National Semiconductor

PRELIMINARY
November 1995

NSAM265SR/NSAM265SF CompactSPEECH™ Digital Speech Processors

General Description

The NSAM265SR and the NSAM265SF are members of National Semiconductor's CompactSPEECH, Digital Speech processors family. These processors provide Digital Answering Machine (DAM) functionality to embedded systems. Both processors are based on the NSAM265.

Unless specified otherwise, all references to the CompactSPEECH processor in this document apply to both the NSAM265SR and the NSAM265SF.

The CompactSPEECH processor integrates the functions of a traditional Digital Signal Processing (DSP) chip and a 16-bit CompactRISC™ embedded Risc processor core. The device contains system support functions such as DRAM Controller, Interrupt Control Unit, Codec Interface, MICROWIRE™ interface, WATCHDOG™ timer and a Clock Generator.

The CompactSPEECH processor operates as a slave peripheral that is controlled by an external microcontroller via a serial MICROWIRE interface. In a typical DAM environment the microcontroller controls the analog circuits, buttons and display, and activates the CompactSPEECH by sending it commands. The CompactSPEECH processor executes the commands and returns status information to the microcontroller.

The CompactSPEECH firmware implements voice compression and decompression, tone detection and generation, message storage management, on-chip speech synthesis for time and day stamp, and support for user-defined voice prompts in various languages.

The NSAM265SR CompactSPEECH supports DRAM/ARAM for message storage while the NSAM265SF supports FLASH/AFLASH. In all other respects, the processors are identical.

The CompactSPEECH implements echo cancellation techniques to support improved DTMF tone detection during message playback.

The CompactSPEECH supports speech synthesis: the technology used to create voice prompts from predefined words and phrases stored in a vocabulary.

The CompactSPEECH can synthesize messages in various languages, in addition to the on-chip English vocabulary, via the International Vocabulary Support (IVS) mechanism. Synthesized messages can be stored on an external ROM. One ROM can contain several vocabularies in various languages. The NSAM265SF can also store vocabularies on FLASH memory. DAM manufacturers can thus create machines that ''speak'' in different languages, simply by using other vocabularies. For more details about IVS, refer to the *IVS User's Manual.*

Features

- Designed around National's 16-bit CompactRISC processor
- 16-bit architecture and implementation
- 20.48 MHz operation
- On-chip DSP Module (DSPM) for high speed DSP operations
- On-chip Codec clock generation and interface
- Power-down mode
- MICROWIRE interface to an external microcontroller
- Storage and management of messages
- Programmable message tag for message categorization, e.g., Mailboxes, InComing Messages (ICM), OutGoing Messages (OGM)
- Skip forward or backward during message playback
- Variable speed playback
- Built-in vocabulary for speech synthesis, and support for external vocabularies. using expansion ROM
- Multi-lingual speech synthesis using International Vocabulary Support (IVS)
- DTMF and single tone generation and detection
- DTMF tone detection during OutGoing Message playback
- Telephone line functions, including busy and dial tone detection
- Real-time clock
- Direct access to message memory
- Supports long-frame and short-frame codecs
- Available in PLCC 68-pin, and PQFP 100-pin packages

NSAM265SR only

- On-chip ARAM/DRAM Controller for 4-Mbit (1M x 4) and 16-Mbit (4M x 4) devices
- 15 minutes recording on a 4-Mbit ARAM
- Supports various ARAM configurations. No glue logic required
- Storage of up to 1600 messages
- Production diagnostics support

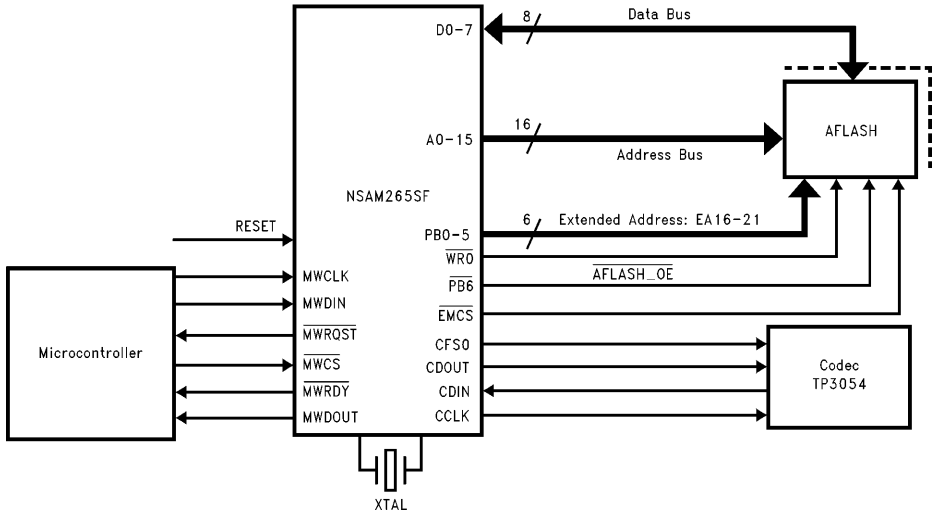
NSAM265SF only

- Supports 4-Mbit and 8-Mbit, byte wide, FLASH/AFLASH devices
- Up to 15 minutes recording on a 4-Mbit FLASH
- Supports various AFLASH configurations. No glue logic required for a single AFLASH configuration
- The number of messages that can be stored is limited only by memory size
- Supports prerecorded IVS and OGM on FLASH

RRD-B30M115/Printed in U. S. A.

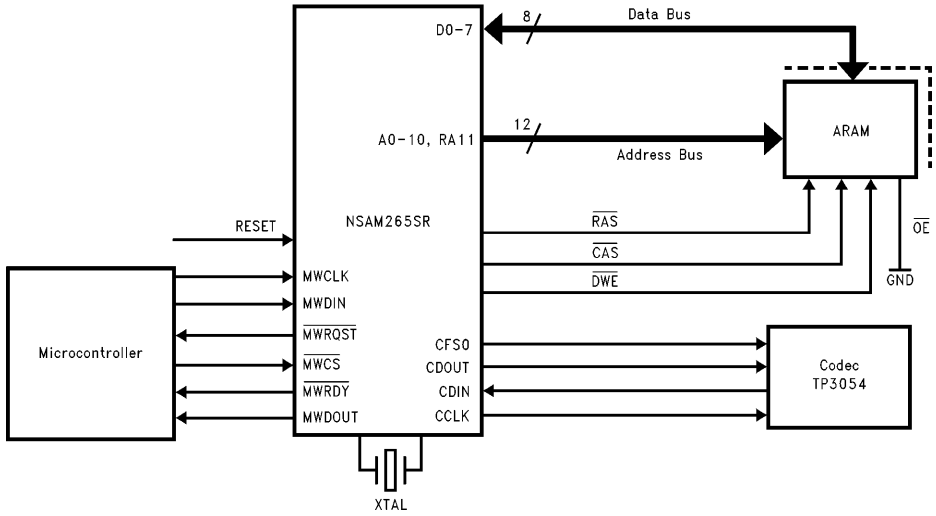
Block Diagrams

NSAM265SF Basic Configuration



TL/EE/12378–1

NSAM265SR Basic Configuration



TL/EE/12378–2

Table of Contents

1.0 Theory of Operation

1.1 OVERVIEW

The CompactSPEECH is a digital speech processor, which provides Digital Answering Machine (DAM) functionality to embedded systems, such as fax machines or stand-alone answering machines.

The CompactSPEECH processor is based on a powerful 16-bit RISC CPU with Digital Signal Processing (DSP) module. However, since the CompactSPEECH firmware is masked in the internal ROM, it requires neither 16-bit nor DSP programming.

The CompactSPEECH processor is designed to operate as a slave peripheral that is controlled by an external microcontroller via a simple, MICROWIRE based, serial link. The microcontroller is responsible for system-level control (e.g., buttons, display, ring detection) while the CompactSPEECH is responsible for speech and tone operations such as recording, playback, tone detection and speech synthesis.

The NSAM265SR CompactSPEECH supports DRAM and Audio grade DRAM (ARAM) as a storage device. ARAMs are cheaper than DRAMs, which reduces the total system cost.

The NSAM265SR CompactSPEECH has an on-chip DRAM controller, which reduces the chip count for complete DAM functionality to three (the CompactSPEECH. ARAM and a codec). The CompactSPEECH supports various ARAM configurations, and automatically recognizes the actual configuration.

The NSAM265SF CompactSPEECH supports FLASH and Audio grade FLASH (AFLASH) as a storage device. A DAM design, which incorporates FLASH technology, does not require a battery backup for the message storage device, thus reducing the system cost and complexity.

When necessary, the CompactSPEECH can be switched to power-down mode, which considerably reduces the power consumption of the whole system.

The CompactSPEECH includes an on-chip English vocabulary for time-and-day stamp announcements and voice prompts. In addition to the on-chip vocabulary, the CompactSPEECH supports external vocabularies (resident on external ROM or AFLASH) which can be used to implement voice prompts in various languages. Several vocabularies can be supported simultaneously.

This chapter describes the features of the CompactSPEECH processor and how they work together.

1.2 THE STATE MACHINE

The CompactSPEECH functions as a state machine. 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).

The CompactSPEECH may be in one of the following states:

RESET

The CompactSPEECH is initialized to this state after a full hardware reset by the $\overline{\text{RESET}}$ signal (see Section 2.2). CompactSPEECH detectors (VOX, call progress tones and DTMF tones) are not active. In all other states, the 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 CompactSPEECH starts executing the command.

PLAY

In this state a message is decompressed, and played.

RECORD

In this state a message is compressed, and recorded into the message memory.

SYNTHESIS

An individual word or a sentence is synthesized from the on-chip, or an external, vocabulary.

TONE__GENERATE

The CompactSPEECH generates single or DTMF tones.

MEMORY__READ

The CompactSPEECH reads a 32-byte block from the message memory and sends it to the external microcontroller.

MEMORY__WRITE

The CompactSPEECH accepts a 32-byte block from the external microcontroller and writes it to the message memory.

MEMORY__FREE

The CompactSPEECH takes memory space that was freed by the DM and DMS commands, and makes it available for new messages. This process occurs only with NSAM265SF where FLASH memory is used for message storage.

After receiving an asynchronous command, (see Section 1.3) such as P (Playback), R (Record), SW (Say Words) or GT (Generate Tone), the CompactSPEECH switches to the appropriate state and executes the command until it is completed, or an S (Stop) or PA (Pause) command is received from the microcontroller.

When an asynchronous command execution is completed, the CompactSPEECH switches to the IDLE state.

Table 1-1 shows the CompactSPEECH commands, the source states in which these commands are valid, and the result states which the CompactSPEECH enters as a result of the command.

1.0 Theory of Operation (Continued)

TABLE 1-1. CompactSPEECH States and Transitions

Command	Description	Source State	Result State
Configuration and Status Commands			
AMAP	Check and Map ARAM	IDLE	IDLE
CFG	Configure CompactSPEECH	RESET	RESET
CVOC	Check Vocabulary	IDLE	IDLE
GEW	Get Error Word	All states	No change
FR	Free Memory	IDLE	MEMORY__FREE
GCFG	Get Configuration	RESET, IDLE	No change
GI	Get Information	PLAY, RECORD, SYNTHESIS, TONE__GENERATE, IDLE, MEMORY__FREE	No change
GMS	Get Memory Status	IDLE	IDLE
GSW	Get Status Word	All states	No change
GTD	Get Time and Day	IDLE	IDLE
INIT	Initialize System	RESET, IDLE	IDLE
INJ	Inject IVS Data	IDLE	IDLE
MR	Memory Reset	IDLE	IDLE
PDM	Go to Power-Down Mode	IDLE	IDLE
RDET	Reset Detectors	MEMORY__FREE, IDLE	No change
S	Stop	All states but RESET	IDLE
SDET	Set Detectors Mask	IDLE, MEMORY__FREE	No change
SETD	Set Time and Day	IDLE, MEMORY__FREE	No change
SV	Set Vocabulary Type	IDLE	IDLE
TUNE	Tune	IDLE, MEMORY__FREE	No change
Speech Commands			
AMSG	Append to Current Message	IDLE	RECORD
GT	Generate Tone	IDLE	TONE__GENERATE
P	Playback	IDLE	PLAY
PA	Pause	PLAY, RECORD, SYNTHESIS, TONE__GENERATE , IDLE*	No change
R	Record	IDLE	RECORD
RES	Resume	PLAY, RECORD, SYNTHESIS, TONE__GENERATE, IDLE*	No change
SAS	Say Argumented Sentence	IDLE	SYNTHESIS
SB	Skip Backward	PLAY, IDLE*	No change
SE	Skip to End of Message	PLAY, IDLE*	No change
SF	Skip Forward	PLAY, IDLE*	No change
SO	Say One Word	IDLE	SYNTHESIS
SPS	Set Playback Speed	PLAY, SYNTHESIS, IDLE, MEMORY__FREE	No change
SS	Say Sentence	IDLE	SYNTHESIS
SW	Say Words	IDLE	SYNTHESIS
VC	Volume Control	PLAY, SYNTHESIS, IDLE, TONE__GENERATE	No change

1.0 Theory of Operation (Continued)

TABLE 1-1. CompactSPEECH States and Transitions (Continued)

Command	Description	Source State	Result State
Message Management Commands			
CMT	Cut Message Tail	IDLE	IDLE
DM	Delete Message	IDLE	IDLE
DMS	Delete Messages	IDLE	IDLE
GL	Get Length	IDLE	IDLE
GMT	Get Message Tag	IDLE	IDLE
GNM	Get Number of Messages	IDLE	IDLE
GTM	Get Tagged Message	IDLE	IDLE
RRAM	Read RAM	IDLE, MEMORY__READ	MEMORY__READ
SMT	Set Message Tag	IDLE	IDLE
WRAM	Write RAM	IDLE, MEMORY__WRITE	MEMORY__WRITE

*Command is valid in IDLE state, but has no effect.

1.3 COMMAND EXECUTION

A CompactSPEECH command is represented by an 8-bit opcode. Some commands have parameters, and some have a return value. Commands are either synchronous or asynchronous.

Synchronous Commands

A synchronous command completes execution before the microcontroller can send a new command (e.g., GMS, GEW).

A command sequence starts when the microcontroller sends an 8-bit opcode to the CompactSPEECH, followed by the command's parameters (if any).

The CompactSPEECH executes the command and, if required, transmits a return value to the microcontroller. Upon completion, the CompactSPEECH 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).

The Status Word

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

The 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 that the $\overline{\text{MWRQST}}$ signal is active, it should issue the GSW (Get Status Word) command which deactivates the $\overline{\text{MWRQST}}$ signal. Then it should test the EV__ERROR bit In the status word, and, if it is set, send the GEW (Get Error Word) command to read the error word for details of the error.

For a detailed description of each of the CompactSPEECH commands, see Section 3.0

1.4 TUNABLE PARAMETERS

The CompactSPEECH processor can be adjusted to your system's requirements. For this purpose the CompactSPEECH supports a set of tunable parameters, which 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 CompactSPEECH's operation, such as silence compression, tone detection, no-energy detection, etc.

Table 3-1 describes all the tunable parameters in detail. Section 3 describes the TUNE command.

1.5 MESSAGES

The CompactSPEECH message manager supports a wide range of applications, which require different levels of DAM functionality.

The message-organization scheme and the message tag, support advanced memory-organization features such as multiple OutGoing Messages (OGMs), mailboxes and the ability to distinguish between InComing Messages (ICMs) and OGMs.

The NSAM265SF can store up to 256 messages per 4 Mbits of AFLASH storage. The NSAM265SR can store up to 100 messages per 4 Mbits of ARAM storage.

A message is the basic unit on which most of the CompactSPEECH commands operate. A CompactSPEECH message, stored in ARAM or AFLASH, can be regarded as a computer file stored on a mass-storage device.

A message is created with either the R or the WRAM (Write RAM) command. When a message is created, it is assigned a time-and-day stamp and a message tag which can be read by the microcontroller.

The R command takes voice samples from the codec, compresses them, and stores them in the message memory.

1.0 Theory of Operation (Continued)

When a message is created with the WRAM command the data to be recorded is provided by the microcontroller and not the codec. The data is transferred directly to the message memory. It is not compressed by the CompactSPEECH voice compression algorithm.

The WRAM command, together with the RRAM (Read RAM) command which enables the microcontroller to read data from the CompactSPEECH, can be used to store data other than compressed voice in the message memory. For example, in the NSAM265SF the AFLASH memory can be used to store a telephone directory.

A message can be played back (P command) and deleted (DM command). Redundant data (e.g., trailing tones or silence) can be removed from the message tail with the CMT (Cut Message Tail) command.

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

Current Message

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

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

1.5.1 Message Tag

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

In the NSAM265SR bits 0–6 are application definable; bits 7–15 are reserved.

In the NSAM265SF bits 0–14 are application definable; bit 15 is the MESSAGE__SAFE bit. The MESSAGE__SAFE bit should be used to record safe (non-volatile) messages (e.g., OGMs) when the NSAM265SF is configured to memory-intensive mode (see Section 1.7.3).

For memory-management reasons, the NSAM265SF must keep at least one FLASH block without safe messages. If there is no such block available during recording of a safe message, or when the microcontroller tries to create a new message with the R or WRAM command, recording stops and the EV__MEMFULL bit in the status word is set.

The GMT (Get Message Tag) and SMT commands may be used to handle message tags.

Note: For the NSAM265SF, message tag bits can only be cleared. Message tag bits are set only when a message is first created. This limitation is inherent in FLASH memories, which only allow bits to be changed from 1 to 0 (changing bits from 0 to 1 requires a special erasure procedure, see Section 1.7.1). However, the main reason for updating an existing tag is to mark a message as old, and this can be done by using one of the bits as a new/old indicator, setting it to 1 when a message is first created, and clearing it when necessary.

1.6 ARAM SUPPORT

The NSAM265SR supports up to two 4-Mbit (1M × 4) ARAM devices or one 16-Mbit (4M x 4) device for storing messages. An ARAM device is actually a bad DRAM, device i.e., it may contain bad bits. The NSAM265SR can use such devices for message recording, without noticeable effect on voice quality, if they conform to the specifications described in Section 2.6.

After an ARAM mapping process (see the AMAP command) which marks bad ARAM rows which can not be used for recording, the NSAM265SR can use the rest of the ARAM space for message recording. A single 1-Mbit × 4 ARAM device holds an average of 15 minutes of recording time, (actual time may vary because of environmental conditions e.g., speech attributes, background noise etc.).

1.7 FLASH SUPPORT

The NSAM265SF CompactSPEECH supports 4-Mbit and 8-Mbit, byte wide, AFLASH devices for storing messages.

A FLASH device is organized in 64 Kbytes blocks. An AFLASH device is a FLASH device, with one or more bad blocks which can not be used for message recording. The NSAM265SF can use such devices for message recording without any affect on voice quality, if they conform to the specifications described in Section 2.9.

There are two major problems imposed by current FLASH technology: block erasure time, and FLASH endurance. Both these limitations are handled by the CompactSPEECH firmware.

1.7.1 Block Erasure

The basic software interface to a DRAM device includes read and write operations. Writing a value to a memory location simply replaces its contents.

In a FLASH environment, an erase operation is also required. You must ensure that a memory location, which was previously written, is erased prior to writing. The basic unit that can be read, or written, is a byte; the basic unit that can be erased is an entire 64 Kbytes block.

Block erasure takes time. The following erasure times are quoted from AMD and INTEL datasheets for devices supported by the NSAM265SF:

INTEL	28F008SA	1.6 sec (typical)	10 sec (max)
AMD	AM29F040	1.5 sec (min)	30 sec (max)

A FLASH memory can not be written while erasure is in progress. During erasure, access to the FLASH is not allowed. The CompactSPEECH, however, accepts commands which do not require FLASH access (e.g., Get Status) during erasure.

1.7.2 Flash Endurance

FLASH memories may be erased a limited number of times. Currently, FLASH manufacturers do not guarantee more than 100,000 erase cycles.

To reduce the effect of this limitation, the memory manager utilizes FLASH 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.

Consider the following extensive usage of all FLASH blocks:

1. Record 15 minutes of messages (until the memory is full).

2. Playback 15 minutes (all the recorded messages).

3. Delete all messages.

Assuming a 4-Mbit FLASH device is used in this manner 24 times a day, the expected life time of the FLASH is:

Flash Lifetime = 100,000 / (24 * 365) = 11.4 years

Thus the FLASH device will last for over ten years, even when used for six hours of recording per day.

Note, that if two 4-Mbit devices are used, then, under the same conditions, each device will last for more than 20 years.

1.7.3. Memory Operating Modes in NSAM265SF

The NSAM265SF supports two operating modes of the FLASH memory manager (selected by the CFG command):

Normal Mode

In this mode, the NSAM265SF always keeps one free block for memory management. The EV__MEMFULL event is set when all but one good AFLASH blocks are full, and recording on the last available free block is not allowed.

Maximum recording time in this mode:

On one 8-Mbit FLASH (no bad blocks) device: 28 minutes and 8 seconds.
On one 4-Mbit FLASH (no bad blocks) device: 13 minutes and 8 seconds.

Memory Intensive Mode

All good blocks are available for message recording. As long as there is one free block available for memory management, this mode is the same as normal mode. The CompactSPEECH sets the EV__MEMLOW event when approximately 20 seconds recording time remain on the one-before-last FLASH block. The EV__MEMFULL event is set only when all the blocks are full.

However, if the last good block is used for recording, the NSAM265SF no longer has a free block for memory management, and therefore may fail to free memory space i.e., execute the FR command. The microcontroller must delete all messages, with the MESSAGE__SAFE bit cleared, to guarantee that at least one block is erased, and can be used for memory management.

When there is no block available for memory management, and you attempt to record a message with the MESSAGE__SAFE bit set, the CompactSPEECH sets the EV__MEMFULL event and does not allow you to record. Thus it is possible to avoid a situation where a safe message must be deleted to free a FLASH block.

Maximum recording time in this mode:

On one 8 Mbits FLASH (no bad blocks) device: 30 minutes.
On one 4 Mbits FLASH (no bad blocks) device: 15 minutes.

1.8 TONE AND NO-ENERGY DETECTION

The CompactSPEECH detects DTMF, busy and dial tones, and no energy (VOX). This enables remote control operations and call progress. Detection is active throughout the operation of the CompactSPEECH. Detection can be configured using the SDET (Set Detectors Mask) command, which controls the reporting of occurrence of tones, and the RDET (Reset Detectors) command which resets the detectors.

