Table of Contents

ISD-T360SB

VoiceDSP™ Digital Speech Processor with Master/Slave, Full-Duplex Speakerphone, Multiple Flash and ARAM/DRAM Support

Preliminary Information

The VoiceDSP™ product family combines multiple digital signal processing functions on a single chip for cost-effective solutions in telephony, automotive and consumer applications.

The VoiceDSP processor offers necessary features to modern telephony products, such as: high-quality, speech record and playback, electrical and acoustic echo cancellation for full-duplex hands-free speakerphone operation.

The ISD-T360SB VoiceDSP can be used in various applications:

- Digital telephony with add-on speech processing: Digital Telephone Answering Machines(DTADs); and full-duplex, hands-free speakerphone operation for ISDN, DECT, Digital Spread Spectrum, and analog cordless applications; CT0/1 Base stations.
- An add-on chip for corded telephones featuring DTAD functions and/or full-duplex, hands-free speakerphone operation.
- Stand-alone digital answering machines with full-duplex, hands-free speakerphone operation.
- Voice memo recording
- Automotive applications employing full-duplex speakerphone operations for hands-free, in-car communications, and for car status and information announcements.

Based on ISD's unique concept which combines 16-bit DSP (Digital Speech Processor) and 16-bit RISC core technology, the ISD-T360SB is a high-performing chip solution for various applications. To facilitate incorporating the VoiceDSP processor, it features system support functions such as an interrupt control unit, codec interface (master, slave), Microwire interface to the system microcontroller, as well as a memory handler for Flash and DRAM memory devices. Design of high-end, price optimized systems are possible with ISD's VoiceDSP flexible system interfaces (codec, microcontroller, and memory management support).

The ISD-T360SB processor operates as a peripheral controlled by the system microcontroller via an enhanced, serial Microwire interface. The system microcontroller typically controls the analog circuits, buttons and display, as well as activates functions through commands. The VoiceDSP executes these commands and returns status information to the Microcontroller.

The VoiceDSP software resides in the on-chip ROM. It includes DSP-based algorithms, system support functions, and a software interface to hardware peripherals.

FEATURES AT A GLANCE

DTAD MANAGEMENT

- Highest quality speech recording in PCM format for music on hold or other OGM (Out Going Message) recording and IVS
- Selectable high-quality speech compression rate of 5.3 Kbit/s, 9.9 Kbits or 16.8 Kbit/s, plus silence compression with each rate
- Up to 16 minutes recording on a 4-Mbit Flash
- High-quality music compression for music on hold (16.8 Kbits)
- Programmable message tag for message categorization, e.g., Mailboxes, InComing Messages (ICM), OutGoing Messages (OGM)
- Message management
- Skip forward or backward during message playback
- Variable speed playback
- Real-time clock: Day of Week, Hours, Minutes
- Multi-lingual speech synthesis using International Vocabulary Support (IVS)
- Vocabularies available in: English, Japanese, Mandarin, German, French and Spanish
- Software Automatic Gain Control (AGC)

SPEAKERPHONE

- Digital full-duplex speakerphone
- Acoustic- and line-echo cancellation
- Continuous on-the-fly monitoring of external and internal conditions (acoustic and line) provides high-quality, hands-free, conversation in a changing environment
- Minimum microcontroller control intervention (Launch-and-forget)
- Supports: On, Off, Mute, and Hold functions

CALL AND DEVICE MANAGEMENT

- Digital volume control
- Least cost routing support (LCR)
- Power-down mode
- 3V to 5V selectable power supply
- 4.096 MHz operation
- DTMF generation and detection
- Telephone line functions, including busy and dial tone detection
- Single tone generation
- DTMF detection during message playback

PERIPHERAL CONTROL

Codec

- µ-Law, A-Law, and 16-bit linear codec input support
- Selectable master/slave codec interface
- Supports two in-coming lines in slave mode without speakerphone for DTAD recording
- Supports up to 16 user selectable speech channels in slave mode
- Supports long-frame and short-frame codecs
- Single/double bit clock rate for slave mode
- On-chip codec clock generation

Memory

- Supports up to four 4-Mbit, four 8-Mbit, or four 16-Mbit Flash devices from Toshiba or Samsung
- Supports up to four 16-Mbit ARAM/DRAM memory devices from Toshiba, Samsung, and Samsung-compatible
- The number of messages that can be stored is limited only by memory size
- Direct access to message memory

- Message storage contains all data in a concatenated chain of memory blocks.
- Memory mapping and product floor test included
- Supports external vocabularies, using Flash memory or expansion ROM

Microwire

- MICROWIRE slave interface to an external microcontroller
- Sophisticated command language to optimize system code size

INTERNATIONAL VOCABULARY SUPPORT (IVS)

For manufacturing recorded voice prompt and speech synthesis, the ISD International Vocabulary Support delivers pre-recorded voice prompts in the same high-quality of the user-recorded speech. For complete control over quality and memory management, the IVS features adjustable speech compressions. In addition, several pre-recorded voice prompt sets are available in various languages for further convenience.

Available Languages:

- English
- Japanese
- Mandarin
- German
- French
- Spanish

Develop a new vocabulary by ISD's voice prompt development tool, the ISD-IVS360. This vocabulary development tool supports various languages and including their unique grammar structures. ISD-IVS360, PC-Windows95™-based program, synthesizes recorded .wav files into the ISD-T360SB's various compression rates (including PCM).

ISD's VoiceDSP products store IVS vocabularies on either Flash memory or expansion ROM memories, thus DTAD manufacturers can design a product for multiple countries, featuring various languages.

For more details about IVS, refer to the *IVS User's Guide*.

Figure 1-1: ISD-T360SB Block Diagram—Basic Configuration with Four 4Mb/8Mb/16Mb NAND Flash Devices (Samsung/Toshiba)

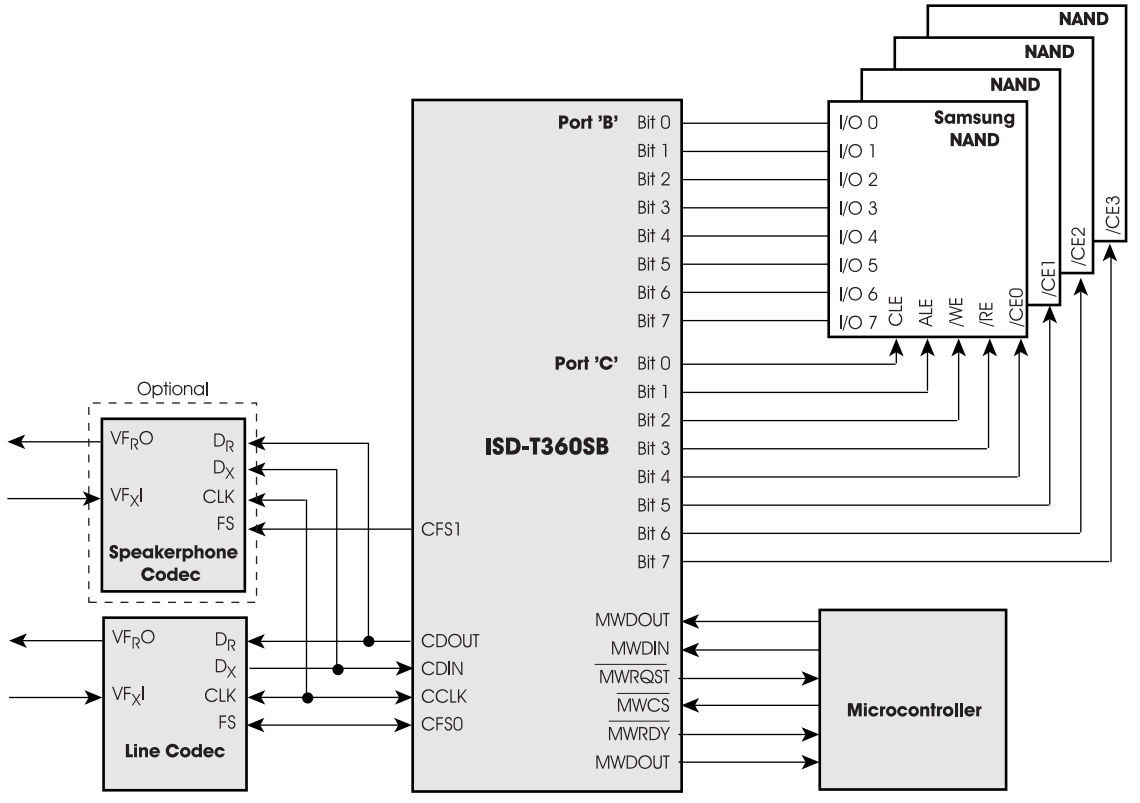


Figure 1-2: ISD-T360SB Block Diagram—Basic Configuration with Four 4Mb Serial Toshiba Flash Devices

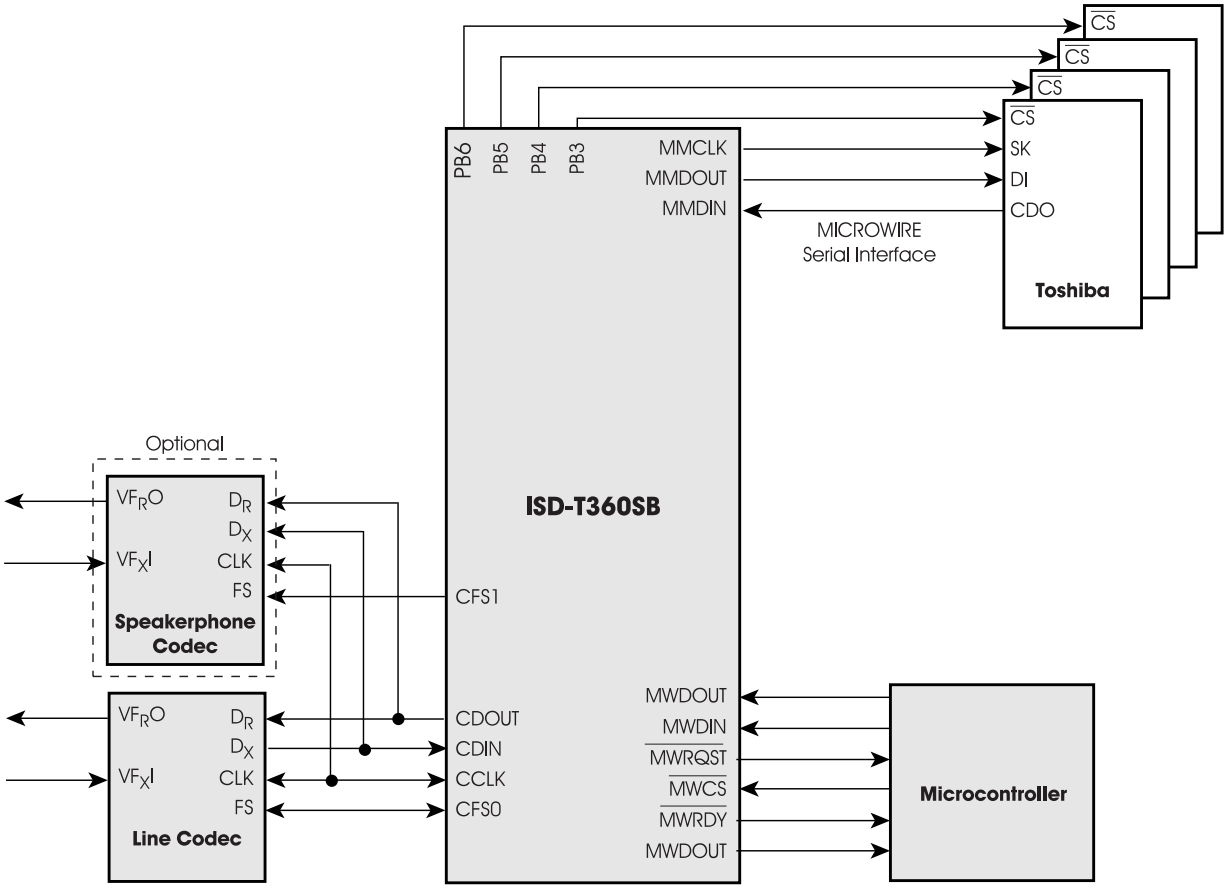
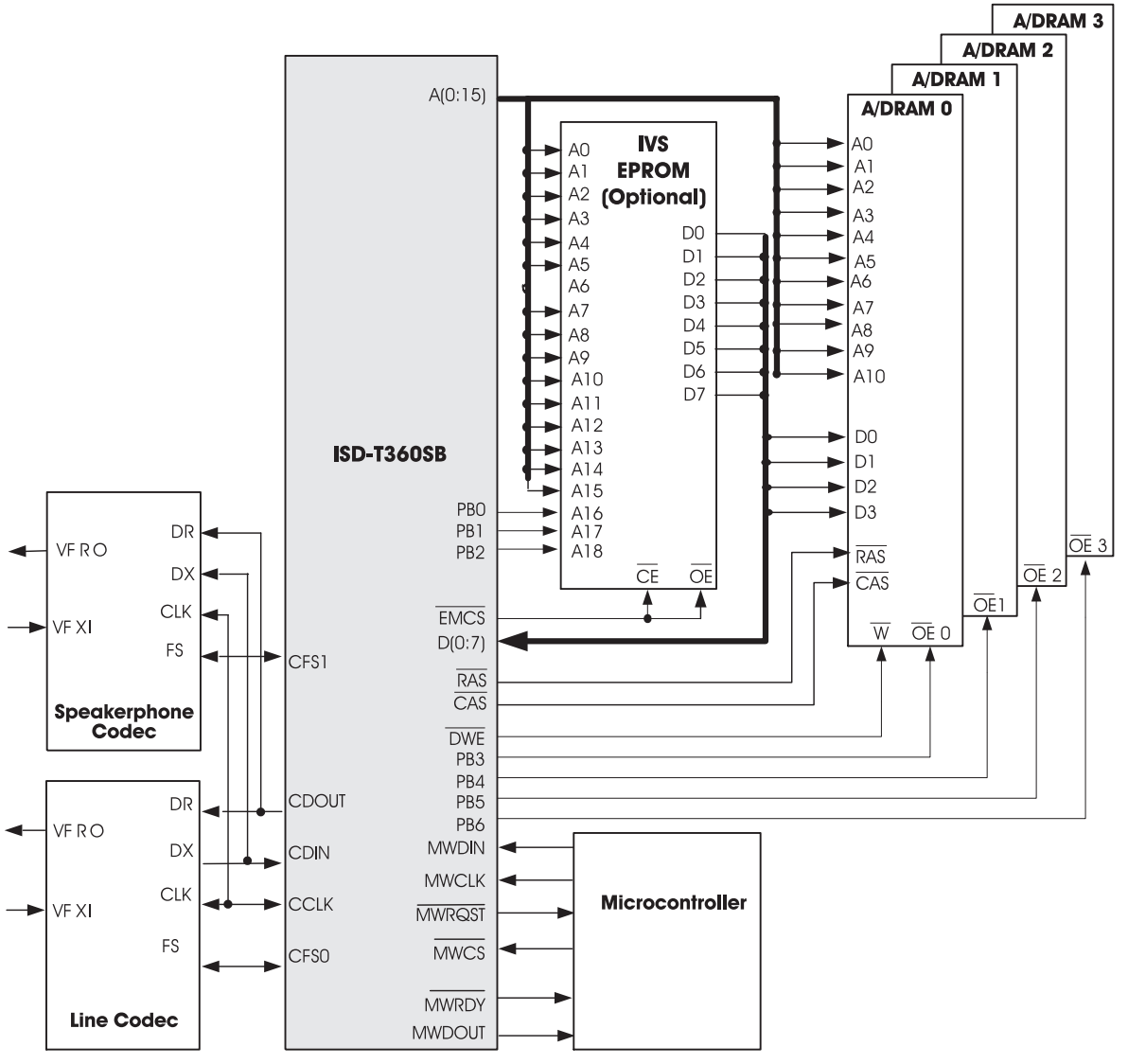


Figure 1-3: ISD-T360SB Block Diagram—Basic Configuration with Four 16Mb ARAM/DRAM Devices (Samsung) and IVS EPROM



Chapter 1—HARDWARE

1.1 PIN ASSIGNMENT

The following sections detail the pins of the ISD-T360SB processor. Slashes separate the names of signals that share the same pin.

1.1.1 PIN-SIGNAL ASSIGNMENT

Table 1-1 shows all the pins, and the signals that use them in different configurations. It also shows the type and direction of each signal.

Figure 1-1: 80-MQFP Package Connection Diagram

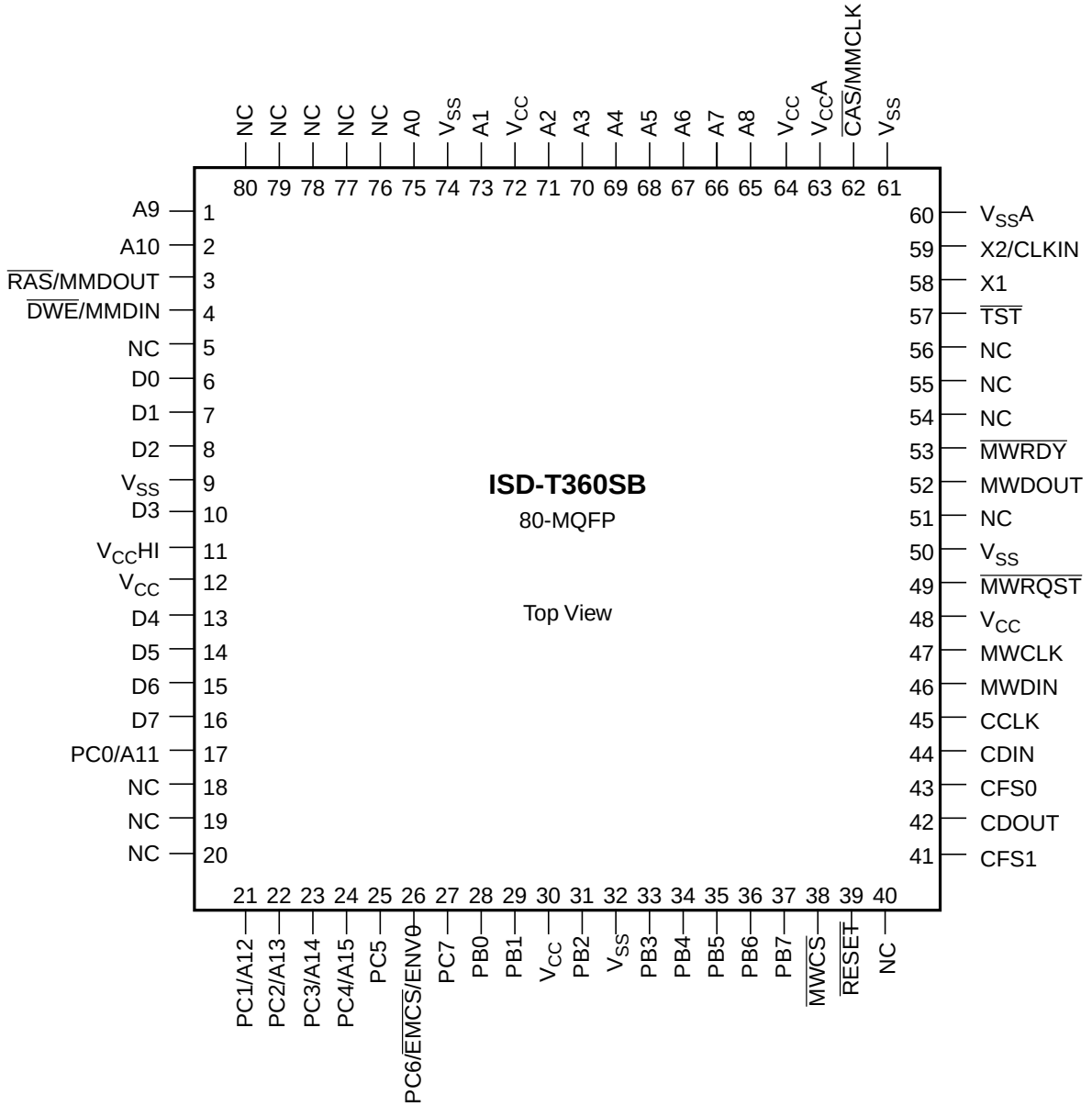


Table 1-1: VoiceDSP Pin Signal Assignment

Pin Name	Signal Name	Type	Description
A(0:15)	A(0:16)	Output	Address bits 0 through 16
$\overline{CAS}$	$\overline{CAS}$	Output	DRAM Column Address Strobe
CCLK	CCLK	I/O	Codec Master/slave Clock
CDIN	CDIN	Input	Data Input from Codec
CDOUT	CDOUT	Output	Data Output to Codec
CFS0	CFS0	I/O	Codec 0 Frame Synchronization
CFS1	CFS1	Output	Codec 1 Frame Synchronization
D(0:7)	D(0:7)	I/O	Data bits 0 through 7
$\overline{DWE}$	$\overline{DWE}$	Output	DRAM Write Enable
$\overline{EMCS}$/ENV0	$\overline{EMCS}$	Output	Expansion Memory Chip Select
$\overline{EMCS}$/ENV0	ENV0	Input	Environment Select
MMCLK	MMCLK	Output	Master MICROWIRE Clock
MMDIN	MMDIN	Input	Master MICROWIRE Data Input
MMDOUT	MMDOUT	Output	Master MICROWIRE DATA Output
MWCLK	MWCLK	Input	MICROWIRE Clock
$\overline{MWCS}$	$\overline{MWCS}$	Input	MICROWIRE Chip Select
MWDIN	MWDIN	Input	MICROWIRE Data Input
MWDOUT	MWDOUT	Output	MICROWIRE DATA Output
$\overline{MWRDY}$	$\overline{MWRDY}$	Output	MICROWIRE Ready
$\overline{MWRQST}$	$\overline{MWRQST}$	Output	MICROWIRE Request Signal
PB(0:7)[3]	PB(0:7)	I/O	Port B, bits 0 through 7
PC(0:7)	PB(0:7)	I/O	Port C, bits 0 through 7
$\overline{RAS}$	$\overline{RAS}$	Output	DRAM Row Address Strobe
$\overline{RESET}$	$\overline{RESET}$	Input	Reset
$\overline{TST}$	$\overline{TST}$	Input	Test pin
V_{CC}	V_{CC}	Power	3.3 V power supply pin
$V_{CC}A$	$V_{CC}A$	Power	3.3 V analog circuitry power supply pin
$V_{CC}HI$	$V_{CC}HI$	Power	5 V power supply pin. Connect to V_{CC} if 3.3 V power supply is used.
V_{SS}	V_{SS}	Power	Ground for on-chip logic and output drivers
$V_{SS}A$	$V_{SS}A$	Power	Ground for on-chip analog circuitry
X1	X1	Oscillator	Crystal Oscillator Interface
X2/CLKIN	X2	Oscillator	Crystal Oscillator Interface

1. *TTL1 output signals provide CMOS levels in the steady state, for small loads.*
2. *Input during reset. CMOS level input.*
3. *Virtual address lines for IVS ROM.*
4. *Chip select lines for Serial Flash devices.*
5. *Schmitt trigger input.*

1.2 DESCRIPTION

This section provides details of the functional characteristics of the VoiceDSP processor. It is divided into the following sections:

- Resetting
- Clocking
- Power-Down Mode
- Power and Grounding
- Memory Interface
- Codec Interface

1.2.1 RESETTING

The $\overline{\text{RESET}}$ pin is used to reset the VoiceDSP processor.

On application of power, $\overline{\text{RESET}}$ must be held low for at least t_{pwr} after V_{CC} is stable. This ensures that all on-chip voltages are completely stable before operation. Whenever $\overline{\text{RESET}}$ is applied, it must also remain active for not less than t_{RST}, see Table 1-10 and Table 1-11. During this period, and for 100 ms after, the $\overline{\text{TST}}$ signal must be high. This can be done with a pull-up resistor on the $\overline{\text{TST}}$ pin.

The value of $\overline{\text{MWRDY}}$ is undefined during the reset period, and for 100 ms after. The microcontroller should either wait before polling the signal for the first time, or the signal should be pulled high during this period.

Upon reset, the ENV0 signal is sampled to determine the operating environment. During reset, the $\overline{\text{EMCS}}$/ENV0 pin is used for the ENV0 input signals. An internal pull-up resistor sets ENV0 to 1.

After reset, the same pin is used for $\overline{\text{EMCS}}$.

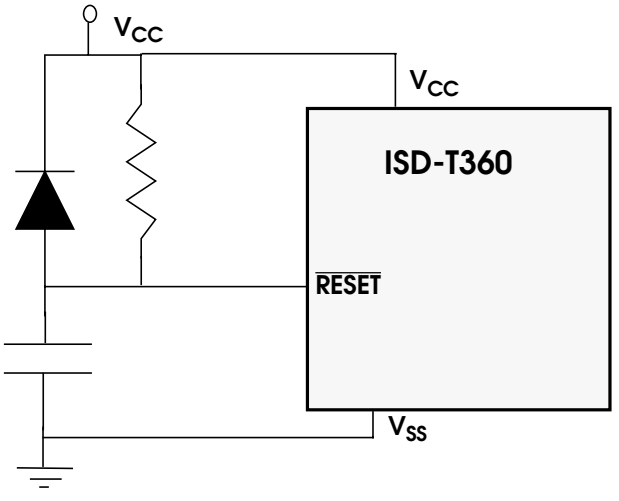
System Load on ENV0

For any load on the ENV0 pin, the voltage should not drop below V_{ENVh} (see Table 1-10 and Table 1-11).

If the load on the ENV0 pin causes the current to exceed 10 µA, use an external pull-up resistor to keep the pin at 1.

Figure 1-2 shows a recommended circuit for generating a reset signal when the power is turned on.

Figure 1-2: Recommended Power-On Reset Circuit



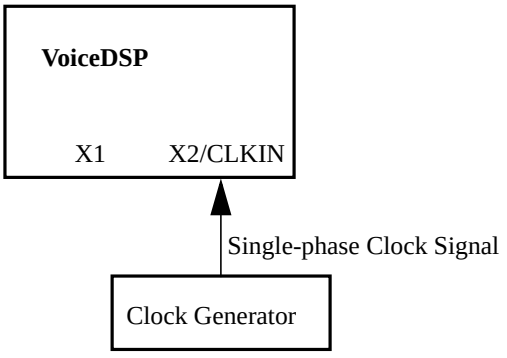
1.2.2 CLOCKING

The VoiceDSP processor provides an internal oscillator that interacts with an external clock source through the X1 and X2/CLKIN pins. Either an external single-phase clock signal, or a crystal oscillator, may be used as the clock source.

External Single-Phase Clock Signal

If an external single-phase clock source is used, it should be connected to the CLKIN signal as shown in Figure 1-3, and should conform to the voltage-level requirements for CLKIN stated in "ELECTRICAL CHARACTERISTICS" on page 1-18.

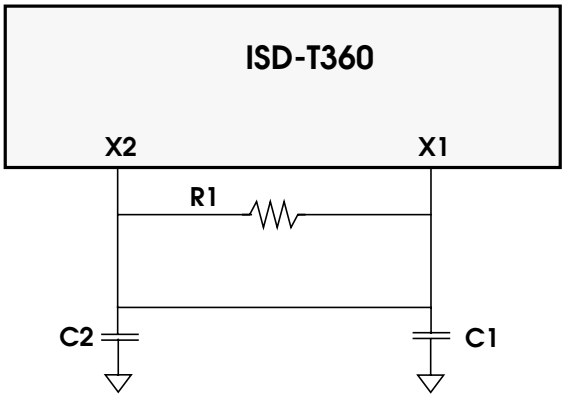
Figure 1-3: External Clock Source



Crystal Oscillator

A crystal oscillator is connected to the on-chip oscillator circuit via the X1 and X2 signals, as shown in Figure 1-4.

Figure 1-4: Connections for an External Crystal Oscillator



Keep stray capacitance and inductance, in the oscillator circuit, as low as possible. The crystal resonator, and the external components, should be as close to the X1 and X2/CLKIN pins as possible, to keep the trace lengths in the printed circuit to an absolute minimum.

You can use crystal oscillators with maximum load capacitance of 20 pF, although the oscillation frequency may differ from the crystal's specified value.

Table 1-2 lists the components in the crystal oscillator circuit

Table 1-2: Components of Crystal Oscillator Circuit

Component	Values	Tolerance
Crystal Resonator	4.096 MHz	
Resistor R1	10 MΩ	5%
Capacitors C1, C2	33 pF	20%

1.2.3 POWER-DOWN MODE

Power-down mode is useful during a power failure or in a power-saving model, when the power source for the processor is a backup battery or in battery-powered devices, while the processor is in idle mode.

In power-down mode, the clock frequency of the VoiceDSP processor is reduced and some of the processor modules are deactivated. As a result, the ISD-T360SB consumes considerably less power than in normal-power mode. Although the VoiceDSP processor does not perform all its usual functions in power-down mode, it does retain stored messages and maintain the date and time.

NOTE In power-down mode all the chip select signals, CS0 to CS3, are set to 1. To guarantee that there is no current flow from these signals to the Flash devices, the power supply to these devices must not be disconnected.

The ISD-T360SB stores messages and all memory management information in Flash or ARAM/DRAM memory. When Flash memory is used for memory management, power does not need to be maintained to the processor to preserve stored messages. When ARAM/DRAM memory is used for message management, preserving stored messages requires a battery back up dur-

ing a power failure. If the power failure extends the life of the battery, the microcontroller should perform the initialization sequence (as described on page 2-4), and use the SETD command to set the date and time.

To keep power consumption low during power-down mode, the $\overline{RESET}$, $\overline{MWCS}$, MWCLK and MWDIN signals should be held above V_{CC} – 0.5 V or below V_{SS} + 0.5 V.

1.2.4 POWER AND GROUNDING

Power Pin Connections

The ISD-T360 can operate over two supply voltage ranges 3.3 V ±10% and 5 V ±10%. The five power supply pins (V_{CC}, V_{SS}, $V_{CC}A$, $V_{SS}A$ and $V_{CC}HI$) must be connected as shown in Figure 1-5 when operating in a 3.3 V environment, and as shown in Figure 1-6 when operating in a 5 V environment. Failure to correctly connect the pins may result in damage to the device.

The Capacitor and Resistor values are given in Table 1-3.

