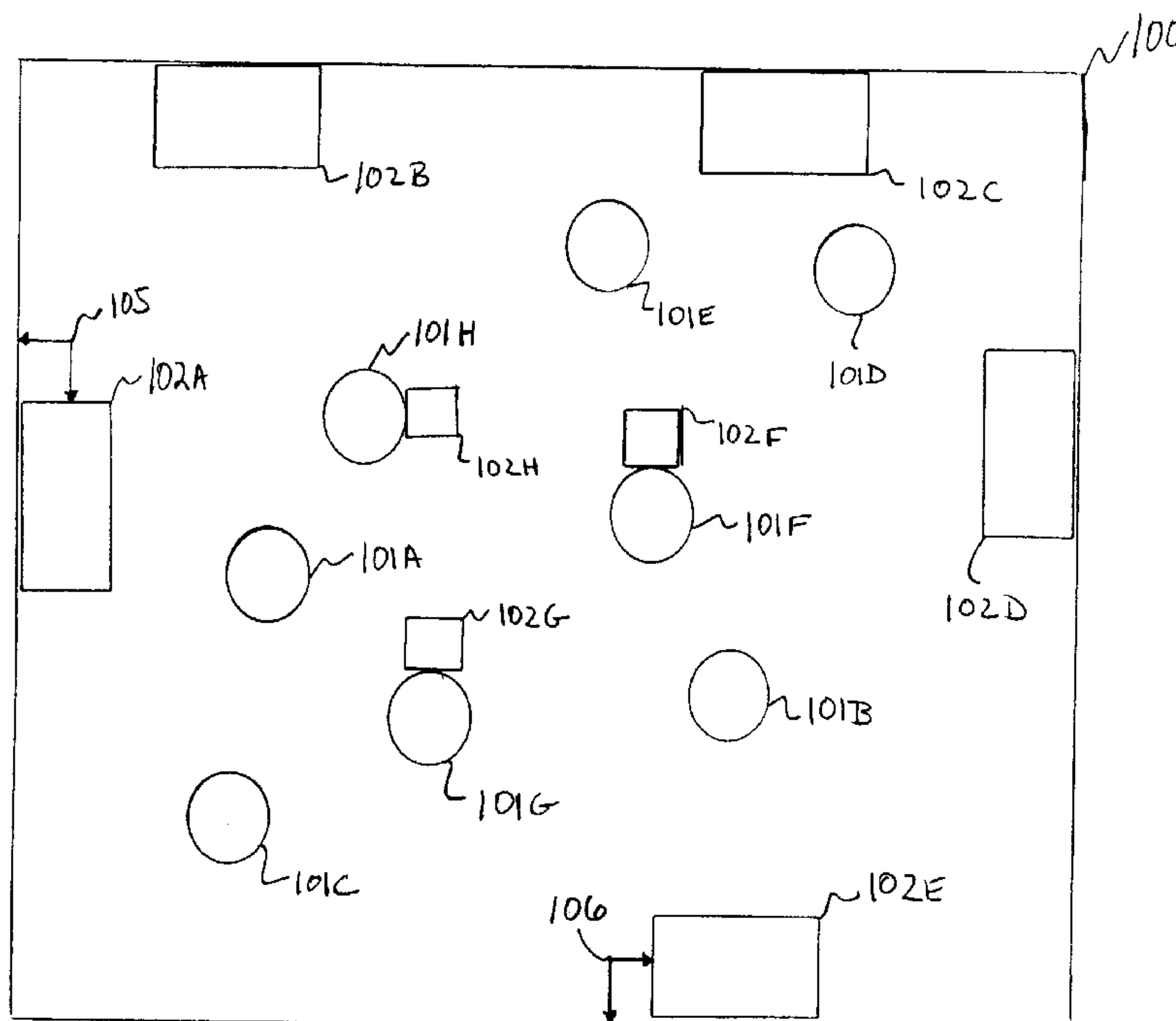




- (72) GEILHUF, MICHAEL, US
(72) MACMILLAN, DAVID, US
(72) BAREL, AVRAHAM, IL
(72) BROWN, AMOS, IL
(72) BOOTSMA, KARIN LISSETTE, US
(72) GADDY, LAWRENCE KENT, US
(72) PYO, PHILLIP PAUL, US
(71) INFORMATION STORAGE DEVICES, INC., US
(51) Int.Cl.⁷ G06F 3/16, G06F 13/38, G10L 15/00
(30) 1999/05/21 (09/316,643) US
(54) **METHODE ET DISPOSITIF DE COMMANDE DE DISPOSITIF A
COMMANDE VOCALE**
(54) **METHOD AND APPARATUS FOR CONTROLLING VOICE
CONTROLLED DEVICES**



(57) Voice controlled devices with speech recognition have user assignable appliance names and default appliance names to address and control the voice controlled devices. Methods of controlling voice controlled devices include addressing a voice controlled device by name and providing a command.

-162-

ABSTRACT OF DISCLOSURE

Voice controlled devices with speech recognition have user assignable appliance names and default appliance names to address and control the voice controlled devices. Methods of controlling voice controlled devices include addressing a voice controlled device by name and providing a command.

-1-

**METHOD AND APPARATUS
FOR
CONTROLLING VOICE CONTROLLED DEVICES**

MICROFICHE APPENDIX

This application contains a microfiche appendix which is not printed herewith entitled "ISD-SR 300, Embedded Speech Recognition Processor" by Information Storage Devices, Inc. which is hereby incorporated by reference, verbatim and with the same effect as though it were fully and completely set forth herein.

FIELD OF THE INVENTION

This invention relates generally to machine interfaces. More particularly, the invention relates to voice user interfaces for devices.

BACKGROUND OF THE INVENTION

Graphical user interfaces (GUIs) for computers are well known. GUIs provide an intuitive and consistent manner for human interaction with computers. Generally, once a person learns how to use a particular GUI, they can operate any computer or device which operates using the same or similar GUI. Examples of popular GUIs are MAC OS by Apple, and MS Windows by Microsoft. GUIs are now being ported to other devices. For example, the MS Windows GUI has been ported from computers to palm tops,

personal organizers, and other devices so that there is a common GUI amongst a number of differing devices. However, as the name implies, GUIs require at least some sort of visual or graphical display and an input device such as a keyboard, mouse, touch pad or touch screen. The displays and the input devices tend to utilize space in an device, require additional components and increase the costs of an device. Thus, it is desirable to eliminate the display and input devices from devices to save costs.

Recently, voice user interfaces (VUIs) have been introduced that utilize speech recognition methods to control a device. However, these prior art VUIs have a number of shortcomings that prohibit them from being universally utilized in all devices. Prior art VUIs are usually difficult to use. Prior art VUIs usually require some sort of display device such as an LCD, or require a manual input device such as keypads or buttons, or require both a display and a manual input device. Additionally, prior art VUIs usually are proprietary and restricted in use to a single make or model of hardware device, or a single type of software application. They usually are not widely available, unlike computer operating systems, and accordingly software programmers can not write applications that operate with the VUI in a variety of device types. Commands associated with prior art VUIs are usually

customized for that single type of device or software application. Prior art VUIs usually have additional limitations in supporting multiple users such as how to handle personalization and security. Furthermore, prior art VUIs require that a user know of the existence of the device in advance. Prior art VUIs have not provided ways of determining the presence of devices. Additionally, prior art VUIs usually require a user to read instruction manuals or screen displayed commands to become trained in their use. Prior art VUIs usually do not include audible methods for a user to learn commands. Furthermore, a user may be required to learn how to use multiple prior art VUIs when utilizing multiple voice controlled devices due to a lack of standardization.

Generally, devices controlled by VUIs continue to require some sort of manual control of functions. With some manual control required, a manual input device such as a button, keypad or a set of buttons or keypads is provided. To assure proper manual entry, a display device such as an LCD, LED, or other graphics display device may be provided. For example, many voice activated telephones require that telephone numbers be stored manually. In this case a numeric keypad is usually provided for manual entry. An LCD is usually included to assure proper manual entry and to display the status of the device. A speech synthesis or voice

feedback system may be absent from these devices. The addition of buttons and display devices increases the manufacturing cost of devices. It is desirable to be able to eliminate all manual input and display from devices in order to decrease costs. Furthermore, it is more convenient to remotely control devices without requiring specific buttons or displays.

Previously, devices were used by few. Additionally they used near field microphones to listen locally for voices. Many prior devices were fixed in some manner or not readily portable or were server based systems. It is desirable to provide voice control capability for portable devices. It is desirable to provide either near field or far field microphone technology in voice controlled devices. It is desirable to provide low cost voice control capability such that it is included in more devices. However, these desires raise a problem when multiple users of multiple voice controlled devices are in the same area. With multiple users and multiple voice controlled devices within audible range of each other, it makes it difficult for voice controlled devices to discern which user to accept commands from and respond to. For example, consider the case of voice controlled cell phones where one user in an environment of multiple users wants to call home. The user issues a voice activated call home command. If more than one voice controlled cell phone audibly hears the call home

command, multiple voice controlled cell phones may respond and start dialing a home telephone number. Previously this was not as significant a problem because there were few voice controlled devices.

Some voice controlled devices are speaker dependent. Speaker dependency refers to a voice controlled device that requires training by a specific user before it may be used with that user. A speaker dependent voice controlled device listens for tonal qualities in how phrases are spoken. Speaker dependent voice controlled devices do not lend themselves to applications where multiple users or speakers are required to use the voice controlled device. This is because they fail to efficiently recognize speech from users that they have not been trained by. It is desirable to provide speaker independent voice controlled devices with a VUI requiring little or no training in order to recognize speech from any user.

In order to achieve high accuracy speech recognition it is important that a voice controlled device avoid responding to speech that isn't directed to it. That is, voice controlled devices should not respond to background conversation, to noises, or to commands to other voice controlled devices. However, filtering out background sounds must not be so effective that it also prevents recognition of speech directed to the voice controlled device. Finding the right mix of

rejection of background sounds and recognition of speech directed to a voice controlled device is particularly challenging in speaker-independent systems. In speaker-independent systems, the voice controlled device must be able to respond to a wide range of voices, and therefore can not use a highly restrictive filter for background sounds. In contrast, a speaker-dependant system need only listen for a particular person's voice, and thus can employ a more stringent filter for background sounds. Despite this advantage in speaker dependant systems, filtering out background sounds is still a significant challenge.

In some prior art systems, background conversation has been filtered out by having a user physically press a button in order to activate speech recognition. The disadvantage of this approach is that it requires the user to interact with the voice controlled device physically, rather than strictly by voice or speech. One of the potential advantages of voice controlled devices is that they offer the promise of true hands-free operation. Elimination of the need to press a button to activate speech recognition would go a long way to making this hands-free objective achievable.

Additionally, in locations with a number of people talking, a voice controlled device should disregard all speech unless it is directed to it. For example, if a person says to another person "I'll call John", the

-7-

cellphone in his pocket should not interpret the "call John" as a command. If there are multiple voice controlled devices in one location, there should be a way to uniquely identify which voice controlled device a user wishes to control. For example, consider a room that may have multiple voice controlled telephones - perhaps a couple of desktop phones, and multiple cellphones - one for each person. If someone were to say "Call 555-1212", each phone may try to place the call unless there was a means for them to disregard certain commands. In the case where a voice controlled device is to be controlled by multiple users, it is desirable for the voice controlled device to know which user is commanding it. For example, a voice controlled desktop phone in a house may be used by a husband, wife and child. Each would could have their own phonebook of frequently called numbers. When the voice controlled device is told "Call Mother", it needs to know which user is issuing the command so that it can call the right person (i.e. should it call the husbands mother, the wife's mother, or the child's mother at her work number?). Additionally, a voice controlled device with multiple users may need a method to enforce security to protect it from unauthorized use or to protect a user's personalized settings from unintentional or malicious interactions by others (including snooping, changing, deleting, or adding to the settings). Furthermore, in a

location where there are multiple voice controlled devices, there should be a way to identify the presence of voice controlled devices. For example, consider a traveler arriving at a new hotel room. Upon entering the hotel room, the traveler would like to know what voice controlled devices may be present and how to control them. It is desirable that the identification process be standardized so that all voice controlled devices may be identified in the same way.

In voice controlled devices, it is desirable to store phrases under voice control. A phrase is defined as a single word, or a group of words treated as a unit.

This storing might be to set options or create personalized settings. For example, in a voice-controlled telephone, it is desirable to store people's names and phone numbers under voice control into a personalized phone book. At a later time, this phone book can be used to call people by speaking their name (e.g. "Cellphone call John Smith", or "Cellphone call Mother").

Prior art approaches to storing the phrase ("John Smith") operate by storing the phrase in a compressed, uncompressed, or transformed manner that attempts to preserve the actual sound. Detection of the phrase in a command (i.e. detecting that John is to be called in the example above) then relies on a sound-based comparison between the original stored speech sound and the spoken

command. Sometimes the stored waveform is transformed into the frequency domain and / or is time adjusted to facilitate the match, but in any case the fundamental operation being performed is one that compares the actual sounds. The stored sound representation and comparison for detection suffers from a number of disadvantages. If a speaker's voice changes, perhaps due to a cold, stress, fatigue, noisy or distorting connection by telephone, or other factors, the comparison typically is not successful and stored phrases are not recognized. Because the phrase is stored as a sound representation, there is no way to extract a text-based representation of the phrase. Additionally, storing a sound representation results in a speaker dependent system. It is unlikely that another person could speak the same phrase using the same sounds in a command and have it be correctly recognized. It would not be reliable, for example, for a secretary to store phonebook entries and a manager to make calls using those entries. It is desirable to provide a speaker independent storage means. Additionally, if the phrases are stored as sound representations, the stored phrases can not be used in another voice controlled device unless the same waveform processing algorithms are used by both voice controlled devices. It is desirable to recognize spoken phrases and store them in a representation such that, once stored, the phrases can

-10-

be used for speaker independent recognition and can be used by multiple voice controlled devices.

Presently computers and other devices communicate commands and data to other computers or devices using modem, infrared or wireless radio frequency transmission. The transmitted command and/or data are usually of a digital form that only the computer or device may understand. In order for a human user to understand the command or data it must be decoded by a computer and then displayed in some sort of format such as a number or ASCII text on a display. When the command and/or data are transmitted they are usually encoded in some digital format understood by the computer or devices or transmitting equipment. As voice controlled devices become more prevalent, it will be desirable for voice controlled devices to communicate with each other using human-like speech in order to avoid providing additional circuitry for communication between voice controlled devices. It is further desirable to allow multiple voice controlled devices to exchange information machine-to-machine without human user intervention.

BRIEF SUMMARY OF THE INVENTION

The present invention includes a method, apparatus and system as described in the claims. Briefly, a

-11-

standard voice user interface is provided to control various devices by using standard speech commands. The standard VUI provides a set of standard VUI commands and syntax for the interface between a user and the voice controlled device. The standard VUI commands include an identification phrase to determine if voice controlled devices are available in an environment. Other standard VUI commands provide for determining the names of the voice controlled devices and altering them.

Voice controlled devices are disclosed. A voice controlled device is defined herein as any device that is controlled by speech, which is either audible or non-audible. A voice controlled device may also be referred to herein as an appliance, a machine, a voice controlled appliance, a voice controlled electronic device, a name activated electronic device, a speech controlled device, a voice activated electronic appliance, a voice activated appliance, a voice controlled electronic device, or a self-identifying voice controlled electronic device.

In order to gain access to the functionality of voice controlled devices, a user communicates to the voice controlled device one of its associated appliance names after a period of relative silence. The appliance name may be a default name or a user-assignable name. The voice controlled device may have a plurality of

user-assignable names associated with it for providing personalized functionality to each user.

Other aspects of the present invention are described in the detailed description.

BRIEF DESCRIPTIONS OF THE DRAWINGS

FIG. 1A is an illustration of an environment containing voice controlled devices of the present invention.

FIG. 1B is an illustration of remote communications with the voice controlled devices in the environment illustrated in FIG. 1A.

FIG. 2 is an illustration of exemplary voice controlled devices.

FIG. 3 is a detailed block diagram of the voice controlled device of the present invention.

FIG. 4 is a detailed block diagram of a voice communication chip.

FIG. 5 is a block diagram of the standard voice user interface of the present invention.

FIGS. 6A-6C are flow charts of the core command structure for the standard voice user interface of the present invention.

FIGs. 6D-6E are flow charts of the telephone command structure for the standard voice user interface of the present invention.

FIG. 7 is a flow chart of the "Store Name" telephone command structure for the standard voice user interface of the present invention.

FIG. 8 is a flow chart of the "Delete Name" telephone command structure for the standard voice user interface of the present invention.

FIGs. 9A-9B are flow charts of the "GETYESNO" function for the standard voice user interface of the present invention.

FIGs. 10A-10C are flow charts of the "GETRESPONSE" function for the standard voice user interface of the present invention.

FIG. 11 is a flow chart of the "GETRESPONSEPLUS" function for the standard voice user interface of the present invention.

FIG. 12 is a flow chart of the "LISTANDSELECT" function for the standard voice user interface of the present invention.

FIG. 13 is a block diagram of a pair of voice controlled devices communicating using the standard

voice user interface of the present invention.

Like reference numbers and designations in the drawings indicate like elements providing similar functionality.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be obvious to one skilled in the art that the present invention may be practiced without these specific details. In other instances well known methods, procedures, components, and circuits have not been described in detail so as not to unnecessarily obscure aspects of the present invention.

The present invention includes a method, apparatus and system for standard voice user interface and voice controlled devices. Briefly, a standard voice user interface is provided to control various devices by using standard speech commands. The standard VUI provides a set of core VUI commands and syntax for the interface between a user and the voice controlled device. The core VUI commands include an identification phrase to determine if voice controlled devices are available in an environment. Other core VUI commands

-15-

provide for determining the names of the voice controlled devices and altering them.

Voice controlled devices are disclosed. A voice controlled device is defined herein as any device that is controlled by speech, which is either audible or non-audible. Audible and non-audible are defined herein later. A voice controlled device may also be referred to herein as an appliance, a machine, a voice controlled appliance, a voice controlled electronic device, a name activated electronic device, a speech controlled device, a voice activated electronic appliance, a voice activated appliance, a voice controlled electronic device, or a self-identifying voice controlled electronic device.

The present invention is controlled by and communicates using audible and non-audible speech. Speech as defined herein for the present invention encompasses a) a signal or information, such that if the signal or information were passed through a suitable device to convert it to variations in air pressure, the signal or information could be heard by a human being and would be considered language, and b) a signal or information comprising actual variations in air pressure, such that if a human being were to hear the signal, the human would consider it language. Audible speech refers to speech that a human can hear unassisted. Non-audible speech refers to any encodings

-16-

or representations of speech that are not included under the definition of audible speech, including that which may be communicated outside the hearing range of humans and transmission media other than air. The definition of speech includes speech that is emitted from a human and emitted from a machine (including machine speech synthesis, playback of previously recorded human speech such as prompts, or other forms).

Prompts which are communicated by a voice controlled device and phrases which are communicated by a user may be in languages or dialects other than English or a combination of multiple languages. A phrase is defined herein as a single word, or a group of words treated as a unit. A user, as defined herein, is a human or a device, including a voice activated device. Hence "a user's spoken phrase", "a user issuing a command", and all other actions by a user include actions by a device and by a human.

Voice controlled devices include some type of speech recognition in order to be controlled by speech.

Speech recognition and voice recognition are used synonymously herein and have the same meaning.

Preferably, speaker independent speech recognition systems are used to provide the speech recognition capability of the voice controlled devices. Speaker independent speech recognitions systems are responsive

-17-

to speaker-independent representations of speech. In the preferred embodiment, a speaker-independent representation of speech is a phonetic representation of speech. However, other speaker-independent representations of speech may also be used in accordance with the present invention.

In order to gain access to the full functionality of a voice controlled device with the present invention, a user must communicate to the voice controlled device one of its associated appliance names. The appliance name may include one or more default names or one or more user-assignable names. A voice controlled device may have a plurality of user-assignable names associated with it in order to provide personalized functionality to each user.

Additionally, the present invention provides a way to leave a speech recognition engine on throughout ongoing conversations (including local conversations or those over a telephone link), without having it be falsely triggered by background noise or speech that is not directed to it. To accomplish this, the invention makes use of a naming scheme for voice controlled devices provided by the standard VUI of the present invention. In general, unless a voice controlled device is addressed by its appliance name, it will disregard all speech. (There are a couple of special exceptions to this rule that will be discussed later.) In certain

-18-

cases the criteria for recognizing a command may be further tightened requiring a voice controlled device to be addressed by its user-assigned appliance name. A voice controlled device may have multiple users, each of whom assign it a unique appliance name using commands of the standard VUI of the present invention. When a voice controlled device is addressed by one of its user-assigned names, the voice controlled device can determine both that it is being addressed, and which user is addressing it. This allows the voice controlled device to use the personalized settings for that particular user. For example, a voice-activated telephone might have four different user-assigned names (e.g. Aardvark, Platypus, Socrates, and Zeus), and each user might have a different telephone number associated with the phonebook entry for Mother. When the first user says "Aardvark call mother", the first user's mother is called. When the second user says "Platypus Call Mother", the second user's mother is called. The command "Geronimo call Mother" would not be acted on by this voice controlled device, since Geronimo is not one of its appliance names.

Another aspect of the present invention improves the recognition accuracy of voice controlled devices. The present invention collectively improves recognition accuracy by requiring first a period of relative silence prior to a phrase directed at the voice controlled

device, second the appliance name, and third a valid command. Complete silence is not necessary but a relative silence is needed, where relative silence is defined as a sound level that is quieter than the sound level while the phrase is being spoken. The specific period of relative silence required, and the allowed decibel difference between the relative silence and the sound intensity of the spoken phrase directed at the voice controlled device, will depend on the type of voice controlled device being built, its intended operating environment, the capabilities of the speech recognition system used, and other factors. In some cases, the duration and / or decibel difference of relative silence required may also be varied by the voice controlled device or associated circuits or software, so as to maximize the recognition accuracy obtained in that particular circumstance. In accordance with the standard VUI, each user can assign a voice controlled device a unique name or use a default appliance name. After communicating the appliance name to a voice controlled device, a command must be spoken. Valid input at this point includes special phrases like "Help" or "Cancel", which are part of the standard VUI grammar. If a valid command is not recognized, the voice controlled device rejects the entire sequence and returns to the state where it is waiting for silence. Additionally, depending on the command, one or more

additional phrases, typically representing modifiers to the command, may be provided or required (for example, the phone number in the command sequence "<silence> Call 555-1212"). Valid phrases at this point also include special phrases like "Help" or "Cancel", which are part of the standard VUI grammar. Failure to detect valid phrases after the command within a short period of time can be used as a basis for rejecting the entire command sequence, or for prompting the user to clarify his intentions. Either way, this serves as an additional level of accuracy checking. Alternatively, if a phrase is not detected during the short period of time after the command, the command may be performed anyway.

Voice controlled devices can be identified either by visual identification, or acoustic identification, or both. Acoustic identification is defined as including both audible and non-audible communications with the voice controlled device. Audible and non-audible are defined elsewhere. Visual identification can occur through use of a standard logo or other visual identifier. A blinking LED is another example of a visual identifier. Visual identification is particularly appropriate for voice controlled devices that do not have a speech recognition engine that is always turned on. For example, to minimize battery consumption, battery operated voice controlled devices may require the user to push a switch (or its equivalent, such as

-21-

flipping open a flip-type cellphone) to activate the speech recognition engine. Acoustic identification only works for voice controlled devices that are actively listening for recognizable commands.