DTMF

DTMF detection may be reported at starting point, ending point, or both. The report is made through the status word (for further details, see GSW command in Section 3.2).

The DTMF detector specifications as measured on the line input using the NSV-AM265-DAA board are summarized below (see Table 1-2).

Echo Cancellation

Echo cancellation is a technique used to improve the performance of DTMF tone detection during speech synthesis, tone generation and OGM playback. For echo cancellation to work properly, AGC must not be active in parallel. Thus, to take advantage of echo cancellation, the microcontroller must control the AGC, i.e., disable the AGC during PLAY, SYNTHESIS and TONE__GENERATE states and enable it again afterwards. If AGC can not be disabled, do not use echo cancellation. The microcontroller should use the CFG command to activate/deactivate echo cancellation. (For further details, see Section 3.2.)

Echo cancellation applies only to DTMF tones. Busy and dial tones detection is not affected by this technique.

TABLE 1-2. DTMF Detector Specifications

	Play	Record/ Idle
Detection Sensitivity (Note A)	−33 with Echo Canceller −23 w/o Echo Canceller	−40 dBm
Accepted DTMF Length	>50 ms	>40 ms
Frequency Tolerance	±1.5%	±1.5%
S/N Ratio	12 dB	12 dB
Minimum Spacing (Note B)	>45 ms	>45 ms
Normal Twist	8 dB	8 dB
Reverse Twist (Note C)	4 dB or 8 dB	4 dB or 8 dB

Note A: Performance depends on DAA design.

Note B: If the interval between two consecutive 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 DTMF tones is between 20 ms and 45 ms, separate detection is unpredictable.

Note C: Determined according to the DTMF__REV__TWIST tunable parameter value.

1.0 Theory of Operation (Continued)

Other Detectors

Detection of busy and dial tones, and no-energy is controlled by tunable parameters. You should tune the thresholds of these parameters to fit your hardware. For more information see the TUNE command in Section 3.2.

Dial and busy tone detectors work with a band pass filter that limits the frequency range in which tones can be detected to 0 Hz−1100 Hz. Its frequency response is illustrated in *Figure 1-1* and the busy tone cadences in *Figure 1-2*.

Tone Generation

The CompactSPEECH can generate DTMF tones and single-frequency tones from 300 Hz to 2000 Hz in increments of 100 Hz. CompactSPEECH tone generation conforms with EIA-470-RS standard. Note however, that you may have to change the value of some tunable parameters in order to meet the standard specifications since the energy level of generated tones depends on the analogue circuits being used.

- Tune the DTMF__TWIST__LEVEL parameter to control the twist level of the generated DTMF tones.
- Use the VC command and tune the TONE__GENERATION__LEVEL parameter to control the energy level at which these tones are generated.
- Use the GT command to specify the DTMF tones and the frequency at which single tones are generated.

1.9 SPEECH SYNTHESIS

Speech synthesis enables you to announce prerecorded voice prompts, or form a sentence by combining individual words. A built-in is supplied with the CompactSPEECH for announcing the number of messages and the time and day.

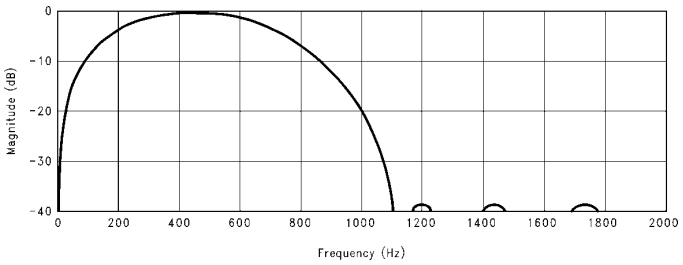
The main speech synthesis features are:

- On-chip English vocabulary for announcing the number of messages and the time and day.
- Additional vocabularies on ROM, or FLASH (NSAM265SF only).
- International Vocabulary Support (IVS) without changing the microcontroller code.
- Up to 220 words in each vocabulary. Each word can be announced separately.
- Sentences announced from a predefined table of sentences.
- On-the-fly sentence announcement.

1.9.1 Explanation of Terms

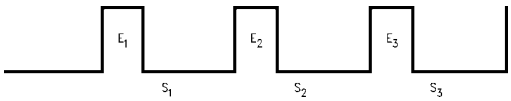
The following terms are used throughout this document:

Vocabulary	A complete set of *words* and *sentences*. *Words* are arranged in a *word table*. *Sentences* are arranged in a *sentence table*.
Word Table	A set of *words*, arranged in a table, as part of a *vocabulary*.
Sentence Table	A set of *sentences*, arranged in a table, as part of a *vocabulary*. The structure of a *sentence table* is described in detail in the *International Vocabulary Support User's Manual*. The Internal Vocabulary includes a built-in table with two entries.
Word	An entry in a *word table* which represents a spoken word or phrase. A *word* is played with the SW (Say Words), or SO (Say One word) commands.
Sentence	A series of *words*, synthesized from a *vocabulary*. A *sentence* is defined as an entry in the *sentence table*, and is synthesized with the SS (Say Sentence) or SAS (Say Argumented Sentence) command, or is synthesized ''on-the-fly'' with the SW command.
Internal Vocabulary	The *vocabulary*, in English, which resides on the CompactSPEECH's internal ROM.
External Vocabulary	An optional *vocabulary* which resides on external ROM or (NSAM265SF only) FLASH.
International Vocabulary Support (IVS)	A method that enables the same CompactSPEECH command to synthesize *sentences* with the same meaning, but in different languages, from separate *external vocabularies*.



TL/EE/12378–3

FIGURE 1-1. Busy and Dial Tone Band Pass Filter Frequency Response



TL/EE/12378–47

$[E_1 - E_3] < 90$ ms $[S_1 - S_3] < 90$ ms $100 < E_i < 1700$ ms $50 < S_i < 1300$ ms

FIGURE 1-2. Busy Tone Detector—Cadence Specification

1.0 Theory of Operation (Continued)

1.9.2 Internal Vocabulary

Table 1-3 summarizes the words in the standard internal English speech synthesis vocabulary.

Sentences

Currently, two built-in sentences, 0 and 1, are supported: *Time and Day*, and *You Have*. The CompactSPEECH provides specific commands to synthesize these sentences when an Internal Vocabulary is used. When using External Vocabularies, sentences can be defined in a sentence table, and accessed directly via the SAS command.

Use the SS command to play a sentence without an argument. Use the SAS command to play a sentence with an argument. A sentence can have only one argument.

Table 1-4 summarizes the sentences in the standard internal English speech synthesis vocabulary.

TABLE 1-3. Internal Vocabulary—Word Table

Index	Word	Comment
0	oh	(as in eight oh five pm-8:05 PM)
1	one	
2	two	
3	three	
. . .	. . .	Numbers four through 18
19	nineteen	
20	twenty	(at end of word, as in 8:20 PM)
21	thirty	(at end of word)
22	forty	(at end of word)
23	fifty	(at end of word)
24	twenty	(in middle of word, as in 8:23 PM)
25	thirty	(in middle of word)
26	forty	(in middle of word)
27	fifty	(in middle of word)
28	AM	
29	PM	
30	Monday	
31	Tuesday	
32	Wednesday	
33	Thursday	
34	Friday	
35	Saturday	
36	Sunday	
37	no	(as in: you have no messages)
38	messages	
39	message	(as in: you have one message)
40	you have	
41	end of messages	

Sentences

Currently, two built-in sentences, 0 and 1, are supported: *Time and Day*, and *You Have*. The CompactSPEECH provides specific commands to synthesize these sentences when an Internal Vocabulary is used. When using External Vocabularies, sentences can be defined in a sentence table, and accessed directly via the SAS command.

Use the SS command to play a sentence without an argument. Use the SAS command to play a sentence with an argument. A sentence can have only one argument.

TABLE 1-4. Internal Vocabulary—Sentence Table

Index	Sentence	Argument	Comment
0	Time and Day	*time__day__option*	Announces one of two sentences, according to the argument: If *time__day__option* is 0, synthesize the actual current time and day. If *time__day__option* is not 0, synthesize the current message time and day stamp.
1	You Have	*num__of__msgs*	Announces the number of messages according to the argument. The 1-byte *num__of__msgs* may be any value from 0 through 59. If *num__of__msgs* is 0, the word *no* is synthesized instead of a number.

1.0 Theory of Operation (Continued)

1.9.3 External (International) Vocabularies

Tools for creation of external vocabularies are available. With these tools voice format files can be compressed, and both numbers and sentences can be composed to comply with the grammar of a specific language.

The CompactSPEECH supports external vocabularies which you can easily tailor for country-specific applications. Every language has its own sentence structure, and its own mechanism for composing numbers. Therefore, the information on the sentence structure and number composition is a part of the external vocabulary. A method, IVS, has been developed which uses this information to compose a complete sentence.

The information stored in the external vocabulary, together with this retrieval method, can be used to compose sentences or phrases in various languages, or to implement a voice menu, or command voice prompts. The additional vocabulary can reside on either external ROM or (NSAM265SF only) AFLASH.

IVS enables you to have the same program on the controller to support operation with several languages. You have only to switch to another table, containing another language, and the machine ''speaks'' the new language.

The Rule

Before using a specific table, set the currently used vocabulary with the command SV (Set Vocabulary Type). Until the next invocation of this command, the selected vocabulary is used when invoking any synthesis command.

IVS Structure

It is possible to have several vocabularies on one external ROM or FLASH device, and the controller can switch between them.

1.10 INITIALIZATION

Use the following procedures to initialize the CompactSPEECH processor:

Normal Initialization

1. Reset the CompactSPEECH by activating the $\overline{\text{RESET}}$ signal. (See Section 2.2.)
2. Issue a GSW command, and check that the EV__RESET bit in the status word is set.
3. Issue a GCFG (Get Configuration) command to figure out what CompactSPEECH knows about your environment.
4. Issue a CFG (Configure CompactSPEECH) command to change the configuration according to your environment.
5. Issue an INIT (Initialize System) command to initialize the CompactSPEECH.
6. Issue a GMS (Get Memory Status) command to determine the size of the memory.

   (Optional, NSAM265SF only) Issue an FR command to free potentially available memory.
7. Issue a AMAP (Check and Map ARAM) 3 command to test and map the ARAM (Required only on NSAM265SR).
8. Use the TUNE command to set all hardware parameters.

Production Test (for the NSAM265SR only)

To save time on the production line of the final product, a set of diagnostics is available. Production testing is the primary use of the AMAP command.

The following procedure is recommended:

1. Reset the CompactSPEECH by activating the $\overline{\text{RESET}}$ signal. (See Section 2.2.)
2. Issue a GSW command, and check that the EV__RESET bit in the status word is set.
3. Issue a GCFG (Get Configuration) command to figure out what CompactSPEECH knows about your environment.
4. Issue a CFG (Configure CompactSPEECH) command to change the configuration according to your environment.
5. Issue an INIT command.
6. Issue a AMAP 0 command to verify that the correct number of ARAMs are connected.
7. Issue a AMAP 1 command as part of the tests to verify connectivity to all ARAMs.
8. Record and playback a short message (up to 5 seconds) as part of the tests to verify connectivity and functionality of the codec interface.
9. If you use an external vocabulary, choose the appropriate table and playback a synthesized sentence (e.g., ''*You have no messages*'') as part of the tests to check connectivity to the ROM(s) holding the External vocabulary.

2.0 Functional Description

2.1 INTRODUCTION

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

Resetting
The serial interface
Codec interface
Memory (DRAM/FLASH) accesses
Memory (DRAM/FLASH) configurations
IVS access
Clocking
Power-down mode
Power and grounding

The processor signals mentioned in this section are described in Appendix A.

2.2 RESETTING

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

On application of power, $\overline{\text{RESET}}$ must be held low for at least 30 ms after V_{CC} is stable. This is to ensure 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 30 ms. During these 30 ms, and for 100 μs after, the $\overline{\text{TST}}$ signal must be high. This can be done by putting a pull-up resistor on the $\overline{\text{WRO}}/\overline{\text{TST}}$ pin.

The value of $\overline{\text{MWRDY}}$ is undefined during the 30 ms reset period and for 100 μs 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 operation 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.

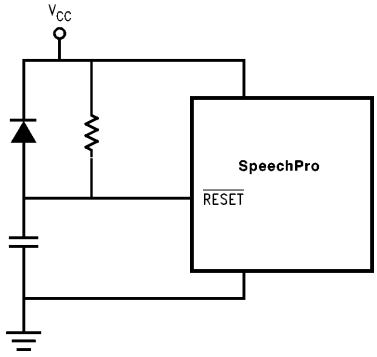
After reset, the same pin is used for $\overline{\text{EMCS}}$.

System Load on ENV0

The load connected to the ENV0 pin should not allow the voltage on ENV0 to drop below V_{ENVh}.

If the load caused by the ENV0 pin exceeds 10 μA, use an external pull-up resistor to keep the pin at 1.

2.0 Functional Description (Continued)



TL/EE/12378–4

FIGURE 2-1. Recommended Power-On Reset Circuit

2.3 THE SERIAL MICROWIRE INTERFACE

The CompactSPEECH supports the MICROWIRE synchronous serial communication protocol. The communication protocol used by the CompactSPEECH is an extension of this protocol scheme. The microcontroller is the protocol master and provides the clock for the protocol. The CompactSPEECH 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 CompactSPEECH and the microcontroller.

Communication is handled in bursts of eight bits (one byte). In each burst the CompactSPEECH is able to receive and transmit eight bits of data. After eight bits have been transferred, an internal interrupt is issued for the CompactSPEECH to process the byte, or to prepare another byte for sending. In parallel the CompactSPEECH sets (1) the $\overline{\text{MWRDY}}$ signal 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 asserted (0) by the CompactSPEECH. When the CompactSPEECH transmits data, it expects to receive the value 0xAA as an echo after each transmitted byte. The CompactSPEECH reports any status change by asserting the $\overline{\text{MWRQST}}$ signal (0).

If a parameter of a CompactSPEECH command is bigger than one byte, the Most Significant Byte (MSB) should be transmitted first by the microcontroller. If a return value is bigger than one byte, the MSB is transmitted first by CompactSPEECH.

Note: Although the CompactSPEECH does not enforce a lower limit on the bit transfer rate, the time between bytes within the same command must not exceed 2 ms.

2.3.1 Signal Description

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

Input Signals

MWDIN

MICROWIRE Data In. Used for input only, for transferring data from the host to the CompactSPEECH.

MWCLK

This signal 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 asserted (0) to indicate that the CompactSPEECH is accessed. Asserting the $\overline{\text{MWCS}}$ causes the CompactSPEECH to start driving the MWDOUT with bit 7 of the transmitted value. Negating the signal resets the transfer-bit counter of the protocol, so it can be used as a synchronization between the CompactSPEECH and the microcontroller.

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

Output Signals

MWDOUT

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

$\overline{\text{MWRDY}}$

MICROWIRE Ready. When active (0), this signal indicates that the CompactSPEECH is ready to transfer (receive or transmit) another byte of data. This signal is deactivated (1) by the CompactSPEECH upon each byte transfer completion. It remains deactivated, while the CompactSPEECH 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 asserted by 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 (1), after the CompactSPEECH 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 CompactSPEECH firmware.

2.3.2 Signal Use in the Interface Protocol

After reset, the $\overline{\text{MWRQST}}$ signal is activated (0) and the $\overline{\text{MWRDY}}$ signal is activated (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 CompactSPEECH reads data from MWDIN on every rising edge of MWCLK. CompactSPEECH 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 mechanism after some propagation delay), or it is a bit of a value the CompactSPEECH 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 one.

2.0 Functional Description (Continued)

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

1. An active (0) $\overline{\text{MWRDY}}$ signal signals the microcontroller that the last eight bits of data transferred to/from the CompactSPEECH were accepted and processed (see below).
2. The $\overline{\text{MWRDY}}$ signal is deactivated (set to 1 by the CompactSPEECH) after 8 bits of data were transferred to/from the CompactSPEECH. The bit is set following the falling edge of the eighth MWCLK clock-cycle.
3. The $\overline{\text{MWRDY}}$ signal is activated (by the CompactSPEECH) when the CompactSPEECH 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 CompactSPEECH is ready to send another byte, or to receive a new command.

   The $\overline{\text{MWRDY}}$ signal is activated (set to 0) after reset and protocol timeout. (See Section 2.3.3.)

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

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

Figure 2-2 illustrates the sequence of activities during a MICROWIRE data transfer.

2.3.3 Interface Protocol Error Handling

Interface Protocol Time-Outs

When the time between two consecutive byte transmissions, or bytes reception within the same command or return value, exceeds two milliseconds after the $\overline{\text{MWRDY}}$ signal is asserted, a time-out event occurs, and the CompactSPEECH 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 (sets it to 0).
4. Activates the $\overline{\text{MWRDY}}$ signal (sets it to 0).
5. Waits for a new command. (After a time-out occurs, the microcontroller must wait at least four milliseconds before issuing the next command.)

Echo Mechanism

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

Note: When a command has a return value, the CompactSPEECH transmits bytes of the return value instead of the echo value.

When the CompactSPEECH is transmitting a byte, it expects to receive the value 0xAA as an echo. Upon detection of an error the CompactSPEECH activates the $\overline{\text{MWRQST}}$ signal, and sets the ERR__COMM bit in the error word.

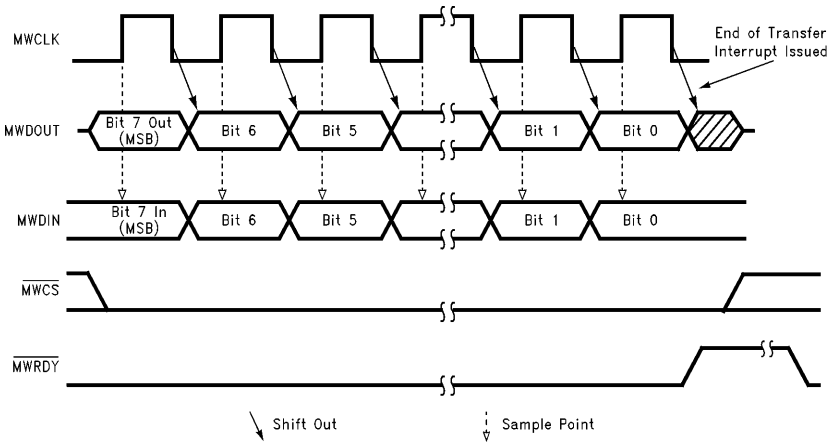
2.4 CODEC INTERFACE

The CompactSPEECH provides an on-chip interface to a serial codec. The interface supports codec operation in long or short-frame formats. The format is selected with the CFG command.

The codec interface uses four signals—CDIN, CDOUT, CCLK and CFS0.

Data is transferred to the codec through the CDOUT pin. Data is read from the codec through the CDIN pin.

Data transfer between the CompactSPEECH and the serial codec starts by the CompactSPEECH asserting (high) the CFS0 frame sync signal. After one clock cycle, the CompactSPEECH de-asserts CFS0, data from the CompactSPEECH is sent to the codec through CDOUT, and simultaneously data from the codec is sent to the CompactSPEECH through CDIN.



TL/EE/12378–5

FIGURE 2-2. Sequence of Activities During a MICROWIRE Byte Transfer

2.0 Functional Description (Continued)

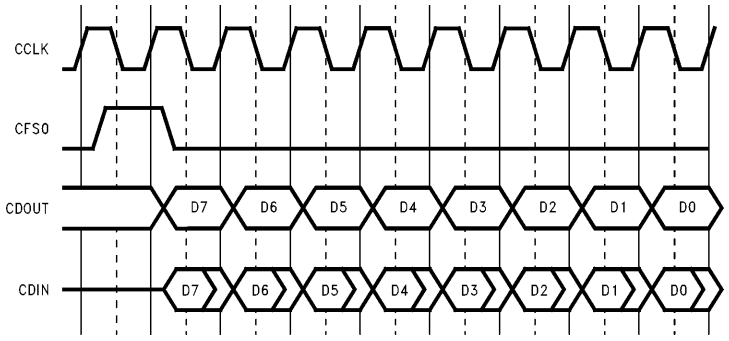


FIGURE 2-3. Codec Protocol - Short Frame

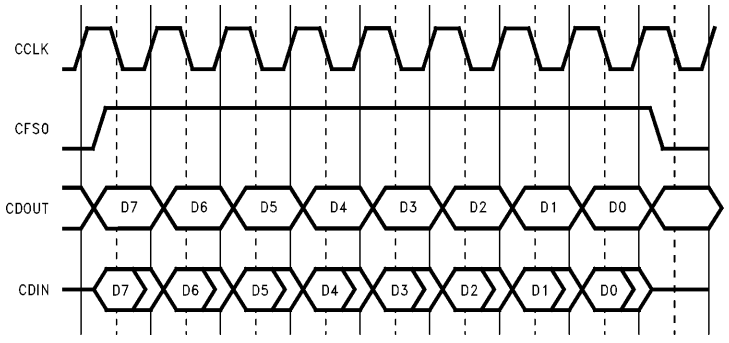


FIGURE 2-4. Codec Protocol - Long Frame

2.5 DRAM/ARAM Access (NSAM265SR)

Reading from DRAM

A read bus cycle starts at T1, when the row address is driven on the address bus and the data bus is in TRI-STATE®. $\overline{DWE}$ is inactive (1) throughout the bus cycle. One idle cycle (TI) is guaranteed before a DRAM bus cycle.

In the next clock cycle (T2W1), $\overline{RAS}$ is asserted (0). T2W2 and T2W3 follow the T2W1 bus cycle. At T2W3 the column address is driven onto the address bus A(0:10) (RA11 is in TRI-STATE). Six T2W cycles and one T2 cycle follow. At the first T2W cycle, the $\overline{CAS}$ signal is asserted (0). At the end of T2, the NSAM265SR samples the data.

The bus cycle is terminated with a T3 cycle, when $\overline{RAS}$ and $\overline{CAS}$ signals become inactive (1). To ensure enough DRAM pre-charge time, the DRAM read bus cycle is separated by at least three clock cycles from any other DRAM bus cycle (read. write or refresh). Other bus cycles can start after at least one TI (Idle) cycle.

For more details, refer to the timing diagram. *Figure A-10*, and AC/DC specifications in ELECTRICAL CHARACTERISTICS.

Writing to DRAM

A write bus cycle starts at T1, when the row address is driven on the address bus and the data bus is in TRI-STATE. One idle cycle (TI) is guaranteed before a DRAM bus cycle.