Table 1-3: Components of Supply Circuit

Component	Values	Tolerance
Resistor R1, R2	10 Ω	5%
Capacitors(1) C1, C2 Capacitors C3, C4, C5, C6, C7	1.0 µF Tantalum 0.1 µF Ceramic	20%

1. All capacitors represent two parallel capacitors at the values 1.0 µF and 0.1 µF.

Figure 1-5: 3.3 V Power Connection Diagram

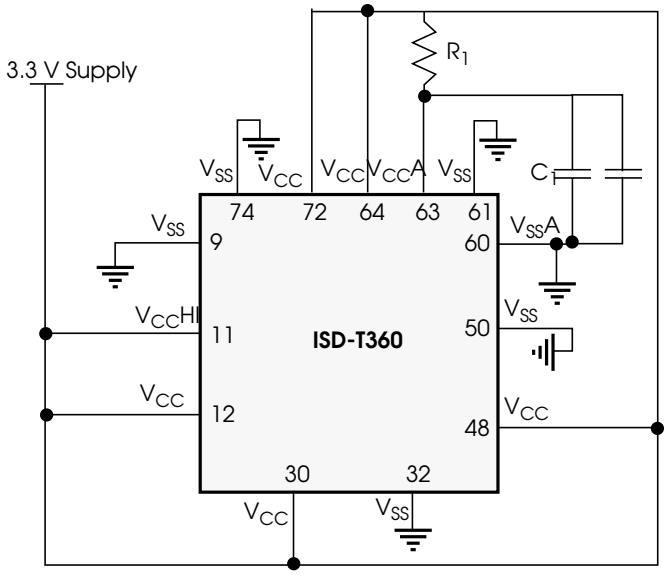
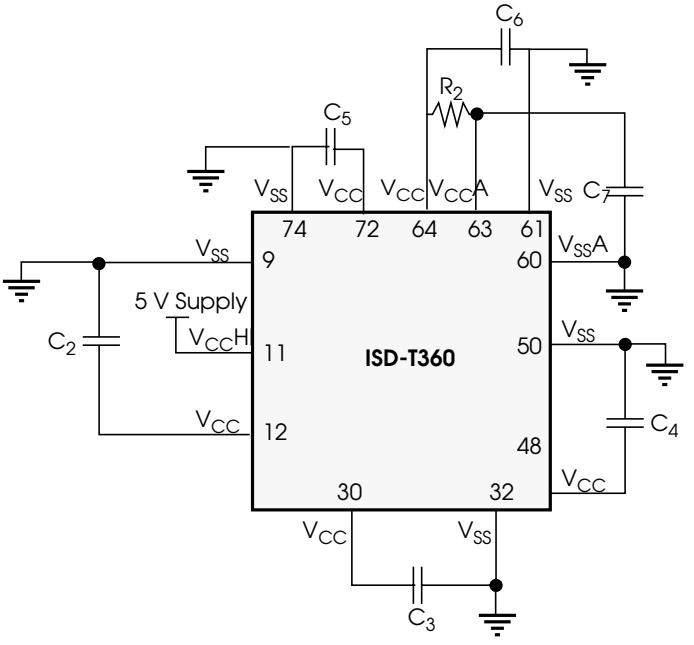


Figure 1-6: 5 V Power Connection Diagram



For optimal noise immunity, the power and ground pins should be connected to V_{CC} and the ground planes, respectively, on the printed circuit board. If V_{CC} and the ground planes are not used, single conductors should be run directly from each V_{CC} pin to a power point, and from each GND pin to a ground point. Avoid daisy-chained connections. The VoiceDSP does not perform its usual functions in power-down mode but it still preserves stored messages, maintains the time of day and generates ARAM/DRAM refresh cycles.

When you build a prototype, using wire-wrap or other methods, solder the capacitors directly to the power pins of the VoiceDSP processor socket, or as close as possible, with very short leads.

1.2.5 MEMORY INTERFACE

Flash Support

The ISD-T360SB VoiceDSP supports Flash devices for storing recorded data, thus, power can be disconnected to the ISD-T360SB without losing data. The ISD-T360SB supports serial and semi-parallel Flash device interfaces, such as TC58V16BFT, TC5816BFT, TC58A040F, KM29N040T, KM2928000T/IT, and KM29216000AT/AIT. The ISD-T360SB may be connected to up to four Flash devices, resulting with maximum recording storage of 16-Mbits x 4 = 64 Mbits (up to 4 hours of recording time).

The following flash devices are supported:

Table 1-4: Supported Flash Devices

Manufacturer	Memory Device Name	Characteristics	Operating Voltage	Memory Size	Conversion Type
Toshiba	TC58V16BFT	2Mx8	3.3 V	16-Mbit LV	µ-Law
Toshiba	TC5816BFT	2Mx8	5 V	16-Mbit	A-Law
Toshiba	TC58A040F	Serial	5V	4-Mbit	µ-Law, A-Law
Samsung	KM29N040T	512Kx8	5 V	4-Mbit	µ-Law, A-Law, LV
Samsung	KM29W8000T/IT	1Mx8	5 V	8-Mbit	µLaw
Samsung	KM29W16000AT /AIT	2Mx8	5 V	16-Mbit	A-Law

Internal Memory Organization

The Flash devices detailed in Table 1-4 divide internally into basic 4-Kbyte block units. The ISD-T360SB uses one block on each device for memory management, leaving the rest of the blocks available for recording. Using at least one block for a single recorded message yields a maximum of NUM_OF_BLOCKS_IN_MEM – 1 (see Table 2-10 for parameter definition) messages per device.

Upon initialization the ISD-T360SB activates a sifting algorithm to detect defected blocks. Defected blocks are defined as blocks with over 10 bad nibbles (a nibble consists of 4 bits). The defected blocks are marked as UNUSED and excluded from the list of available blocks for recording.

Toshiba Serial Flash

The VoiceDSP processor supports up to four TC58A040F, 4Mbit, serial interface, Flash memory devices for storing messages. The TC58A040F is organized as an array of 128 blocks, with a dedicated, read-only, bad block map programmed by the manufacturer and located in the last block. This map is used by the ISD-T360SB to define the available blocks for recording. The ISD-T360SB uses the VoiceDSP master MICROWIRE interface to communicate, serially, with the Flash devices, while selecting the current Flash device using PB3-PB6. Connecting less than four Flash devices require connecting the Flash devices sequentially, starting from PB3 up to PB6 (see Figure 1-7). Refer to Figure 1-34for the Master MICROWIRE timing diagram.

Figure 1-7: Memory Interface with Four Toshiba 4Mbit Serial Flash Devices and Optional Voice IVS EPROM

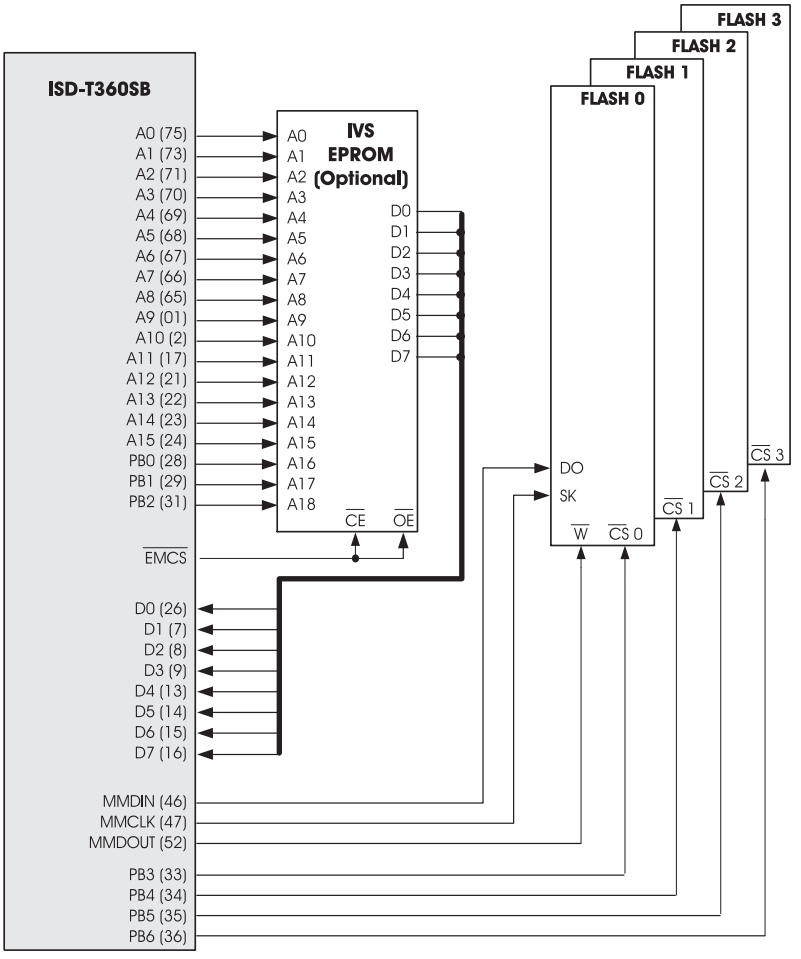
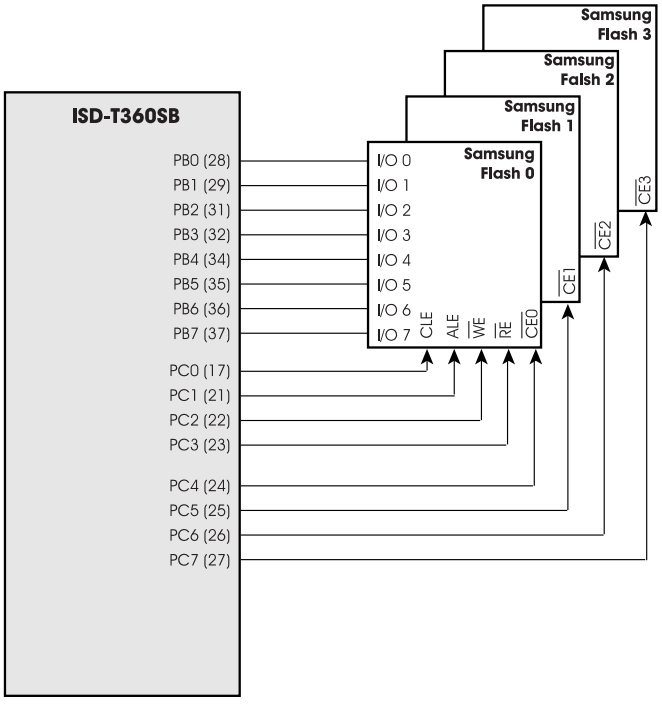


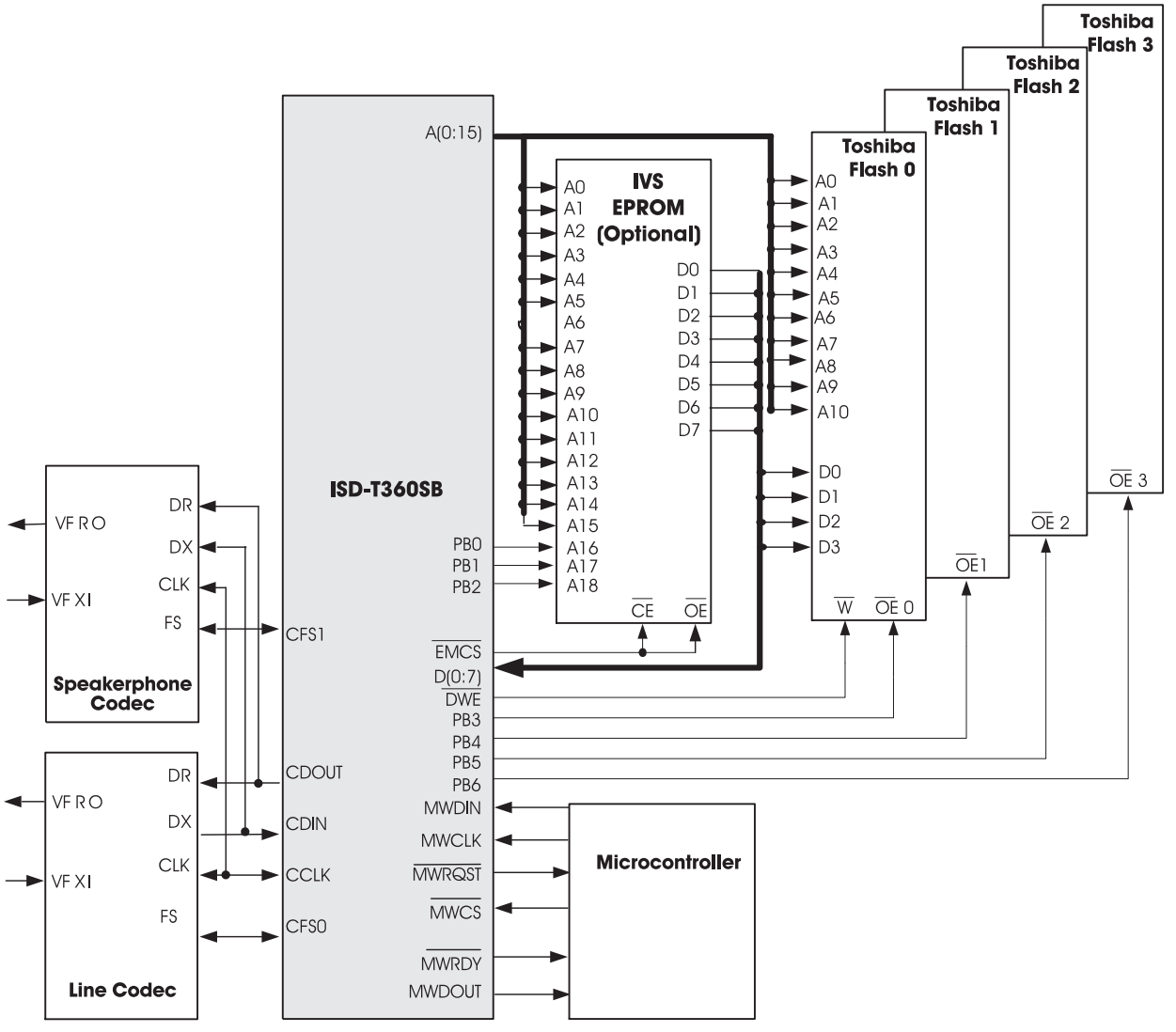
Figure 1-8: Memory Interface with Four 4MB/16 Mbit, NAND Flash Devices (Samsung, Toshiba)



NAND Flash (Samsung, Toshiba)

The VoiceDSP processor supports up to four, semi-parallel interface, Flash memory devices for storing messages. Flash device with semi-parallel interface uses a single 8bit I/O port to set the address and access the data. The ISD-T360SB supports three types of Flash volumes (4Mbit, 8Mbit and 16Mbit as listed in Table 1-4) while all the connected Flash devices must be of the same type. Ports B and C are used to connect ISD-T360SB to the Flash devices using port B for address and data transfer and port C for communication control and chip select. Connecting less than four Flash devices require connecting the Flash devices sequentially, starting from PC4 up to PC7 (see Figure 1-8). The ISD-T360SB scans the Flash devices upon initialization, sifting out the bad blocks, and marking them in a special map, located in the last block of each device.

Figure 1-9: Memory Interface with Four Toshiba 4Mbit Serial Flash Devices and Optional IVS EPROM



Flash Endurance

A Flash memory may be erased a limited number of times. To maximize the Flash use, the memory manager utilizes the Flash's blocks evenly (i.e., each block is erased more or less the same number of times), to ensure that all blocks have the same lifetime. Refer to the respective Flash memory device data sheets for specific endurance specifications.

A VoiceDSP processor message uses at least one block.

The maximum recording time depends on four factors:

1. The basic compression rate (5.3 Kbit/s, 9.9Kbit/s, or 16.8Kbit/s).
2. The amount of silence in the recorded speech.
3. The number of bad blocks.

The number of recorded messages. (The basic memory allocation unit for a message is a 4-Kbyte block, which means that half a block on average is wasted per recorded message).

Table 1-5: Recording Time with 15% Silence Compression

Memory Size	Compression Rate	Total Recording Time
4 Mbit	5.3 Kbit/s	14.9 Minutes
4 Mbit	9.9 Kbit/s	8.1 Minutes
4 Mbit	16.8 Kbit/s	4.8 Minutes
8 Mbit	5.3 Kbit/s	29.8 Minutes
8 Mbit	9.9 Kbit/s	16.2 Minutes
8 Mbit	16.8 Kbit/s	9.6 Minutes
16 Mbit	5.3 Kbit/s	59.6 Minutes
16 Mbit	9.9 Kbit/s	32.4 Minutes
16 Mbit	16.8 Kbit/s	19.2 Minutes
32 Mbit	5.3 Kbit/s	119.2 Minutes
32 Mbit	9.9 Kbit/s	64.8 Minutes
32 Mbit	16.8 Kbit/s	38.4 Minutes

ARAM/DRAM Support

The VoiceDSP processor supports up to four, 16-Mbit, ARAM/DRAM devices for storing messages. The ISD-T360 connects to the ARAM/DRAM device using address buses A0–A11 and data buses D0–D3. This connection allows access to 2^{22} nibbles (16-Mbit) on each device. The ISD-T360SB selects the current ARAM/DRAM device using PB3–PB6 as described in Figure 1-10. Using less than four ARAM/DRAM devices requires connecting the devices sequentially, starting from PB3 up to PC6. $\overline{RAS}$ and $\overline{CAS}$ are connected in parallel to all the ARAM/DRAM devices and are used to refresh the memory. The difference between ARAM and DRAM resides with the amount of bad cells on the device and the device performance. While DRAM has no bad cells, ARAM contains certain level of impurity and thus requires testing and mapping of the bad blocks upon the initialization of the ISD-T360SB. Although there are no real blocks on the ARAM device, the ISD-T360SB emulates virtual "blocks" on the ARAM device (as if it was a Flash device), tests these "blocks" and marks them in a special map on the last "block" on each device. This test is required only when using ARAM devices (as opposed to DRAM devices that require no testing due to lack of bad blocks). The virtual division to blocks simplifies the use of ARAM/DRAM devices and allows the use of the same set of commands for Flash and ARAM/DRAM.

Another major difference between ARAM/DRAM and Flash devices is that the internal mapping in the ARAM is lost upon power off. Thus, the initialization process needs to take place after each power reset. The mapping is not lost when entering and exiting the Power-Down Mode.

Refer to Figure 1-24 through Figure 1-28 for Timing Diagrams of ARAM/DRAM Read, Write, Refresh in Normal Mode and Refresh in Power-Down Mode Cycles.

ROM Interface

IVS vocabularies can be stored in either Flash memory and/or ROM. The VoiceDSP processor supports IVS ROM devices through an Expansion Memory mechanism. Up to 64 Kbytes (64K x 8) of Expansion Memory are directly supported. Nevertheless, the processor uses bits of the on-chip port (PB) to further extend the 64 Kbytes address space up to 0.5 Mbytes address space.

ROM is connected to the VoiceDSP processor using the data bus, D(0:7), the address bus, A(0:15), the extended address signals, EA(16:18), and Expansion Memory Chip Select, $\overline{EMCS}$, controls. The number of extended address pins to use may vary, depending on the size and configuration of the ROM. ISD-T360SB configured with semi-parallel Flash memory can not support extension ROM.

Figure 1-10: Memory Interface with Four 16-Mbit ARAM/DRAM Devices (Samsung, Toshiba) and Optional IVS EPROM

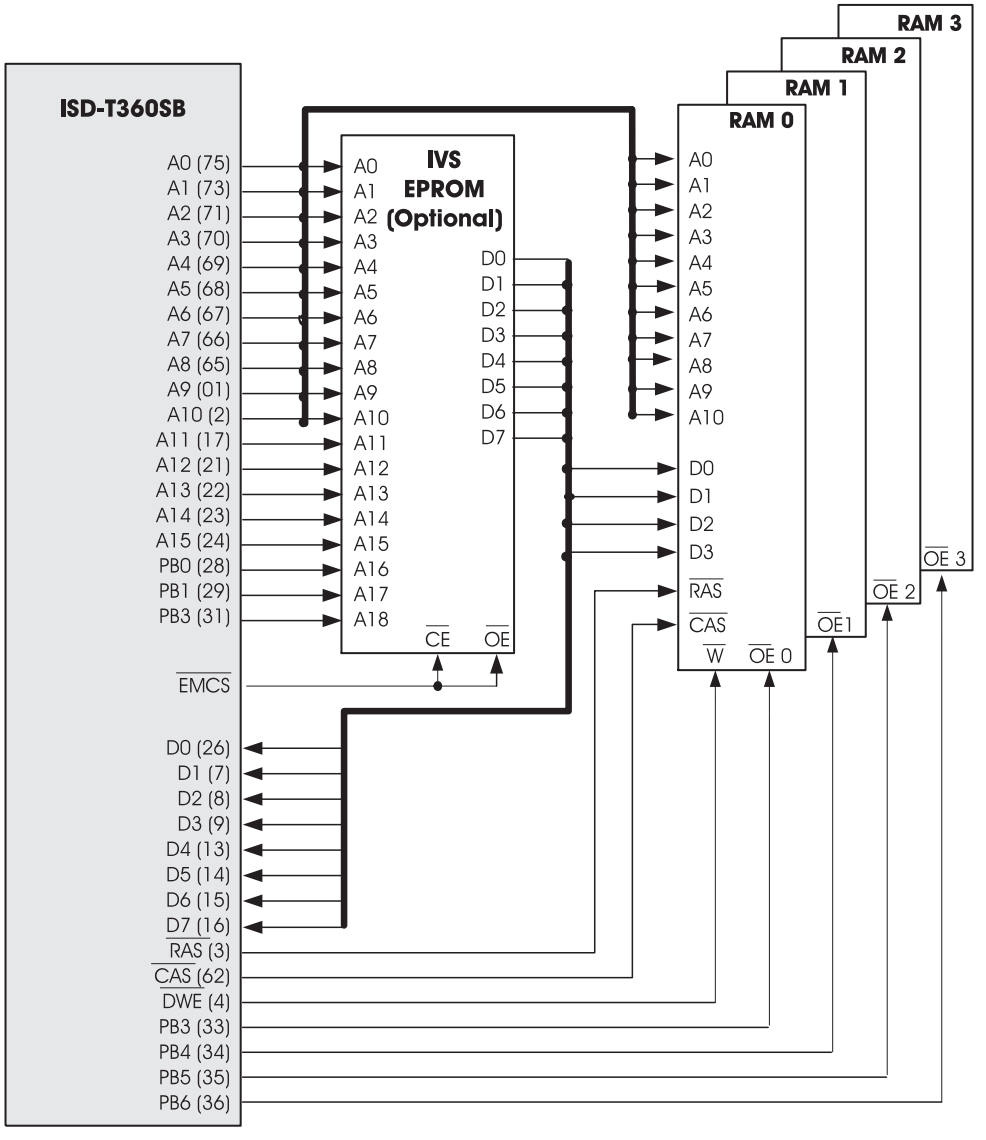


Table 1-6: Supported DRAM Devices

Manufacturer	Memory Device Name	Characteristics	Operating Voltage	Memory Size
Samsung	KM44C4004CS-6	EDO 4Mx4	5 V	16-Mbit EDO LV
Samsung	KM44V4004CS-6	EDO 4Mx4	3.3 V	16-Mbit EDO
Toshiba	TP	EDO 4Mx4	5V, 3.3V	16-Mbit EDO, LV

1.2.6 THE CODEC INTERFACE

The ISD-T360 provides an on chip interface for analog and digital telephony, supporting master and slave codec interface modes. In master mode, the ISD-T360 controls the operation of the codec for use in analog telephony. In the slave mode, the ISD-T360 codec interface is controlled by an external source. This mode is used in digital telephony (i.e., ISDN or DECT lines). The slave mode is implemented with respect to IOM-2™/ CGI specifications.

See Table 1-7 for codec options for the ISD-T360SB (ISD supports compatible codecs in addition to those listed below).

The codec interface supports the following features:

- Master Mode or Slave Mode.
- 8- or 16-bit Channel Width.
- Long (Variable) or Short (Fixed) Frame Protocol.
- Single or Double Bit (Slave Mode Only) Clock Rate.
- Single or Dual Channel Codecs
- One or Two Codecs
- Multiple Clock And Sample Rates.
- One or Two Frame Sync Signals

This codec interface uses five signals: CDIN, CDOUT, CCLK, CFS0, and CFS1. The CDIN, CDOUT, CCLK, and CFS0 pins are connected to the first codec. The second codec is connected to CDIN, CDOUT, CCLK, and CFS1 pins. Data is transferred to the codec through the CDOUT output pin. Data is read from the codec through the CDIN input pin. The CCLK and CFS0 pins are output in Master Mode and input in Slave Mode. The CFS1 is an output pin.

Short Frame Protocol

When the short frame protocol is configured, eight or sixteen data bits are exchanged with each codec in each frame (i.e., the CFS0 cycle). Data transfer begins when CFS0 is set to 1 for one CCLK cycle. The data is then transmitted, bit by bit, via the CDOUT pin. Concurrently, the received data is shifted in through the CDIN pin. Data is shifted one bit per CCLK cycle. After the last bit has been shifted, CFS1 is set to 1 for one CCLK cycle. Then, the data from the second codec is shifted out via CDOUT, concurrently with the inward shift of the data received via CDIN.

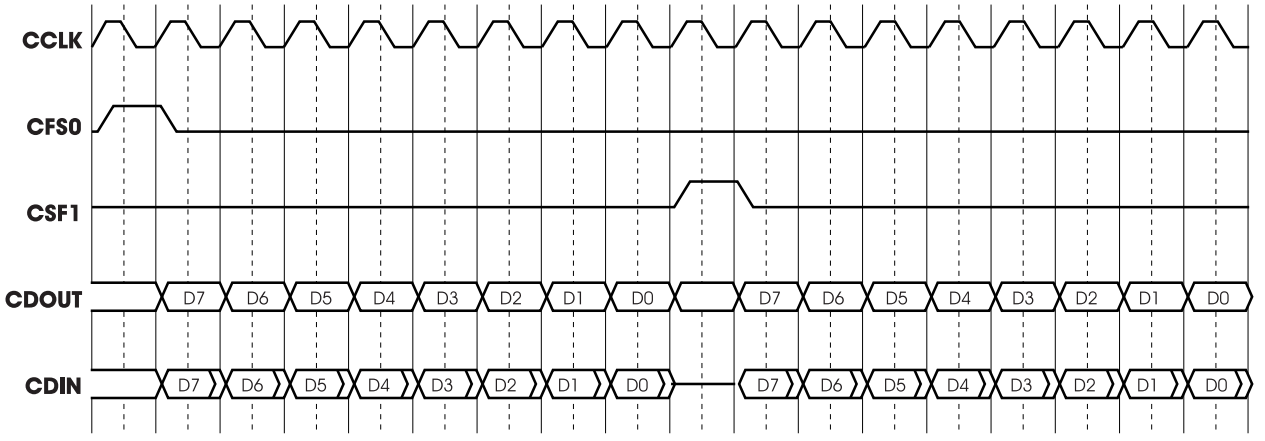
Long Frame Protocol

When long frame protocol is configured, eight or sixteen data bits are exchanged with each codec, as for the short frame protocol. However, for the long frame protocol, data transfer starts by setting CFS0 to 1 for eight or sixteen CCLK cycles. Short or long frame protocol is available in both Master and Slave modes. Figure 1-11 illustrates an example of short frame protocol with an 8-bit channel width.

Table 1-7: Supported Codec Devices

Manufacturer	Codec Device Name	Characteristics	Operating Voltage	Conversion Type
National Semiconductor	TP3054	Single codec	5 V	μ-Law
National Semiconductor	TP 3057	Single codec	5 V	A-Law
Oki	MSM7533V	Dual codec	5 V	μ-Law, A-Law
Oki	MSM 7704	Dual codec	3.3 V	μ-Law, A-Law, LV
Macronix	MX93002FC	Dual codec	5 V	μ-Law
Lucent	T7502	Dual codec	5 V	A-Law
Lucent	T7503	Dual codec	5 V	μ-Law

Figure 1-11: Codec Protocol-Short Frame—8-Bit Channel Width



Channel Width

The Codec interface supports both 8-bit and 16-bit channel width in Master and Slave Modes.

Slave Mode

The VoiceDSP supports digital telephony applications including DECT and ISDN by providing a Slave Mode of operation. In Slave Mode operation, the CCLK signal is input to the ISD-T360 and controls the frequency of the codec interface operation. The CCLK may take on any frequency between 500 KHz and 4 MHz. Both long and short frame protocol are supported with only the CFS1 output signal width affected. The CFS0 input signal must be a minimum of one CCLK cycle.

In slave mode, a double clock bit rate feature is available as well. When the codec interface is configured to double clock bit rate, the CCLK input signal is divided internally by two and the resulting clock used to control the frequency of the codec of the codec interface operation.

This interface supports ISDN protocol with one bit clock rate or double bit clock rate. The exact format is selected with the CFG command. The slave codec interface uses four signals: CDIN, CDOUT, CCLK, and CFS0. The CDIN, CCLK, and CFS0 input pins and the CDOUT output pins are connected to the ISDN/DECT agent. Data is transferred to the VoiceDSP through the CDIN pin and read out through the CDOUT pin. The CFS0 pin is used to define the start of each frame (see below) the source of that signal is at the master side. The CCLK is used for bit timing of CDIN and CDOUT. The rate of the CCLK is configured via the CFG command and can be twice of the data rate or at the data rate. The source of that signal is at the master side.

