left channel data is being input if cleared or Right channel data if set, on the A Group serial ports.

WSB **Word Select B** bit indicates Left channel data is being output if cleared or Right channel data if set, on the B Group serial ports.

Q **Quotient** bit is used with the divide iteration instructions.

SS **Sticky Scaling** bit is set if any data transferred through the Data Shifter has a magnitude of greater than 0.25. This indicates the potential for overflow in the next pass of an FFT on the data. It is cleared only by a processor RESET or by an explicit instruction to clear it.

SL **Sticky Limiting** bit is set whenever limiting takes place in the Arithmetic Unit or during a data transfer through the Data Shifter. It is cleared only by a processor RESET or by an explicit instruction to clear it.

SV **Sticky Overflow** bit is set whenever the Overflow bit is set except for the compare instructions. It is cleared only by a processor RESET or by an explicit instruction to clear it.

V **Overflow** bit is set if an overflow results from any operation in the Arithmetic Unit. Overflow is determined if any number can not be properly represented in its destination register.

C **Carry** bit is set if a carry results from an addition or a borrow results from a subtraction in the ALU, or results from shifts in the barrel shifter of the Arithmetic Unit.

N **Negative** bit is set if the most significant bit of the destination register is set, otherwise it is cleared.

Z **Zero** bit is set if the entire result of an Arithmetic Unit operation in its destination register is zero.

DBX Register

The data bus extension register (DBX) is a 12-bit register that permits full use of the 32-bit internal memories as data or the external memory for data storage when that memory is 32-bits wide. When reading data from 32-bit wide external or internal memory to a 20-bit register, the least significant 20 bits are loaded into the destination register. The most significant 12 bits are loaded into the DBX register. When writing data from a 20-bit register to the 32-bit external memory, the least significant 20 bits are driven by the specified source register, while the most significant 12 bits are driven by the DBX register. When the DBX is specified as the destination or source in a transfer, the least significant 12 bits are read into or loaded from the DBX.

Z Register

The Z register is a 20-bit general register on the Data Bus that can be a source or destination for parallel transfers. However it is used by the assembler for the JSR macroinstruction so it must be used with caution.

Address Register File

Each of the twenty-five 20-bit registers of the Address Generator Unit's register file can be a source or destination for parallel transfers on the Data Bus. If not used for address generation they may be used as general registers.

Auxiliary Registers

In addition to the primary data flow and control flow of instructions between functional units on the two data buses, there is the secondary control flow with the general and auxiliary registers for initialization and maintenance of operation. The following auxiliary registers are not directly addressable on the Data Bus, but are accessed by register-to-register parallel transfers.

Table 5. Auxiliary Registers

Name	Description
ICR	ICE control
IDR	ICE data
ISR	ICE status
IRR	ICE response
BKP1	Breakpoint 1 instruction address
BKP2	Breakpoint 2 instruction address
BKP3	Breakpoint 3 data address
BCT1	Breakpoint 1 counter
BCT2	Breakpoint 2 counter
BCR	Breakpoint control
BSR	Breakpoint status
IMR	Interrupt mask
GPIOC	General purpose I/O control
SPMODE	Serial ports mode
SPPR	Serial ports prescaler
SPAS	Serial ports A scaler
SPBS	Serial ports B scaler
SPIMODE	Serial host interface mode
SPISTAT	Serial host interface status
SPITX	Serial host interface transmit
SPIRX	Serial host interface receive

Serial Host Interface

The serial host interface provides a low-cost, low-bandwidth interface to a host processor for down-loading RAM programs and basic operating commands. The ZR38001 always operates as a slave and transfers are program interrupt driven. The signals and protocol are the industry standard serial peripheral interface (SPI compatible). Transfers are full-duplex serial between eight-bit word registers. The bootstrap ROM program will expect the host to down-load RAM program through the serial host interface if it does not find an external byte-wide EPROM at reset time.

Interface signals are data input (SI), data output (SO), clock input (SCK) and slave select ($\overline{\text{SS}}$) which provides transfer synchronization as well. Four auxiliary registers receive data (SPIRX), transmit data (SPITX), determine clock polarity (SPIMODE) and provide control flags (SPISTAT) for the interrupt driven operation.

In Circuit Emulation Interface

The ZR38001's In Circuit Emulation (ICE) capability for both hardware and software debugging is provided through four test pins (TDI, TDO, TCK,TMS) using a standard JTAG interface. This interface is serviced by routines in the on-chip Program/Data ROM and the highest priority interrupt. This provides register and memory read and set commands for hardware debugging. Three breakpoint address-detection registers and two count registers with interrupt additionally provide for real-time program debugging capability in the ICE.

Reset and Interrupt Inputs

The processor can be reset only by asserting the $\overline{\text{RESET}}$ signal input pin externally. On the initial power-up it must be asserted for a minimum of 128 clock cycles with proper supply voltage operating conditions. Operation starts 4096 cycles after the rising edge. After power-up, any reset must be asserted for at least five clock cycles but less than 128 clock cycles. Operation starts 4 cycles after the rising edge at the selected reset service routine location in memory.

The external interrupt input signal $\overline{\text{INT}}$ is edge-sensitive and must remain asserted for two clock cycles to set the internal INT flag. This flag is cleared as the interrupt service routine starts so that any new interrupt condition must allow INT to go high and then low again for another interrupt to be generated.

Oscillator and Clock Inputs

The XTI and XTO signals jointly supply the processor clock, either as an input from a TTL system clock or as the crystal connection to enable the internal oscillator. The maximum frequency is 33.3 MHz and the minimum is 5 MHz. An internal phase-locked-loop (PLL) doubles this to generate clocking for the two bus and functional units operations per instruction cycle. The external clock is applied to XTI, while the external crystal connection is as shown in Figure 7. A parallel-resonant fundamental-mode crystal should be used.

Table 6. ZR38001 Signal Summary

Name	Number	Type*	Description
A(19-0)	20	O/T	Address bus of parallel I/O port
D(31-0)	32	I/O/T	Data bus of parallel I/O port
CS	1	I/O/T	Chip select enable for address selection on parallel I/O port
RD	1	I/O/T	Read enable for read operation on parallel I/O port
WR	1	I/O/T	Write enable for write operation on parallel I/O port
HREQ	1	I	Request by host for access to parallel I/O port
HACK	1	O	Acknowledgment to host of access to parallel I/O port granted
SDA	1	I	Data input for serial port A
SDB	1	O	Data output for serial port B
SDC	1	O	Data output for serial port C
SDD	1	O	Data output for serial port D
SDE	1	I	Data input for serial port E or GPI4
SDF	1	I	Data input for serial port F or GPI5
WSA/FSA	1	I/O	Word select or frame synchronization input/output for A group serial ports
WSB/FSB	1	I/O	Word select or frame synchronization input/output for B group serial ports
SCKA	1	I/O	Clock input/output for A group serial ports
SCKB	1	I/O	Clock input/output for B group serial ports
MCK/SCKIN	1	I/O	Master clock output or input for B group serial ports
GPIO(1-0)	2	I/O	General purpose input or output ports
SI	1	I	Host serial interface data input
SO	1	O	Host serial interface data output
SCK	1	I	Host serial interface clock input
SS	1	I	Host serial interface slave select input
TDI	1	I	ICE test interface data input
TDO	1	O/T	ICE test interface data output
TCK	1	I	ICE test interface clock input
TMS	1	I	ICE test interface mode select
INT	1	I	External interrupt request input
RESET	1	I	Reset input to start operation in known state
MMAP	1	I	Input selects internal ROM or external memory location for reset and interrupt service routines block
XTI	1	I	External clock input or connection to external crystal
XTO	1	O	Other connection to external crystal
VCC	21	Power	+5 volt power supply
VGND	27	Power	Power supply ground

* O = output, I = input, T = tristatable. After reset, all I/O pins are inputs and tristatable pins are three-stated.

