SIX-CHANNEL DOLBY DIGITAL SURROUND PROCESSOR

FEATURES

- Multi-Function
  - Decode of six-channel Dolby AC-3 Digital Surround Sound
  - Two-channel AC-3 to four-channel Dolby Pro Logic
  - Two-channel PCM to four-channel Dolby Pro Logic
  - One- or two-channel PCM to two-channel mono or stereo
  - Pink noise generator function for testing of all channels
  - Programmable center and surround channel delays
  - Added power of the programmable ZR38001 core processor for custom product features
- Low-Cost Solution
  - Programmable internal I/O clock generators
  - Internal oscillator with x2 PLL for use with 33 MHz crystal
  - No host microcontroller required
  - Simple bit-serial SPI compatible host interface
  - Wait-state generation for low-cost external memory
  - 128-pin Plastic Quad Flat Pack (PQFP) packaging
- Flexible System Component
  - Internal ROM contains all decoder algorithms
  - Control by commands from host microprocessor or the user designed core processor program
  - User designed program in external boot-strap ROM, internal masked ROM or down-loaded from host
  - User designed program adds product distinctions to industry standard decoder functions
- Flexible Audio Input/Output
  - Master or slave I/O with programmable internal clocks
  - 16-, 18- or 20-bit word sizes
  - Formats: I^2S, EIAJ, non-delayed data or frameless
  - Protocols: AC-3/IEC 958 (S/PDIF) or unformatted
  - Sample rates: 44.1 kHz, 48 kHz or 32 kHz
- 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

- The complete solution for Dolby multi-channel decoding for:
  - CD and Digital Audio Tape consumer electronics
  - Television set-top boxes
  - Multimedia computing
  - Virtual Reality
  - Professional digital audio

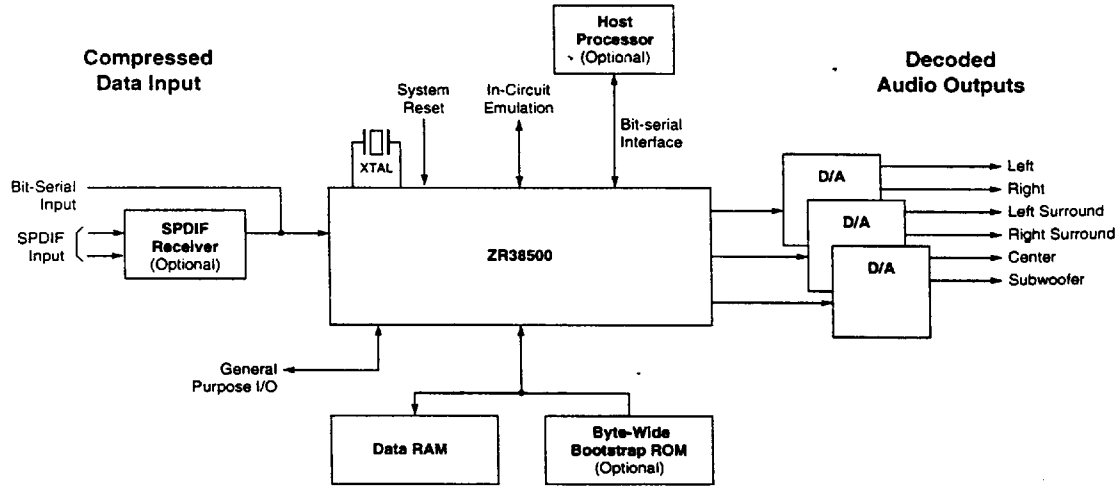


Figure 1. ZR38500 Configuration Diagram

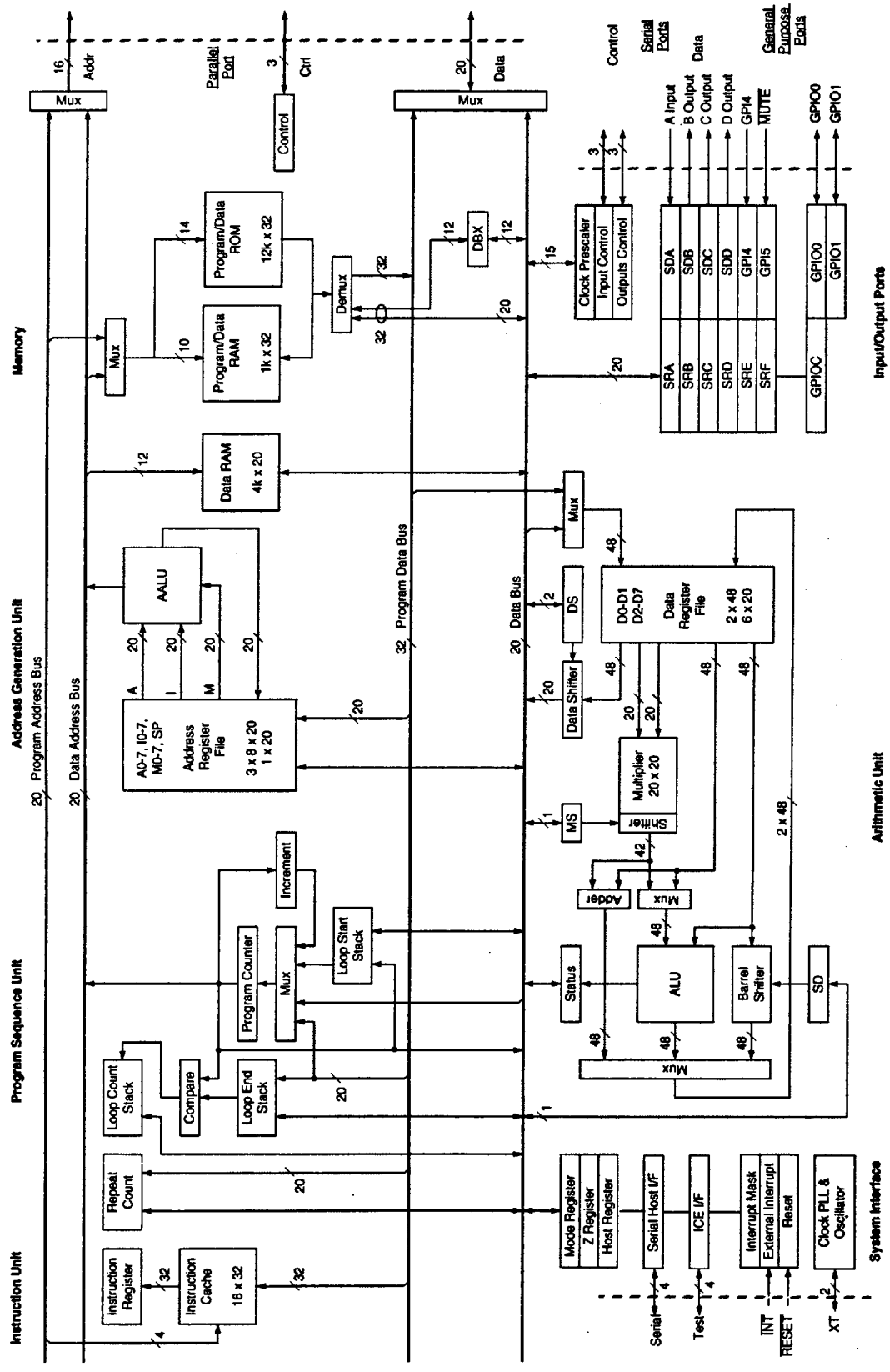


Figure 2. ZR38500 Detailed Block Diagram

GENERAL DESCRIPTION

The Zoran ZR38001 is the first programmable digital signal processor capable of real-time single-chip decoding of the Dolby Laboratories AC-3 six-channel digital surround sound algorithm. The ZR38500 is a derivative design with AC-3 and a selection of Dolby algorithms pre-programmed in internal ROM. Useful in a number of different configurations, it provides even lower cost system solutions than a ZR38001 yet retains the programmability.

From a single compressed digital signal stream (in a variety of formats), the ZR38500 generates all of the signals for virtually all serial audio D/A converters to produce the decoded analog outputs. The available decoding functions, illustrated in Figure 3, are:

- Compressed six-channel Dolby AC-3 to the six decoded channels. Loudspeaker configuration, center and surround delay and dynamic range are all selectable
- Compressed two-channel AC-3 to the four decoded channels of Dolby Pro Logic. Surround delay and loudspeaker configuration are selectable
- Two-channel PCM to the four generated channels of Dolby Pro Logic. Surround delay and loudspeaker configuration are selectable
- One- or two-channel PCM with stereo or mono loudspeaker configuration and mixing selectable.
- An additional function is pink noise generation for system testing on any selected channel.

These functions can be used on their own in a very cost-effective way, but an additional power of the ZR38500 is that they may be embedded in user written code with other functions that execute on the core processor. An additional internal mask-programmable ROM of 1.5 Kwords and/or 1 Kwords of RAM are available for these coded functions.

The ZR38500 is as flexible in its hardware configuration as in its functional operation. Figure 1 shows the possibilities. The basic configuration is the ZR38500 with its clock crystal, a RAM for I/O buffering and delay memory, the desired input connection and the appropriate number of output D/A converters.

This basic configuration is all that is required when using a custom mask-programmed version. The user written program in ROM executes the desired functions with the selected parameters after being reset by the system. Control and interaction with the system can take place through the four single-bit user-defined general-purpose I/O ports.

The basic configuration's operation can be achieved without resort to custom mask-programming by using the optional external byte-wide bootstrap ROM shown. The user code in the ROM is loaded after reset and executed from the internal program/data RAM.

If there is a host microprocessor in the system, only the basic configuration need be used. A connection is made with the host through an inexpensive bit-serial peripheral interface (SPI compatible). User code may be then be down-loaded from the host or the decoder functions and their parameters may be selected through a simple command structure.

The ZR38500 is also useful in many different digital audio applications because of the flexibility of its input/output ports. Three sample rates of 48, 44.1 and 32 kHz are supported with the processor being selectable as either slave or master for input and outputs separately. Programmable dividers are used when master I/O clocking is derived from the ZR38500's crystal clock. This master clock is output for system use or the input divider can be used with an external clock. Word size can be 16, 18 or 20 bits with single or multiple word frames. Formats can be I^2S with word select or frame synchronization, EIAJ with LSB justification or ones with non-delayed data.

Special protocols supported are the AC-3 data stream in IEC 958 (S/PDIF) format and the continuous or frameless types used in telecommunications.

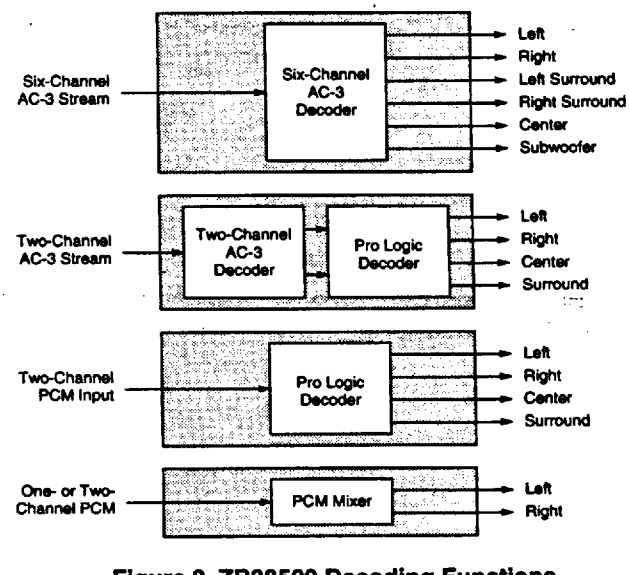
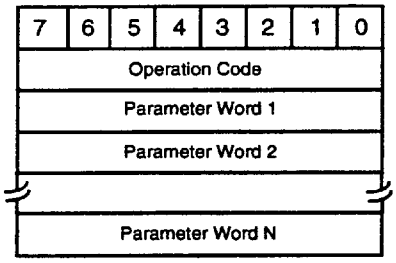


Figure 3. ZR38500 Decoding Functions

DECODER FUNCTIONAL DESCRIPTION

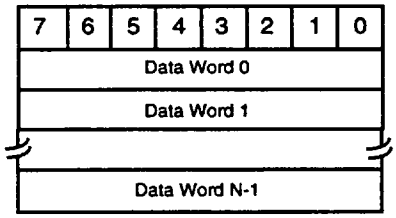
The ZR38500 decoding functions are selected by the commands shown in Tables 1 and 2, with their parameter word sets. These are the commands and parameters sent by a microprocessor over the serial host interface when a host is used. The program subroutines and parameter tables for the AC-3, Pro Logic and Pink Noise functions are similar. They are called by a user's program executing on the core processor.