In the next clock cycle (T2W1), $\overline{RAS}$ is asserted (0) and the data is available on the data bus (except for D2). T2W2 and T2W3 follow the T2W1 cycle. At T2W2, $\overline{DWE}$ is asserted (0) to indicate the write operation.

In the next cycle (T2W3) the column address is driven onto the address bus A(0:10) and RA11. Then six T2W cycles and one T2 cycle follow. At the first T2W cycle, $\overline{CAS}$ is asserted (0).

The bus cycle is terminated by a T3 cycle, when $\overline{DWE}$, $\overline{RAS}$ and $\overline{CAS}$ signals become inactive (1). To provide enough DRAM pre-charge time, the DRAM write bus cycle is separated by at least three clock cycles from any other DRAM bus cycle (read, write or refresh). Other bus cycles can start after at least one TI cycle.

For more details, refer to the timing diagram *Figure A-11* and AC/DC specifications in ELECTRICAL CHARACTERISTICS.

2.5.1 Refreshing a DRAM/ARAM

The NSAM265SR generates DRAM refresh bus cycles in both normal operation and power-down modes. In both cases, a clock, generated by the clock generator sets the refresh rate to $1/384$ of the oscillator frequency.

2.0 Functional Description (Continued)

In Normal Operation Mode

In normal operation mode, a refresh bus cycle starts at T2W1RF, by asserting $\overline{CAS}$ (0). $\overline{DWE}$ stays inactive (1) throughout the transaction. Nine cycles follow the T2W1RF cycle: T2W2RF, T2W3RF, six T2WRF cycles and a T2 cycle. The $\overline{RAS}$ signal is asserted (0) at the first T2WRF cycle.

The transaction is terminated at T3RF, when $\overline{RAS}$ and $\overline{CAS}$ signals become inactive (1). To ensure enough DRAM precharge time, the DRAM refresh bus cycle is separated by at least three clock cycles from any other DRAM bus cycle (read, write or refresh). Other bus cycles can be performed in parallel with a refresh transaction and/or the three cycles following the refresh transaction.

In Power-Down Mode

In power-down mode, a DRAM refresh cycle starts when the $\overline{CAS}$ signal is held asserted (0) for 32 oscillator cycles. $\overline{RAS}$ is asserted (0) 16 oscillator cycles after the $\overline{CAS}$ signal is asserted (0). After another 16 oscillator cycles both the $\overline{RAS}$ and $\overline{CAS}$ signals become inactive (1). In power-down mode, $\overline{DWE}$ stays inactive (1).

In power-down mode, A(0:10) and RA11 stay inactive (0), and D(0:1) and D(3:7) are in TRI-STATE.

2.6 DRAM/ARAM SPECIFICATIONS (NSAM265SR)

The NSAM265SR supports three types of ARAM devices:

- 1M x 4 bits organized in 1024 rows x 1024 columns
- 4M x 4 bits organized in 4096 rows x 1024 columns
- 4M x 4 bits organized in 2048 rows x 2048 columns:

ARAM Device Type	Number of Rows	Clean Rows	Bad Nibbles in One Row
4 Mbits	1024	4 (0 to 3)	<0.5%
16 Mbits	4096	16 (0 to 15)	<0.5%
16 Mbits	2048	4 (0 to 3)	<0.5%

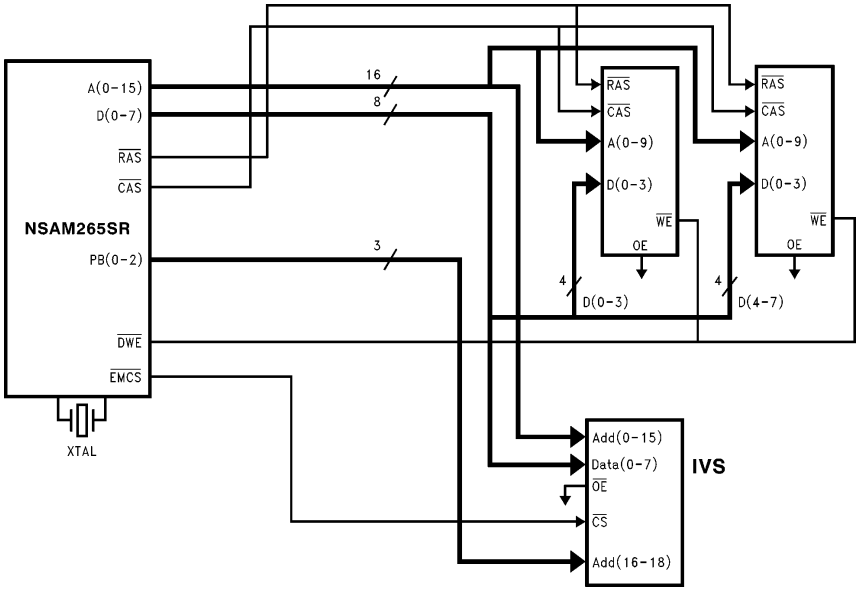
- Clean rows are ARAM rows without any defects.
- An ARAM row containing more bad nibbles than allowed is counted as a bad row.
- A nibble which contains one or more bad bits is counted as one bad nibble.
- The maximum number of bad rows permitted is 200. When the AMAP 3 command is issued, and there are more than 200 bad rows, the ERR__ARAM bit in the error word is set.

2.7 DRAM/ARAM CONFIGURATIONS (NSAM265SR)

During power-up the NSAM265SR automatically determines the memory configuration used for message storage. The AMAP command returns the number of ARAMs.

The following memory configurations are recognized:

Number of ARAMs	Number of Messages	Data Pins	Number of Address Lines ROWS x COLS	Address Lines
1 (4 Mbits)	100	D0–D3	10 x 10	A0–A9
2 (4 Mbits)	400	D0–D7	10 x 10	A0–A9
1 (16 Mbits)	400	D0–D3	12 x 10	A0–A10, RA11
2 (16 Mbits)	1600	D0–D7	12 x 10	A0–A10, RA11
1 (16 Mbits)	200	D0–D3	11 x 11	A0–A10
2 (16 Mbits)	800	D0–D7	11 x 11	A0–A10



TL/EE/12378–8

FIGURE 2-5. Connections for Two 1-Mbit × 4 ARAMs

2.0 Functional Description (Continued)

2.8 FLASH/AFLASH ACCESS (NSAM265SF)

The NSAM265SF CompactSPEECH supports off-chip memory devices through Expansion Memory. Up to 64 Kbytes (64K × 8) of Expansion Memory are supported directly. Nevertheless, the CompactSPEECH uses bits of the on-chip I/O port (PB) to further extend the 64 Kbytes address space, and with additional glue logic it can access up to 2 Mbytes address space.

The Expansion Memory mechanism is used to connect the NSAM265SF CompactSPEECH to AFLASH and IVS ROM devices.

ROM is connected to the CompactSPEECH using the data bus, D(0:7), the address bus, A(0:15), the extended address signals, EA(16:21), (EA21 serves as the ROM Output Enable signal, $\overline{\text{ROM\_OE}}$), and Expansion Memory Chip Select, $\overline{\text{EMCS}}$ controls. If AFLASH is connected (NSAM265SF only), the $\overline{\text{FLASH\_OE}}$ signal is also used. The number of extended address pins to use may vary, depending on the size and configuration of the ROM and AFLASH devices.

Reading from Expansion Memory

An Expansion Memory read bus cycle starts at T1, when the data bus is in TRI-STATE, and the address is driven on the address bus. $\overline{\text{EMCS}}$ is asserted (0) on a T2W1 cycle. This cycle is followed by three T2W cycles and one T2 cycle. Data is sampled by the NSAM265SF at the end of the T2 cycle.

The transaction is terminated at T3, when $\overline{\text{EMCS}}$ becomes inactive (1). The address remains valid until T3 is complete. A T3H cycle is added after the T3 cycle. The address remains valid until the end of T3H.

$\overline{\text{WR0}}$ is inactive (1) during the read bus cycle.

2.9 FLASH/AFLASH SPECIFICATIONS

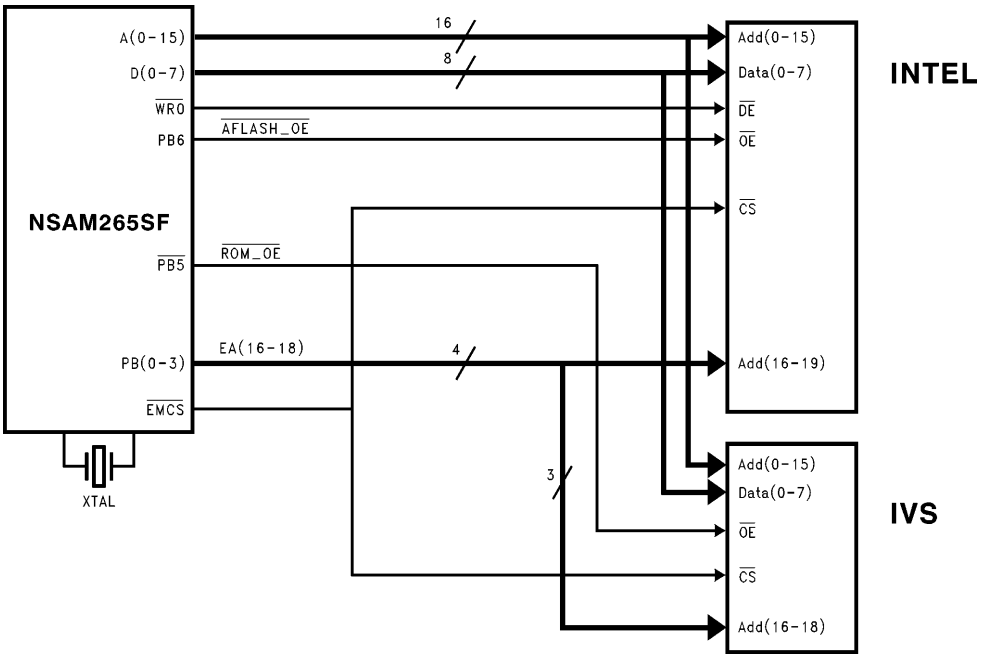
The NSAM265SF supports either the AMD Am29F040 Sector Erase Audio Flash 4-Mbit Memory or the Intel 28F008SA 8-Mbit AFLASH memory as message storage devices. These devices are organized in 64-Kbytes blocks and support up to 100,000 block-erase cycles. For the exact features of these devices please refer to the devices' specifications.

AFLASH devices contain bad blocks. The NSAM265SF supports devices with more good blocks than bad blocks i.e., at least five good blocks on the AMD device, and at least nine good blocks on the Intel device.

2.10 FLASH/AFLASH CONFIGURATIONS

The NSAM265SF supports up to two 28F008SA devices, or up to four 29F040 devices. The actual configuration is selected by the CFG command.

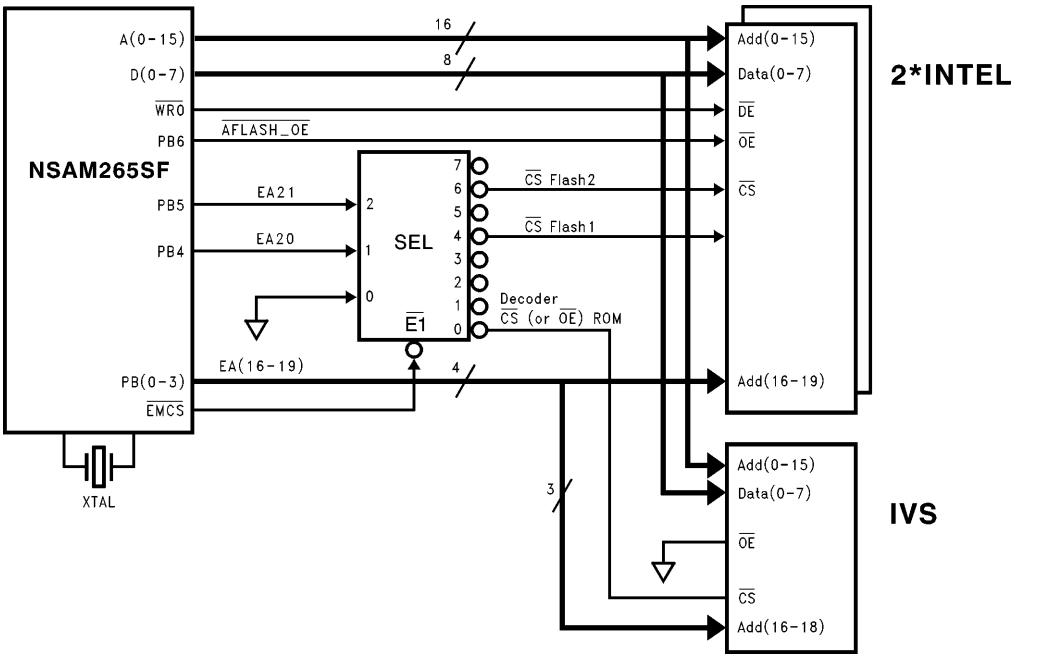
Figures 2-6, 2-7, 2-8 and *2-9* illustrate the minimum and maximum configurations for each device in an environment that contains an IVS ROM.



TL/EE/12378–9

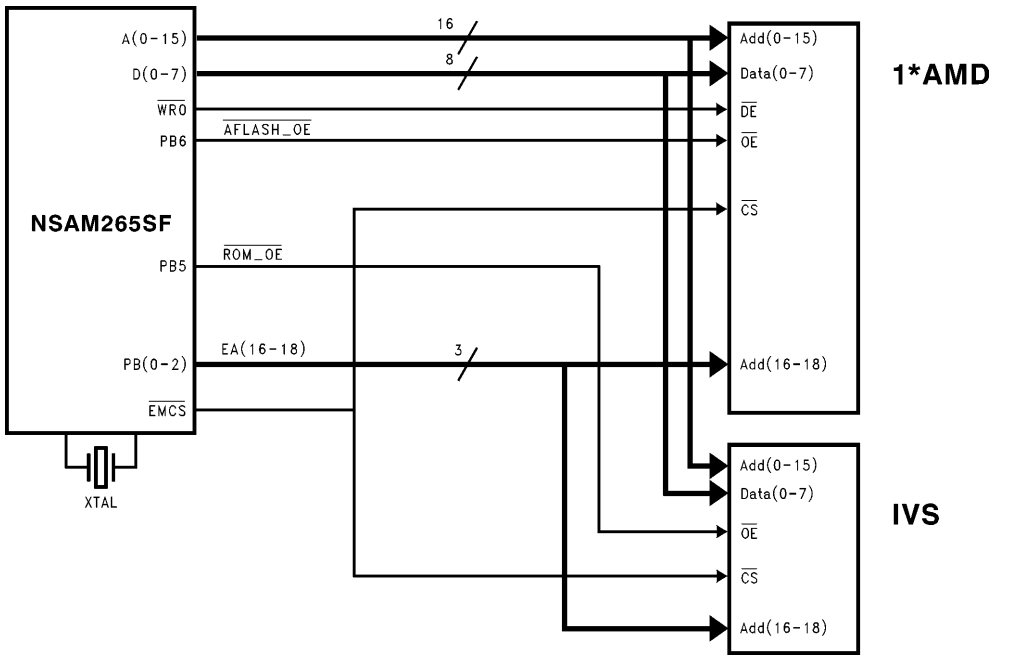
FIGURE 2-6. One Intel Flash Device: No External Glue Logic

2.0 Functional Description (Continued)



TL/EE/12378–10

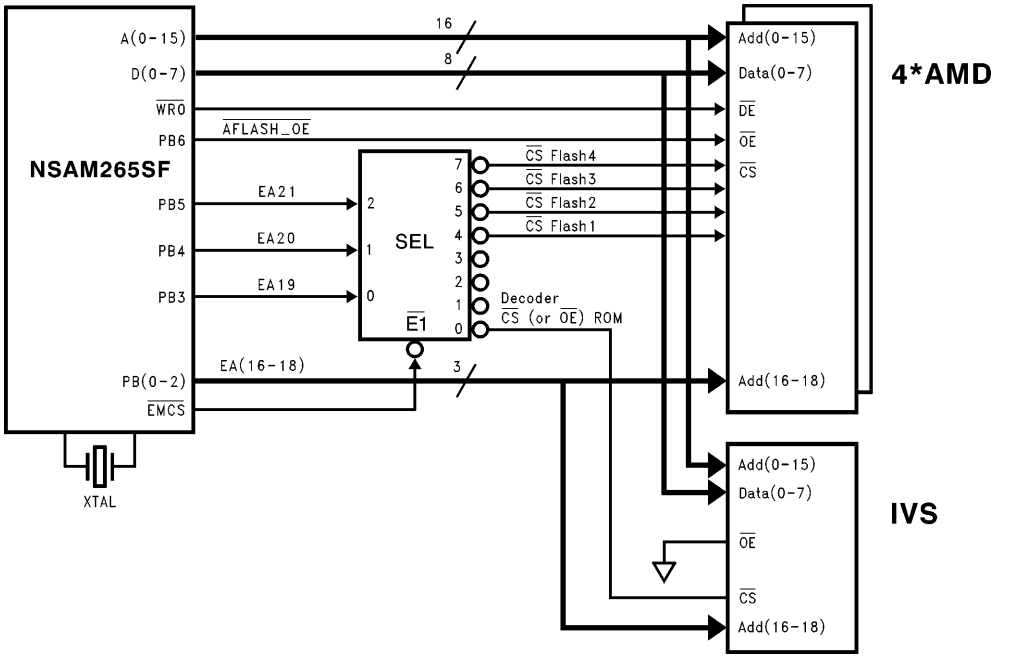
FIGURE 2-7. Two Intel Flash Devices: External Logic—One Decoder



TL/EE/12378–11

FIGURE 2-8. One AMD Flash Device: No External Glue Logic

2.0 Functional Description (Continued)



TL/EE/12378–12

FIGURE 2-9. Four AMD Flash Devices: External Logic—One Decoder

2.11 IVS ACCESS

IVS vocabularies reside on either ROM or FLASH (NSAM265SF only). The IVS tool is used to compile a vocabulary definition into a binary file that should be burnt into the memory device. If more than one vocabulary is used the binary files are concatenated to form a single, ''ready-to-burn'' binary file.

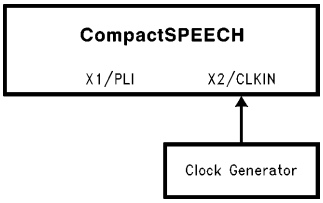
For more details about the IVS tool, and how to program IVS vocabularies on ROM and FLASH. See the *IVS User's Manual*.

2.12 CLOCKING

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

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 2-10*, and should conform with the voltage level requirements for CLKIN stated in ELECTRICAL CHARACTERISTICS.



TL/EE/12378–13

FIGURE 2-10. External Clock Source

2.0 Functional Description (Continued)

Crystal Oscillator

A crystal resonator is connected to the on-chip oscillator circuit via the X1 and X2 signals, as shown in *Figure 2-11*.

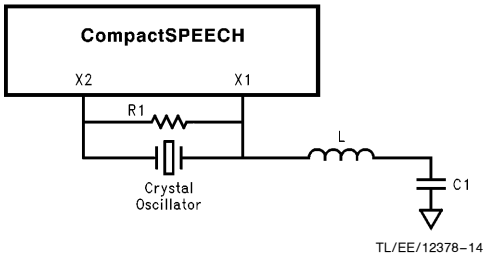


FIGURE 2-11. Connections for an External Crystal Oscillator

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

Crystals with maximum load capacitance of 20 pF may be used, although the oscillation frequency may differ from the crystal's specified value.

Table 2-1 defines the components in the crystal oscillator circuit.

2.13 POWER-DOWN MODE

Power-down mode is useful during a power failure, when the power source for the CompactSPEECH is a backup battery, or in battery powered devices, while the CompactSPEECH is idle. Note that there is no need to battery backup the AFLASH devices in the NSAM265SF.

In power-down mode, the clock frequency of the CompactSPEECH is reduced and some of the processor modules are deactivated. As a result, the CompactSPEECH consumes much less power than in normal-power mode. The CompactSPEECH does not perform its usual functions in power-down mode, but it still preserves stored messages and maintains the time of day.

The NSAM265SF stores messages, and all memory management information, in FLASH memory. Thus, there is no need to maintain the power to the processor to preserve stored messages. If the microcontroller's real-time clock (and *not* the NSAM265SF's real-time clock) is used to maintain the time and day, neither the FLASH nor the NSAM265SF require battery backup during power failure. In this case, when returning to normal mode. the microcontroller should perform the initialization sequence. as described in Section 1.10, and use the SETD command to set the time and day.

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

The PDM (Go To Power-down Mode) command switches the CompactSPEECH to power-down mode. It may only be issued when the CompactSPEECH is in the IDLE state. If it is necessary to switch to power-down mode from any other state, the controller must first issue an S command to switch the CompactSPEECH to the IDLE state, and then issue the PDM command. Sending any command while in power-down mode will return the CompactSPEECH to normal operation mode.

2.14 POWER AND GROUNDING

The CompactSPEECH processor requires a single 5V power supply, applied on the V_{CC} pins.

The grounding connections are made on the GND pins.

For optimal noise immunity, the power and ground pins should be connected to V_{CC} and ground planes, respectively, on the printed circuit board. If V_{CC} and 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.

Use decoupling capacitors to keep the noise level to a minimum. Standard 0.1 μF ceramic capacitors can be used for this purpose. They should attach to V_{CC}, GND pins as close as possible to the CompactSPEECH.

During prototype using wire-wrap or similar methods, the capacitors should be soldered directly to the power pins of the CompactSPEECH socket, or as close as possible, with very short leads.

Design Notes

When constructing a board using high frequency clocks with multiple line switching, special care should be taken to avoid resonances on signal lines. A separate power and ground layer is recommended. Switching times of under 5 ns are possible on some lines. Resonant frequencies should be maintained well above the 200 MHz frequency range on signal paths by keeping traces short and inductance low. Loading capacitance at the end of a transmission line contributes to the resonant frequency and should be minimized if possible. Capacitors should be located as close as possible across each power and ground pair near the CompactSPEECH.