TYPICAL CONFIGURATIONS

Figure 7 shows a ZR38001 in a typical stand-alone configuration with a variety of additional external circuits. Additional connections are shown for a two-channel digital audio A/D as a source and a slave D/A, an external data/program RAM, an external data ROM and an external bootstrap byte-wide ROM.

The standard ZR38001, without being custom ordered with a user's program, has a $\overline{\text{RESET}}$ bootstrap loading routine in its internal ROM. In initial system design, users have the option of using this routine or writing their own located in an external program ROM (not shown). The internal loading routine reads from an external byte-wide ROM (shown) which has the to-be-executed program and data. After loading, the program can reside in the internal or external 32-bit RAM (shown) while data can be in the internal 20-bit RAM as well as the same external RAM. Additional fixed data may be stored in an external 20-bit data ROM as shown. For moderate size external memories of 128K-words or below, the external address decoding is very simple because of the large 20-bit address space available.

Figure 8 shows a ZR38001 in a typical system configuration with a host processor. Both the serial and parallel connections are shown with the host so that sharing of the optional external program/data RAM can be illustrated. In this configuration with the MMAP pin low the internal ROM bootstrap will check for the byte-wide external ROM. Not finding that, it will then expect to down-load program from the host through the serial connection. With MMAP high execution could start from previously loaded code in the external RAM.

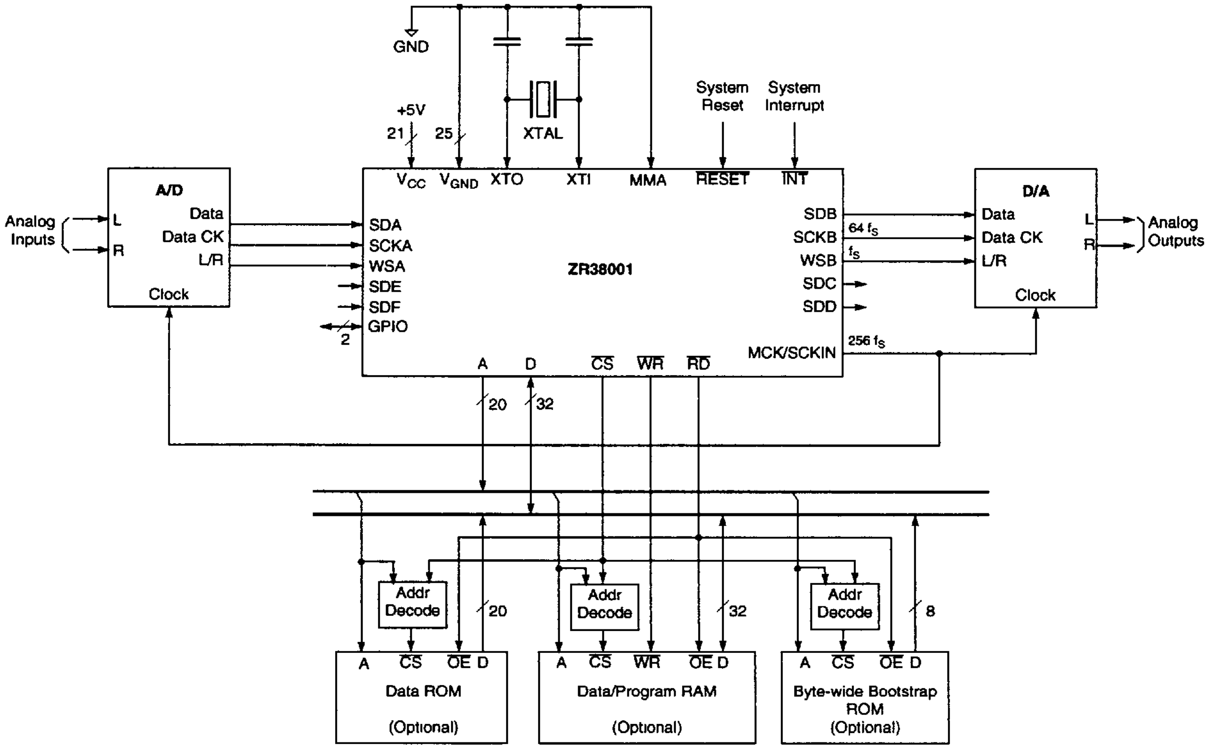


Figure 7. ZR38001 Typical Stand-Alone Configuration

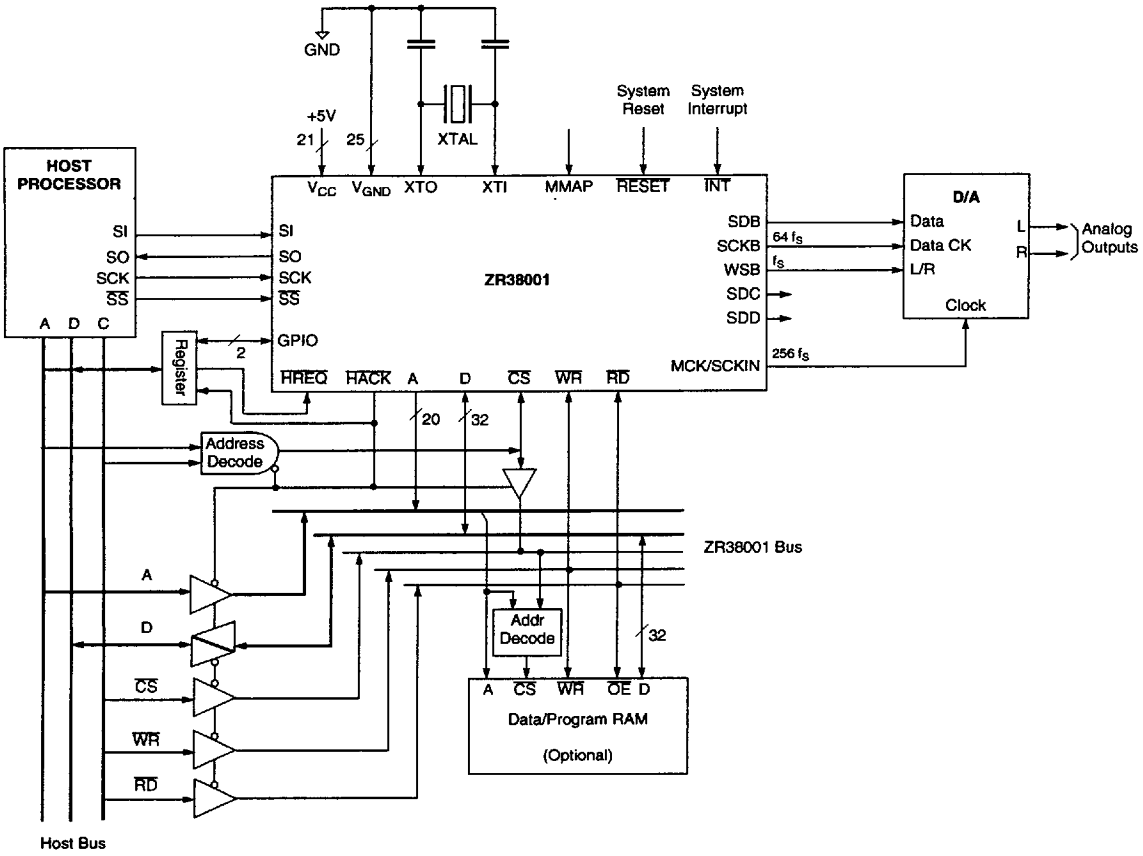


Figure 8. ZR38001 Configuration With Parallel And Serial Host I/F And Shared Memory