Transfers with a host and the ZR38500 are full duplex with the host being the master. For every command sent from the host a word is received back from the ZR38500. The commands are sent only from the host and are of the following form:



There are two classes of commands: those that Write to the decoder and those that Read back from the decoder. The Write commands may have parameter words in addition to the basic operation code. The Write commands are of three types as shown in Table 1: those that choose functions, those that govern operation (e.g., STOP or PLAY) and those that set-up operation (e.g., Configure).

The word received back from the ZR38500 when the host initiates a transfer will be zero or the response to a command. Responses are of the form:



The responses are of two types, a Reply to READ commands following specific Write commands or the Progress responses to each Write command.The Progress response is always the number of still expected parameters (EXPECT) followed by the response and interpreter status (ISTATUS).

Words sent from Host

1	2	3				time
SETIO	Para 1	READ	NOP	NOP	NOP	NOP

Words sent from ZR38500

		1	2	3		
0	0	EXPECT=1	SETIOR	ISTATUS	0	0

All responses are delayed by one word as shown in this example of the SETIO command which has both a parameter word (Para 1) and a Reply data word (SETIOR). Note that the zeros returned are a response to the NOPs sent by the host.

The following descriptions explain the ZR38500's functions as well as the specifics of the commands used and their responses.

AC-3 Decoder Function AC3

	7	6	5	4	3	2	1	0
Command	1	0	0	0	0	1	0	1
Parameter 1	PRLG		SIF	1	COMP		DMM	
Parameter 2	0	AB	PBCFG		SW	OCFG		
Parameter 3	CDLY			SRDLY				
Parameter 4	HDR							
Parameter 5	LDR							
Parameter 6	0	0	0	0	0	RPC		
Parameter 7	PSFH							
Parameter 8	PSFL							

The AC-3 Decoder function includes normal six-channel AC-3 and the two-channel AC-3 with Pro Logic output. Selection is made in the command for speaker configuration, dynamic range compression and downmixing, delays and error concealment strategy.

PRLG	**Pro Logic** output: 0 = Off, 1 = On, 2 = Selected automatically based on input stream information.
SIF	**Serial Input Format**: 0 = Non-formatted, 1 = AC-3 S/PDIF protocol.
COMP	**Compression and Dialog Normalization.** 0 = Off, 1 = Dialog normalization mode, 2 = Line-out mode (dialog normalization plus high level compression), 3 = RF modulation mode (peak level compression).
DMM	**Dual Mono Mode** output selection when the two input channels are unrelated: 0 = Stereo, 1 = Mono channel 0 to both, 2 = Mono channel 1 to both, 3 = Mono channels 0 and 1 summed and scaled to both.
AB	**Auto-Balance** for Pro Logic output: 0 = On, 1 = Off.
PBCFG	**Pro Logic Bass Configuration** redirection: 0 = No redirection, 1 = Redirect center bass to left and right, 2 = Redirect left, right and center bass to subwoofer, 3 = Redirect center bass to subwoofer.
SW	**Subwoofer** output channel: 0 = Off, 1 = On.
OCFG	**Output Speaker Configuration** of Front/Surround number of speakers: 0 = 2/0 Surround Compatible, 1 = 1/0, 2 = 2/0 Normal, 3 = 3/0, 4 = 2/1, 5 = 3/1, 6 = 2/2, 7 = 3/2. In 2/0 configurations, a mono input is directed to both output channels.
CDLY	**Center Delay** in one millisecond steps from zero to five.
SRDLY	**Surround Delay** in one millisecond steps from zero to 31 milliseconds.

HDR	**High Dynamic Range** scale factor controlling the depth of high-level compression. A two's complement fraction between 0.00 and 0.FE where 0.00 is no high-level compression and 0.FE is full compression.
LDR	**Low Dynamic Range** scale factor controlling the depth of low-level compression. A two's complement fraction between 0.00 and 0.FE where 0.00 is no low-level compression and 0.FE is full compression.
RPC	**Repeat Count** before muting. Maximum number of consecutive block repeats before muting output.
PSFH	**PCM Scale Factor High.** Output scale factor,16-bit two's complement fraction between 0.0000 and 0. FFFE. The high byte.
PSFL	**PCM Scale Factor Low.** Output scale factor low byte.

Pro Logic Decoder Function — PROL

	7	6	5	4	3	2	1	0
Command	1	0	0	0	0	1	1	0
Parameter 1	0	AB	PBCFG		SW	OCFG		
Parameter 2	CDLY				SRDLY			
Parameter 3	PSFH							
Parameter 4	PSFL							

The Pro Logic Decoder function decodes two-channel PCM into four-channel Pro Logic output. Selection is made in the command for speaker configuration, scale factor and delay. The parameter descriptions are the same as for AC-3 except for OCFG where there are no Surround Compatible, Mono or Stereo selections.

PCM Mixer Function — PCM

	7	6	5	4	3	2	1	0
Command	1	0	0	0	0	1	0	0
Parameter 1	0	0	0	0	0	0	DMM	
Parameter 2	PSFH							
Parameter 3	PSFL							

The PCM Mixer function transfers two-channel PCM input to two-channel PCM output. Selection is made for output modes, including mixing, and scale factor. The parameter descriptions are the same as for AC-3.

Pink Noise Generator Function — PNG

	7	6	5	4	3	2	1	0
Command	1	0	0	0	0	0	1	1
Parameter 1	0	0	L	C	R	LS	RS	SW
Parameter 2	PSFH							
Parameter 3	PSFL							

The Pink Noise Generator function produces pseudo-random noise sequence outputs. Selection is made for output channel and scale factor. The parameter descriptions are the same as for AC-3 with the addition of the output channels.

L	**Left** channel output.
C	**Center** channel output.
R	**Right** channel output.
LS	**Left Surround** channel output.
RS	**Right Surround** channel output.
SW	**Subwoofer** channel output.

User Function — USER

	7	6	5	4	3	2	1	0
Command	1	0	0	0	1	0	0	0
Parameter 1	Parameter Word 1							
Parameter 8	Parameter Word 8							

The USER command passes eight bytes of parameters to a user programmed function as described in the ZR38500 Application Programming Interface (API).

Resume Operation — PLAY

	7	6	5	4	3	2	1	0
Command	1	0	0	0	1	0	1	0

The PLAY command resumes operation of the selected function and unmutes the output after a STOP or STOPF command.

Mute Operation — MUTE

	7	6	5	4	3	2	1	0
Command	1	0	0	0	1	0	1	1

The MUTE command mutes the output without stopping the operation of the selected function.

Unmute Operation — UNMUTE

	7	6	5	4	3	2	1	0
Command	1	0	0	0	1	0	0	1

The UNMUTE command restores the muted output while continuing the operation of the selected function.

Stop Operation — STOP

	7	6	5	4	3	2	1	0
Command	1	0	0	0	1	1	0	0

The STOP command stops operation of the selected function and mutes the output. Data in the input buffer is preserved but new data is ignored.

Stop Operation and Flush — STOPF

	7	6	5	4	3	2	1	0
Command	1	0	0	0	1	1	0	1

The STOPF command stops operation of the selected function and mutes the output. Data in the input buffer is flushed out and new data is ignored.

Return Status — STAT

	7	6	5	4	3	2	1	0
Command	1	0	0	0	1	1	1	0

The STAT command allows the N words of STATR status information to be returned to the host by sending N READ commands.

No Operation — NOP

	7	6	5	4	3	2	1	0
Command	1	0	0	0	0	0	0	0

A NOP is not a command and does not affect operation except to cause the ZR38500 to return a response word to the host.

Return Version Number — VER

	7	6	5	4	3	2	1	0
Command	1	0	0	0	0	0	0	1

The VER set-up command allows the four words of the ROM version number (VERR) to be returned to the host by sending four READ commands.

Set Configuration — CFG

	7	6	5	4	3	2	1	0
Command	1	0	0	0	0	0	1	0
Parameter 1	WFA	WFB	0	AW	BW	0	WAIT	
Parameter 2	0	DB	DA	CB	MB	MA	TB	TA
Parameter 3	SB	SA	FRB			FRA		
Parameter 4	SR		MPE	0	0	0	ISP	OSP
Parameter 5	0	0	0	INW		OUTW		RAW
Parameter 6	ICS							
Parameter 7	OCS							
Parameter 8	0	0	0	SPPR				

The CFG setup command determines the input, output and external memory configurations for the ZR38500. The I/O format parameter fields (DB, DA, TA, SB, SA) of parameter words 1-3 can be set from Table 4 under the input/output ports description.

WFA	**Word/Frame** synchronization for input: 0 = Frame, 1 = Word.
WFB	**Word/Frame** synchronization for outputs: 0 = Frame, 1 = Word.
AW	**Input Word:** 0 = Wide (20 or 18 bits), 1 = Narrow (16 bits).
BW	**Output Word:** 0 = Wide (20 or 18 bits), 1 = Narrow (16 bits).
WAIT	**Wait-state** cycles for external delay memory: 0 = None, 1 = One, 3 = Seven.
CB	**Clock** source for outputs: 0 = SCKIN pin, 1 = Internal, using OCS and SPPR scalers with crystal oscillator.
MB	**Master** mode for outputs clocking: 0 = Slave. 1 = Master.
MA	**Master** mode for input clocking: 0 = Slave. 1 = Master.
FRB	**Frame** size (bits) for outputs: 0 = 16, 1 = 32, 2 = 64, 3 = 128, 4 = 192, 5 = 256, 6 = 193. Normal value = 1.
FRA	**Frame** size (bits) for input: 0 = 16, 1 = 32, 2 = 64, 3 = 128, 4 = 192, 5 = 256, 6 = 193. Normal value = 1.
SR	**Sample Rate:** 0 = 48 kHz, 1 = 44.1 kHz, 2 = 32 kHz.
MPE	**Mute Pin Enable:** 0 = Mute determined by host command, 1 = Mute determined by $\overline{\text{MUTE}}$ input pin.
ISP	**Input Word Select Polarity:** 0 = Left is WS low, 1 = Left is WS high.
OSP	**Output Word Select Polarity:** 0 = Left is WS low, 1 = Left is WS high.
INW	**Input Word:** 0 = 20 bits, 1 = 18 bits, 2 = 16 bits.
OUTW	**Output Word:** 0 = 20 bits, 1 = 18 bits, 2 = 16 bits.
RAW	**RAM Word:** 0 = Wide (16, 18 or 20 bits), 1 = Narrow (8 or 9 bits).
ICS	**Input Clock Scaler.** When input is master, SCKA clock output is at a frequency $f_A = f_{PS}/(2 \cdot ICS)$.

OCS	**Outputs Clock Scaler**. When outputs are slaves, SCKB output is at a frequency $f_B = f_{SCKIN}/(2 \cdot OCS)$. When outputs are a master, $f_B = f_{PS}/(2 \cdot OCS)$ and MCK clock output is at a frequency is $f_{MCK} = f_{PS}$.
SPPR	**Clock Pre-Scaler**. For internal sourced clocks, $f_{PS} = 2f_{XTAL}/SPPR$.

Load Program — BOOT

	7	6	5	4	3	2	1	0
Command	1	0	0	1	0	0	0	0
Parameter 1	Program Word 0							
Parameter N	Program Word N-1							

The BOOT set-up command allows serial loading and running of a program from the host. The format is given in the ZR38000 Family User's Manual and includes the start address, number of words and instructions. Execution transfers to the start address with no further action by the ROM resident executive.

Interpret Instruction — INTRP

	7	6	5	4	3	2	1	0
Command	1	0	0	1	0	0	0	1
Parameter 1	Instruction [31:24]							
Parameter 2	Instruction [23:16]							
Parameter 3	Instruction [15:8]							
Parameter 4	Instruction [7:0]							

The INTRP set-up command allows serial loading and running of a single ZR38001 core processor instruction from the host. After the single execution control transfers to the ROM resident executive with further commands possible from the host.

Set and Return I/O Register — SETIO

	7	6	5	4	3	2	1	0
Command	1	0	0	1	0	0	1	0
Parameter 1	GPIOC	GPIO						

The SETIO set-up command allows configuring and setting the output general purpose single-bit registers. The GPIO field in the parameter word is exclusive-or-ed with the six-bit GPIO register and the output bits set with the result. The input general purpose single-bit registers are read and returned in the SETIOR reply to the host following a READ command.