Acoustic identification is accomplished by a user saying an identification phrase. An example of an identification phrase is "What is out there?". A voice controlled device may have one or more identification phrases. Any voice controlled device that hears its identification phrase responds to identify its presence.

In accordance with the standard VUI, the response is a random delay of up to 2 seconds of silence, followed by a standard signal (for example, one or more tones or beeps or other sounds), then at least one of the voice controlled device's appliance names, and any applicable basic operation instructions (e.g. "<beep> I am Telephone. You can say Telephone help."). In order to coordinate responses from multiple voice controlled devices in the same communication environment, each voice controlled device must during its silence period listen for another voice controlled device's response, the start of which is marked by the standard signal. Detection of the other voice controlled device's standard signal can be accomplished by any means that is convenient, including by the voice recognition system, by a DSP, by a microprocessor, or by special circuitry. In the event another voice controlled device starts

-22-

responding during this silence period, the listening voice controlled device must restart its silence timing after the responding voice controlled device finishes. In the event two voice controlled devices start responding at approximately the same time [for example, so that they're standard signals overlap in time], they both must back off for a new randomly selected silence delay, but this time the delay must be of up to twice the length of the previous silence delay, but not to exceed 16 seconds.

In order to restrict which voice controlled devices respond to an identification phrase, a user may include a voice controlled device's name in the identification phrase. For example, one could say "Socrates are you out there?" to see if a voice controlled device named Socrates was nearby. Similarly, one could say "Clock are you out there" which would cause all voice controlled devices with an appliance name of Clock (whether a default appliance name or a user appliance name) to respond. A possible variation is that voice controlled devices may respond with some response other than their names, as for example, might be needed for security reasons.

A voice controlled device may use both visual and acoustic identification methods. For example, even though a speech recognition engine is continuously on, it may still display the visual logo and / or other

visual identifier. Similarly, in a voice controlled device that requires manual activation of the speech engine, once enabled, the engine could then be responsive to the command "What is out there?"

In another aspect of the present invention, the initial storage of a user's spoken phrase (for example, when making a new phonebook entry under voice control) is processed by the speaker-independent speech recognition engine of the voice controlled devices. This engine returns a speaker-independent phonetic representation of the phrase. This speaker-independent phonetic representation is what is stored.

When a command is issued by a user, it is also processed by the speaker-independent speech recognition engine of the present invention. This could be the same speaker-independent engine use for storing the original entries, or a completely different speaker-independent engine. In either case, the engine returns a speaker-independent phonetic representation of the command sequence. This speaker-independent phonetic representation can be compared to earlier stored phonetic representations to determine whether the command is recognizable.

By converting both the stored spoken entries and any commands to speaker-independent phonetic representation a number of advantages are provided.

- Recognition will be reliable even if the user's voice

has changed, perhaps due to a sickness, stress, fatigue, transmission over a noisy or distorting phone link, or other factors that might change a human user's or machine user's speech. Text-based information can be stored and then recognized.

- Recognition will be reliable even if some other user had stored the original voice phrase.
- Recognition can be speaker-independent, even for user-stored commands and phrases.
- Stored entries originating from text sources and from different speakers can all be combined and reliably for recognition.
- The use of speaker-independent phonetic representations facilitates upgrading to improved recognition engines as they become available. Improved speech recognition engines can use existing stored information without impacting reliability or requiring re-storage, since all stored entries are held in phonetic form. New information stored using the improved speech recognition engines can be used on equipment with older recognition engines. Old and new generations of equipment can interoperate without prior coordination by using phonetic representations. This allows, for example, two PDAs to exchange voice-stored phonebook entries and provide reliable recognition to the new users of that information.

Finally, there are no legacy restrictions to hold back or restrict future development of speaker-independent recognition engines as long as they can create phonetic representations, unlike waveform-storage based systems, which must always be able to perform exactly the same legacy waveform transformations.

VOICE CONTROLLED DEVICES

Referring now to FIG. 1A, environment 100 is illustrated. Environment 100 may be any communication environment such as an office, a conference room, a hotel room, or any location where voice controlled devices may be located. Within environment 100, there are a number of human users 101A-101H, represented by circles. Also within the environment 100, are voice controlled devices 102A-102H, represented by squares and rectangles, each operationally controlled by the standard voice user interface (VUI) of the present invention. Voice controlled devices 102A-102E, represented by rectangles, are fixed within the environment 100. Voice controlled devices 102F-102H, represented by squares, are mobile voice controlled devices that are associated with human users 101F-101H respectively. Voice controlled devices 102A-102H may be existing or future devices. Voice controlled devices 102A-102E may be commonly associated with a user's automobile, home, office, factory, hotel or other

locations where human users may be found.

Alternatively, if the voice controlled devices 102A-102E are to be controlled by non-audible speech, voice controlled devices may be located anywhere.

In the present invention, the standard VUI allows a user to associate a user-assignable name with these voice controlled devices 102A-102H. The user-assignable name of the voice controlled device may be generic such as telephone, clock, or light. Alternatively, the name may be personalized such as those ordinarily given to humans such as John, Jim, or George. In either case, the voice controlled devices 102A-102H while constantly listening will not respond to commands until it recognizes one of its names (user-assigned or default).

Although any name can be assigned to a voice controlled device, to minimize confusion between the voice controlled device and real people, users may choose to use unusual names such as Aardvark or Socrates, which are unlikely to occur during normal conversation. With reference to Figure 1A, consider the environment 100 to be a conference room where human users 101A-101H are meeting. Further assume that voice controlled device 102A is a telephone having speaker phone capabilities in the conference room 100 and the appliance name is Telephone. The human user such as 101A would first call out the name of the Telephone before desiring to give commands to that voice controlled device. By providing

names to the voice controlled devices, the voice controlled devices can properly respond to given commands and avoid confusion between multiple users and voice controlled devices. The voice controlled device may be a telephone, an organizer, a calculator, a light fixture, a stereo system, a microwave oven, a TV set, a washer, a dryer, a heating system, a cooling system, or practically any system. Voice controlled devices 102A-102H may include an audible communications interface (ACI) in order to listen to commands and data input from human users 101A-101H and audibly notify a user that the command or data was properly interpreted and executed. Voice controlled devices 102A-102H further include a speech recognition and synthesis system (SRS). The speech recognition of the SRS provides for interpreting speech in different dialects independent of which user is speaking, and independent of whether the user is a human or device. While the preferred embodiments of the present invention utilize a speaker independent voice recognition system, the present invention is also compatible with speaker dependent voice recognition systems. The SRS may operate with one or more than one language. The speech synthesis of the SRS provides for generation of speech responses, status commands, or data by the voice controlled devices which may be audibly communicated or non-audibly communicated. Speech synthesis, also referred to herein as speech generation,

-28-

is defined herein to include any method of responding with speech (audible or non-audible), including but not limited to, speech recording, storage and playback systems, pre-recorded vocabulary systems with playback, sophisticated speech synthesis systems generating utterances from a combination of characters, and some combination of the above. Preferably the voice controlled devices contain both a speech recording, storage and playback system and a pre-recorded vocabulary system with playback.

Voice controlled devices 102A-102H may optionally include an communications interface (ECI) for providing remote control of voice controlled device via wireless or wired means using non-audible voice or speech. As illustrated in FIG. 1A, voice controlled device 102A has a connection 105 for connection to a telephone system. In this manner, the voice controlled device 102A may remotely communicate to a user and accept and acknowledge commands. Referring now to FIG. 1B, the human user 101I communicates by telephone 112 over the wired or wireless transmission media 114 over the telephone company switch 116. The telephone company switch 116 is connected by a wire means or wireless means through connection 105 to the voice controlled device 102A. Telephone 112 may be a wireless or wired telephone. In this matter, human user 101I may remotely interface to a voice controlled device 102A within a

communications environment 100. Alternatively, a voice controlled device such as voice controlled device 102E may be remotely controlled over a network by a remote computer 118. In this case, a remote human user 101J can send voice commands or instructions through remote computer 118 which is coupled to the voice controlled device 102E through the network connection 120 and connection 106. The network connection 120 may be a wireless or wired connection, realtime or store-and-forward, through a computer network such as the Internet. There are a wide variety of ways that a remote user can be connected to a voice controlled device, including but not limited to, the use of wired and wireless connections. Wired connections may include, but are not limited to, realtime communications systems such as the telephone system and realtime Internet connections, store-and-forward systems such as email of voice representations and other non-realtime Internet protocols. Wireless systems may include, but are not limited to, radio and infrared systems. Any of these alternatives can include circuit-based systems and packet-based systems, and can include analog and digital systems. Any of these alternatives can be used with or without various modulation and/or encoding and/or encryption schemes.

Referring now to Figure 2, exemplary voice controlled devices 102I-102M are illustrated. The voice

-30-

controlled device 102I is exemplary of white goods such as freezers, refrigerators, washers, dryers, air conditioners, heating units, microwave ovens, ovens, and stoves. Voice controlled device 102J is exemplary of voice controlled devices requiring an optional communication interface (ECI). This may include voice controlled devices for consumer electronics such as television, video cassette recorders, stereos, camcorders, tape recorders, dictation units, alarm clocks, and clock radios as well as telephone products such as standard wired telephones, telephone answering machines, light switches, alarm systems, computing devices, Internet access devices, and servers, etc. Voice controlled device 102K is exemplary of portable or wireless systems such as cellular telephones, walkman style systems, camcorders, and personal digital systems. Voice controlled device 102L is exemplary of automobile voice controlled systems such as car cellular telephone systems, automobile radio systems, car navigation systems, HAV (heating, air conditioning and ventilation) systems, and other control systems for an automobile. Voice controlled device 102M is exemplary of remote controlled devices, such as voicemail systems.

Voice controlled device 102I includes an audible communications interface (ACI) 202, a speech recognition and synthesis system (SRS) 204, and an appliance peripheral and control circuit (APCC) 206. The ACI 202

-31-

is coupled to SRS 204 and SRS 204 is coupled to APCC 206. In the voice controlled device 102I, ACI 202 is its primary means of speech communication.

Voice controlled device 102J includes ACI 202, SRS 204, APCC 206, communications interface (ECI) 207, and connection 208. ACI 202 is coupled to SRS 204. APCC 206 is coupled to SRS 204. ECI 207 couples to SRS 204 and connection 208 couples to the ECI 207. Voice controlled device 102J can alternatively communicate using speech or voice communication signals through ACI 202 or ECI 207. Voice controlled device 102K includes ACI 202, SRS 204, APCC 206, and an antenna 209.

Voice controlled device 102K can communicate using audible speech signals through the ACI 202 or using encoded speech signals through the ECI 207. ECI 207 couples to APCC 206. ECI 207 also couples to Connection 212. Connection 212 could, for example, be an antenna or infrared port. Voice controlled device 102L also includes an ACI 202, SRS 204, APCC 206, and an antenna 209. ACI 202 couples to SRS 204. SRS 204 couples to APCC 206. Antenna 209 couples to APCC 206. Voice controlled device 102L can communicate by means of ACI 202 and APCC 206 through antenna 209.

Voice controlled device 102M includes an APCC 206, SRS 204, an ECI 207, and connection 210. Connection 210 may be a wired or wireless connection, including an antenna. SRS 204 couples to APCC 206 and also to ECI

207. Connection 210 couples to ECI 207. Voice controlled device 102M can communicate via ECI 207 over connection 210.

The APCC 206 represents the elements of the voice controlled device 102 that are to be controlled. For example, in the case of white goods, the items to be controlled may be temperature, a time setting, a power setting, or a cycle depending on the application. In the case of consumer electronics, the APCC 206 may consist of those items normally associated with buttons, switches, or knobs. In the case of telephone products, the APCC 206 may represent the buttons, the dials, the display devices, and the circuitry or radio equipment for making wired or wireless calls. In the case of automobile systems, the APCC 206 may represent instrumentation panels, temperature knobs, navigational systems, the automobile radios channels, volume, and frequency characteristics.

Referring now to FIG. 3, the voice controlled device 102 is illustrated. Voice controlled device 102, illustrated in FIG. 3, is exemplary of the functional blocks within voice controlled devices described herein.

Voice controlled device 102 includes the ACI 202, the APCC 206 and the SRS 204. The voice controlled device 102 may also have an ECI 207 such as ECI 207A or ECI 207B.

The ACI 202 illustrated in FIG. 3 includes

-33-

microphone 303, speaker 304, and amplifiers 305. The SRS 204 as illustrated in FIG. 3 includes the voice communication chip 301, coder/decoder (CODEC) 306 and 308, host microcontroller 310, power supply 314, power on reset circuit 316, quartz crystal oscillator circuit 317, memory 318, and memory 328. The SRS 204 may optionally include an AC power supply connection 315, an optional keypad 311 or an optional display 312. For bidirectional communication of audible speech, such as for local commands, prompts and data, the speech communication path is through the VCC 301, CODEC 306, and the ACI 202. For bidirectional communication of non-audible speech, such as for remote commands, prompts and data, the non-audible speech communication path is through the VCC 301, CODEC 308, ECI 207A or the VCC 301, host microcontroller 310, APCC 206, and ECI 207B. The ECI 207 may provide for a wired or wireless link such as through a telephone network, computer network, internet, radio frequency link, or infrared link.

Voice communication chip 301 provides the voice controlled device 102 with a capability of communication via speech using the standard voice user interface of the present invention. Microphone 303 provides the voice controlled device 102 with the capability of listening for audible speech, such as voice commands and the device's appliance names. Microphone 303 may be a near field or far field microphone depending upon the

application. For example, near field microphones may be preferable in portable cell phones where a user's mouth is close while far field microphones may be preferable in car cell phones where a user's mouth is a distance away. Speaker 303 allows the voice controlled device 102 to respond using speech such as for acknowledging receipt of its name or commands. Amplifiers 305 provides amplification for the voice or speech signals received by the microphone 303. Additionally, the amplifiers 305 allow amplification of representations of voice signals from the CODEC 306 out through the speakers 303 such that a human user 101 can properly interface to the voice controlled device 102.

Microphone 303 and Speaker 304 are each transducers for converting between audible speech and representations of speech. CODEC 306 encodes representations of speech from the ACI 202 into an encoded speech signal for VCC 301. In addition, CODEC 306 decodes an encoded speech signal from the VCC 301 into an representation of speech for audible communication through the ACI 202.

Alternatively, non-audible speech signals may be bi-directionally communicated by the voice controlled device 102. In this case, VCC 301 provides encoded speech signals to CODEC 308 for decoding. CODEC 308 decodes the encoder speech signal and provides it to the ECI 207A for communication over the connection 105.

Speech signals may be received over the connection 105 and provided to the ECI 207A. The ECI 207A couples the speech signals into the CODEC 308 for encoding. CODEC 308 encodes the speech signals into encoded speech signals, which are coupled into the VCC 301.

Speech signals may also be electronically communicated through the APCC 206. Speech signals from the VCC 301 for transmission are passed to the microcontroller 310. Microcontroller 310 couples these into the APCC 206, which transmits the speech signals out to the ECI 207B. Speech signals to be received by the voice controlled device 102 may be received by the ECI 207B and passed to the APCC 206. The APCC 206 then may couple these received speech signals to the microcontroller 310, which passes these onto the VCC 301 for recognition.

The voice controlled device 102 controls the APCC 206 by means of signals from the host microcontroller 310. The host microcontroller 310 is coupled to the APCC 206 to facilitate this control. Voice controlled device 102 may optionally have a keypad 311 coupled to the microcontroller 310 as a further input means. Keypad may be a power button, a push to talk button or a security code input means, in addition to optionally being used to input other information. Voice controlled device 102 may optionally include a display 312 coupled to the host microcontroller 310 in order to visually

display its status or other items of interest to a user.

However, the voice controlled device can function generally without the optional keypad 311 or the optional display 312.

The voice controlled device 102 includes power supply 314. Power supply 314 may generate power from a DC supply source or an AC supply source, or from both. The source of DC supply may be a battery, solar cell, or other DC source. In the case of an AC supply source, the optional AC power cord 315 is provided. VCA 102 includes a power on reset circuit 316 to reset its system when the power supply 314 is turned on.

Quartz crystal oscillator circuit 317 in conjunction with other circuitry within the VCC 301 provides an accurate oscillation input to the VCC 301 for generation of clock signals.

Memory 318 is coupled to VCC 301 and provides rewritable non-volatile and volatile memory as well as a read only memory. These typically are a flash RAM, a static RAM, and a ROM. Memory 318 is used to store programs as well as store pre-recorded and recorded phrases. Additionally, memory 318 provides scratch memory for program operation. As is standard practice in the industry, the types of memories used may vary depending on the specific voice controlled device being constructed. Program storage for the present invention may be permanent, as with a ROM, non-volatile but

changeable, as with a flash, or volatile, as in a RAM, in which case the program could be downloaded from a non-volatile memory, or from a remote source.

Memory 328 may be volatile memory, non-volatile memory, or a mixture. If only volatile memory is used, its contents can be downloaded from another location for initialization. The size and capabilities of Memory 328 will depend on the type of voice controlled device being built. Alternatively, memory may be substituted in some cases for a type of magnetic, optical or other type of storage medium.

In the voice controlled device 102, VCC 301 may additionally include the functionality of the host microcontroller 310 such that only one processing unit is contained within the voice controlled device 102. Similarly, the APCC 206, codecs 306 and / or 308, ECI 207A, ECI 207B, memory 318, memory 328, amplifiers 305, or other elements maybe integrated into VCC 301, as is customary in the industry as ever-increasing levels of integration are achieved.

Referring now to FIG. 4, a block diagram of the voice communication chip (VCC) 301 is illustrated. The voice communication chip 301 is an integrated circuit and includes the processing units 402, memory units 403, a Bus and Memory Controller (BMC) 404, a bus adapter 405, and Peripherals 406. The voice communication chip 301 is further described in the microfiche appendix

entitled "ISD-SR 300, Embedded Speech Recognition Processor" by Information Storage Devices, Inc. The processing units 402 includes a microprocessor and a digital signal processing module (DSPM). The memory units 403 include a DSPM random access memory (RAM) 407, a system RAM 408, and a read only memory (ROM) 409. The peripherals 406 include I/O ports 420, an Interrupt Control Unit (ICU) 422, a coder/de-coder (CODEC) interface 424, a Pulse Width Modulator (PWM) 426, a MICROWIRE interface 428, Master MICROWIRE controller 430, a reset and configuration controller 432, a clock generator 434 and a WATCHDOG timer 436. In order to communicate effectively, the voice communication chip 301 includes a core bus 415 and a peripheral bus interconnecting the components as shown in FIG. 4.

The microprocessor 416 is a general purpose 16-bit microprocessor core with a RISC architecture. The microprocessor 416 is responsible for integer arithmetic logic and program control. The DSP Module (DSPM) 418 performs DSP arithmetic. ROM 409 and system RAM 408 are used for the storage of programs and data. DSPM RAM 407 can be accessed directly by the DSPM 418. When the DSPM 418 is idle, the microprocessor 416 can access the DSPM RAM 407.

The Bus and Memory Controller (BMC) 404 controls access to off-chip devices, such as DRAM, Expansion Memory, off-chip Base Memory and I/O Expansion. The I/O

ports 420 provide the interface to devices coupled to the voice communication chip 301. The I/O ports 420 represents twenty-six I/O pins of the voice communication chip 301. Using the internal ROM 409 for program memory without expansion options, sixteen I/O pins can be individually configured for input or output, eight I/O pins dedicated for output only and two I/O pins dedicated for input only. The ICU 422 provides the capability of processing five maskable interrupts (four internal and one external) and three internal Non-Maskable Interrupts (NMIs). The CODEC interface 424 provides a direct interface to one CODEC device 306 in the case of ACI 202 only or two CODEC devices 306 and 308 in the case of ACI 202 and ECI 207A. The Pulse Width Modulator (PWM) 426 generates a square wave with a fixed frequency and a variable duty cycle. The MICROWIRE interface 428 allows serial communication with the host microcontroller 310. The Master MICROWIRE controller 430 allows interface to serial flash memory and other peripherals. The reset and configuration block 432 controls definition of the environment of the voice communication chip 301 during reset and handles software controlled configurations. Some of the functions within the voice communication chip 301 are mutually exclusive. Selection among the alternatives is made upon reset or via a Module Configuration register. The clock generator 434

interfaces to the quartz crystal oscillator circuit 317 to provide clocks for the various blocks of the voice communication chip including a real-time timer. The clock generator can also be used to reduce power consumption by setting the voice communication chip 301 into a powerdown mode and returning it into normal operation mode when necessary. When the voice communication chip 301 is in power-down mode, some of its functions are disabled and contents of some registers are altered. The watchdog timer 436 generates a non-maskable interrupt whenever software loses control of the processing units 402 and at the expiration of a time period when the voice communication chip 301 is in a power-down mode.

STANDARD VOICE USER INTERFACE

Similar to computer operating systems providing a GUI, the standard voice user interface (VUI) can be thought as being provided by a standard VUI operating system code. The standard VUI operating across a wide array of voice controlled devices allows a user to interface any one of the voice controlled devices including those a user has never previously interacted with. Once a user is familiar with the standard VUI, they can walk up to and immediately start using any voice controlled device operating with the standard VUI. The standard VUI operating system code has specific

-41-

standardized commands and procedures in which to operate a voice controlled device. These standardized commands and procedures are universal to machines executing the standard VUI operating system code. Voice controlled application software, operating with the standard VUI operating system code, can be written to customize voice controlled devices to specific applications. The voice controlled application software has voice commands specific to the application to which the voice controlled device is used. A particular voice controlled device may also have additional special features that extend the core capabilities of the standard VUI.

Some of the standard VUI functionality in the core VUI include a way to discover the presence of voice controlled devices, a core common set of commands for all voice controlled devices, a way to learn what commands (both core commands and appliance-specific commands) the voice controlled device will respond to, a vocalized help system to assist a user without the use of a manual or display, a way to personalize the voice controlled device to a user with user assignable settings, security mechanisms to control use of voice controlled devices to authorized users and protect user assignable settings and information from other users, and standard ways for a user to interact with voice controlled devices for common operations (e.g. selecting

yes or no, listing and selecting items from a list of options, handling errors gracefully, etc.).

The standard VUI includes an API (Applications Programming Interface) to allow software developers to write custom voice controlled applications that interface and operate with the standard VUI and extend the voice controlled command set.

Referring now to FIG. 5, a block diagram illustrates the Software 500 for controlling Voice Controlled Device 102 and which provides the standard VUI and other functionality. The Software 500 includes Application Code 510, a VUI software module 512 and a Vocabulary 524. Application code 510 may be further modified to support more than one application, representing multiple application code modules, to provide for further customization of a voice controlled device 102. The Vocabulary 524 contains the phrases to be detected. The phrases within the Vocabulary are divided into groups called Topics, of which there may be one or more. In Figure 5, the Vocabulary 524 consists of two Topics, Topic 551 and Topic 552.

Typically, Application Code 510 interfaces to the VUI software 512 through the Application Programming Interface (API) 507. The VUI software 512 provides special services to the Application Code 510 related to voice interface, including recognition and prompting. The interrelationship between the VUI software 512 and

the application code 510 is analogous to that between Microsoft's MS Windows and Microsoft Word. Microsoft Windows provides special services to Microsoft Word related to displaying items on a screen and receiving mouse and keyboard inputs.

Generally, the Application Code 510 may be stored in host memory and executed by the host microcontroller 310. However, the functionality of the host microcontroller 310 can be embedded into the VCC 301 such that only one device or processor and one memory or storage device is needed to execute the code associated with the software 500.