Table 1-8: Typical Codec Applications

Application	Codec Type	No. of Channels	Master/ Slave	Channel Width (No. Bits)	Long/ Short Frame Protocol	Bit Rate	CCLK Freq. (MHz)	Sample Rate (Hz)	No. of Frame Syncs
Analog µ-Law	single	1	Master	8	short or long	1	2.048	8000	1
ISDN—8 bit digital—A-Law	dual	2	Slave	8	short	1 or 2	2.048	8000	1
Linear	single	1	Master	16	short	1	2.048	8000	1
IOM-2/GCI	single or dual	1–2	Slave	8	short	1 or 2	1.536	8000	1
266 Compatibility	single or dual	1 or 2	Master	8	long or short	1	2.048	8000	1 or 2

Figure 1-12: Codec Interface with One Single Codec, NSC TP3054, for Single Line Operation

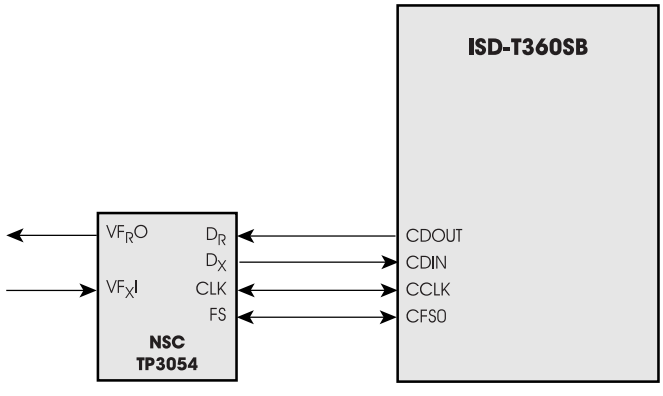


Figure 1-13: Codec Interface Diagram with Two, Single Codecs, NSC TP3054, and NSC TP3057, for Speakerphone Operation

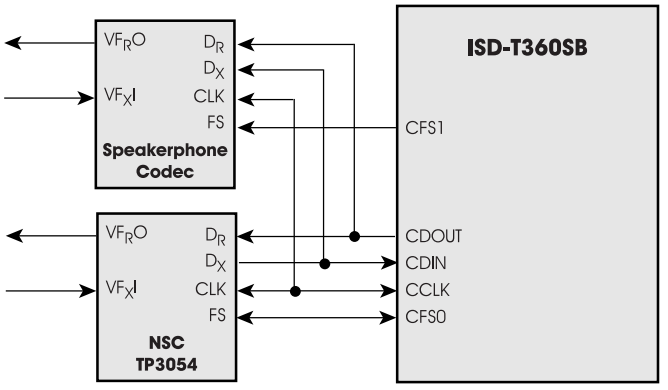


Figure 1-14: Codec Interface for Dual Line or Single Line and Speakerphone Operation with OKI Dual Codec

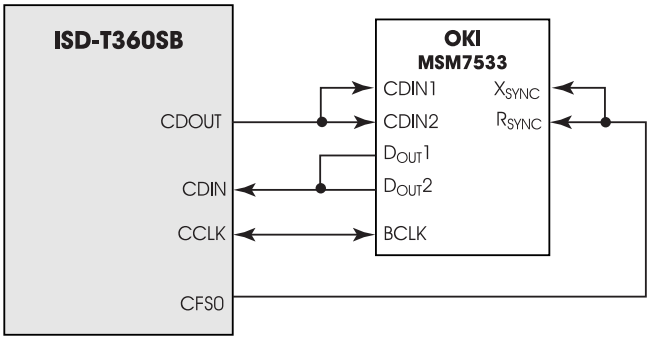


Figure 1-15: Codec Interface for Dual Line or Single Line and Speakerphone with Lucent Dual Codec

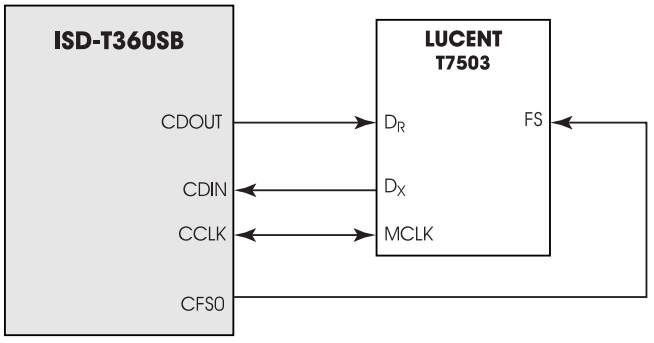
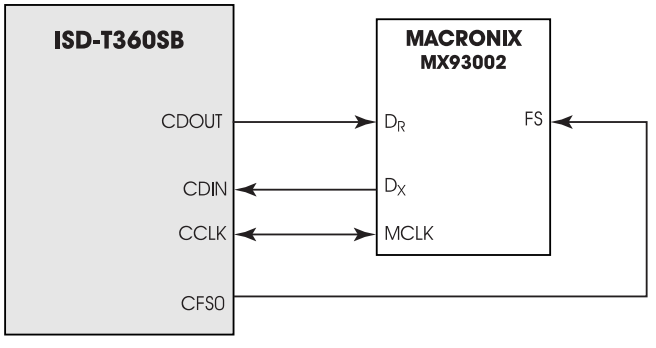


Figure 1-16: Codec Interface for Dual Line or Single Line and Speakerphone Operation with Macronix Dual Codec



1.3 SPECIFICATIONS

1.3.1 ABSOLUTE MAXIMUM RATINGS

Storage temperature	–65˚C to +150˚C
Temperature under bias	0˚C to +70˚C
All input and output voltages with respect to GND	–0.5 V to +6.5 V

NOTE Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to the conditions specified below.

1.3.2 ELECTRICAL CHARACTERISTICS

T_A = 0ºC to +70ºC, V_{CC} = 5 V ±10% Or 3.3 V ±10%, GND = 0 V

Table 1-9: Electrical Characteristics—Preliminary Information
(All Parameters with Reference to V_{CC} = 3.3 V)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_X	X1 and X2 Capacitance[1]			17.0		pF
I_{CC1}	Active Supply Current	Normal Operation Mode, Running Speech Applications[2]		40.0	80.0	mA
I_{CC2}	Standby supply current	Normal Operation Mode, DSPM Idle[2]		30.0		mA
I_{CC3}	Power-down Mode Supply Current	Power-down Mode[2,3]			0.7	mA
I_L	Input Load Current	0 V ≤ V_{IN} ≤ V_{CC}	–5.0		5.0	µA
I_O (Off)	Output Leakage Current (I/O pins in Input Mode)	0 V ≤ V_{OUT} ≤ V_{CC}	–5.0		5.0	µA
t_{CASa}	$\overline{CAS}$ Active	After R.E. CTTL, T1 or T2W3			12.0	
t_{CASh}	$\overline{CAS}$ Hold	After R.E. CTTL	0.0			
t_{CASia}	$\overline{CAS}$ Inactive	After R.E. CTTL, T3 or TERF			12.0	
t_{CASLw}	DRAM, PDM, $\overline{CAS}$ Width	At 0.8 V, Both Edges	600.0			
t_{DWEa}	$\overline{DWE}$ Active	After R.E. CTTL, T2W2			12.0	
t_{DWEh}	$\overline{DWE}$ Hold	After R.E. CTTL			12.0	
t_{DWEia}	$\overline{DWE}$ Inactive	After R.E. CTTL, T3			12.0	
t_{RASa}	$\overline{RAS}$ Active	After R.E. CTTL, T2W1 or T2WRF			12.0	
t_{RASh}	$\overline{RAS}$ Hold	After R.E. CTTL	0.0			
t_{RASia}	$\overline{RAS}$ Inactive	After R.E. CTTL, T3 or T3RF			12.0	
t_{RASLw}	DRAM PDM, $\overline{RAS}$ Width	At 0.8 V, Both Edges	200.0			
t_{RLCL}	DRAM PDM $\overline{RAS}$ Low, after $\overline{CAS}$ Low	F.E. $\overline{CAS}$ to F.E. $\overline{RAS}$	200.0			
t_{WRa}	$\overline{WR0}$ Active	After R.E. CTTL, T1			t_{CTp}/2+2	

Table 1-9: Electrical Characteristics—Preliminary Information
(All Parameters with Reference to V_{CC} = 3.3 V)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WRCSh}	$\overline{WR0}$ Hold after $\overline{EMCS}$[4]	R.E. $\overline{EMCS}$ R.E. to R.E. $\overline{WR0}$	10.0			
t_{WRh}	$\overline{WR0}$ Hold	After R.E. CTTL	$t_{CTp}/2–6$			
t_{WRia}	$\overline{WR0}$ Inactive	After R.E. CTTL, T3			$t_{CTp}/2+2$	
V_{ENVh}	ENV0 Input, High Voltage		2.0			V
V_{Hh}	CMOS Input with Hysteresis, Logical 1 Input Voltage		2.1			V
V_{HI}	CMOS Input with Hysteresis, Logical 0 Input Voltage				0.8	V
V_{Hys}	Hysteresis Loop Width[1]		0.5			V
V_{IH}	TTL Input, Logical 1 Input Voltage		2.0		V_{CC}+0.5	V
V_{IL}	TTL Input, Logical 0 Input Voltage		–0.5		0.8	V
V_{OH}	Logical 1 TTL, Output Voltage	I_{OH} = –0.4 mA	2.4			V
V_{OHWC}	MMCLK, MMDOUT and $\overline{EMCS}$ Logical 1, Output Voltage	I_{OH} = –0.4 mA	2.4			V
		I_{OH} = –50 µA[5]	V_{CC}–0.2			V
V_{OL}	Logical 0, TTL Output Voltage	I_{OL} = 4 mA			0.45	V
		I_{OL} = 50 µA[5]			0.2	V
V_{OLWC}	MMCLK, MMDOUT and $\overline{EMCS}$ Logical 0, Output Voltage	I_{OL} = 4 mA			0.45	V
		I_{OL} = 50 µA[5]			0.2	V
V_{XH}	CLKIN Input, High Voltage	External Clock	2.0			V
V_{XL}	CLKIN Input, Low Voltage	External Clock			0.8	V

1. *Guaranteed by design.*
2. *I_{OUT} =0, T_A 25˚C, V_{CC} = 3.3 V for V_{CC} pins and 3.3 V or 5 V on V_{CCHI} pins, operating from a 4.096 MHz crystal and running from internal memory with Expansion Memory disabled.*
3. *All input signals are tied to 0 (above V_{CC} – 0.5 V or below V_{SS} + 0.5 V), except ENV0, which is tied to V_{CC}.*
4. *Measured in power-down mode. The total current driven, or sourced, by all the VoiceDSP processor's output signals is less than 50 µA.*
5. *Guaranteed by design, but not fully tested.*

1.3.3 SWITCHING CHARACTERISTICS—PRELIMINARY

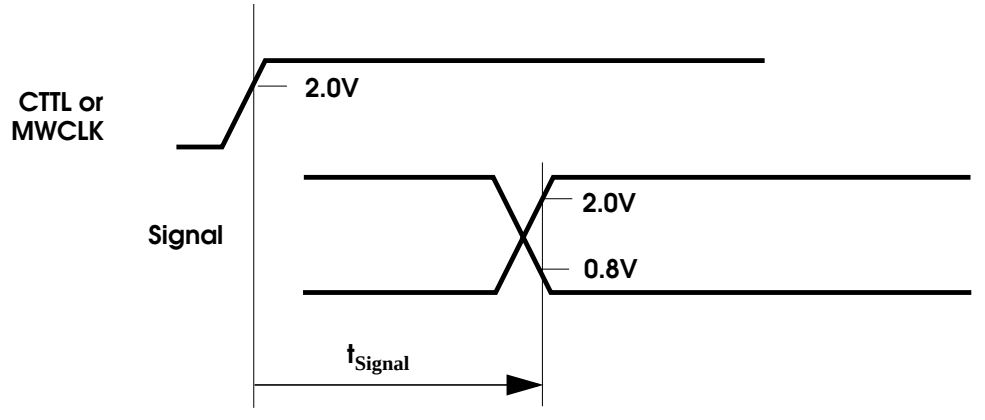
Definitions

All timing specifications in this section refer to 0.8 V or 2.0 V on the rising or falling edges of the signals, as illustrated in Figure 1-17 through Figure 1-23, unless specifically stated otherwise.

Maximum times assume capacitive loading of 50pF. CLKIN crystal frequency is 4.096 MHz.

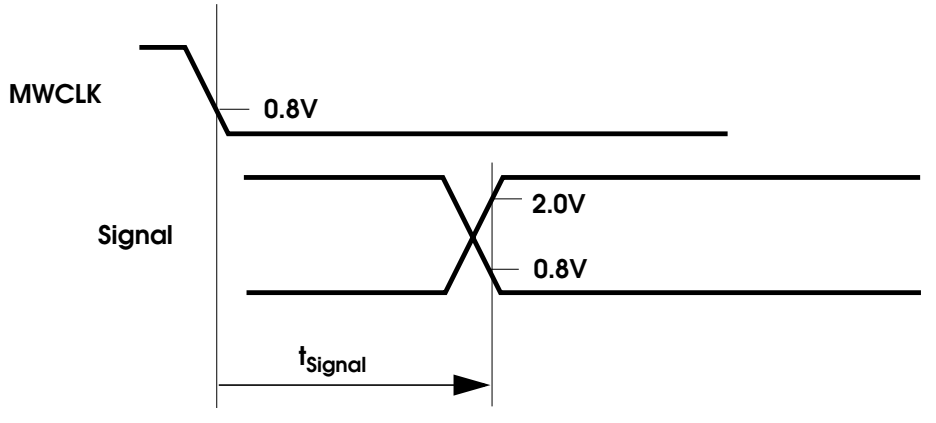
NOTE CTTL is an internal signal and is used as a reference to explain the timing of other signals. See Figure 1-37.

Figure 1-17: Synchronous Output Signals (Valid, Active and Inactive)



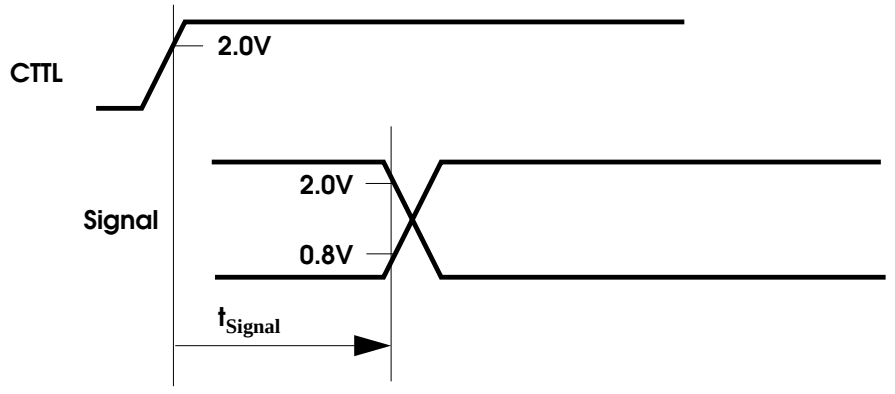
NOTE: *Signal valid, active or inactive time, after a rising edge of CTTL or MWCLK.*

Figure 1-18: Synchronous Output Signals (Valid)



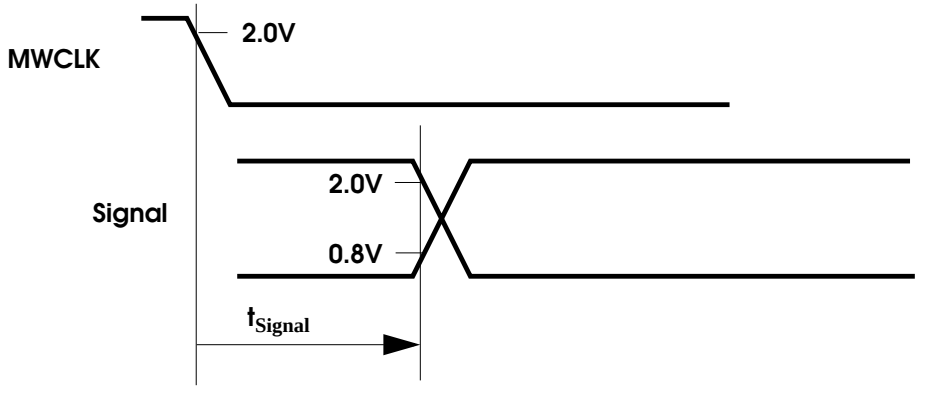
NOTE: *Signal valid time, after a falling edge of MWCLK.*

Figure 1-19: Synchronous Output Signals (Hold), after Rising Edge of CTTL



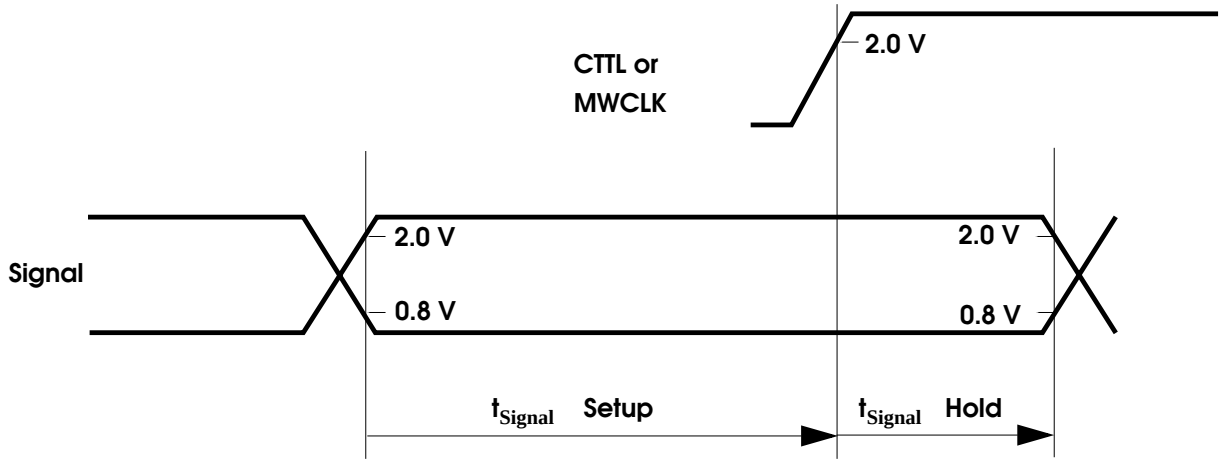
NOTE: *Signal hold time, after a rising edge of CTTL.*

Figure 1-20: Synchronous Output Signals (Hold), after Falling Edge of MWCLK



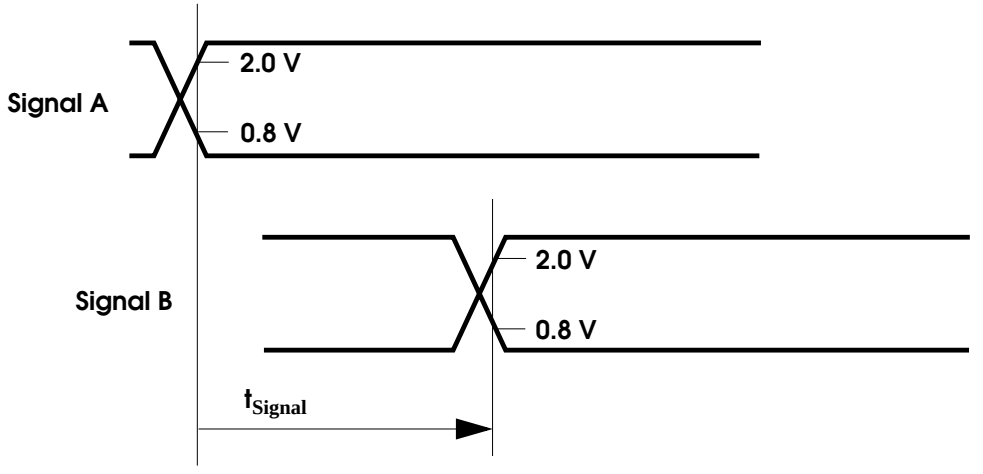
NOTE: *Signal hold time, after a falling edge of MWCLK.*

Figure 1-21: Synchronous Input Signals



NOTE: *Signal setup time, before a rising edge of CTTL or MWCK, and signal hold time after a rising edge of CTTL or MWCK*

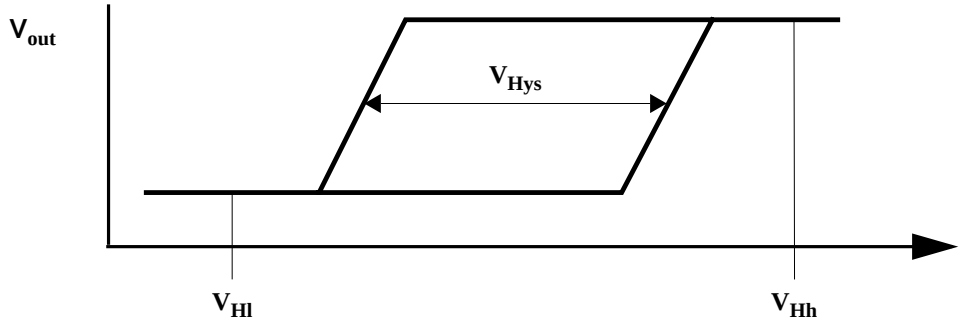
Figure 1-22: Asynchronous Signals



NOTE: *Signal B starts after rising or falling edge of signal A.*

The $\overline{\text{RESET}}$ has a Schmitt trigger input buffer. Figure 1-23 shows the input buffer characteristics.

Figure 1-23: Hysteresis Input Characteristics



1.3.4 SYNCHRONOUS TIMING TABLES

In this section, R.E. means Rising Edge and F.E. means Falling Edge.

Table 1-10: Output Signals—Preliminary

Symbol	Figure	Description	Reference Conditions	Min (ns)	Max (ns)
t_{Ah}		Address Hold	After R.E. CTTL	0.0	
t_{Av}		Address Valid	After R.E. CTTL, T1		12.0
t_{CCLKa}		CCLK Active	After R.E. CTTL		12.0
t_{CCLKh}		CCLK Hold	After R.E. CTTL	0.0	
t_{CCLKia}		CCLK Inactive	After R.E. CTTL		12.0
t_{CDOh}		CDOUT Hold	After R.E. CTTL	0.0	
t_{CDOv}		CDOUT Valid	After R.E. CTTL		12.0
t_{CTp}		CTTL Clock Period[1]	R.E. CTTL to next R.E. CTTL	30.5	250,000
t_{EMCSa}		$\overline{\text{EMCS}}$ Active	After R.E. CTTL, T2W1		12.0
t_{EMCSh}		$\overline{\text{EMCS}}$ Hold	After R.E. CTTL	0.0	
t_{EMCSia}		$\overline{\text{EMCS}}$ Inactive	After R.E. CTTL T3		12.0
t_{FSa}		CFS0 Active	After R.E. CTTL		25.0
t_{FSh}		CFS0 Hold	After R.E. CTTL	0.0	
t_{FSia}		CFS0 Inactive	After R.E. CTTL		25.0
t_{MMCLKa}		Master MICROWIRE Clock Active	After R.E. CTTL		12.0
t_{MMCLKh}		Master MICROWIRE Clock Hold	After R.E. CTTL	0.0	
$t_{MMCLKia}$		Master MICROWIRE Clock Inactive	After R.E. CTTL		12.0
t_{MMDOh}		Master MICROWIRE Data Out Hold	After R.E. CTTL	0.0	
t_{MMDOv}		Master MICROWIRE Data Out Valid	After R.E. CTTL		12.0
t_{MWDOf}		MICROWIRE Data Float[1]	After R.E. $\overline{\text{MWCS}}$		70.0
t_{MWDOh}		MICROWIRE Data Out Hold[2]	After F.E. MWCLK	0.0	
t_{MWDOnf}		MICROWIRE Data No Float[2]	After F.E. $\overline{\text{MWCS}}$	0.0	70.0
t_{MWDOv}		MICROWIRE Data Out Valid[2]	After F.E. MWCLK		70.0
t_{MWITOp}		MWDIN to MWDOUT	Propagation Time		70.0
t_{MWRDYa}		$\overline{\text{MWRDY}}$ Active	After R.E. of CTTL	0.0	35.0
$t_{MWRDYia}$		$\overline{\text{MWRDY}}$ Inactive	After F.E. MWCLK	0.0	70.0
t_{PABCh}		PB and $\overline{\text{MWRQST}}$	After R.E. CTTL	0.0	
t_{PABCv}		PB and $\overline{\text{MWRQST}}$	After R.E. CTTL, T2W1		12.0

1. In normal operation mode, t_{CTp} must be 48.8 ns; in power-down mode, t_{CTp} must be 50,000 ns.
2. Guaranteed by design, but not fully tested.

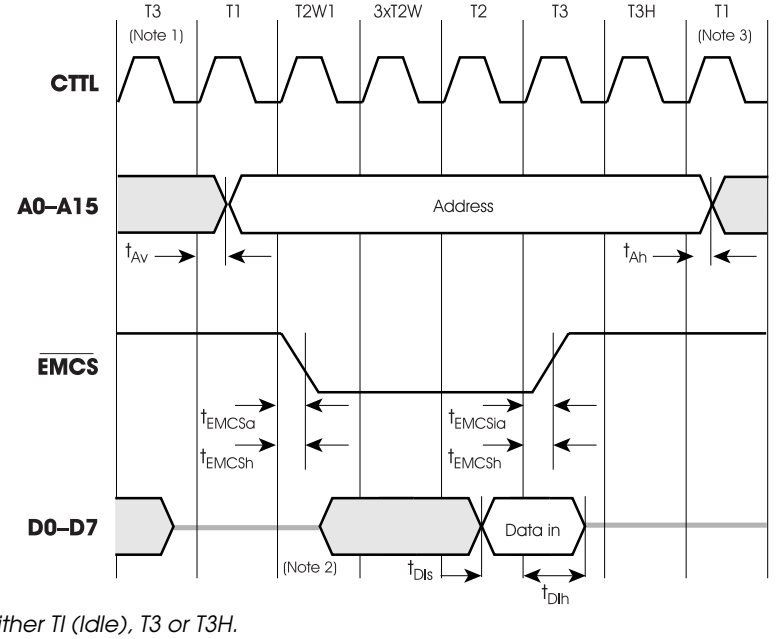
Table 1-11: Input Signals—Preliminary

Symbol	Figure	Description	Reference Conditions	Min (ns)	Max (ns)
t_{CCLKSp}		Codec Clock Period (slave)	R.E. CCLK to next R.E. CCLK	244	
t_{CCLKSh}		Codec Clock High (slave)	At 2.0 V (both edges)	120	
t_{CCLKSl}		Codec Clock Low (slave)	At 0.8 V (both edges)	120	
t_{CDIh}		CDIN Hold	After R.E. CTTL	0.0	
t_{CDIs}		CDIN Setup	Before R.E. CTTL	11.0	
t_{CFS0Ss}		CFS0 Setup	Before R.E. CCLK	TBD	
t_{CFS0Sh}		CFS0 Hold	After R.E. CCLK	TBD	
t_{DIh}		Data in Hold (D0:7)	After R.E. CTTL T1, T3 or TI	0.0	
t_{DIs}		Data in Setup (D0:7)	Before R.E. CTTL T1, T3 or TI	15.0	
t_{MMDINh}		Master MICROWIRE Data In Hold	After R.E. CTTL	0.0	
t_{MMDINs}		Master MICROWIRE Data In Setup	Before R.E. CTTL	11.0	
t_{MWCKh}		MICROWIRE Clock High (slave)	At 2.0 V (both edges)	100.0	
t_{MWCKl}		MICROWIRE Clock Low (slave)	At 0.8 V (both edges)	100.0	
t_{MWCKp}		MICROWIRE Clock Period (slave)[1]	R.E. MWCLK to next R.E. MWCLK	2.5 µs	
t_{MWCLKh}		MWCLK Hold	After $\overline{MWCS}$ becomes inactive	50.0	
t_{MWCLKs}		MWCLK Setup	Before $\overline{MWCS}$ becomes active	100.0	
t_{MWCSh}		$\overline{MWCS}$ Hold	After F.E. MWCLK	50.0	
t_{MWCSs}		$\overline{MWCS}$ Setup	Before R.E. MWCLK	100.0	
t_{MWDIh}		MWDIN Hold	After R.E. MWCLK	50.0	
t_{MWDIs}		MWDIN Setup	Before R.E. MWCLK	100.0	
t_{PWR}		Power Stable to $\overline{RESET}$ R.E.[2]	After V_{CC} reaches 4.5 V	30.0 ms	
t_{RSTw}		$\overline{RESET}$ Pulse Width	At 0.8 V (both edges)	10.0 ms	
t_{Xh}		CLKIN High	At 2.0 V (both edges)	$t_{X1p}/2 - 5$	
t_{Xl}		CLKIN Low	At 0.8 V (both edges)	$t_{X1p}/2 - 5$	
t_{Xp}		CLKIN Clock Period	R.E. CLKIN to next R.E. CLKIN	244.4	

1. Guaranteed by design, but not fully tested in power-down mode.
2. Guaranteed by design, but not fully tested.

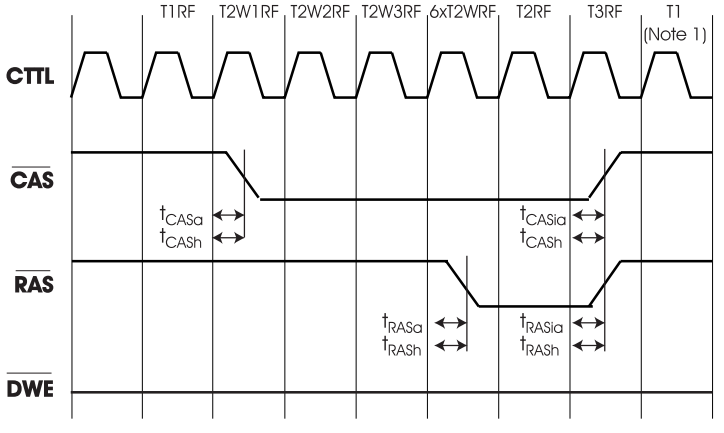
1.3.5 TIMING DIAGRAMS

Figure 1-24: ROM Read Cycle Timing



1. *This cycle may be either TI (Idle), T3 or T3H.*
2. *Data can be driven by an external device at T2W1, T2W, T2 and T3.*
3. *This cycle may be either TI (Idle) or T1.*

Figure 1-25: ARAM/DRAM Refresh Cycle Timing (Normal Operation)



1. *This cycle may be either TI (Idle) or T1 of any non-DRAM bus cycle. If the next bus cycle is a DRAM one, T3RF is followed by three TI (Idle) cycles.*