Bit 0 in the GPIO field is the GPIO0 port, bit 1 is the GPIO1 port, bit 2 and bit 3 are reserved, bit 4 is GPI4 and bit 5 is GPI5.

Setting bit 6 in the GPIOC field makes GPIO0 an output and setting bit 7 makes GPIO1 an output.

In normal operation with the resident ROM decoding functions, GPI5 is the $\overline{\text{MUTE}}$ input signal. This leaves GPIO0-1 and GPI4 free for user definition.

Load Memory Data — POKE

	7	6	5	4	3	2	1	0
Command	1	0	0	1	0	0	1	1
Parameters 1-3	Start Address [23:0]							
Parameters 4-7	Number of Words [31:0]							
Parameter 8	Data Word 0 [31:24]							
Parameter 7+4N	Data Word N-1 [7:0]							

The POKE set-up command allows serial loading of data and program from the host to the ZR38500 memory. N 32-bit words are loaded at the 24-bit start address. After the loading control transfers to the ROM resident executive with further commands possible from the host.

Return Memory Data — PEEK

	7	6	5	4	3	2	1	0
Command	1	0	0	1	0	1	0	0
Parameters 1-3	Start Address [23:0]							
Parameters 4-7	Number of Words (N) [31:0]							

The PEEK set-up command allows serial reading of data and program from the ZR38500 memory to the host. N 32-bit words are read from the 24-bit start address where N is a 20-bit number. 4N READ commands must be sent from the host to transfer all PEEKR reply data. After reading, control transfers to the ROM resident executive with other commands possible from the host.

Read — READ

	7	6	5	4	3	2	1	0
Command	1	0	0	0	0	0	0	0

The READ command returns a single Reply word to the host after the STAT, VER, SETIO or PEEK commands have been issued.

Status Reply — STATR

	7	6	5	4	3	2	1	0
Data Word 0	STATUS				OPCODE			
Data Word 1	RST			AC3DST			AC3IST	
Data Word 2	SR			IDR				EW
Data Word 3	0					EF	CCFG	
Data Word 4	BSID						BSM	
Data Word 5	CM		SM		DS		C	OR
Data Word 6	0				DN2			
Data Word 7	0				DN			
Data Word 8	LC2							
Data Word 9	LC							
Data Word 10	P2	RT2			ML2			
Data Word 11	P	RT			ML			

The STATR reply is a response to the STAT command. The STATUS, OPCODE and RST fields in data words 0-1 are returned for all functions by using two READ commands. The AC-3 STATR is twelve data words where data words 2-11 are recovered from the input AC-3 data stream.

STATUS	**Global Status** of function in operation: 0 = No errors, 1 = New command, updated status not available yet, 2 = Operation error in Data Word 1.
OPCODE	**Opcode** of function in operation.
RST	**Run Status**: 0 = Running, 1 = Stopped.
AC3DST	**AC-3 Decode Status**: 0 = No errors, 1 = Input status nonzero, last output block was repeated, 2 = Input status nonzero, outputs were muted, 3 = Unsupported bit-stream ID revision, 4 = Unsupported number of channels in input stream, 5 = Unsupported number of input streams.
AC3IST	**AC-3 Frame Information Status**: 0 = No errors, 1 = Invalid frame sync, 2 = Invalid sample rate, 3 = Invalid data rate.
SR	**Sample Rate**: 0 = 48 kHz, 1 = 44.1 kHz, 2 = 32 kHz.
IDR	**Input Data Rate** in kbits per second: 0 = 32, 1 = 40, 2 = 48, 3 = 56, 4 = 64, 5 = 80, 6 = 96, 7 = 112, 8 = 128, 9 = 160, 10 = 192, 11 = 224, 12 = 256, 13 = 320, 14 = 384, 15 = 448, 16 = 512, 17 = 576, 18 = 640.
EW	**Extra Word** packed: 0 = No, 1 = Yes for 44.1 kHz sample rate only.
EF	**Effects** channel for low frequency: 0 = No, 1 = Yes.
CCFG	**Coding Configuration**: 0 = Dual mono mode, 1 = 1/0, 2 = 2/0, 3 = 3/0, 4 = 2/1, 5 = 3/1, 6 = 2/2, 7 = 3/2.
BSID	**Bit-Stream Identification** number of five bits.
BSM	**Bit-Stream Mode**: 0 = Main audio service, 1 = Main audio service minus dialog, 2 = Associated service; visually impaired, 3 = Associated service; hearing impaired, 4 = Associated service; dialog, 5 = Associated service; commentary, 6 = Associated service; emergency flash.
CM	**Center Mix** level: 0 = -3 dB, 1 = -4.5 dB, 2 = -6 dB.
SM	**Surround Mix** level: 0 = -3 dB, 1 = -6 dB, 2 = None.
DS	**Dolby Surround** mode: 0 = No indication, 1 = Not Dolby Surround encoded, 2 = Dolby Surround encoded.
C	**Copyright**: 0 = Not copyright protected, 1 = Copyright protected.
OR	**Original**: 0 = Copy of an original bit-stream, 1 = Original bit-stream.
DN2	**Dialog Normalization for Channel 2** in dual mono.
DN	**Dialog Normalization** value for normal operation.
LC2	**Language Code for Channel 2** in dual mono.
LC	**Language Code** for normal operation.
P2	**Production Information for Channel 2** in dual mono operation. 0 = Does not exist, 1 = Does exist.
RT2	**Room Type for Channel 2** in dual mono operation. 0 = Not indicated, 1 = Large, 2 = Small.
ML2	**Mix Level for Channel 2** in dual mono operation.
P	**Production Information** in normal operation. 0 = Does not exist, 1 = Does exist.
RT	**Room Type** in normal operation. 0 = Not indicated, 1 = Large, 2 = Small.
ML	**Mix Level** value in normal operation.

Version Number Reply — VERR

	7	6	5	4	3	2	1	0
Data Word 0	ROM Version Number [31:24]							
Data Word 1	ROM Version Number [23:16]							
Data Word 2	ROM Version Number [15:8]							
Data Word 3	ROM Version Number [7:0]							

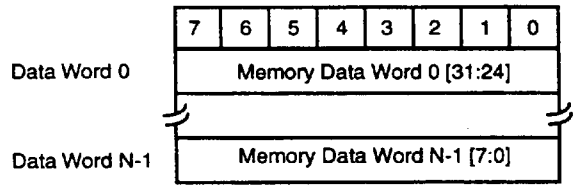
The VERR reply is a response to the VER command. It is four data words of the version number of the ROM resident executive transferred by using four READ commands.

Set I/O Register Reply — SETIOR

	7	6	5	4	3	2	1	0
Data Word 0	GPIOC	GPIO Register						

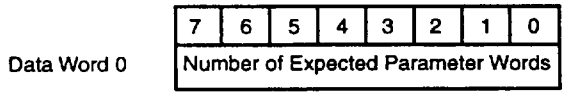
The SETIOR reply is a response to the SETIO command. It is one data word with the most recent contents of the GPIO and GPIOC registers that is transferred by one READ command.

Data Memory Reply — PEEKR



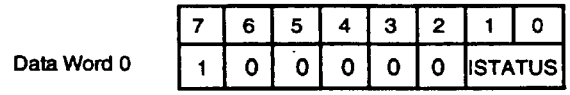
The PEEKR reply is a response to the PEEK command. It is N data words with the contents of the specified ZR38500 memory locations that is transferred by 4N READ commands.

Expected Parameters — EXPECT



EXPECT is a progress response to any command sent from the host. It is one data word returned to the host with the number of parameter words still expected by the ZR38500 at the time it was sent. It is not sent when the expected number is zero.

Interpreter Status — ISTAT



ISTAT is a progress response to any command sent from the host. It is one data word returned to the host indicating any error detected during decoding or interpretation of the command. It is returned only after the command and all parameter words are received. Errors during execution appear in the STATR reply.

ISTATUS **Interpreter Status:** 0 = No errors, 1 = Invalid opcode, 2 = Invalid parameters, 3 = Not ready to accept new commands.

Table 1: Command Summary (Host to ZR38500 Transfers)

Class	Name	Operation code	Number of parameter words	Description
Write				Commands to ZR38500 to perform a specific function or operation
	AC3	85	8	Select AC-3 decoder function, either six- or two-channel input
	PROL	86	4	Select Pro Logic decoder function with PCM input
	PCM	84	3	Select PCM mixer function
	PNG	83	3	Select pink noise generator function
	USER	88	8	Select user defined function
	PLAY	8A	0	Resume selected function operation and unmute audio output
	MUTE	8B	0	Mute audio output without stopping the selected operation
	UNMUTE	89	0	Restore muted audio output while continuing the selected operation
	STOP	8C	0	Stop operation, retain data in input buffer and mute audio output
	STOPF	8D	0	Stop operation, flush the data in the input buffer and mute audio output
	STAT	8E	0	Return decoder status information using the READ command
	NOP	80	0	Not a command, does not affect operation. Will return a Progress response.
	VER	81	0	Return 32-bit ROM version number using the READ command
	CFG	82	8	Configure the ZR38500 I/O to the specific system hardware
	BOOT	90	N	Load and execute the N parameter words of bootstrap program
	INTRP	91	4	Interpret: load and execute four parameter words as ZR38001 instruction
	SETIO	92	1	Set, test and return general purpose single-bit I/O registers
	POKE	93	7+4N	Load N 32-bit words to the core processor RAM at the given start address
	PEEK	94	7	Read N 32-bit words from core processor RAM at the given start address
Read				Commands to ZR38500 to return Reply words to the host
	READ	00	0	Command to ZR38500 to return a Reply word after specific commands

Table 2: Response Summary (ZR38500 to Host Transfers)

Class	Name	Operation code	Number of data words	Description
Reply				Data words returned to the host as the result of sending specific commands followed by READ commands
	STATR	-	Variable	Decoder status information read by STAT command
	VERR	-	4	Four byte version number of ROM read by VER command
	SETIOR	-	1	Two bits of general purpose I/O registers set and read by SETIO command
	PEEKR	-	4N	N 32-bit words from core processor RAM specified by PEEK command
Progress				Data words returned to host in the normal process of sending any command
	EXPECT	-	1	Expected number of parameter words still to be received from host
	ISTATUS	-	1	Interpreter status

DECODER OPERATION

There are two methods of controlling the operation of the ZR38500 for decoding. One is using the command structure described in the previous section with a host microprocessor. The other is calling the decoding function subroutines through the Applications Programming Interface (API) within a user written program on the ZR38500's core processor. Either allows for user written programs to execute in addition to the audio decoding functions. The second method is recommended only to the advanced user.

This section describes operation principles when using the specific command structure and configuration choices in the standard ROM coded program. Following sections describe the operation of the core processor and its full range of options as a programmable device.

START-UP

After being reset by the system $\overline{\text{RESET}}$ signal, the ZR38500 will check for an external program ROM to load. Not finding this it will start execution from the internal ROM and await commands from the host. Normal host operation would confirm the ROM version number and then configure the ZR38500 to match the system and desired operation. To provide user functions individual core processor instructions may be executed using the INTRP command or whole programs loaded and executed with the BOOT command. Data required by the programs may be exchanged with the host with the PEEK and POKE commands or the USER command.

INPUT/OUTPUT CONFIGURATION

Using core processor instructions, the I/O can be configured using all of the flexibility of the processor interfaces for special applications. However, the configuration (CFG) and set I/O (SETIO) commands accommodate most needs.

Connections to the data stream input, the output D/A converters and the single-bit general purpose registers are made through the input/output ports. There are eight single-bit ports: GPIO(1-0), digital audio input Port A, digital audio output Ports B, C and D, GPI4, and $\overline{\text{MUTE}}$ or GPI5. Four transfer data in various bit-serial formats (Ports A through D) and four as programmable bits in an internal register.

Serial Ports

The bit-serial ports serve a variety of peripheral device conventions. Their operation is determined solely by the Set Configuration (CFG) command. The input Port A and output Ports B-D have separate clocking systems and may be individually selected with the ZR38500 acting as a master or a slave. A master clock output has a programmable rate as do the two internally generated input and output master bit-rate clocks.

If input port A is a master, SCKA is at a frequency $f_A = f_{PS}/(2 \cdot ICS)$ where $f_{PS} = 2f_{XTAL}/SPPR$ is derived from the internal oscillator and its clock pre-scaler. The dividers ICS and SPPR are fields in the configuration command parameter words. Likewise, when output ports B-D are master, SCKB is at a frequency $f_B = f_{PS}/(2 \cdot OCS)$ and the master clock output (MCK/SCKIN) is at the frequency f_{PS}. The outputs are unique in that when operating as a master their clock outputs can also be derived from an externally supplied master clock input (MCK/SCKIN) with the programmable divider rate OCS. This selection is made with the CB field in the configuration command. These choices are summarized in Table 3.