All phrases that can be recognized, including those phrases for the core and application specific commands, are included in the Vocabulary 524. The VUI software module 512 can directly access the vocabulary phrases, for example for use during recognition. The VUI software module 512 can also processes tokens. Tokens abstractly relate to the phrases within the Topics 551-552. Tokens are integer numbers. For example, the phrase for 'dial' might have a token value of '5', and the phrase for 'hangup' might have a token value of '6'. There is a token value assigned to every phrase that can be recognized. Because the VUI software module 512 can process tokens related to the vocabulary file 524, it can refer to phrases without having to directly access them. This makes it possible to change languages (from

-44-

English to French, etc.) without modifying the VUI software module 502. Thus, the standard VUI will function using different dialects or languages simply by modifying the vocabulary file 524.

Core capabilities of the standard VIU operating in a voice controlled device allow a user to: name the voice controlled device, identify the presence of voice controlled devices, activate a user's previously stored personalized preferences, recover from misrecognitions by canceling an operation, use a Help function to identify the commands and options that can be used with the voice controlled device, use a standard core set of commands and use other additional commands, confident that they follow a standard syntax. (Although the syntax of commands is common, the specific list of commands on any voice controlled device will depend on the nature of the voice controlled device). The standard VUI also includes standard functions for the following user interactions for the API: GETYESNO - Accepting a Yes / No response from the user; GETRESPONSE - Accepting an arbitrary input from the user; GETRESPONSEPLUS - Accepting an arbitrary input from the user, with enhanced error recovery features; LISTANDSELECT - Providing the user with a list of choices, and allowing the user to select one; and ACOUSTICADDWORD - Adding a phrase that can thereafter be recognized.

In order to properly function with the standard

-45-

VUI, the SRS 204 of the voice controlled device 102 can provide continuous recognition of speech and digits when powered up. However, pauses exceeding certain durations may be recognized by the SRS 204 as marking the end of a command or providing an indication that an incomplete command sequence has been received.

NAMES

A key element of the standard VUI of the present invention is that each voice controlled device has one or more appliance names, each of which is a phrase. The initial appliance name is a default name for a voice controlled device programmed by the manufacturer at the factory. However, users can generally assign a user-assigned appliance name of their choosing to a voice controlled device. Naming a voice controlled device is different from other kinds of naming, such as naming people. A person has a single (first) name that can be used by everyone who wants to talk with them. In contrast, with naming of voice controlled devices, every user of a voice controlled device usually gives the voice controlled device a different, unique name. Accordingly, a voice controlled device may have as many names as it has users.

When a user addresses a voice controlled device by name two things happen. First, when the voice controlled device recognizes one of its names, the voice

controlled device is notified that it is being addressed and will need to listen for a command. Second, since each user usually employs a different name for a voice controlled device, it is informed of a user's identity (speaker identification). If a user has stored preferences related to the functionality of the voice controlled device, the voice controlled device can personalize itself to the preferences of that user.

To illustrate this naming concept, consider the following example of a desktop telephone, the voice controlled device, having two users. User 1 has named the phone "Aardvark" and user 2 named the phone "Platypus". If the phone hears "Aardvark Call Mom", the phone will recognize that it is being addressed by user 1 and it should use User 1's phonebook. Accordingly, it will dial the number for "Mom" programmed by User 1. Similarly, if the phone hears "Platypus Call Mom", it will recognize that user 2 is addressing it, and it will dial the number for "Mom" programmed by user 2.

In order to minimize false recognition, it is preferable that users assign names to the voice controlled devices that are generally not spoken during normal speech. Choosing unusual names helps ensure that two voice controlled devices within audible range of each other don't have identical names (perhaps assigned by different users). A maximum time limit for saying the phrase name may be required in some cases due to memory

limitations in the voice controlled device.

Referring now to FIGs. 6A-6E, flow charts of the detailed operation of the standard VUI with voice controlled devices 102 are described. In the flow charts of FIGs. 6A-6E, a solid box shows a phrase communicated by a user (placed in quotes) or a user action (no quotes). A dotted box shows a phrase communicated by the voice controlled device (in quotes) or an action taken (no quotes). In the case where there is a solid box directly below a dotted box, a path exiting from the right of a dotted box is taken if the action within the current dotted box is completed normally and the path to the solid box below a dotted box is taken if an unusual event occurs. Generally, the solid box directly below the dotted box indicates the unusual event.

STANDARD VUI COMMAND SYNTAX

Referring now to FIG. 6A, the general syntax for all voice commands is:

<silence><name> <command> <modifiers & variables>.

The <silence> is a period of relative silence during which the user is not speaking although background noise and background speech may still be present. The <name> is the appliance name associated with a voice controlled

device 102. The <command> is an operation that a user wants performed. The <modifiers & variables> consist of additional information needed by some commands. The SRS 204 recognizes the elements in their syntax in order for a user to control voice controlled devices.

Most voice controlled devices will continuously listen for the voice command sequence. When a voice controlled device hears its <name>, it knows that the following <command> is intended for it. Since each user has a different <name> for a voice controlled device, the <name> also uniquely identifies the user, allowing the voice controlled device to select that user's personalization settings. Commands include core VUI commands included with all voice controlled devices, and commands specific to a given application, all of which are stored within the vocabulary 524.

Requiring <silence> before detection of <name> helps prevent false detection of <name> during normal conversational speech (i.e. during periods when the user is speaking conversationally to other users and not to the voice controlled device). In all cases, the duration of <silence> can be configured by the manufacturer and can range from 0 (no <silence> required) to a second or more. Typically it will be about a quarter of a second.

Examples of voice command sequences that might be used with a voice controlled device such as a telephone

named Aardvark include "Aardvark Call The Office", "Aardvark Dial 1-800-55-1212", and "Aardvark Hang-up". (In the command examples and descriptions provided, for the sake of brevity the <silence> is often not shown, and even where it is shown or described, the option always exists of a manufacturer choosing to use a silence duration of zero.)

There are two special cases where the command syntax is permitted to differ from the general syntax. The first special case is in voice controlled devices that do not continuously listen for <silence><name>. For example, in some battery operated applications, power consumption limitations may require the VCC 301 in the voice controlled device 102 to be powered down during idle periods. Another example is a voice controlled device located where false recognition of a name would have undesirable results, for example, a desktop phone in a conference room during a presentation. A third example is voice controlled devices where there is a high risk of false recognition, for example, where multiple conversations can be heard.

For these types of situations, an alternate command syntax is used in conjunction with a button or switch of some type. The first alternate command syntax is:

<activation of a switch> <silence (optional)>
<name> <command> <modifiers & variables>.

In this syntax, the <activation of a switch> means the user presses a button or performs some other mechanical act (e.g. opening a flip-style cell phone) to activate the recognition capability.

A second special case is where the user normally enters a series of commands in quick succession. For these cases, the user can identify themselves once to the voice controlled device using a password protection method, or by issuing a command that includes the voice controlled device's appliances <name>, and thereafter continue entering commands. The second alternate command syntax (in this example, for three successive commands) is:

<silence> <name> <command> <modifiers & variables as needed>

<silence> <name (optional)> <command> <modifiers & variables as needed>

<silence> <name (optional)> <command> <modifiers & variables as needed>

With this syntax, the user can issue a series of commands without having to constantly repeat the voice controlled device's appliances <name>. However, the user is permitted to say the <name> at the start of a command. Note that in this syntax, the <silence> is required to properly recognize the spoken <name> or <command>.

When either of the first or second alternate

syntaxes is used, it is desirable to ensure that if a new user starts working with the voice controlled device, they are properly identified. This can be ensured by explicitly requiring the <name> after a period of inactivity or after power-up of the voice controlled device or other similar protocol.

STANDARD CORE VUI COMMANDS

There are a number of standard core commands included in the vocabulary 524 of voice controlled devices 102 operating using the standard VUI. FIGs. 6A-8 illustrate the syntax of the following commands.

Referring to FIG. 6A, at start 600, the appliance name, <name>, of a voice controlled device is usually spoken prior to a command. Any of the voice controlled device's appliances names can be spoken whenever the voice controlled device is listening for a command. If the <name> is not followed by a command within some period of time, the voice controlled device will go back to return to start 600 in its original idle state. This is indicated by the solid box Silence of N seconds. N in this case is a programmable value usually application dependent and assigned by the voice controlled device manufacturer. After supplying the appliance name, a user is granted access to further commands of the standard VUI operating on the voice controlled device at 601.

The syntax of the Help command is:

<name> Help <command (optional)>

or

Help <command (optional)>

The help command can be invoked at any time, including when any other command can be given, or whenever the voice controlled device is waiting for a response. If the Help command is issued while the voice controlled device is waiting for a valid command, Help must be preceded with <name> if the voice controlled device requires a <name> before other commands. If the Help command is requested while the voice controlled device is waiting for any other type of response, <name> does not need to precede the Help command. In all cases where <name> is not required before Help, if the user says "<name> Help", the use of <name> does not generate an error.

The help function is context sensitive - whenever Help is requested, the voice controlled device responds with a description of the available options, given the current context of the voice controlled device. If Help is requested when the voice controlled device is listening for a command, the voice controlled device will respond with its state and the list the commands that it can respond to (e.g. "At Main menu. You can say") Further detail on any specific command can be obtained with the "Help <command>" syntax (e.g. "Help

Dial", "Help Call", and even "Help Help"). If "Help" is requested while the voice controlled device is waiting for some type of non-command response (e.g. "Say the name"), then the voice controlled device will respond with a statement of the voice controlled device's current status, followed by a description of what it is waiting for (e.g. "Waiting for user response. Say the name of the person whose phonebook entry you wish to create, or say Nevermind to cancel.").

The syntax of the cancellation command is:

<name (optional)> Nevermind

or

<name (optional)> Cancel

The Nevermind or Cancel command can be issued whenever the voice controlled device is executing a command and waiting for a response from the user. Nevermind or Cancel causes the voice controlled device to cancel the current command and respond with a statement that the operation has been cancelled (e.g. "Cancelled."). If Nevermind or Cancel is issued while the voice controlled device is waiting for a command, it can be ignored. The use of <name> with Nevermind or Cancel is optional - it works identically whether or not <name> is spoken.

The syntax of the return to main menu command is:

<name> Main Menu

For voice controlled devices that have submenus of commands, <name> Main Menu returns the user to the main

menu and causes a response of "At Main menu." or the like. This command provides an easy way for the user to return to a known point from any submenu. The Main Menu command does not have to be recognized in voice controlled devices that only have one menu, but is a mandatory command for voice controlled devices with submenus.

Changing Voice Controlled Device Names

In some cases it may be desirable to change the user-assigned name of a voice controlled device. Referring now to FIGs. 6A-6B, the syntax of the Change Name command is:

<old name> Change Your Name

This command allows a user to name or rename a voice controlled device. When a voice controlled device is new, it has at least one default factory programmed appliance name (e.g. Telephone). Most voice controlled devices have the capability of supporting one or more user-assignable appliance names. A user can name the appliance name by saying "<factory programmed name> Change your name" (e.g. "Telephone change your name").

The voice controlled device will then ask for the new name to be repeated and then change its name. This process can be repeated once for each user-assignable name. For example, consider a 4-user telephone that can be assigned four user-assignable appliance names. A

-55-

user may execute the four name changes with the commands: "Telephone change your name " followed by the dialog to set the name for user 1 to (for example) Aardvark. "Telephone change your name " followed by the dialog to set the name for user 2 to (for example) Barracuda. "Telephone change your name" followed by the dialog to set the name for user 3 to (for example) Coyote. "Telephone change your name " followed by the dialog to set the name for user 4 to (for example) Doggone. If the user attempted to change a fifth user-assignable name in sequence with the command ("Telephone change your name "), it would result in an error message because all available user-assignable appliance names were assigned. Note that the voice controlled device always responds to the factory programmed name, even if all user-assigned names are defined. Accordingly, in this example of a fifth attempt, the voice controlled device still recognizes the "Telephone" factory programmed name - it is just unable to assign a fifth new user-assignable appliance name.

An existing user-assignable appliance name can also be changed with the "Change Your Name " command. Continuing the above example, "Aardvark change your name " would alter the appliance's name for the first user (for example, it could be changed to Platypus), and leave the other three user names unchanged. Similarly, "Platypus change your name " followed by a dialog to set

the name to "Telephone" would reset the first user name to the factory-programmed default.

Identification of Voice Controlled Devices

As voice controlled devices proliferate, it is important that users be capable of readily identifying what, if any, voice controlled devices are present when they enter a new environment. For example, a user walks into a hotel room that has a number of devices. In order to use them a user needs to know which devices are voice controlled devices. Additionally a user needs to know the appliance names in order to properly control them. Beside being audibly identified, voice controlled devices can be identified visually as well as by using a logo signifying a voice controlled device utilizing the standard VUI.

Acoustic identification works when voice controlled devices are actively listening for recognizable commands. In most cases, this means the voice controlled device is constantly listening and attempting recognition. Typically, these voice controlled devices will be AC powered, since the power drain from continuous recognition will be unacceptable for most battery operated voice controlled devices. Referring to FIG. 6A and 6C, the acoustic identification is accomplished by a user communicating an identification phrase to command the voice controlled device. The

identification phrase "What Is Out There?" or some other suitable identification phrase may be used for causing the voice controlled devices to identify themselves.

The syntax of the standard VUI Identification phrase is:

<silence> What Is Out There?

In response to this query, any voice controlled device that hears the question must respond. The typical voice controlled devices response is a random delay of up to 2 seconds of relative silence, followed by a beep (the standard signal) , and the response "You can call me <name>", where <name> is the factory-programmed name that can be used to address the voice controlled device.

In the telephony voice controlled device example described above, a response might be "<beep> You can call me Telephone."

Referring to FIG. 6C, during the random delay of up to 2 seconds, each responding voice controlled device listens for another voice controlled device's response (specifically, for another voice controlled device's beep). In the event another voice controlled device starts responding (as evidenced by a beep) during this silence period, the listening voice controlled device must restart its silence timing after the responding voice controlled device finishes. In the event two voice controlled devices start responding at the same time (overlapping beeps), they both must back off for a

new randomly selected silence delay. However, this time the random delay may be greater than the first, up to twice the length of the previous silence delay. In any event, the delay should not exceed 16 seconds. Additional back off periods for further conflict resolution is provided if other voice controlled devices respond.

Referring to FIG. 6A, the syntax of the Request User-Assignable Names command is:

<name> Tell Me Your Name

or

<name> Tell Me Your Names

If security permits, any user-programmed <name> or the default <name> can be used. The Request User-Assignable Names command is used to ask a voice controlled device to list all the user-programmed <names> that it will respond to. If security permits, the voice controlled device communicates each use-programmed name in a list fashion. Between each user-assigned name it pauses for a moment. During this pause a user may communicate a command to the voice controlled device and it will be executed as if given with that user-programmed <name>. For example consider the telephony voice controlled device example above. The command "Telephone Tell Me Your Name" provided after a pause will cause the telephone to respond by saying "I have been named Aardvark, (pause) Barracuda (pause), Coyote (pause), and

Doggone (pause)." During the pause that followed the voice controlled device saying "Coyote", a user may say "Call Mom", in which case the phone calls user Coyote's Mom (assuming that a phone number for Mom had been previously stored by user Coyote).

SECURITY CONSIDERATIONS

The command for Requesting User Assignable names raised the issue of security in the voice controlled devices. In some cases it is necessary to limit access to a voice controlled device to authorized users. Various methods of security protection can be employed in a voice controlled device which are supported by the standard VUI.

The simplest and least secure security protection is provided through the VUI's naming capability. In this case every user is required to choose a unique name for a voice controlled device. The user assigned appliance names are kept confidential within the voice controlled device and only changed or deleted by a user.

In this manner the appliance name can be used to provide basic security. However, there are many shortcomings with this approach. First, the user must typically repeat the name before issuing each command, which makes it easy for someone to overhear the name, resulting in a loss of security. Second, most voice controlled devices will include a capability for

deleting or changing a user's name for the device. It is preferable to make deletions and changes easy to perform. Additionally changes may need to be performed by someone other than that particular user. For example, the user may have forgotten the name he originally assigned to the voice controlled device, or the user may have stopped using the device and not be available to delete his settings. In the case of using the appliance name as security, there is an inherent conflict between the need for ease of use in changing a name and the quality of security.

A greater level of security can be achieved by requiring the user to say a secret numeric sequence, password or phrase in order to gain access to the voice controlled device. The login might be required when the user starts using the voice controlled device after some period of inactivity, or based on some other criteria. A disadvantage of this approach is that the spoken numeric sequence or phrase might be overheard. Another security alternative is to require the user to enter the numeric sequence, password, or phrase on a keypad such as optional keypad 311. Although this introduces additional hardware, it eliminates the risk of a secret code being overheard by another. A variety of other security options are also possible, including use of a physical key or a security card (e.g. magnetic stripe or smartcard).

Additional security is provided by automatic cancellation or termination of user access to the voice controlled device. In some cases access may be automatically cancelled after every command execution. In other cases automatic cancellation of access may occur following some period of inactivity, power-down or reset, completion of some operation (e.g. in a phone, at the end of a call), or upon the specific request of a user by use of a "Cancel Access" command.

APPLICATION-SPECIFIC COMMANDS

The standard VUI provides each voice controlled device with a number of application specific commands. The application specific commands provided by the standard VUI are associated with telephone and answering machine applications. Additional application specific commands can be programmed for and included in the vocabulary by a manufacturer.

General guidelines for developing commands for the standard VUI are as follows. Sub-menus should be limited in number and organized around logical groups of commands. For example, a telephone TAD might have a main menu that included telephony functions, a submenu for phonebook management, and another submenu for TAD functions.

The number of commands in any menu or submenu should generally be limited to ten or less to minimize

complexity. The help function should clearly describe the available commands.

Complex commands should be broken down into manageably small units. Command phrases should be selected that ensure high recognition success. The standard VUI commands have been selected to ensure high recognition accuracy. Care should be exercised when creating of a custom vocabulary to avoid using confusable phrases.

For destructive events (delete, etc.), user-confirmation of the correct entry and verification of the operation should be requested.

TELEPHONY VOCABULARY

Referring now to FIGs. 6D-6E, 7, and 8, flow charts for the telephony vocabulary for the standard VUI are illustrated. The telephony vocabulary is particularly for telephony voice controlled devices such as desktop telephones, cellular telephones, cellular telephone car kits, and cordless phones. The SRS 204 of the present invention is capable of recognizing the commands in the telephony vocabulary and converting them into recognized tokens for control of the telephony voice controlled devices. The telephony vocabulary includes all the standard VUI Core Commands and the following application specific commands.

The syntax of the Call command is:

-63-

<name> Call <voicetag>

or

<name> Call <digits>

The Call command is used to dial a specific phone number, expressed either as a series of digits or as a phonebook voicetag. The <digits> can be any list of numeric digits. The telephony voice controlled device allows for the synonyms "oh" for zero, and "hundred" for zero-zero to be enabled. The sequence of <digits> can contain embedded pauses. However, if a pause exceeds a programmable duration, the sequence is terminated and the command executed after recognition of a pause that exceeds a duration set by the system designer. The telephony voice controlled device response to a Call command should be "Calling <digits>" or "Calling <voicetag>" with the recognized digits or recognized voicetag voiced to verify accurate recognition. The "Cancel" command can be used to cancel the calling operation in the event of misrecognition.

The syntax of the Dial command is:

<name> Dial <voicetag>

or

<name> Dial <digits>

The Dial command is the same as the Call command.

The syntax of the Answer command is:

<name> Answer

This command is used to answer an incoming call. The

response prompt is "Go ahead".

The syntax of the Hangup command is:

<name> Hangup

This command is used to hangup an active call. The response prompt is a high-pitched beep.

The syntax of the Redial command is:

<name> Redial

This command is used to redial a number. The response is "Redialing <digits>" or "Redialing <voicetag>", depending on whether the previous Call or Dial command was to <digits> or a <voicetag>. If there was no earlier call made, the response is "Nothing to redial".

The syntax of the Store command is:

<name> Store

The Store command is in the phonebook submenu and is used to add a new voicetag.

The syntax of the Delete command is:

<name> Delete

The Delete command is in the phonebook submenu and is used to delete a voicetag.

The syntax of the Mute command is:

<name> Mute

This command mutes the microphone. The response by the voice controlled device is "Muted".

The syntax of the Online command is:

-65-

<name> Online

This command unmutes the microphone. The response is "Online".

Prompts can be communicated by the voice controlled devices to request a response from the user. Prompts may be communicated (i.e. prompting) by a speech synthesizer, playback of pre-recorded speech or other means. The prompts in the telephone vocabulary include the following context-sensitive help prompts:

"Calling <digits> <voicetag>"	"Please say the name you want to call"	"Online"
"Dialing <digits> <voicetag>"	"Please start over"	"one"
"Go ahead"	"My name is now <name>"	"two"
"Goodbye" (for the hangup command)	"Redialing <digits> <voicetag>"	"three"
"Cancelled"	"Sorry, I didn't understand"	"four"
"Please say the name you want to delete"	"Please say the name again"	"five"
"Are you sure you want to delete <voicetag>?"	"Name change canceled"	"six"
"<voicetag>deleted"	"The names did not match"	"seven"
"Please say the new	"Please repeat the	"eight"

-66-

name"	number"	
"Please repeat the new name"	"The number for <voicetag> is <digits>. Is this correct?"	"nine"
"Please say the number for <voicetag>"	"The number for <voicetag> has been stored"	"zero"
"That name is not in the phone book"	"Do you want to store it now?"	"hundred"
	"Muted"	"Nothing to redial"
		"Star"
		"Flash"
		"Pound"

In addition to these prompts, the voice controlled devices can generate a number of different tones or beeps. These include a medium pitch beep (e.g. 200 millisecond, 500 Hz. sine wave), a low pitched beep (e.g. a buzzer sound or 250 millisecond, low frequency beep signifying erroneous entry) and a high pitched beep (e.g. 200 milliseconds, 1200 Hz. sine wave). Other sounds are possible and would be within the intended scope of the present invention.

Vocabulary For Telephone Answering Voice Controlled

-67-

Device

In addition to the forgoing, application specific commands for the standard VUI enable a user to interface to a telephone answering voice controlled device using voice commands. A user can manage message functions and obtain remote access from a telephone answering voice controlled device without using a keypad. The following lists the additional voice commands to be included in the vocabulary 224 for telephone answering voice controlled device.

<name> Play new	<name> Rewind <n>	<name> Stop
<name> Play all	<name> Record Greeting	<name> Play Greeting
<name> Delete this	<name> Record message	<name> Room monitor
<name> Delete all messages	<name> Answer On	<name> Password <password phrase>
<name> Forward <n>	<name> Answer Off	

Automobile Control Vocabulary

Additional specific commands for the standard VUI enable a user to interface to automobile accessories using voice control. Two primary areas for automotive voice control include the control of interior accessories and control of entertainment systems. Automotive accessories include environmental controls, windows, door locks, and interior lights. It is preferable that "Mission critical" elements in an

-68-

automobile, such as steering, braking, acceleration, and exterior lights not be controlled by voice due to potential safety concerns if misrecognition occurs. Entertainment controls are used primarily for a DC player/changer and for the radio.

The automobile control vocabulary 224 for voice controlled devices includes Air conditioning, Fan speed, Temperature, Driver window, Passenger window, Left rear window, Right rear window, Windows, Door locks, Wipers, Low, Medium, High, Increase, Decrease, Set, Reset, Cancel, Clear, Recall, On, Off, Colder, and Warmer.