PRODUCT SUPPORT

DOCUMENTATION

This data sheet is a summary description of the ZR38001's functional operation and instruction set. It includes the complete electrical, timing and physical description. The complete source of information on its operation and instruction set is the "ZR38000 Family Users Manual and Programming Reference". Also available are the "ZR38000 Family Simulator User's Manual" and the "ZR38000 Family Assembler/Linker User's Manual."

SOFTWARE

Two software development tools provide all that is necessary to write, assemble, link, simulate, and debug programs for the ZR38001. They run on a 386 or 486 PC under Microsoft Windows. The Assembler/Linker translates the assembly language code, including macros, to object code which can be linked with data files and other object code to generate a complete executable program file. The Simulator accurately executes the program file while permitting full displays of registers and memory along with single-step operation and breakpoints for debugging. Both are of modern design being highly interactive and with macro and symbolic naming support throughout.

The Assembler/Linker will also generate ROM code in the form required for a custom ordered version of the ZR38001.

DEVELOPMENT BOARD

The ZR38001 Development Board is a PC/AT compatible add-in board and software for real-time operation, testing and debugging of ZR38001 programs. The hardware is an ISA bus compatible board with a ZR38001, 128Kx 32-bit words of zero wait state external memory, an interface to external analog front-ends, and a fully buffered data and control interface to the PC. The software, in conjunction with the ZR38001 Assembler/Linker, provides single-step operation, breakpoints, interactive modification and display of registers and memory, and operation on data files or real-time data.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature .... -65°C to +150°C

Supply Voltage to Ground
Potential Continuous...-0.5V to +7.0V

DC Voltage Applied to Outputs for
High Impedance Output State ....-0.5V to +5.5V

DC Input Voltage.. -0.5V to V_{CC}+0.5V

DC Output Current, into Outputs
(not to exceed 200mA total)................................... 20mA/output

DC Input Current..-10mA to +3.0mA

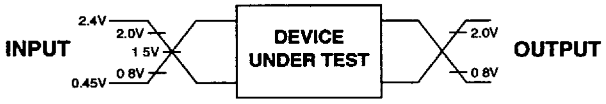
NOTE: Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGE

Temperature.. 0°C ≤ T_A ≤ +70°C
Supply Voltage...4.75V ≤ V_{CC} ≤ 5.25V

DC CHARACTERISTICS

Symbol	Parameter	Min	Typ	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5	–	0.8	V	
V_{IH}	Input High Voltage	2.0	–	V_{CC} + 0.5	V	
V_{IHS}	Input High Voltage of SCKIN	2.2	–	–	V	
V_{OL}	Output Low Voltage	–	–	0.4	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4	–	–	V	I_{OH} = -400µA
I_{CC}	Power Supply Current	–	400	500	mA	f = 33 MHz, V_{CC} = 5.25V
I_{LI}	Input Leakage Current	–	–	±10	µA	
I_{LO}	Output Leakage Current	–	–	±10	µA	
I_{WPU}	Weak Pull-Up Leakage Current	-20	-50	-100	µA	
I_{PU1}	Pull-Up PU1 Leakage Current	-100	-270	-500	µA	
I_{PU2}	Pull-Up PU2 Leakage Current	-300	-580	-1200	µA	
C_{IN}	Input Capacitance	–	–	10	pF	
C_{IO}	I/O and Output Capacitance	–	–	10	pF	



During AC testing, inputs are driven at 0.4V and 2.4V levels. Unless otherwise specified, switching times are measured from the 1.5V level of DCLK to the 0.8V or 2.0V levels at the input/output.

Figure 1. AC Testing Input, Output

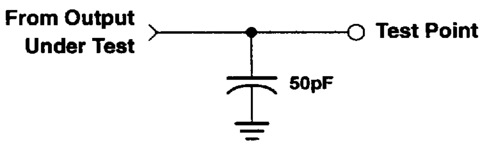


Figure 2. Normal AC Test Load

AC CHARACTERISTICS

Memory Read (External Clock Frequency = 33 MHz)

Input Requirements		Min	Max	Units	Notes
5	Data in hold from $\overline{\text{RD}}$ rising edge	0	–	ns	
7	$\overline{\text{RD}}$ low to data valid	–	6	ns	Note 1
9	Address stable, $\overline{\text{CS}}$ low to data valid	–	14	ns	Note 1

1. These specifications are for zero wait-state operation. For operation with wait states, add 1 or 7 external clock periods, as appropriate.

Output Characteristics		Min	Max	Units	Notes
1	Read cycle width	30	–	ns	Note 1
2	Address hold from $\overline{\text{RD}}$ rising edge	1	–	ns	
3	Address setup to $\overline{\text{RD}}$ falling edge	3	–	ns	
4	$\overline{\text{RD}}$ pulse width	15	–	ns	Note 1
6	$\overline{\text{RD}}$ after $\overline{\text{RD}}$ recovery time	8	–	ns	
8	$\overline{\text{WR}}$ after $\overline{\text{RD}}$ recovery time	8	–	ns	

1. These specifications are for zero wait-state operation. For operation with wait states, add 1 or 7 external clock periods, as appropriate.

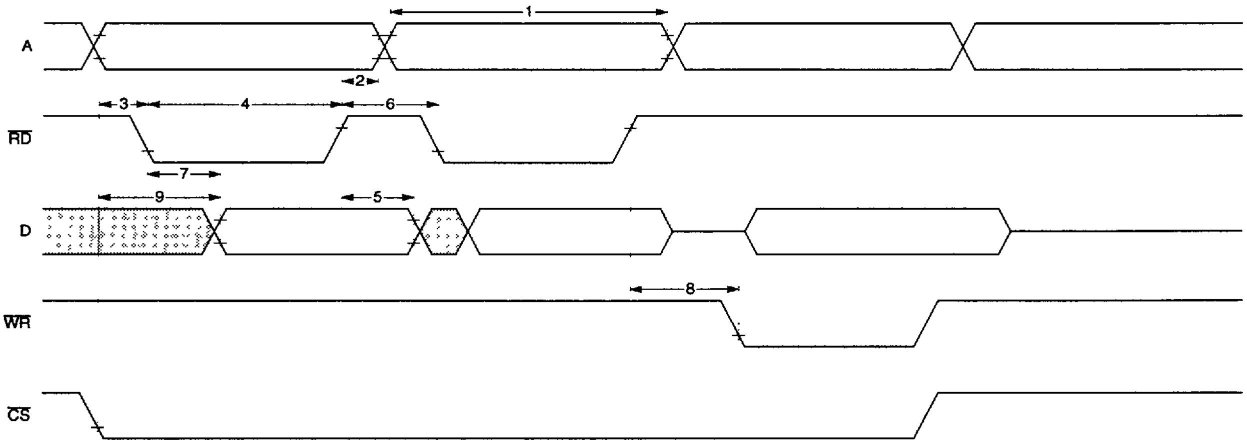


Figure 3. Memory Read

Memory Write (External Clock Frequency = 33 MHz)