Many choices are possible for the bit-serial port formats and word sizes. The five most commonly used formats are summarized in Table 4 along with the selectable word sizes. Waveforms for each are illustrated in Figures 4-8. Clocking for both master and slave operation is shown. The transitions marked are the edges where data changes when the ZR38500 is a master or where the data is sampled when it is a slave. Settings for the appropriate fields in the configuration command are also summarized.

Word select (WS) or frame synchronization (FS) is chosen with the WFA and WFB fields. The polarity of the WS signal is chosen with the ISP and OSP fields. Either polarity is acceptable on any of the word select formats 0-2. The DA and DB fields choose formats where the serial data (SD) is not delayed by one bit-

clock from the WS or FS signal. The TA field chooses the frameless input operation of Figure 7 when AW = 1. The input then is sampled every SCKA and an interrupt generated after 16 bits have been received. Note that when a master the FS signal shown is, in fact, generated and if FS is asserted when a slave it will resynchronize the data as shown. The SA and SB fields with AW, BW = 0, choose the least-significant-bit-justified 32-bit word format of the EIAJ standard of Figure 5.

Table 3: Serial Ports A & B Clocking Command Summary

Function	Input Port A	Input Port B
Master or slave clocking mode	MA field	MB field
External master clock input SCKIN/MCK pin (f_{SCKIN})	None	CB field = 0
Internal master clock output SCKIN/MCK pin ($f_{MCK} = f_{PS}$)	None	CB field = 1
5-bit pre-scaler for internal master clock and MCK pin output	SPPR field ($f_{PS} = 2f_{XTAL}$/SPPR)	
Internal 8-bit master clock scalers for SCKA and SCKB. CB = 1.	ICS field ($f_A = f_{PS}$ / [2•ICS])	OSC field ($f_B = f_{PS}$ / [2•OCS])
External 8-bit master clock scaler for SCKB. CB = 0.	None	OCS field ($f_B = f_{SCKIN}$ / [•OCS])

Table 4: Input and Output Format Selection

Format		Figure	Configuration Command Fields						
			WFA or WFB	ISP or OSP	DA or DB	TA	SA or SB	INW Input Word Size	OUTW Output Word Size
0	I^2S	4	1	0	0	0	0	16, 18, 20 bits	16, 18, 20 bits
1	EIAJ	5	1	1	0	0	1	16, 18, 20 bits	16, 18, 20 bits
2	Non-Delayed	6	1	1	1	0	0	16, 18, 20 bits	16, 18, 20 bits
3	Frameless	7	1	0	0	1	0	16 bits	not applicable
4	Frame Sync	8	0	0	0	0	0	16, 20 bits	16, 20 bits

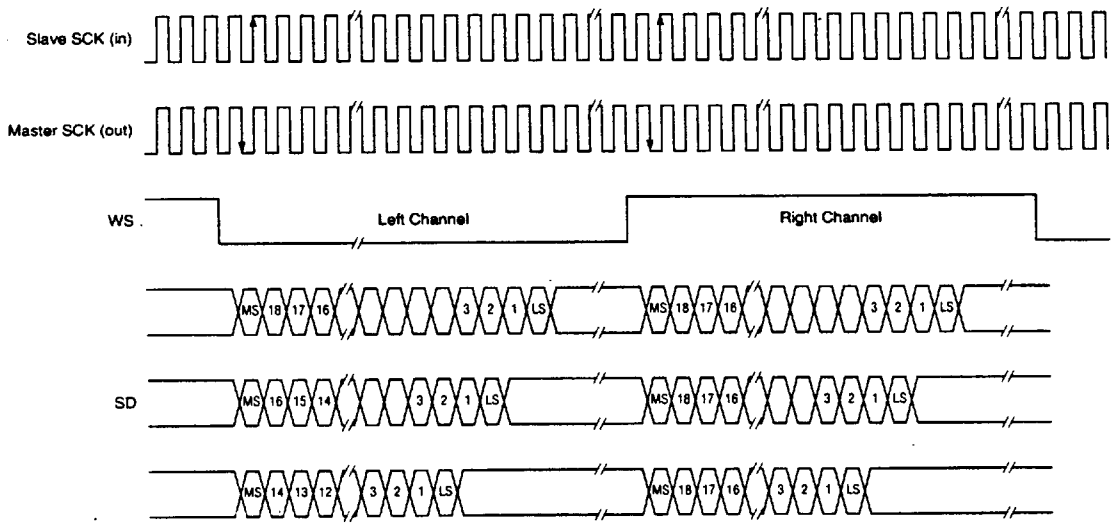


Figure 4. I^2S Input/Output – Format 0

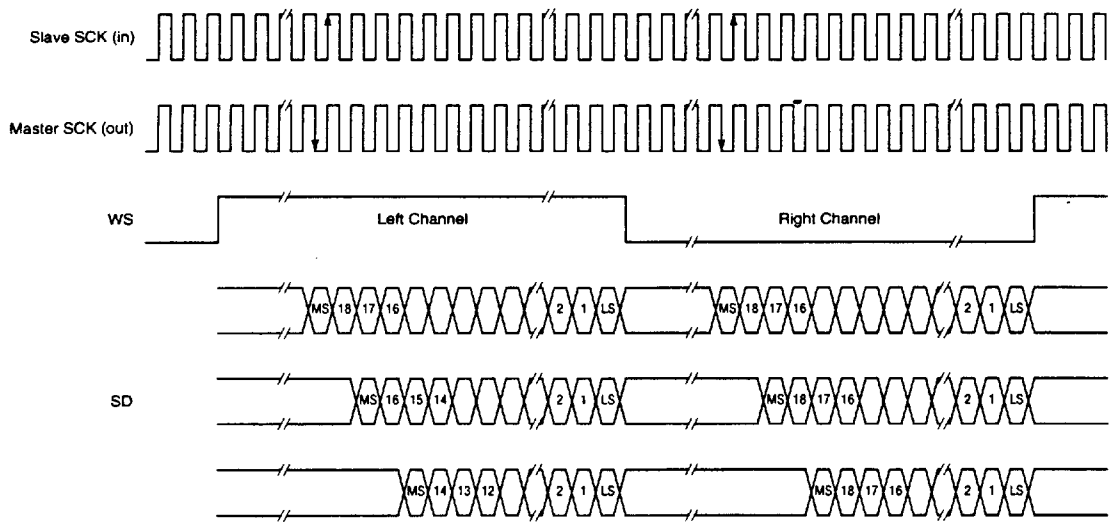


Figure 5. EIAJ Input/Output – Format 1

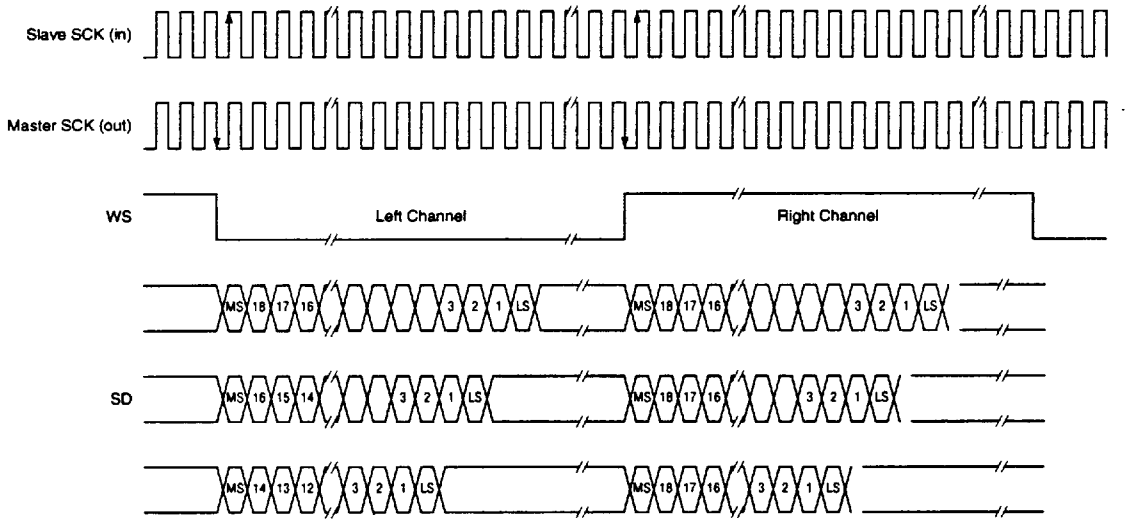


Figure 6. Non-Delayed Input/Output – Format 2

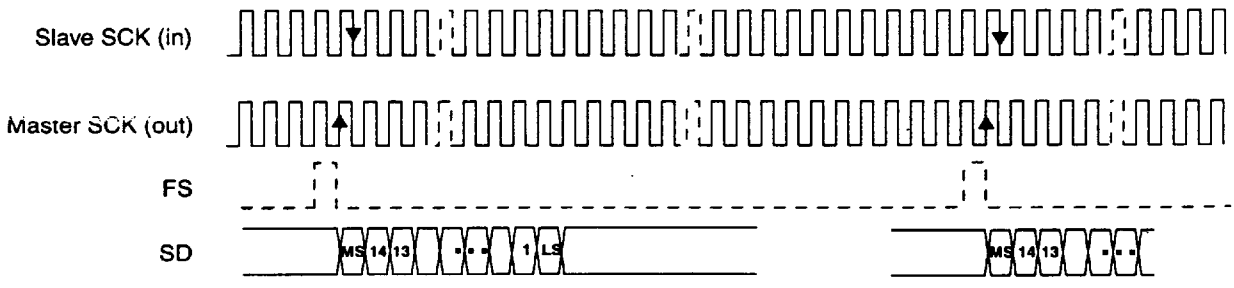


Figure 7. Frameless Input. Format 3.

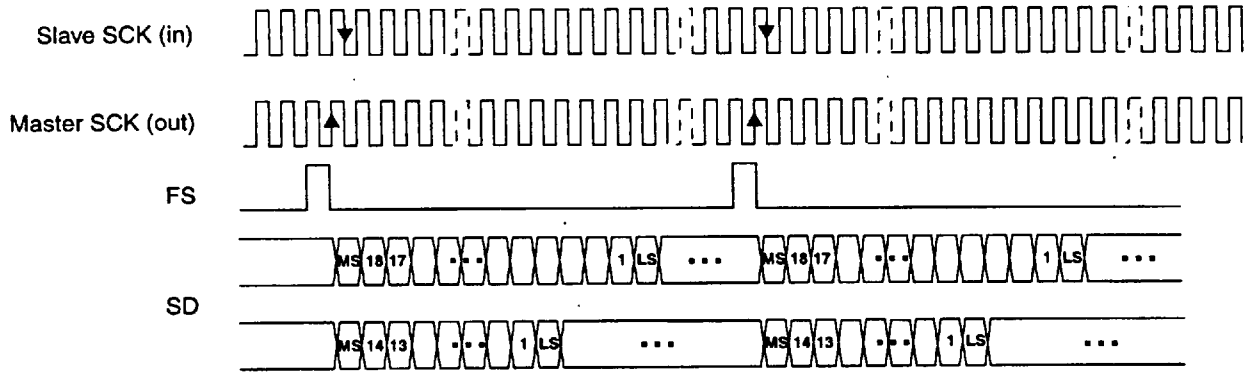


Figure 8. Frame Sync Input/Output. Format 4.

General Purpose Ports

There are four single-bit general purpose ports normally configured as inputs: GPIO(1-0) and GPI(5-4). GPI5 is normally used as the $\overline{\text{MUTE}}$ input signal. If the $\overline{\text{MUTE}}$ pin is not enabled in the configuration command with the MPE bit, then it may also be used as a user defined input. GPIO(1-0) may be configured as user defined outputs by setting the GPIOC field in the SETIO command. Inputs pins GPIO(1-0) and GPI(5-4) are sampled and read from the GPIO register by the host with the SETIO command. At the same time it can set the state of output pins GPIO(1-0) if they have been configured as outputs.

PROCESSOR GENERAL DESCRIPTION

With its versatile internal architecture (see Figure 2), general purpose instruction set and high speed, the ZR38500's core processor is also capable of executing many other types of algorithms for a wide variety of DSP applications. These algorithms can add differentiating product features to the basic audio decoding functions. With the ZR38500's state-of-the-art performance these additional features take little processing time or program memory.