STANDARD USER INTERFACE FUNCTIONS FOR THE API

The standard VUI of the present invention includes standard functions for user interactions, which are accessed by an applications programming interface (API).

These standard functions for the API include GETYESNO, GETRESPONSE, GETRESPONSEPLUS, and LISTANDSELECT which are used by custom software developers to develop applications that operate on top of the standard VUI of the present invention. FIGs. 9A-9B, 10A-10C, 11, and 12 are flow charts illustrating the functionality of these standard user interface functions within the standard VUI. Briefly, the GETYESNO function is for prompting and accepting a positive (Yes) or negative (No) response from a user. The GETRESPONSE function is for prompting and accepting an input from a user that corresponds to

an expected list of responses. The GETRESPONSEPLUS function is for prompting and accepting input from a user similar to the GETRESPONSE function but includes enhanced error recovery features. The LISTANDSELECT function provides a user with a list of choices and allows the user to select one. The operation of the GETYESNO, GETRESPONSE, GETRESPONSEPLUS, and LISTANDSELECT are adapted from "Debouncing the Speech Button: A Sliding Capture Window Device for Synchronizing Turn-Taking" by Bruce E. Balentine et al, International Journal of Speech Technology, 1997. FIG. 9A illustrates the use of a Yes/No menu and FIG. 9B illustrates how to resolve a rejection or a bad recognition. FIG. 10A illustrates the initiation or begin window for the GETRESPONSE and GETRESPONSEPLUS functions. FIG. 10B illustrates the speech startup or open window functionality for the GETRESPONSE and GETRESPONSEPLUS functions. FIG. 10C illustrates the end recognition or close window functionality for the GETRESPONSE and GETRESPONSEPLUS functions. FIG. 11 illustrates the dual capture window functionality for the GETRESPONSEPLUS function. FIG. 12 illustrates the menu list functionality for the LISTANDSELECT function.

Referring to FIGs. 9A-9B, the GETYESNO user interface function is used to ask the user a question and to accept a positive or negative response such as "Yes" or "No" (or the equivalent phrases in other

-70-

languages). The parameters associated with the GETYESNO are the QUESTION and a TIMEOUT period. The question parameter is a voice prompt to the user which asks a question that can be answered positively or negatively such as "yes" or "no". The Timeout parameter is the number of seconds to wait for a response before flagging that a response was not detected. The voice controlled device returns a byte value depending upon the response or outcome. A 0 is returned if "No" response is detected. A 1 is returned if a "Yes" response was detected. A 17 is returned if a response was not detected in the allowed time indicating a Timeout error. An 18 is returned if a response was detected, but it was not recognizable indicating an out-of-vocabulary-word error.

Referring to FIGs. 10A-10C, GETRESPONSE user interface function plays a Prompt to a user that solicits a response and waits for the response. GETRESPONSE looks for a spoken response that matches a topic within a list known as TopicList. GETRESPONSE either returns an array of recognized tokens, or an error indicator. The parameters associated with the GETRESPONSE are Prompt, Timeout, STS_Sound, and TopicList. The Prompt parameter is the initial prompt to be played to the user. The Timeout parameter is the number of milliseconds to wait for a response before

-71-

flagging that a response was not detected. The STS_Sound parameter (Spoke-Too-Soon Sound) is the sound or prompt to be played if a user speaks before the Prompt finishes playing. Typically, the STS_Sound will be a short tone or beep sound rather than a spoken phrase. The parameter TopicList is the vocabulary subset for the list of topics which the SRS 204 should use to identify the spoken response. The voice controlled device returns a pointer to an integer array.

If the recognition of a response associated with the TopicList was successful, the first element in the array is the number of tokens returned and the following elements in the array are the tokens for each identified speech element (one or more words). Element 1 is the Number of tokens returned. Elements 2 through n+1 are the Token values for each speech element recognized. For example, consider the phrase "Telephone Dial Office". If the token value for the speech element "Telephone" is 7, for the speech element "Dial" is 12, and for the speech element "Office" is 103, then if they are all recognized successfully, the complete array returned would be four elements long with the values 3, 7, 12, 103. If the recognition of the response was not successful, the array is two elements long. The first element is set to zero and the second element indicates the type of error that occurred. In this case, Element 1 is set to 0 indicating that an error was detected.

-72-

Element 2 is set to 17 indicating that a response was not detected in the allowed time (Timeout error) or 18 indicating that a response was detected, but it was not recognizable (out-of-vocabulary-word error). The array returned for a timeout error is two elements long with values 0, 17 and the array returned for an out-of-vocabulary-word error is two elements long with values 0, 18.

Referring to FIG. 11, GETRESPONSEPLUS user interface function plays a Prompt to a user that solicits a response and waits for the response. GETRESPONSEPLUS is similar to GETRESPONSE in that it plays a Prompt for the user and then waits for a spoken response. However, GETRESPONSEPLUS includes the capability to play prompts to recover from error situations where the user has not spoken or has excessive noise in the background. GETRESPONSEPLUS listens for a spoken response that matches the topics in TopicList. GETRESPONSEPLUS either returns an array of recognized tokens, or an error indicator. The parameters for GETRESPONSEPLUS are Initial_Prompt, Timeout, STS_Sound, TopicList, MaxTries, Intervene_Prompt, Repeat_Prompt, and the Help_Prompt. The Initial_Prompt parameter is the initial prompt to be played to a user to solicit a response. The Timeout parameter is the number of milliseconds to wait for a response before flagging that a response was not

-73-

detected. The STS_Sound prompt is a sound or prompt to be played if user speaks before Prompt finishes playing. Typically, STS_Sound prompt will be a short tone or beep sound rather than a spoken phrase. The parameter TopicList is the vocabulary subset for the list of topics which the SRS 204 should use to identify the spoken response. The MaxTries parameter is the maximum number of times GETRESPONSEPLUS will re-prompt the user in an effort to get a good recognition. If recognition does not occur after MaxTries, GETRESPONSEPLUS will return and indicate an error. The Intervene_Prompt parameter is a prompt played to ask the user to repeat himself (e.g. "There was too much noise. Please repeat what you said."). This prompt is played when there was too much noise during the previous recognition attempt.

The Repeat_Prompt parameter is the prompt played to ask the user to repeat what was just said (e.g. "Please repeat what you said"). This prompt is used when a spoke-too-soon error occurred. The Help_Prompt parameter is the prompt played when the user seems to need further instructions, including when the user says nothing. The voice controlled device returns a pointer to an integer array upon completion of the user interface function. If the recognition of a response associated with the TopicList was successful, the first element in the array is the number of tokens returned and the following elements in the array are the tokens

-74-

for each identified speech element (one or more words).

Element 1 is n the Number of tokens returned.

Elements 2 through n+1 are the Token values for each speech element recognized. For example, consider the phrase "Telephone Dial Office". If the token value for the speech element "Telephone" is 7, for the speech element "Dial" is 12, and for the speech element "Office" is 103, then if they are all recognized successfully, the complete array returned would be four elements long with the values 3, 7, 12, 103. If recognition was not successful, the array is four elements long. The first element is zero. The second element indicates the most recent type of error that occurred. The third through fifth elements indicate the number of times each type of error occurred between when GETRESPONSEPLUS was called to when GETRESPONSEPLUS returned. In this case Element 1 has a value of 0 indicating that an error was detected. Element 2 has a value of 17 indicating that a response was not detected in the allowed time (TimeOut error) or 18 indicating that a response was detected, but it was not recognizable (out-of-vocabulary-word error) or 19 indicating that a spoke-to-soon error was detected. Element 3 has a value of x indicating the number of times a TimeOut error was detected. Element 4 has a value of y indicating the number of times an out-of-vocabulary-word error was detected. Element 5 has a

value of z indicating the number of times a spoke-too-soon error was detected.

Referring to FIG. 12, LISTANDSELECT user interface function first plays a Prompt. Then it plays each prompt in array ListOfMenuPrompts, pausing after each for a PauseTime. During these pauses, the recognizer listens for a spoken response that matches the topics in TopicList. LISTANDSELECT either returns an array of recognized tokens, or an error indicator. The parameters for LISTANDSELECT include Initial_Prompt, Timeout, STS_Sound, TopicList, ListOfMenuPrompts, PauseTime, and the Help_Prompt. The Initial_Prompt parameter is the initial prompt to be played to the user. The TimeOut parameter is the number of milliseconds to wait for a response, after playing all the prompts in ListOfMenuPrompts, or before flagging that a response was not detected. The STS_Sound parameter is the sound or prompt to be played if user speaks before a prompt finishes playing. Typically, STS_Sound will be a short tone or beep sound rather than a spoken phrase. The parameter TopicList is the vocabulary subset for the list of topics which the SRS 204 should use to identify the spoken response. The ListOfMenuPrompts parameter is an array of prompts which will be played one at a time. The first element in the array is a count of the number of prompts in ListOfMenuPrompts. The PauseTime parameter is the time

-76-

to pause after playing each prompt in ListOfMenuPrompts.

The PauseTime parameter has a value in milliseconds.

The Help_Prompt parameter is the prompt played when the user seems to need further instructions, including when the user says nothing. The voice controlled device returns a pointer to an integer array upon completion of the user interface function. If recognition was successful, the first element in the array is the number of tokens returned, and the following elements in the array are the tokens for each identified speech element (one or more words). Element 1 has a value of n indicating the number of tokens returned. Elements 2 through n+1 have a value of x indicating the token values for each speech element recognized. If recognition was not successful, the array is two elements long. The first element is zero. The second element indicates the type of error that occurred. In this case, Element 1 has a value of 0 indicating that an error was detected. Element 2 has a value of 17 indicating a response was not detected in the allowed time (Timeout error) or 18 indicating that a response was detected, but it was not recognizable (out-of-vocabulary-word error).

The ACOUSTICADDWORD function is used by application software to allow a user to add a phrase, also called a

-77-

voicetag, into the voice controlled device. These phrases can later be recognized using the GETRESPONSE and GETRESPONSEPLUS functions. The ACOUSTICADDWORD function can be used, for example, in a telephone to create dial-by-name entries. By storing a person's name ("John Smith") or identity ("Mother") or other distinguishing phrase ("My office number") with ACOUSTICADDWORD, a person could later call the number by saying "Call John Smith", "Call Mother", or "Call my office number".

ACOUSTICADDWORD stores the voicetag into a specified TopicList. In its operation, ACOUSTICADDWORD plays a prompt, receives and records a voicetag, verifies the voicetag, then stores the voicetag. AcousticAddWord has the ability to recover from errors by re-checking the voicetag more than once. AcousticAddWord checks and returns an error to the user in the event of duplication. The parameters for ACOUSTICADDWORD include Initial_Prompt, Timeout, STS_Sound, TopicList, MaxTries, Repeat_Prompt, Intervene_Prompt, Error_Prompt, Ok_Prompt, and Help_Prompt. The Initial_Prompt parameter is the initial prompt to be played to a user, such as "Say the new name" in the example of storing names in a voice controlled telephone's phonebook. The Timeout parameter is the number of milliseconds to wait before flagging a response that a failure was detected. The STS_Sound

(Spoke-Too_Soon Sound) parameter is the sound or prompt to be played if user speaks before the Prompt finishes playing. Typically, the STS_Sound will be a short tone or beep sound rather than a spoken phrase. The parameter TopicList is the vocabulary subset for which the SRS 204 should store the new voicetag in. The MaxTries parameter is the maximum number of times AcousticAddWord will re-prompt the user in an effort to get a good recognition. If recognition does not occur after MaxTries, AcousticAddWord will return an error indication. The Repeat_Prompt parameter is the prompt played to ask the user to repeat what was just said (e.g. "Please repeat what you said"). This prompt is used when a spoke-too-soon error occurred. The Intervene_Prompt parameter is a prompt played to ask the user to repeat himself (e.g. "There was too much noise. Please repeat what you said."). This prompt is played when there was too much noise during the previous recognition attempt. Error_Prompt parameter is the prompt played when the repeated name does not match the initial name, or if the name is a duplicate (e.g. "Please try again."). The OK_Prompt parameter is the prompt played when the new name has been successfully recorded and stored (e.g. "<name> is now stored in the address book"). The Help_Prompt parameter is the prompt played when the user seems to need further instructions, including when the user says nothing. The voice controlled device returns a

-79-

pointer to an integer array upon completion of the user interface function. If the recognition of a response associated with the AcousticAddWord was successful, the array is seven elements long. Element 1 is a value of 1 indicating successful recognition. Element 2 is a value indicating the token number assigned by the SRS 204, which corresponds to the voicetag that was stored. Element 3 is a pointer to a recorded copy of the voicetag. Element 4 is a value indicating the number of timeout errors that occurred. Element 5 is a value indicating the number of times there was a failure to match the name. Element 6 is a value indicating the number of times spoke-too-soon occurred. Element 7 is a value indicating the number of times the help prompt was played. If recognition was not successful, the array is six elements long. The first element is zero. The second element indicates the most recent type of error that occurred. The third through fifth elements indicate the number of times each type of error occurred between when AcousticAddWord was called to when AcousticAddWord returned. The sixth element indicates the number of times the help prompt was played. In this case, Element 1 is a value of indicating that an error was detected. Element 2 has a value of 17 indicating that a response was not detected in the allowed time (Timeout error); 18 indicating that a response was detected, but it was not recognizable (Noise error); 19

-80-

indicating that a spoke-to-soon error was detected; 20 indicating a Recognition failure (no match on repeat); or 21 indicating a Voicetag list already full. Element 3 is a value of x indicating the number of times a TimeOut error was detected. Element 4 is a value of y indicating the number of times a recognition error was detected. Element 5 is a value of z indicating the number of times a spoke-too-soon error was detected. Element 6 is a value indicating the number of times the help prompt was played.

ETIQUETTE FOR VOICE CONTROLLED DEVICES

The standard VUI includes an etiquette for voice controlled devices. Generally, voice controlled devices (also referred to as machines) should conduct themselves like well-behaved guests.

However, human factors and human issues involved in living with voice controlled devices are largely unexplored. In designing voice controlled devices, the following suggestions should be considered.

Machine Requests to Humans

Machines can ask humans to do things. Any request should be polite. For example, a voice activated cellular telephone might ask to be placed in its charger when its batteries are running low. Humans should always have the option to refuse a machine's request,

-81-

and the machine should politely accept that, unless the machine considers the situation threatening to human life or valuable data, in which case its protests can be more urgent.

Machines That Use the Telephone On Their Own

If a voice controlled device answers the telephone, or places a call to a human user, it should clearly identify itself as a machine if there is any risk of it being considered human.

Recording User Speech

No machine should record or transcribe a human user's conversations unless those humans present are aware that this is occurring.

Volume Levels

Machines should modulate their volume levels in response to ambient noise levels, unless specifically overridden by a human. Machines should be sensitive to when humans want them to be silent (for example, when humans are sleeping). Machines shouldn't babble needlessly, and should permit a user barge-in as a means to silence them.

Machine-to-Machine Communication

FIG. 13 is a block diagram of a pair of voice

controlled devices 102M and 102N (each also referred to as a machine) communicating, neither, one or both of which could be using the standard voice user interface 500 of the present invention in the communication environment 1300. Voice controlled devices can talk to each other to find out what other voice controlled devices are present, what kinds of information they understand, and to exchange information. For example, a voice controlled TV may ask a voice controlled VCR about necessary settings for it to operate. Machine-to-machine communication between voice controlled devices occurs in both audible and non-audible formats. Essentially, machine-to-machine communication using speech may occur over any speech-compatible media, including sound waves through air, conventional telephone links, Internet voice links, radio voice channels, and the like. Machine-to-machine communication can occur where none of the machines, some of the machines, or all of the machines include the VUI of the present invention.

Using the standard VUI, a voice controlled device can locate other voice controlled devices within a communications environment in a number of ways. These include overhearing a human interact with another machine, overhearing a machine interact with another machine, explicitly requesting nearby machines to identify themselves by using the identification phrase "<silence> What is out there?", explicitly seeking a

-83-

specific class of machines (e.g. all clocks) by addressing them by a name category "<silence> Clock are you out there?", or explicitly seeking a specific machine (e.g. a clock named Socrates) by addressing it by name "<silence> Socrates are you out there?".

In the first two cases, the process of listening to other conversations would reveal the other machines' names. In the other three cases the machines within earshot who respond to the "are you out there" command would respond with their names. In the last two cases, the "What is out there?" command is restricted to certain classes of machines and a specific named machine thereby limiting the number of machines that will respond to the command. Once the name of the target voice controlled device is known, the initiating voice controlled device can issue other commands (e.g. "Socrates what time is it?") to the other.

In some cases, a voice controlled device may need to talk to another voice controlled device, one or both of which may not adhere to the above protocol. In these cases, the machines can be explicitly programmed to issue the correct commands and recognize appropriate responses. A simple example of this interaction would be a voice controlled device with voice recognition capability and a telephone voice interface dialing a voice-based service such as a spoken report of the time, and simply capturing the desired data (the time).

The preferred embodiments of the present invention for METHOD AND APPARATUS FOR STANDARD VOICE USER INTERFACE AND VOICE CONTROLLED DEVICES are thus described. While the preferred embodiments of the present invention utilize a speaker independent voice recognition system, the present invention is also compatible with speaker dependent voice recognition systems. While the present invention has been described in particular embodiments, the present invention should not be construed as limited by such embodiments, but rather construed according to the claims that follow below.

MICROFICHE APPENDIX

ISD CONFIDENTIAL INFORMATION

**ISD-SR300**

Embedded Speech Recognition Processor

Advanced Information

SPEECH RECOGNITION IS NOW EASY TO ADD TO COMMAND AND CONTROL APPLICATIONS**VOICE RECOGNITION FEATURES**

- Speech recognition processor optimized for command and control applications
- Complete voice recognition subsystem requiring no host processor overhead
- Supports both speaker-independent and speaker-dependent continuous speech
 - Up to 65 speaker-dependent user-defined voicetags for voice-activated speed dialing
 - Supports up to four independent users through keyword activation and assigned phone books
 - Up to 30 speaker-independent application specific commands
 - Continuous digit recognition (0 through 9)
 - Natural Number digit recognition "oh" for zero, eight-hundred for 800, etc.)
 - Recognition of 'star', 'pound', and 'flash' for voice-dialing applications
- Recognition always active
 - Allows for voice control of on-the-air services
- Zero power voicetag storage
- User-friendly application-specific Voice User Interface (VUI)
 - True hands-free control
 - Activation by voice
 - Standardized, easy to use interface for voice activated appliances
 - Minimizes training by users
 - Accelerated application development by providing standard interface software
- Flexible API provides high level commands suitable for a wide variety of applications
- 99% recognition accuracy for both speaker-independent commands and Continuous digits.
- Uses Hidden Markov Models and probabilistic modeling to recognize a wide variety of speakers.

IMPORTANT NOTICE: This product concept and specifications are preliminary and subject to change without notice. Please contact ISD before using this information in any product design.

February 1999

iSD . 2045 Hamilton Avenue, San Jose, CA 95125 . TEL: 408/369-2400 . FAX: 408/369-2422 . <http://www.isd.com>

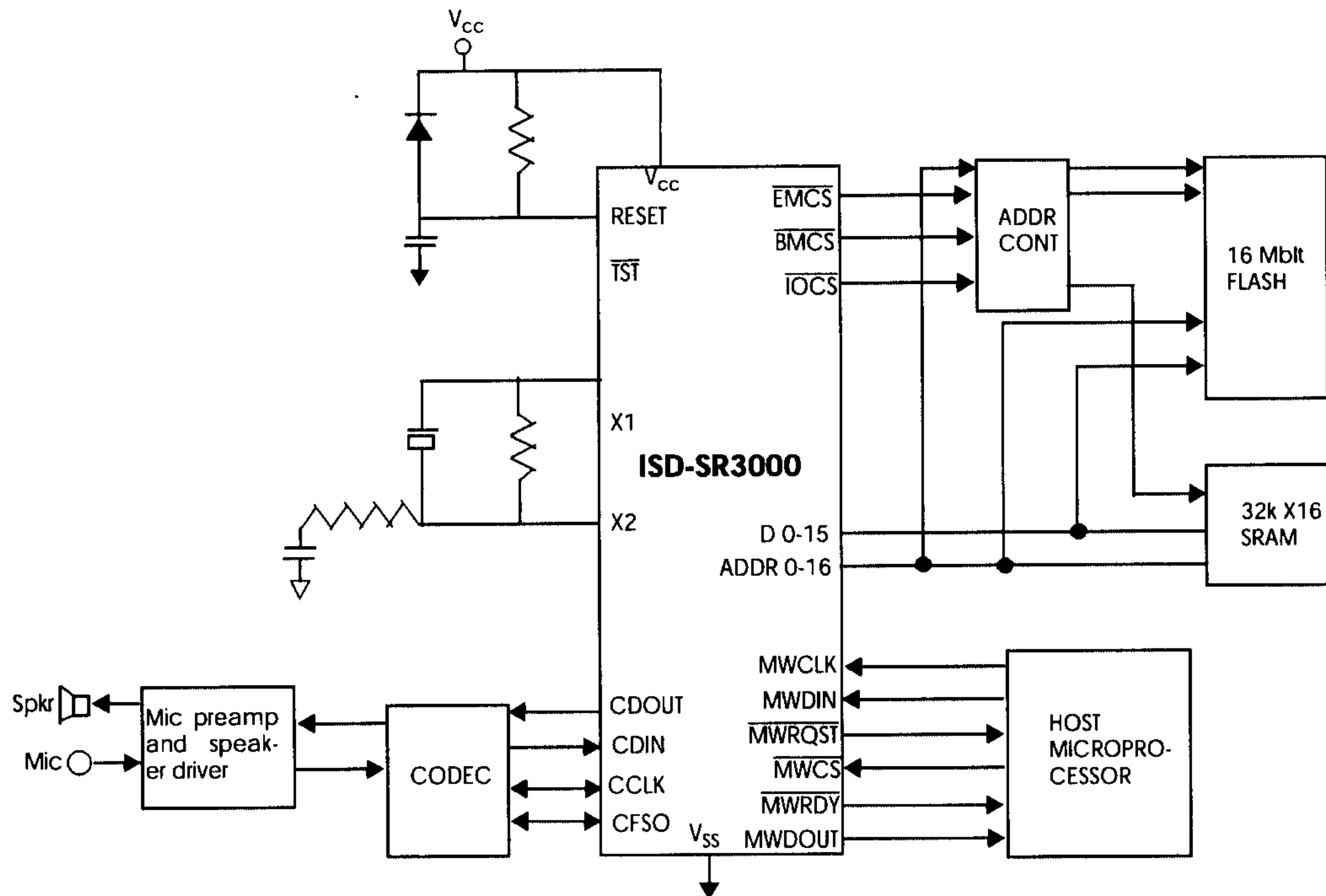
ISD CONFIDENTIAL INFORMATION

RECOGNITION ENGINE

- Combination DSP module and RISC processor optimized for speech recognition
- Interfaces to μ -Law, A-Law, or linear voice CODEC
- Serial interface to host microcontroller
- Single +5V or +3.3V power supply
- Quiescent current: 40mA
- Power-down current: 1mA
- Package: 80-pin QFP