Figure 1-26: ARAM/DRAM Power-Down Refresh Cycle Timing

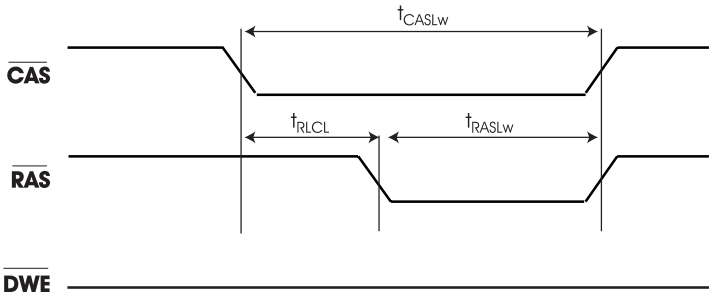
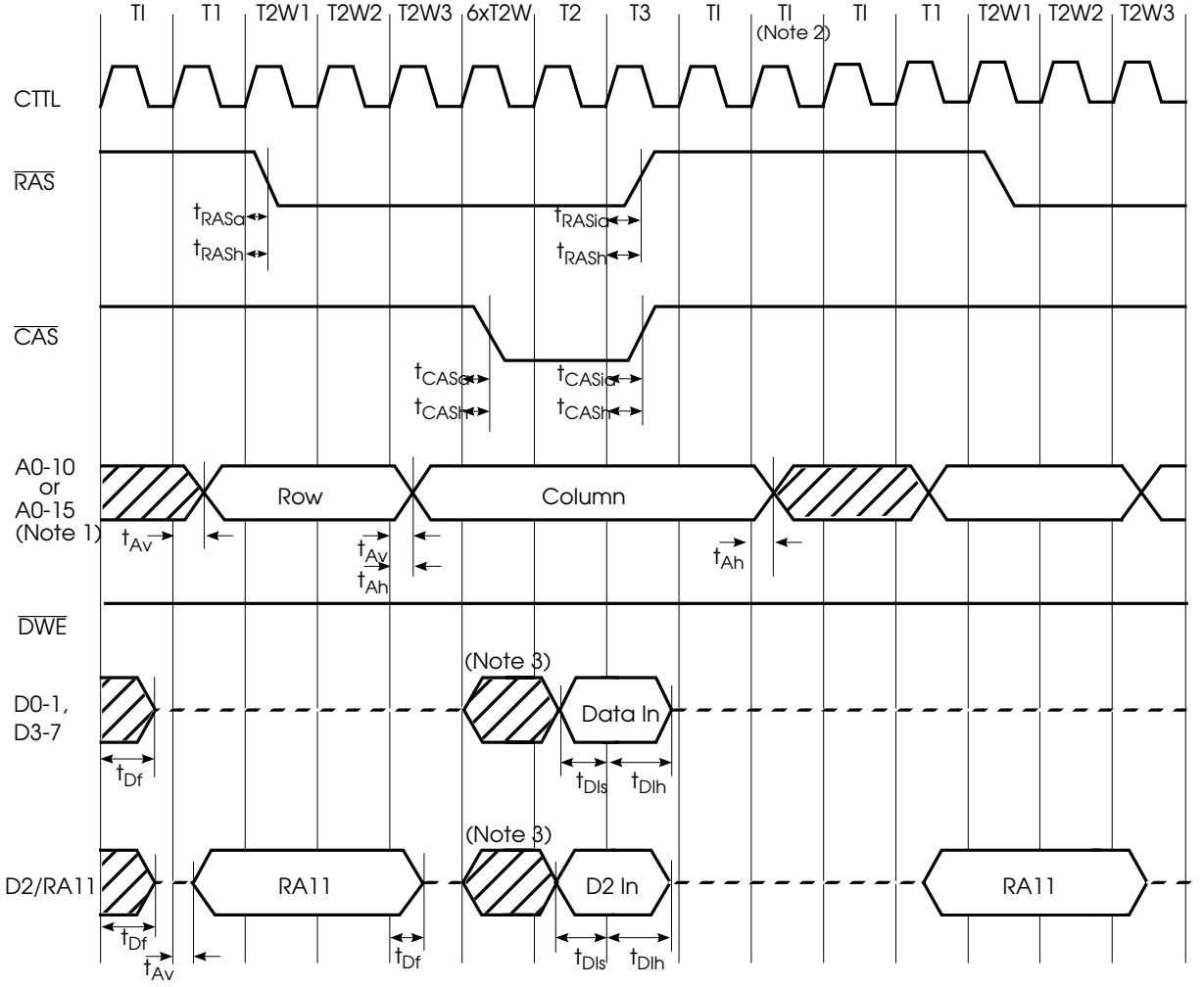
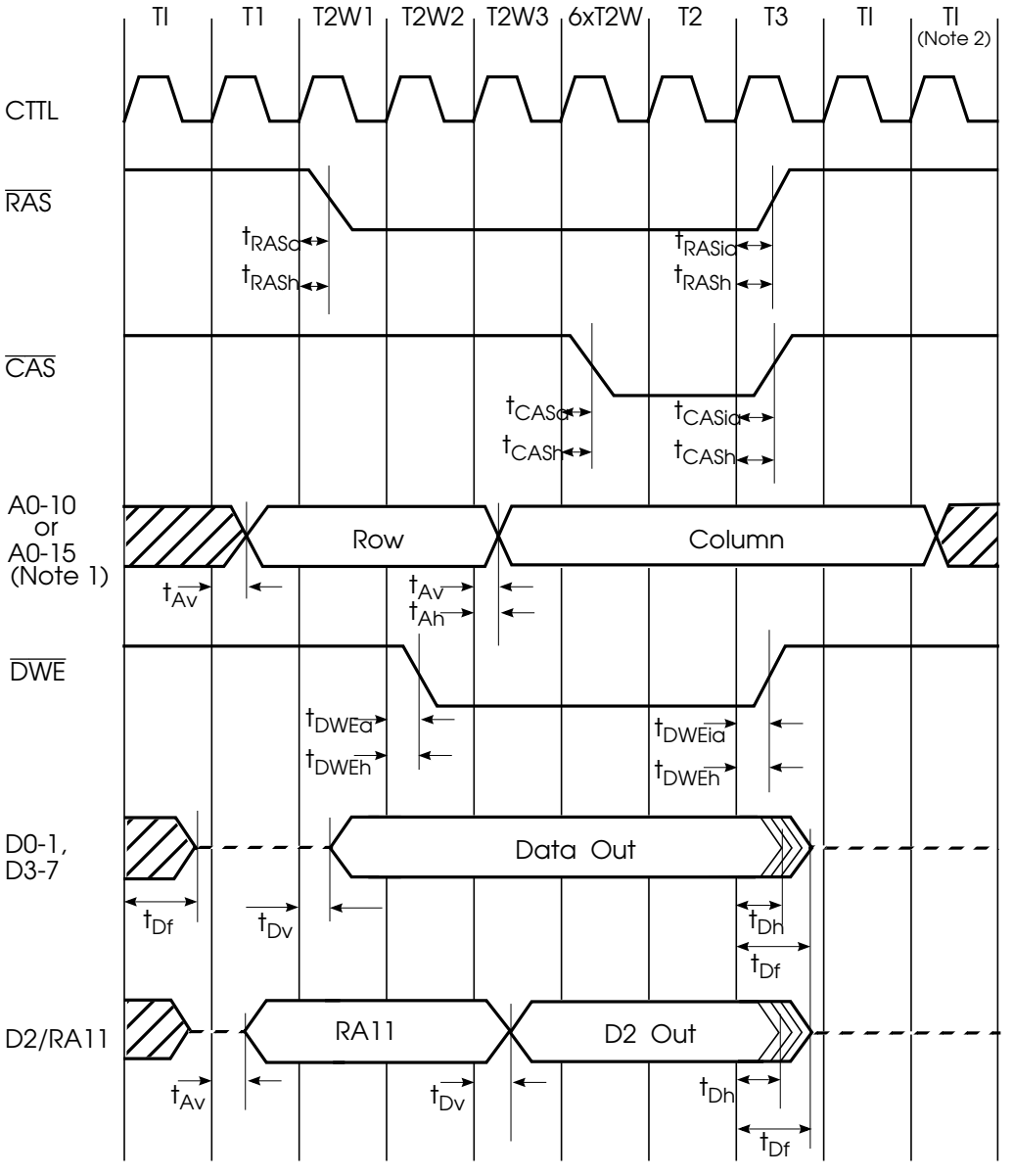


Figure 1-27: DRAM Read Cycle Timing



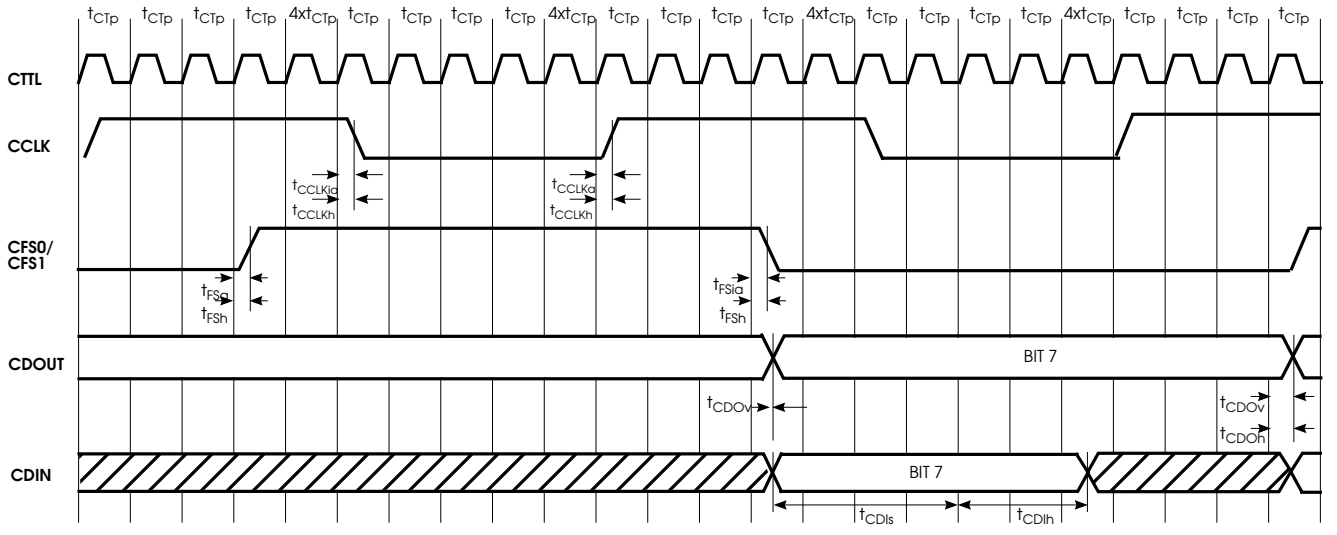
1. *A0-A10 in the IRE environment, when Expansion Memory is disabled; otherwise A0-A15.*
2. *This cycle may be either TI (Idle) or T1 of any non-DRAM bus cycle. If the next bus cycle is to DRAM, T3 is followed by three TI (Idle) cycles.*
3. *An external device can drive data from T2W3 to T3.*

Figure 1-28: DRAM Write Cycle Timing



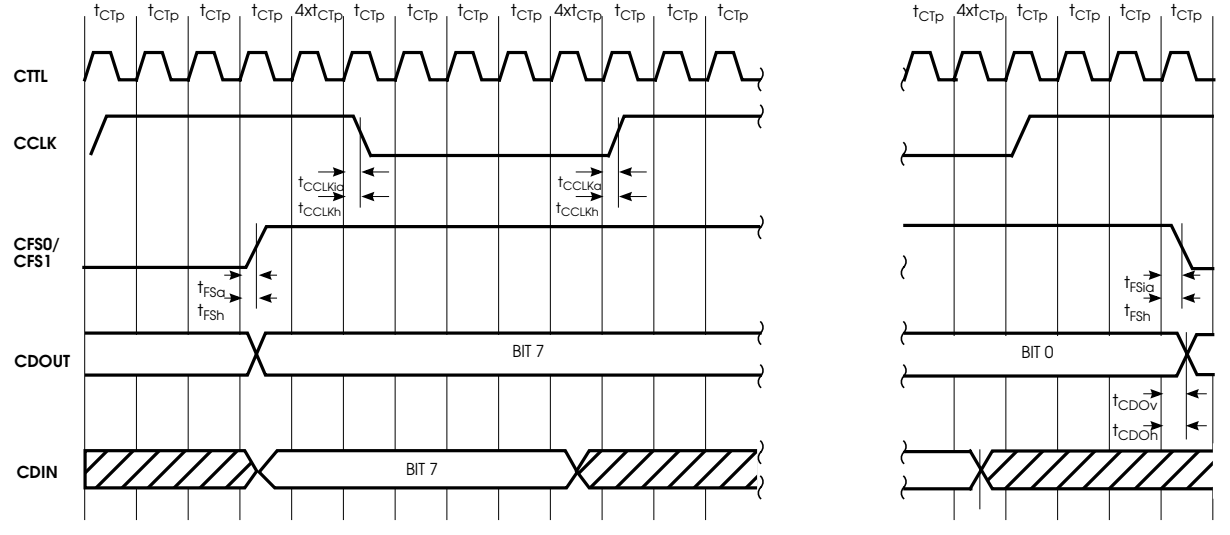
1. *A0–A10 in the IRE environment, when Expansion Memory is disabled; otherwise A0–A15.*
2. *This cycle may be either TI (Idle) or T1 of any non-DRAM bus cycle. If the next bus cycle is to DRAM, T3 is followed by three TI (idle) cycles.*

Figure 1-29: Codec Short Frame Timing



NOTE: *The CCLK and CFS0 timing is shown for Master Mode only. For Slave Mode, see Figure 1-31.*

Figure 1-30: Codec Long Frame Timing



NOTE: *The CCLK and CFS0 timing is shown for Master Mode only. For Slave Mode, see Figure 1-31.*

Figure 1-31: Slave Codec CCLK and CFS0 Timing

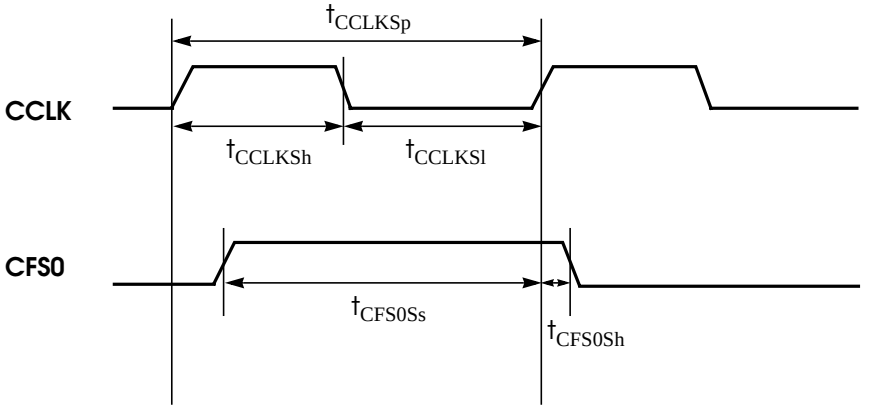


Figure 1-32: MICROWIRE Transaction Timing—Data Transmitted to Output

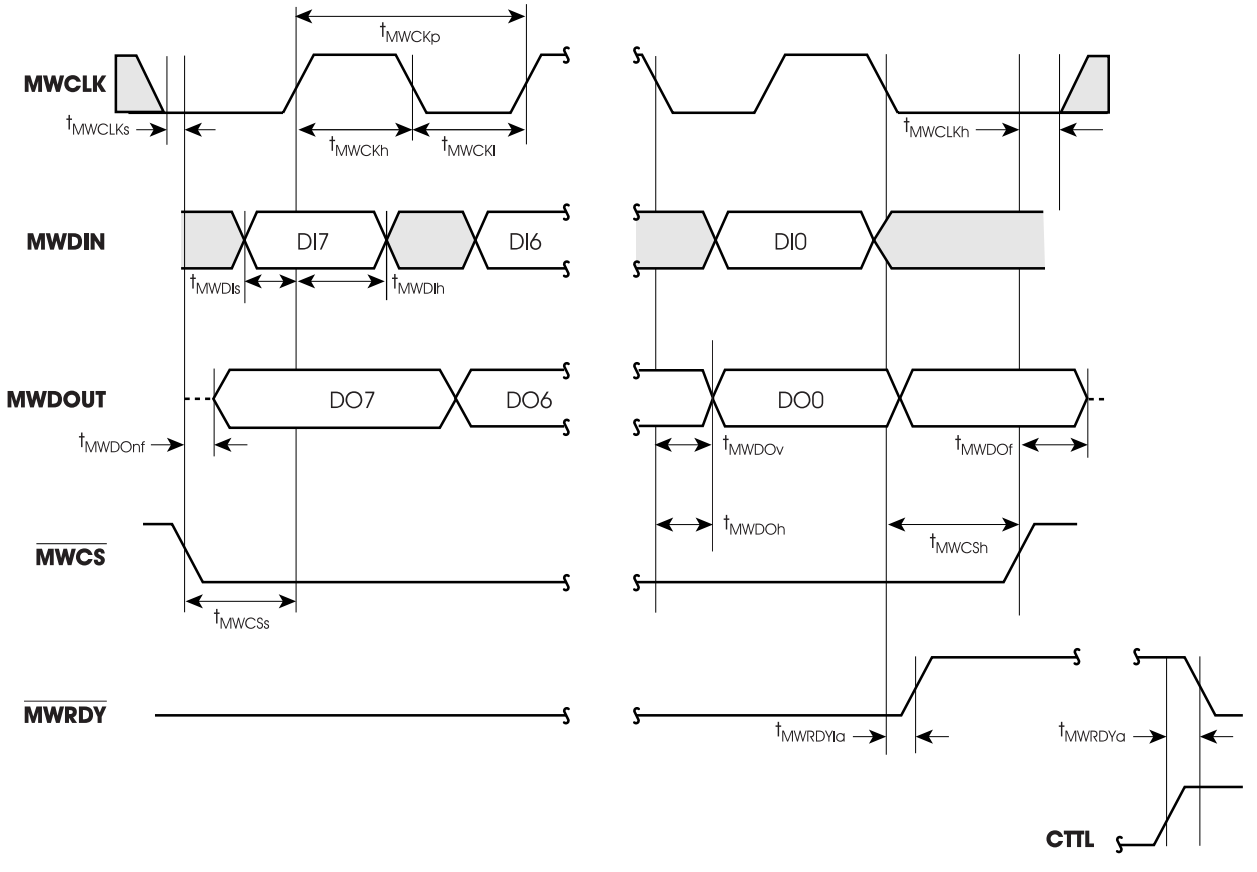


Figure 1-33: MICROWIRE Transaction Timing—Data Echoed to Output

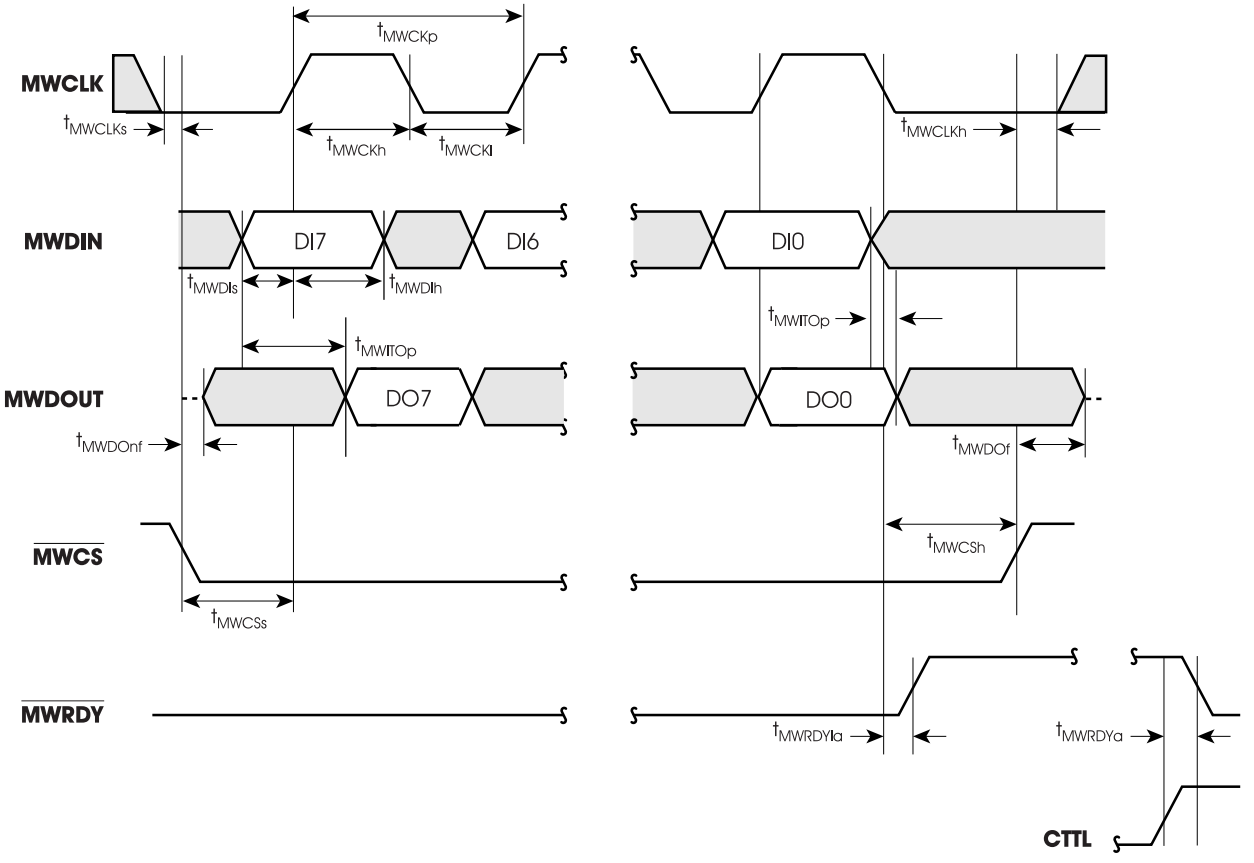


Figure 1-34: Master MICROWIRE Timing

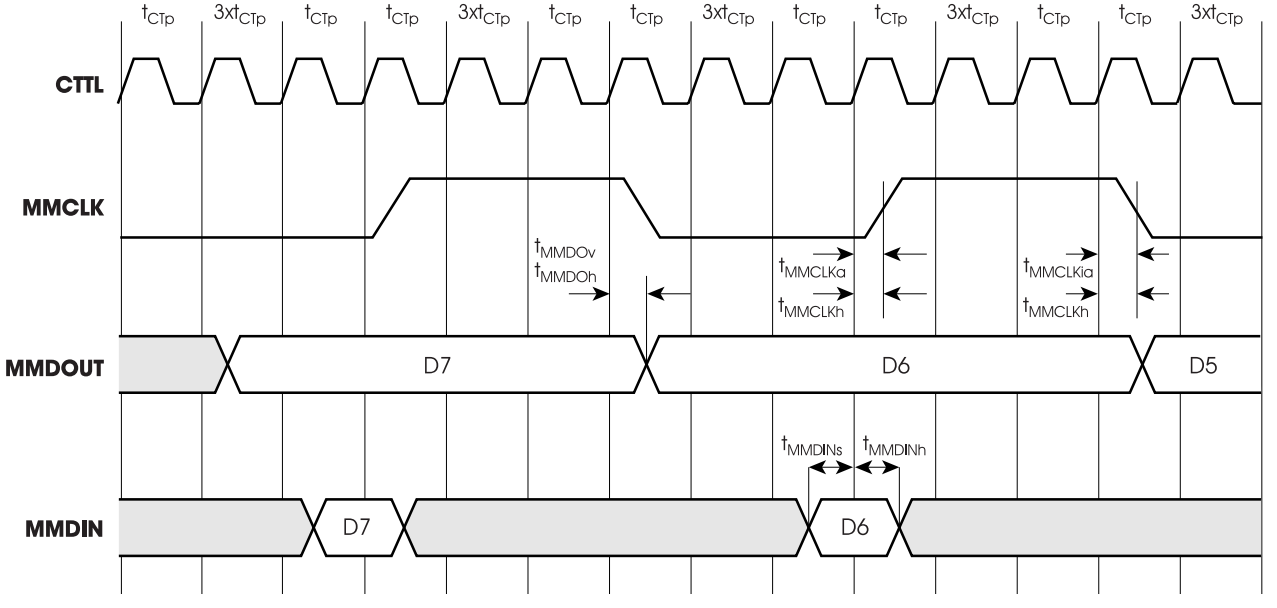
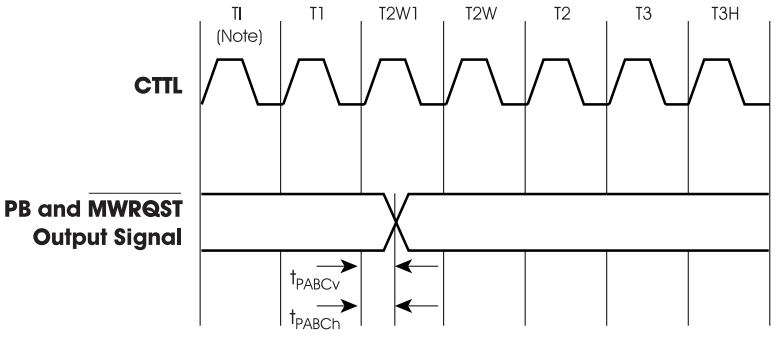


Figure 1-35: Output Signal Timing for Port PB and $\overline{\text{MWRQST}}$



NOTE: *This cycle may be either TI (Idle), T2, T3 or T3H.*

Figure 1-36: CLKIN Timing

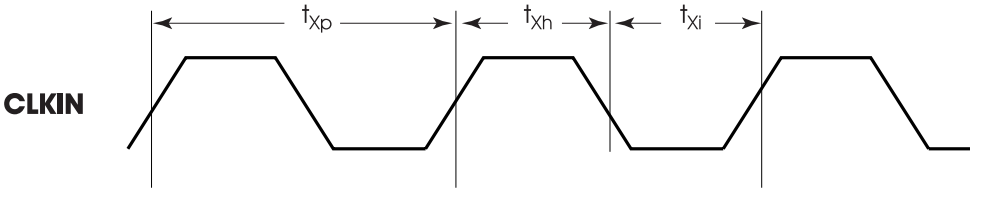


Figure 1-37: CTTL Timing

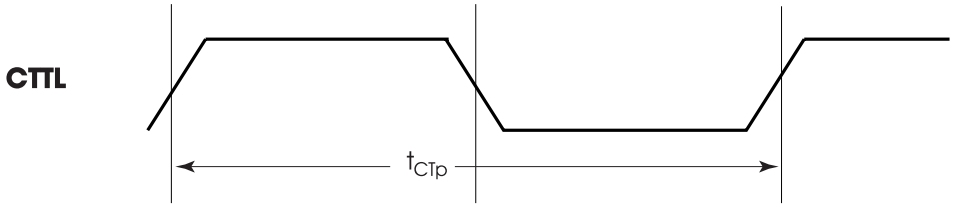


Figure 1-38: Reset Timing When Reset Is Not at Power-Up

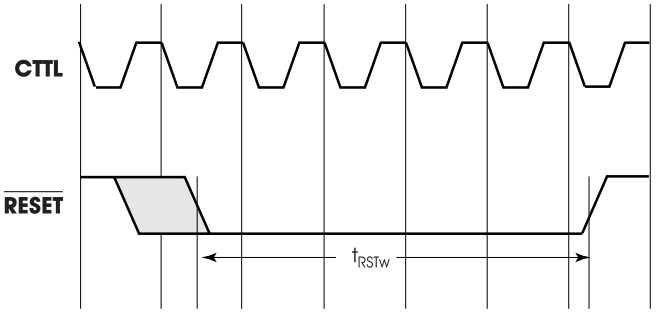
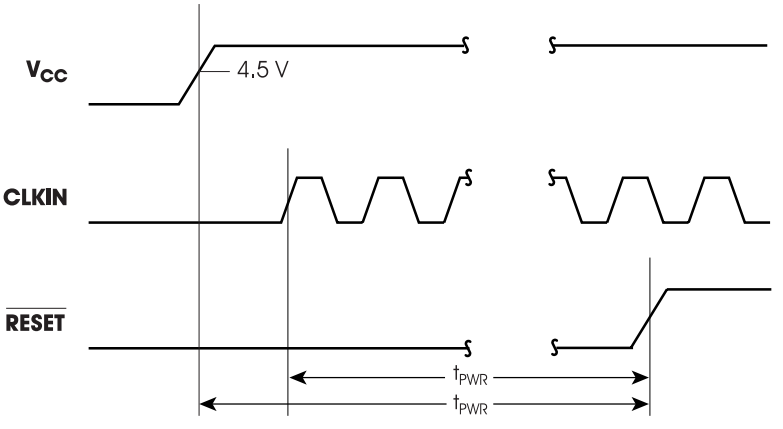


Figure 1-39: Reset Timing When Reset Is at Power-Up



Chapter 2—SOFTWARE

The VoiceDSP software resides in the on-chip ROM. It includes DSP-based algorithms, system support functions and a software interface to hardware peripherals.

2.1 SYSTEM OPERATION

This section provides details of the system support functions and their principle operation. It is divided into the following subjects:

- The State Machine
- Command Execution
- Event Handling
- Message Handling
- Tone Generation
- Initialization and Configuration
- Power Down Mode (PDM)

2.1.1 THE STATE MACHINE

The ISD-T360SB operates in two modes, normal mode (DTAD) and speakerphone mode. To change the mode use the Set Speakerphone Mode (SSM) command. The VoiceDSP processor functions as a state machine under each mode. It changes state either in response to a command sent by the microcontroller, after execution of the command is completed, or as a result of an internal event (e.g. memory full or power failure). For more information see "VoiceDSP PROCESSOR COMMANDS—QUICK REFERENCE TABLE" on page 2-21. The VoiceDSP states are desctibed below:

RESET

The VoiceDSP processor is initialized to the RESET state after a full hardware reset by the $\overline{\text{RESET}}$ signal (See "RESETTING" on page 1-3). In this state the processor detectors (VOX, constant energy, call progress tones and DTMF) are not active. In all other states, these detectors are active. (See the SDET and RDET commands for further details).

IDLE

This is the state from which most commands are executed. As soon as a command and all its parameters are received, the processor starts executing the command.

PLAY

In this state a message is decompressed (unless stored in PCM format), and played back.

RECORD

In this state a message is compressed (unless stored in PCM format) and recorded into the message memory.

SYNTHESIS

An individual word or a sentence is synthesized from an external vocabulary in this state.

TONE_GENERATE

In the TONE_GENERATE state, the VoiceDSP processor generates single or DTMF tones.

MSG_OPEN

The VoiceDSP processor either reads or writes 32 bytes to the message memory, or sets the message read/write pointer on a 32 byte boundary.

2.1.2 COMMAND EXECUTION

A VoiceDSP processor command is represented by an 8-bit opcode. Some commands have parameters and some have return values. Commands are either synchronous or asynchronous.

SYNCHRONOUS COMMANDS

A synchronous command must complete execution before the microcontroller can send a new command (e.g. GMS, GEW). A command sequence begins when the microcontroller sends an 8-bit opcode to the processor, followed by the command's parameters (if any). The VoiceDSP processor then executes the command and, if required, transmits a return value to the microcontroller. Upon completion, the processor notifies the microcontroller that it is ready to accept a new command.

ASYNCHRONOUS COMMANDS

An asynchronous command starts execution in the background and notifies the microcontroller, which can send more commands while the current command is still running (e.g. R, P). After receiving an asynchronous command, such as P (Playback), R (Record), SW (Say Words) or GT (Generate Tone), the VoiceDSP processor switches to the appropriate state and executes the command until finished or a S (Stop) or PA (Pause) command is received from the microcontroller. When completed, the EV_NORMAL _END event is set and the processor switches to the IDLE state.

"VoiceDSP PROCESSOR COMMANDS—QUICK REFERENCE TABLE" on page 2–21 displays all the processor commands, the valid source states in which these commands are valid, and the states resulting from the command.

2.1.3 EVENT HANDLING

STATUS WORD

The 16-bit status word indicates events that occur during normal operation. The VoiceDSP processor activates the $\overline{\text{MWRQST}}$ signal, to indicate a change in the status word. This signal remains active until the processor receives a GSW (Get Status Word) command.

For detailed description of the Status Word and the meaning of each bit, see "GSW Get Status Word" on page 2-33.

ERROR WORD

The 16-bit error word indicates errors that occurred during execution of the last command. If an error is detected, the command is not processed; the EV_ERROR bit in the status word is set to 1, and the $\overline{\text{MWRQST}}$ signal is activated.

ERROR HANDLING

When the microcontroller detects the active $\overline{\text{MWRQST}}$ signal, it issues the GSW command, deactivating the $\overline{\text{MWRQST}}$ signal. Then, the microcontroller tests the EV_ERROR bit in the status word, and, if set, sends the GEW (Get Error Word) command to read the error word for details.

For detailed description of the Error Word and the meaning of each bit, see "GEW Get Error Word" on page 2-30.

2.1.4 MESSAGE HANDLING

A message is the basic unit on which most of the VoiceDSP commands operate. A VoiceDSP processor message, stored on a memory device (Flash or ARAM/DRAM), can be regarded as a computer file stored on a mass-storage device.

The ISD-T360SB manages messages for a wide range of applications, which require different levels of DTAD functionality. The VoiceDSP processor features advanced memory-organization features such as multiple OutGoing Messages (OGMs), mailboxes, and the ability to distinguish between InComing Messages (ICMs) and OGMs.

A message is created with either the R (Record) or the CMSG (Create Message) command. Once created, the message is assigned a time-and-day stamp and a message tag which is read by the microcontroller. The R command takes voice samples from the codec, compresses them, and stores them in the message memory.