A 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.
- 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 ZR38500's four-level zero-overhead loop and repeat instructions, produce very compact code. Most instructions execute in a single cycle.

The ZR38500 uses an internal clock rate of 66 MHz to achieve 33-million instructions per second (33-MIPS) performance. This allows accessing internal data memory twice per instruction cycle. An internal phase-locked loop (PLL) 2x clock multiplier circuit permits a 33 MHz external crystal or input clock to be used.

The ZR38500's optimized 20-bit (120 dB) data precision make it particularly well suited for compact disk-quality audio applications including audio equalization, special effects and audio mixing 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), ZR38500-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 ZR38500'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 ZR38500 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 ZR38500 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.

The ZR38500 has many built-in memory resources. A large 1k x 32-bit program/data RAM is available on-chip with an additional 1.5k of program/data space usable in the mask programmable 12k x 32-bit ROM. The already large internal 4k x 20-bit data RAM can be extended off-chip via the 20-bit external data bus and 16-bit memory address bus, allowing it to address up to 64k

data words in a unified address space. A separate address space allows an addition 64k data words of external ROM also. Programmable wait-states accommodate lower-cost slow external memories and byte-wide configurations can be used for lower chip count if desired.

PROCESSOR FUNCTIONAL DESCRIPTION

ARCHITECTURAL OVERVIEW

Figure 2 shows the detailed functional units of the ZR38500 processor. 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 smaller single external buses for external data memory. 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 ZR38500 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 9. 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.

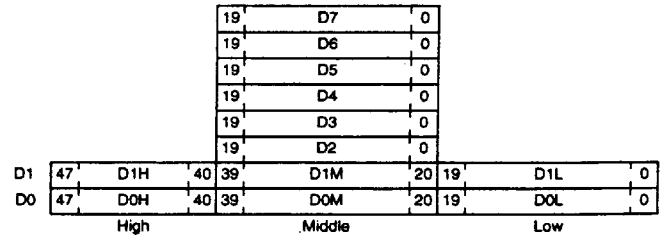


Figure 9. Data Register File

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 10. 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 10. 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. When the RAM is written to the most significant 12 bits are loaded at the same time from the Data Bus Extension (DBX) register. When the RAM is read as data the DBX register is loaded with the most significant 12 bits of data. The DBX register can also be loaded or read as a general register with data in the least significant 12 bits.

The ROM is always at locations Hex E0000 to E2FFF in memory space on both Address Buses. The standard ZR38500 product has the ROM coded with the digital audio decoder functions and a bootstrap program for accepting commands from a host or loading an operating program into RAM from a byte-wide external ROM. The upper 1.5 Kwords of internal 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 mask programmed internal ROM when the ZR38500 is used with an external byte-wide bootstrap ROM.

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 11.

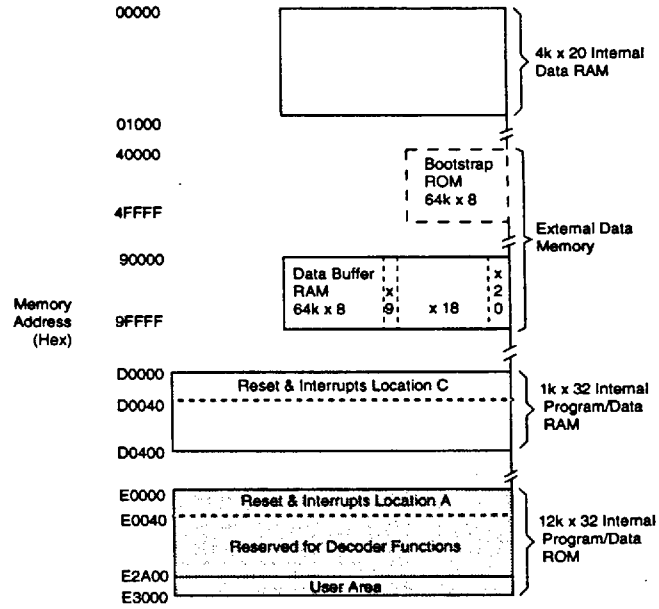


Figure 11. Program/Data Memory Map

External

Data memory is extended externally on the Parallel Port in the two 64k memory address spaces shown in Figure 11. Internal address buses and data buses are multiplexed into single smaller buses for external memory. Thus only one external data transfer can take place at a time. Also, only a single transfer can be made in each instruction cycle due to the slower external memories. This memory cycle-time can be lengthened by insert-

ing 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. The addresses shown are the internal 20-bit ones. Externally the two 16-bit addresses for the ROM and RAM are determined by the MS memory select signal.

The optional external bootstrap ROM is 8-bits wide and right justified on the data bus. Various widths of memory can be used for the RAM data buffer required for decoder operation. As shown for the four choices of 8, 9, 18 or 20 bits, the data buffer RAM must be left justified on the data bus.

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 11 these can be located at the lowest portion of the on-chip 12k x 32-bit ROM or 1k x 32-bit RAM. This is selected by the MM and PM bits in the Mode Register as follows:

Table 5: Reset and Interrupt Start Locations

MM bit	PM bit	Reset & Interrupts Location	Start Address (Hex)
0	0	A - Internal ROM	E0000
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 ZR38500 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 E0000 before unconditionally jumping to the beginning of the Reset and Interrupt block (shown in Figure 12) to start executing the reset service routine. The complete service routine may be read from an external bootstrap device and in turn, executed.

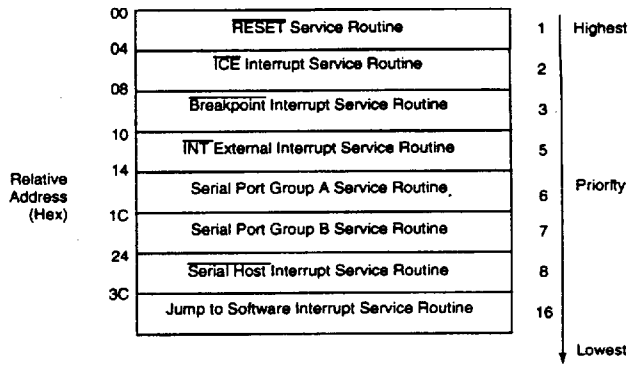


Figure 12. Reset and Interrupt Block Memory Map

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 seven hardware interrupts and the software Interrupt, have their corresponding vector addresses and the priority shown in Figure 12. 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 or internal RAM or ROM), 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 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. 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 ZR38500 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 6 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 7. It is here that the full power of the ZR38500 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:

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 6: Instruction Set Summary

Instructions				
Arithmetic Unit			Address Generation Unit	Program Sequence Unit
Arithmetic	Logic	Multiplier		
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

Table 7: Instruction Class Summary Table

I. Single operand ALU operations with parallel transfer operations

Class Code	Op-code	Oper-and	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

Connections to external memory and peripherals are made through the input/output ports. There is a single 20-bit parallel port and eight single-bit ports; four transfer data in various bit-serial formats and four as programmable bits in an internal register.

Parallel Port

The external memory and parallel interface consists of the 16-bit address bus A(15-0), the 20-bit bi-directional data bus D(19-0), and the control signals MS, $\overline{RD}$ and $\overline{WR}$.

MS (A18 internally) is asserted high whenever there is an access to external ROM as opposed to RAM. This eliminates external address decoding in most systems. $\overline{RD}$ is asserted during an external read cycle, and can be used as an output enable for memory. $\overline{WR}$ is asserted during an external write cycle and can be used as a write enable for memory.

The ZR38500 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, the $\overline{RD}$ and $\overline{WR}$ 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, $\overline{RD}$ is also active.

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

Serial Ports

The serial ports are flexible on the ZR38500 to serve a wide variety of applications and peripheral device conventions. The input 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 I²S format, the frame-less time-division-multiplex (TDM) format and the 20-bit word in an LSB justified 32-bit frame of the Japanese DAC format are all supported.

Port A is always data input, 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. In normal decoder operation of the ZR38500 the input is in A group and all outputs are in B group. 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 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 advanced by one bit interval for the non-I^2S 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

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. The other two ports are input-only.

Table 8: Serial Ports Function Summary

Function	A Group	B Group
Grouping: AB = 0	1 Input (A)	3 Outputs (B, C, D)
Grouping: AB = 1	1 Input (A), 1 Output (D)	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^2S format	Yes	Yes
TDM format	Yes	Yes
Japanese DAC format	Yes	Yes

SYSTEM INTERFACE

The system interface consists of all external signal functions other than Input/Output Ports 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 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; B Group is ports B, C. When cleared, A group is port A; 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.
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.

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

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.
WS	**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 for data. When reading data from 32-bit wide 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 internal RAM, 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 9: 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 ZR38500 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 with parallel eight-bit registers. The bootstrap ROM can accept commands or 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 ($\overline{SS}$) 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. Operation is shown in Figure 13.

In Circuit Emulation Interface

The ZR38500'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 $\overline{RESET}$ 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 processor will not, however, accept a serial host command and return a response until 200 instruction cycles have elapsed.

The external interrupt input signal $\overline{INT}$ 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 con-

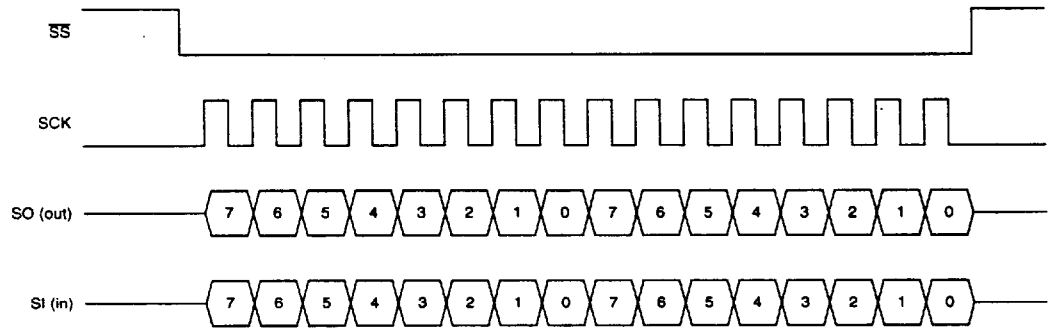


Figure 13. Example Two-Byte Transfer with Host Interface (SCKP = 0)

nection to enable the internal oscillator. The maximum frequency is 33 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 14. A parallel-resonant fundamental-mode crystal should be used.

Table 10: ZR38500 Signal Summary

Name	Number	Type*	Description
A(15-0)	16	O	Address bus of parallel I/O port
D(19-0)	20	I/O	Data bus of parallel I/O port
MS	1	O	Memory select enable for $\overline{\text{RAM}}$ /ROM selection on parallel I/O port
RD	1	O	Read enable for read operation on parallel I/O port
WR	1	O	Write enable for write operation on parallel I/O port
SDA	1	I	Serial data input. Port A.
WSA/FSA	1	I/O	Word select or frame synchronization for input port A. An output when a master, an input when a slave.
SCKA	1	I/O	Serial clock for input port A. An output when a master, an input when a slave.
SDB	1	O	Serial left and right data output. Port B.
SDC	1	O	Serial left and right surround data output. Port C.
SDD	1	O	Serial center and sub-woofer data output. Port D.
WSB/FSB	1	I/O	Word select or frame synchronization for output ports. An output when a master, an input when a slave.
SCKB	1	I/O	Serial clock for output ports. An output when a master, an input when a slave.
MCK/SCKIN	1	I/O	Master clock output or master clock input for output ports
GPIO0	1	I/O	Can be programmed as general purpose input/output pin GPIO0
GPIO1	1	I/O	Can be programmed as general purpose input/output pin GPIO1
GPI4	1	I	Can be programmed as general purpose input pin GPI4. [1]
$\overline{\text{MUTE}}$	1	I	Mutes audio output or pin can be programmed as GPI5. [2]
SI	1	I	Host serial interface data input
SO	1	O	Host serial interface data output
SCK	1	I	Host serial interface clock input
$\overline{\text{SS}}$	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
$\overline{\text{INT}}$	1	I	External interrupt request input
$\overline{\text{RESET}}$	1	I	Reset input to start operation in known state
XTI	1	I	External clock input or connection to external crystal
XTO	1	O	Other connection to external crystal
VCC		Power	+5 volt power supply
GND		Power	Power supply ground

* O = output, I = input, T = tristatable. After reset, all I/O type pins are inputs and O/T type pins are tristated.