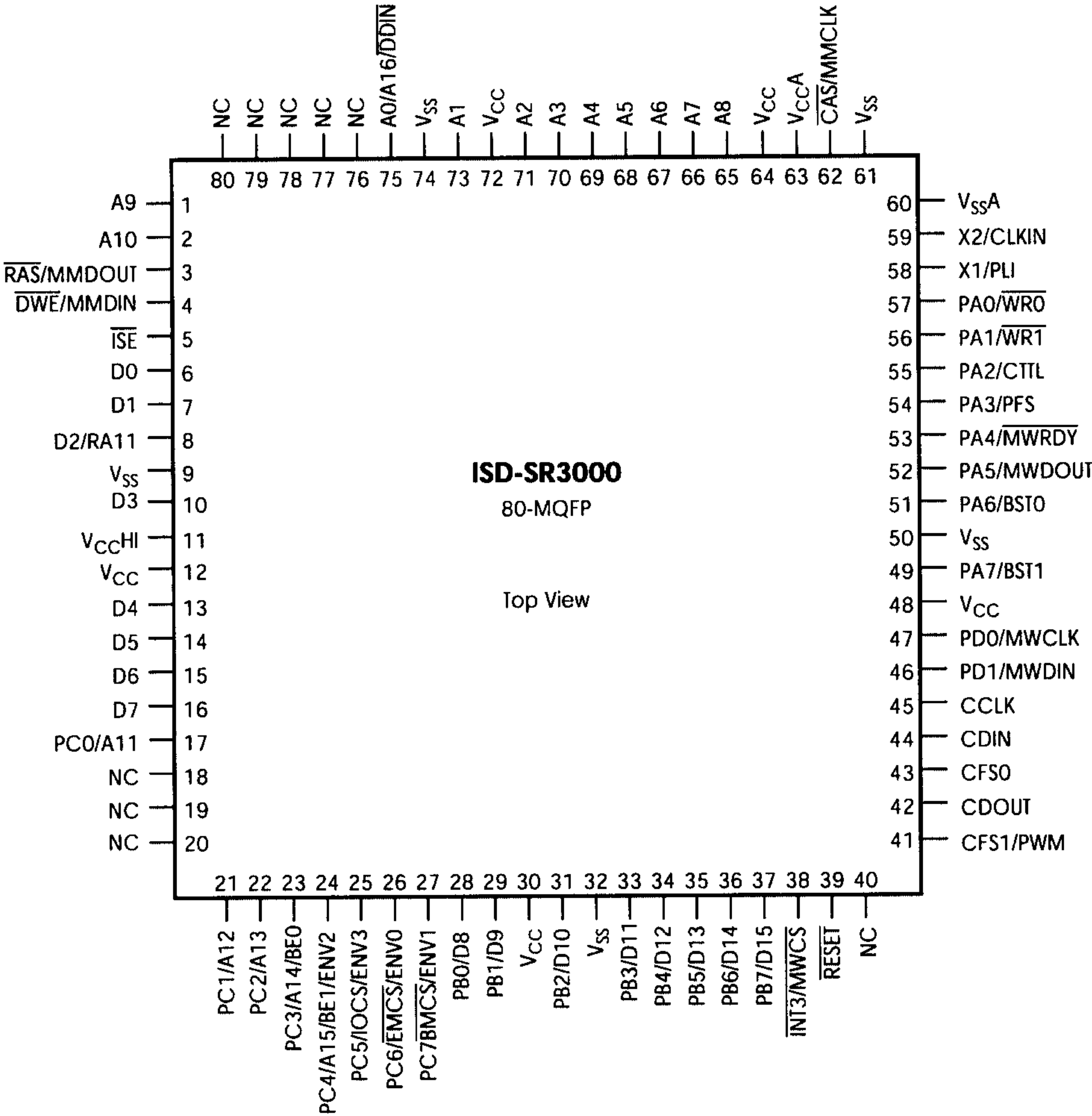
VOICE ACTIVATED APPLICATIONS

- Command and control applications where voice control is preferable to keypad control
- Desktop or cordless phones
- Cellular handsets
- Cellular car kits
- Automobile navigation or interior accessory control
- Home appliance and entertainment control.

Figure i: Stand-Alone Speech Recognition System Diagram

Chapter 1 HARDWARE

Figure 1-1: 80-MQFP Package Connection Diagram



NOTE: Pins marked NC should not be connected.

ISD CONFIDENTIAL INFORMATION

PIN ASSIGNMENT

The following sections detail the pins of the ISD-SR3000 processor. Slashes separate the names of signals that share the same pin.

PIN SIGNAL ASSIGNMENT

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

Table 1-1: ISD-SR3000 Pin—Signal Assignment

Pin	Signal Type	Signal Name	Direction
A(0–15)	TTL	A(0–15)	Output
CCLK	TTL	CCLK	Output
CDIN	TTL	CDIN	Input
CDOUT	TTL	CDOUT	Output
CFS0	TTL	CFS0	Output
D(0–15)	TTL	D(0–15)	Input/Output
EMCS/ENVO	TTL ¹ CMOS ²	EMCS ENVO	Output Input
MWCLK	TTL	MWCLK	Input
MWCS	TTL ³	MWCS	Input
MWDIN	TTL	MWDIN	Input
MWRDY	TTL	MWRDY	Output
MWRQST	TTL	MWRQST	Output
MWDOUT	TTL	MWDOUT	Output
BMCS	TTL	BMCS	Output
IOCS	TTL	IOCS	Output
RESET	Schmitt ³	RESET	Input
TST	TTL	TST	Input
V _{CC}	Power	V _{CC}	
V _{SS}	Power	V _{SS}	
X1	XTAL	X1	OSC
X2/CLKIN	XTALTTL	X2 CLKIN	OSC Input

1. TTL1 output signals provide CMOS levels in the steady state for small loads.
2. Input during RESET, CMOS level input.
3. Schmitt trigger input.

ISD CONFIDENTIAL INFORMATION

Table 1-2: ISD-SR3000 Pin—Signal Assignment (Continued)

	Pin	Signal	Type	Description
1	A9	A9	O	Address bit 9
2	A10	A10	O	Address bit 10
3	MMDOUT	MMDOUT	O	Master MICROWIRE data output
4	MMDIM	MMDIN	I	Master MICROWIRE data input
5	ISE			
6	D0	D0	I/O	Data bit 0
7	D1	D1	I/O	Data bit 1
8	D2/RA11	D2	I/O	Data bit 2
9	V _{ss}	V _{ss}	Power	Ground for on-chip logic and output drivers
10	D3	D3	I/O	Data bit 3
11	V _{cc} HI	V _{cc} HI	Power	Power: +3.3V or +5V for on-chip voltage regulator
12	V _{cc}	V _{cc}	Power	Power: +3.3V or +5V for on-chip logic and output drivers
13	D4	D4	I/O	Data bit 4
14	D5	D5	I/O	Data bit 5
15	D6	D6	I/O	Data bit 6
16	D7	D7	I/O	Data bit 7
17	PC0/A11	PC0	I/O	Port C, bit 0
		A11	O	Address bit 11
18	NC	NC		Do not connect
19	NC	NC		Do not connect
20	NC	NC		Do not connect
21	PC/A12	PC1	O	Port C, bit 1
		A12	O	Address bit 12
22	PC2/A13	PC2	O	Port C, bit 2
		A13	O	Address bit 13
23	PC3/A14/BE0	PC3	O	Port C, bit 3
		A14	O	Address bit 14
		BE0	O	Byte enable bit 0
24	PC4/A15BE1ENV2	PC4	O	Port C, bit 4
		A15	O	Address bit
		BE1	O	Byte enable bit 1
		ENV2	O	Environment select bit 2

ISD CONFIDENTIAL INFORMATION

Table 1-2: ISD-SR3000 Pin—Signal Assignment (Continued)

Pin	Pin Name	Signal Name	Type	Description
25	PC5/IOCS/ENV3	PC5	O	Port C, bit 5
		IOCS	O	I/O expansion chip select
		ENV3	O	Environment select bit 3
26	PC6/EMCS/ENV0	PC6	O	Port C, bit 6
		EMCS	O	Expansion memory chip select
		ENV0	O	Environment select bit 0
27	PC7/BMCS/ENV1	PC7	O	Port C, bit 7
		BMCS	O	Base memory chip select
		ENV1	O	Environment select bit 1
28	PB0/D8	PB0	O	Port B, bit 0
		D8	I/O	Data bit 8
29	PB1/D9	PB1	O	Port B, bit 1
		D9	I/O	Data bit 9
30	V _{CC}	V _{CC}	Power	Power: +3.3V or +5V for on-chip logic and output drivers
31	PB2/D10	PB2	O	Port B, bit 2
		D10	I/O	Data bit 10
32	V _{SS}	V _{SS}	Power	Ground for on-chip logic and output drivers
33	PB3/D11	PB3	O	Port B, bit 3
		D11	I/O	Data bit 11
34	PB4/D12	PB4	O	Port B, bit 4
		D12	I/O	Data bit 12
35	PB5/D13	PB5	O	Port B, bit 5
		D13	I/O	Data bit 13
36	PB6/D14	PB6	O	Port B, bit 6
		D14	I/O	Data bit 14
37	PB7/D15	PB7	O	Port B, bit 7
		D15	I/O	Data bit 15
38	INT3/MWCS	INT3	I	External interrupt
		MWCS	I	MICROWIRE chip select
39	RESET	RESET	I	Reset
40	NC	NC		Do not connect
41	CFS1/PWM	CFS1	O	CODEC 1 Frame synchronization
		PWM	O	Pulse width modulation

Table 1-2: ISD-SR3000 Pin-Signal Assignment (Continued)

Pin	Pin Name	Signal Name	Type	Description
42	CDOUT	CDOUT	O	Data output to CODEC
43	CFSO	CFSO	I/O	CODEC 0 Frame synchronization
44	CDIN	CDIN	I	Data input from CODEC
45	CCLK	CCLK	I/O	CODEC Master/slave clock
46	PD1/MWDIN	PD1	I	Port D, bit 1
		MWDIN	I	MICROWIRE data input
47	PDO/MWCLK	PDO	I	Port D, bit 0
		MWCLK	I	MICROWIRE clock
48	V _{CC}	V _{CC}	Power	Power: +3.3V or +5V for on-chip logic and output drivers
49	PA7/BST1	PA7	I/O	Port A, bit 7
		BST1	O	Bus status bit 1
50	V _{SS}	V _{SS}	Power	Ground for on-chip logic and output drivers
51	PA6/BST0	PA6	I/O	Port A, bit 6
		BST0	O	Bus status bit 0
52	PA5/MWOUT	PA5	I/O	Port A, bit 5
		MWOUT	O	MICROWIRE data output
53	PA4/MWRDY	PA4	I/O	Port A, bit 4
		MWRDY	O	MICROWIRE ready
54	PA3/PFS	PA3	I/O	Port A, bit 3
		PFS		
55	PA2/CTTL	PA2	I/O	Port A, bit 2
		CTTL	O	CPU Clock
56	PA1/WR1	PA1	I/O	Port A, bit 1
		WR1		
57	PA0/WR0	PA0	I/O	Port A, bit 0
		WR0		
58	X1/PL1	X1	Osc,	Crystal oscillator interface
		PL1		
59	X2/CLKIN	X2	Osc,	Crystal oscillator interface
		CLKIN	I	Oscillator clock input
60	V _{SSA}	V _{SSA}	Power	Ground for on-chip analog circuitry
61	V _{SS}	V _{SS}	Power	Ground for on-chip logic and output drivers
62	CAS/MMCLK	CAS	O	DRAM column address strobe

ISD CONFIDENTIAL INFORMATION

Table 1-2: ISD-SR3000 Pin-Signal Assignment (Continued)

Pin	Pin Name	Signal Name	Type	Description
		MMCLK	O	Master MICROWIRE clock
63	V _{ccA}	V _{ccA}	Power	Power: +3.3V or +5V for on-chip analog circuitry
64	V _{cc}	V _{cc}	Power	Power: +3.3V or +5V for on-chip logic and output drivers
65	A8	A8	O	Address bit 8
66	A7	A7	O	Address bit 7
67	A6	A6	O	Address bit 6
68	A5	A5	O	Address bit 5
69	A4	A4	O	Address bit 4
70	A3	A3	O	Address bit 3
71	A2	A2	O	Address bit 2
72	V _{cc}	V _{cc}	Power	Power: +3.3V or +5V for on-chip logic and output drivers
73	A1	A1	O	Address bit 1
74	V _{ss}	V _{ss}	Power	Ground for on-chip logic and output drivers
75	A0/A16/DDIN	A0	O	Address bit 0
		A16	O	Address bit 16
		DDIN	O	Data direction
76	NC	NC		Do not connect
77	NC	NC		Do not connect
78	NC	NC		Do not connect
79	NC	NC		Do not connect
80	NC	NC		Do not connect

SIGNAL DESCRIPTION

The following signals are used for the interface protocol. Input and output are relative to the ISD-SR3000.

INPUT SIGNALS

MWDIN

MICROWIRE Data In. Used for input only, for transferring data from the microcontroller to the ISD-SR3000.

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 MWDOOUT on the clock falling edge. The signal should remain low when switching $\overline{MWCS}$.

$\overline{MWCS}$

MICROWIRE Chip Select. The $\overline{MWCS}$ signal is cleared to 0, to indicate that the ISD-SR3000 is being accessed. Setting $\overline{MWCS}$ to 1 causes the ISD-SR3000 to start driving MWDOOUT with bit 7 of the transmitted value. Setting the $\overline{MWCS}$ signal resets the transfer-bit counter of the protocol, so the signal can be used to synchronize between the ISD-SR3000 and the microcontroller.

To prevent false detection of access to the ISD-SR3000 due to spikes on the MWCLK signal, use this chip select signal, and toggle the MWCLK input signal, only when the ISD-SR3000 is accessed.

OUTPUT SIGNALS

MWDOOUT

MICROWIRE Data Out. Used for output only, for transferring data from the ISD-SR3000 to the microcontroller. When the ISD-SR3000 receives data it is echoed back to the microcontroller on this signal, unless the received data is 0xAA. In this case, the ISD-SR3000 echoes a command's return value.

$\overline{MWRDY}$

MICROWIRE Ready. When active (0), this signal indicates that the ISD-SR3000 is ready to transfer (receive or transmit) another byte of data.

This signal is set to 1 by the ISD-SR3000 after each byte transfer has been completed. It remains 1, while the ISD-SR3000 is busy reading the byte, writing the next byte, or executing the received command (after the last parameter has been received). $\overline{MWRDY}$ is cleared to 0 after reset. For proper operation after a hardware reset, this signal should be pulled up.

$\overline{MWRQST}$

MICROWIRE Request. When active (0), this signal indicates that new status information is available. $\overline{MWRQST}$ is deactivated (set to 1), after the ISD-SR3000 receives a GSW (Get Status Word) command from the microcontroller. After reset, this signal is active (0) to indicate that a reset occurred. $\overline{MWRQST}$, unlike all the signals of the communication protocol, is an asynchronous line that is controlled by the ISD-SR3000 firmware.

ISD CONFIDENTIAL INFORMATION

SIGNAL USE IN THE INTERFACE PROTOCOL

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

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

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

While the $\overline{\text{MWCS}}$ signal is active (0), the ISD-SR3000 reads data from MWDIN on every rising edge of MWCLK. ISD-SR3000 also writes every bit back to MWDOOUT. This bit is either the same bit which was read from MWDIN (in this case it is written back as a synchronization echo after some propagation delay), or it is a bit of a value the ISD-SR3000 transmits to the microcontroller (in this case it is written on every falling edge of the clock).

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

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

1. Active (0) $\overline{\text{MWRDY}}$ signals the microcontroller that the last eight bits of data transferred to/from the voice module were accepted and processed (see below).
2. The $\overline{\text{MWRDY}}$ signal is deactivated (set to 1 by the ISD-SR3000) after 8-bits of data were transferred to/from the ISD-SR3000. The bit is set following the falling edge of the eighth MWCLK clock-cycle.

3. The $\overline{\text{MWRDY}}$ signal is activated (cleared to 0) by the ISD-SR3000 when it is ready to receive the first parameter byte (if there are any parameters) and so on till the last byte of parameters is transferred. An active $\overline{\text{MWRDY}}$ signal after the last byte of parameters indicates that the command was parsed and (if possible) executed. If that command has a return value, the microcontroller must read the value before issuing a new command.
4. When a return value is transmitted, the $\overline{\text{MWRDY}}$ signal is deactivated after every byte, and activated again when the ISD-SR3000 is ready to send another byte, or to receive a new command.

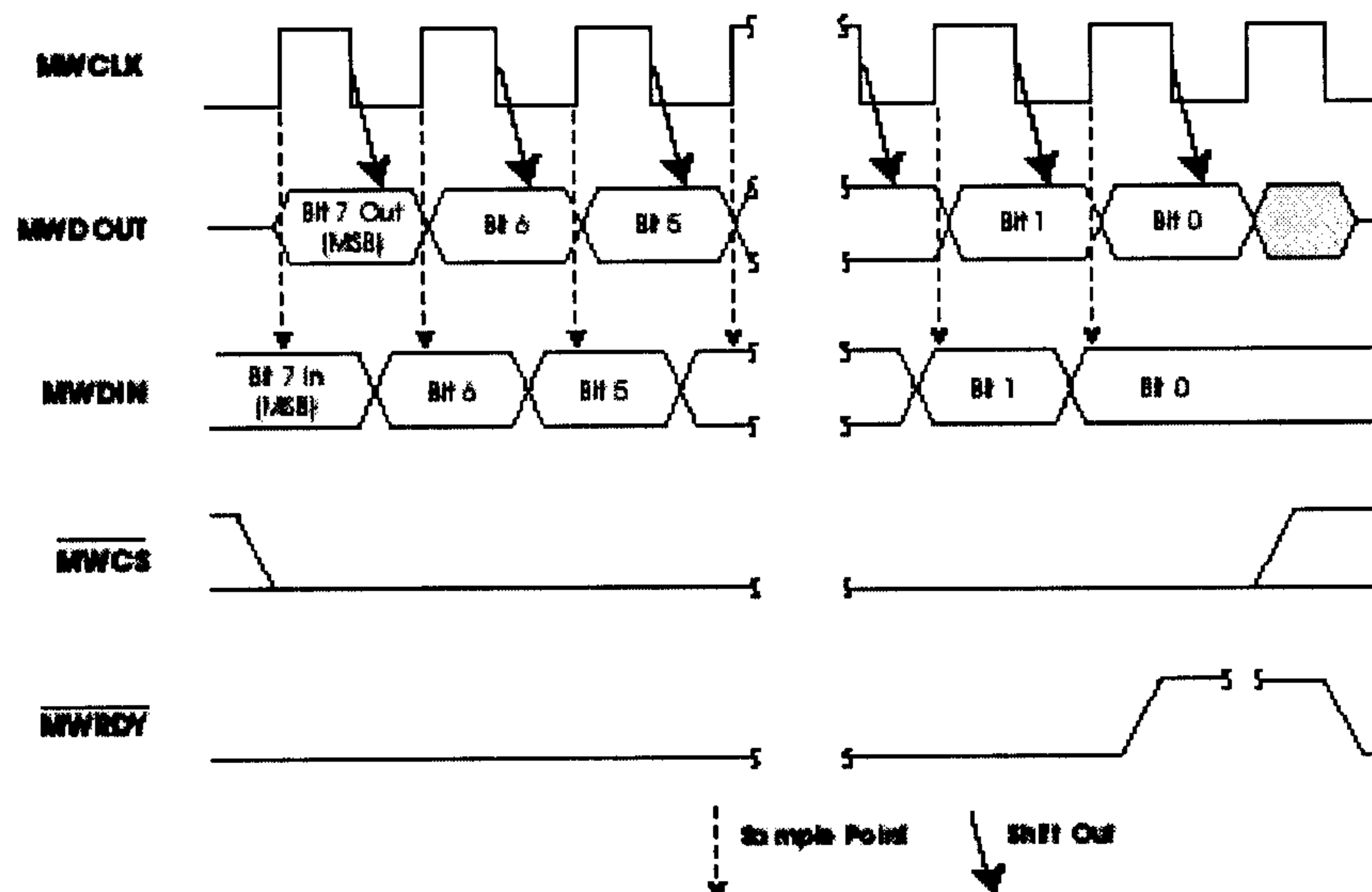
The $\overline{\text{MWRDY}}$ signal is activated (cleared to 0) after reset, and after a protocol time-out. (See "INTERFACE PROTOCOL ERROR HANDLING" on page 1-5.)

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

1. The $\overline{\text{MWRQST}}$ signal is activated (cleared to 0), when the status word is changed.
2. The $\overline{\text{MWRQST}}$ signal remains active (0), until the ISD-SR3000 receives a GSW command.

Figure 1-2 illustrates the sequence of activities during a MICROWIRE data transfer.

Figure 1-2: Sequence of Activities During a MICROWIRE Byte Transfer



INTERFACE PROTOCOL ERROR HANDLING

Interface Protocol Time-outs

Depending on the ISD-SR3000's state, if more than 100 milliseconds elapse between the assertion of the $\overline{\text{MWRDY}}$ signal and the transmission 8th bit of the next byte pertaining to the same command transaction, a time-out event occurs, and the ISD-SR3000 responds as follows:

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

Echo Mechanism

The ISD-SR3000 echoes back to the microcontroller all the bits received by the ISD-SR3000. Upon detection of an error in the echo, the microcontroller should stop the protocol clock, which eventually causes a time-out error (i.e., ERR_TIMEOUT bit is set in the error word).

NOTE When a command has a return value, the ISD-SR3000 transmits bytes of the return value instead of the echo value.

The ISD-SR3000 transmits a byte as an echo when it receives the value 0xAA from the microprocessor. Upon detection of an error the ISD-SR3000 activates the $\overline{\text{MWRQST}}$ signal, and sets the ERR_COMM bit in the error word.

ISD CONFIDENTIAL INFORMATION

FUNCTIONAL DESCRIPTION

This section provides details of the functional characteristics of the ISD-SR3000 processor. It is divided into the following sections:

- Resetting
- Clocking
- Power-down mode
- Power and grounding
- CODEC interface

RESETTING

The $\overline{\text{RESET}}$ pin is used to reset the ISD-SR3000 processor.