When a message is created with the CMSG command the data to be recorded is provided by the microcontroller, via the WMSG (Write Message) command and not through the codec. Here, the data is transferred directly to the message memory, and not compressed by the ISD-T360SB voice compression algorithm.

WMSG, RMSG (Read Message) and SMSG (Set Message Pointer) are message-data access commands used to store and read data to or from any location in the message memory (see "VoiceDSP PROCESSOR COMMANDS—QUICK REFERENCE TABLE" on page 2-21 for more details). Using these commands, the microcontroller utilizes messages for features such as a Telephone Directory and playing back (P command) or deleting (DM command) a message. Remove redundant data (e.g., trailing tones or silence) from the message tail with the CMT (Cut Message Tail) command.

The PA (Pause) and RES (Resume) commands suspend the P and R commands, respectively, and then resume them from where they were suspended.

CURRENT MESSAGE

The GTM (Get Tagged Message) command selects the current message. Most message handling commands (P, DM, RMSG), operate on the current message.

Deleting the prevailing message does not cause a different message to become current; the current message is undefined. If you issue the GTM command to skip to the next message, the first message, newer than the just deleted message, becomes the current message.

MESSAGE TAG

Each message has a 2-byte message tag which used to categorize messages, and implement such features as OutGoing Messages, mailboxes, and different handling of old and new messages.

The tag is created during the R (Record) command. Use the GMT (Get Message Tag) and SMT (Set Message Tag) commands to handle message tags.

NOTE *Message tag bits can only be cleared and are set only when a message is first created. This limitation, inherent in Flash memories, allows bits to be changed only from 1 to 0 (changing bits from 0 to 1 requires a special erasure procedure). However, the usual reason for updating an existing tag is to mark a message as old. This can be done when a message is first created by using one of the bits as a new/old indicator, setting the bit to 1 and later clearing it when necessary.*

2.1.5 TONE GENERATION

The VoiceDSP processor generates DTMF tones and single-frequency tones from 300Hz to 3000Hz in increments of 100Hz. The ISD-T360SB tone generation conforms to the EIA-470-RS standard. Note, however, that value of some tunable parameters may need adjusting to meet the standard specifications since the energy level of generated tones depends on the analog circuits used.

1. Tune the DTMF_GEN_TWIST_LEVEL parameter to control the twist level of the generated DTMF tones.
2. Use the VC (Volume Control) command, and tune the TONE_GEN_LEVEL parameter, to control the energy level at which these tones are generated.
3. Use the GT (Generate Tone) command to specify the DTMF tones, and the frequency at which single tones are generated.

Refer to table 2-5, VC command and GT command of the Command Description section for furhter details of the relevant tunable parameters and commands.

NOTE The DTMF detector performance is degraded during tone generation, especially if the frequency of the generated tone is close to the frequency of one of the DTMF tones.

2.1.6 INITIALIZATION AND CONFIGURATION

Use the following procedures to initialize the VoiceDSP processor:

NORMAL INITIALIZATION

Reset the VoiceDSP processor by activating the RESET signal. (See "RESETTING" on page 1-3.)

1. Issue a CFG (Configure VoiceDSP processor) command to change the configuration according to your environment.
2. Issue an INIT (Initialize System) command to initialize the VoiceDSP firmware.
3. Issue a series of TUNE commands to adjust the VoiceDSP processor to the requirements of your system.

TUNABLE PARAMETERS

The VoiceDSP processor can be adjusted to the specific system's requirements using a set of tunable parameters. These parameters are set to their default values after reset and can be later modified with the TUNE command. By tuning these parameters, you can control various aspects of the VoiceDSP processor's operation, such as silence compression, tone detection, and no-energy detection.

Tables 2-4 to 2-11 of the Command Description section describe all the tunable parameters in detail.

2.1.7 POWER-DOWN MODE

The PDM (Go To Power-Down Mode) command switches the ISD-T360SB to power-down mode. The purpose of the PDM command is to save power during buttery operation, or for any other power saving cause. During power-down mode only basic functions, such as ARAM/DRAM refresh and time and date update, are active (for more details refer to POWER-DOWN MODE description on page 1-4).

This PDM command may only be issued when the processor is in the IDLE mode (for an explanation of the ISD-T360SB states, see "Command Execution" on page 2–2). If it is necessary to switch to power-down mode from any other state, the controller must first issue a S (Stop) command to switch the processor to the IDLE state, and then issue the PDM command. Sending any command while in power-down mode resets the VoiceDSP processor detectors, and returns it to normal operation mode.

NOTE *Entering or exiting power-down mode can distort the real-time clock by up to 500 µs. Thus, to maintain the accuracy of the real-time clock, enter or exit the power-down mode as infrequently as possible.*

2.2 PERIPHERALS

This section provides details of the peripherals interface support functions and their principle operation. It is divided into the following subjects:

- Microcontroller Interface (Slave MICROWIRE)
- Memory Interface
- Codec Interface

2.2.1 MICROCONTROLLER INTERFACE

MICROWIRE/PLUS™ is a synchronous serial communication protocol minimizes the number of connections, and thus the cost, of communicating with peripherals.

The VoiceDSP MICROWIRE interface implements the MICROWIRE/PLUS interface in slave mode, with an additional ready signal. It enables a microcontroller to interface efficiently with the VoiceDSP processor application.

The microcontroller is the protocol master and provides the clock for the protocol. The VoiceDSP processor supports clock rates of up to 400 KHz. This transfer rate refers to the bit transfer; the actual throughput is slower due to byte processing by the VoiceDSP processor and the microcontroller.

Communication is handled in bursts of eight bits (one byte). In each burst the VoiceDSP processor is able to receive and transmit eight bits of data. After eight bits have been transferred, an internal interrupt is issued for the VoiceDSP processor to process the byte, or to prepare another byte for sending. In parallel, the VoiceDSP processor sets $\overline{\text{MWRDY}}$ to 1, to signal the microcontroller that it is busy with the byte processing. Another byte can be transferred only when the $\overline{\text{MWRDY}}$ signal is cleared to 0 by the VoiceDSP processor. When the VoiceDSP processor transmits data, it expects to receive the value 0xAA before each transmitted byte. The VoiceDSP processor reports any status change by clearing the $\overline{\text{MWRQST}}$ signal to 0.

If processor command's parameter is larger than one byte, the microcontroller transmits the Most Significant Byte (MSB) first. If a return value is larger than one byte, the VoiceDSP processor transmits the MSB first.

The following signals are used for the interface protocol. Input and output are relative to the VoiceDSP processor.

INPUT SIGNALS

MWDIN

MICROWIRE Data In. Used for input only, for transferring data from the microcontroller to the VoiceDSP processor.

MWCLK

MICROWIRE Clock. Serves as the synchronization clock during communication. One bit of data is transferred on every clock cycle. The input data is available on MWDIN and is latched on the clock rising edge. The transmitted data is output on MWDOUT on the clock falling edge. The signal should remain low when switching $\overline{\text{MWCS}}$.

$\overline{\text{MWCS}}$

MICROWIRE Chip Select. The $\overline{\text{MWCS}}$ signal is cleared to 0, to indicate that the VoiceDSP processor is being accessed. Setting $\overline{\text{MWCS}}$ to 1 causes the VoiceDSP processor to start driving MWDOUT with bit 7 of the transmitted value. Setting the $\overline{\text{MWCS}}$ signal resets the transfer-bit counter of the protocol, so the signal can be used to synchronize between the VoiceDSP processor and the microcontroller.

To prevent false detection of access to the VoiceDSP processor due to spikes on the MWCLK signal, use this chip select signal, and toggle the MWCLK input signal, only when the VoiceDSP processor is accessed.

OUTPUT SIGNALS

MWDOUT

MICROWIRE Data Out. Used for output only, for transferring data from the VoiceDSP processor to the microcontroller. When the VoiceDSP processor receives data it is echoed back to the microcontroller on this signal, unless the received data is 0xAA. In this case, the VoiceDSP processor echoes a command's return value.

$\overline{\text{MWRDY}}$

MICROWIRE Ready. When active (0), this signal indicates that the VoiceDSP processor is ready to transfer (receive or transmit) another byte of data.

This signal is set to 1 by the VoiceDSP processor after each byte transfer has been completed. It remains 1, while the VoiceDSP processor is busy reading the byte, writing the next byte, or executing the received command (after the last parameter has been received). $\overline{\text{MWRDY}}$ is cleared to 0 after reset. For proper operation after a hardware reset, this signal should be pulled up.

$\overline{\text{MWRQST}}$

MICROWIRE Request. When active (0), this signal indicates that new status information is available. $\overline{\text{MWRQST}}$ is deactivated (set to 1), after the VoiceDSP processor receives a GSW (Get Status Word) command from the microcontroller. After reset, this signal is active (0) to indicate that a reset occurred. $\overline{\text{MWRQST}}$, unlike all the signals of the communication protocol, is an asynchronous line that is controlled by the VoiceDSP firmware.

SIGNAL USE IN THE INTERFACE PROTOCOL

After reset, both $\overline{\text{MWRQST}}$ and $\overline{\text{MWRDY}}$ are cleared to 0.

The $\overline{\text{MWRQST}}$ signal is activated to indicate that a reset occurred. The EV_RESET bit in the status register is used to indicate a reset condition.

The GSW command should be issued after reset to verify that the EV_RESET event occurred, and to deactivate the $\overline{\text{MWRQST}}$ signal.

While the $\overline{\text{MWCS}}$ signal is active (0), the VoiceDSP processor reads data from MWDIN on every rising edge of MWCLK. VoiceDSP processor also writes every bit back to MWDOUT. This bit is either the same bit which was read from MWDIN (in this case it is written back as a synchronization echo after some propagation delay), or it is a bit of a value the VoiceDSP processor transmits to the microcontroller (in this case it is written on every falling edge of the clock).

When a command has more than one parameter/return-value, the parameters/return-values are transmitted in the order of appearance. If a parameter/return-value is more than one byte long, the bytes are transmitted from the most significant to the least significant.

The $\overline{\text{MWRDY}}$ signal is used as follows:

1. Active (0) $\overline{\text{MWRDY}}$ signals the microcontroller that the last eight bits of data transferred to/from the voice module were accepted and processed (see below).
2. The $\overline{\text{MWRDY}}$ signal is deactivated (set to 1 by the VoiceDSP processor) after 8-bits of data were transferred to/from the VoiceDSP processor. The bit is set following the falling edge of the eighth MWCLK clock-cycle.
3. The $\overline{\text{MWRDY}}$ signal is activated (cleared to 0) by the VoiceDSP processor when it is ready to receive the first parameter byte (if there are any parameters) and so on till the last byte of parameters is transferred. An active $\overline{\text{MWRDY}}$ signal after the last byte of parameters indicates that the command was parsed and (if possible) executed. If that command has a return value, the microcontroller must read the value before issuing a new command.
4. When a return value is transmitted, the $\overline{\text{MWRDY}}$ signal is deactivated after every byte, and activated again when the VoiceDSP processor is ready to send another byte, or to receive a new command.
5. The $\overline{\text{MWRDY}}$ signal is activated (cleared to 0) after reset, and after a protocol time-out. (See "INTERFACE PROTOCOL TIME-OUTS")

The $\overline{\text{MWRQST}}$ signal is used as follows:

1. The $\overline{\text{MWRQST}}$ signal is activated (cleared to 0), when the status word is changed.
2. The $\overline{\text{MWRQST}}$ signal remains active (0), until the VoiceDSP processor receives a GSW command.

Figure 1-32 and Figure 1-33 illustrate the sequence of activities during a MICROWIRE data transfer between VoiceDSP and the microcontroller.

INTERFACE PROTOCOL TIME-OUTS

Depending on the VoiceDSP processor's state, if more than 100 milliseconds elapse between the assertion of the $\overline{\text{MWRDY}}$ signal and the transmission 8th bit of the next byte pertaining to the same command transaction, a time-out event occurs, and the VoiceDSP processor responds as follows:

1. Sets the error bit in the status word to 1.
2. Sets the EV_TIMEOUT bit in the error word to 1.
3. Activates the $\overline{\text{MWRQST}}$ signal (clears it to 0).
4. Activates the $\overline{\text{MWRDY}}$ signal (clears it to 0).
5. Waits for a new command. (After a time-out occurs, i.e., the microcontroller received $\overline{\text{MWRQST}}$ during the command transfer, or result reception, the microcontroller must wait at least four milliseconds before issuing the next command.)

ECHO MECHANISM

The VoiceDSP processor echoes back to the microcontroller all the bits received by the VoiceDSP processor. Upon detection of an error in the echo, the microcontroller should stop the protocol clock, which eventually causes a time-out error (i.e., ERR_TIMEOUT bit is set in the error word).

NOTE When a command has a return value, the VoiceDSP processor transmits bytes of the return value instead of the echo value.

The VoiceDSP processor transmits a byte as an echo when it receives the value 0xAA from the microprocessor. Upon detection of an error the VoiceDSP processor activates the $\overline{\text{MWRQST}}$ signal, and sets the ERR_COMM bit in the error word.

2.2.2 MEMORY INTERFACE

DEVICE NUMBER AND TYPE

The VoiceDSP processor supports various types of Flash memory and ARAM/DRAM devices. Up to four devices may be connected to the VoiceDSP, where all the connected devices must be of the same type. Each memory device may be of 4Mbit, 8Mbit or 16Mbit; thus a total of 64Mbit non-volatile memory may be connected for message storage (up to 4 hours of voice recording).

See "MEMORY INTERFACE" on page 1-6, for detailed description of the supported Flash and ARAM/DRAM devices and the hardware connectivity.

Use the CFG command to define the type and number of installed memory devices (see "CFG Configure VoiceDSP config_value" on page 2-25).

MEMORY DEVICE SIZE

The memory manager handles the memory devices in basic units of 4Kbyts blocks. This approach is defined due to the nature of Flash devices where the basic unit that can be erased is a 4Kbytes block. This constraint is not relevant for ARAM/DRAM devices, but the concept is maintained for simplicity and consistency.

Memory blocks cannot be shared by different voice messages. Therefore, the maximum number of messages per memory device, equals to the number of memory blocks minus one (one block per device is used for memory management).

The size of the connected memory devices, is defined by the number of memory blocks in each device. Refer to tunable parameter index 62, in Table 2-10, for detailed description of the available number of blocks for Flash and ARAM/DRAM devices.

PRODUCTION LINE TESTING

In many cases it is desired to test the ISD-T360SB in the production line as part of the whole application. Usually in these cases, the testing time is an important factor and should be minimized as possible. The initialization time of the memory devices is significant and should be avoided during production (Refer to Table 1-4). Therefore, a dedicated parameter is defined in order to allow a production line testing while using a small part of the real connected memory size.

It should be noted that in case of power failure during the production line testing, the connected memory devices should be replaced, and the process should be repeated. Refer to parameter index 63, in Table 2-10, for further explanation of the production line testing.

ARAM QUALITY

ARAM is an Audio Grade RAM device, which implies that some percentage of the ARAM bits are defected and their content is undefined. Unlike Flash devices, where the defected bits can be mapped out, in case of ARAM specific bits cannot be mapped out; only memory blocks can be mapped out.

Therefore, it should be noted that using ARAM as a voice storage device, will result in audible distortions. It is the user responsibility to define the maximum allowed bad nibbles (4 bits) in a memory block. If the number of bad nibbles exceeds the defined limitation, the specific block is mapped out and is not used for message recording. Refer to tunable parameter index 64, in Table 2-10, for further details of the ARAM quality level definition.

2.2.3 CODEC INTERFACE

SUPPORTED FUNCTIONALITY

The VoiceDSP processor supports analog and digital telephony in various configurations. For analog telephony the VoiceDSP operates in master mode, where it provides the clock and the synchronization signals. It supports a list of single channel and dual channel codecs, as listed in Table 1-7. For digital telephony the VoiceDSP operates in slave mode, where the control signals are provided by an external source.

The codec interface is designed to exchange data in short frame format as well as in long frame format. The channel width may be either 8 bits (u-Law format or A-Law format), or 16 bits (linear format). In slave mode the clock may be divided by two, if required (two bit rate clock mode).

The VoiceDSP support up to 2 voice channels, where the line should be connected as channel 0 (in master mode or in slave mode - depends on the configuration), and the speakerphone (speaker and microphone) should be connected as channel 1 or as channel 2, depends on the configuration (channel 1 and channel 2 are always connected as master).

See "The Codec Interface" on page 1-13, for detailed description of the supported codec devices and the hardware connectivity.

Use the CFG command to define the codec mode (master or slave), the data frame format (short or long), the channel width (8 bits or 16 bits), the clock bit rate (single or dual) and the number and type of codecs (one or two, single channel or dual channel). See "CFG Configure VoiceDSP config_value" on page 2-25.

DATA CHANNELS TIMING

Especially in digital telephony, but also in analog telephony when speakerphone is connected, the channels data may be delayed from the synchronization signal by variable number of clock cycles. In order to allow full flexibility of the data delay relative to the synchronization signal, and the delay between the two synchronization signals, a set of registers is provided. setting the delay parameters of these registers defines the exact timing of all the codec interface signals.

Refer to tunable parameters index 65 to index 69, in Table 2-11, for detailed description of the delay registers and their significance.

2.3 ALGORITHM FEATURES

This section provides details of the VoiceDSP algorithms and their principle operation. It is divided into the following subjects:

- VCD (Voice Compression and Decompression)
- DTMF Detection
- Tone and Energy Detection (Call Progress)
- Speakerphone
- Speech Synthesis

2.3.1 VCD (VOICE COMPRESSION AND DECOMPRESSION)

The VoiceDSP processor implements a state of the art VCD algorithm of the CELP family. The algorithm provides 3 compression rates that can be selected dynamically (actually, the algorithm supports more compression rates). PCM recording (no compression) is also provided.

The lowest compression rate of 5.3 Kbit/s enables about 30 minutes of recording on an 8-Mbit device (depending on the relative silence period). The mid-quality compression rate of 9.9 Kbit/s provides about 16 minutes of voice recording time. The highest compression rate of 16.8 Kbit/s, the highest quality recording, stores up to 10 minutes on a 8-Mbit device. For detailed information about recording times refer to table 1-5.

Before recording each message, the microcontroller selects one of the three compression rates, or PCM recording, with the compression_rate parameter of the R (Record) command. During message playback the VoiceDSP processor reads this one byte parameter and selects the appropriate speech decompression algorithm.

IVS vocabularies can be prepared in either of the three compression rates, or in PCM format, using the IVS tool. All messages in a single vocabulary must be recorded using the same algorithm. (See the *IVS User's Guide* for more details). During speech synthesis, the VoiceDSP processor automatically selects the appropriate speech decompression algorithm.

SILENCE COMPRESSION

A Voice Activity Detector (VAD) is used in order to detect periods of silence during the compression of the recorded message. Silence is treated differently than normal voice by the compression algorithm. It is compressed to about 1.0 Kbit/s. The compressed silence contains data that allows to generate comfort noise during message playback. The comfort noise generation is important because the human ear is not used to "real" silence while listening to messages.

Various tunable parameters are available in order to optimally tune the VAD. The silence compression may be turned Off, though it is planned to remain On continuously. For more details refer to table 2-4 of the Command Description section.

NOTE *The silence compression should be turned Off when ARAM devices are used for voice storage. Otherwise, unpredictable results are expected during message playback.*

SW AGC

A SoftWare Automatic Gain Control (SW AGC) algorithm is activated with the compression module in order to regulate the input signal to a dynamic range that will provide higher compression quality. The algorithm senses the energy level and updates the signal gain in order to amplify low energy signals and to avoid signal saturation. The SW AGC feature eliminates the need for an external HW AGC, thus reducing hardware costs and complexity. Hardware Gain Control may be used to avoid signal saturation prior to sampling the signal.

A tunable parameter determines the maximum allowed gain for the SW AGC algorithm. The SW AGC may be turned Off, though it is planned to remain On continuously. For more details refer to table 2-4 of the Command Description section.

VARIABLE SPEED PLAYBACK

This feature increases or decreases the speed of messages and synthesized messages during playback. Use the SPS (Set Playback Speed) to set the speed of message playback. The new speed applies to all recorded messages and synthesized messages (only if synthesized using IVS), until changed by another SPS command. If this command is issued while the VoiceDSP processor is in the PLAY state, the speed also changes for the message currently being played.

The speedup / slowdown algorithm is designed to maintain the pitch of the original speech. This approach provides the same speech tone while playback speed varies.

PCM RECORDING

The VoiceDSP is capable of recording data in PCM format (that is the original samples format either in 8 bits u-Law format, 8 bit A-law format or 16 bits linear format). The PCM data uses more storage space, but it provides the highest quality for OGM, music-on-hold or IVS data. The PCM recording may be selected as one of the available compression rates during the R command (compression_rate = 0).

Silence Compression and variable Speed Playback are not feasible during PCM recording and playback since this feature skips the compression algorithm.

2.3.2 DTMF DETECTION

The VoiceDSP processor detects DTMF, which enables remote control operations. Detection is active throughout the operation of the VoiceDSP processor. Detection can be configured using the SDET (Set Detectors Mask) command, which controls the reporting of the occurrence of tones, and the RDET (Reset Detectors) command which resets the detectors. The accuracy of the tone length, as reported by the tone detectors, is ±10 ms.

DTMF detection may be reported at the starting point, ending point, or both. The report is made through the status word (for further details, see GSW command).

For further details about tunable parameters refer to table 2-6 of the Command description section.

The DTMF detector performance, as measured on the line input using an ISD-DS360-DAA board, is summarized below (see Table 2-1).

Table 2-1: DTMF Detector Performance[1]

	Play/IVS Synthesis	Record/Idle
Detection Sensitivity	Performance depends on the message being played.[2]	–34 dBm
Accepted DTMF Length[3]	>50 ms	>40 ms
Frequency Tolerance	±1.5%	±1.5%
S/N Ratio	12 dB	12 dB
Minimum Spacing[4]	>50 ms	>45 ms
Normal Twist	8 dB	8 dB
Reverse Twist[5]	4 dB or 8 dB	4 dB or 8 dB

1. *Performance depends on the DAA design. For reliable DTMF detection:*
   - A hardware echo-canceler, that attenuates the echo by at least 6 dBm, is required during playback.
   - The HW AGC, if present, must be disabled during playback.
2. *Performance with echo canceler is 10 dB better than without echo canceler. For a silent message, Detection sensitivity is –34 dBm, with echo canceler.*
3. *Tune parameters 60 and 61 may improve DTMF detection sensitivity. for more details refer to the parameters description in Table 2-6.*
4. *The accuracy of reported DTMF tones is ±10 ms.*
5. *If the interval between two consecutive identical DTMF tones is less than, or equal to, 20 ms, the two are detected as one long DTMF tone. If the interval between two consecutive identical DTMF tones is between 20 ms and 45 ms, separate detection is unpredictable.*
6. *Determined by the DTMF_REV_TWIST tunable parameter value.*

DTMF SW AGC

In order to remove the linkage between the HW AGC and the detection level of the DTMF detector, two new tunable parameters are added. These tunable parameters define the gain of the SW AGC for DTMF signals.

DTMF_DET_AGC_IDLE - SW AGC for DTMF detection in Idle and Record states. When incrementing this tunable by 1, the dynamic range is increased by 3 dB.

DTMF_DET_AGC_PLAY - SW AGC for DTMF detection in Play and Tone_Generate states. When incrementing this tunable by 1, the dynamic range is increased by 3 dB.

ECHO CANCELLATION

Echo cancellation is a technique used to improve the performance of DTMF detection during speech synthesis, tone generation, and OGM playback. For echo cancellation to work properly, HW AGC must not be active in parallel. Thus, to take advantage of echo cancellation, the microcontroller must control the HW AGC, if exists, (i.e., disable the HW AGC during PLAY, SYNTHESIS and TONE_GENERATE states and enable it again afterwards). If HW AGC can not be disabled, do not use echo cancellation.

The microcontroller should use the CFG command to activate/deactivate echo cancellation.

NOTE Normally, a HW AGC is not required with The ISD-T360SB, since SW AGC is active for the VCD algorithm, DTMF detection and the speakerphone module.

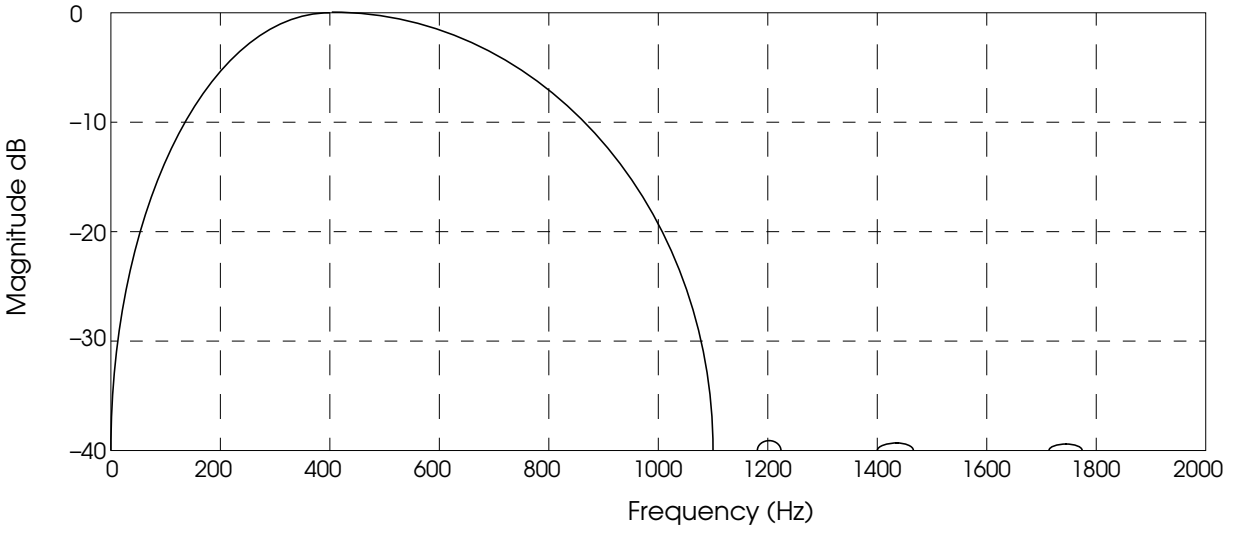
2.3.3 TONE AND ENERGY DETECTION (CALL PROGRESS)

The VoiceDSP processor detects busy and dial tones, constant energy level, and no-energy (VOX). This enables call progress tracking. Detection is active throughout the operation of the VoiceDSP processor. Detection can be configured using the SDET (Set Detectors Mask) command, which controls the reporting of the occurrence of tones, and the RDET (Reset Detectors) command which resets the detectors. The accuracy of the tone length, as reported by the tone detectors, is ±10 ms.

TUNABLE PARAMETERS

Tunable parameters control the detection of busy and dial tones, constant energy level (in the frequency range 200–3400Hz), and no-energy. These parameters should be tuned to fit the system hardware. In addition, changes may be required to the tunable parameters according to the setting (On or Off) of the HW Automatic Gain Control (HW AGC), if exists. For more information refer to tables 2-7, 2-8 of the Command Description section.

Figure 2-1: Busy and Dial-Tone Band-Pass Filter Frequency Response



BUSY AND DIAL TONES

Busy and dial-tone detectors work with a band-pass filter that limits the frequency range in which tones can be detected to 0–1100Hz. Figure 1-1 shows the frequency response of this band-pass filter.

The design of the busy-tone detector allows very high flexibility in detecting busy tones with varying cadences.

The tunable parameters are divided into five sets:

1. Busy Tone On-time and Off-time Range Specification:

   BUSY_DET_MIN_ON_TIME
   BUSY_DET_MIN_OFF_TIME
   BUSY_DET_MAX_ON_TIME
   BUSY_DET_MAX_OFF_TIME

2. Busy Tone Cadence Control Specification

   BUSY_DET_VERIFY_COUNT
   BUSY_DET_TONE_TYPE
   BUSY_DET_DIFF_THRESHOLD

   BUSY_DET_VERIFY_COUNT determines the number of On/Off cadences that detector should detect before reporting busy tone presence.

   BUSY_DET_DIFF_THRESHOLD describes the maximum allowed difference between two compared On or Off periods, as determined by the BUSY_DET_TONE_TYPE tunable parameter.

   BUSY_DET_TONE_TYPE specifies the type of cadences that are supported.

   Legal values are:

   Two cadences only
   Three cadences only
   Both two and three cadences.

   The acceptance criteria for two cadences:

   (E1–E3) < BUSY_DET_DIFF_THRESHOLD
   and
   (S1–S3) < BUSY_DET_DIFF_THRESHOLD

   The acceptance criteria for three cadences:

   (E1–E4) < BUSY_DET_DIFF_THRESHOLD
   and
   (S1–S4) < BUSY_DET_DIFF_THRESHOLD

3. Busy and Dial Tone Energy Thresholds

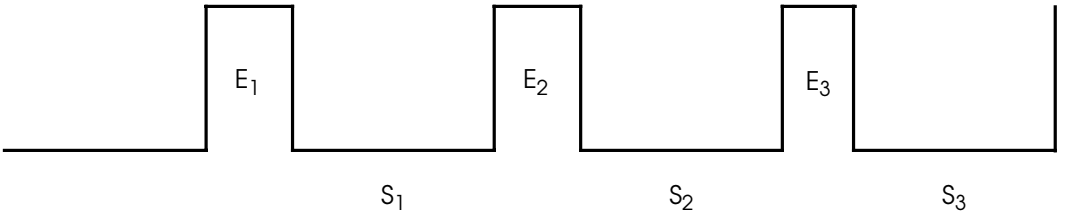
   TONE_DET_ON_ENERGY THRESHOLD

   TONE_DET_OFF_ENERGY THRESHOLD

4. Busy Detection Time

   BUSY_DET_MIN_TIME

Figure 2-2: Busy-Tone Detector—Default Cadence Specification



$(E_1 - E_3) < 100$ ms $(S_1 - S_3) < 100$ ms $100 < E_i < 1680$ ms $70 < S_i < 1220$ ms

CONSTANT ENERGY

The constant-energy detector reports the presence of constant energy in the range 200Hz to 3400Hz. It is intended to detect both white and pink noise and can be used to detect line disconnection during recording.

It is recommend to use the constant energy mechanism in conjunction with the no-energy (VOX) mechanism.

The following tunable parameters control the operation of the constant-energy detector:

CONST_NRG_DET_TIME_COUNT
CONST_NRG_DET_TOLERANCE_TIME
CONST_NRG_DET_LOW_THRESHOLD
CONST_NRG_DET_HIGH_THRESHOLD

NO ENERGY (VOX)

The no-energy detector reports when the energy in the frequency range 200Hz to 3400Hz remains below a preprogrammed threshold for a preprogrammed time-out. A programmable tolerance is allowed.

It is recommend to use the no-energy (VOX) mechanism in conjunction with the constant-energy mechanism.