1. This pin can also be used as serial data input port E.
2. This pin can also be used as serial data input port F.

TYPICAL CONFIGURATIONS

Figure 14 shows a ZR38500 in a typical low-cost stand-alone configuration without a host microprocessor. A data RAM is used for surround delay and a byte-wide ROM for loading the user written program that governs the decoder operation. A 32k x 16 RAM is shown in the typical configuration, although only 16k words are required. The RAM data width is 16 bits for 16-bit output precision, and 20 bits for 20-bit output precision. The ROM is shown as optional because it could be eliminated by using a custom addition to the internal masked ROM of up to 1.5k x 32-bits.

The compressed data stream is input through a SPDIF receiver that acts as a clock master for the output D/As. This clocking master at a x256 master clock rate is derived from the initial source of the compressed signal. The internal clock divider on the ZR38500 generates the clocking for the three slave D/As that provide the six-channel audio output.

The standard ZR38500, without being custom ordered with a user's program, has a $\overline{\text{RESET}}$ bootstrap loading routine in its internal ROM. The internal loading routine reads from an external byte-wide ROM (shown) which has the to-be-executed program and data. The boot-strap recognizes the external ROM rather than waiting for commands from a host that does not exist in this configuration.

Figure 15 shows a ZR38500 in a typical system configuration with a host processor. At reset time the internal ROM bootstrap will check for the byte-wide external ROM. Not finding that, it will then expect to down-load program commands from the host through the serial connection. A resistive pull-up on any one data line D(7-0) assures that an external ROM will not be found.

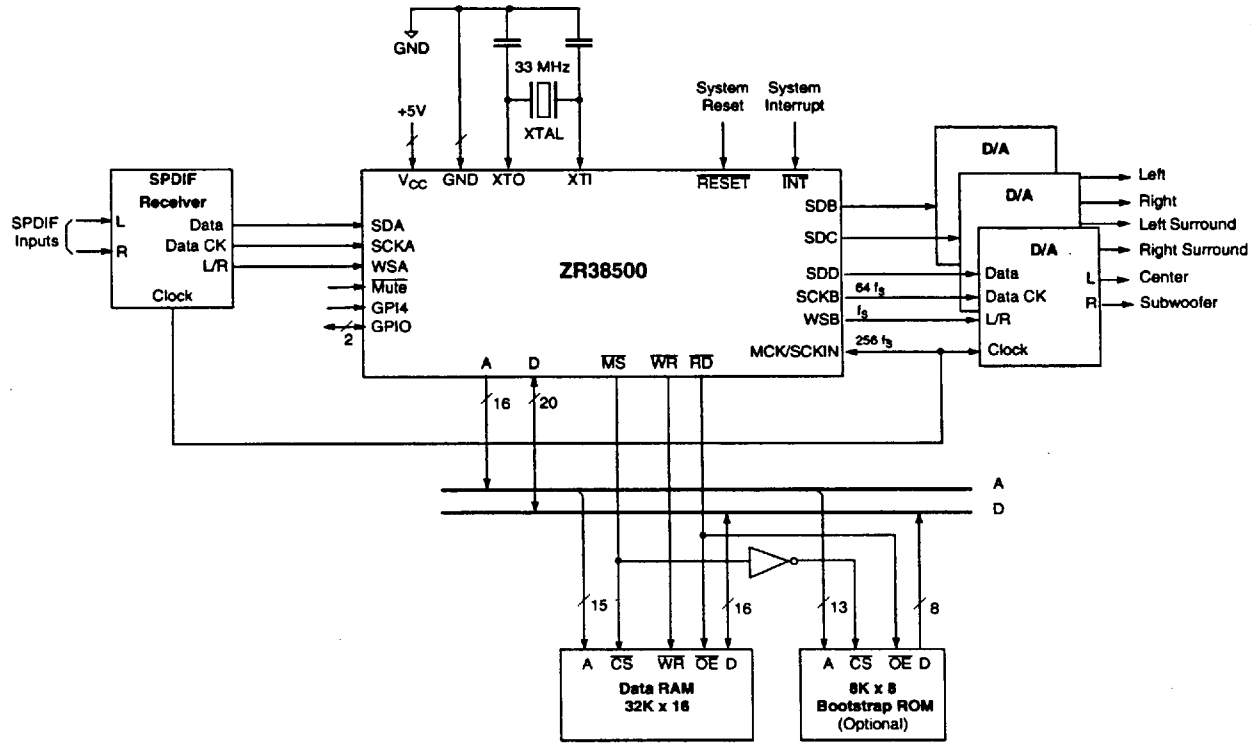


Figure 14. ZR38500 Typical Stand-Alone Configuration

PRODUCT SUPPORT

DOCUMENTATION

This data sheet is a summary description of the ZR38500's functional operation and instruction set. It includes the complete electrical, timing and physical description. The complete source of information on its programmed operation and instruction set is the "ZR38000 Family Users Manual and Programming Reference". The "ZR38500 Application Programming Interface (API)" should be consulted for writing user programs which call the decoding routines in the internal ROM. 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 ZR38500. 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 ZR38500.

DEVELOPMENT BOARD

The ZR38001 Development Board is a PC/AT compatible add-in board and software for real-time operation, testing and debugging of ZR38500 programs. The limitation on external memory of the ZR38500 must be observed when using this board but otherwise instructions will be the same.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.

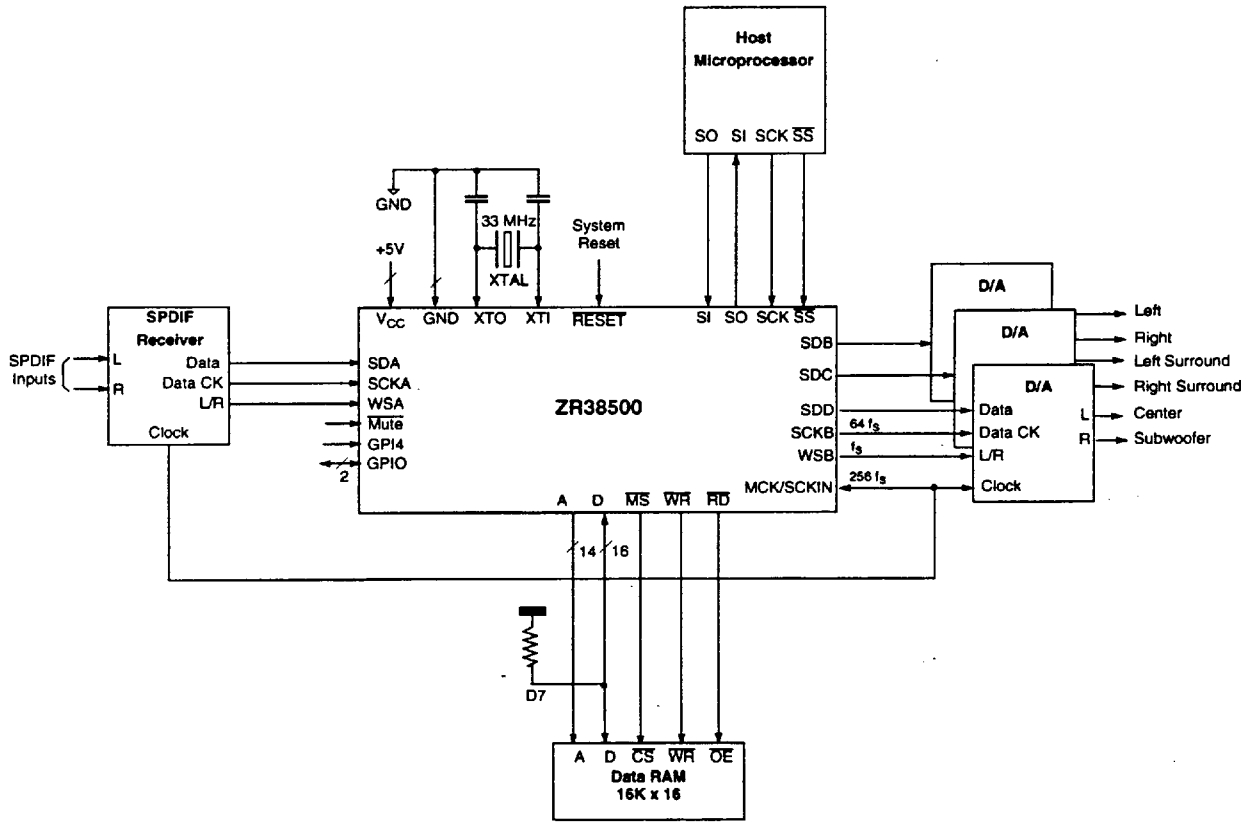


Figure 15. ZR38500 Typical Configuration With Host

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 ..... $0°C \le T_A \le +70°C$
Supply Voltage ..... $4.75V \le V_{CC} \le 5.25V$

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	2.6	–	–	V	
V_{OL}	Output Low Voltage	–	–	0.4	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4	–	–	V	I_{OH} = 0.4 mA
I_{CC}	Power Supply Current	–	350	400	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_{PU}	Pull-Up 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 16. AC Testing Input, Output

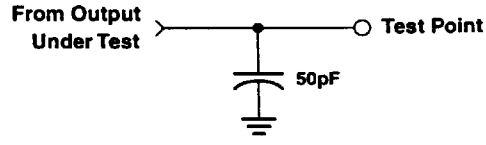


Figure 17. Normal AC Test Load

AC CHARACTERISTICS

Memory Read (External Clock Frequency = 33 MHz)

	Input Requirements	Min	Max	Units	Notes
5	Data in hold from $\overline{RD}$ rising edge	0	--	ns	
7	$\overline{RD}$ low to data valid	--	6	ns	Note 1
9	Address stable, $\overline{CS}$ low to data valid	--	12	ns	Note 1

	Output Characteristics	Min	Max	Units	Notes
1	Read cycle width	30	--	ns	Note 1
2	Address memory select hold from $\overline{RD}$ rising edge	1	--	ns	
3	Address memory select setup to $\overline{RD}$ falling edge	3	--	ns	
4	$\overline{RD}$ pulse width	16	--	ns	Note 1
6	$\overline{RD}$ after $\overline{RD}$ recovery time	8	--	ns	
8	$\overline{WR}$ after $\overline{RD}$ recovery time	8		ns	

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

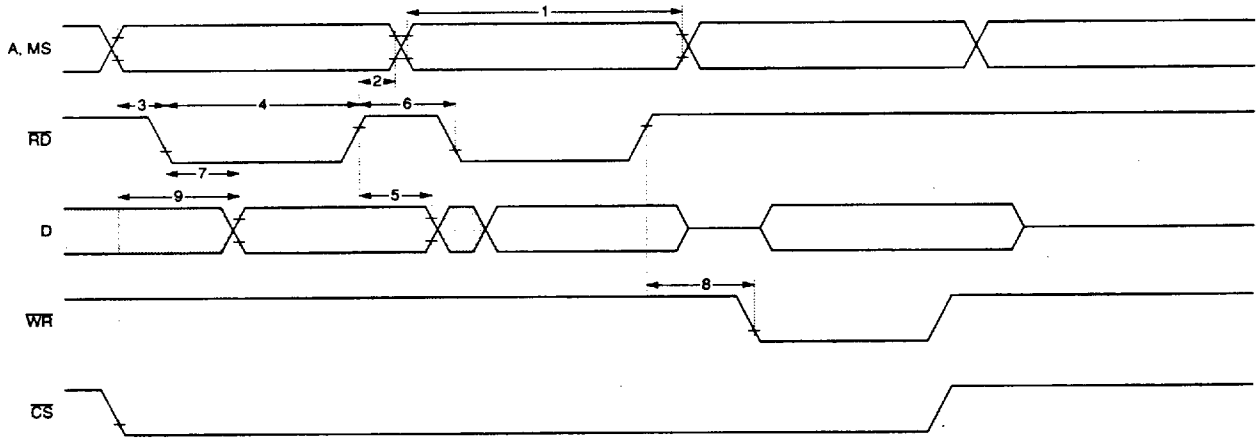


Figure 18. Memory Read

Memory Write (External Clock Frequency = 33 MHz)

Output Characteristics		Min	Max	Units	Notes
10	Write cycle width	30	–	ns	Note 2
11	Address memory select setup to $\overline{WR}$ falling edge	3	–	ns	
12	$\overline{WR}$ pulse width	15	–	ns	Note 2
13	Address memory select hold from $\overline{WR}$ rising edge	1	–	ns	
14	Data out setup to $\overline{WR}$ rising edge	8	–	ns	Note 2
15	Data out hold from $\overline{WR}$ rising edge	2	–	ns	
16	$\overline{WR}$ low to data out enabled	3	–	ns	
17	$\overline{WR}$ after $\overline{WR}$ recovery time	8	–	ns	
18	$\overline{RD}$ after $\overline{WR}$ recovery time	8	–	ns	
19	Data disable to $\overline{RD}$ after $\overline{WR}$	5	–	ns	