TABLE 2-1. Components of a Crystal Oscillator Circuit

Component	Parameters	Values	Tolerance
Crystal Oscillator	Resonance Frequency Third Overtone Type Maximum Serial Resistance Maximum Shunt Capacitance Maximum Load Capacitance	40.96 MHz Parallel AT-Cut 50Ω 7 pF 12 pF	N/A
Resistor R1		10 MΩ	5%
Capacitor C1		1000 pF	20%
L (Inductance)		3.9 μH	10%

3.0 Command Set

This section describes the CompactSPEECH commands, their parameters and return values. The NSAM265SR and NSAM265SF support the same command set. Some commands, however, behave differently for ARAM and AFLASH.

3.1 COMMANDS SUMMARY

The following table shows the CompactSPEECH commands. These commands are described in greater detail in the next section.

Notes: The command type is shown in the column headed **Type S/A**, where A indicates an Asynchronous command, and S indicates a Synchronous Command.

The current message is not always defined (e.g., after execution of the DMS command). The CompactSPEECH behavior while executing commands that operate on the current message when it is not defined is unpredictable. Use the GTM command to set the current message as required.

Command Mnemonic	Type S/A	Description	Opcode Hex	Command Parameters		Return Value	
				Description	Length Bytes	Description	Length Bytes
AMAP	S	Check and Map ARAM	06	Action__Number	1	Test Result	1
AMSG	A	Append to Current Message	27	None		None	
CFG	S	Configure CompactSPEECH	01	Config__Value	2	None	
CMT	S	Cut Message Tail	26	Length of Time	2	None	
CVOC	S	Check Vocabulary	2B	None		Test Result	1
DM	S	Delete Message	0A	None		None	
DMS	S	Delete Messages	0B	Tag__Ref, Tag__Mask	2 + 2	None	
FR	A	Free Memory	08	None		None	
GCFG	S	Get Configuration Value	02	None		Version, Config.__Value	2
GEW	S	Get Error Word	1B	None		Error Word	2
GI	S	Get Information Item	25	Index	1	Value	2
GL	S	Get Length	19	None		Message Length	2
GMS	S	Get Memory Status Type	12	Type	1	Recording Time Left	2
GMT	S	Get Message Tag	04	None		Message Tag	2
GNM	S	Get Number of Messages	11	Tag__Ref, Tag__Mask	2 + 2	Number of Messages	1
GSW	S	Get Status Word	14	None		Status Word	2
GT	A	Generate Tone Tone	0D	Tone or DTMF	1	None	
GTD	S	Get Time and Day	0E	Time/Day Option	1	Time/Day	2
GTM	S	Get Tagged Message	09	Tag__Ref, Tag__Mask, Dir	2 + 2 + 1	Message Found	1
INIT	S	Initialize System	13	None		None	
INJ	S	Inject IVS Data	29	N, $Byte_1$ . . . $Byte_n$	4 + n	None	
MR	S	Memory Reset	2A	None		None	
P	A	Playback	03	None		None	
PA	S	Pause	IC	None		None	
PDM	S	Go to Power-Down Mode	1A	None		None	
R	A	Record Message	0C	Message Tag	2	None	
RDET	S	Reset Detectors	2C	Detectors Reset Mask	1	None	
RES	S	Resume	1D	None		None	
RRAM	S	Read RAM	18	None		Data	32
S	S	Stop	00	None		None	
SAS	A	Say Argumented Sentence	1E	Sentence__n arg_1	1 + 1	None	
SB	S	Skip Backward	23	Length of Time	2	None	

3.0 Command Set (Continued)

Command Mnemonic	Type S/A	Description	Opcode Hex	Command Parameters		Return Value	
				Description	Length Bytes	Description	Length Bytes
SDET	S	Set Detectors Mask	10	Detectors Mask	1	None	
SE	S	Skip to End of Message	24	None		None	
SETD	S	Set Time and Day	0F	Time/Day	2	None	
SF	S	Skip Forward	22	Length of Time	2	None	
SMT	S	Set Message Tag	05	Message Tag	2	None	
SO	A	Say One Word	07	Word Number	1	None	
SPS	S	Set Playback Speed	16	Speed Value	1	None	
SS	A	Say Sentence	1F	Sentence__n	1	None	
SV	S	Set Vocabulary Type	20	Mode, Id	1 + 1	None	
SW	A	Say Words	21	N, $Word_1$ . . . $Word_n$	1 + n	None	
TUNE	S	Tune Index	15	Index, Value	1 + 2	None	
VC	S	Volume Control	28	Increment/Decrement	1	None	
WRAM	S	Write RAM	17	Message Tag, Data	2 + 32	None	

3.2 COMMANDS DESCRIPTION

The commands are listed in alphabetical order.

Each command indicates by a ✓ the chips for which it is intended. Commands which are not intended for a particular chip are valid for that chip, but have no effect. Chip-specific features of a command which is intended for both chips are indicated by the chip name in the margin.

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.

AMAP **Check and Map ARAM *action__number***

NSAM265SR ✓ NSAM265SF□

This command runs diagnostics on the ARAMs, and returns a 1-byte result. The following actions are performed according to the 1-byte *action__number* parameter:

Action 0

Checks the ARAM configuration. The return value is encoded as follows:

Bits 0–1 Number of ARAM devices (1 or 2)

Bits 2–3 Number of ARAM columns in multiples of 1k units (1 or 2)

Bits 4–6 Number of ARAM rows in multiples of 1k units (1, 2 or 4)

Action 1

This action is intended to be carried out on the production line. It performs the following tests:

— Address line test

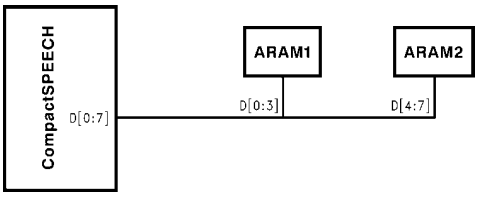
— Scan one row from start to end, and from end to start

— Write to line X and check that line Y is not written. ($\overline{RAS}$/$\overline{CAS}$ not connected test)

— Walking 1 and walking 0 on the data lines

Action 1 then performs Action 3 (see below) on the first 60 rows only. This allows you to record a short memo, of up to 5-seconds duration, to check the system and codec. Execution time is less than 1 second.

This action returns 0 if the ARAMs pass the connectivity test, or the ID of the first ARAM to fail. The return value is 1-byte long. IDs are allocated according to the following scheme:



TL/EE/12378–15

Action 2

Performs a one-pass test, clears the entire ARAM to 0, and marks all the bad rows in the internal row table. If more than 0.5% nibbles in a row are bad, the row is mapped as bad. If more than 200 of the rows are bad, the EV__ERROR bit in the status word and the ERR__ARAM bit in the error word are set to 1.

This operation does not detect ARAM cross-talk problems.

Action 2 of the AMAP command should be invoked immediately after the INIT command, and is valid only in the IDLE state. If this command is not issued, the ARAM is considered as a DRAM (no errors).

The test takes up to 20 seconds.

Action 3

Performs a two-pass test to map, and return the number of bad rows. If more than 0.5% nibbles in a row are bad. the row is mapped as bad. If more than 5% of the rows are bad, the EV__ERROR bit in the status word and the ERR__ARAM bit in the error word are set to 1. This diagnostic also detects ARAM cross-talk problems.

3.0 Command Set (Continued)

The test takes up to 40 seconds.

If *action__number* is not in the 0.3 range, ERR__PARAM is set in the error word.

NSAM265SF

The return value is 0.

AMSG Append to Current Message

NSAM265SR□ NSAM265SF ✓

Skips to the end of the current message and starts recording. The CompactSPEECH state changes to RECORD. The AMSG 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.

EV__MEMFULL is also set in the NSAM265SF if there is only one AFLASH block available for recording, and the MESSAGE__SAFE bit in the current message tag is set.

CFG Configure CompactSPEECH *config__value*

NSAM265SR ✓ NSAM265SF ✓

This command is used to configure the CompactSPEECH in various hardware environments. The command should be used to change the default CompactSPEECH configuration.

The *config__value* parameter is encoded as follows:

Bit 0 Codec configuration.

0: short frame format (default)

1: long frame format. (Guaranteed by design, but not tested.)

Bit 1 Reserved.

Bit 2 Echo cancellation control.

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 AGC during playback. A system with AGC that can not be controlled (i.e., enabled/disabled) by the microcontroller must not turn on this bit.

Bit 3 Memory utilization mode (NSAM265SF only).

0: Normal (default)

1: Intensive

For more details about memory operating modes see Section 1.7.3. The NSAM265SR ignores this bit.

Bits 4–5 FLASH device protocol (NSAM265SF only).

00: AMD Am29040 and compatibles (default)

01: INTEL 28F008SA and compatibles

10: Reserved

11: Reserved

Bits 6–7 FLASH device size and organization (NSAM265SF only).

00: 4-Mbit byte wide (default)

01: 8-Mbit byte wide

10: Reserved

11: Reserved

Bits 8–10 Number of installed FLASH devices (NSAM265SF only).

Valid range 1 . . 4 AMD

1 . . 2 INTEL

Default is 1.

Bits 11–15 Reserved—Must be set to 0.

The following pseudo C code demonstrates the initialization sequence to be executed by a microcontroller that supports the NSAM265SF with two Intel 8-Mbit devices, IVS and a long format codec in a system with AGC that can't be disabled. Memory utilization mode is normal.

```
#define USING_AFLASH   0x0010

#define CODEC_LONG    0x0001
#define INTEL          0x0010
#define SIZE_8         0x0040
#define NUM_DEV       (2 << 8)

void init_CompactSPEECH()
{
     unsigned int rcfg_val, cfg_val;

     rcfg_val = (GCFG() >> 8);/* get current configuration */
     if (!(rcfg_val & USING_AFLASH))
         panic("This CompactSPEECH device does not support AFLASH");
     else {
         cfg_val = CODEC_LONG|INTEL|SIZE_8|NUM_DEV;
         CFG(cfg_val); /* 0x0251 */
         INIT();
     }
}
```

TL/EE/12378–16

3.0 Command Set (Continued)

CMT **Cut Message Tail *time__length***

NSAM265SR ✔ NSAM265SF ✔

Cut *time__length* units, each of 10 ms duration, off the end of the current message. The maximum value of *time__length* is 6550. Cut-time accuracy is 0.14 seconds.

Note: If *time__length* is longer than or equal to the total duration of the message the EV__NORMAL__END event is set in the status word and the message becomes empty (i.e., message length is 0), but is not deleted. Use the DM (Delete Message) or DMS (Delete Messages) command to delete the message.

CVOC **Check Vocabulary**

NSAM265SR ✔ NSAM265SF ✔

Checks the correctness of the current selected vocabulary.

The purpose of this command is to check the correctness of IVS vocabularies which were pre-programmed to a FLASH device.

If the vocabulary data is OK or if the current vocabulary is the internal vocabulary, the return value is 1. Otherwise the return value is 0.

DM **Delete Message**

NSAM265SR ✔ NSAM265SF ✔

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 will be selected as the current message.

NSAM265SR

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

NSAM265SF

The memory space released by the deleted message is potentially available (see the GMS command) but, in practice, is available only after the FR command has been executed.

DMS **Delete Messages *tag__ref tag__mask***

NSAM265SR ✔ NSAM265SF ✔

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 to select a different message to be the current message.

NSAM265SR

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

NSAM265SF

The memory space released by the deleted message is potentially available (see the GMS command) but, in practice, is available only after the FR command has been executed.

FR **Free Memory**

NSAM265SR□ NSAM265SF ✔

Frees memory space for recording. When a message is deleted with a DM or DMS command, the memory space which was occupied by the deleted message is marked DELETED but is not available for recording new messages. The process performed by the FR command makes this memory space available for recording.

When execution of the FR command starts, the state of the CompactSPEECH changes to MEMORY__FREE. When the process is completed the CompactSPEECH sets the EV__NORMAL__END event, and activates the $\overline{\text{MWRQST}}$ signal.

The process can be stopped with the S command. The typical execution time of an S command following an FR command is 1 second. Since the process involves erasure of AFLASH blocks, the maximum execution time of an S command following an FR command is according to the AFLASH device specification (i.e., 3 seconds + single block erasure time).

Possible schemes for using FR in a DAM environment:

- When the CompactSPEECH is idle for a few seconds after a sequence of one or more DM or DMS commands. A typical DAM session involves listening to ICMs and deleting some of them. The microcontroller assumes that such a session has been completed if the CompactSPEECH is idle for a preset time, and should then issue the FR command.
- During initialization (after the INIT command has been executed). We recommend sending a GMS command to see if there is sufficient potential additional recording time to justify sending the FR command.

We recommend suspending FR execution (with the S command) if a ring is detected, to minimize the time taken by the CompactSPEECH to prepare for ICM recording.

Note: After FR execution has been completed, the current message is undefined. Use the GTM command to select the current message.

GCFG **Get Configuration Value**

NSAM265SR ✔ NSAM265SF ✔

This command returns a sequence of two bytes with the following information:

Bits 0–7 Magic number which specifies the CompactSPEECH firmware version

Bits 8–9 **Memory type**

00: ARAM

01: AFLASH

10, 11: Reserved

The command should be used together with the CFG and INIT commands during CompactSPEECH initialization. See the CFG command for more details and an example of a typical initialization sequence.

GEW **Get Error Word**

NSAM265SR ✔ NSAM265SF ✔

Returns the 2-byte error word.

The 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 low).

3.0 Command Set (Continued)

15 8	7	6	5	4	3	2	1	0
Reserved	ERR__ INVALID	ERR__ TIMEOUT	ERR__ COMM	ERR__ ARAM	ERR__ PARAM	ERR__ COMMAND	ERR__ OPCODE	Reserved

The GEW command reads the error word. The error word is cleared during reset and before execution of all commands, except GSW and GEW.

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.

The bits of the error word are used as follows:

ERR__OPCODE

Illegal opcode. The command opcode is not recognized by the CompactSPEECH.

ERR__COMMAND

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

ERR__PARAM

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

ERR__ARAM

ARAM test failed. This test is performed only during execution of the AMAP command on the NSAM265SR.

ERR__COMM

MICROWIRE communication error.

ERR__TIMEOUT

Time-out error. More than 2 ms elapsed between the arrival of two consecutive bytes (for commands that have parameters).

ERR__INVALID

Command can't be performed in current context.

GI **Get Information *item***

NSAM265SR ✓ NSAM265SF ✓

Returns the 16-bit value specified by *item* from one of the internal registers of the CompactSPEECH.

item may be one of the following:

0: The duration of the 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. The duration includes the time required to verify that the DTMF tone has ended.

1: The duration of the last detected busy tone in 10 ms units. This value is only valid if EV__BUSY of the status register is set to 1. Otherwise, it is undefined.

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 1-1*. 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*.

GL **Get Length**

NSAM265SR ✓ NSAM265SF ✓

Returns the length of the current message in multiples of 32 bytes.

This information enables the controller to read the entire message from memory using the RRAM command. The returned value is a 16-bit unsigned integer.

NSAM265SR

The returned value includes the entire last ARAM row of the message, even if the message occupies only a portion of the last row. Since an ARAM row includes 512, 1024 or 2048 bytes (depending on the memory configuration), the returned value may be up to 511, 1023 or 2047 bytes bigger than the actual message length.

NSAM265SF

The returned value includes the entire last AFLASH segment of the message, even if the message occupies only a portion of the last segment. Since an AFLASH sector includes 2048 bytes, the returned length may be up to 2047 bytes bigger than the actual message length.

The minimum length of a message is one segment, i.e., an empty message occupies 2 kbytes (the message length is: 2048/32 = 64).

GMS **Get Memory Status *type***

NSAM265SR ✓ NSAM265SF ✓

Returns the estimated total remaining recording time in seconds as a 16-bit unsigned integer. This estimate assumes no silence compression: a real recording may be longer, according to the amount of silence detected and compressed.

The return value is dependent on the value of the *type* parameter as follows:

0: The actual remaining recording time is returned.

1: The potential additional recording time is returned.

Deleted messages hold memory space that can not be used for recording until the FR command is executed. The total amount of recording time used by such messages is the potential additional recording time. (In the NSAM265SR this time is always zero.)

2: The sum of the actual and potential recording times is returned.

The return value is unpredictable for any other value of *type*.

GMT **Get Message Tag**

NSAM265SR ✓ NSAM265SF ✓

Returns the 16-bit tag associated with the current message.

NSAM265SF

Bits 7–15 are meaningful only in the NSAM265SF.

3.0 Command Set (Continued)

15	14	13	12 11	10	9	8	7	6	5	4	3 0
EV__ DTMF	EV__ RESET	EV__ VOX	Reserved	EV__ MEMLOW	EV__ DIALTONE	EV__ BUSY	EV__ ERROR	EV__ MEMFULL	EV__ NORMAL__ END	EV__ DTMF__ END	EV__ DTMF__ DIGIT

GNM Get Number of Messages *tag__ref tag__mask*

NSAM265SR ✔ NSAM265SF ✔

Returns the number of 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.

The *tag__ref* and *tag__mask* parameters are each two bytes; the return value is also 2-byte long.

For example, if *tag__ref* = 42_{16}, and *tag__mask* = $3F_{16}$, the number of existing old messages whose user-defined tag is 2 is returned. See Section 1.5 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.

GSW Get Status Word

NSAM265SR ✔ NSAM265SF ✔

Returns the 2-byte status word.

The Status Word

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

The status word is cleared during reset and by the GSW command.

The bits in the status word are used as follows:

EV__DTMF__DIGIT

DTMF digit. A value indicating a detected DTMF digit. (See 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, end of garbage collection, etc.

EV__MEMFULL

1 = Memory is full.

EV__ERROR

1 = Error detected in the last command. You can use the GEW command to return the error code.

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

NSAM265SF only. 1 = Memory is almost full (i.e., 20 seconds recording time remain, assuming no silence, on the one-before-last FLASH block) when working in memory intensive mode.

EV__VOX

1 = a period of silence (no energy) was detected on the telephone line during recording.

EV__RESET

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

EV__DTMF

1 = Started detection of a DTMF tone.

GT Generate Tone *tone*

NSAM265SR ✔ NSAM265SF ✔

Generates the tone specified by the 1-byte *tone* parameter, until an S command is received.

Specify the tone by setting the bits of *tone* 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 3–20.

The value in bits 1–5 is multiplied by 100 to generate the required frequency (300 Hz through 2000 Hz)

Bits 6, 7 0

CompactSPEECH does not check the validity of the tone specification. Invalid specifications yields unpredictable results.

3.0 Command Set (Continued)

GTD **Get Time and Day *time__day__option***

NSAM265SR ✓ NSAM265SF ✓

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 stamp.

Any other *time__day__option* returns the time stamp of the current message.

Time of day 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 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.

GTM **Get Tagged Message *tag__ref tag__mask dir***

NSAM265SR ✓ NSAM265SF ✓

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.

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

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

Note: 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.

If *dir* is 0 and 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, trying to go beyond the first (last) message during skip-previous (skip-next) operation results with the first (last) message selected as the current message and 0 (FALSE) is returned.

INIT **Initialize System**

NSAM265SR ✓ NSAM265SF ✓

This command should be executed after the CompactSPEECH is configured (see CFG and GCFG commands).

Performs a soft reset of the CompactSPEECH as follows:

- Initializes the message directory information.
  NSAM265SR: all messages are deleted.
  NSAM265SF: messages are not deleted. To delete the messages, use the DM and DMS commands.
- Sets the detectors mask to 0.
- Sets the playback speed to normal (0).
- In the NSAM265SR, marks the entire ARAM as good (no defective rows).
- In the NSAM265SF, completes a garbage collection operation to ensure consistency of the FLASH memory manager (required only if the command is executed after power failure which happened during garbage collection). This may take up to 4 seconds + single FLASH block erasure time.
- Switches to the IDLE state.
- Activates (sets to low) the $\overline{\text{MWDRY}}$ signal.

The current message is undefined after INIT execution.

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

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

NSAM265SR NSAM265SF ✓

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

The purpose of the command is to enable to program FLASH devices on the production line with IVS vocabulary data. The command is optimized for speed and therefore all CompactSPEECH detectors are suspended during the command execution. The CVOC command can be used 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 cannot be used for recording even if only one byte of the block contains IVS data (e.g., if the vocabulary size is 64k + 100 bytes. TWO blocks of the FLASH will not be available for message recording).

MR **Memory Reset**

NSAM265SR NSAM265SF ✓

Erases all good AFLASH blocks and initializes CompactSPEECH (i.e., does exactly what the INIT command does). Bad blocks and blocks which are used for IVS vocabularies are not erased.

The execution time of the command is dependent on the number of good AFLASH blocks and the block erasure time of the AFLASH device.

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

P **Playback**

NSAM265SR ✓ NSAM265SF ✓

Begins playback of the current message. The CompactSPEECH state changes to PLAY. When playback is complete. the CompactSPEECH sets the EV__NORMAL__END bit in the status word, and activates (sets to low) the $\overline{\text{MWRQST}}$ signal. Playback can be paused with the PA command, and can be resumed later with the RES command.

PA **Pause**

NSAM265SR ✓ NSAM265SF ✓

Suspends the execution of the current Say command, R, P, or GT command. The PA command does not change the state of the CompactSPEECH; execution can be resumed with the RES command. The S command switches the CompactSPEECH to the IDLE state.

Note: DTMF and tone detectors remain active during Pause.

PDM **Go To Power-Down Mode**

NSAM265SR ✓ NSAM265SF ✓

Switches the CompactSPEECH to power-down mode (see Section 2.13 for details).

3.0 Command Set (Continued)

R **Record *tag***

NSAM265SR ✓ NSAM265SF ✓

Records a new message with message tag *tag*. The CompactSPEECH 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.

NSAM265SF

EV__MEMFULL is also set in the NSAM265SF if there is only one AFLASH block available for recording. and the MESSAGE__SAFE bit in the *tag* parameter is set.

NSAM265SR

If an attempt to record more than the maximum number of messages is made, an ERR__INVALID error is reported.

Note: A time/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.

RDET **Reset Detectors *detectors reset mask***

NSAM265SR ✓ NSAM265SF ✓

Resets the CompactSPEECH 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 1–4 Reserved. Must be set to 0.

Bit 5 Reset the no energy (VOX) detector.

Bit 6 Reset the DTMF detector.

Bit 7 Reserved. Must be set to 0.

RES **Resume**

NSAM265SR ✓ NSAM265SF ✓

Resumes the activity that was suspended by the PA command.

RRAM **Read RAM**

NSAM265SR ✓ NSAM265SF ✓

Returns 32 bytes from the current message. The first RRAM command issued returns the first 32 bytes of the current message. Subsequent RRAM commands return the next following 32 bytes from the message until the end of the message. The command sequence can be stopped by the S command.