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

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

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

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

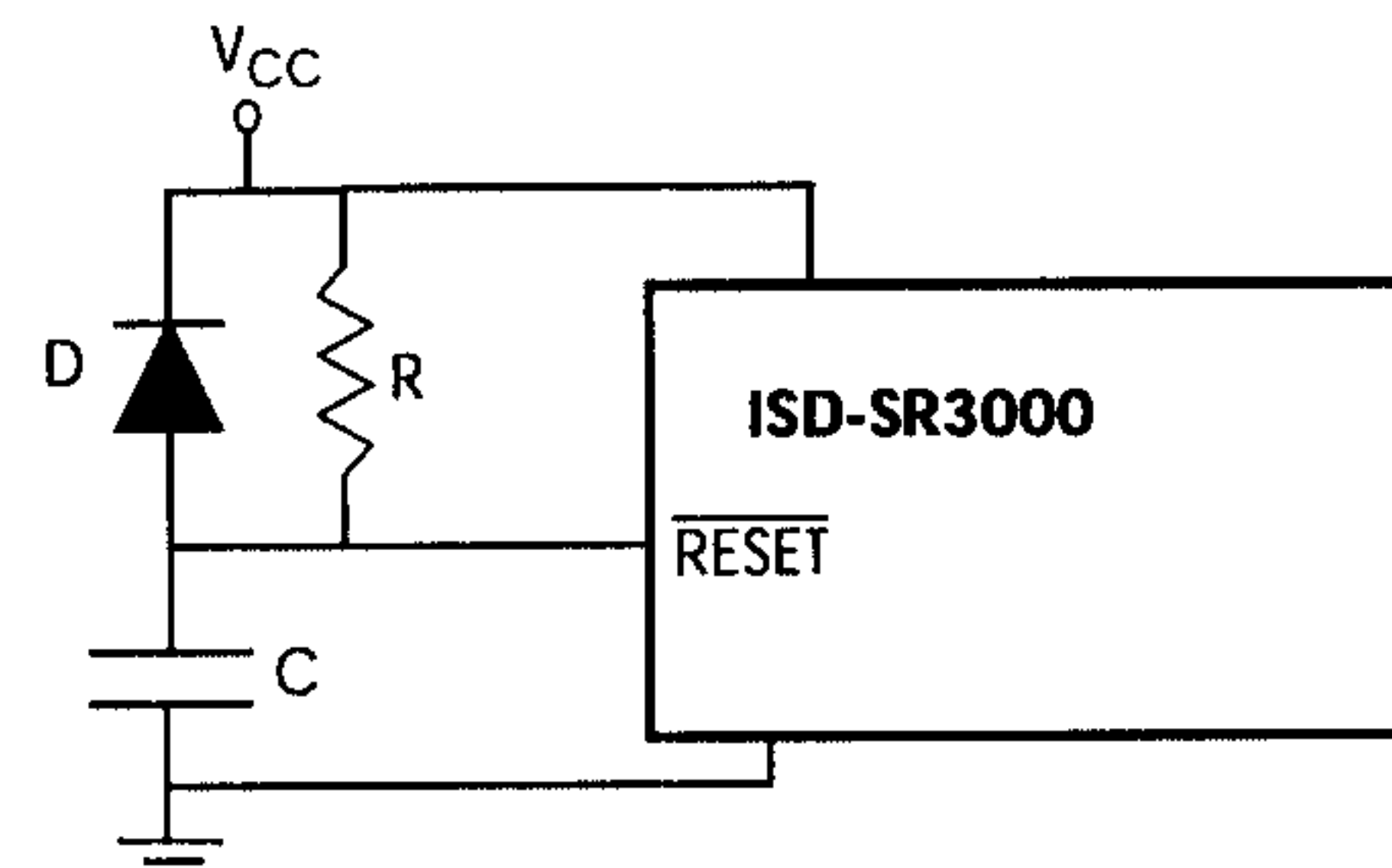
System Load on ENV0

For any load on the ENV0 pin, the voltage should not drop below V_{ENVh} .

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

Figure 1-3 shows a recommended circuit for generating a reset signal when the power is turned on.

Figure 1-3: Recommended Power-On Reset Circuit

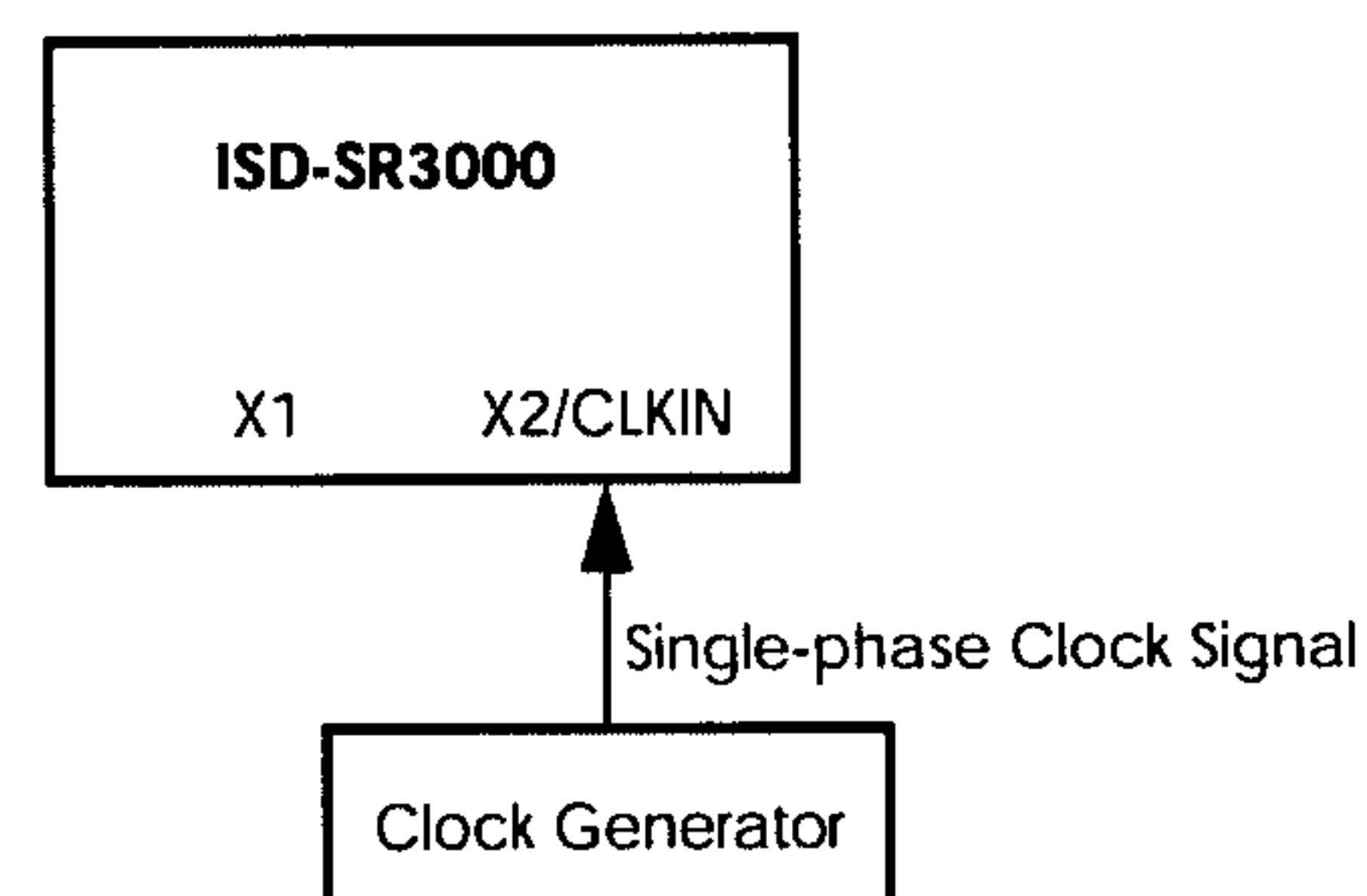
**CLOCKING**

The ISD-SR3000 processor provides an internal oscillator that interacts with an external clock source through the X1 and X2/CLKIN pins. Either an external single-phase clock signal, or a crystal oscillator, may be used as the clock source.

External Single-Phase Clock Signal

If an external single-phase clock source is used, it should be connected to the CLKIN signal as shown in Figure 1-4, and should conform to the voltage-level requirements for CLKIN stated in "Electrical Characteristics" on page 1-11.

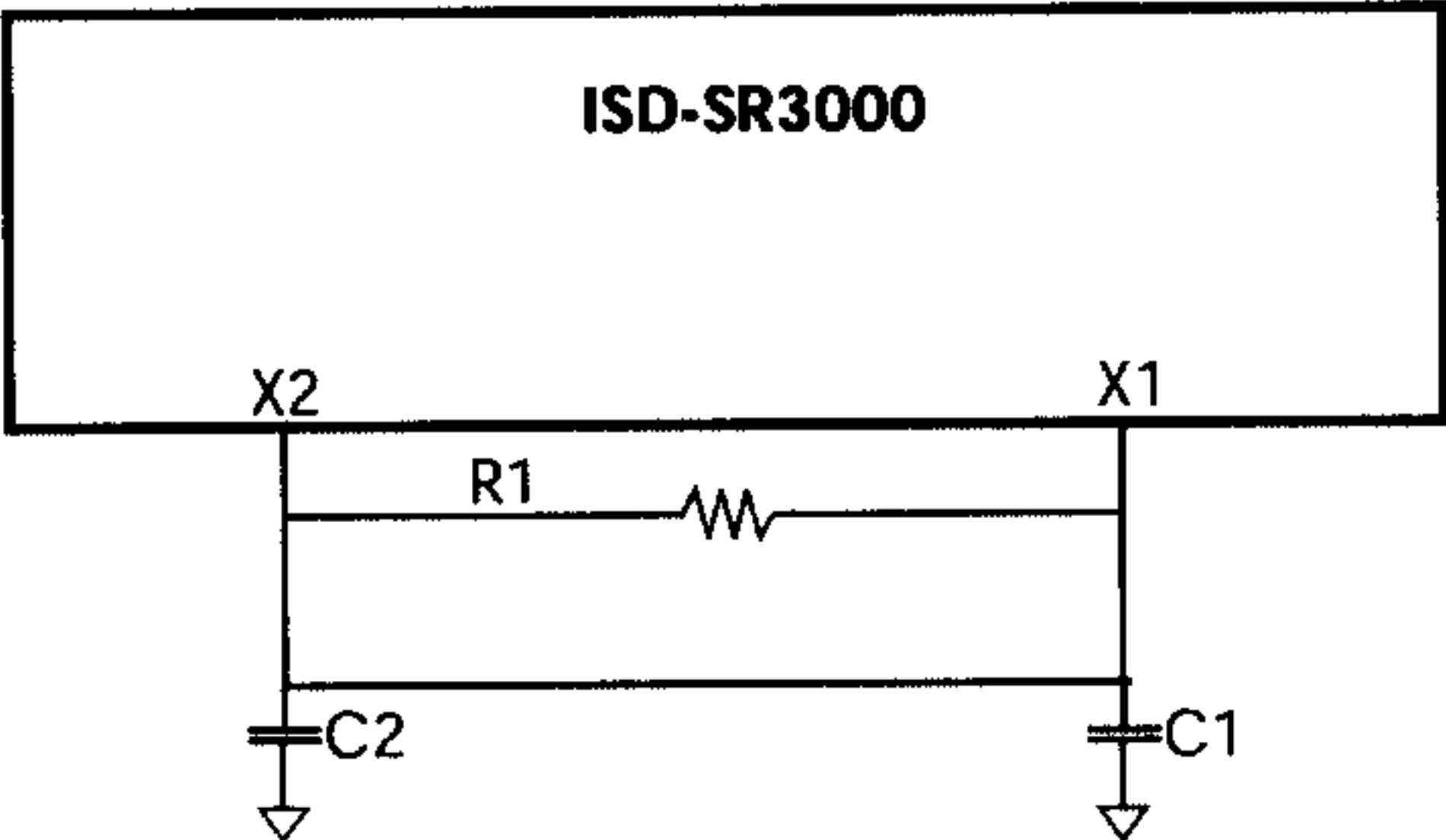
Figure 1-4: External Clock Source



Crystal Oscillator

A crystal oscillator is connected to the on-chip oscillator circuit via the X1 and X2 signals, as shown in Figure 1-5.

Figure 1-5: Connections for an External Crystal Oscillator



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

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

Table 1-3 lists the components in the crystal oscillator circuit.

Table 1-3: Crystal Oscillator Component List

Crystal Resonator	Resonance Frequency	4.096 MHz	
Resistor R1		10MΩ	5%
Capacitors C1, C2		33pF	20%

ISD CONFIDENTIAL INFORMATION

POWER-DOWN MODE

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

In power-down mode, the clock frequency of the ISD-SR3000 processor is reduced and some of the processor modules are deactivated. As a result, the ISD-SR3000 processor consumes considerably less power than in normal-power mode.

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

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

The ISD-SR3000 stores voice tags in Flash memory. When Flash memory is used for storage, power does not need to be maintained to the processor to preserve stored messages.

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

The PDM (Go To Power-down Mode) command switches the ISD-SR3000 to power-down mode. (For an explanation of the ISD-SR3000 processor commands, see Table 2-1.) This command may only be issued when the processor is in the idle mode. (For an explanation of the SR3000 states, see "Command Execution" on page 45.) If it is necessary to switch the power-down mode from any other state, the controller must first issue an S command to switch the processor to the idle state, and then issue the PDM command. Sending any command while in power-down mode resets the ISD-SR3000 processor detectors, and returns it to normal operation mode.

THE CODEC INTERFACE

The ISD-SR3000 provides an on chip interface for analog and digital telephony, supporting master and slave CODEC interface modes. In master mode, the ISD-SR3000 controls the operation of the CODEC for use in analog telephony. In the slave mode, the ISD-SR3000 CODEC interface is controlled by an external source. This mode is used in digital telephony (i.e., ISDN or DECT lines). The slave mode is implemented with respect to IOM-2™/CGI specifications.

See Table 1-4 for CODEC options for the ISD-SR3000 (ISD supports compatible CODECS in addition to those listed below).

The CODEC interface supports the following features:

- Master Mode or Slave Mode.
- 8- or 16-bit channel width.
- Long (Variable) or Short (Fixed) Frame Protocol.
- Single or Double bit clock rate.
- Single or Dual Channel CODECS
- One or Two CODECS
- Multiple clock and sample rates.
- One or Two frame sync signals

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

SHORT FRAME PROTOCOL

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

LONG FRAME PROTOCOL

When long frame protocol is configured, eight or sixteen data bits are exchanged with each CODEC, as for the short frame protocol. However, for the long frame protocol, data transfer starts by setting CFSO to 1 for eight or sixteen CCLK cycles. Short or long frame protocol is available in both Master and Slave modes.

ISD CONFIDENTIAL INFORMATION

Table 1-4:Supported CODEC Devices

National Semiconductor	TP3054	Single CODEC	5V	μ -Law
National Semiconductor	TP3057	Single CODEC	5V	A-Law
OKI	MSM7533V	Dual CODEC	5V	μ -Law, A-Law
OKI	MSM7704	Dual CODEC	3.3V	μ -Law, A-Law, LV
Macronix	MX93002FC	Dual rail CODEC	5V	μ -Law
Lucent	T7502	Dual CODEC	5V	A-Law
Lucent	T7503	Dual CODEC	5V	μ -Law

Channel Width

The CODEC interface supports both 8-bit and 16-bit channel width in Master and Slave Modes. Figure 1 shows how the CODEC interface signals behave when short frame protocol is configured.

SLAVE MODE

The ISD-SR3000 supports digital telephony applications including DECT and ISDN by providing a Slave Mode of operation. In Slave Mode operation, the CCLK signal is input to the SR-3000 and controls the frequency of the CODEC interface operation. The CCLK may be any frequency between 500 kHz and 4 MHz. Both long and short frame protocols are supported with only the CFS1 output signal width affected. The CFSO input signal must be a minimum of one CCLK cycle.

ISD CONFIDENTIAL INFORMATION

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

This interface supports ISDN protocol with one bit clock rate or double bit clock rate. The exact format is selected with the CFG command. The slave CODEC interface uses four signals: CDIN, CDOUT, CCLK, and CFSO. The CDIN, CCLK, and CFSO input pins and the CDOUT output pins are connected to the ISDN/DECT agent. Data is transferred to the ISD-SR3000 through the CDIN pin and read out through the CDOUT pin. The CFSO pin is used to define the start of each frame (see below). The source of that signal is at the master side. The CCLK is used for bit timing of CDIN and CDOUT. The rate of the CCLK is configured via the CFG command and can be twice the data rate, or at the data rate. The source of that signal is at the master side.

The CODEC interface is enabled while the system is in normal operation mode and when MCFG.CDE = 1. It is disabled when MCFG.CDE = 0, during reset, or whenever the system is in power down mode.

Table 1-5:Typical CODEC Applications

Analogue	single	1	Master	8	short or long	1	2.048	8000	1
ISDN--8bit digital--A-Law	dual	2	Slave	8	short	1 or 2	2.048	8000	1
Linear	single	1	Master	16	short	1	2.048	8000	1
IOM-2/GCI	single or dual	1-3	Slave	8	short	1 or 2	1.536	8000	1

ISD CONFIDENTIAL INFORMATION

SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Storage temperature	-65°C to +150°C
Temperature under bias	-40°C to +85°C
All input or output voltages, with respect to GND	-0.5 V to +6.5 V

ELECTRICAL CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5 V ±10%, GND = 0 V

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

Table 1-6:Electrical Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C _X	X1 and X2 Capacitance ¹			17.0		pF
I _{CC1}	Active Supply Current	Normal Operation Mode, Running Speech Applications ²		40.0	80.0	mA
I _{CC2}	Standby Supply Current	Normal Operation Mode, DSPM Idle ²		30.0		mA
I _{CC3}	Power-down Mode Supply Current	Power-Down Mode ^{2,3}			0.7	µA
I _L	Input Load Current ¹	0 V ≤ V _{IN} ≤ V _{CC}	-5.0		5.0	µA
I _O (Off)	Output Leakage Current (I/O pins in Input Mode) ¹	0 V ≤ V _{OUT} ≤ V _{CC}	-5.0		5.0	µA
V _{Hh}	CMOS Input with Hysteresis, Logical 1 Input Voltage		2.1			V
V _{Hi}	CMOS Input with Hysteresis, Logical 0 Input Voltage				0.8	V
V _{Hys}	Hysteresis Loop Width ⁴		0.5			V
V _{IH}	TTL Input, Logical 1 Input Voltage		2.0		V _{CC} + 0.5	V
V _{IL}	TTL Input, Logical 0 Input Voltage		-0.5		0.8	V
V _{OH}	Logical 1 TTL, Output Voltage	I _{OH} = -0.4 µA	2.4			V

ISD CONFIDENTIAL INFORMATION

Table 1-6: Electrical Characteristics (Continued)

V _{OHWC}	MMCLK, MMDOUT and $\overline{\text{EMCS}}$ Logical 1, Output Voltage	I _{OH} = -0.4 μA	2.4			V
		I _{OH} = -50 μA ⁵	V _{CC} - 0.2			V
V _{OL}	Logical 0, TTL Output Voltage	I _{OL} = 4 mA			0.45	V
		I _{OL} = 50 μA ⁵			0.2	V
V _{OLWC}	MMCLK, MMDOUT and $\overline{\text{EMCS}}$ Logical 0, Output Voltage	I _{OL} = 4 mA			0.45	V
		I _{OL} = 50 μA ⁵			0.2	V
V _{XH}	CLKIN Input, High Voltage	External Clock	2.0			V
V _{XL}	CLKIN Input, Low Voltage	External Clock			0.8	V

- 1. Maximum 20 μA for all pins together.
- 2. I_{OUT} = 0, T_A = 25°C, V_{CC} = 5 V, operating from a 4.096 MHz crystal and running from internal memory with Expansion Memory disabled.
- 3. All input signals are tied to 1 or 0 (above V_{CC} - 5 V or below V_{SS} = 5 V.)
- 4. Guaranteed by design.
- 5. Measured in power-down mode. The total current driven or sourced by all the ISD-SR3000 processor's output signals is less than 50 μA .

ISD CONFIDENTIAL INFORMATION

SWITCHING CHARACTERISTICS

Definitions

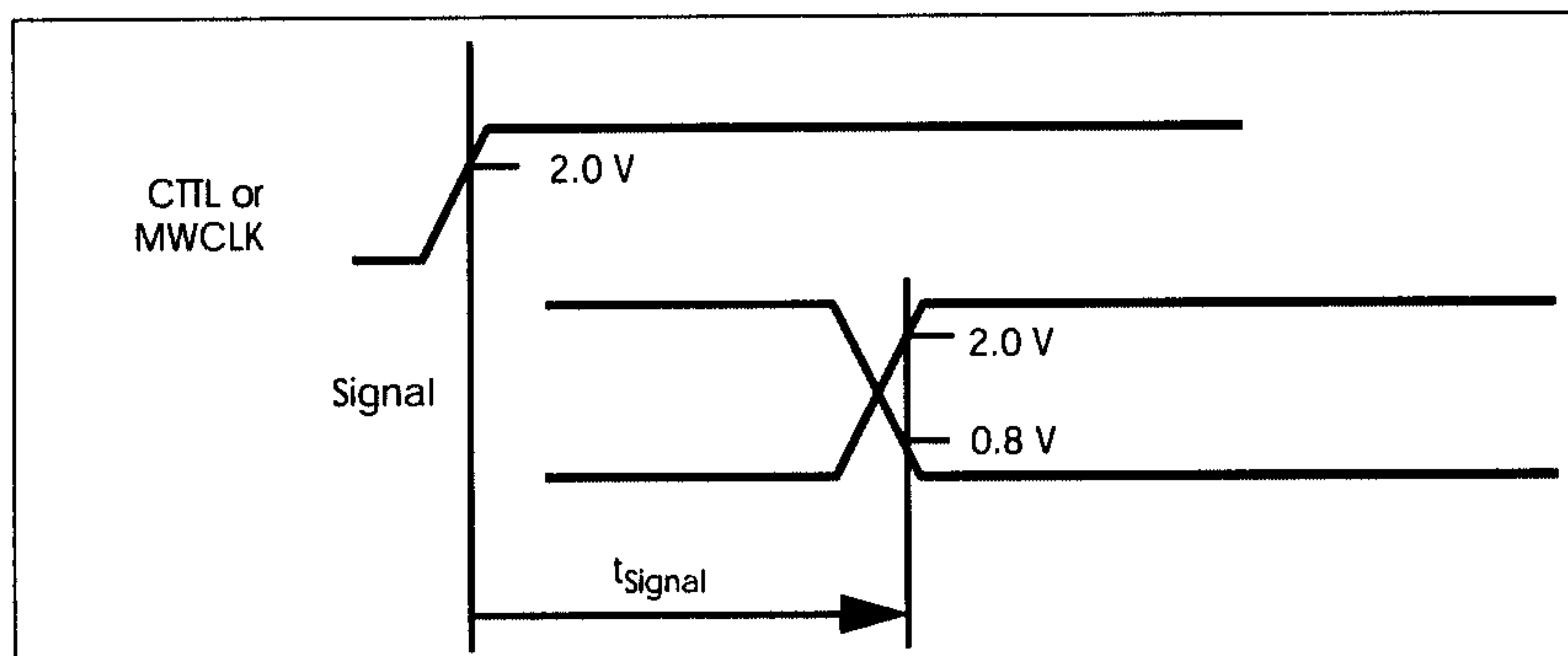
All timing specifications in this section refer to 0.8 V or 2.0 V on the rising or falling edges of the signals, as illustrated in Figure 1-7 through 1-12, unless specifically stated otherwise.

Maximum times assume capacitive loading of 50 pF.

CLKIN crystal frequency is 4.096 MHz.