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

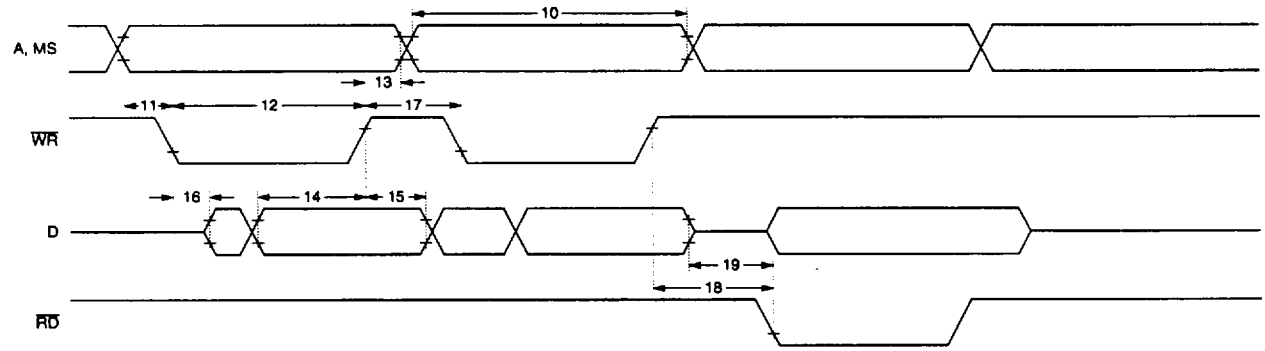


Figure 19. Memory Write

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	5	–	ns	Slave Mode (MA = 0)
23	FSA hold time from SCKA falling edge	7	–	ns	Slave Mode (MA = 0)
24	SDA setup time to SCKA falling edge	8	–	ns	
27	SDA hold time from SCKA falling edge	8	–	ns	

	Output Characteristics	Min	Max	Units	Notes
70	FSA output delay from SCKA rising edge	–	15	ns	Master Mode (MA = 1)
71	SCKA period	8 x t_{XTI}	2^{16} x t_{XTI}	ns	Master Mode (MA = 1)

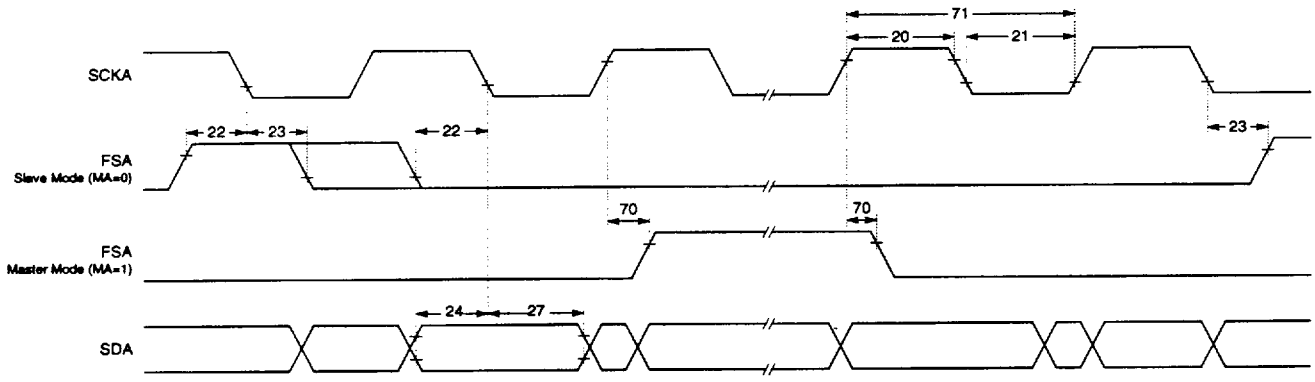


Figure 20. A-Group Serial Ports (Frame Sync Mode)

A-Group Serial Ports (Word Select Mode)

	Input Requirements	Min	Max	Units	Notes
31	WSA setup time to SCKA rising edge	5	–	ns	Slave Mode (MA=0)
32	WSA hold time from SCKA rising edge	6	–	ns	Slave Mode (MA=0)
72	WSA hold time from SCKA falling edge	3	–	ns	SPMODE: DA = 1, MA = 0
33	SDA setup time to SCKA rising edge	8	–	ns	
36	SDA hold time from SCKA rising edge	7	–	ns	

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

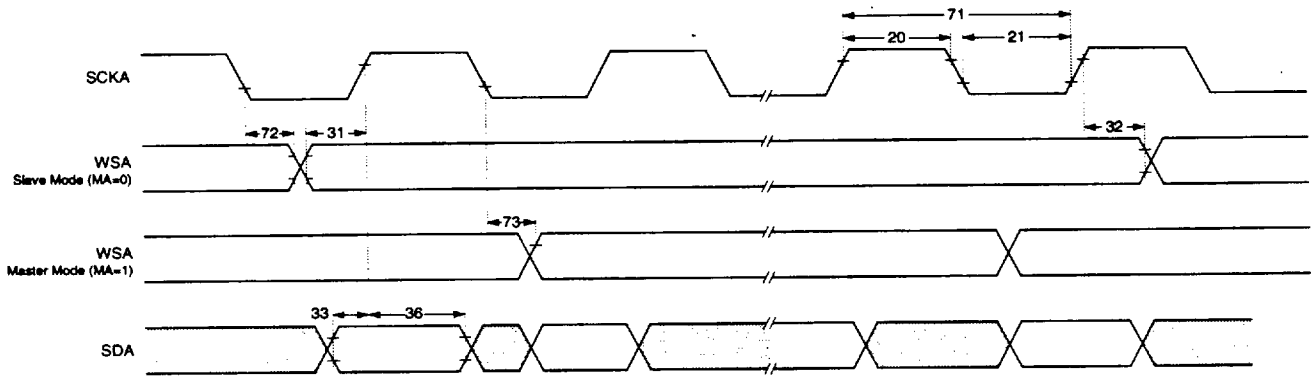


Figure 21. A-Group Serial Ports (Word Select Mode)

B-Group Serial Ports (Frame Sync Mode)

	Input Requirements	Min	Max	Units	Notes
41	SCKIN low width	t_{XTI}	--	ns	CB = 0
42	SCKIN high width	t_{XTI}	--	ns	CB = 0
75	SCKB high width	t_{XTI}	--	ns	Slave Mode (MB = 0)
76	SCKB low width	t_{XTI}	--	ns	Slave Mode (MB = 0)
78	FSB setup time to SCKB falling edge	5	--	ns	Slave Mode (MB = 0)
79	FSB hold time from SCKB falling edge	5	--	ns	Slave Mode (MB = 0)

	Output Characteristics	Min	Max	Units	Notes
40	SCKIN period (t_{SCKIN})	$2t_{XTI}$	$16t_{XTI}$	ns	CB = 1
43	SCKB period	$4t_{SCKIN}$	$4096t_{SCKIN}$	ns	Master Mode (MB = 1), CB = 0
44	FSB delay from SCKB rising edge	--	15	ns	Master Mode (MB = 1)
45	SDB, SDC, SDD delay from SCKB rising edge	--	20	ns	

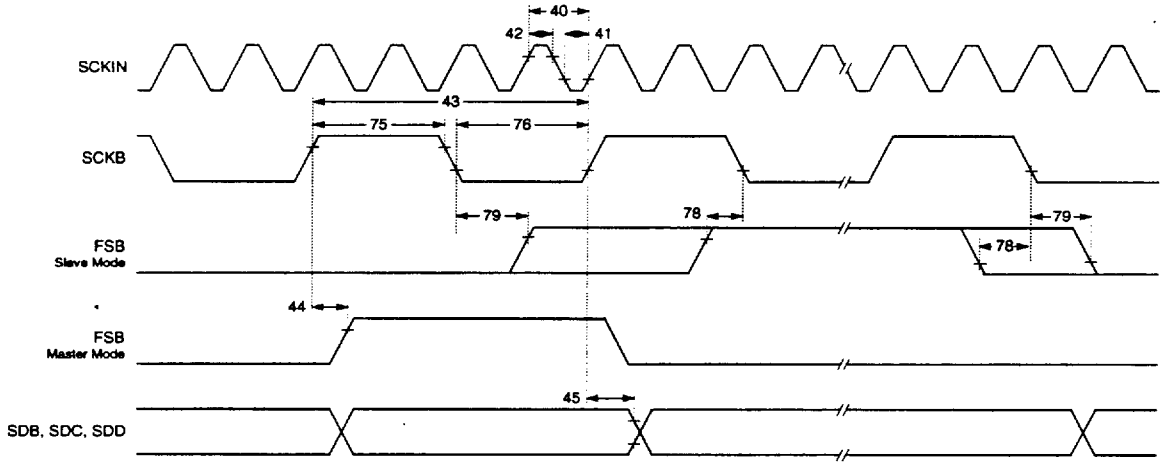


Figure 22. B-Group Serial Ports (Frame Sync Mode)

B-Group Serial Ports (Word Select Mode)

Input Requirements		Min	Max	Units	Notes
81	WSB setup to SCKB rising edge	5	–	ns	Slave Mode (MB = 0)
82	WSB hold from SCKB rising edge	6	–	ns	Slave Mode (MB = 0)
83	WSB hold time from SCKB falling edge	6	–	ns	DB=1, Slave Mode (MB = 0)

Output Characteristics		Min	Max	Units	Notes
50	WSB delay from SCKB falling edge	–	15	ns	Master Mode (MB=1)
51	SDB, SDC, SDD delay from SCKB falling edge	–	25	ns	SDD in Mode AB=0
52	SDB, SDC, SDD delay from WSB	–	20	ns	SPMODE: DB=1, MB=0; SDD in Mode AB=0

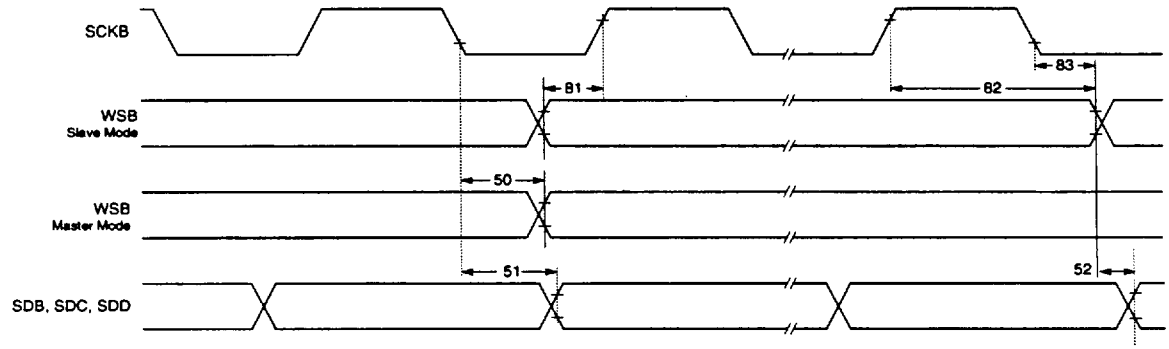


Figure 23. B-Group Serial Ports (Word Select Mode)

Serial Host Interface Timing

Characteristics		Min	Max	Units	Notes
111	SCK clock period	$6t_{XTI}$	–	ns	Note 1, 2
112	SCK clock high width	$3t_{XTI}$	–	ns	
113	SCK clock low width	$3t_{XTI}$	–	ns	
114	$\overline{SS}$ setup time to first SCK edge	10	–	ns	
115	$\overline{SS}$ hold time from last edge of SCK	10	–	ns	
116	SI setup time to SCK active edge	10	–	ns	
117	SI hold time from SCK active edge	10	–	ns	
118	$\overline{SS}$ negation to data Hi-Z	–	10	ns	
119	SO delay from SCK active edge	–	15	ns	

1. SCK polarity is controlled by field SCKP of register SPIMODE. The polarity shown in Figure 24 corresponds to SCKP=0.
2. When using the ROM-based serial host control shell, the period of SCK should be larger than 320 t_{XTI}.