Output Characteristics		Min	Max	Units	Notes
10	Write cycle width	30	–	ns	Note 2
11	Address setup to $\overline{\text{WR}}$ falling edge	3	–	ns	
12	$\overline{\text{WR}}$ pulse width	15	–	ns	Note 2
13	Address hold from $\overline{\text{WR}}$ rising edge	1	–	ns	
14	Data out setup to $\overline{\text{WR}}$ rising edge	9	–	ns	Note 2
15	Data out hold from $\overline{\text{WR}}$ rising edge	2	–	ns	
16	$\overline{\text{WR}}$ low to data out enabled	3	–	ns	
17	$\overline{\text{WR}}$ after $\overline{\text{WR}}$ recovery time	8	–	ns	
18	$\overline{\text{RD}}$ after $\overline{\text{WR}}$ recovery time	8	–	ns	
19	Data disable to $\overline{\text{RD}}$ after $\overline{\text{WR}}$	6	–	ns	

2. These specifications are for zero wait-state operation. For operation with wait states, add 1 or 7 external clock periods, as appropriate.

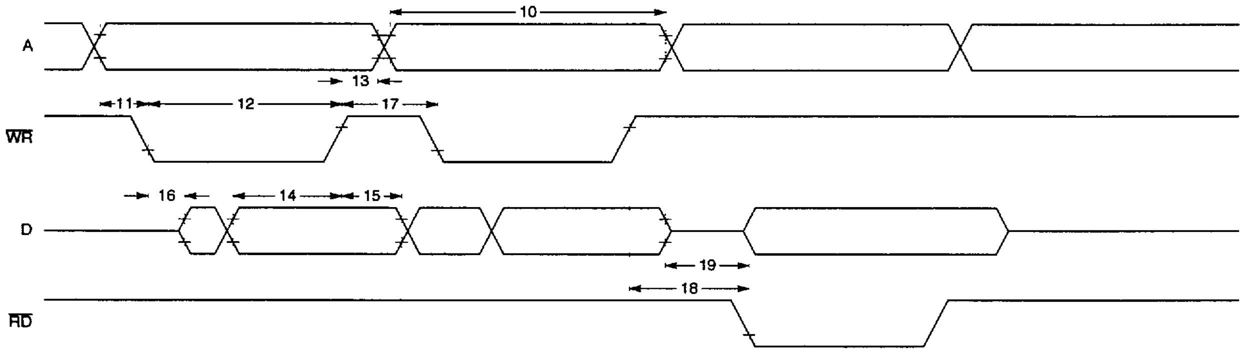


Figure 4. Memory Write

A-Group Serial Ports (Frame Sync Mode)

Input Requirements		Min	Max	Units	Notes
20	SCKA high width	t_{XTI}	–	ns	Slave Mode (MA = 0)
21	SCKA low width	t_{XTI}	–	ns	Slave Mode (MA = 0)
22	FSA setup time to SCKA falling edge	10	–	ns	Slave Mode (MA = 0)
23	FSA hold time from SCKA falling edge	10	–	ns	Slave Mode (MA = 0)
24	SDA setup time to SCKA falling edge	10	–	ns	
25	SDE setup time to SCKA falling edge	10	–	ns	
26	SDF setup time to SCKA falling edge	10	–	ns	Mode AB = 0
27	SDA hold time from SCKA falling edge	10	–	ns	
28	SDE hold time from SCKA falling edge	10	–	ns	
29	SDF hold time from SCKA falling edge	10	–	ns	Mode AB = 0

Output Characteristics		Min	Max	Units	Notes
30	SDD delay from SCKA rising edge	–	30	ns	Mode AB = 1
70	FSA output delay from SCKA rising edge	–	15	ns	Master Mode (MA = 1)
71	SCKA period	$8t_{XTI}$	–	ns	Master Mode (MA = 1)

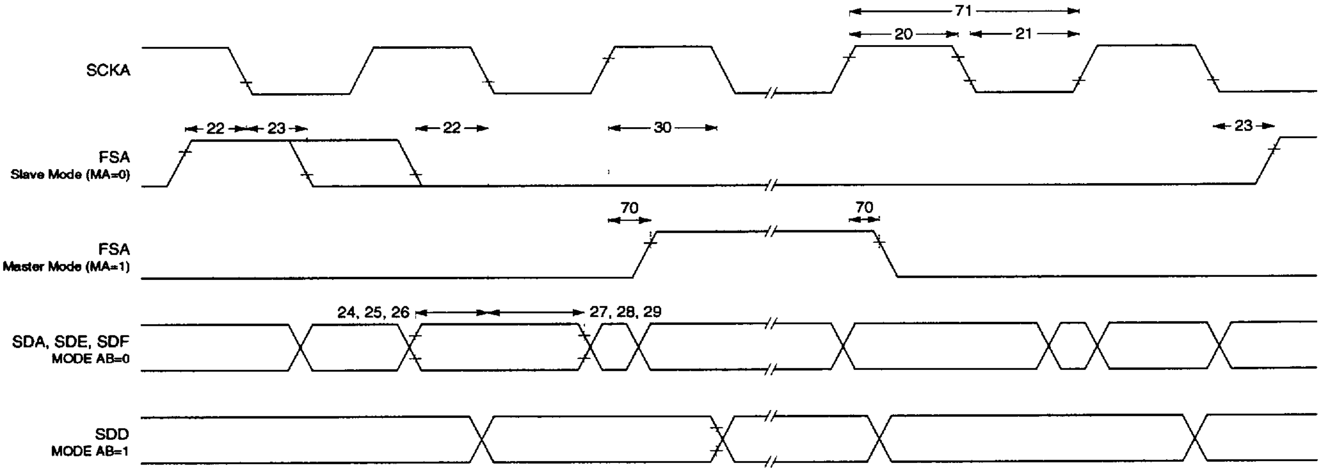


Figure 5. A-Group Serial Ports (Frame Sync Mode)

A-Group Serial Ports (Word Select Mode)

Input Requirements		Min	Max	Units	Notes
31	WSA setup time to SCKA rising edge	10	–	ns	Slave Mode (MA=0)
32	WSA hold time from SCKA rising edge	10	–	ns	Slave Mode (MA=0)
72	WSA hold time from SCKA falling edge	10	–	ns	SPMODE DA=1, Slave Mode
33	SDA setup time to SCKA rising edge	10	–	ns	
34	SDE setup time to SCKA rising edge	10	–	ns	
35	SDF setup time to SCKA rising edge	10	–	ns	Mode AB=0
36	SDA hold time from SCKA rising edge	10	–	ns	
37	SDE hold time from SCKA rising edge	10	–	ns	
38	SDF hold time from SCKA rising edge	10	–	ns	Mode AB=0