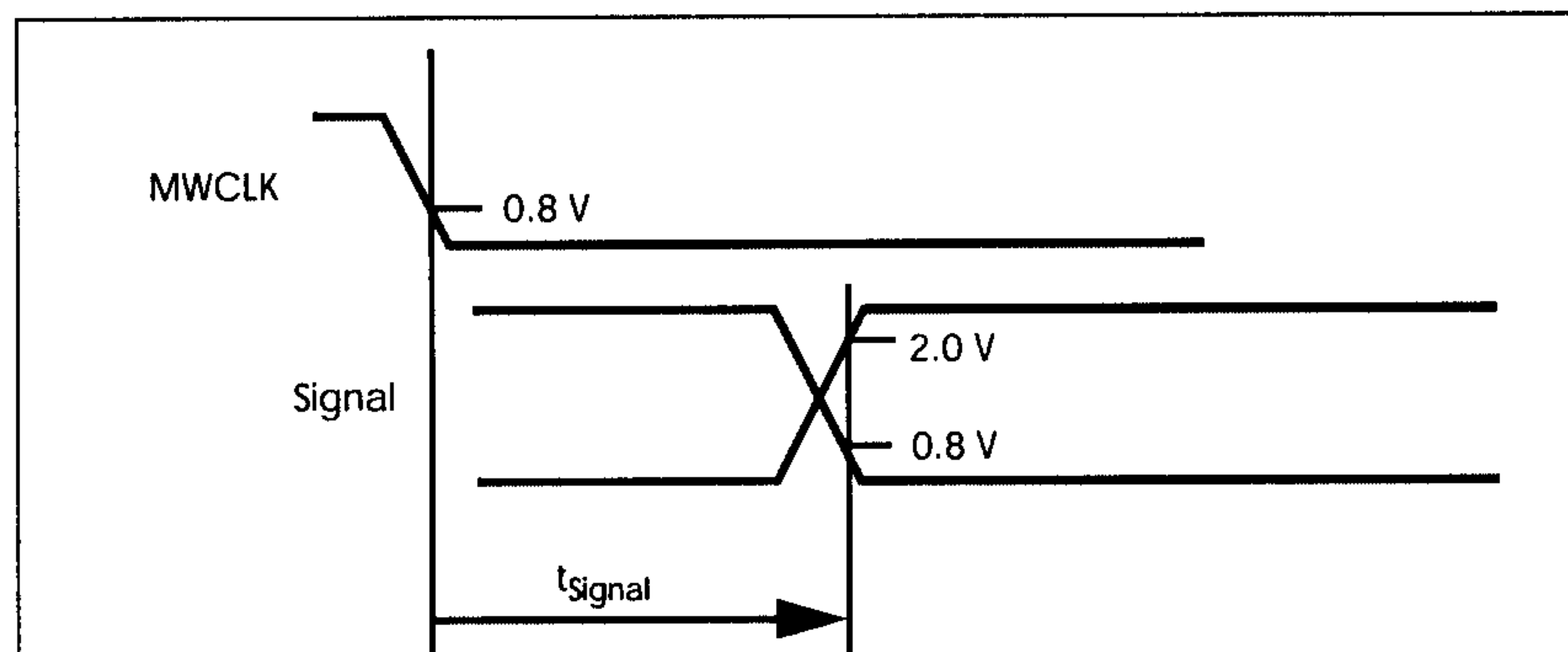
NOTE CCTL is an internal signal and is used as a reference to explain the timing of other signals. See Figure 1-21.

Figure 1-6: Synchronous Output Signals (Valid, Active and Inactive)



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

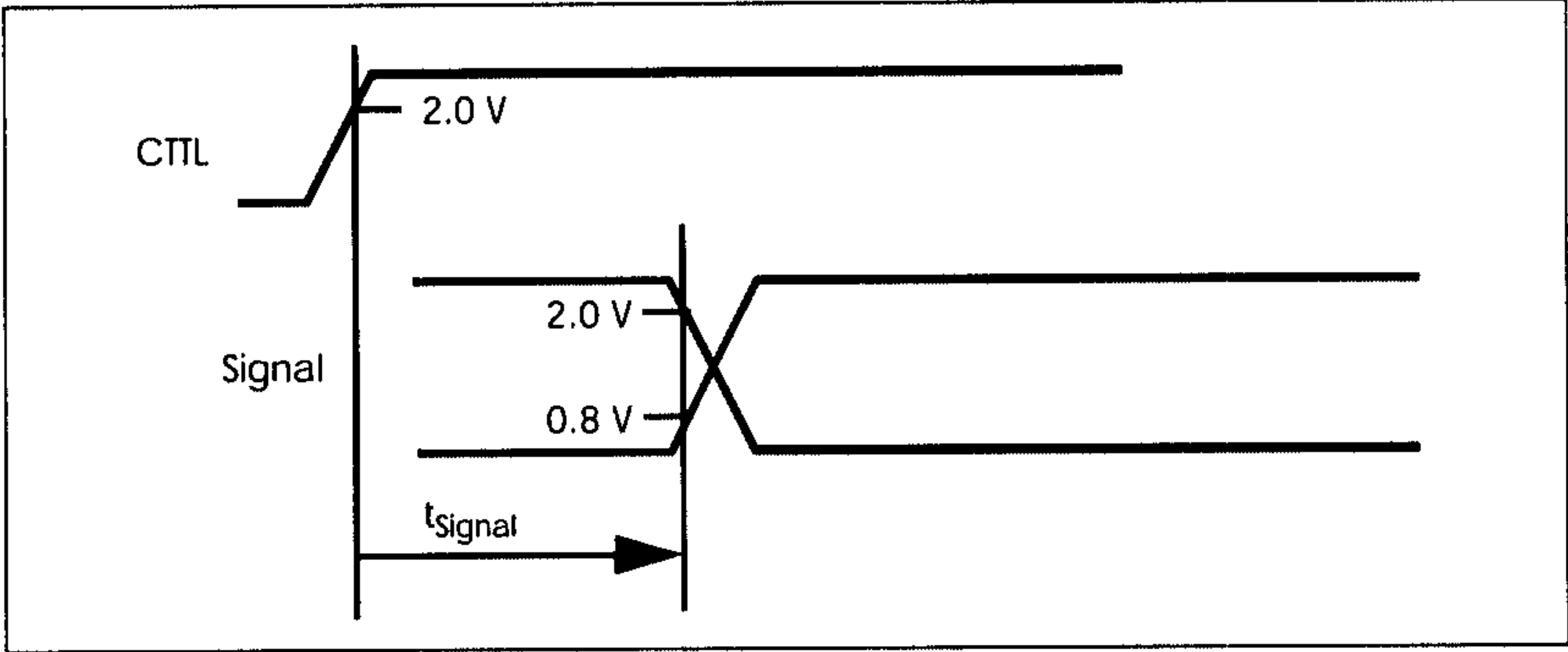
Figure 1-7: Synchronous Output Signals (Valid)



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

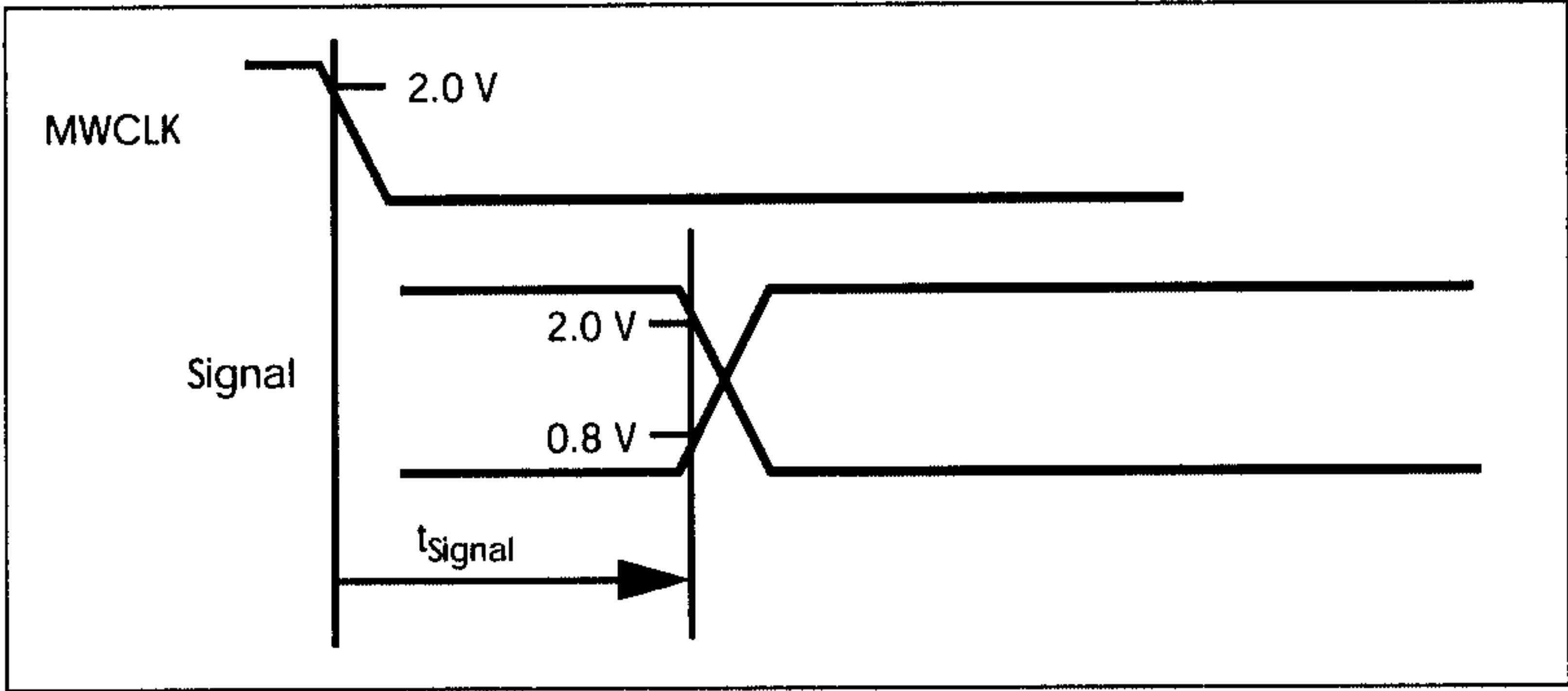
NOTE Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to the conditions specified below.

Figure 1-8: Synchronous Output Signals (Hold), After a Rising Edge of CTTL



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

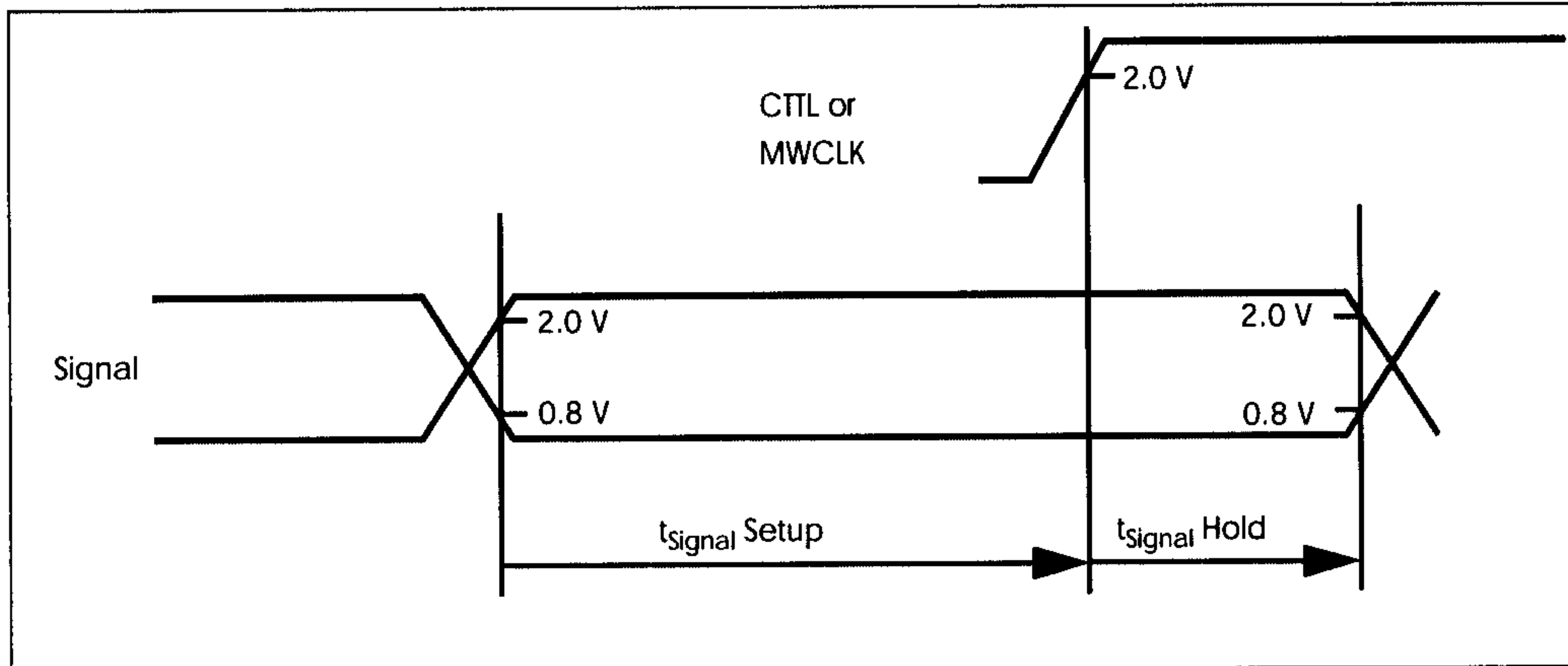
Figure 1-9: Synchronous Output Signals (Hold), After a Falling Edge of MWCLK



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

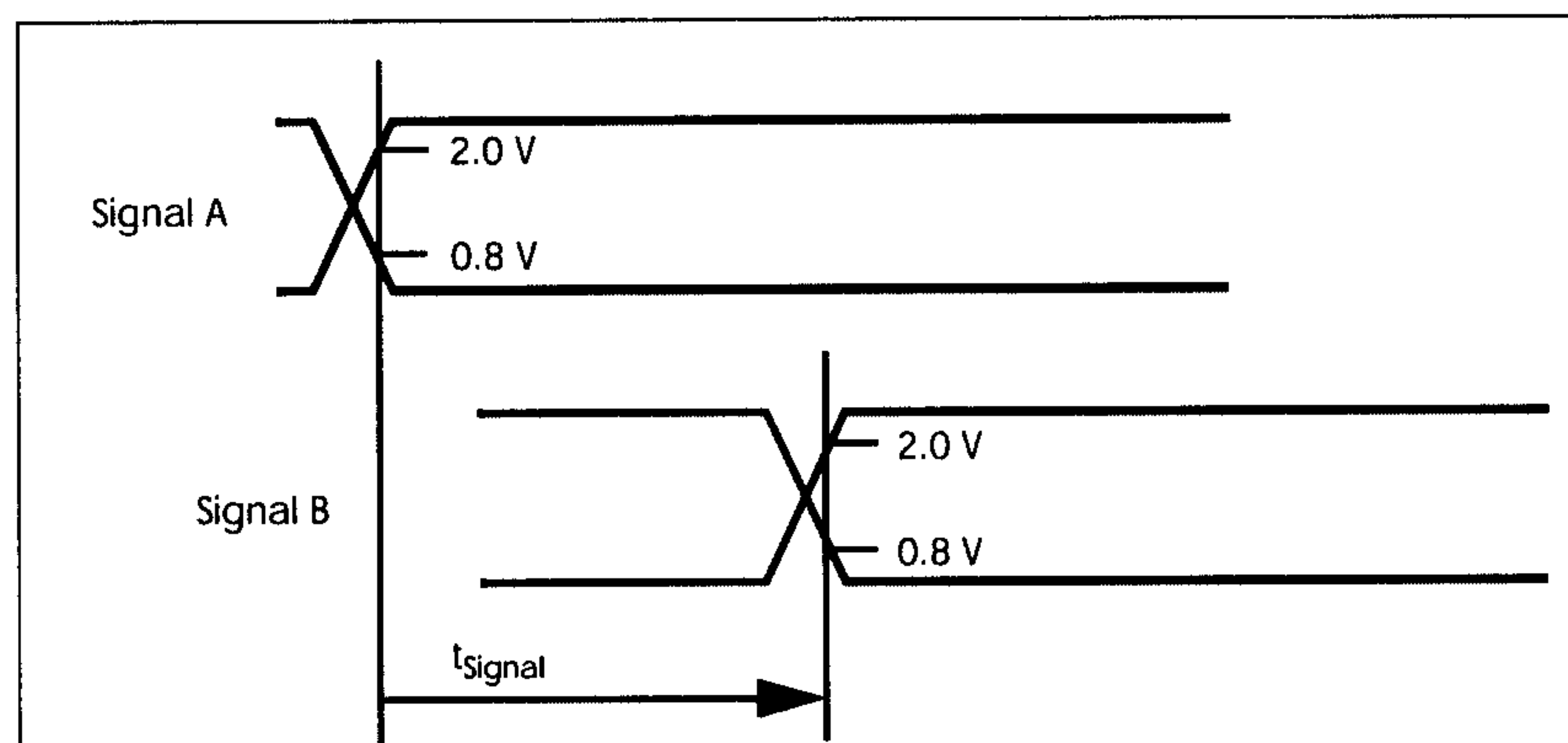
ISD CONFIDENTIAL INFORMATION

Figure 1-10: Synchronous Input Signals



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

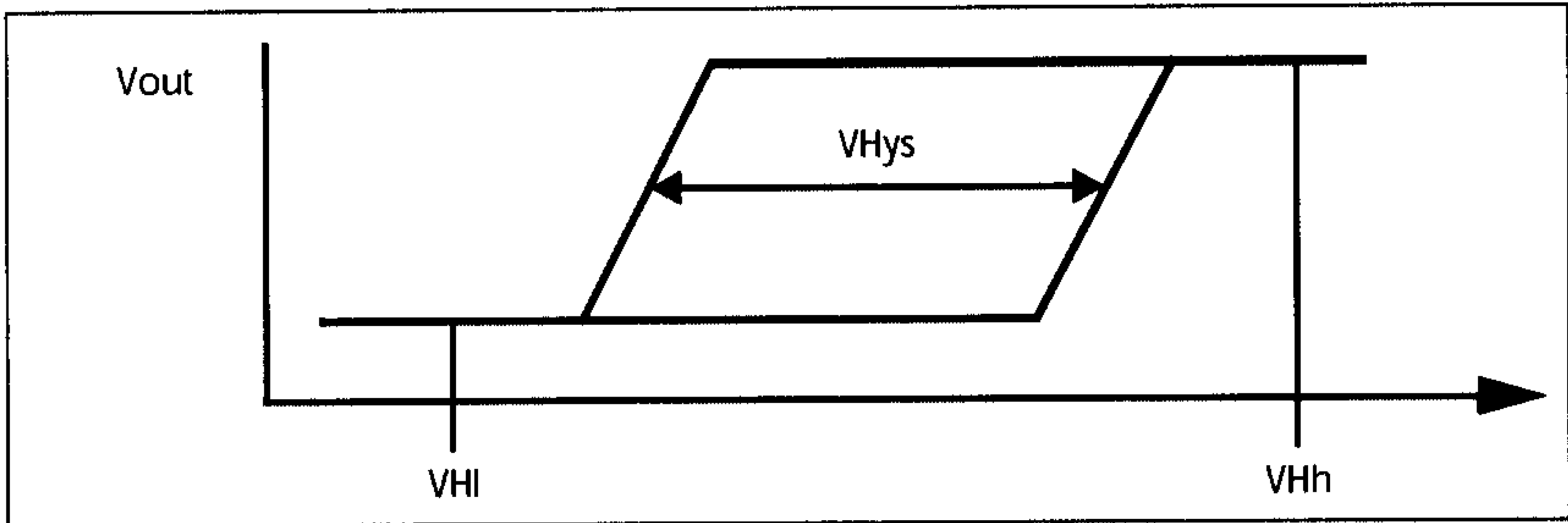
Figure 1-11: Asynchronous Signals



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

The $\overline{\text{RESET}}$ signal has a Schmitt trigger input buffer. Figure 1-12 shows the characteristics of the input buffer.

Figure 1-12: Hysteresis Input Characteristics



SYNCHRONOUS TIMING TABLES

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

Output Signals

Table 1-7: Output Signals

Symbol	Pin	Signal Name	Condition	Min	Max
t_{Ah}	1-20	Address Hold	After R.E. CTIL	0.0	
t_{Av}	1-20	Address Valid	After R.E. CTIL, T1		12.0
t_{CCLKa}	1-11	CCLK Active	After R.E. CTIL		12.0
t_{CCLKh}	1-11	CCLK Hold	After R.E. CTIL	0.0	
t_{CCLKia}	1-11	CCLK Inactive	After R.E. CTIL		12.0
t_{CDOh}	1-11	CDOOUT Hold	After R.E. CTIL	0.0	
t_{CDOv}	1-11	CDOOUT Valid	After R.E. CTIL		12.0
t_{CTp}	1-21	CTIL Clock Period ¹	R.E. CTIL to next R.E. CTIL	25.0	50,000
t_{EMCSa}	1-20	$\overline{EMCS}$ Active	After R.E. CTIL, T2W1		12.0
$t_{EMCS h}$	1-20	$\overline{EMCS}$ Hold	After R.E. CTIL	0.0	
t_{EMCSia}	1-20	$\overline{EMCS}$ Inactive	After R.E. CTIL T3		12.0
t_{FSa}	1-11	CFS0 Active	After R.E. CTIL		25.0
$t_{FS h}$	1-11	CFS0 Hold	After R.E. CTIL	0.0	
t_{FSia}	1-11	CFS0 Inactive	After R.E. CTIL		25.0
t_{MMCLKa}		Master MICROWIRE Clock Active	After R.E. CTIL		12.0

ISD CONFIDENTIAL INFORMATION

Table 1-7: Output Signals (Continued)

Symbol	Pin	Signal Description	Transition	Setup (ns)	Hold (ns)
t_{MMCLKh}		Master MICROWIRE Clock Hold	After R.E. CTIL	0.0	
$t_{MMCLKia}$		Master MICROWIRE Clock Inactive	After R.E. CTIL		12.0
t_{MMDOh}		Master MICROWIRE Data Out Hold	After R.E. CTIL	0.0	
t_{MMDOv}		Master MICROWIRE Data Out Valid	After R.E. CTIL		12.0
t_{MWDOF}	1-4	MICROWIRE Data Float ²	After R.E. $\overline{MWCS}$		70.0
t_{MWDOh}	1-4	MICROWIRE Data Out Hold ²	After F.E. MWCK	0.0	
t_{MWDOvf}	1-4	MICROWIRE Data No Float ²	After F.E. $\overline{MWCS}$	0.0	70.0
t_{MWDOv}	1-4	MICROWIRE Data Out Valid ²	After F.E. MWCK		70.0
t_{MWIOP}	1-13	MWDIN to MWDOUT	Propagation Time		70.0
t_{MWRDYa}	1-4	$\overline{MWRDY}$ Active	After R.E. of CTIL	0.0	35.0
$t_{MWRDYia}$	1-4	$\overline{MWRDY}$ Inactive	After F.E. MWCLK	0.0	70.0
t_{PABCh}	1-14	PB and $\overline{MWRQST}$	After R.E. CTIL	0.0	
t_{PABCV}	1-14	PB and $\overline{MWRQST}$	After R.E. CTIL, T2W1		12.0

1. In normal operation mode t_{Ctp} must be 25.0ns; in power-down mode, t_{Ctp} must be 50,000 ns.
 2. Guaranteed by design, but not fully tested.