The following tunable parameters control the operation of the no-energy (VOX) mechanism:

VOX_DET_ENERGY_THRESHOLD
VOX_DET_TIME_COUNT
VOX_DET_TOLERANCE_TIME

2.3.4 FULL-DUPLEX SPEAKERPHONE

The speakerphone feature lets the user communicate through a telephone line, using the unit's speaker and the microphone instead of its handset. The speakerphone processes signals sent from the line to the speaker, and from the microphone to the line. It also performs the necessary switching, attenuation and echo cancellation on the signals present on the line/speaker.

The ISD-T360SB speakerphone is simple to use; it requires no special hardware or training for the echo cancelers. The gain control is fully digital, which eliminates the need for analog gain control hardware.

The speakerphone features two types of echoes, the electrical echo (line or circuit) and the acoustic echo. The electrical echo is a result of an imperfect impedance match between the 4- to 2-wire interface (hybrid) and the line impedance. The electrical echo, relatively short term, has a transfer function that varies slowly. The second echo, the acoustic echo, is a line impedance returning from the speaker to the microphone. This echo is relatively long term, and its transfer function may vary quite quickly if anyone, or anything, moves in the room. Both echoes must be canceled to achieve a high-quality hands-free system.

For more details of the speakerphone tunable parameters refer to table 2-9 of the Command Description section.

SPEAKERPHONE TERMINOLOGY

Send Path

The signal path from the microphone (near-end speaker) to the line (far-end listener). The microphone is the input port, and line-out is the output port of this signal path.

Receive Path

The signal path from the line (far-end speaker) to the loudspeaker (near-end listener). The line-in is the input port, and the speaker is the output port for this signal path.

AEC

Acoustic Echo Controller. The part in the speakerphone algorithm that controls the echo in the sendpath.

EEC

Electric Echo Controller. The part in the speakerphone algorithm that controls the echo in the receive path.

SPEAKERPHONE MODES OF OPERATION

Full-Duplex (ON)

The speakerphone works in full-duplex mode, meaning both parties can speak and hear each other simultaneously. In this mode both the acoustic and electric echo controllers are active. The VoiceDSP processor tone detectors are not active in this mode.

Mute

In this mode of operation, the speakerphone generates silence to the line. The near-end listener can hear the far-end speaker but not vice versa. Tone detectors are not active.

Hold

During Hold mode interrupts from both codecs are stopped. Neither side can hear each other.

Restart

In Restart mode the speakerphone re-initializes itself to the last speakerphone mode (full-duplex, transparent or mute). This mode should be used to resume the speakerphone operation after Hold mode or when there is a significant change in the environmental conditions (e.g., parallel pickup) that may affect the speakerphone quality.

Transparent

While in Transparent mode, the speakerphone works in full-duplex mode but without echo cancellation.

Samples from the microphone are transferred to the line, and samples from the line are transferred to the speaker, with no processing. This mode should be used only for tuning and testing the system.

Listen

In Listen mode the line is audible on the speaker, and the processor tone detectors are active.

During Listen mode, dialing with the GT command and call progress is possible, since the busy and dial tone detectors are active.

The following pseudo-code demonstrates how to make a call from speakerphone mode:

Figure 2-3: Speakerphone Pseudo Code Representation

```
while () {
    EV = wait_event()
    case EV of:
        skpr_button_pressed:
            if (speakerphone_on) {
                SSM 0   // Put VoiceDSP in idle mode
                first_digit = TRUE
                deactivate_digit_timeout_event()
            else
                SSM 1   // Put VoiceDSP in full-duplex speakerphone mode
        digit_pressed:
            if (first_digit) {
                SSM 4       // Enter LISTEN mode
            first_digit = FALSE
            }
            GT <dtmf_of_digit> // Dial the digit
            S           // Stop. Note that after the S command
                        // the VoiceDSP is still in speakerphone mode
            enable_digit_timeout_event() // To "guess" when dialing is completed.
        digit_timeout_event:
            SSM 1       // Dialing is completed, Go back to full-duplex mode
            deactivate_digit_timeout_event()
}
```

2.3.5 SPEECH SYNTHESIS

Speech synthesis is the technology used to create messages out of predefined words and phrases stored in a vocabulary.

There are two kinds of predefined messages: fixed messages (voice menus in a voice-mail system) and programmable messages (time-and-day stamp, or the *You have n messages* announcement in a DTAD).

A vocabulary includes a set of predefined words and phrases, needed to synthesize messages in any language. Applications which support more than one language require a separate vocabulary for each language.

INTERNATIONAL VOCABULARY SUPPORT (IVS)

IVS is a mechanism by which the VoiceDSP processor utilizes several vocabularies stored on an external storage device. IVS enables the ISD-T360SB to synthesize messages with the same meaning, but in different languages, from separate vocabularies.

IVS Features

- Multiple vocabularies stored on a single storage device.
- Plug-and-play. The same microcontroller code is used for all languages.
- Synthesized and recorded messages use the same voice compression algorithm to achieve equal quality.
- Argumented sentences. (For example: *You have <n> messages.*)

- Auto-synthesized time-and-day stamp (driven by the VoiceDSP processor's clock).
- Support for various language and sentence structures:
  - One versus many. (For example: You *have one message* versus *You have two messages*.)
  - None versus many. (For example: *You have no messages* versus *You have two messages*.)
  - Number synthesis (English—*Eighty* versus French—*Quatre-vingt*).
  - Word order (English—*Twenty-one* versus German—*Einundzwanzig*).
  - Days of the week (Monday through Sunday versus Sunday through Saturday).

VOCABULARY DESIGN

There are several issues, sometimes conflicting, which must be addressed when designing a vocabulary.

Vocabulary Content

If memory space is not an issue, the vocabulary could contain all the required sentences, each recorded separately.

If memory space is a concern, the vocabulary must be compact; it should contain the minimum set of words and phrases required to synthesize all the sentences. The least memory is used when phrases and words that are common to more than one sentence are recorded only once, and the IVS tool is used to synthesize sentences out of them.

A good combination of sentence quality and memory space is achieved if you take the "compact" approach, and extend it to solve pronunciation problems. For example, the word twenty is pronounced differently when used in the sentences *You have twenty messages* and *You have twenty-two messages*. To solve this problem, words that are pronounced differently should be recorded more than once, each in the correct pronunciation.

Vocabulary Recording

When recording vocabulary words, there is a compromise between space and quality. On one hand, the words should be recorded and saved in a compressed form, and you would like to use the best voice compression for that purpose. On the other hand, the higher the compression rate, the worse the voice quality.

Another issue to consider is the difference in voice quality between synthesized and recorded messages (e.g., between time-and-day stamp and ICMs in a DTAD environment). It is more pleasant to the human ear to hear both messages have the same sound quality.

Vocabulary Access

Sometimes compactness and high quality are not enough. There should be a simple and flexible interface to access the vocabulary elements. Not just the vocabulary but the code to access the vocabulary should be compact.

When designing for a multi-lingual environment, there are even more issues to consider. Each vocabulary should be able to handle language-specific structures and designed in a cooperative way with the other vocabularies so that the code to access each vocabulary is the same. When you use the command to synthesize the sentence *Monday. 12:30 P.M.,* you should not care in what language it is going to be played back.

IVS VOCABULARY COMPONENTS

This section describes the basic concept of an IVS vocabulary, its components, and the relationships between them.

Basic Concepts

An IVS vocabulary consists of words, sentences, and special codes that control the behavior of the algorithm which VoiceDSP processor uses to synthesize sentences.

Word Table

The words are the basic units in the vocabulary. Create synthesized sentences by combining words in the vocabulary. Each word in the vocabulary is given an index which identifies it in the word table.

Note that, depending on the language structures and sentences synthesized, you may need to record some words more than once in the vocabulary. For example, if you synthesize the sentences: *you have twenty messages* and *you have twenty-five messages*, the word *twenty* is pronounced differently. They should, therefore, be defined as two different words.

Number Tables

The number tables allow you to treat numbers differently depending on the context.

Example 1: The number 1 can be announced as *one* as in *message number one* or as first as in *first message*.

Example 2: The number 0 can be announced as *no* as in *you have no messages* or as *oh* as in *monday, eight oh five* A.M.

A separate number table is required for each particular type of use. The number table contains the indices of the words in the vocabulary that are used to synthesize the number. Up to nine number tables can be included in a vocabulary.

Sentence Table

The sentence table describes the predefined sentences in the vocabulary. The purpose of this table is to make the microcontroller that drives the VoiceDSP processor independent of the language being synthesized. For example, if the Flash and/or ROM memory contain vocabularies in various languages, and the first sentence in each vocabulary means you have n messages, the microcontroller switches languages by issuing the following command to VoiceDSP processor:

```
SV <storage_media>, <vocabulary_id> -Select
a new vocabulary
```

The microcontroller software is thus independent of the grammar of the language in use. The sentences consist of words, which are represented by their indices in the vocabulary.

Sentence 0

All sentences but one are user defined. The VoiceDSP processor treats the first sentence in the sentence table (sentence 0) specially, to support time-and-day stamp. The processor assumes that the sentence is designed for both system time, and message time-and-day stamp announcement, and has one argument which is interpreted as follows:

0	System time is announced
1	The time-and-day stamp of the current message is announced.

Example 1: When the microcontroller sends the command: SAS 0, 0

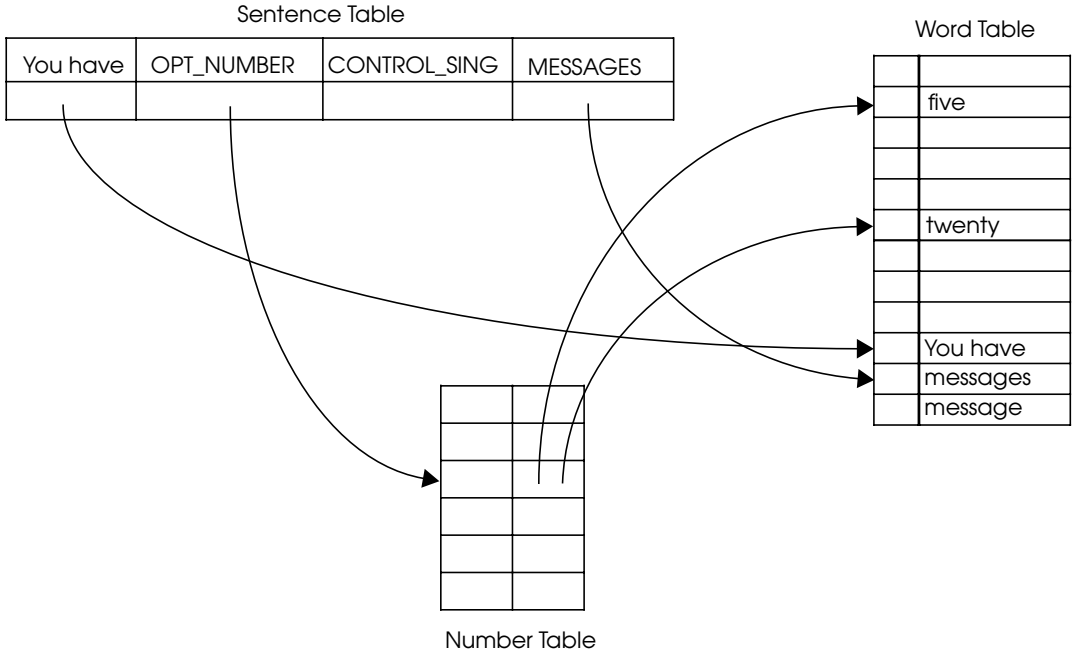
The system time and day is announced.

Example 2: When the microcontroller sends the command: SAS 0, 1

The current message time-and-day stamp is announced.

The following Figure 2-4 shows the interrelationship between the three types of tables.

Figure 2-4: The Interrelationship between the Word, the Number, and the Sentence Tables



Control and Option Codes

The list of word indices alone cannot provide the entire range of sentences that the VoiceDSP processor is able to synthesize. IVS control and option codes send special instructions to control the speech synthesis algorithm's behavior in the processor.

For example, if the sentence should announce the time of day, the VoiceDSP processor should be able to substitute the current day and time in the sentence. These control words do not represent recorded words, rather they instruct the processor to take special actions.

THE IVS TOOL

The IVS tool includes two utilities:

1. The DOS-based IVS Compiler
2. IVSTOOL for Windows. A Windows 3.1/95 based utility

The tools help create vocabularies for the VoiceDSP processor. They take you from designing the vocabulary structure, through defining the vocabulary sentences, to recording the vocabulary words.

IVS Compiler

The IVS compiler runs on MS-DOS (version 5.0 or later) and enables you to insert your own vocabulary, (i.e., basic words and data used to create numbers and sentences, as directories and files in MS-DOS). The IVS compiler then outputs a binary file containing that vocabulary. In turn, this information can be burned into an EPROM or Flash memory to be used by the VoiceDSP software.

NOTE The IVS data cannot be stored in EPROM when semi-parallel Flash is used (Samsung or Toshiba).

IVS Voice Compression

Each IVS vocabulary can be compiled with either the 5.3 Kbit/s, the 9.9 Kbit/s or the 16.8Kbit/s voice compression algorithm, or in PCM format. Define the compression rate before compilation. The VoiceDSP processor automatically selects the required voice decompression algorithm when the SV command chooses the active vocabulary.

Graphical User Interface (GUI)

The IVS package includes a Windows utility to assist the vocabulary designer to synthesize sentences. With this utility, you can both compose sentences and listen to them.

HOW TO USE THE IVS TOOL WITH THE VOICEDSP PROCESSOR

The IVS tool creates IVS vocabularies, and stores them as a binary file. This file is burnt into a ROM device or programmed into a Flash memory device using the INJ (Inject IVS) command. The VoiceDSP processor SO (Say One Word) command is used to select the required vocabulary. The SW (Say Words), SO, SS (Say Sentence) and SAS (Say Argumented Sentence) commands are used to synthesize the required word or sentence. The typical vocabulary-creation process is as follows:

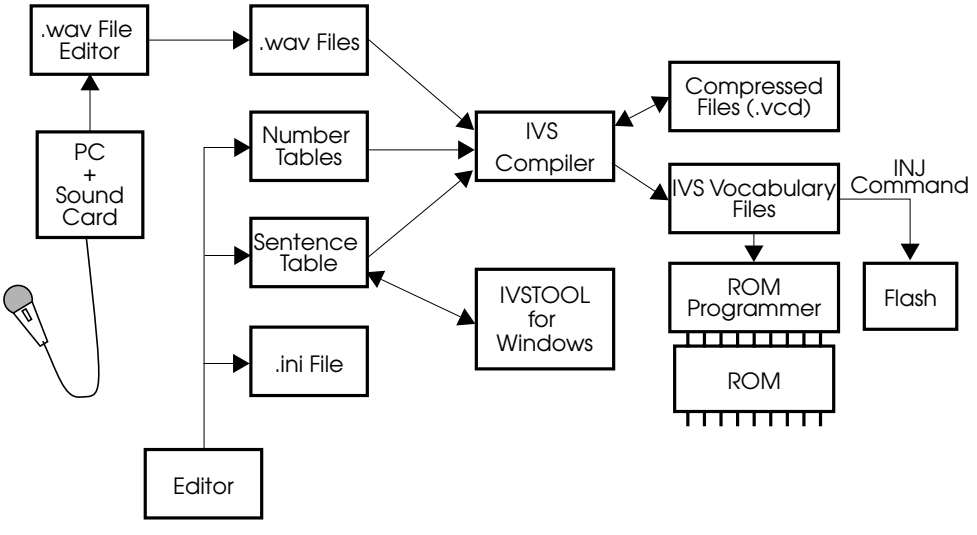
1. Design the vocabulary.
2. Create the vocabulary files (as described in detail below). Use VISTULA for Windows 3.1/95 to simplify this process.
3. Record the words using any standard PC sound card and sound editing software, that can create .wav files.
4. Run the IVS compiler to compress the .wav files, and compile them and the vocabulary tables into an IVS vocabulary file.
5. Repeat steps 1 to 4 to create a separate IVS vocabulary for each language that you want to use.
6. Burn the IVS vocabulary files into a ROM (or Flash memory) device. Use the INJ (Inject IVS) command to program the data into a Flash device.

NOTE *The IVS data cannot be stored in EPROM when semi-parallel Flash is used (Samsung or Toshiba).*

Once the vocabulary is in place, the speech synthesis commands of the VoiceDSP processor can be used to synthesize sentences.

Figure 2-5 shows the vocabulary-creation process for a single table on a ROM or Flash memory device.

Figure 2-5: Creation of an IVS Vocabulary



2.4 VOICEDSP PROCESSOR COMMANDS—QUICK REFERENCE TABLE

Table 2-2: Speech Commands

Command		Description	Opcode Hex	Source State	Result State	Command Parameters		Return Value	
Name	S/A					Description	Bytes	Description	Bytes
CCIO	S	Configure Codec I/O	34	RESET, IDLE	No change	Config_value	1	None	-
CFG	S	Configure VoiceDSP	01	RESET	No change	Config_value	3	None	-
CMSG	S	Create Message	33	IDLE	MSG_OPEN	Tag, Num_of_ blocks	2+2	None	-
CMT	S	Cut Message Tail	26	IDLE	No change	Time_length	2	None	-
CVOC	S	Check Vocabulary	2B	IDLE	No change	None	-	Test result	1
DM	S	Delete Message	0A	IDLE	No change	None	-	None	-
DMS	S	Delete Messages	0B	IDLE	No change	Tag_ref, Tag_mask	2 + 2	None	-
GCFG	S	Get Configuration Value	02	RESET, IDLE	No change	None	-	Version	1
GEW	S	Get Error Word	1B	All states	No change	None	-	Error word	2
GI	S	Get Information item	25	PLAY, RECORD, SYNTHESIS, TONE_ GENERATE, IDLE	No change	Item	1	Item value	2
GL	S	Get Length	19	IDLE	No change	None	-	Message length	2
GMS	S	Get Memory Status	12	IDLE	No change	None	-	Recording time left	2
GMT	S	Get Message Tag	04	IDLE	No change	None	-	Message tag	2
GNM	S	Get Number of Messages	11	IDLE	No change	Tag_ref, Tag_mask	2 + 2	Number of messages	2
GSW	S	Get Status Word	14	All states	No change	None	-	Status word	2
GT	A	Generate Tone	0D	IDLE	TONE_ GENERATE	Tone (single Tone or DTMF)	1	None	-
GTD	S	Get Time and Day	0E	IDLE	No change	Time_day_ option	1	Time and day	2
GTM	S	Get Tagged Message	09	IDLE	No change	Tag_ref, Tag_mask,Dir	2+2+1	Message found	1
GTUNE	S	Get Tunable Parameter	06	IDLE, RESET	No change	Index	1	Parameter_ value	2
INIT	S	Initialize System	13	RESET, IDLE	IDLE	None	-	None	-
INJ	S	Inject IVS data	29	IDLE	No change	N, $byte_1$...$byte_n$	4 + n	None	-
MR	S	Memory Reset	2A	RESET, IDLE	No change	None	-	None	-
P	A	Playback	03	IDLE	PLAY	None	-	None	-

Table 2-2: Speech Commands (Continued)

Command		Description	Opcode Hex	Source State	Result State	Command Parameters		Return Value	
Name	S/A					Description	Bytes	Description	Bytes
PA	S	Pause	1C	PLAY,RECORD, SYNTHESIS, TONE_GENERATE, IDLE*	No change	None	-	None	-
PDM	S	Go To Power-Down Mode	1A	IDLE	No change	None	-	None	-
R	A	Record Message	0C	IDLE	RECORD	Tag (message Tag), Compression_ rate	2 + 1	None	-
RDET	S	Reset Detectors	2C	IDLE	No change	Detectors_ reset_mask	1	None	-
RES	S	Resume	1D	PLAY,RECORD, SYNTHESIS, TONE_GENERATE IDLE*	No change	None	-	None	-
RMSG	S	Read Message	32	IDLE, MSG_OPEN	MSG_OPEN	None	-	Data	32
S	S	Stop	00	All states but RESET	IDLE	None	-	None	-
SAS	A	Say Argumented Sentence	1E	IDLE	SYNTHESIS	Sentence_n, Arg	1 + 1	None	-
SB	S	Skip Backward	23	PLAY, IDLE*	No change	Time_length	2	None	-
SDET	S	Set Detectors Mask	10	IDLE	No change	Detectors_ mask	1	None	-
SE	S	Skip to End of Message	24	PLAY, IDLE*	No change	None	-	None	-
SETD	S	Set Time and Day	0F	IDLE	No change	Time_and_ day	2	None	-
SF	S	Skip Forward	22	PLAY, IDLE*	No change	Time_length	2	None	-
SMSG	S	Set Message Pointer	30	IDLE, MSG_OPEN	MSG_OPEN	Num_of_ pages	2	None	-
SMT	S	Set Message Tag	05	IDLE	No change	Message_tag	2	None	-
SO	A	Say One Word	07	IDLE	SYNTHESIS	Word_number	1	None	-
SPS	S	Set Playback Speed	16	PLAY, SYNTHESIS, IDLE	No change	Speed	1	None	-
SS	A	Say Sentence	1F	IDLE	SYNTHESIS	Sentence_n	1	None	-

Table 2-2: Speech Commands (Continued)

Command		Description	Opcode Hex	Source State	Result State	Command Parameters		Return Value	
Name	S/A					Description	Bytes	Description	Bytes
SV	S	Set Vocabulary Type	20	IDLE	No change	Type, Id	1 + 1	None	-
SW	A	Say Words	21	IDLE	SYNTHESIS	N, $word_1$...$word_n$	1 + N	None	-
TUNE	S	Tune Parameters	15	IDLE, RESET	No change	Index, Parameter_ value	1 + 2	None	-
VC	S	Volume Control	28	PLAY, SYNTHESIS, IDLE, TONE_GENERATE	No change	Vol_level (increment/ decrement)	1	None	-
WMSG	S	Write Message	31	IDLE, MSG_OPEN	MSG_OPEN	Data	32	None	-

NOTE: ** Command is valid in IDLE state, but has no effect.*
S = Synchronous command
A= Asynchronous command

Table 2-3: Speakerphone Commands

Command		Description	Opcode Hex	Source State	Result State	Command Parameters		Return Value	
Name	S/A					Description	Bytes	Description	Bytes
GEW	S	Get Error Word	1B	TONE_GENERATE, IDLE	No change	None	-	Error word	2
GSW	S	Get Status Word	14	TONE_GENERATE, IDLE	No change	None	-	Status word	2
GT	A	Generate Tone	0D	IDLE	TONE_GENERATE	Tone (Single Tone or DTMF)	1	None	-
RDET	S	Reset Detectors	2C	IDLE	No change	Detectors_reset_mask	1	None	-
S	S	Stop	00	TONE_GENERATE, IDLE	IDLE	None	-	None	-
SDET	S	Set Detectors Mask	10	IDLE	No change	Detectors_ mask	1	None	-
SSM	S	Set Speakerphone Mode	2F	IDLE	No change	Mode	1	None	-
VC	S	Volume Control	28	TONE_GENERATE, IDLE	No change	Vol_level (increment/ decrement)	1	None	-

NOTE: ** Command is valid in IDLE state, but has no effect.*
S = Synchronous command
A = Asynchronous command

2.5 COMMAND DESCRIPTION

The commands are listed in alphabetical order.

The execution time for all commands, when specified, includes the time required for the microcontroller to retrieve the return value, where appropriate.

The execution time does not include the protocol timing overhead, i.e., the execution times are measured from the moment that the command is detected as valid until the command is fully executed.

NOTE: Each command description includes an example application of the command. The examples show the opcode issued by the microcontroller and the response returned by the VoiceDSP processor. For commands which require a return value from the processor, the start of the return value is indicated by a thick vertical line.

CCIO Configure Codec I/O *config_value*

Configures the voice sample paths in various states. It should be used to change the default VoiceDSP processor configuration. It is relevant only when two codecs are used and speakerphonr is connected.

The *config_value* parameter is encoded as follows:

Bit 0

Loopback control.

0 Loopback disabled (default).

1 Loopback enabled. In the RECORD state, the input samples are echoed back, unchanged (i.e., no volume control), to the same codec.

Bit 1

Codec input control.

0 Input is received via the line codec (default).

1 Input is received via the speakerphone codec.

Bits 2–3

Codec output control.

00 In PLAY, IDLE, SYNTHESIS and TONE_GENERATE states, output is to both codecs. In RECORD state, output is to the non-input codec (no volume control). If the loopback control bit is set, output in RECORD state is to both codecs as well (default).

01 Output in all states is to the line codec.

10 Output in all states is to the speakerphone codec.

11 Output in all states is to both codecs.

Bits 4–7

Reserved.

Example

CCIO 01			
Byte sequence:	Microcontroller	34	01
	VoiceDSP	34	01
Description:	Enable loopback		

CFG Configure VoiceDSP *config_value*

Configures the VoiceDSP processor in various hardware environments. It should be used to change the default processor configuration.

The *config_value* parameter is encoded as follows:

Bits 0–3

Memory type.

0000 No Memory (default).

0001 A/DRAM.

0010 Reserved.

0011 Toshiba Serial Flash.

0100 Samsung Semi-Parallel Flash.

0101 Toshiba Semi-Parallel Flash.

0110 Reserved.

0111 Reserved.

Bits 4–5

Number of installed memory devices.

00 1 (Default)

01 2

10 3

11 4

Bit 6-14

Reserved.

Bit 15

Echo Cancellation Control (for DTMF Detection).

0 Echo cancellation off (default).

1 Echo cancellation is on during playback.

Echo cancellation improves the performance of DTMF detection during playback. Echo cancellation can be turned on only with a system that can disable HW AGC (if present) during playback. A system featuring HW AGC, that cannot be controlled by the microcontroller (i.e., disabled or enabled), must not turn on this bit.

Bit 16

Clock bit rate (in Slave Mode only).

0 One bit rate clock (default).

1 Two bit rate clock.

Bit 17

Codec configuration.

0 Short-frame format (default).

1 Long-frame format (guaranteed by design but not tested).

Bits 18–19

Codec type.

00 16-bit linear (default).

01 μ-Law.

10 A-Law.

Bit 20

Codec interface mode.

0 Master codec interface (default).

1 Slave codec interface.

Bits 21–22

Number and type of codecs

00 One single codec (default).

01 Two single codecs.

10 One dual codec.

11 Reserved.

The codecs should be connected as follows:

Telephone line or equivalent - always connected as channel 0.

Speaker and microphone - connected as channel 1 in case of one dual codec (not applicable in slave mode), connected as channel 2 in case of two single codecs.

Bit 23

Reserved.

Example

CFG 144013					
Byte sequence:	Microcontroller	01	14	40	13
	VoiceDSP	01	14	40	13
Description:	Configure the VoiceDSP to work with: Single codec in Slave Mode and A-Law compressed samples. Data in Short Frame format and Single Bit Rate interface. Two Serial Toshiba Flash devices. Echo Cancellation for DTMF detectiopn is On.				

CMSG Create Message *tag num_of_blocks*

Creates a new message with a message tag *tag*, allocates *num_of_blocks* 4-Kbytes blocks for the new message, and sets the message pointer to the beginning of the message data. CMSG switches the VoiceDSP processor to the MSG_OPEN state.

The memory space available for the message data is computed as follows:

(127 x *num_of_blocks* – 2) x 32 bytes.

Once a message is open (the processor is in the MSG_OPEN state), the message pointer can be set to any position on a page (32 bytes) boundary within the message with the SMSG command. Modify the message contents with the WMSG command, and read with the RMSG command.

The microcontroller must issue an S command to close the message and switch the VoiceDSP processor to the IDLE state.

If the memory is full, EV_MEMFULL is set in the status word and no message is created. If the memory is not full but there is insufficient memory and the processor cannot allocate more memory space, EV_MEMLOW is set in the status word and no message is created.

Example

CMSG 0101 0001						
Byte sequence:	Microcontroller	33	01	01	00	01
	VoiceDSP	33	01	01	00	01
Description:	Create a new message with a tag=0101, and allocate 1 block (4 Kbytes) for its data.					

CMT Cut Message Tail *time_length*

Cut *time_length* units, in 10 ms segments, off the end of the current message. The maximum value of *time_length* is 6550. In case of silence, cut-time accuracy is 0.1 to 0.2 seconds (depends on compression rate).

NOTE *If time_length is longer than the total duration of the message, the EV_NORMAL_END event is set in the status word and the message becomes empty but not deleted. Use the DM (Delete Message), or DMS (Delete Messages), command to delete the message.*

A compressed frame represents 21 ms of speech, thus the minimum meaningful parameter is 3, (i.e., a 30 ms cut).

CMT 1 or CMT 2 have no effect.The CMT command can not be used on data messages

Example

CMT 02BC				
Byte sequence:	Microcontroller	26	02	BC
	VoiceDSP	26	02	BC
Description:	Cut the last seven seconds of the current message.			

CVOC Check Vocabulary

Checks (checksum) if the IVS data was correctly programmed to the ROM or Flash device.

If the vocabulary data is correct the return value is 1. Otherwise the return value is 0. If the current vocabulary is undefined, ERR_INVALID is reported.

Example

CVOC			
Byte sequence:	Microcontroller	2B	AA
	VoiceDSP	2B	01
Description:	Check the current vocabulary. The VoiceDSP processor responds that the vocabulary is OK.		

DM Delete Message

Deletes the current message. Deleting a message clears its message tag.

Deleting the current message does not cause a different message to become current. The current message is undefined. If, for example, you issue the GTM command to skip to the next message, the first message that is newer than the just deleted message is selected as the current message.

The memory space released by the deleted message is immediately available for recording new messages.

Example

DM		
Byte sequence:	Microcontroller	0A
	VoiceDSP	0A
Description:	Delete current message.	

DMS Delete Messages *tag_ref tag_mask*

Deletes all messages whose message tags match the tag_ref parameter. Only bits set in tag_mask are compared i.e., a match is considered successful if:

message tag and *tag_mask* = *tag_ref* and *tag_mask*

where and is a bitwise AND operation.

After the command completes execution, the current message is undefined. Use the GTM command to select a message to be the current message.

The memory space released by the deleted message is immediately available for recording new messages.

Example

DMS FFC2 003F						
Byte sequence:	Microcontroller	0B	FF	C2	00	3F
	VoiceDSP	0B	FF	C2	00	3F
Description:	Delete all old incoming messages from mailbox Number 2 in a system where the message tag is encoded as follows: Bits 0–2: mailbox ID 8 mailboxes indexed: 0 to 7 Bit 3: new/old message indicator 0—Message is old 1—Message is new Bits 4–5: message type 00—ICM/memo 01—OGM 10—Call transfer message Bits 6–15: not used *Note: the description of the tag is an example only. All bits of the tag are user-definable.*					

GCFG Get Configuration Value

Returns a sequence of one byte with the following information:

Bits 0–7

Magic number, which specifies the VoiceDSP firmware version.

Example

GCFG			
Byte sequence:	Microcontroller	02	AA
	VoiceDSP	02	01
Description:	Get the VoiceDSP processor magic number. The VoiceDSP processor responds that it is Version 1.		

GEW Get Error Word

Returns the 2-byte error word.

ERROR WORD

The 16-bit error word indicates errors that occurred during execution of the last command. If an error is detected, the command is not processed; the EV_ERROR bit in the status word is set to 1, and the $\overline{\text{MWRQST}}$ signal is activated (set to 0).