Output Characteristics		Min	Max	Units	Notes
39	SDD delay from SCKA falling edge	–	30	ns	Mode AB=1
73	WSA output delay from SCKA falling edge	–	15	ns	Master Mode (MA=1)

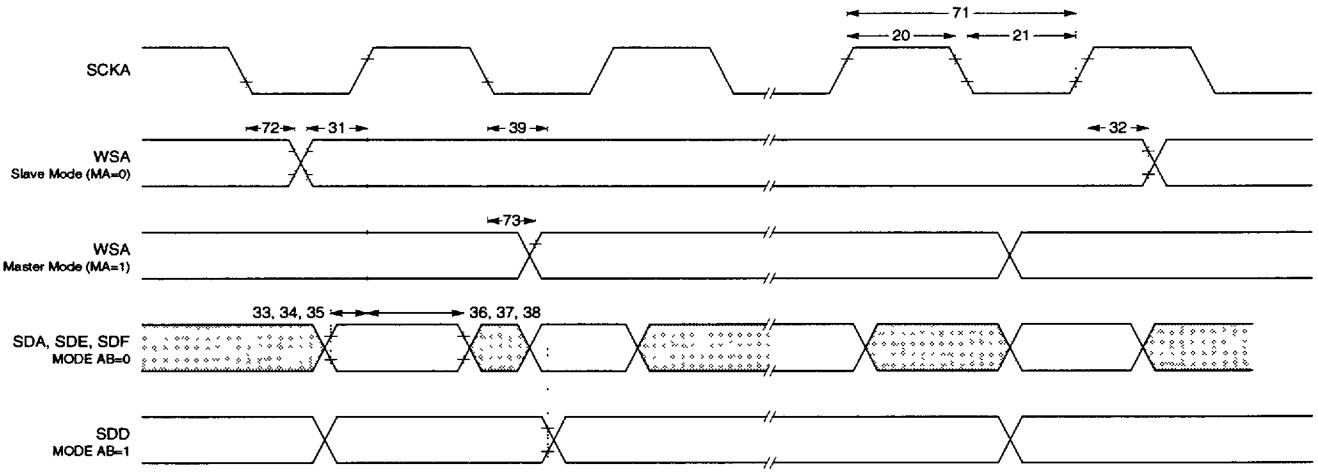


Figure 6. A-Group Serial Ports (Word Select Mode)

B-Group Serial Ports (Frame Sync Mode)

Input Requirements		Min	Max	Units	Notes
40	MCK/SCKIN period (t_{SCKIN})	$2t_{XTI}$	–	ns	When CB=1
41	MCK/SCKIN low width	35	–	ns	When CB=0
42	MCK/SCKIN high width	15	–	ns	When CB=0
75	SCKB high width	t_{XTI}	–	ns	Slave Mode (MB=0)
76	SCKB low width	t_{XTI}	–	ns	Slave Mode (MB=0)
48	SDF setup time to SCKB falling edge	10	–	ns	Mode AB=1
49	SDF hold time from SCKB falling edge	10	–	ns	Mode AB=1
78	FSB setup time to SCKB falling edge	15	–	ns	Slave Mode (MB=0)
79	FSB hold time from SCKB falling edge	15	–	ns	MB=0

Output Characteristics		Min	Max	Units	Notes
43	SCKB period	$4t_{SCKIN}$	$4096t_{SCKIN}$	ns	Master Mode (MB=1)
44	FSB delay from SCKB rising edge	–	15	ns	Master Mode (MB=1)
45	SDB delay from SCKB rising edge	–	15	ns	
46	SDC delay from SCKB rising edge	–	15	ns	
47	SDD delay from SCKB rising edge	–	15	ns	Mode AB=0

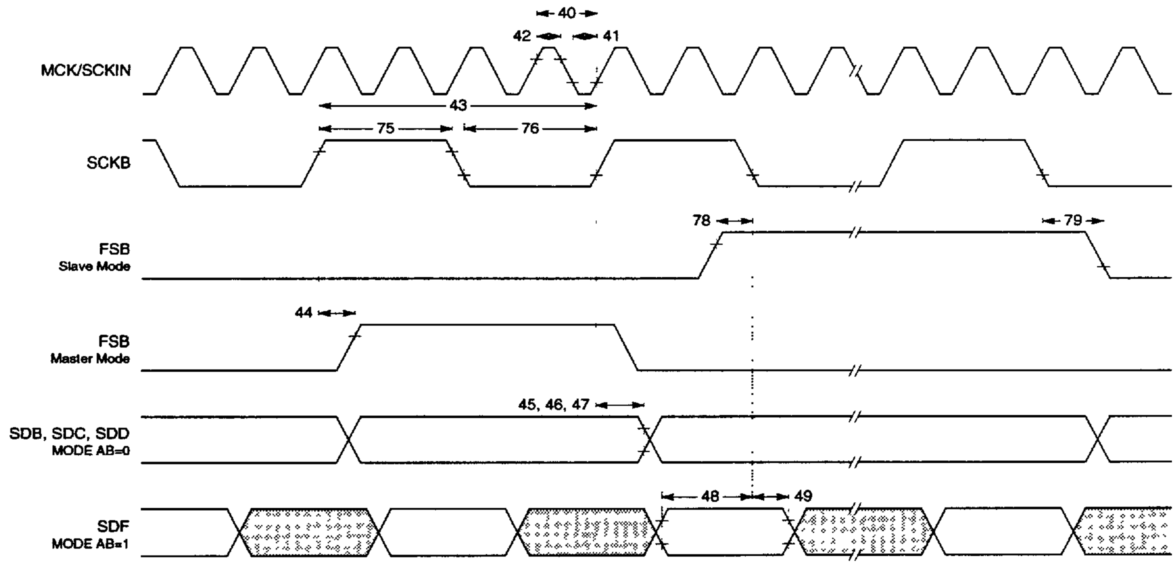


Figure 7. B-Group Serial Ports (Frame Sync Mode)

B-Group Serial Ports (Word Select Mode)

Input Requirements		Min	Max	Units	Notes
54	SDF setup time to SCKB rising edge	10	–	ns	Mode AB=1
55	SDF hold time from SCKB rising edge	10	–	ns	Mode AB=1
81	WSB setup to SCKB rising edge	15	–	ns	Slave Mode
82	WSB hold from SCKB rising edge	15	–	ns	Slave Mode
83	WSB hold time from SCKB falling edge	15	–	ns	DB=1, Slave Mode

Output Characteristics		Min	Max	Units	Notes
50	WSB delay from SCKB falling edge	–	15	ns	Master Mode (MB=1)
51	SDB delay from SCKB falling edge	–	15	ns	
52	SDC delay from SCKB falling edge	–	15	ns	
53	SDD delay from SCKB falling edge	–	15	ns	Mode AB=0