Notes: When the end of the message is detected during RRAM execution, the CompactSPEECH sets the EV__NORMAL__END bit in the status word. If the current message was created with the WRAM command, the 32-byte return value should be ignored.

When using WRAM and RRAM to write and read messages of arbitrary length, the microcontroller is responsible to mark the actual end of the message (e.g., with a delimiter string)

The next RRAM command after the end of the message is reached, starts again from the beginning of the current message.

S **Stop**

NSAM265SR ✓ NSAM265SF ✓

Stops execution of the current command, and switches the CompactSPEECH to the IDLE state. S may be used to stop execution of all the Say commands, and the FR, P, R, GT, RRAM and WRAM commands.

SAS **Say Argumented Sentence *sentence__n arg***

NSAM265SR ✓ NSAM265SF ✓

SAS announces sentence number *sentence__n* of the currently selected vocabulary. and passes *arg* to it. *sentence__n* and *arg* are each 1-byte long.

When playing is complete. the CompactSPEECH sets the EV__NORMAL__END bit in the status word, and activates the $\overline{\text{MWRQST}}$ signal.

If you are using an internal vocabulary, you can synthesize the two built-in sentences. These sentences are *Time and Day*, and *You Have*. These sentences are designated 0 and 1, respectively, in the internal vocabulary sentence table.

Sentence 0

If the internal vocabulary is selected, the built-in *Time and Day* sentence is synthesized. If an external vocabulary is selected, it is assumed that the *Time and Day* sentence is defined as sentence 0 in the sentence table. For example, use SAS 0,0 to synthesize the current time and day.

Sentence 1

If the internal vocabulary is selected, the built-in *You Have* sentence is synthesized. If an external vocabulary is selected, it is assumed that the *You Have* sentence is defined as sentence 1 in the sentence table. For example, use SAS 1,5 to synthesize the sentence *You Have 5 Messages*.

If sentence n is not defined in the current vocabulary, ERR__PARAM is set in the error word.

For further information on sentences 0 and 1, and their options, see Table 1-3.

SB **Skip Backward *time__length***

NSAM265SR ✓ NSAM265SF ✓

Retreats in the current message *time__length* units, each of 0.2 seconds duration, and cause message playback to pause. *time__length* is a 2-byte parameter that may have any value up to 320, i.e., 64 seconds. The skip accuracy is 0.14 seconds. This command is meaningful only in the PLAY state. The RES command must be issued to continue playback.

If the beginning of the message is detected during execution of the SB command, execution of this command is terminated, the EV__NORMAL__END bit in the status register is set, the $\overline{\text{MWRQST}}$ signal is activated, and the CompactSPEECH switches to the IDLE state. If time__length is greater than 320, ERR__PARAM is set in the error word.

SDET **Set Detectors Mask *detectors__mask***

NSAM265SR ✓ NSAM265SF ✓

Controls the reporting of detection for tones and VOX 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.

3.0 Command Set (Continued)

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.

Bits 2–4 Reserved. Must be set to 0.

Bit 5 Report detection of no energy (VOX) on the line.

(The VOX attributes are specified with the tunable parameters VOX__TIME__COUNT and VOX__ENERGY__LEVEL.)

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).

SE **Skip to End of Message**

NSAM265SR ✓ NSAM265SF ✓

This command is valid only in 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.

SETD **Set Time and Day *time__and__day***

NSAM265SR ✓ NSAM265SF ✓

Sets the system time of day as specified by bits 0–13 in the 2-byte *time__and__day* parameter. The *time__and__day* parameter is encoded as follows:

Bits 0–2 Day of the week (1 through 7)

When the internal vocabulary is used, the first day of the week is Monday.

Bits 3–7 Hour of the day (0 through 23)

Bits 8–13 Minute of the hour (0 through 59)

Bits 14–15 These bits must be set to 1.

If time__and__day is not valid, ERR__PARAM is set in the error word.

SF **Skip Forward *time__length***

NSAM265SR ✓ NSAM265SF ✓

Advances in the current message *time__length* units, each of 0.2 seconds duration, and causes message playback to pause. *time__length* is a 2-byte parameter that may have any value up to 320, i.e., 64 seconds. The skip accuracy is 0.14 seconds. This command is meaningful only in the PLAY state. The RES command must be issued 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 register is set, the $\overline{\text{MWRQST}}$ signal is activated, and CompactSPEECH switches to the IDLE state.

SMT **Set Message Tag *message__tag***

NSAM265SR ✓ NSAM265SF ✓

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 Section 1.5.

To change the tag of a message, we recommend that you read the message tag, modify it, and write it back.

If the current message is undefined results are unpredictable.

NSAM265SF

Bits in the message tag may be cleared, but not set.

NSAM265SR

Bits 7–15 of the tag are ignored.

SO **Say One Word *word__number***

NSAM265SR ✓ NSAM265SF ✓

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.

When playback of the selected word has been completed, the CompactSPEECH sets the EV__NORMAL__END bit in the status word, and activates the $\overline{\text{MWRQST}}$ signal.

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.

SPS **Set Playback Speed *speed***

NSAM265SR ✓ NSAM265SF ✓

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

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

Note that a negative *speed* value represents an increase in speed, a positive value represents a decrease in speed.

The change in speed is approximate and dependent on the recorded data.

If *speed* is not in the -6 to $+6$ range, ERR__PARAM is set in the error word.

SS **Say Sentence *sentence__n***

NSAM265SR ✓ NSAM265SF ✓

Say sentence number *sentence__n* of the currently selected vocabulary. *sentence__n* is 1-byte long.

When playing has been completed, the CompactSPEECH sets the EV__NORML__END bit in the status word, and activates the $\overline{\text{MWRQST}}$ signal.

If *sentence__n* is not defined in the current vocabulary, ERR__PARAM is set in the error word.

3.0 Command Set (Continued)

SV **Set Vocabulary Type *type id***

NSAM265SR ✔ NSAM265SF ✔

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

0: Internal vocabulary (default)

1: External vocabulary in ROM

2: External vocabulary in AFLASH (NSAM265SF only)

all others: Reserved

If *type* is 0, *id* is ignored.

Each external vocabulary table has a unique id which is part of the vocabulary internal header (See the IVS User's Manual for more details). If *type* is 1 or 2, the CompactSPEECH 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.

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

NSAM265SR ✔ NSAM265SF ✔

This command accepts the number of words to synthesize (*n*, 1 byte), followed by the indexes of the words (1 byte each) in the current vocabulary. *n* can be 1 to 8. It plays the words. On completion, the EV__NORMAL__END bit in the status word is set, and the $\overline{\text{MWRQST}}$ signal goes low.

If one of the words 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.

TUNE **Tune *index parameter__value***

NSAM265SR ✔ NSAM265SF ✔

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 CompactSPEECH 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.

Table 3-1 describes the tunable parameters, their index numbers and their default values.

TABLE 3-1. Tunable Parameters

Index	Parameter Name	Description	Default
0–3	Reserved		—
4	__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	__SIL__THRESHOLD__STEP	Defines the adaptive threshold changes step. If this threshold is too low, the threshold converges too slowly. If this threshold is too high, silence detection is too sensitive to any noise. Legal values: 3 to 48.	12
6	__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, speech is detected if there is a burst of noise. If it is too high words may be partially cut off. Legal values: 1 to 3.	2
7	__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. silence is detected. Legal values: 8 to 31.	15
8	__SIL__ENABLE	Silence compression control. 0 turns silence compression off.	1
9	__ENERGY__FACTOR	Determines the energy level used to synthesize silence. The default value means that the energy level of the synthesized silence will be the same as the energy level of the recorded silence. If you divide (multiply) the default value by two, the synthesized silence will be 6 dB less (more) than the level of the recorded silence. Legal values: 1024 to 16384.	8192

3.0 Command Set (Continued)

TABLE 3-1. Tunable Parameters (Continued)

Index	Parameter Name	Description	Default
10	VOX__ENERGY__THRESHOLD	This constant determines the minimum energy level at which voice is detected. Below this level, it is interpreted as silence. If you divide (multiply) the value by 2 you get 3 dB decrease (increase) in the threshold. Legal values: 0 to 65535.	12
11	Reserved		—
12	VOX__TIME__COUNT	This constant, in units of 10 ms, determines the period of silence before the CompactSPEECH reports silence. Legal values: 0 to 65535.	700
13–14	Reserved		—
15	VOICE__SYNTHESIS__LEVEL	Controls the energy level at which internal vocabulary words are output. 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 VOICE__SYNTHESIS__LEVEL and the VOL__LEVEL variable, controlled by the VC (Volume Control) command. The synthesized speech is distorted when the level is set too high. The valid range is: $0 \leq$ VOICE__SYNTHESIS__LEVEL + VOL__LEVEL ≤ 12.	6
16	TONE__GENERATION__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__GENERATION__LEVEL and the VOL__LEVEL variable, controlled by the VC command. The tones are distorted when the level is set too high. The valid range is: $0 \leq$ TONE__GENERATION__LEVEL + VOL__LEVEL ≤ 12.	6
17	Reserved.		—
18	TONE__TIME__COUNT	Controls the duration of a tone before it is reported as a dial tone, in 10 ms units. Legal values: 0 to 65535.	700
19	TONE__ON__ENERGY__THRESHOLD	Minimum energy level at which busy and dial tones are detected as ON (after 700 Hz filtering). If you divide (multiply) the value by 2 you get 3 dB decrease (increase) in the threshold. Legal values: 0 to 65535.	160
20	TONE__OFF__ENERGY__THRESHOLD	Maximum energy level at which busy and dial tones are detected as OFF (after 700 Hz filtering). If you divide (multiply) the value by 2 you get 3 dB decrease (increase) in the threshold. Legal values: 0 to 65535.	110
21	VCD__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. The speech is distorted when the level is set too high. The valid range is: $0 \leq$ VCD__LEVEL + VOL__LEVEL ≤ 12.	6
22	VOX__TOLERANCE__TIME	Controls the maximum energy period, in 10 ms units, that does NOT reset the vox detector. Legal values: 0 to 255.	3
23	MIN__BUSY__DETECT__TIME	Minimum time period for busy detection, in 10 ms units. Legal values: 0 to 65535.	600

3.0 Command Set (Continued)

TABLE 3-1. Tunable Parameters (Continued)

Index	Parameter Name	Description	Default
24	ECHO__DELAY	The near echo delay in samples. For example, the default value is computed as follows: The near echo delay is assumed 500 μs. Since the sampling rate is 8000 Hz (i.e., 125 μs per sample), the value of ECHO__DELAY is 4. Legal values: 0 to 255.	4
25	Reserved		—
26	DTMF__REV__TWIST	Controls the reverse twist level at which CompactSPEECH detects DTMF tones. While the normal twist is set at 8 dB, the reverse twist can be either 8 dB (default) or 4 dB (If this parameter is set to 1).	0
27	DTMF__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 on the output of a TP3054 codec (before the DAA) connected to the CompactSPEECH 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 tone during meaurements was 6 (TONE__GENERATION__LEVEL + VOL__LEVEL = 6). The default level means that the high tone is at −14.3 dBv and the low tone at −12.4 dBv which gives DTMF twist level of 1.9 dB.	66

VC **Volume Control *vol__level***

NSAM265SR ✔ NSAM265SF ✔

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 3-1.

For example, if the tunable variable VCD__LEVEL is 6, and *vol__level* is −2, then the output level equals VCD__LEVEL + *vol__level* = 4.

WRAM **Write RAM tag, *data***

NSAM265SR ✔ NSAM265SF ✔

This command creates a new message with a message tag *tag*. The following 32 bytes of *data* are stored as the new message data in the message memory.

The WRAM command switches the CompactSPEECH to the MEMORY__WRITE state. As long as it remains in this state, each subsequent WRAM command appends new message data to the end of the previous data. The CompactSPEECH remains in the MEMORY__WRITE state until an S command is issued. Note that while the CompactSPEECH is in MEMORY__WRITE state the *tag* parameter is ignored.

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

EV__MEMFULL will be also set in the NSAM265SF if there is only one AFLASH block available for recording and the MESSAGE__SAFE bit in the *tag* parameter is set.

NSAM265SR

If an attempt to record more than the maximum number of messages is made, an ERR__INVALID error is reported.

Appendix A
Device Specifications

This appendix describes the pins and signals of the NSAM265SR and NSAM265SF CompactSPEECH processors, specifies its maximum ratings and its electrical characteristics, and describes its timing.

A.1 PIN ASSIGNMENT

The following sections list the pins of the NSAM265SR and NSAM265SF CompactSPEECH processors. They indicate which signals use the same pin and under what conditions each signal is enabled for that pin. Slashes separate the names of signals that share the same pin.

A.1.1 Pin-Signal Assignment

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

TABLE A-1. CompactSPEECH Pin-Signal Assignment under Different Conditions

Pin Name	Type	Signal Name	I/O
A(0:15)	TTL	A(0:15)	Output
$\overline{CAS}$ (Note A)	TTL1 (Note B)	CAS	Output
CCLK	TTL	CCLK	Output
CDIN	TTL	CDIN	Input
CDOUT	TTL	CDOUT	Output
CFS0	TTL	CFS0	Output
D(0:1)	TTL	D(0:1)	I/O
D2/RA11	TTL	D2/RA11	I/O
D(3:7)	TTL	D(3:7)	I/O
$\overline{DWE}$	TTL1 (Note B)	$\overline{DWE}$	Output
$\overline{MWCS}$	TTL (Note C)	$\overline{MWCS}$	Input
$\overline{WR0}$/$\overline{TST}$	TTL	$\overline{WR0}$ TST	Output Input
$\overline{MWDRY}$	TTL	$\overline{MWDRY}$	I/O
$\overline{MWRQST}$	TTL	$\overline{MWRQST}$	I/O
MWDOUT	TTL	MWDOUT	Output
PB(0:5) (Note D)	TTL	EA(16:21)	Output
PB6 (Note E)	TTL	$\overline{AFLASH\_\_OE}$	Output
$\overline{EMCS}$/ENV0	TTL1 (Note B)	$\overline{EMCS}$	Output
	CMOS (Note F)	ENV0	Input
MWCLK	TTL	MWCLK	Input
MWDIN	TTL	MWDIN	Input
$\overline{RAS}$ (Note A)	TTL1 (Note B)	$\overline{RAS}$	Output
$\overline{RESET}$	Schmitt (Note C)	$\overline{RESET}$	Input
V_{CC}	Power	V_{CC}	
V_{SS}	Power	V_{SS}	
X1	XTAL	X1	OSC
X2/CLKIN	XTAL TTL	X2 CLKIN	OSC Input

Note A: Used in NSAM265SR only. N.C. in NSAM265SF.

Note B: TTL1 output signals provide CMOS levels in the steady state, for small loads.

Note C: Schmitt trigger input.

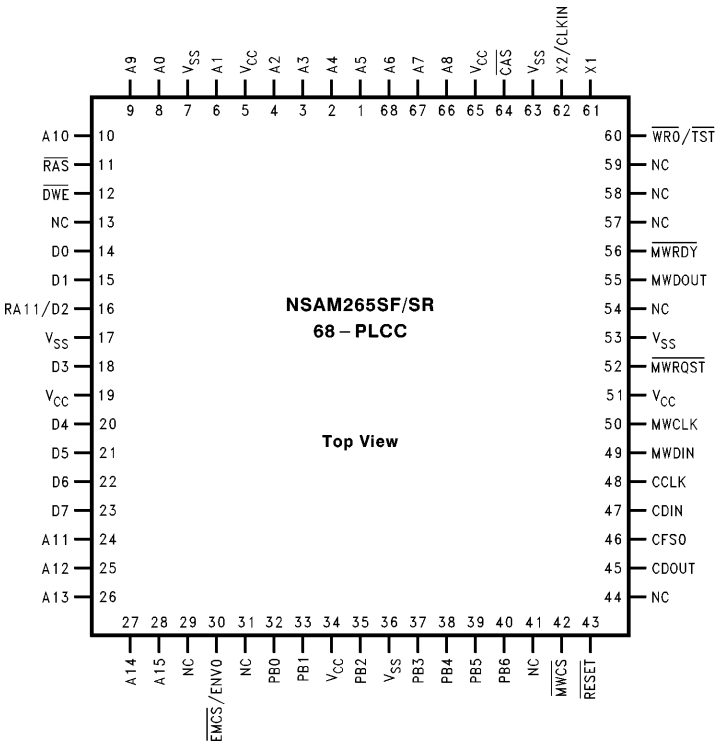
Note D: NSAM265SR and NSAM265SF—virtual address lines for IVS ROM.
NSAM265SF only—virtual address lines for FLASH.

Note E: Used in NSAM265SF only—enables/disable FLASH.

Note F: Input during reset, CMOS level input.

Appendix A Device Specifications (Continued)

A.1.2 PIN ASSIGNMENT IN THE 68-PLCC PACKAGE

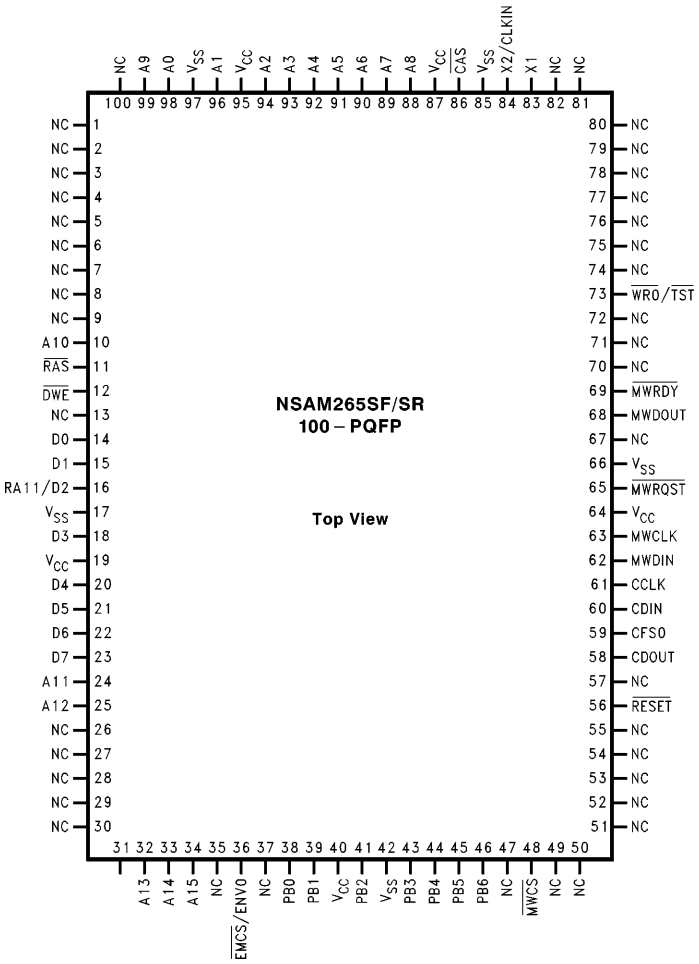


TL/EE/12378–17

FIGURE A-1. 68-PLCC Package Connection Diagram

Appendix A Device Specifications (Continued)

A.1.3 PIN ASSIGNMENT IN THE 100-PQFP PACKAGE



TL/EE/12378–18

FIGURE A-2. 100-PQFP Package Connection Diagram

Appendix A Device Specifications (Continued)

A.2 ABSOLUTE MAXIMUM RATINGS

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Storage Temperature	−65°C to +150°C
Temperature Under Bias	0°C to +70°C
All Input or Output Voltages, with Respect to GND	−0.5V to +6.5V

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 those conditions specified below.

A.3 ELECTRICAL CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 10\%$, GND = 0V

Symbol	Parameter	Conditions	Min	Typ	Max	Units
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_{XH}	CLKIN Input, High Voltage	External Clock	2.0			V
V_{XL}	CLKIN Input, Low Voltage	External Clock			0.8	V
V_{ENVh}	ENV0 High Level, Input Voltage		3.6			V
V_{Hh}	CMOS Input with Hysteresis, Logical 1 Input Voltage		3.6			V
V_{HI}	CMOS Input with Hysteresis, Logical 0 Input Voltage				1.1	V
V_{hys}	Hysteresis Loop Width (Note A)		0.5			V
V_{OH}	Logical 1 TT, Output Voltage	$I_{OH} = -0.4$ mA	2.4			V
V_{OHWC}	$\overline{RAS}$, $\overline{CAS}$, $\overline{DWE}$ and $\overline{EMCS}$ Logical 1, Output Voltage	$I_{OH} = -0.4$ mA	2.4			V
		$I_{OH} = -50\ \mu$A (Note B)	$V_{CC} - 0.2$			V
V_{OL}	Logical 0, TTL Output Voltage	$I_{OL} = 4$ mA			0.45	V
		$I_{OL} = 50\ \mu$A (Note B)			0.2	V
V_{OLWC}	$\overline{RAS}$, $\overline{CAS}$, $\overline{DWE}$ and $\overline{EMCS}$ Logical 0, Output Voltage	$I_{OL} = 4.0$ mA			0.45	V
		$I_{OL} = 50\ \mu$A (Note B)			0.2	V
I_L	Input Load Current (Note C)	$0V \le V_{IN} \le V_{CC}$	−5.0		5.0	
I_O (Off)	Output Leakage Current (I/O Pins in Input Mode) (Note C)	$0V \le V_{OUT} \le V_{CC}$	−5.0		5.0	μA
I_{CC1}	Active Supply Current	Normal Operation Mode Running Speech Applications (Note D)		65	80	mA
I_{CC2}	Standby Supply Current	Normal Operation Mode, DSPM Idle (Note D)		40		mA
I_{CC3}	Power-Down Mode Supply Current	Power-Down Mode (Notes D and E)			1.5	mA
C_X	X1 and X2 Capacitance (Note A)			17		pF

Note A: Guaranteed by design.

Note B: Measured in power-down mode. The total current driven or sourced by all the CompactSPEECH's output signals is less than 50 μA.

Note C: Maximum 20 μA for all pins together.

Note D: $I_{OUT} = 0$, $T_A = 25°C$, $V_{CC} = 5V$, operating from a 40.96 MHz crystal and running from internal memory with Expansion Memory disabled.

Note E: All input signals are tied to 1 or 0 (above $V_{CC} - 0.5$ or below $V_{SS} + 0.5V$).