The GEW command reads the error word. The error word is cleared during reset and after execution of the GEW command.

If errors ERR_COMMAND or ERR_PARAM occur during the execution of a command that has a return value, the return value is undefined. The microcontroller must still read the return value, to ensure proper synchronization.

15 9	8	7	6	5	4	3	2	1	0
Res	Res	ERR_ INVALID	ERR_ TIMEOUT	ERR_ COMM	Res	ERR_ PARAM	ERR_ COMMAND	ERR_ OPCODE	Res

The bits of the error word are used as follows:

ERR_OPCODE

Illegal opcode. The VoiceDSP processor does not recognize the opcode.

ERR_COMMAND

Illegal command sequence. The command is not legal in the current state.

ERR_PARAM

Illegal parameter. The value of the parameter is out of range, or is not appropriate for the command.

ERR_COMM

Microcontroller MICROWIRE communication error.

ERR_TIMEOUT

Time-out error. Depending on the VoiceDSP processor's state, more than 100 milliseconds elapsed between the arrival of two consecutive bytes (for commands that have parameters).

ERR_INVALID

Command can not be performed in current context.

Example

GEW					
Byte sequence:	Microcontroller		1B	AA	AA
	VoiceDSP		1B	00	02
Description:	Get the VoiceDSP processor error word (typically sent after GSW when EV_ERROR is reported in the status word). The VoiceDSP processor responds: ERR_OPCODE:				

GI Get Information *item*

Returns the 16-bit value specified by item from one of the internal registers of the VoiceDSP processor.

item may be one of the following:

0 The duration of the last detected DTMF tone, in 10 ms units. The return value is meaningful only if DTMF detection is enabled, and the status word shows that a DTMF tone was detected.

1 The duration of the last detected busy tone in 10 ms units.

2 The energy level of the samples in the last 10 ms.

3 The energy level of the samples, in the last 10 ms, that are in the frequency range described in Figure 2-1. The return value is meaningful only if one of the tone detectors is enabled (bits 0,1 of the detectors mask; see the description of SDET command).

The return value is unpredictable for any other value of item.

Example

GI 0						
Byte sequence:	Microcontroller	25	00	AA	AA
	VoiceDSP	25	00	00	06
Description:	Get the duration of the last detected DTMF tone. The VoiceDSP processor responds: 60 ms.					

GL Get Length

Returns the length of the current message in multiples of 4 Kbytes (blocks).

The returned value includes the message directory information (64 bytes for the first block and 32bytes for every other block), the message data, and the entire last block of the message, even if the message occupies only a portion of the last block. Since a memory block includes 4096 bytes, the returned length may be bigger than the actual message length by up to 4095 bytes. The minimum length of a message is one block.

Example

GL				
Byte sequence:	Microcontroller	19	AA	AA
	VoiceDSP	19	00	04
Description:	Get the length of the current message. The VoiceDSP processor responds: 4 i.e., the message occupies 16384 (4 * 4096) bytes.			

GMS Get Memory Status

Returns the total remaining memory blocks as a 16bit unsigned integer. The estimated remaining recording time may be calculated as follows:

Time = (Num_of_blocks x 4096 x 8) / (Compression_rate x 1000)

This estimate assumes no silence compression: a real recording may be longer, according to the amount of silence detected and compressed.

Example

GMS				
Byte sequence:	Microcontroller	12	AA	AA
	VoiceDSP	12	00	28
Description:	Return the remaining memory blocks. The VoiceDSP responds: 40 blocks.			

GMT Get Message Tag

Returns the 16-bit tag associated with the current message. If the current message is undefined, ERR_VALID is reported.

Example

GMT				
Byte sequence:	Microcontroller	04	AA	AA
	VoiceDSP	04	00	0E
Description:	Get the current message tag. In a system where the message tag is encoded as described in the DMS command, the VoiceDSP processor return value indicates that the message is a new ICM in mailbox Number 6.			

GNM Get Number of Messages *tag_ref tag_mask*

Returns the number of messages whose message tags match the *tag_ref* parameter. Only bits set in *tag_mask* are compared, a match is considered successful if:

message tag and *tag_mask* = *tag_ref* and *tag_mask*

where and is a bitwise AND operation.

The *tag_ref* and *tag_mask* parameters are each two bytes; the return value is also two bytes long.

See "Message Tag" on page 2-3 for a description of message-tag encoding. If *tag_mask* = 0, the total number of all existing messages is returned, regardless of the *tag_ref* value.

Example

GNM FFFE 0003								
Byte sequence:	Microcontroller	11	FF	FE	00	03	AA	AA
	VoiceDSP	11	FF	FE	00	03	00	05
Description:	Get the number of messages which have bit 0 cleared, and bit 1 set, in their message tags. VoiceDSP processor responds that there are five messages which satisfy the request.							

GSW Get Status Word

Returns the 2-byte status word.

STATUS WORD

The VoiceDSP processor has a 16-bit status word to indicate events that occur during normal operation. The VoiceDSP processor asserts the $\overline{\text{MWRQST}}$ signal (clears to 0), to indicate a change in the status word. This signal remains active until the VoiceDSP processor receives a GSW command.

The status word is cleared during reset, and upon a successful GSW command.

15	14	13	12	11	10	9	8	7	6	5	4	3 0
EV_ DTMF	EV_ RESET	EV_ VOX	EV_ CONST_ NRG	Res	EV_ MEMLOW	EV_ DIALTONE	EV_ BUSY	EV_ ERROR	EV_ MEMFULL	EV_ NORMAL_ END	EV_ DTMF_ END	EV_ DTMF_ DIGIT

The bits in the status word are used as follows:

EV_DTMF_DIGIT

DTMF digit. A value indicating a detected DTMF digit. (See the description of DTMF code in the GT command.)

EV_DTMF_END

1 = Ended detection of a DTMF tone. The detected digit is held in EV_DTMF_DIGIT.

EV_NORMAL_END

1 = Normal completion of operation, e.g., end of message playback.

EV_MEMFULL

1 = Memory is full.

EV_ERROR

1 = Error detected in the last command. You must issue the GEW command to return the error code and clear the error condition.

EV_BUSY

1 = Busy tone detected. Use this indicator for call progress and line disconnection.

EV_DIALTONE

1 = Dial tone detected. Use this indicator for call progress and line disconnection.

EV_MEMLOW

1 = Not enough memory. (See CMSG command for further details.)

EV_CONST_NRG

1 = A period of constant energy was detected. Use this indicator for line disconnection. (See CONST_NRG_TIME_COUNT in Table 2-8.)

EV_VOX

1 = A period of silence (no energy) was detected on the telephone line. Use this indicator for line disconnection. (See VOX_TIME_COUNT in Table 2-8.)

EV_RESET

When the VoiceDSP processor completes its power-up sequence and enters the RESET state, this bit is set to 1, and the $\overline{\text{MWRQST}}$ signal is activated (cleared to 0). Normally, this bit changes to 0 after performing the INIT command. If this bit is set during normal operation of the VoiceDSP processor, it indicates an internal VoiceDSP processor error. The microcontroller can recover from such an error by reinitializing the system.

EV_DTMF

1 = Started detection of a DTMF tone.

Example

GSW				
Byte sequence:	Microcontroller	14	AA	AA
	VoiceDSP	14	00	40
Description:	Get the VoiceDSP processor Status Word (typically sent after the $\overline{\text{MMRQST}}$ signal is asserted by the VoiceDSP processor which indicates a change in the status word). The VoiceDSP processor responds that the memory is full.			

GT Generate Tone *tone*

Generates the tone specified by the 1-byte tone parameter. The VoiceDSP state changes to TONE_GENERATE. The tone generation continues until an S command is received.

A DTMF or a single frequency tone may be generated as shown:

To generate a DTMF encode the bits as follows:

Bit 0

1

Bits 1–4

DTMF code.

Where the DTMF code is encoded as follows:

Value (Hex)	DTMF Digit
0 to 9	0 to 9
A	A
B	*
C	#
D	B
E	C
F	D

Bits 5–7

0

To generate a single frequency tone encode the bits as follows:

Bit 0

0

Bits 1-5

3–30

The value in bits 1–5 is multiplied by 100 to generate the required frequency (300Hz–3000Hz).

Bits 6-7

0

The VoiceDSP processor does not check for the validity of the tone specification. Invalid specification yields unpredictable results.

Example

GT 20			
Byte sequence:	Microcontroller	0D	20
	VoiceDSP	0D	20
Description:	Generate a single-frequency 1600Hz tone.		

GTD Get Time and Day *time_day_option*

Returns the time and day as a 2-byte value. *time_day_option* may be one of the following:

0 Get the system time and day.

1 Get the current message time-and-day stamp.

Any other *time_day_option* returns the time-and-day stamp of the current message.

Time and day are encoded as follows:

Bits 0–2

Day of the week (1 through 7).

Bits 3–7

Hour of the day (0 through 23).

Bits 8–13

Minute of the hour (0 through 59).

Bits 14–15

00 The time was not set before the current message was recorded.

11 The time was set, i.e., the SETD (Set Time of Day) command was executed.

NOTE If the current message is undefined, and time_day_option is 1, an ERR_INVALID error is reported.

Example

GTD 1					
Byte sequence:	Microcontroller	0E	01	AA	AA
	VoiceDSP	0E	01	E8	29
Description:	Get the current message time-and-day stamp. The VoiceDSP processor responds that the message was created on the first day of the week at 5:40 A.M. The return value also indicates that the SETD command was used to set the system time and day before the message was recorded. *Note: If the SAS command is used to announce the time-and-day stamp, "Monday" is announced as the first day of the week. For an external vocabulary, the announcement depends on the vocabulary definition (See the* IVS User's Manual *for more details).*				

GTM Get Tagged Message *tag_ref tag_mask dir*

Selects the current message, according to instructions in *dir*, to be the first, n^{th} next or n^{th} previous message which complies with the equation:

message tag and *tag_mask* = *tag_ref* and *tag_mask*.

where and is a bitwise AND operation.

dir is one of the following:

0 Selects the first (oldest) message.

-128 Selects the last (newest) message.

n Selects the n^{th} next message starting from the current message.

–n Selects the n^{th} previous message starting from the current message.

To select the n^{th} message with a given tag to be the current message you must first select the first message that complies with the above equation, and then issue another GTM command with n – 1 as a parameter, to skip to the n^{th} message.

NOTE *To select the n^{th}, or $-n^{th}$, message with a given tag to be the current message you must first select the first message (dir=0), or the last message (dir=-128), that complies with the above equation, and then issue another GTM command with n-1 (for next message), or -n+1 (for previous message), as a parameter, to skip to the n^{th}, or $-n^{th}$, message respectively.*

\f a message is found, it becomes the current message and 1 (TRUE) is returned. If no message is found, the current message remains unchanged and 0 (FALSE) is returned.

If dir is not 0, and the current message is undefined the return value is unpredictable. After the command execution the current message may either remain undefined or changed to any existing message. The only exception is when the GTM command is executed just after the DM command. (See the DM command for further details.)

To access the n^{th} message, when n > 127, a sequence of GTM commands is required.

Example

GTM FFCE 003F 0								
Byte sequence:	Microcontroller	09	FF	CE	00	3F	00	AA
	VoiceDSP	09	FF	CE	00	3F	00	01

Description:	Select the oldest of the new ICMs, in mailbox number 6, to be the current message, for a system where the message tag is encoded as described in the example for the DMS command.
	The VoiceDSP processor returns a value indicates that there is such a message.
	The following pseudo-code demonstrates how to play all new ICMs in mailbox number 6. The messages are marked as old after being played:
	
	Return_val = GTM(FFCE, 003F, 00) /*Get the oldest message with the defined tag*/
	While (ReturnVal == TRUE)
	Begin
	P() /* Play */
	Message_tag = GMT() /* Get message tag */
	SMT(FFF7) /* Mark the message as `old' */
	GTM(FFCE, 003F, 01) /* Get next message with the same tag */
	End							

GTUNE Get Tune *index*

Gets the value of the tunable parameter identified by *index* (one byte) as the 2-byte value, *parameter_value*. This command may be used to read and identify the parameter value that was set to tune the VoiceDSP.

If *index* does not point to a valid tunable parameter, ERR_PARAM is set in the error word.

The GTUNE command may be used in IDLE state or RESET state. If TUNE command was not used to set the tunable parameters, then the GTUNE command will read the default parameter value.

Tables 2-4 to 2-11 describe the tunable parameters, their index numbers and their default values.

Example

GTUNE 17					
Byte sequence:	Microcontroller	06	17	AA	AA
	VoiceDSP	06	17	02	BC
Description:	Get the minimum period for busy detection ComactSPEECH responds: 700 (7 seconds).				

INIT Initialize System

Execute this command after the VoiceDSP processor has been configured (see CFG command). INIT performs a soft reset of the VoiceDSP as follows:

- Initializes the message directory information.
- Messages are not deleted. To delete the messages, use the DM and DMS commands.
- Sets the detectors mask to 0.
- Sets the volume level that is controlled by the VC command, to 0.
- Sets the playback speed to normal (0).
- Switches to the IDLE state.
- Initializes the tone detectors.

The current message is undefined after INIT execution.

The tunable parameters are not affected by this command. They are set to their default values only during RESET.

Example

INIT		
Byte sequence:	Microcontroller	13
	VoiceDSP	13
Description:	Initialize the VoiceDSP processor.	

INJ Inject IVS Data *n byte*$_1$*. . . byte*$_n$

Injects vocabulary data of size *n* bytes to good Flash blocks.

This command programs Flash devices, on a production line, with IVS vocabulary data. It is optimized for speed; all VoiceDSP processor detectors are suspended during execution of the command. Use the CVOC command to check whether programming was successful.

If there is not enough memory space for the vocabulary data, ERR_PARAM is set in the error word, and execution stops.

Flash blocks that include IVS data can not be used for recording, even if only one byte of the block contains IVS data (e.g., if the vocabulary size is 4K + 100 bytes, two blocks of the Flash are not available for message recording).

Example

INJ 00000080 Data							
Byte sequence:	Microcontroller	29	00	00	00	80	Vocabulary Data
	VoiceDSP	29	00	00	00	80	Echo of Data
Description:	Inject 128 bytes of vocabulary data.						

MR Memory Reset

Erases all memory blocks and initializes the VoiceDSP processor (does exactly what the INIT command does). Bad blocks, and blocks which are used for IVS vocabularies, are not erased. This command can be issued in either RESET or IDLE states.

NOTE When Memory Reset is used in RESET state, it must be issued after the CFG command is issued, or the memory type and number of devices are not defined. In this case the result is unpredictable.

NOTE The command erases all messages and should be used with care.

Example

MR		
Byte sequence:	Microcontroller	2A
	VoiceDSP	2A
Description:	Erase all memory blocks.	

P Playback

Begins playback of the current message. The VoiceDSP processor state changes to PLAY. When playback is complete, the VoiceDSP processor sets the EV_NORMAL_END bit in the status word, and activates (clears to 0) the $\overline{\text{MWRQST}}$ signal. The state then changes to IDLE. Playback can be paused with the PA command, and can be resumed later with the RES command. Playback can be stopped with the S command.

If the current message is undefined, ERR_INVALID is reported.

Example

P		
Byte sequence:	Microcontroller	03
	VoiceDSP	03
Description:	Play the current message.	

PA Pause

Suspends the execution of the current GT, P, R, SAS, SO, SW, or SS command. The PA command does not change the state of the VoiceDSP processor; execution can be resumed with the RES command.

NOTE DTMF and tone detectors remain active during Pause.

Example

PA		
Byte sequence:	Microcontroller	1C
	VoiceDSP	1C
Description:	Suspend playback of current message.	

PDM Go To Power-Down Mode

Switches the VoiceDSP processor to power-down mode (see "POWER-DOWN MODE" on page 1-4 for details). Sending any command while in power-down mode resets the processor detectors, and returns it to normal operation mode.

NOTE If an event report is pending ($\overline{MWRQST}$ is active) and not processed by the microcontroller prior to issuing the PDM command, the event is lost.

Example

PDM		
Byte sequence:	Microcontroller	1A
	VoiceDSP	1A
Description:	Put the VoiceDSP processor in power-down mode.	

R Record *tag compression_rate*

Records a new message with message tag *tag* and compression rate *compression_rate*. The VoiceDSP processor state changes to RECORD. The R command continues execution until stopped by the S command. Recording can be paused with the PA command, and can be resumed later with the RES command.

If the memory becomes full, recording stops and EV_MEMFULL is set in the status word.

See "Message Tag" on page 2-3 for a description of message-tag encoding.

The compression rate may be defined as 0 for PCM recording or either 1, 3, 6 for compression rates 5.3 Kbits/sec, 9.9 Kbits/sec, 16.8 Kbits/sec respectively. See " VCD" on page 2-10 for a description of the compression algorithm.

NOTE A time-and-day stamp is automatically attached to each message. Before using the R command for the first time, use the SETD command. Failure to do so results in undefined values for the time-and-day stamp.

Example of a typical recording session:

- (ICM) The microcontroller detects the first ring.
- (ICM, OGM, memo) The microcontroller sends the R command.

Example

R 000E 03					
Byte sequence:	Microcontroller	0C	00	0E	03
	VoiceDSP	0C	00	0E	03
Description:	Record a new ICM in mailbox Number 6 in a system where the message tag is encoded as described in the example of the DMS command. The compression rate is defined as 9.9 Kbits/sec.				

RDET Reset Detectors *detectors_reset_mask*

Resets the VoiceDSP processor tone and energy detectors according to the value of the detectors_reset_mask parameter. A bit set to 1 in the mask, resets the corresponding detector. A bit cleared to 0 is ignored.

The 1-byte *detectors_reset_mask* is encoded as follows:

Bit 0

Reset the busy and dial tone detectors.

Bits 1–3

Reserved. Must be cleared to 0.

Bit 4

Reset the constant energy detector.

Bit 5

Reset the no energy (VOX) detector.

Bit 6

Reset the DTMF detector.

Bit 7

Reserved. Must be cleared to 0.

Example

RDET 20			
Byte sequence:	Microcontroller	2C	20
	VoiceDSP	2C	20
Description:	Reset the VOX detector.		

RES Resume

Resumes the activity that was suspended by the PA, SB, or SF commands.

Example

RES		
Byte sequence:	Microcontroller	1D
	VoiceDSP	1D
Description:	Resume playback which was suspended by either the PA, SF or SB command.	

RMSG Read Message

Returns 32 bytes of data from the current position of the message pointer, and advances the message pointer by 32 bytes.

If the VoiceDSP processor was in the IDLE state, the command opens the current message, switches the VoiceDSP processor to the MSG_OPEN state, sets the message pointer to the beginning of the message data, and returns the 32 bytes of data.

The microcontroller must issue an S command to close the message, and switch the VoiceDSP processor to the IDLE state.

If the current message is undefined, ERR_INVALID is reported.

Trying to read beyond the end of the message sets the EV_NORMAL_END event, and the VoiceDSP processor switches to the IDLE state. In this case, the return value is undefined and should be ignored.

Example

RMSG					
Byte sequence:	Microcontroller	32	AA	AA	...
	VoiceDSP	32	32 bytes of data		
Description:	Read 32 bytes from the current message memory.				

S Stop

Stops execution of the current command and switches the VoiceDSP processor to the IDLE state. S may be used to stop the execution of CMSG, SMSG, WMSG, RMSG and all asynchronous commands.

Example

S		
Byte sequence:	Microcontroller	00
	VoiceDSP	00
Description:	Stop current activity (e.g., playback, recording) and change the VoiceDSP processor to IDLE state.	

SAS Say Argumented Sentence *sentence_n arg*

Announces sentence number *sentence_n* of the currently selected vocabulary, and passes *arg*. *sentence_n* and *arg* are each 1-byte long. The VoiceDSP processor state changes to SYNTHESIS.

When playing is complete, the VoiceDSP processor sets the EV_NORMAL_END bit in the status word, and activates the $\overline{\text{MWRQST}}$ signal. The state then changes to IDLE.

If the current vocabulary is undefined, ERR_INVALID is reported.

Example

SAS 00 03				
Byte sequence:	Microcontroller	1E	00	03
	VoiceDSP	1E	00	03
Description:	Announce the first sentence in the sentence table of the currently selected vocabulary with '3' as the actual parameter.			

SB Skip Backward *time_length*

Skips backward in the current message time_length units, in 0.2 second segments, and pauses message playback. The RES command must be issue to continue playback. *time_length* is a 2-byte parameter that can have any value up to 320 (64 seconds). The skip accuracy is five percent. SB is meaningful only in the PLAY state.

If the beginning of the message is detected during the execution of the SB command, execution terminates, the EV_NORMAL_END bit in the status register sets, the $\overline{\text{MWRQST}}$ signal activates, and the processor switches to the IDLE state.

If *time_length* is greater than 320, ERR_PARAM is set in the error word.

Playback speed does not affect the behavior of this command.

Example

SB 0019				
Byte sequence:	Microcontroller	23	00	19
	VoiceDSP	23	00	19
Description:	Skip backwards five seconds from the current position in the message being played.			

SDET Set Detectors Mask *detectors_mask*

Controls the reporting of detection of tones and energy detectors according to the value of the *detectors_mask* parameter. A bit set to 1 in the mask, enables the reporting of the corresponding detector. A bit cleared to 0 disables the reporting.

Disabling reporting of a detector does not stop or reset the detector.

The 1-byte *detectors_mask* is encoded as follows:

Bit 0

Report detection of a busy tone.

Bit 1

Report detection of a dial tone.

Bit 2-3

Reserved. Must be cleared to 0.

Bit 4

Report detection of a constant energy.

Bit 5

Report detection of no energy (VOX) on the line.

Bit 6

Report the ending of a detected DTMF.

Bit 7

Report the start of a detected DTMF (up to 40 ms after detection start).

Example

SDET A3			
Byte sequence:	Microcontroller	10	B3
	VoiceDSP	10	B3
Description:	Set reporting of all VoiceDSP processor detectors, except for end-of-DTMF.		

SE Skip to End of Message

This command is valid only in the PLAY state. When invoked, playback is suspended (as for the PA command), and a jump to the end of the message is performed. Playback remains suspended after the jump.

Example

SE		
Byte sequence:	Microcontroller	24
	VoiceDSP	24
Description:	Skip to end of current message.	

SETD Set Time and Day *time_and_day*

Sets the system time and day as specified by the 2-bytes *time_and_day* parameter. The *time_and_day* parameter is encoded as follows:

Bits 0–2

Day of the week (1 through 7).

Bits 3–7

Hour of the day (0 through 23).

Bits 8–13

Minute of the hour (0 through 59).

Bits 14–15

Must be set to 1.

If *time_and_day* value is not valid, ERR_PARAM is set in the error word.

Example

SETD DE09				
Byte sequence:	Microcontroller	0F	DE	09
	VoiceDSP	0F	DE	09
Description:	Set time and day to Monday 1.30 A.M. (where Monday is the first day of the week)			

SF Skip Forward *time_length*

Skips forward in the current message *time_length* units, in 0.2 second segments, and causes message playback to pause. The RES command must be issue to continue playback. *time_length* is a 2-byte parameter that can have any value up to 320 (64 seconds). The skip accuracy is five percent. This command is meaningful only in the PLAY state. The RES command must be issue to continue playback.

If the end of the message is detected during execution of SF, execution of the command is terminated the EV_NORMAL_END bit in the status word is set, the $\overline{\text{MWRQST}}$ signal is activated and the processor switches to the IDLE state.

If time_length is greater than 320, ERR_PARAM is set in the error word.

Playback speed does not affect the behavior of this command.

Example

SF 0019				
Byte sequence:	Microcontroller	22	00	19
	VoiceDSP	22	00	19
Description:	Skip forward five seconds from the current position in the message being played.			

SMSG Set Message Pointer *num_of_pages*

Sets the message pointer to *num_of_pages* x 32 bytes from the beginning of the current message data.

If the VoiceDSP processor was in the IDLE state, the command opens the current message and switches the VoiceDSP processor to the MSG_OPEN state. The microcontroller must issue an S command to close the message, and switch the VoiceDSP processor to the IDLE state.

If *num_of_pages* x 32 is greater than the message length, EV_NORMAL_END is set in the status word, the message pointer is set to the end of the message, and the VoiceDSP processor switches to the IDLE state.

If the current message is undefined, ERR_INVALID is reported.

Example

SMSG 000A				
Byte sequence:	Microcontroller	30	00	0A
	VoiceDSP	30	00	0A
Description:	Set the message pointer to 10 pages (320 bytes) from the beginning of the current message data.			

SMT Set Message Tag *message_tag*

Sets the tag of the current message. The 2-byte *message_tag* can be used to implement mailbox functions by including the mailbox number in the tag, or to handle old and new messages differently by using one bit in the tag to mark the message as old or new. See "Message Tag" on page 2-3.

To change the message tag, you should first get the tag using the GMT command, read the tag, modify it, and write it back.

NOTE Message tag bits can only be cleared. Message tag bits are set only when a message is first created.

If the current message is undefined, ERR_INVALID is reported.

Example

SMT FFF7				
Byte sequence:	Microcontroller	05	FF	F7
	VoiceDSP	05	FF	F7
Description:	Mark the current message as old in a system where the message tag is encoded as described in the example of the DMS command. *Note that the VoiceDSP processor ignores bits in the tag which are set to 1; only bit 3 is modified in the message tag.*			

SO Say One Word *word_number*

Plays the word number *word_number* in the current vocabulary. The 1-byte *word_number* may be any value from 0 through the index of the last word in the vocabulary. The VoiceDSP processor state changes to SYNTHESIS.

When playback of the selected word has been completed, the VoiceDSP processor sets the EV_NORMAL_END bit in the status word, and activates the $\overline{\text{MWRQST}}$ signal. The state then changes to IDLE.

If *word_number* is not defined in the current vocabulary, or if it is an IVS control or option code, ERR_PARAM is set in the error word.

If the current vocabulary is undefined, ERR_INVALID is reported.

Example

SO 00			
Byte sequence:	Microcontroller	07	00
	VoiceDSP	07	00
Description:	Announce the first word in the word table of the currently selected vocabulary.		

SPS Set Playback Speed *speed*

Sets the speed of message playback as specified by speed. The new speed applies to all recorded messages and synthesized messages (only if synthesized using IVS), until changed by another SPS command. If this command is issued while the VoiceDSP processor is in the PLAY state, the speed also changes for the message currently being played.

Speed may be one of 13 values, from –6 to +6. A value of 0 represents normal speed.

NOTE A negative speed value represents an increase in speed, a positive value represents a decrease in speed.

NOTE The playback speed control is not applicable when the stored messages or the IVS data are not compressed (Stored in PCM format).

The change in speed is approximate, and depends on the recorded data. In any case, if i < j, playback speed with parameter i is the same or faster than with parameter j.

If *speed* is not in the –6 to +6 range, ERR_PARAM is set in the error word.

Example

SPS FB			
Byte sequence:	Microcontroller	16	FB
	VoiceDSP	16	FB
Description:	Set playback speed to –5.		

SS Say Sentence *sentence_n*

Say sentence number *sentence_n* of the currently selected vocabulary. *sentence_n* is 1-byte long. The VoiceDSP processor state changes to SYNTHESIS.

If the sentence has an argument, 0 is passed as the value for this argument.

When playing has been completed, the VoiceDSP processor sets the EV_NORMAL_END bit in the status word, and activates the $\overline{\text{MWRQST}}$ signal. The state then changes to IDLE.

If *sentence_n* is not defined in the current vocabulary, ERR_PARAM is set in the error word.

If the current vocabulary is undefined, ERR_INVALID is reported.

Example

SS 00			
Byte sequence:	Microcontroller	1F	00
	VoiceDSP	1F	00
Description:	Announce the first sentence in the sentence table of the currently selected vocabulary.		

SSM Set Speakerphone Mode *mode*

Sets the speakerphone to the *mode* mode of operation. The command is valid when the VoiceDSP processor is in IDLE state. *mode* can be one of:

0	OFF	Deactivate the speakerphone, and return the VoiceDSP processor to normal operation mode.
1	ON	Put the VoiceDSP processor in speakerphone mode and activate speakerphone in full-duplex mode i.e., with full cancellation of both the acoustic and the electrical echoes. Tone detectors are not active. Gains in the Send and Receive paths are set by the relevant tunable parameters.
2	TRANSPARENT	Activate the speakerphone with no echo cancellation (this mode is used for system tuning).
3	MUTE	Activate the speakerphone, while generating silence to the line. The near-end-listener can hear the far-end-speaker, but not vice versa. Tone detectors are not active.
4	LISTEN	The line is audible on the speaker. Tone detectors are active. This mode is used for call generation.
5	Reserved.	
6	RESTART	Restart the current speakerphone mode. This mode differs from ON; it does not require full initialization of the speakerphone. It should be used to resume the speakerphone operation after HOLD mode or to adjust to an environment change (e.g., parallel pickup).
7	HOLD	Stop the codec interrupts. Neither side can hear each other.

See "Full-duplex Speakerphone" on page 2-14 for more details.

NOTE Only commands that are specified in Table 2-3, are active during all speakerphone modes (other than 0).

Example

SSM 01			
Byte sequence:	Microcontroller	2F	01
	VoiceDSP	2F	01
Description:	Put the VoiceDSP processor into Speakerphone mode, and set the speakerphone to full-duplex mode.		

SV Set Vocabulary Type *type id*

Selects the vocabulary table to be used for voice synthesis. The vocabulary type is set according to the 1-byte *type* parameter:

0 For compatibility only.

1 External vocabulary in ROM.

2 External vocabulary in Flash.

3-7 Reserved.

The host is responsible for selecting the current vocabulary, with SV command, before using an SAS, SO, SS or SW command. Each external vocabulary table has a unique id which is part of the vocabulary internal header (See the *IVS User's Guide* for more details). If type is 1 or 2, the VoiceDSP processor searches for the one byte *id* parameter in each vocabulary table header until a match is found.

If the *id* parameter does not point to a valid IVS vocabulary, ERR_PARAM is set in the error word.

Example

SV 02 03				
Byte sequence:	Microcontroller	20	02	03
	VoiceDSP	20	02	03
Description:	Select the vocabulary with vocabulary-id 3, which resides on a Flash, as the current vocabulary.			

SW Say Words *n word$_1$. . . word$_n$*

Plays n words, indexed by *word$_1$* to *word$_n$*. The VoiceDSP processor state changes to SYNTHESIS.

On completion, the EV_NORMAL_END bit in the status word is set, and the $\overline{\text{MWRQST}}$ signal goes low. The state then changes to IDLE.

If one of the words is not defined in the current vocabulary, or if it is an IVS control or option code, or if n > 8, ERR_PARAM is reported.

If the current vocabulary is undefined, ERR_INVALID is reported.

Example

SW 02 00 00					
Byte sequence:	Microcontroller	21	02	00	00
	VoiceDSP	21	02	00	00
Description:	Announce the first word, in the word table of the currently selected vocabulary, twice.				

TUNE Tune *index parameter_value*

Sets the value of the tunable parameter identified by *index* (one byte) to the 2-byte value, *parameter_value*. This command may be used to tune the DSP algorithms to a specific Data Access Arrangement (DAA) interface, or to change other parameters. If you do not use TUNE, the VoiceDSP processor uses default values.