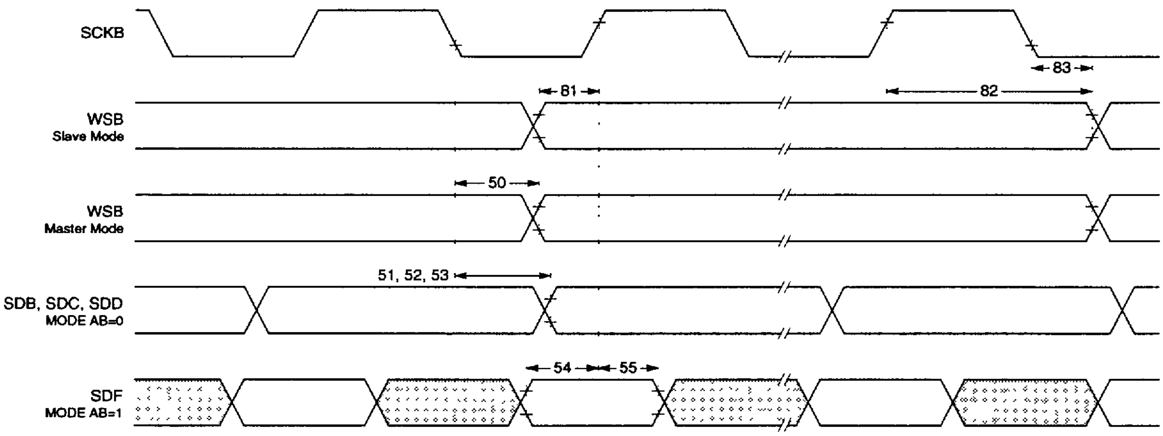


Figure 8. B-Group Serial Ports (Word Select Mode)

Parallel Host Interface Timing

	Characteristics	Min	Max	Units	Notes
101	Assertion of $\overline{CS}$ ($\overline{RD}$, $\overline{WR}$) after $\overline{HACK}$	0	–	ns	
102	$\overline{HREQ}$ deassertion after $\overline{CS}$ ($\overline{RD}$, $\overline{WR}$)	0	–	ns	
103	Data hold time from $\overline{CS}$ and $\overline{WR}$ high	10	–	ns	
104	Data setup time to $\overline{CS}$ and $\overline{WR}$ high	10	–	ns	Write cycle
105	Data out delay from $\overline{CS}$ and $\overline{RD}$ low	–	20	ns	Read cycle
106	Assertion of $\overline{HACK}$ after $\overline{HREQ}$ assertion	$4t_{XTI}$	–	ns	Note 1
107	Deassertion of $\overline{HACK}$ after $\overline{HREQ}$ deassertion	–	$2t_{XTI}$	ns	

1. If a repeat instruction is in progress, $\overline{HACK}$ assertion is delayed until the instruction completes.

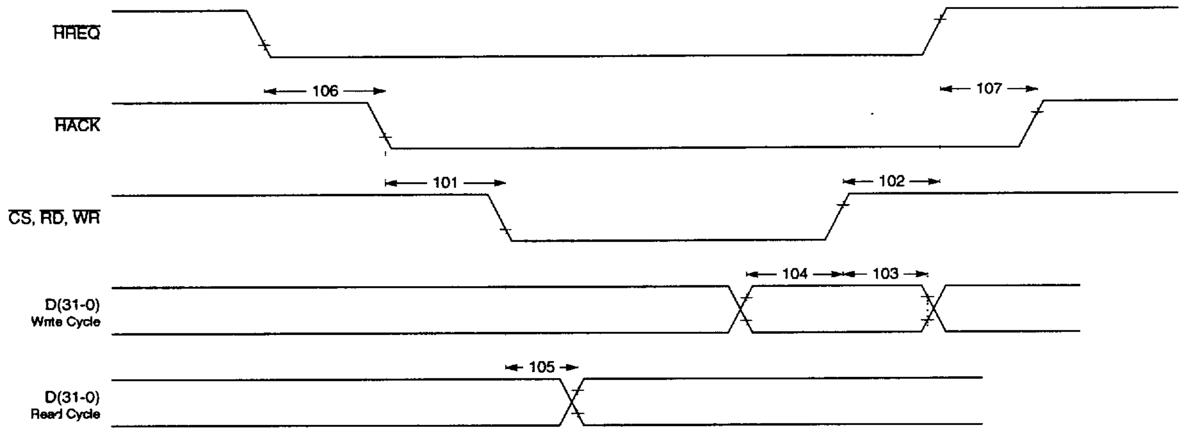


Figure 9. Parallel Host Interface Timing

Serial Host Interface Timing

Characteristics		Min	Max	Units	Notes
111	SCK clock period	$4t_{XTI}$	–	ns	Note 1
112	SCK clock high width	$2t_{XTI}$	–	ns	
113	SCK clock low width	$2t_{XTI}$	–	ns	
114	$\overline{SS}$ setup time to first SCK edge	10	–	ns	
115	$\overline{SS}$ hold time from last edge of SCK	60	–	ns	
116	SI setup time to SCK active edge	10	–	ns	
117	SI hold time from SCK active edge	30	–	ns	
118	$\overline{SS}$ negation to data Hi-Z	–	10	ns	
119	SO delay from SCK active edge	–	15	ns	
110	SO hold time from SCK active edge	10	–	ns	

1. SCK polarity is controlled by field SCKP of register SPIMODE. The polarity shown in Figure 10 corresponds to SCKP=0.

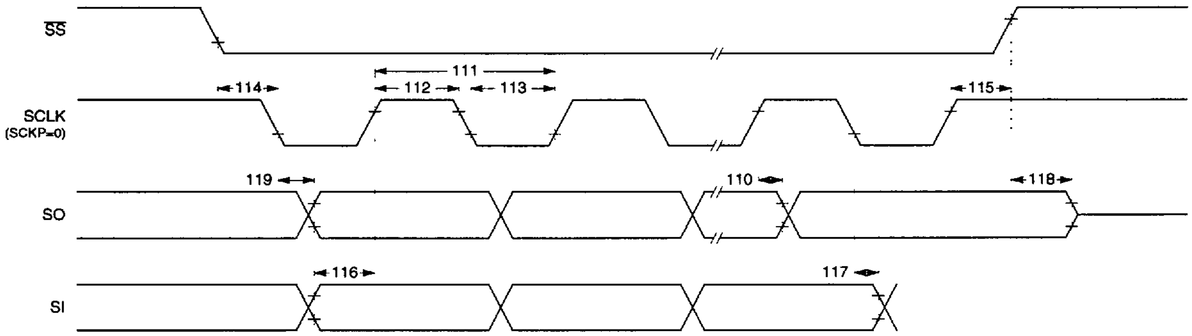


Figure 10. Serial Host Interface Timing

External Clock

		Min	Max	Units	Notes
56	XTI period (t_{XTI})	30	160	ns	
57	XTI high width	12	–	ns	
58	XTI low width	12	–	ns	

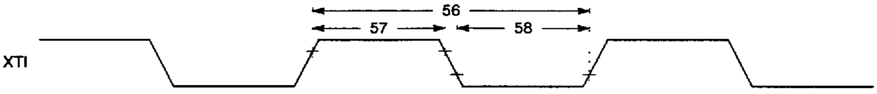


Figure 11. External Clock

Reset

		Min	Max	Units	Notes
59	$\overline{\text{RESET}}$ width, warm reset	$5t_{XTI}$	$127\ t_{XTI}$	ns	
60	$\overline{\text{RESET}}$ width, cold reset	$128t_{XTI}$	–	ns	Note 1
61	Memory bus enable after $\overline{\text{RESET}}$ rising edge, warm reset	t_{XTI} +20	–	ns	
62	Memory bus disable after $\overline{\text{RESET}}$ falling edge	–	$3t_{XTI}$	ns	

1. Applies to the power-up sequence. The rising edge of $\overline{\text{RESET}}$ must occur after the crystal oscillator or external clock frequency and amplitude have stabilized. After the rising edge of a cold $\overline{\text{RESET}}$, 4096 clock cycles are required for initialization of the internal phase locked loop, during which the processor is inactive. Any subsequent reset pulse of 128 clocks or longer is interpreted as a cold reset and will start a new initialization of the phase locked loop.