A.4 SWITCHING CHARACTERISTICS

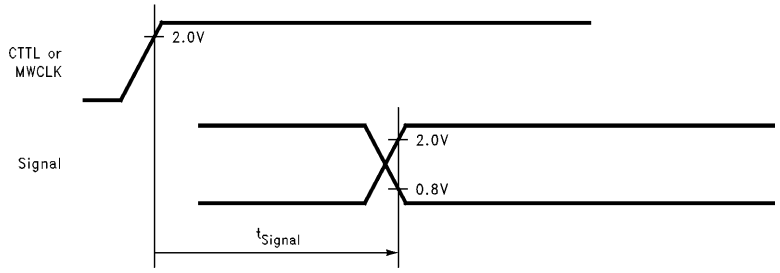
A.4.1 DEFINITIONS

All timing specifications in this section refer to 0.8V or 2.0V on the rising or falling edges of the signals, as illustrated in *Figures A-3* through *A-10*, unless specifically stated otherwise.

Maximum times assume capacitive loading of 50 pF.

CLKIN crystal frequency is 40.96 MHz.

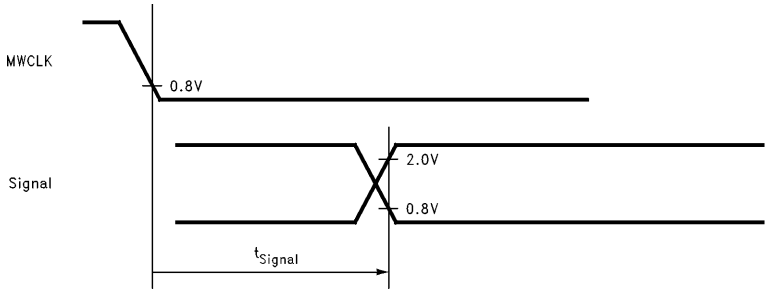
Note: CTTL is an internal signal and is used as a reference to explain the timing of other signals. See *Figure A-21*.



TL/EE/12378–19

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

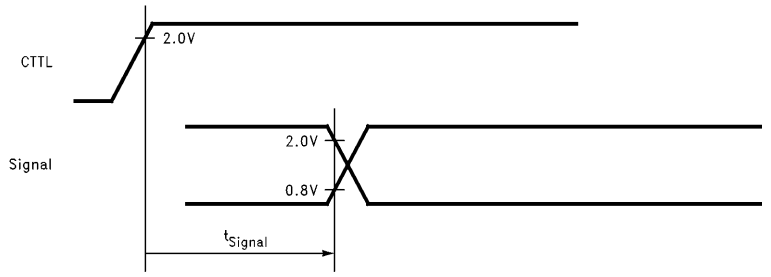
FIGURE A-3. Synchronous Output Signals (Valid, Active and Inactive)



TL/EE/12378–20

Signal valid time, after a falling edge of MWCLK.

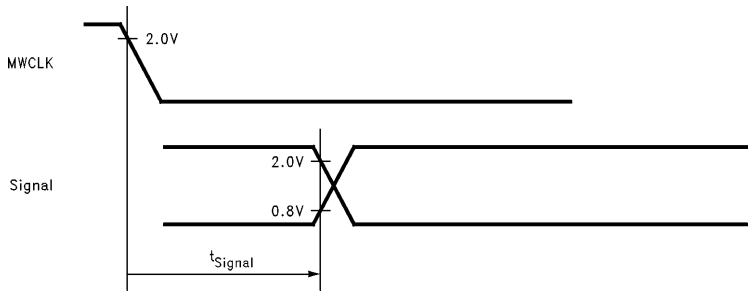
FIGURE A-4. Synchronous Output Signals (Valid)



TL/EE/12378–21

Signal hold time, after a rising edge of CTTL.

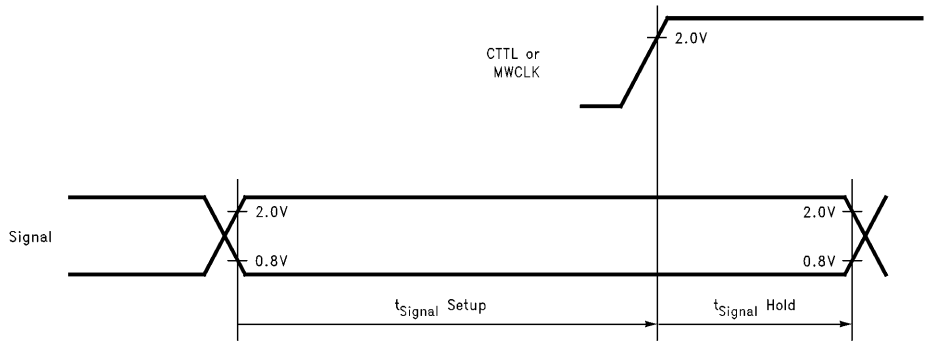
FIGURE A-5. Synchronous Output Signals (Hold)



TL/EE/12378–22

Signal hold time, after a falling edge of MWCLK.

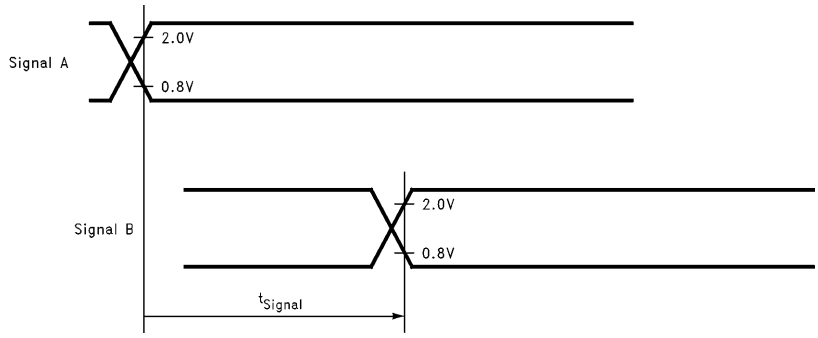
FIGURE A-6. Synchronous Output Signals (Hold)



TL/EE/12378–23

Signal setup time, before a rising edge of CTTL or MWCLK, and signal hold time after a rising edge of CTTL or MWCLK.

FIGURE A-7. Synchronous Input Signals

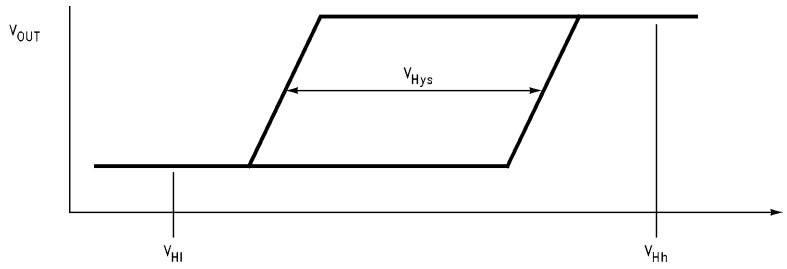


TL/EE/12378–24

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

FIGURE A-8. Asynchronous Signals

$\overline{\text{RESET}}$ has Schmitt trigger input buffers. *Figure A-9* shows the input buffer characteristics.



TL/EE/12378–25

FIGURE A-9. Hysteresis Input Characteristics

A.4.2 SYNCHRONOUS TIMING TABLES

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

TABLE A-2. Timing for Output Signals

Symbol	Figure	Description	Reference Conditions	Min (ns)	Max (ns)
t_{Ah}	A-10	Address Hold	After R.E. CTTL	0.0	
t_{Av}	A-10	Address Valid	After R.E. CTTL, T1 or T2W3		12.0
t_{CASa}	A-10	$\overline{CAS}$ Active	After R.E. CTTL, T2W or T2W1RF		12.0
t_{CASh}	A-10	$\overline{CAS}$ Hold	After R.E. CTTL	0.0	
t_{CASia}	A-10	$\overline{CAS}$ Inactive	After R.E. CTTL, T3 or T3RF		12.0
t_{CASLw}	A-13	DRAM, PDM, $\overline{CAS}$ Width	At 0.8V, Both Edges	600.0	
t_{CCLKa}	A-14	CCLK Active	After R.E. CTTL		12.0
t_{CCLKh}	A-14	CCLK Hold	After R.E. CTTL	0.0	
t_{CCLKia}	A-14	CCLK Inactive	After R.E. CTTL		12.0
t_{CDOh}	A-14	CDOUT Hold	After R.E. CTTL	0.0	
t_{CDOv}	A-14	CDOUT Valid	After R.E. CTTL		12.0
t_{CTp}	A-21	CTTL Clock Period (Note A)	R.E. CTTL to next R.E. CTTL	48.8	50,000
t_{DCSh}	A-17	Data Hold after $\overline{EMCS}$ (Note B)	R.E. $\overline{EMCS}$ to R.E. Data Float	10.0	
t_{Df}	A-11	Data and RA11 Float (D0:7) (Note B)	After R.E. CTTL, T3 or T3H	$t_{CTp}/2 - 6$	$t_{CTp}/2 + 12$
t_{Dh}	A-11	Data Hold (D0:7)	After R.E. CTTL, T3 or T3H	$t_{CTp}/2 - 6$	
t_{Dv}	A-11	Data Valid (D0:7)	After R.E. CTTL, T2 or T2W1		12.0
t_{DWEa}	A-11	$\overline{DWE}$ Active	After R.E. CTTL, T2W2		12.0
t_{DWEh}	A-11	$\overline{DWE}$ Hold	After R.E. CTTL	0.0	
t_{DWEia}	A-11	$\overline{DWE}$ Inactive	After R.E. CTTL, T3		12.0
t_{EMCSa}	A-16	$\overline{EMCS}$ Active	After R.E. CTTL, T2W1		12.0
t_{EMCSh}	A-16	$\overline{EMCS}$ Hold	After R.E. CTTL	0.0	
t_{EMCSia}	A-16	$\overline{EMCS}$ Inactive	After R.E. CTTL T3		12.0
t_{FSa}	A-14	CFS0 Active	After R.E. CTTL		25.0
t_{FSh}	A-14	CFSO Hold	After R.E. CTTL	0.0	
t_{FSia}	A-14	CFSO Inactive	After R.E. CTTL		25.0
t_{MWDOf}	A-18	MICROWIRE Data Float (Note B)	After R.E. $\overline{MWCS}$		70.0
t_{MWDOh}	A-18	MICROWIRE Data Out Hold (Note B)	After F.E. MWCK	0.0	
t_{MWDOnf}	A-18	MICROWIRE Data No Float (Note B)	After F.E. $\overline{MWCS}$	0.0	70.0
t_{MWDOv}	A-18	MICROWIRE Data Out Valid (Note B)	After F.E. MWCK		70.0
t_{MWITOp}	A-19	MWDIN to MWDOUT	Propagation Time		70.0
t_{MWDRYa}	A-18	$\overline{MWDRY}$ Active	After R.E. of CTTL	0.0	35.0
$t_{MWDRYia}$	A-18	$\overline{MWDRY}$ Inactive	After F.E. MWCLK	0.0	70.0
t_{PABCh}	A-20	PB and $\overline{MWRQST}$	After R.E. CTTL	0.0	
t_{PABCv}	A-20	PB and $\overline{MWRQST}$	After R.E. CTTL, T2W1		12.0
t_{RASa}	A-10	$\overline{RAS}$ Active	After R.E. CTTL, T2W1 or T2WRF		12.0

Appendix A Device Specifications (Continued)

TABLE A-2. Timing for Output Signals (Continued)

Symbol	Figure	Description	Reference Conditions	Min (ns)	Max (ns)
t_{RASh}	A-10	$\overline{RAS}$ Hold	After R.E. CTTL	0.0	
t_{RASia}	A-10	$\overline{RAS}$ Inactive	After R.E. CTTL, T3 or T3RF		12.0
t_{RASLw}	A-13	DRAM PDM, $\overline{RAS}$ Width	At 0.8V, Both Edges	200.0	
t_{RLCL}	A-13	DRAM PDM $\overline{RAS}$ Low, after $\overline{CAS}$ Low	F.E. $\overline{CAS}$ to F.E. $\overline{RAS}$	200.0	
t_{WRa}	A-17	$\overline{WR0}$ Active	After R.E. CTTL, T1		$t_{CTp}/2 + 12$
t_{WRCSh}	A-17	$\overline{WR0}$ Hold after $\overline{EMCS}$ (Note B)	R.E. $\overline{EMCS}$ R.E. to R.E. $\overline{WR0}$	10.0	
t_{WRh}	A-17	$\overline{WR0}$ Hold	After R.E. CTTL	$t_{CTp}/2 - 6$	
t_{WRia}	A-17	$\overline{WR0}$ Inactive	After R.E. CTTL, T3		$t_{CTp}/2 + 12$

Note A: In normal operation t_{CTp} must be 48.8 ns; in power-down mode, t_{CTp} must be 50,000 ns.

Note B: Guaranteed by design, but not fully tested.

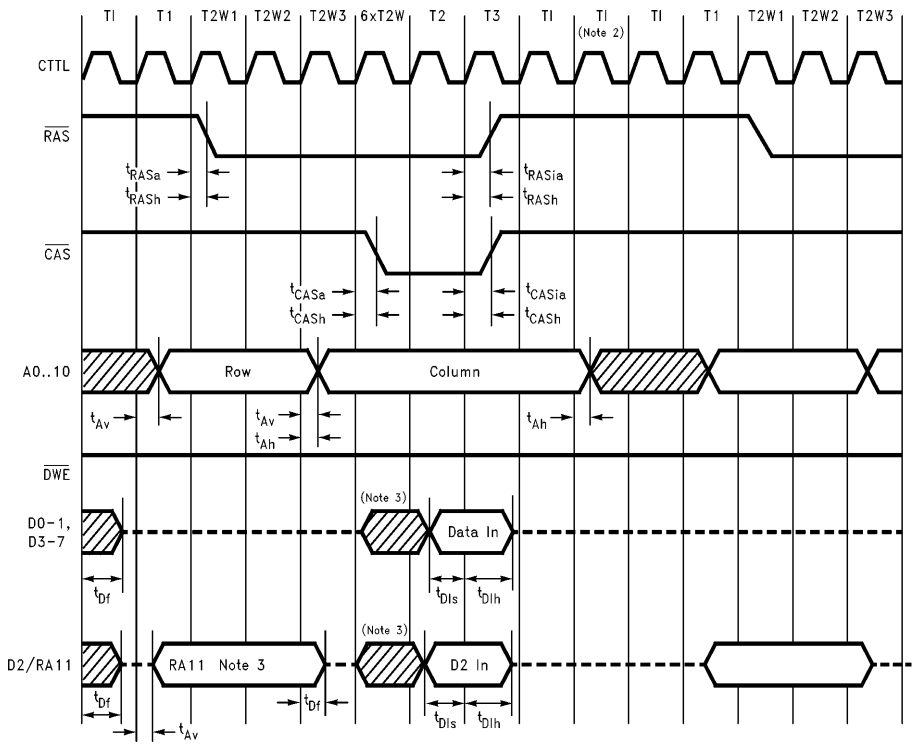
Table A-3. Input Signals

Symbol	Figure	Description	Reference Conditions	Min (ns)
t_{CDIh}	A-14	CDIN Hold	After R.E. CTTL	0.0
t_{CDIs}	A-14	CDIN Setup	Before R.E. CTTL	11.0
t_{DIh}	A-10	Data in Hold (D0:7)	After R.E. CTTL T1, T3 or TI	0.0
t_{DIs}	A-10	Data in Setup (D0:7)	Before R.E. CTTL T1, T3 or TI	15.0
t_{MWCKh}	A-18	MICROWIRE Clock High (Slave)	At 2.0V (Both Edges)	100.0
t_{MWCKl}	A-18	MICROWIRE Clock Low (Slave)	At 0.8V (Both Edges)	100.0
t_{MWCKp}	A-18	MICROWIRE Clock Period (Slave) (Note A)	R.E. MWCLK to next R.E. MWCLK	2.5 ms
t_{MWCLKh}	A-18	MWCLK Hold	After $\overline{MWCS}$ becomes Inactive	50.0
t_{MWCLKs}	A-18	MWCLK Setup	Before $\overline{MWCS}$ becomes Active	100
t_{MWCSh}	A-18	$\overline{MWCS}$ Hold	After F.E. MWCLK	50.0
t_{MWCSs}	A-18	$\overline{MWCS}$ Setup	Before R.E. MWCLK	100.0
t_{MWDIh}	A-18	MWDIN Hold	After R.E. MWCLK	50.0
t_{MWDIs}	A-18	MWDIN Setup	Before R.E. MWCLK	100.0
t_{PWR}	A-23	Power Stable to $\overline{RESET}$ R.E. (Note B)	After V_{CC} reaches 4.5V	30 ms
t_{RSTw}	A-22	$\overline{RESET}$ Pulse Width	At 0.8V (Both Edges)	10 ms
t_{XIh}	A-21	CLKIN High	At 2.0V (Both Edges)	$t_{X1p}/2 - 5$
t_{XIl}	A-21	CLKIN Low	At 0.8V (Both Edges)	$t_{X1p}/2 - 5$
t_{XIp}	A-21	CLKIN Clock Period	R.E. CLKIN to next R.E. CLKIN	24.4

Note A: Guaranteed by design, but not fully tested in power-down mode.

Note B: Guaranteed by design. but not fully tested.

A.4.3 TIMING DIAGRAMS



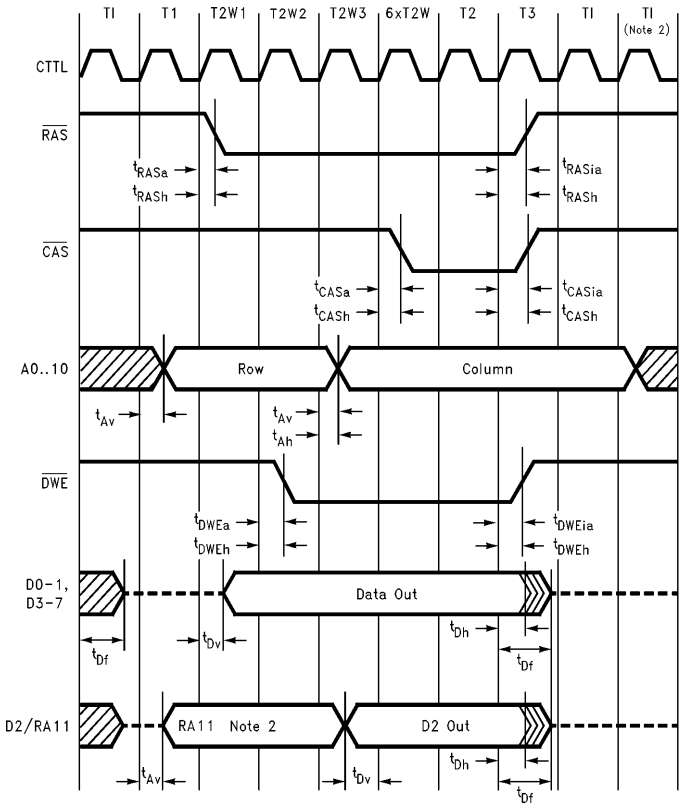
TL/EE/12378–26

Note 1: 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.

Note 2: An external device can drive data from T2W3 to T3.

Note 3: An external device can not drive data from T1 to T2W3.

FIGURE A-10. DRAM Read Cycle Timing (NSAM265SR only)

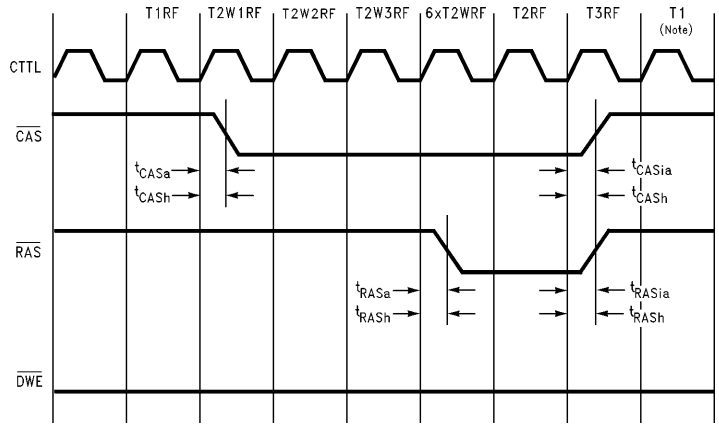


TL/EE/12378–27

Note 1: 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.

Note 2: An external device can not drive data from T1 to T2W3.

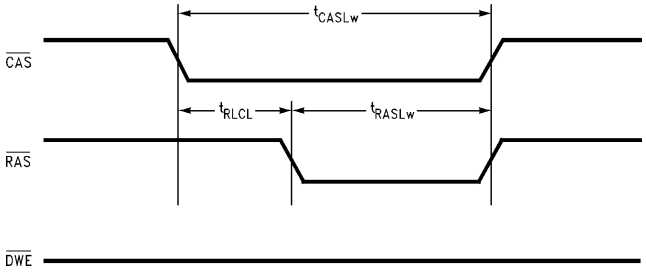
FIGURE A-11. DRAM Write Cycle Timing (NSAM265SR only)



TL/EE/12378–28

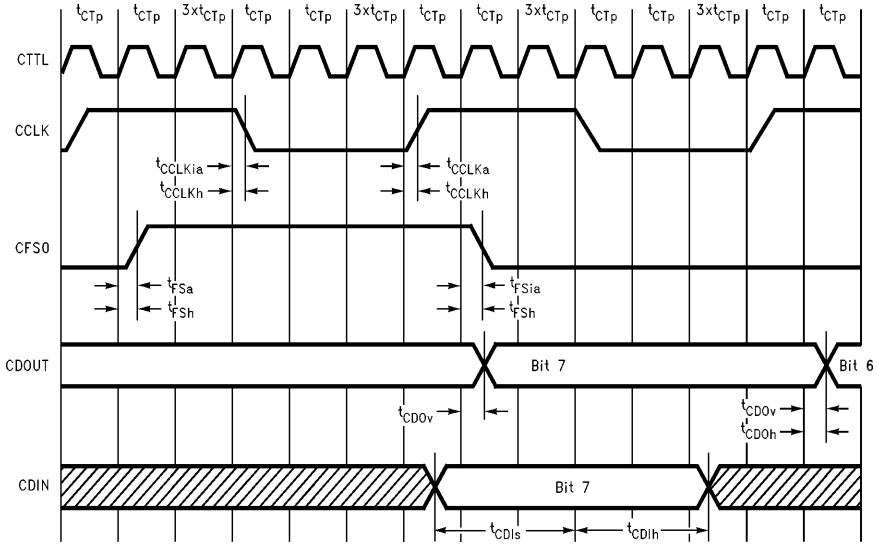
Note: This cycle may be either TI (Idle) or T1 of any non-DRAM bus cycle. If the next bus cycle is a DRAM one, T3RF is followed by three TI (Idle) cycles.