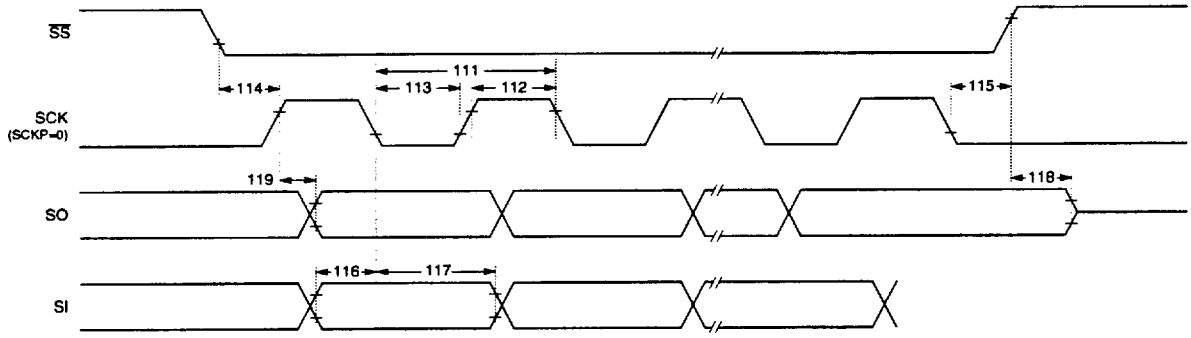


Figure 24. Serial Host Interface Timing

External Clock

		Min	Max	Units	Notes
56	XTI period (t_{XTI})	30	200	ns	Note 1
57	XTI high width	12		ns	
58	XTI low width	12		ns	

1. For six-channel Dolby AC-3 decoding t_{XTI} = 30 ns.

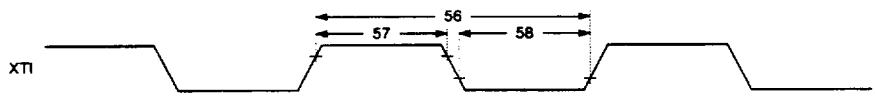


Figure 25. External Clock

Reset

		Min	Max	Units	Notes
59	$\overline{\text{RESET}}$ width, warm reset	$3t_{XTI}$	127 t_{XTI}	ns	
60	$\overline{\text{RESET}}$ width, cold reset	$200t_{XTI}$		ns	Note 1
61	Memory bus enable after $\overline{\text{RESET}}$ rising edge, warm reset	t_{XTI} +10	$2t_{XTI}$ +15	ns	
62	Memory bus disable after $\overline{\text{RESET}}$ falling edge	t_{XTI} +10	$2t_{XTI}$ +15	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.

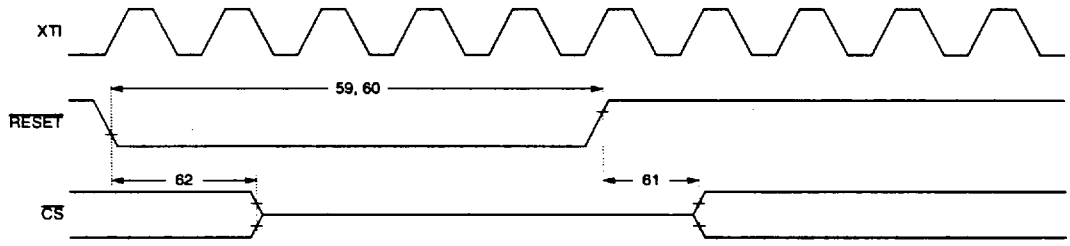


Figure 26. Reset

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.

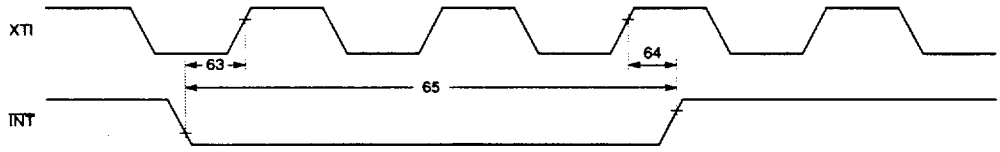


Figure 27. External Interrupt

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	

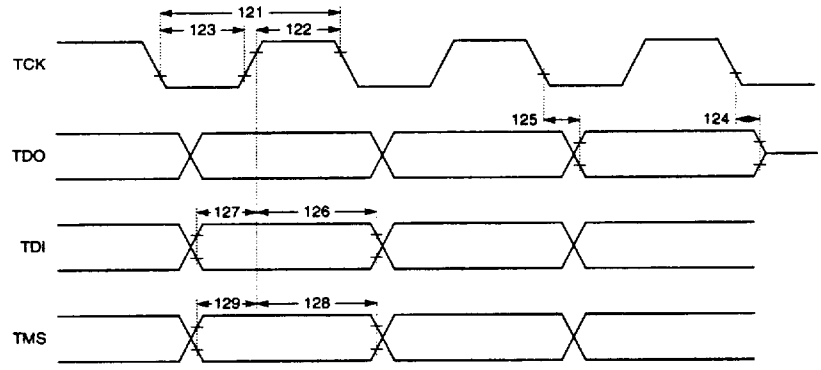


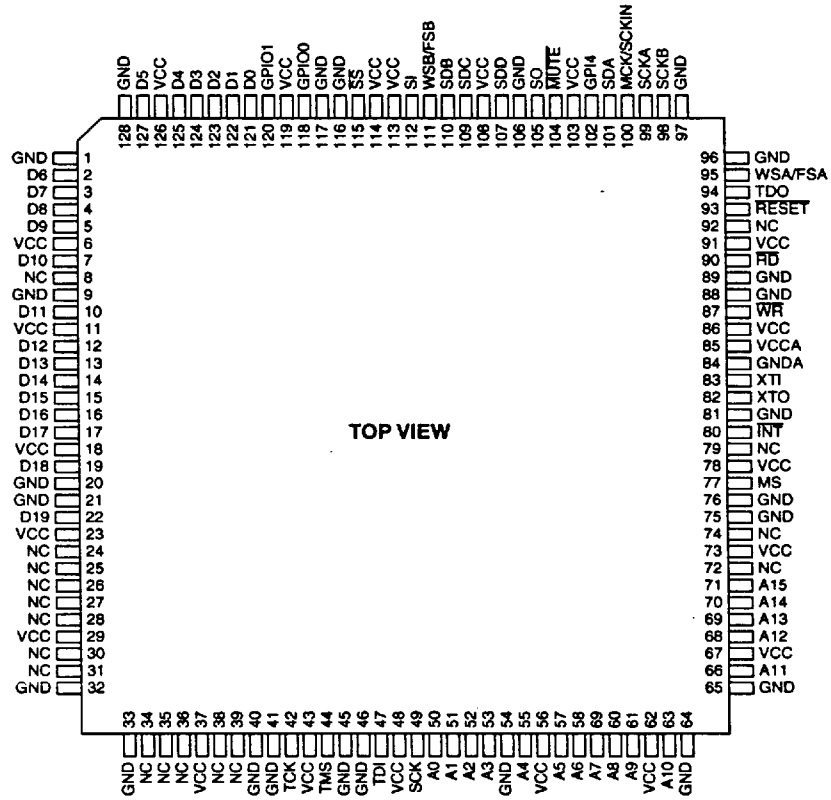
Figure 28. ICE Interface Timing

PINOUT INFORMATION

Table 11: 128-Pin QFP Pin Assignment [1], [2], [3]

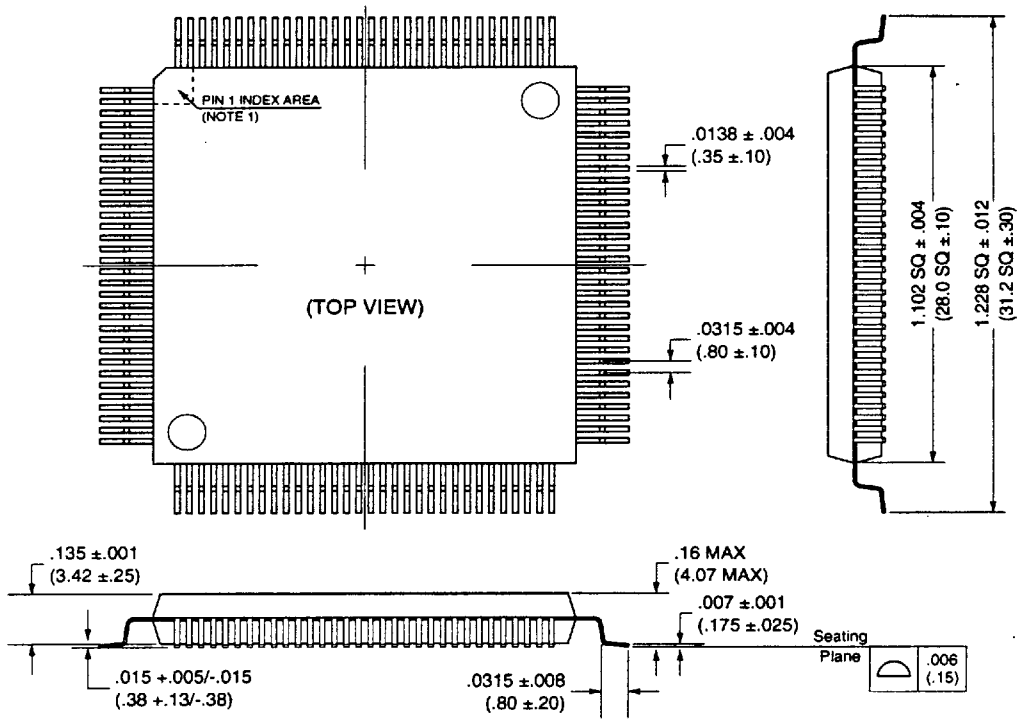
Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type	Pin No	Pin Name	Type
1	GND	P	27	NC	–	53	A3	O/T	79	NC	–	105	SO	P
2	D6	I/O/T	28	NC	–	54	GND	P	80	$\overline{INT}$	I	106	GND	P
3	D7	I/O/T	29	VCC	P	55	A4	O/T	81	GND	P	107	SDD	O
4	D8	I/O/T	30	NC	–	56	VCC	P	82	XTO	O	108	VCC	P
5	D9	I/O/T	31	NC	–	57	A5	O/T	83	XTI	I	109	$\overline{SDC}$	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	I/O
8	NC	–	34	NC	–	60	A8	O/T	86	VCC	P	112	SI	I
9	GND	P	35	NC	–	61	A9	O/T	87	$\overline{WR}$	O	113	VCC	P
10	D11	I/O/T	36	NC	–	62	VCC	P	88	GND	P	114	VCC	P
11	VCC	P	37	VCC	P	63	A10	O/T	89	GND	P	115	$\overline{SS}$	I
12	D12	I/O/T	38	NC	–	64	GND	P	90	$\overline{RD}$	O	116	GND	P
13	D13	I/O/T	39	NC	–	65	GND	P	91	VCC	P	117	GND	P
14	D14	I/O/T	40	GND	P	66	A11	O/T	92	NC	–	118	GPIO0	I/O
15	D15	I/O/T	41	GND	P	67	VCC	P	93	$\overline{RESET}$	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/O	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	NC	–	98	SCKB	I/O	124	D3	I/O/T
21	GND	P	47	TDI	I	73	VCC	P	99	SCKA	I/O	125	D4	I/O/T
22	D19	I/O/T	48	VCC	P	74	NC	–	100	MCK/SCKIN	I/0	126	VCC	P
23	VCC	P	49	SCK	I	75	GND	P	101	SDA	I	127	D5	I/O/T
24	NC	–	50	A0	O/T	76	GND	P	102	GPI4	I	128	GND	P
25	NC	–	51	A1	O/T	77	MS	O/T	103	VCC	P			
26	NC	–	52	A2	O/T	78	VCC	P	104	$\overline{MUTE}$	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).



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

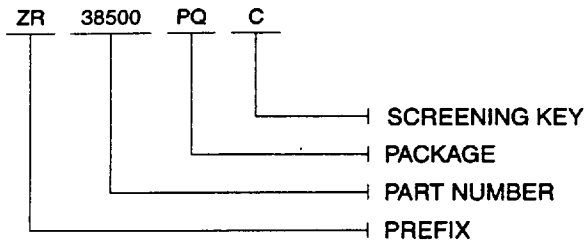
PACKAGE INFORMATION



NOTES: 1. Pin 1 index may be corner 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 29. ZR38500 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
2041 Mission College Blvd.
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: 972-4-8551-551
FAX: 972-4-8551-550