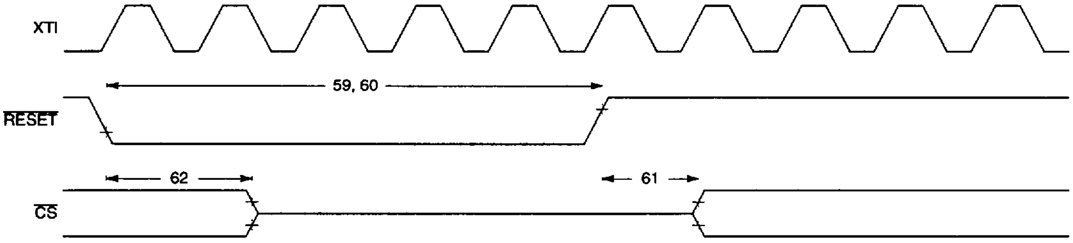


Figure 12. Warm Reset

External Interrupt

		Min	Max	Units	Notes
63	$\overline{\text{INT}}$ setup time	6		ns	Note 1
64	$\overline{\text{INT}}$ hold time	5		ns	Note 1
65	$\overline{\text{INT}}$ width	$2t_{XTI}$		ns	

1. For reference only. Synchronous operation is not required.

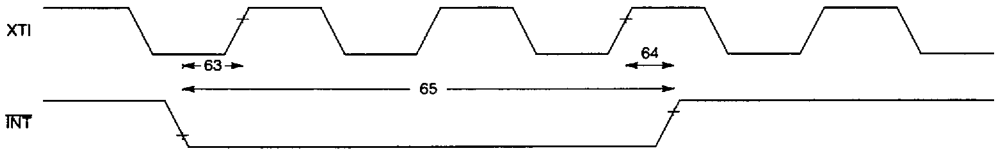


Figure 13. External Interrupt

ICE Interface Timing

Characteristics		Min	Max	Units	Notes
121	TCK clock period	$4t_{XTI}$	–	ns	
122	TCK clock high width	$2t_{XTI}$	–	ns	
123	TCK clock low width	$2t_{XTI}$	–	ns	
124	TDO negation to data Hi-Z	–	15	ns	
125	TDO delay from falling edge of TCK	–	15	ns	
126	TDI hold time from rising edge of TCK	10	–	ns	
127	TDI setup time to rising edge of TCK	5	–	ns	
128	TMS hold time from rising edge of TCK	10	–	ns	
129	TMS setup time to rising edge of TCK	10	–	ns	

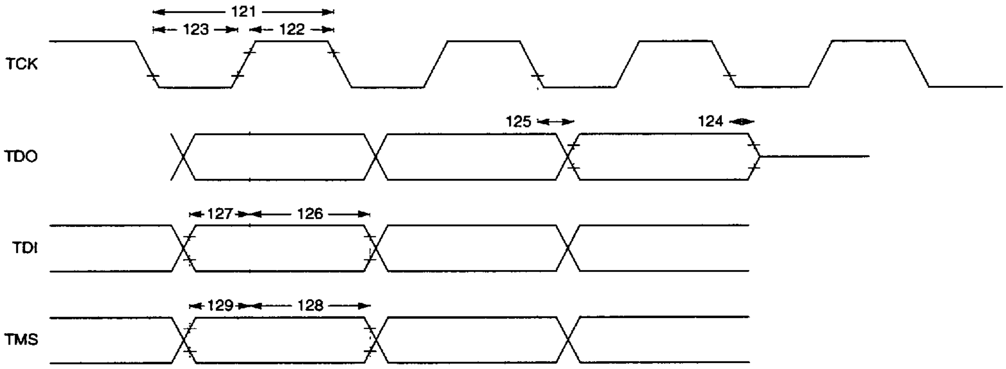


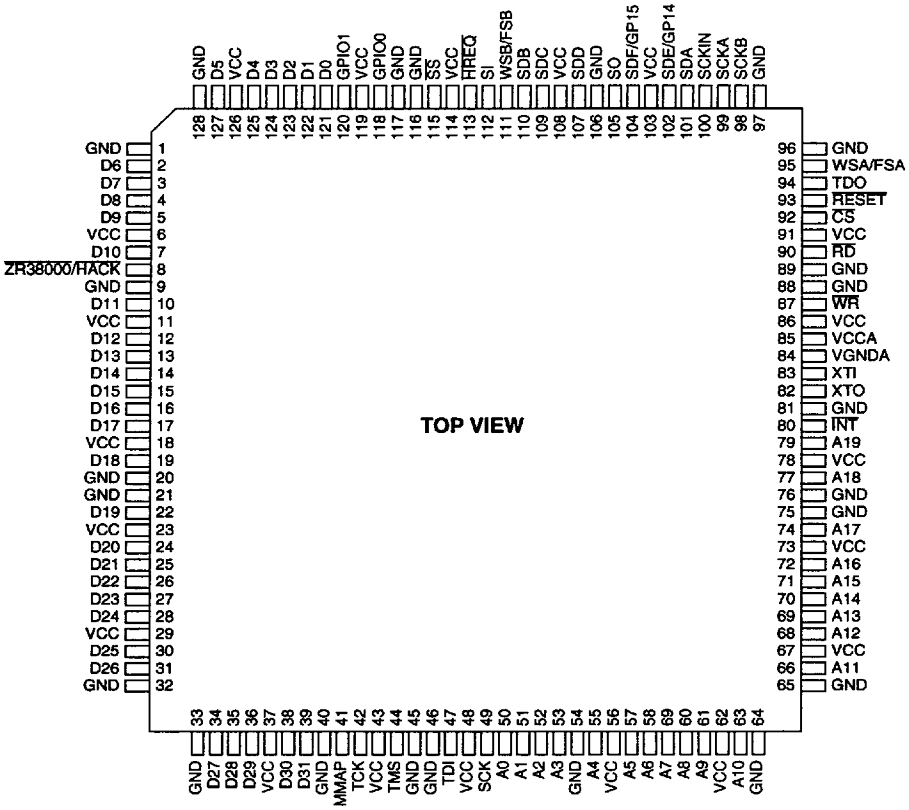
Figure 14. ICE Interface Timing

PINOUT INFORMATION

Table 1. 128-Pin QFP Pin Assignment [1, 2, 3]