FIGURE A-12. DRAM Refresh Cycle Timing (Normal Operation)



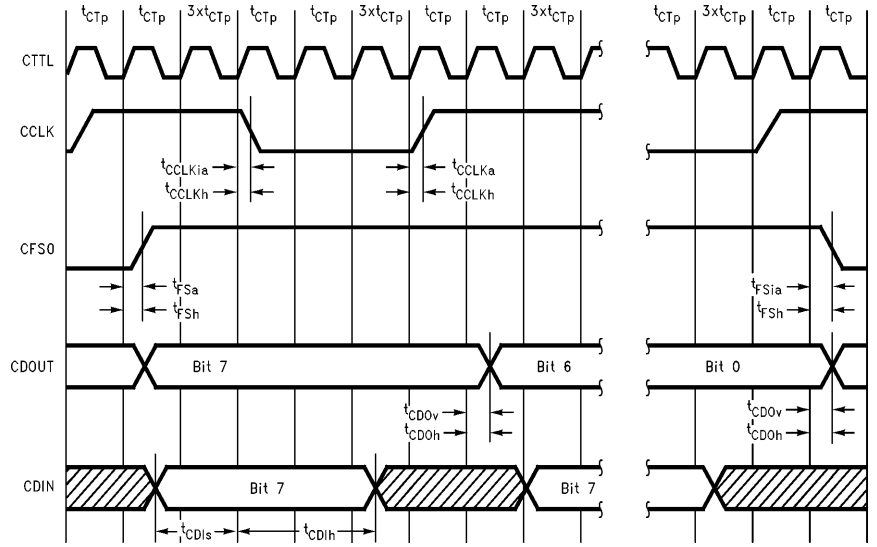
TL/EE/12378–29

FIGURE A-13. DRAM Power-Down Refresh Cycle Timing



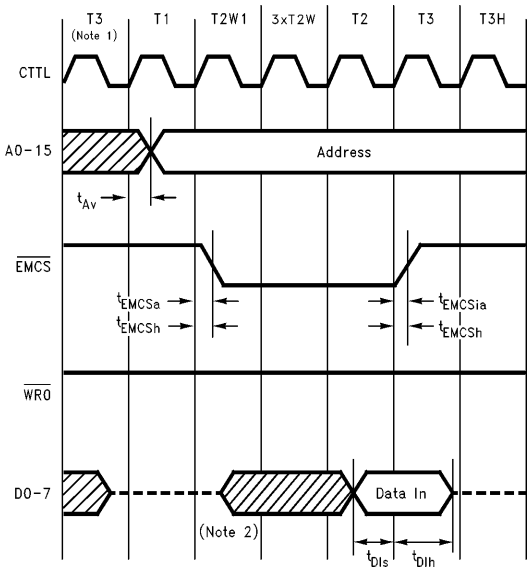
TL/EE/12378–30

FIGURE A-14. Codec Short Frame Timing



TL/EE/12378–31

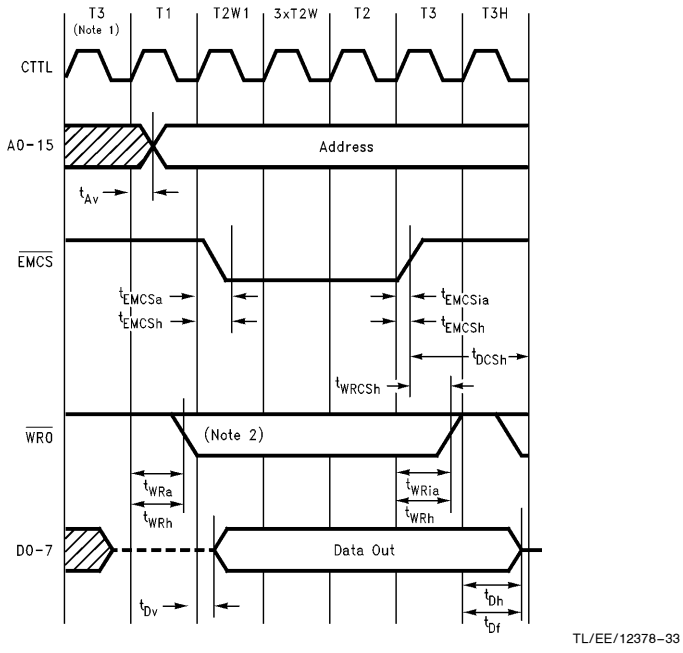
FIGURE A-15. Codec Long Frame Timing



Note 1: This cycle may be either TI (Idle), T3 or T3H.

Note 2: Data can be driven by an external device at T2W1, T2W, T2 and T3.

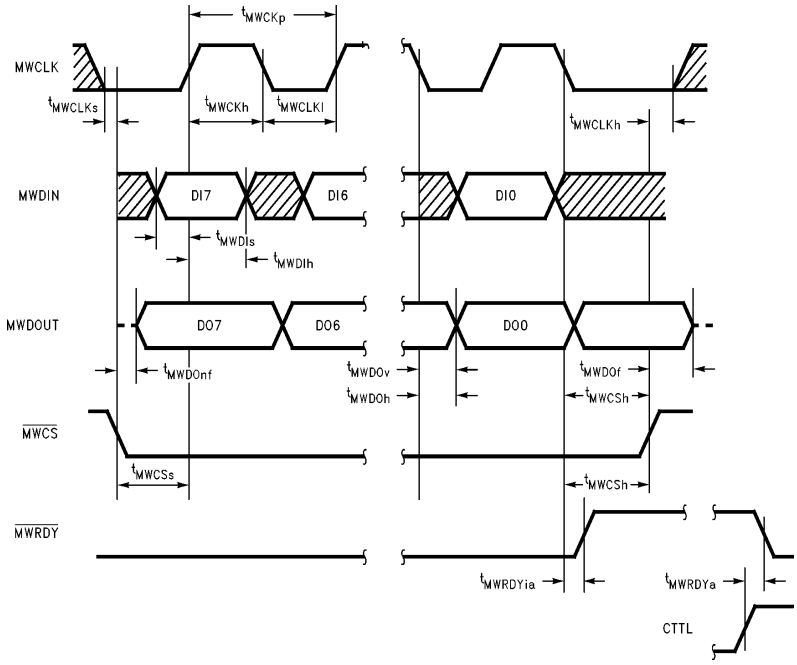
FIGURE A-16. ROM/FLASH Read Cycle Timing



Note 1: This cycle may be either TI (Idle), T3 or T3H.

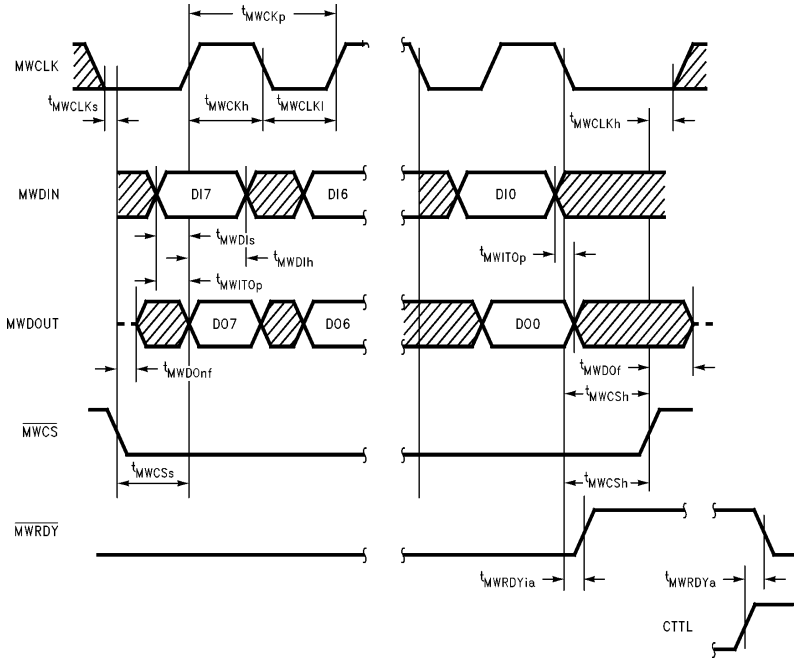
Note 2: Depends on which bytes are written.

FIGURE A-17. FLASH Write Cycle Timing (NSAM265SF only)



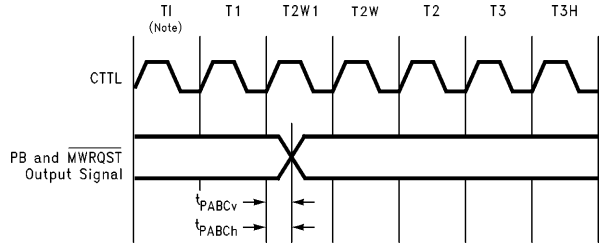
TL/EE/12378–34

FIGURE A-18. MICROWIRE Transaction Timing—Data Transmitted to Output



TL/EE/12378–35

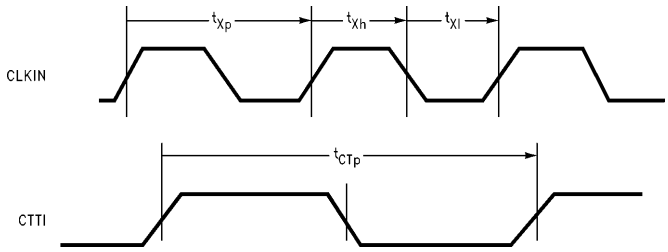
FIGURE A-19. MICROWIRE Transaction Timing—Data Echoed to Output



TL/EE/12378–36

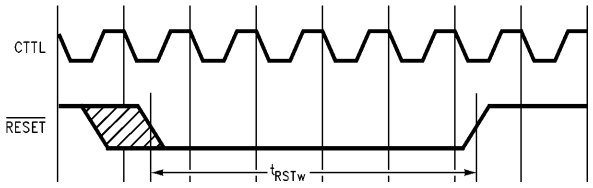
Note 1: This cycle may be either TI (Idle), T2, T3 or T3H.

FIGURE A-20. Output Signal Timing for Port PB and $\overline{\text{MWRQST}}$



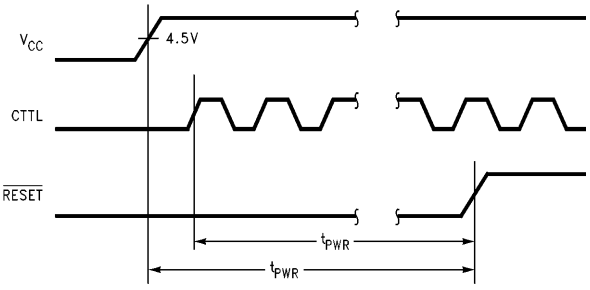
TL/EE/12378–37

FIGURE A-21. CTTL and CLKIN Timing



TL/EE/12378–38

FIGURE A-22. Reset Timing When Reset Is Not At Power-Up



TL/EE/12378–39

FIGURE A-23. Reset Timing When Reset Is At Power-Up

Appendix B

Schematic Diagrams

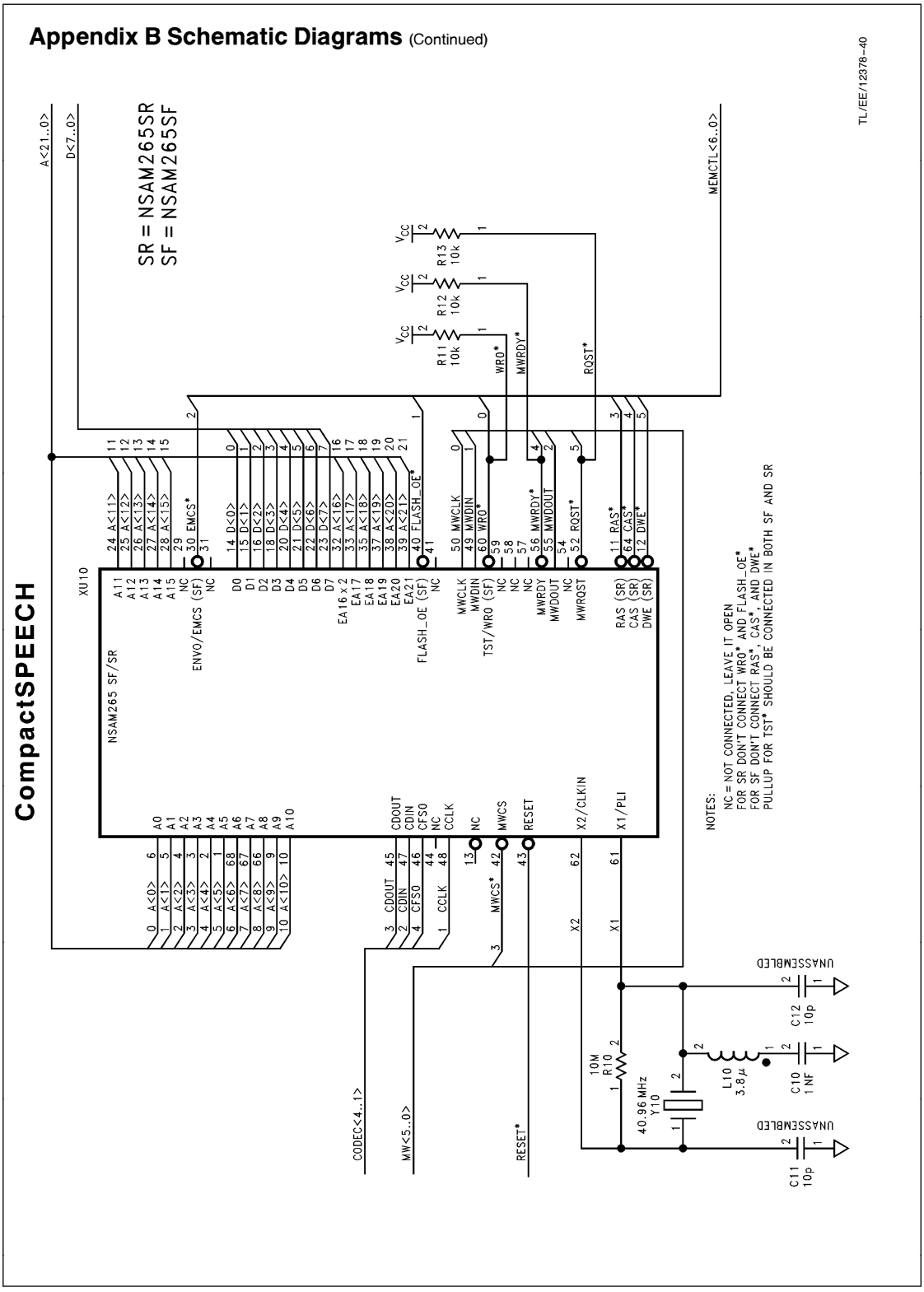
The following schematic diagrams are extracted from a CompactSPEECH demo unit, based on the NSV-AM265-SPAF board, designed by National Semiconductor.

This demo includes three basic clusters:

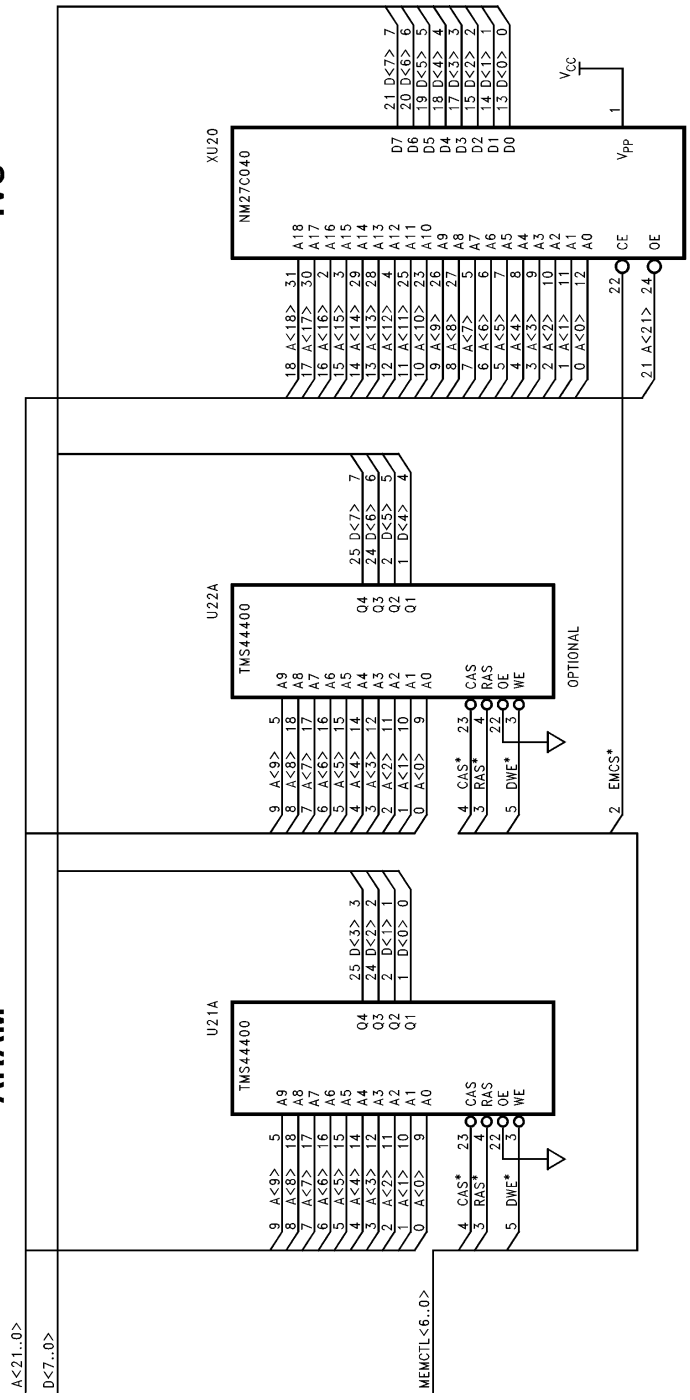
- COP888EEG Microcontroller
- CompactSPEECH cluster, including a TP3054 codec and, either an NSAM265SR controlling two 1M x 4 ARAMs, or an NSAM265SF connected to one (AMD or INTEL) FLASH device.
- User interface that includes one 16-digit LCD, and 16-key (4x4) key-pad.

For more details about the demo please refer to the *NS Digital Answering Machine Demo Operating Instructions*.

Appendix B Schematic Diagrams (Continued)



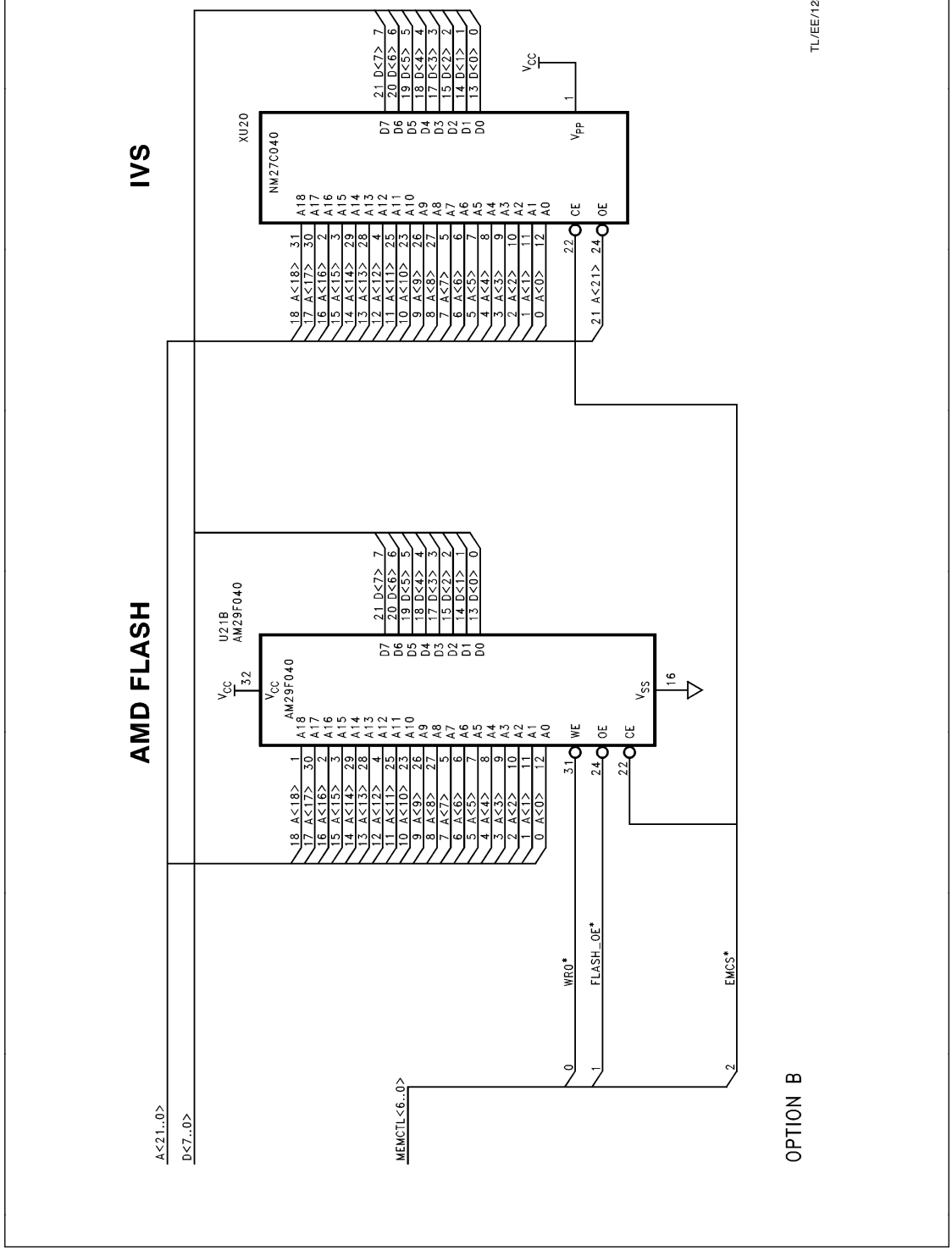
Appendix B Schematic Diagrams (Continued)