If *index* does not point to a valid tunable parameter, ERR_PARAM is set in the error word.

NOTE The tunable parameters are assigned with their default values on application of power. The INIT command does not affect these parameters.

The following tables 2-4 to 2-11 describe the tunable parameters, their index numbers and their default values, grouped by their functionality.

Table 2-4: TUNABLE PARAMETERS: Voice Compression and Decompression (VCD)

Index	Parameter Name	Description	Default
4	Voice Activity Detection (VAD): VAD_SIL_THRESHOLD	Prevents speech from being interpreted as silence. The silence detection algorithm has an adaptive threshold, which is changed according to the noise level. This parameter is, therefore, only the initial threshold level. *Legal values: 9216 to 13824 in 512 (6 dB) steps.*	11264
5	Voice Activity Detection (VAD): VAD_SIL_THRESHOLD_STEP	Defines the adaptive threshold changes step. If this threshold is too low, the threshold converges too slowly. If it is too high, silence detection is too sensitive to any noise. *Legal values: 3 to 48.*	12
6	Voice Activity Detection (VAD): VAD_SIL_BURST_THRESHOLD	The minimum time period for speech detection, during silence. As this threshold increases, the time period interpreted as silence increases. If this threshold is too low, a burst of noise is detected as speech. If it is too high, words may be partially cut off. *Legal values: 1 to 3.*	2

Table 2-4: TUNABLE PARAMETERS: Voice Compression and Decompression (VCD)

Index	Parameter Name	Description	Default
7	Voice Activity Detection (VAD): VAD_SIL_HANG_THRESHOLD	The minimum time period for silence detection, during speech. As this threshold increases, the time period interpreted as silence decreases. If this threshold is too low, words may be partially cut off. If it is too high, no silence is detected. *Legal values: 8 to 31.*	15
8	Voice Activity Detection (VAD): VAD_SIL_ENABLE	Silence compression control. 0 turns silence compression off. *Note: Silence compression must be turned off when using ARAM for voice storage. Otherwise the playback quality is unpredictable.*	1
9	Voice Activity Detection (VAD): VAD_ENERGY_FACTOR	Determines the energy level used to synthesize silence. For the default value, the energy levels of the synthesized silence and the recorded silence are the same. If you divide (multiply) the default value by two, the synthesized silence is 6 dB less (more) than the level of the recorded silence. *Legal values: 1024 to 16384.*	8192
70	SW Automatic Gain Control (SW AGC): SWAGC_ENABLE	SW AGC control. 0 turns SW AGC off.	1
11	SW Automatic Gain Control (SW AGC): SWAGC_FACTOR	Determines the maximum gain that the SW AGC algorithm may use. *Legal values: 0, 1, 2, 4, 8, 16, 32, 64, 128* *Note: Value 0 means the the maximum gain is defined by the algorithm.*	128

Table 2-5: TUNABLE PARAMETERS: Tone Generation and Message Playback

Index	Parameter Name	Description	Default
27	DTMF Generation: DTMF_GEN_TWIST_LEVEL	A one-byte value that controls the twist level of a DTMF tone, generated by the GT command, by controlling the energy level of each of the two tones (low frequency and high frequency) composing the DTMF tone. The Least Significant Nibble (LSN) controls the low tone and the Most Significant Nibble (MSN) controls the high tone. The energy level of each tone, as measured at the output of a TP3054 codec (before the DAA) connected to the VoiceDSP processor is summarized in the following table: *Nibble Value* — *Tone Energy (dB-Volts)* 0 — 0 1 — –17.8 2 — –14.3 3 — –12.9 4 — –12.4 5 — –12.0 6 — –11.9 7 — –11.85 8..15 — –11.85 The volume of the generated DTMF tone during measurements was 6. (TONE_GEN_LEVEL+VOL_LEVEL = 6). For the default level, the high tone is –14.3 dBV and the low tone is –12.4 dBV, which gives a DTMF twist level of 1.9 dB. The energy level of a single generated tone is the level of the low tone.	66
16	Tone Generation: TONE_GEN_LEVEL	Controls the energy level at which DTMF and other tones are generated. Each unit represents 3 dB. The default level is the reference level. For example, if you set this parameter to 4, the energy level is 6 dB less than the default level. The actual output level is the sum of TONE_GEN_LEVEL and the VOL_LEVEL variable, controlled by the VC command. The tones are distorted when the level is set too high. *Legal values: 0 ≤ TONE_GEN_LEVEL + VOL_LEVEL ≤ 12.*	6
21	VCD Playback and Voice Synthesis: VCD_PLAY_LEVEL	Controls the energy during playback and external voice synthesis. Each unit represents 3 dB. The default level is the reference level. For example, if you set this parameter to 4, the energy level is 6 dB less than the default level. The actual output level is the sum of VCD_LEVEL and the VOL_LEVEL variable, controlled by the VC command. Speech is distorted when the level is set too high. *Legal values: 0 ≤ VCD_PLAY_LEVEL + VOL_LEVEL ≤ 12.*	6

Table 2-6: TUNABLE PARAMETERS: DTMF Detection

Index	Parameter Name	Description	Default
17	Energy Level: DTMF_DET_MIN_ENERGY	Minimum energy level at which DTMF tones are detected. If you divide (multiply) the value by 2, the detection sensitivity decreases (increases) by 3 dB. *Legal values: 8 to 4096*	32
24	Echo Canceler: DTMF_DET_ECHO_DELAY	The near-echo delay in samples. The sampling rate is 8000Hz (i.e., 125 ms per sample). *Legal values: 0 to 16.*	4
26	Twist Level: DTMF_DET_REV_TWIST	Controls the reverse twist level at which the VoiceDSP processor detects DTMF tones. While the normal twist is set at 8 dB, the reverse twist can be either 4 dB (default) or 8 dB (if this parameter is set to 1).	0
60	SW AGC: DTMF_DET_AGC_IDLE	SW AGC for DTMF in idle/record modes. When incrementing the tunable by 1, the dynamic range is increased by 3 dB. *Legal values: 0 to 5.*	0
61	SW AGC: DTMF_DET_AGC_PLAY	Software AGC for play mode and tone generation modes. When incrementing the tunable by 1, the dynamic range increases by 3 dB. *Legal values: 0 to 16.*	3

Table 2-7: TUNABLE PARAMETERS: Tone Detection

Index	Parameter Name	Description	Default
18	Dial Tone: TONE_DET_TIME_COUNT	Controls the duration of a tone before it is reported as a dial tone, in 10 msec units. The accuracy of the constant is ±10 ms. *Legal values: 0 to 65535.*	700
19	Busy and Dial Tone: TONE_DET_ON_ENERGY_THRESHOLD	Minimum energy level at which busy and dial tones are detected as ON (after 700Hz filtering). If you divide (multiply) the value by 2 you get about 3 dB decrease (increase) in the threshold. The mapping between energy level and the parameter value is as follows (measured on the codec output when a 400Hz tone was injected to the codec input): *Tunable value* — *Energy threshold (dB-Volts)* 10 — –31.8 20 — –28.6 100 — –21.7 500 — –14.7 8000 — –2.5 *Legal values: 0 to 65535.*	160
20	Busy and Dial Tone: TONE_DET_OFF_ENERGY_THRESHOLD	Maximum energy level at which busy and dial tones are detected as OFF (after 700Hz filtering). If you divide (multiply) the value by 2 you get about 3 dB decrease (increase) in the threshold. The mapping between energy level and the parameter value is the same as for TONE_ON_ENERGY_THRESHOLD *Legal values: 0 to 65535.*	110
23	Busy Tone: BUSY_DET_MIN_TIME	Minimum time period for busy detection, in 10 ms units. The accuracy of the constant is ±10 ms. *Legal values: 0 to 65535.*	600
53	Busy Tone: BUSY_DET_MIN_ON_TIME	Minimum period considered as On period for busy tone detection. Note that for weak signals: (–30 dB and below) the maximum value is 12 (i.e., 120 ms minimum detection time). Unit: 10 ms. Accuracy is ±20 ms. *Legal values: 10 to 1000.*	10
54	Busy Tone: BUSY_DET_MAX_ON_TIME	Maximum period considered as On for busy-tone detection. Unit: 10 ms. Accuracy is ±20 ms. *Legal values: 10 to 1000.*	168

Table 2-7: TUNABLE PARAMETERS: Tone Detection

Index	Parameter Name	Description	Default
55	Busy Tone: BUSY_DET_MIN_OFF_TIME	Minimum period considered as Off for busy-tone detection. Unit: 10 ms. Accuracy is ±20 ms. *Legal values: 5 to 1000.*	7
56	Busy Tone: BUSY_DET_MAX_OFF_TIME	Maximum period considered as On for busy-tone detection. Unit: 10 ms. Accuracy is ±20 ms. *Legal values: 5 to 1000.*	122
57	Busy Tone: BUSY_DET_VERIFY_COUNT	Number of On/Off cadences that must be detected prior to reporting busy-tone presence. *Legal values: 9 to 127.*	9
58	Busy Tone: BUSY_DET_TONE_TYPE	Specifies the type of busy tone to detect: 1 —Two cadences 2 —Three cadences 3 —Both two and three cadences	1
59	Busy Tone: BUSY_DET_DIFF_THRESHOLD	The maximum allowed difference between two compared On or Off periods. Unit: 10 ms. *Legal values: 0 to 1000.*	9

Table 2-8: TUNABLE PARAMETERS: Energy Detection

Index	Parameter Name	Description	Default
10	Silence (VOX): VOX_DET_ENERGY_THRESHOLD	This parameter determines the minimum energy level at which voice is detected. Below this level, it is interpreted as silence. *Legal values: 1 to 32767.*	12
12	Silence (VOX): VOX_DET_TIME_COUNT	This parameter, in units of 10 ms, determines the period of silence before the VoiceDSP processor reports silence. The accuracy of the constant is ±10 ms. *Legal values: 0 to 65535.*	700
22	Silence (VOX): VOX_DET_TOLERANCE_TIME	Controls the maximum energy-period, in 10 ms units, that does NOT reset the vox detector. *Legal values: 0 to 255.*	3
47	Constant Energy: CONST_NRG_DET_TIME_COUNT	Minimum elapsed time until the VoiceDSP processor reports constant energy level. Units: 10 ms. Accuracy: ±10 ms *Legal values: 1 to 65534*	700
48	Constant Energy: CONST_NRG_DET_TOLERANCE_TIME	Variations in constant energy, up to this time, do not reset the constant energy detector. Units: 10 ms. *Legal values: 0 to 255*	5
49	Constant Energy: CONST_NRG_DET_LOW_THRESHOLD	Determines the minimum energy level that is treated as constant energy. The minimum energy is calculated as follows: $(1—1/2^{CONST\_NRG\_DET\_LOW\_THRESHOLD}) * average\_energy$ *Legal values: 1 to 16*	1
50	Constant Energy: CONST_NRG_DET_HIGH_THRESHOLD	Determines the maximum energy level that is treated as constant energy. The maximum energy is calculated as follows: $(1 + 1/2^{CONST\_NRG\_DET\_HIGH\_THRESHOLD}) * average\_energy$ *Legal values: 0 to 16*	1

Table 2-9: TUNABLE PARAMETERS: Speakerphone

Index	Parameter Name	Description	Default
31	Acoustic Echo Canceler (AEC): SP_AEC_PRIORITY_BIAS	Controls the bias in priority between the Send and Receive paths. If send priority-bias is preferred, the value should be greater than zero. For no priority bias, the value should be zero. For priority bias for the Receive path, the value should be negative. Steps are 3 dB each (e.g., +3 is 9 dB bias for the Send path, –2 is 6 dB bias for the Receive path). *Legal values: –4 to 4.*	0
32	Acoustic Echo Canceler (AEC): SP_AEC_COUPLING_ LOSS_THRESHOLD	This parameter limits the acoustic return loss. Its value (SP_AEC_COUPLING_LOSS_THRESHOLD / 32767) is compared with the RMS value of S_{out} divided by the RMS value of R_{in}, during a single-talk event. The loop gain is decreased, if necessary, to control the TCL level. For SP_AEC_COUPLING_LOSS_THRESHOLD = 32767 this loop is disabled. *Legal values: 0 to 32767.*	2047
34	Acoustic Echo Canceler (AEC): SP_AEC_LR_LEVEL	Controls the speakerphone gain from the microphone to the line-out. The total attenuation, or gain, depends on both of the analog gains and this value. The gain is: K * signal where: K = SP_AEC_LR_LEVEL/4096. *Legal values: 0 to 16000.*	14000
36	Acoustic Echo Canceler (AEC): SP_AEC_CLIP_POS	Specifies the positive peak-value at which the analog circuit of the line-out saturates. Codec analog full scale corresponds to µLAW full scale values after expansion.Assume that positive saturation occurs at amplitudes higher than those of a sine wave at X (dBm0). The SP_AEC_CLIP_POS value is set as: SP_AEC_CLIP_POS = $32636 * 10^{((X - 3.17)/20)}$ *Note: a sine wave with amplitude 4 * 8159 = 32636 corresponds to 3.17 dBm0.* Example: For X = –6.2761 dBm0, the value is: SP_AEC_CLIP_POS = $32636 * 10^{((-6.2761 - 3.17)/20)} = 0.3371 * 32636 = 11000$; *Legal values: 0 to 32767.*	16000
37	Acoustic Echo Canceler (AEC): SP_AEC_CLIP_NEG	Specifies the negative peak value at which the analog circuit of the line-out saturates. Codec analog full scale corresponds to µLAW full scale values after expansion.The value of SP_AEC_CLIP_NEG is set as shown for SP_AEC_CLIP_POS, above. *Legal values: –32768 to 0.*	–16000

Table 2-9: TUNABLE PARAMETERS: Speakerphone

Index	Parameter Name	Description	Default
40	Acoustic Echo Canceler (AEC): SP_AEC_ENABLE	Enables/disables the acoustic echo controller. *Legal values: 0 (disable), 1 (enable).*	1
43	Acoustic Echo Canceler (AEC): SP_AEC_VOX_HYST	Controls the hysteresis in near-talker detection. (The speakerphone state machine has a built-in hysteresis mechanism to prevent fluctuations in the talker identification process i.e., identifying the active side.) The value of this parameter is a dimensionless number, which should be evaluated during the tuning process for specific hardware. Larger values for the parameter correspond to a wider hysteresis loop. Negative values increase the probability that the state machine remains in the last state. *Legal values: –127 to 127.*	10
45	Acoustic Echo Canceler (AEC): SP_AEC_DTD_TH	Controls the sensitivity of the system. Low values correspond to high sensitivity, with a greater false alarm probability (i.e., an echo is considered a real talker). High values correspond to low sensitivity, with slower switching. This parameter is affected by the loop gain and the specific hardware characteristics. *Legal values: 0 to 127.*	73
35	Electric Echo Canceler (EEC): SP_EEC_LR_LEVEL	Controls the speakerphone gain from the line-in to the speaker. The total attenuation, or gain, depends on both of the analog gains and this value. The gain is: K * signal where: $K = (SP\_EEC\_LR\_LEVEL/4096) * (2^{(6 + VOL\_LEVEL)/2})$ *Legal values: 0 to 400.*	281

Table 2-9: TUNABLE PARAMETERS: Speakerphone

Index	Parameter Name	Description	Default
38	Electric Echo Canceler (EEC): SP_EEC_CLIP_POS	Specifies the positive peak value at which the analog circuit of the speaker saturates. Codec analog full scale corresponds to µLAW full scale values after expansion. The value of SP_EEC_CLIP_POS is set as shown for SP_AEC_CLIP_POS, above. *Legal values: 0 to 32767.*	16000
39	Electric Echo Canceler (EEC): SP_EEC_CLIP_NEG	Specifies the negative peak value at which the analog circuit of the line-out saturates. Codec analog full scale corresponds to µLAW full scale values after expansion. The value of SP_EEC_CLIP_NEG is set as shown for SP_AEC_CLIP_POS, above. *Legal values: –32768 to 0.*	–16000
41	Electric Echo Canceler (EEC): SP_EEC_ENABLE	Enables/disables the electrical echo controller. *Legal values: 0 (disable), 1 (enable).*	1
44	Electric Echo Canceler (EEC): SP_EEC_VOX_HYST	Controls the hysteresis in far-talker detection. (The speakerphone state machine has a built-in hysteresis mechanism to prevent fluctuations in the talker identification process i.e., identifying the active side.) The value of this parameter is a dimensionless number, which should be evaluated during the tuning process for specific hardware. Larger values for the parameter correspond to a wider hysteresis loop. Negative values increase the probability that the state machine remains in the last state. *Legal values: –127 to 127.*	10
46	Electric Echo Canceler (EEC): SP_EEC_DTD_TH	Controls the sensitivity of the system. Low values correspond to high sensitivity, with a greater false alarm probability (i.e., an echo is considered a real talker). High values correspond to low sensitivity, with slower switching. This parameter is affected by the loop gain and the specific hardware characteristics. *Legal values: 0 to 127.*	82
33	Attenuation: SP_BLOCK_LEVEL	Controls the maximum attenuation level of the speakerphone suppressors. It affects the speakerphone stability and its subjective quality. The maximum attenuation is calculated according to: $SP\_BLOCK\_LEVEL/2^{28}$ *Legal values: 550 to 32000.*	10922
42	Tone Generation: SP_TONE_GEN_ LEVEL	Controls the energy level at which DTMF, and other tones, are generated to the line (codec 0) while the speakerphone is active. Each unit represents 3 dB. *Legal values: 0 ≤ SP_TONE_GEN_LEVEL ≤ 10.* *Note: the energy level at which the tones are generated to the speaker (codec 1) while the speakerphone is active, is controlled by the TONE_GEN_LEVEL tunable parameter and the vol_level.*	6

Table 2-10: TUNABLE PARAMETERS: Memory Support

Index	Parameter Name	Description	Default
62	Memory Device Size: NUM_OF_BLOCKS_IN_MEM	Defines the nubber of blocks (each block is of 4096 bytes) in every memory device (Flash or ARAM/DRAM). The number and type of connected devices are defined by the CFG command. *Flash Device Size (Mbits)* — *Number of Blocks Value* 4 — 128 8 — 256 16 — 512 *ARAM/DRAM Device Size (Mbits)* — *Number of Blocks Value* 16 — 508	128
63	Memory Size for Testing NUM_OF_BLOCKS_FOR_TEST	Defines the nubber of blocks (each block is of 4096 bytes) in every memory device (Flash or ARAM/DRAM) for production line testing purposes. The number should be small to minimize testing time during the production sequence. However, the number of blocks should be larger than the number of expected bad blocks in the memory device. In case of value=0, no productiontest is performed. In any case other than value= 0, the number of blocks is defined by the parameter value, and a production testing cycle is performed after RESET. *Legal values: 0 to 128.* *Note: If power fails during production testing cycle, the memory status is unpredicted. The memory device should be replaced and the production test should be repeated.*	0
64	ARAM Quality Level: MAX_DEFECT_NIBBLES_IN_BLOCK	*Defines the maximum allowed bad nibbles in ARAM block (each block is of 8192 nibbles). A nibble (4 bits) is considered bad if any bit is defected.* If the number of bad nibbles in a block exceeds the maximum allowed value, the block is marked as bad block and is not used for voice storage. *Legal values: 0 to 255.*	0

Table 2-11: TUNABLE PARAMETERS: Codec Support (Samples)

Index	Parameter Name	Description	Default
65	Channel 0 Delay: CFRD0	*The delay of codec channel 0 from Frame Synch 0 (CFS0) to start of valid data.* Legal values: 0 to 255	1
66	Channel 1 Delay: CFRD1	*The delay of codec channel 1 from Frame Synch 0 (CFS0) to start of valid data.* Legal values: 0 to 255	10
67	Channel 2 Delay: CFRD2	*The delay of codec channel 2 from Frame Synch 0 (CFS0) to start of valid data.* Legal values: 0 to 255	10
68	Frame Synch Delay: CFSD	*The delay of Frame Synch 1 (CFS1) from Frame Synch 0 (CFS0).* Legal values: 0 to 255	10
69	Data Valiid Delay: CFET	*The delay between Frame Synch 0 (CFS0) to end of valid data of all channels.* Legal values: 0 to 255	18

Example

TUNE 17 02BC					
Byte sequence:	Microcontroller	15	17	02	BC
	VoiceDSP	15	17	02	BC
Description:	Set the minimum period for busy detection to 700 (7 seconds).				

VC Volume Control *vol_level*

Controls the energy level of all the output generators (playback, tone generation, and voice synthesis), with one command. The resolution is ±3 dB.

The actual output level is composed of the tunable level variable, plus the *vol_level*. The valid range for the actual output level of each output generator is defined in Table 2-5.

For example, if the tunable variable VCD_LEVEL (parameter number 21) is 6, and *vol_level* is –2, then the output level equals VCD_LEVEL + *vol_level* = 4.

Example

VC 04			
Byte sequence:	Microcontroller	28	04
	VoiceDSP	28	04
Description:	Set the volume level to VCD_LEVEL + 4.		

WMSG Write Message *data*

Writes 32 bytes of data to the current position of the message pointer, and advances the message pointer by 32 bytes.

If the VoiceDSP processor is in the IDLE state, the command opens the current message, switches the VoiceDSP processor to the MSG_OPEN state, sets the message pointer to the beginning of the message data, and writes the 32 bytes of data.

To add data at the end of an existing message, issue the SMSG command to the last page of the message. Issue the WMSG command with a buffer consisting of 32 FF bytes (this has no effect on the current data in the page). A subsequent WMSG command adds a new block to the message, and writing continues at the beginning of the new block.

The microcontroller must issue an S command to close the message and switch the VoiceDSP processor to the IDLE state.

NOTE When updating an existing message, bits can only be cleared, but not set.

If the current message is undefined, ERR_INVALID is reported.

Example

WMSG 32 bytes			
Byte sequence:	Microcontroller	31	32 bytes of data to write
	VoiceDSP	31	echo 32 bytes of data
Description:	Write 32 bytes in the message memory.		

Chapter 3—SCHEMATIC DIAGRAMS

3.1 APPLICATION INFORMATION

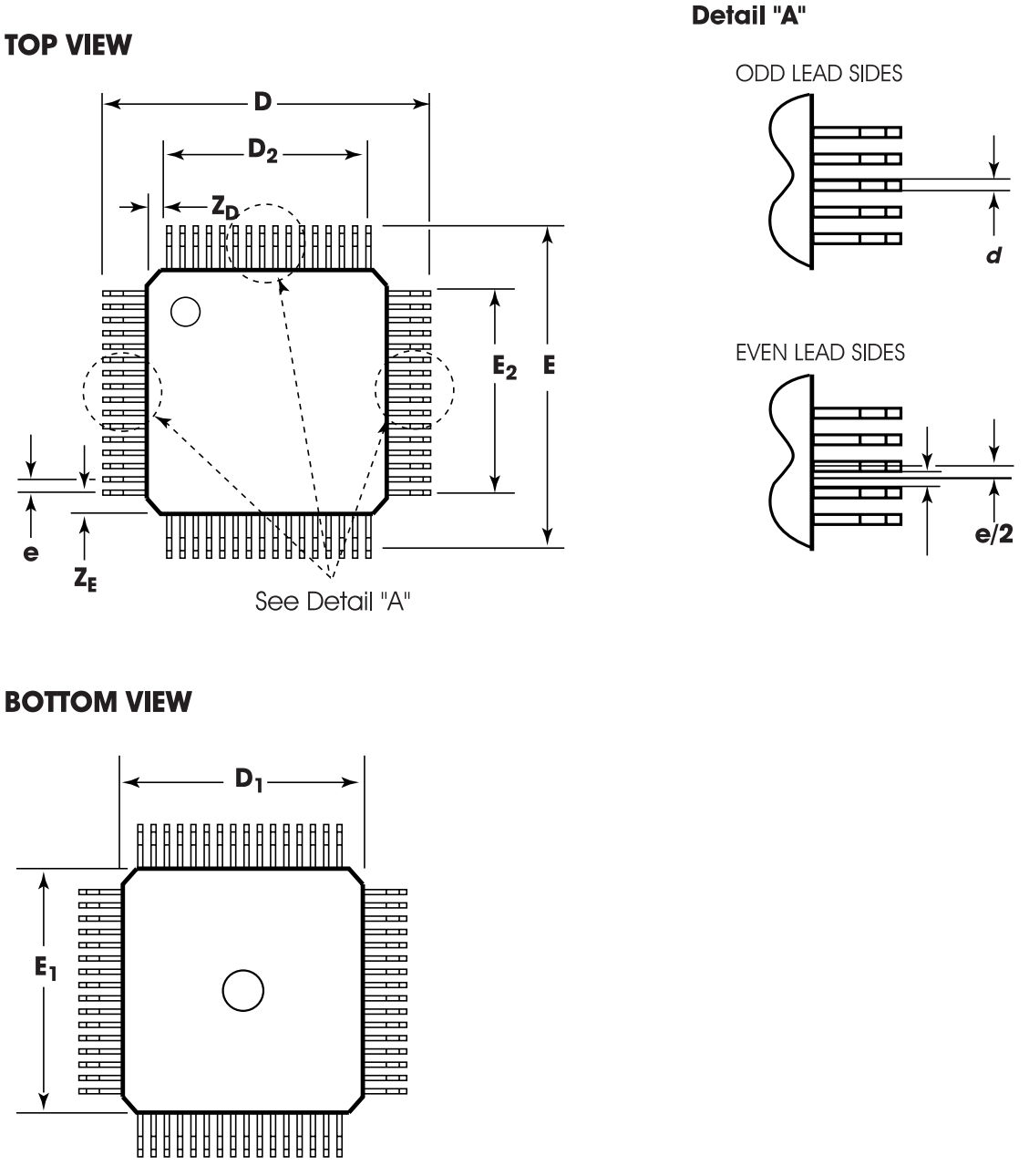
The following schematic diagrams for a VoiceDSP processor Reference design unit.

This reference design includes three basic clusters:

- An 80C51 MicroController.
- VoiceDSP processor cluster, including a TP3054 codec, and an ISDT360SB controlling a Flash device.
- User interface that includes one 16-digit LCD, and a 16-key (4 x 4) keypad.

Chapter 4—PHYSICAL DIMENSIONS

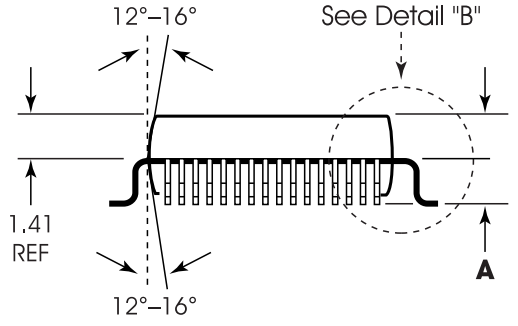
Figure 4-1: 80-Pin Plastic Quad Flat Package, Top and Bottom—Type: Metric PQFP, 14x14 Body



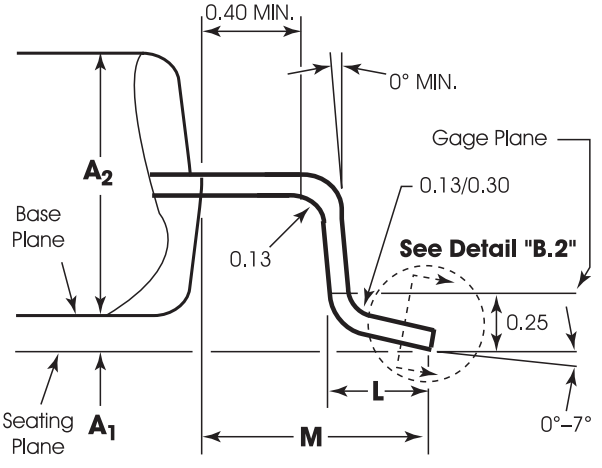
1. *All dimensions are in millimeters. All dimensions and tolerances conform to ANSI Y14.5-1982.*

Figure 4-2: 80-Pin Plastic Quad Flat Package, Side—Type: Metric PQFP, 14x14 Body

SIDE VIEW



Detail "B"



Detail "B.2"

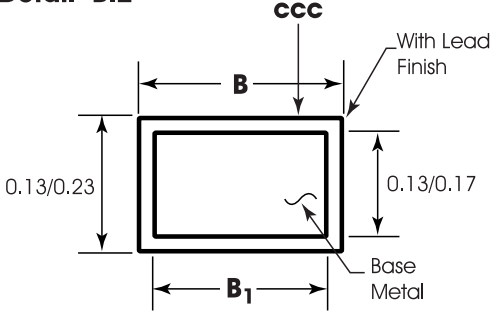


Table 4-1: Packaging Dimensions

Symbol	Min.	Nom.	Max.
A	—	2.82	3.00
A_1	0.10	0.15	0.25
A_2	2.55	2.67	2.75
D		17.20 BSC.	
D_1		14.00 BSC.	
D_2		12.35BSC.	
Z_D		0.825 REF.	
E		17.20 BSC.	
E_1		14.00 BSC.	
E_2		12.35 BSC.	
Z_E		0.825 REF.	
L	0.73	0.88	1.03
N		80	
e		0.65 BSC.	
b	0.22		0.38
b_1	0.22	0.30	0.33
ccc		0.12	

IMPORTANT NOTICES

The warranty for each product of ISD (Information Storage Devices, Inc.), is contained in a written warranty which governs sale and use of such product. Such warranty is contained in the printed terms and conditions under which such product is sold, or in a separate written warranty supplied with the product. Please refer to such written warranty with respect to its applicability to certain applications of such product.

These Product may be subject to restrictions on use. Please contact ISD, for a list of the current additional restrictions on these Product. By purchasing these Product, the purchaser of these Product agrees to comply with such use restrictions. Please contact ISD for clarification of any restrictions described herein.

ISD, reserves the right, without further notice, to change the ISD ChipCorder product specifications and/or information in this document and to improve reliability, functions and design.

ISD assumes no responsibility or liability for any use of the ISD ChipCorder Product. ISD conveys no license or title, either expressed or implied, under any patent, copyright, or mask work right to the ISD ChipCorder Product, and ISD makes no warranties or representations that the ISD ChipCorder Product are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Application examples and alternative uses of any integrated circuit contained in this publication are for illustration purposes only and ISD makes no representation or warranty that such applications shall be suitable for the use specified.

The 100-year retention and 100K record cycle projections are based upon accelerated reliability tests, as published in the ISD Reliability Report, and are neither warranted nor guaranteed by ISD.

Information contained in this ISD ChipCorder data sheet supersedes all data for the ISD ChipCorder Product published by ISD prior to December, 1998.

This data sheet and any future addendum to this data sheet is (are) the complete and controlling ISD ChipCorder product specifications. In the event any inconsistencies exist between the information in this and other product documentation, or in the event that other product documentation contains information in addition to the information in this, the information contained herein supersedes and governs such other information in its entirety.

2045 Hamilton Ave.
San Jose, California 95125-5904
Tel: 408/369-2400
Fax: 408/369-2422
http://www.isd.com

Part No. 2201298D5008