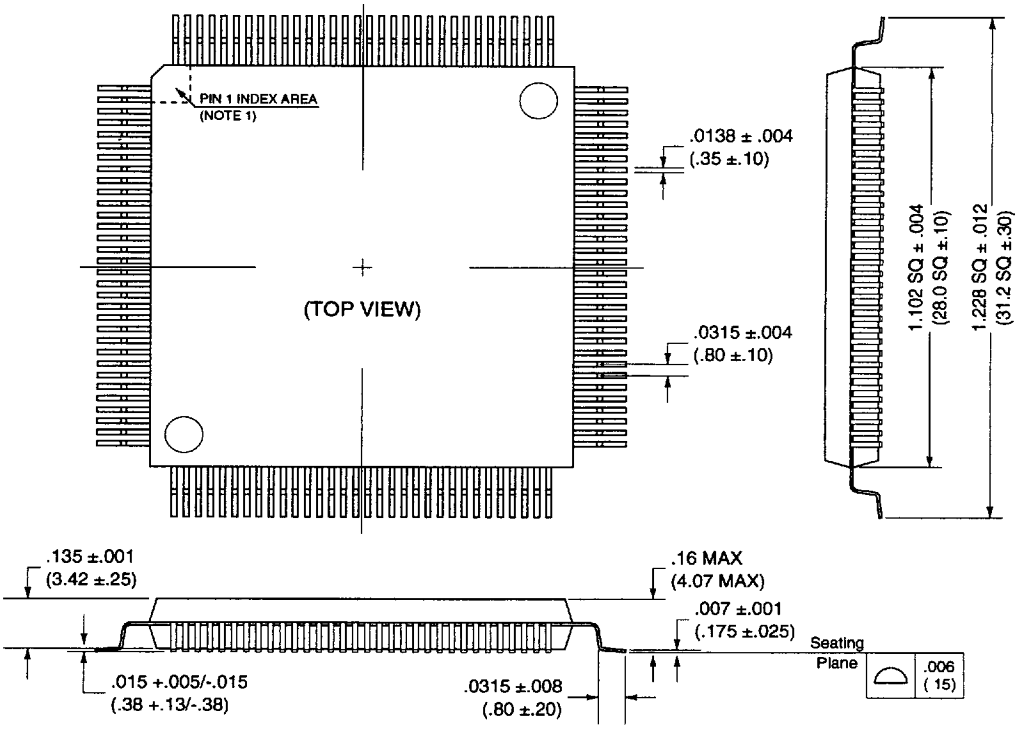
Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type
1	GND	P	27	D23	I/O/T	53	A3	O/T	79	A19	O/T	105	SO	P
2	D6	I/O/T	28	D24	I/O/T	54	GND	P	80	$\overline{\text{INT}}$	I	106	GND	P
3	D7	I/O/T	29	VCC	P	55	A4	O/T	81	GND	P	107	SDD	O
4	D8	I/O/T	30	D25	I/O/T	56	VCC	P	82	XTO	O	108	VCC	P
5	D9	I/O/T	31	D26	I/O/T	57	A5	O/T	83	XTI	I	109	SDC	O
6	VCC	P	32	GND	P	58	A6	O/T	84	GNDA	P	110	SDB	O
7	D10	I/O/T	33	GND	P	59	A7	O/T	85	VCCA	P	111	WSB/FSB	O
8	$\overline{\text{ZR30800/HACK}}$	I/O [4]	34	D27	I/O/T	60	A8	O/T	86	VCC	P	112	SI	I
9	GND	P	35	D28	I/O/T	61	A9	O/T	87	$\overline{\text{WR}}$	O	113	$\overline{\text{HREQ}}$	I
10	D11	I/O/T	36	D29	I/O/T	62	VCC	P	88	GND	P	114	VCC	P
11	VCC	P	37	VCC	P	63	A10	O/T	89	GND	P	115	$\overline{\text{SS}}$	I
12	D12	I/O/T	38	D30	I/O/T	64	GND	P	90	$\overline{\text{RD}}$	O	116	GND	P
13	D13	I/O/T	39	D31	I/O/T	65	GND	P	91	VCC	P	117	GND	P
14	D14	I/O/T	40	GND	P	66	A11	O/T	92	$\overline{\text{CS}}$	O	118	GPIO0	I/O
15	D15	I/O/T	41	MMAP	I	67	VCC	P	93	$\overline{\text{RESET}}$	I	119	VCC	P
16	D16	I/O/T	42	TCK	I	68	A12	O/T	94	TDO	O/T	120	GPIO1	I/O
17	D17	I/O/T	43	VCC	P	69	A13	O/T	95	WSA/FSA	I	121	D0	I/O/T
18	VCC	P	44	TMS	I	70	A14	O/T	96	GND	P	122	D1	I/O/T
19	D18	I/O/T	45	GND	P	71	A15	O/T	97	GND	P	123	D2	I/O/T
20	GND	P	46	GND	P	72	A16	O/T	98	SCKB	O	124	D3	I/O/T
21	GND	P	47	TDI	I	73	VCC	P	99	SCKA	I	125	D4	I/O/T
22	D19	I/O/T	48	VCC	P	74	A17	O/T	100	SCKIN	I	126	VCC	P
23	VCC	P	49	SCK	I	75	GND	P	101	SDA	I	127	D5	I/O/T
24	D20	I/O/T	50	A0	O/T	76	GND	P	102	SDE/GP14	I	128	GND	P
25	D21	I/O/T	51	A1	O/T	77	A18	O/T	103	VCC	P			
26	D22	I/O/T	52	A2	O/T	78	VCC	P	104	SDF/GP15	I			

[1] Unused inputs (I) must be connected to VCC, if active low, or to GND, if active high.
[2] Unused outputs (O), three-state (T) and NC pins, should be left unconnected.
[3] All data pins are internally pulled up to VCC using weak pull-up transistors (WPU).
[4] Strong pull-up to VCC (PU2).



N.C. = No connect. Leave this pin unconnected.

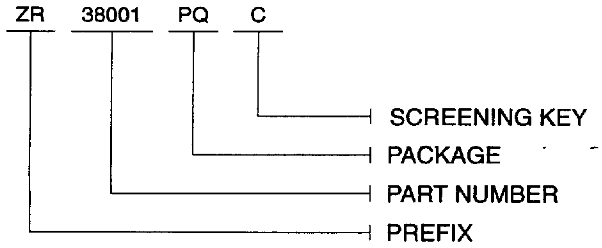
PACKAGE INFORMATION



NOTES: 1. Pin 1 index may be comer chamfer, dot or both
2. Principal dimensions in inches, dimensions in brackets in (millimeters).
3. Top and bottom mold marks shown for indication only. Location may vary except for Pin 1 index area.

Figure 15. ZR38001 Plastic Quad Flat Pack Dimensions

ORDERING INFORMATION



PACKAGE

PQ - Plastic Quad Flat Pack (EIAJ)

SCREENING KEY

C - 0°C to +70°C (V$_{CC}$ = 4.75V to 5.25V)

SALES OFFICES

- ***U.S. Headquarters***
  Zoran Corporation
  1705 Wyatt Drive
  Santa Clara, CA 95054 USA
  Telephone: 408-986-1314
  FAX: 408-986-1240

- ***Israel Design Center***
  Zoran Microelectronics, Ltd.
  Advanced Technology Center
  P.O. Box 2495
  Haifa, 31024 Israel
  Telephone: 972-4-551-551
  FAX: 972-4-551-550

Trademarks:
All brand, product, and company names are trademarks or registered trademarks of their respective companies.

The material in this data sheet is for information only. Zoran Corporation assumes no responsibility for errors or omissions and reserves the right to change, without notice, product specifications, operating characteristics, packaging, etc. Zoran Corporation assumes no liability for damage resulting from the use of information contained in this document.