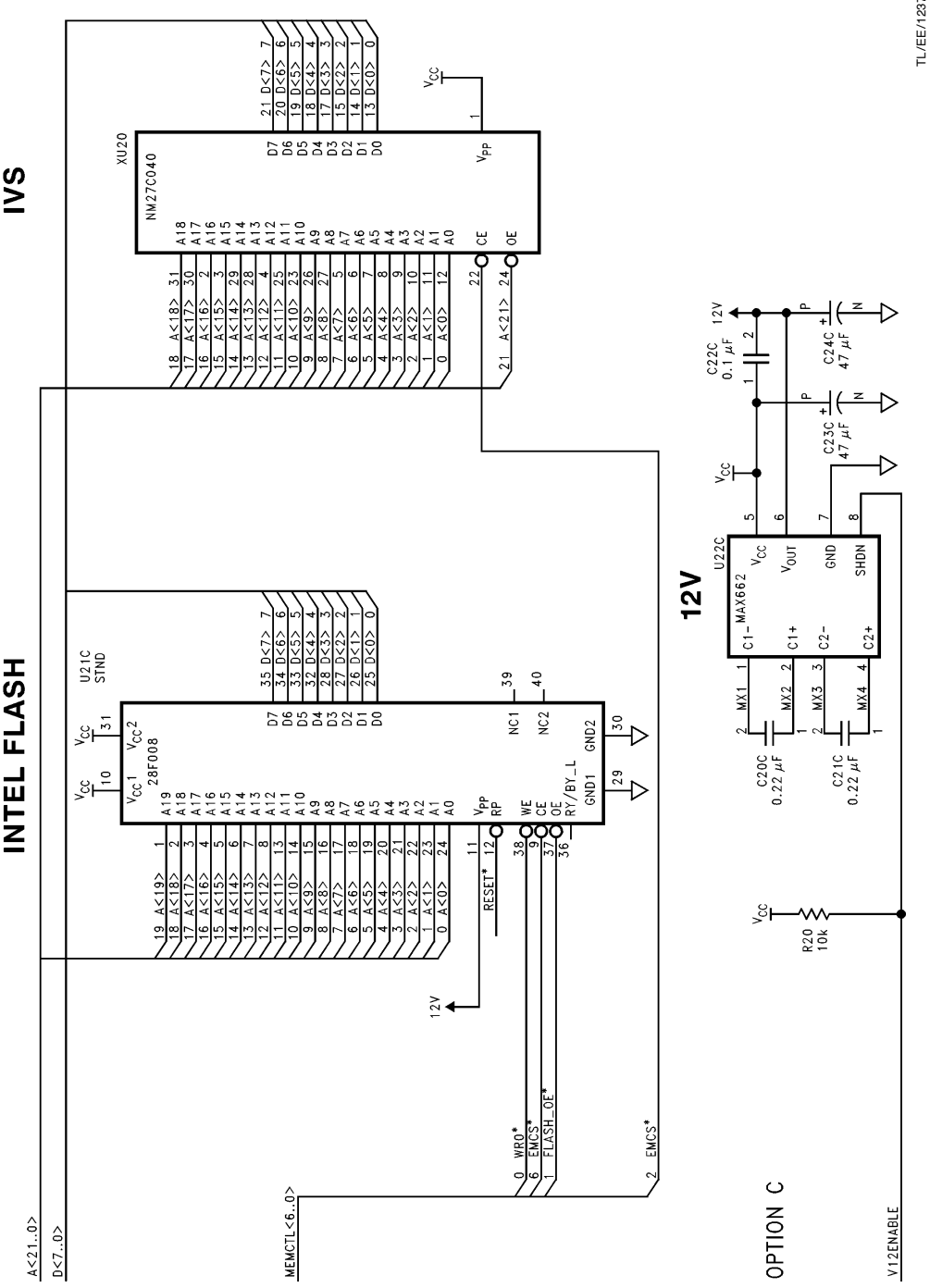
TL/EE/12378–41

OPTION A

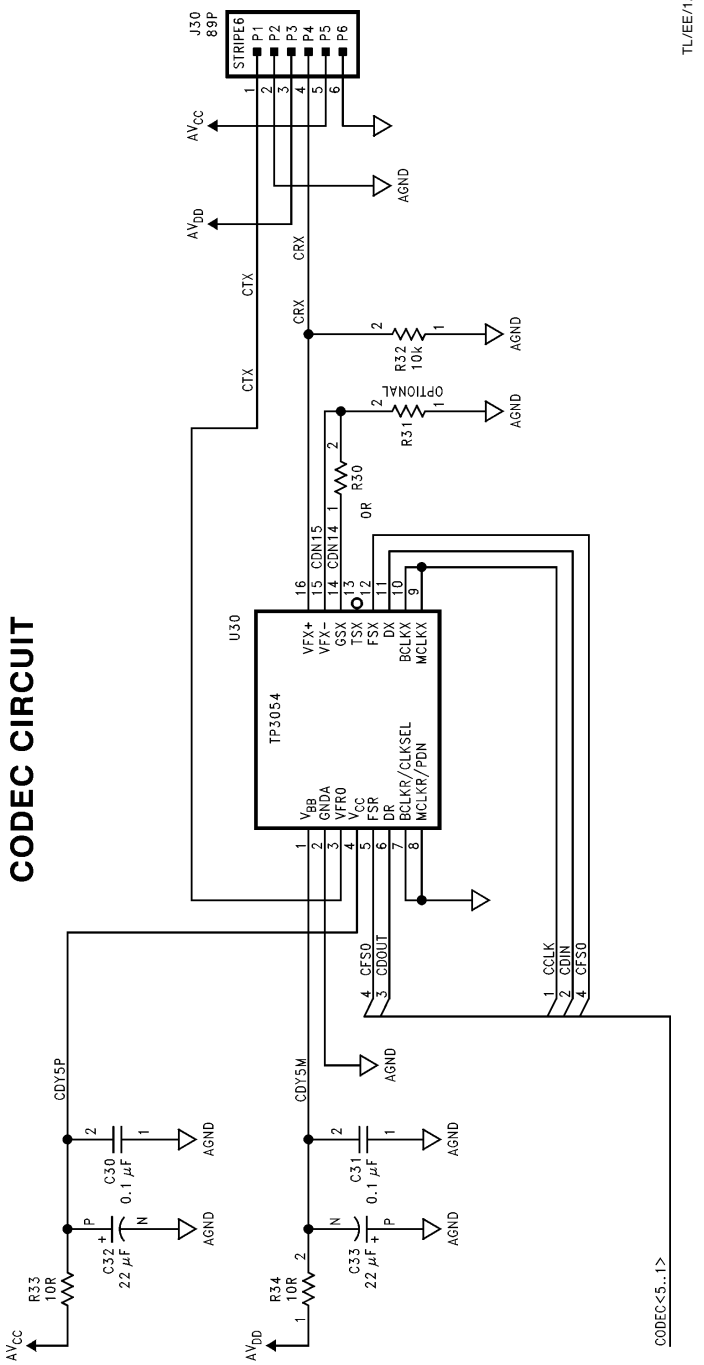
Appendix B Schematic Diagrams (Continued)



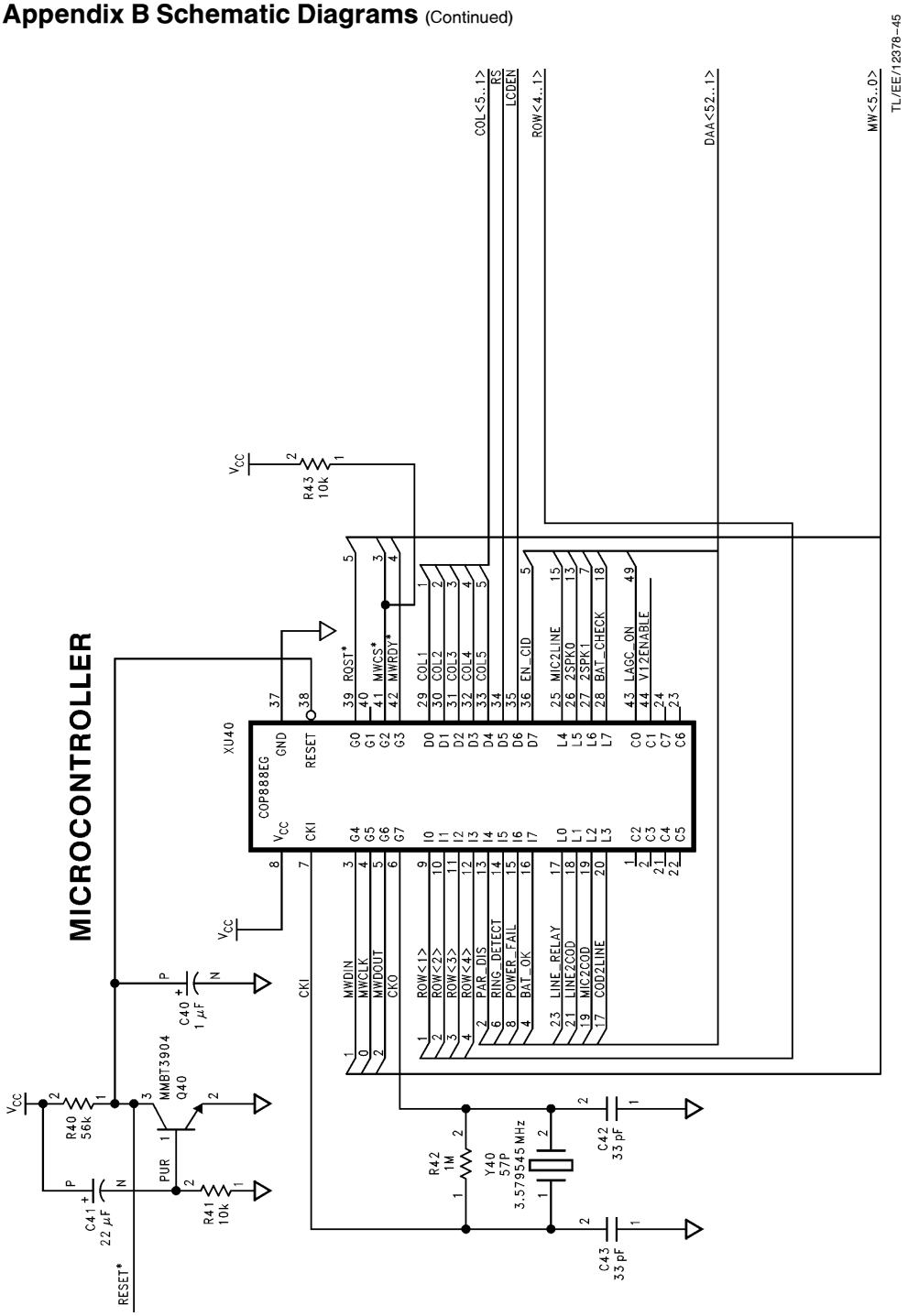
Appendix B Schematic Diagrams (Continued)



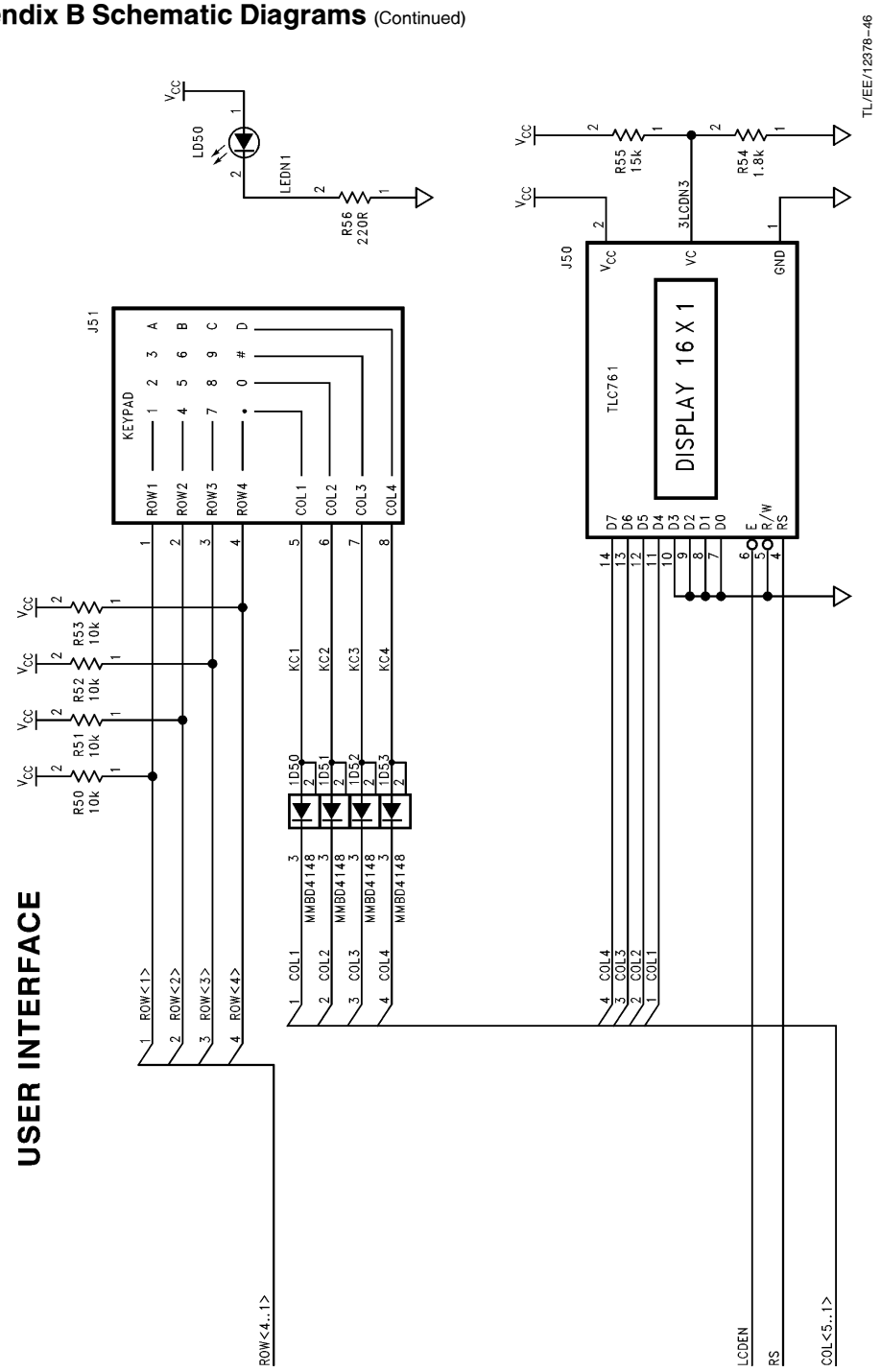
Appendix B Schematic Diagrams (Continued)

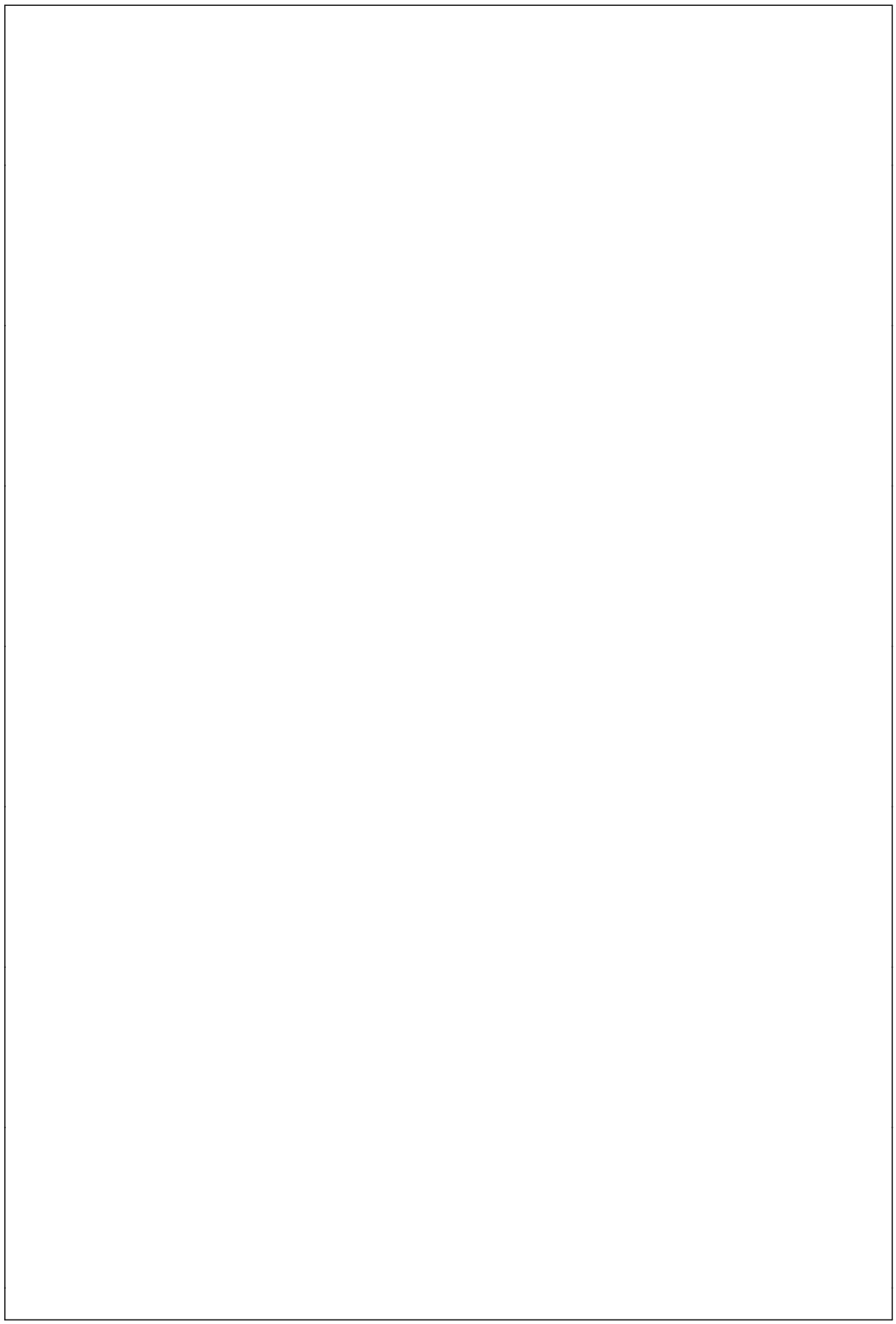


Appendix B Schematic Diagrams (Continued)

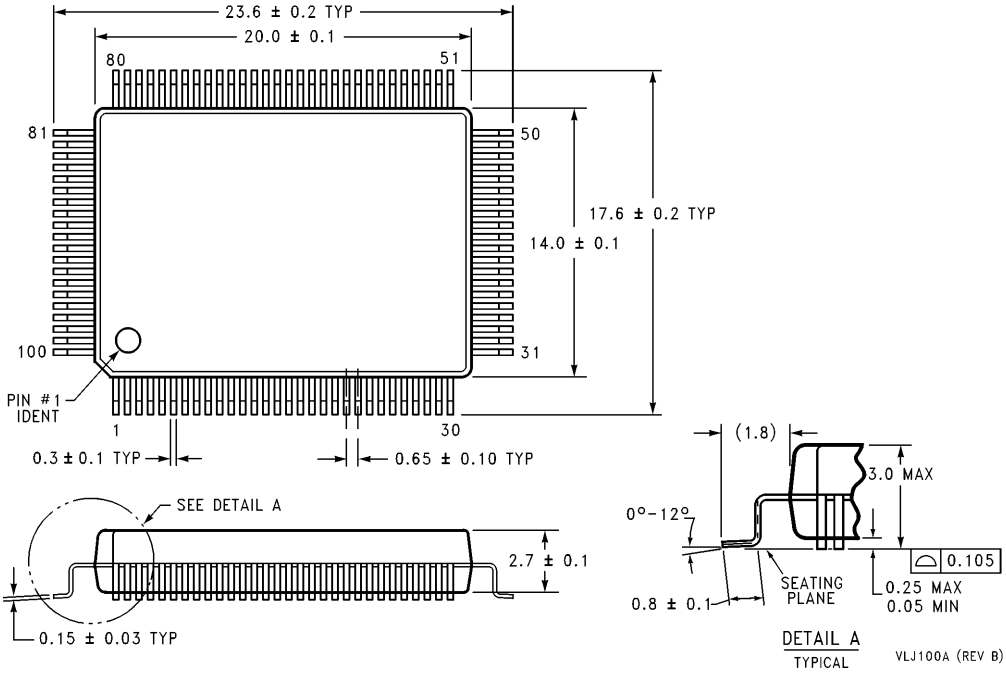


Appendix B Schematic Diagrams (Continued)



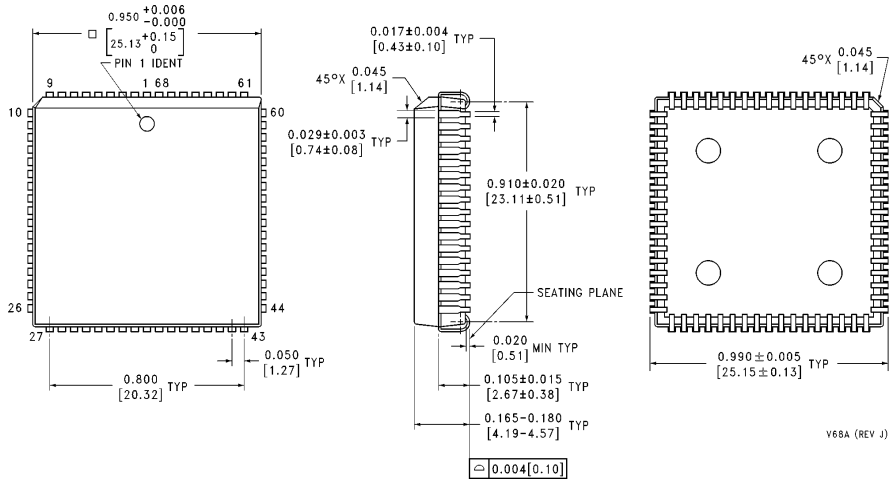


Physical Dimensions inches (millimeters)



100-Pin Molded Plastic Quad Flat Package (EIAJ)
Order Number NSAM265SRA/SFA
NS Package Number VLJ100A

Physical Dimensions inches (millimeters) (Continued)



68-Pin Plastic Leaded Chip Carrier (V)
Order Number NSAM265SRA/SFA
NS Package Number V68A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408