Input Signals

Table 1-8: Input Signals

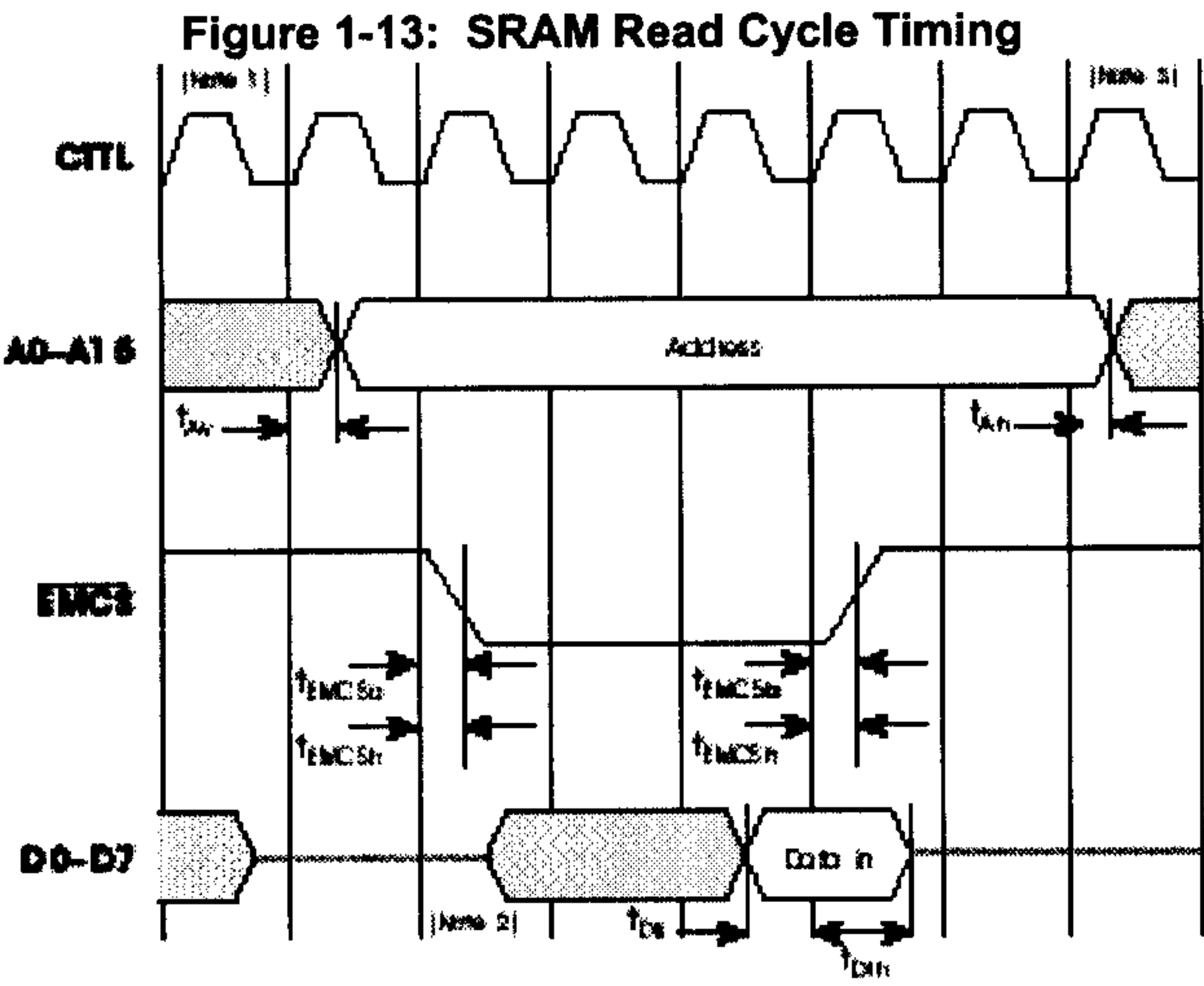
Symbol	Pin	Signal Description	Transition	Setup (ns)
t_{CDih}	1-11	CDIN Hold	After R.E. CTIL	0.0
t_{CDIs}	1-11	CDIN Setup	Before R.E. CTIL	11.0
t_{DIh}		Data in Hold (D0:7)	After R.E. CTIL T1, T3 or TI	0.0
t_{DIIs}		Data in Setup (D0:7)	Before R.E. CTIL T1, T3 or TI	15.0
t_{MMDINh}		Master MICROWIRE Data In Hold	After R.E. CTIL	0.0
t_{MMDINs}		Master MICROWIRE Data In Setup	Before R.E. CTIL	11.0
t_{MWCKh}	1-4	MICROWIRE Clock High (slave)	At 2.0 V (both edges)	100.0
t_{MWCKl}	1-4	MICROWIRE Clock Low (slave)	At 0.8 V (both edges)	100.0
t_{MWCKp}	1-4	MICROWIRE Clock Period (slave) ¹	R.E. MWCLK to next R.E. MWCLK	2.5 μ s
t_{MWCLKh}	1-4	MWCLK Hold	After $\overline{MWCS}$ becomes inactive	50.0
t_{MWCLKs}	1-4	MWCLK Setup	Before $\overline{MWCS}$ becomes active	100.0

Table 1-8: Input Signals (Continued)

Symbol	Pin	Signal	Condition	Value
t_{MWCSH}	1-4	MWCS Hold	After F.E. MWCLK	50.0
t_{MWCSs}	1-4	MWCS Setup	Before R.E. MWCLK	100.0
t_{MWDih}	1-4	MWDIN Hold	After R.E. MWCLK	50.0
t_{MWDIs}	1-4	MWDIN Setup	Before R.E. MWCLK	100.0
t_{PWR}	1-23	Power Stable to $\overline{RESET}$ R.E. ²	After V_{CC} reaches 4.5 V	30.0 ms
t_{RSTW}	1-23	$\overline{RESET}$ Pulse Width	At 0.8 V (both edges)	10.0 ms
t_{Xh}	1-21	CLKIN High	At 2.0 V (both edges)	$t_{X1p}/2 - 5$
t_{Xl}	1-21	CLKIN Low	At 0.8 V (both edges)	$t_{X1p}/2 - 5$
t_{Xp}	1-21	CLKIN Clock Period	R.E. CLKIN to next R.E. CLKIN	24.4

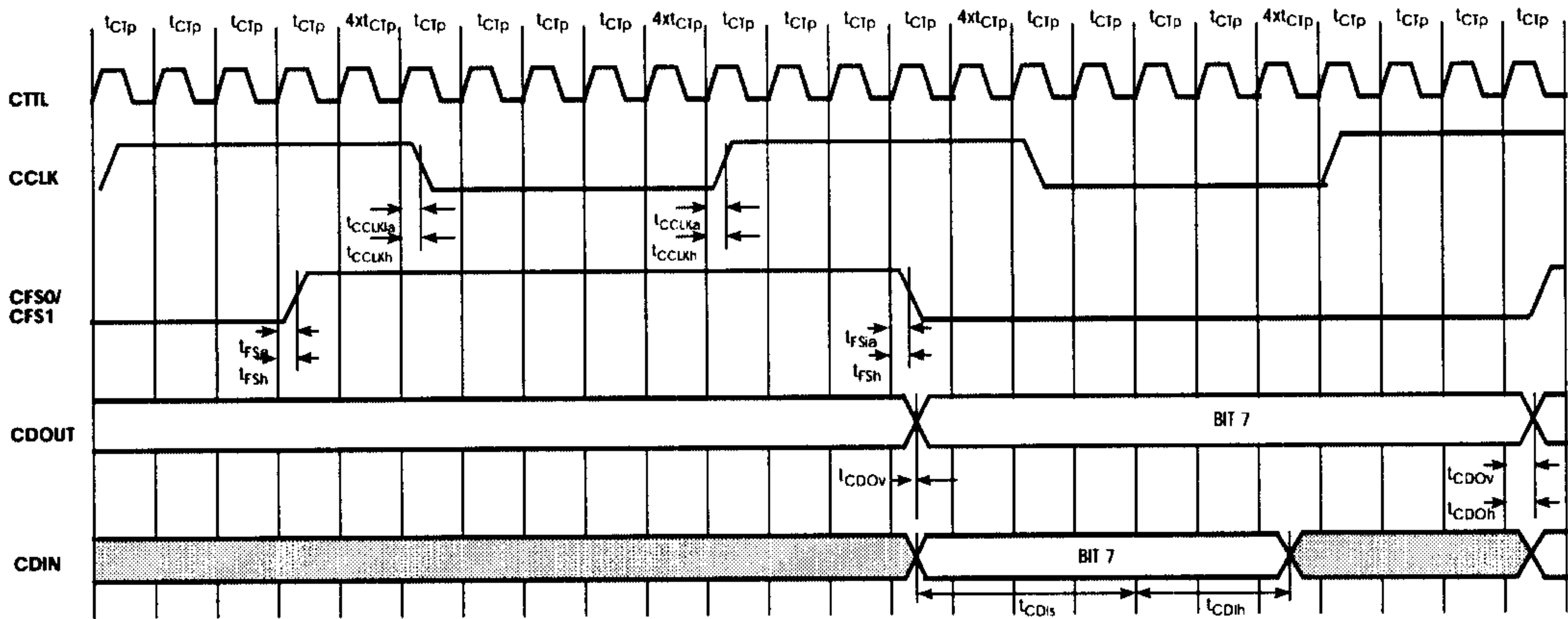
1. Guaranteed by design, but not fully tested in power-down mode.
2. Guaranteed by design, but not fully tested.

TIMING DIAGRAMS



ISD CONFIDENTIAL INFORMATION

Figure 1-14: CODEC Short Frame Timing



NOTE: This cycle may be either T1 (Idle), T2, T3 or T3H.

Figure 1-15: CODEC Long Frame Timing

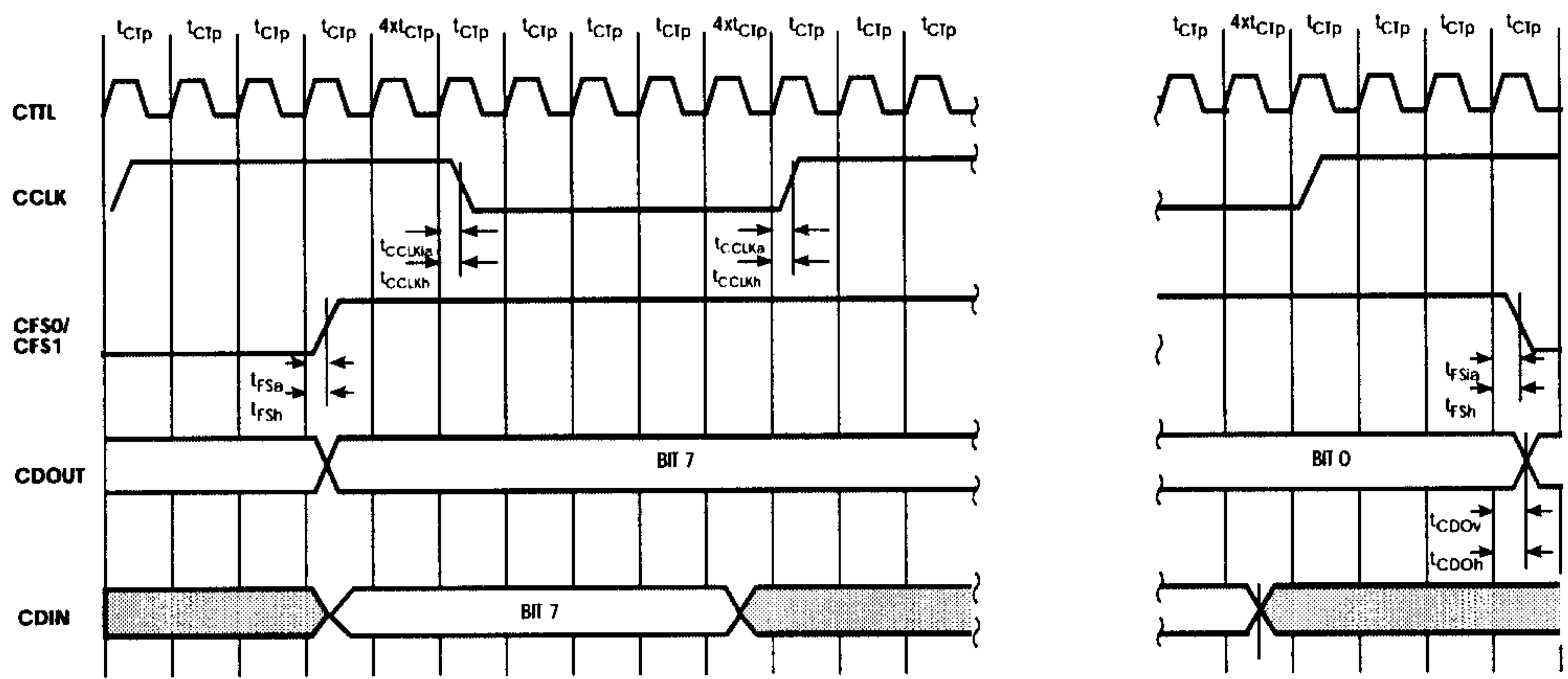


Figure 1-16: Slave CODEC CCLK and CFSO Timing

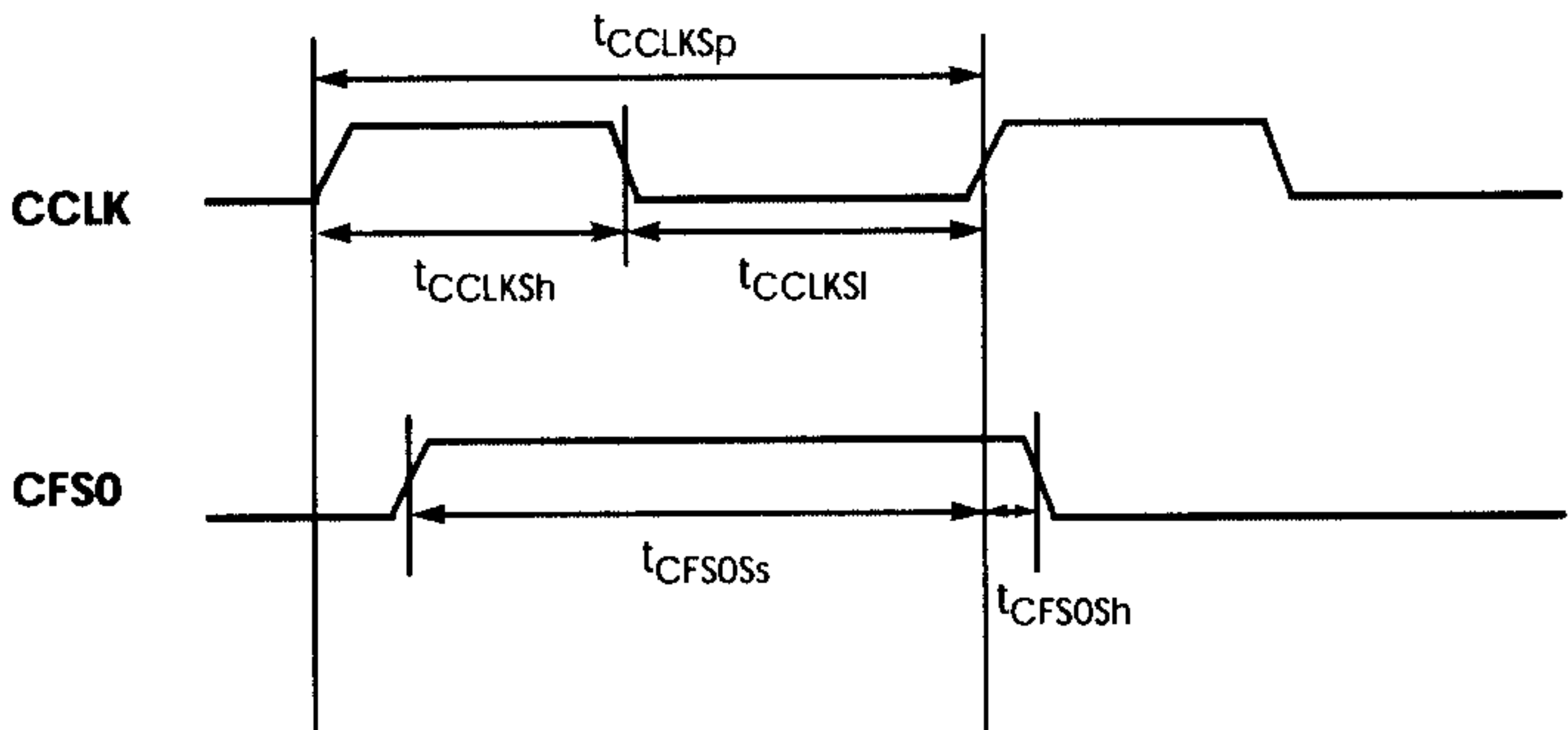
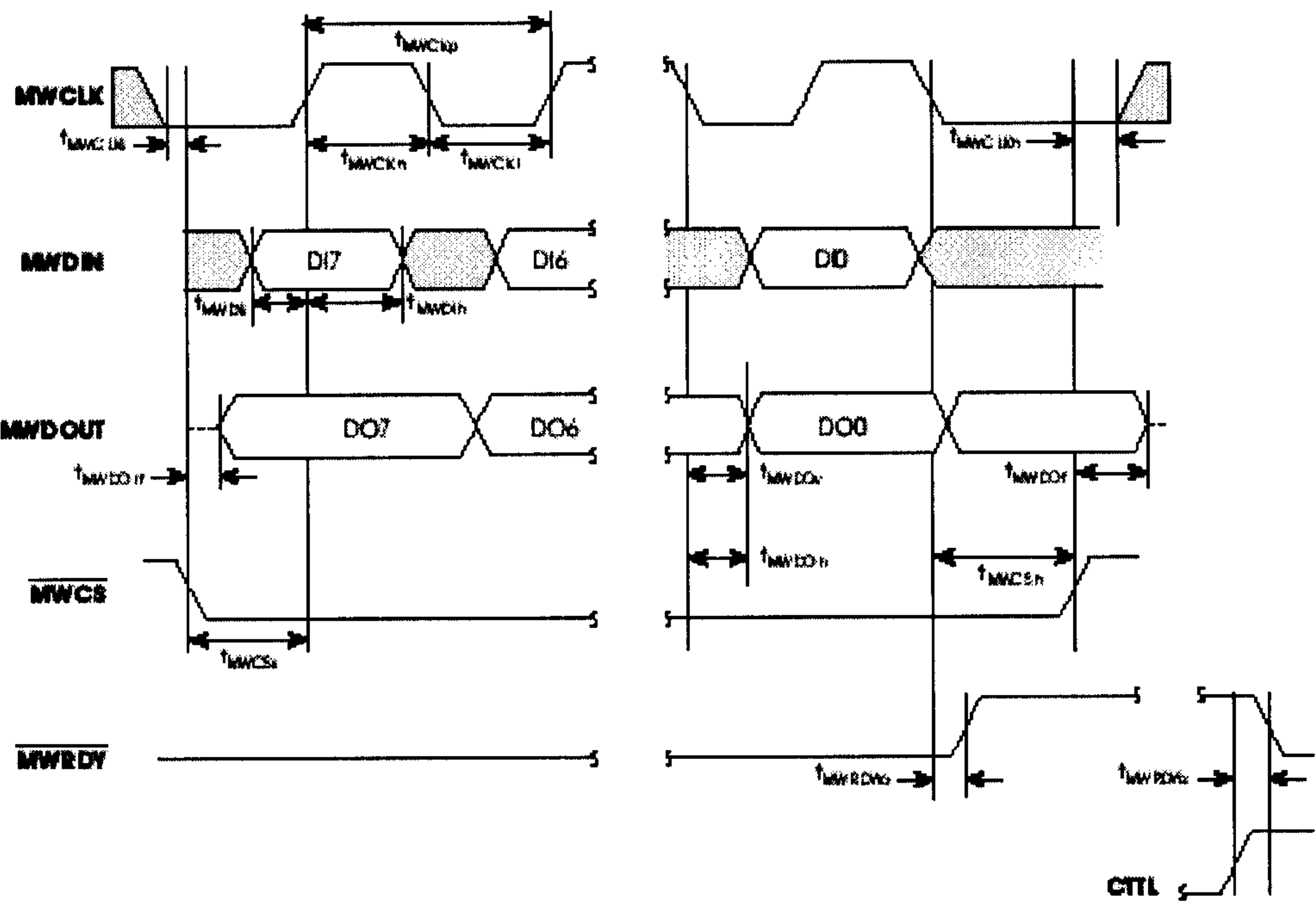


Figure 1-17: MICROWIRE Transaction Timing--Data Transmitted to Output



ISD CONFIDENTIAL INFORMATION

Figure 1-18: MICROWIRE Transaction Timing--Data Echoed to Output

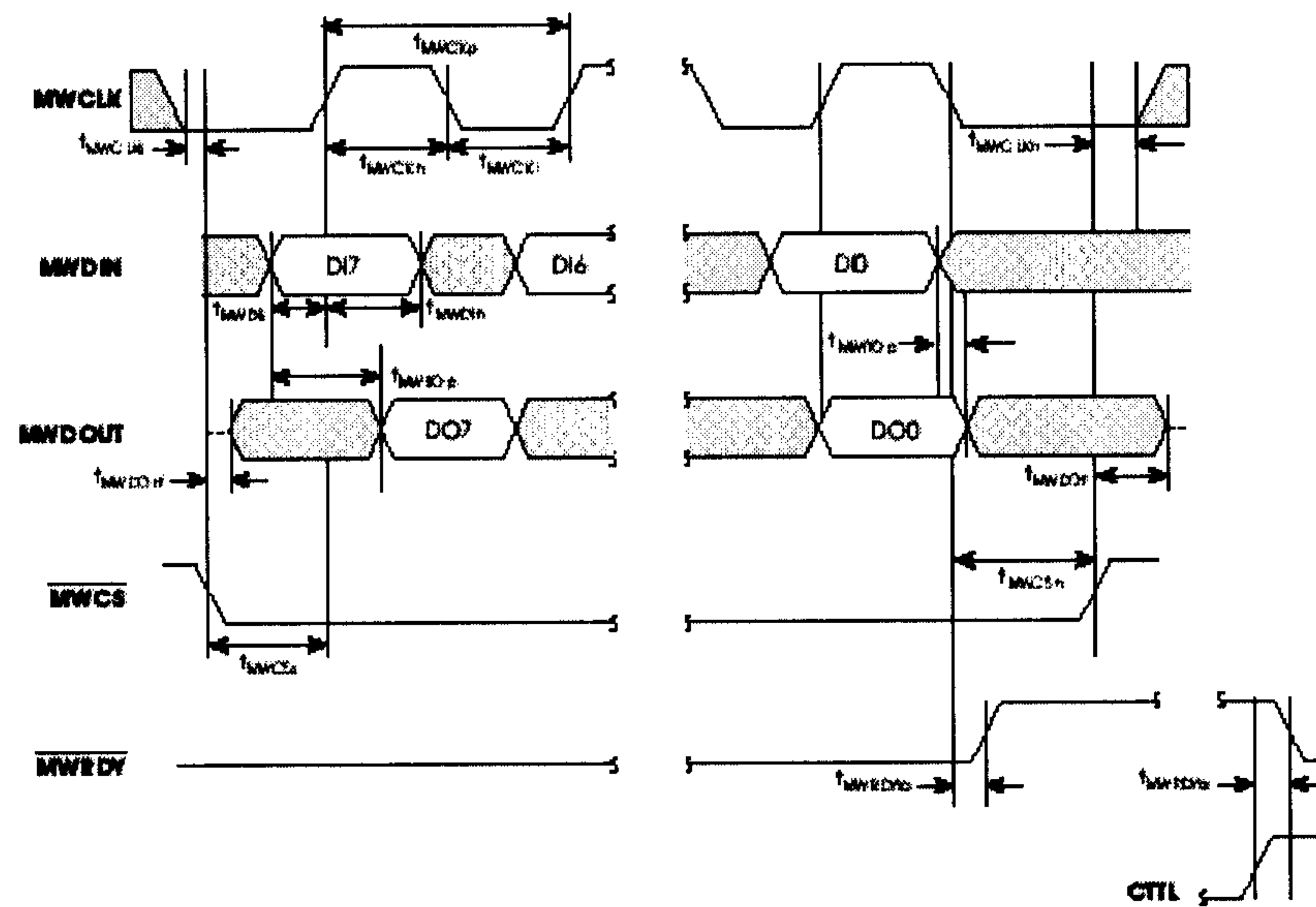


Figure 1-19: Master MICROWIRE Timing

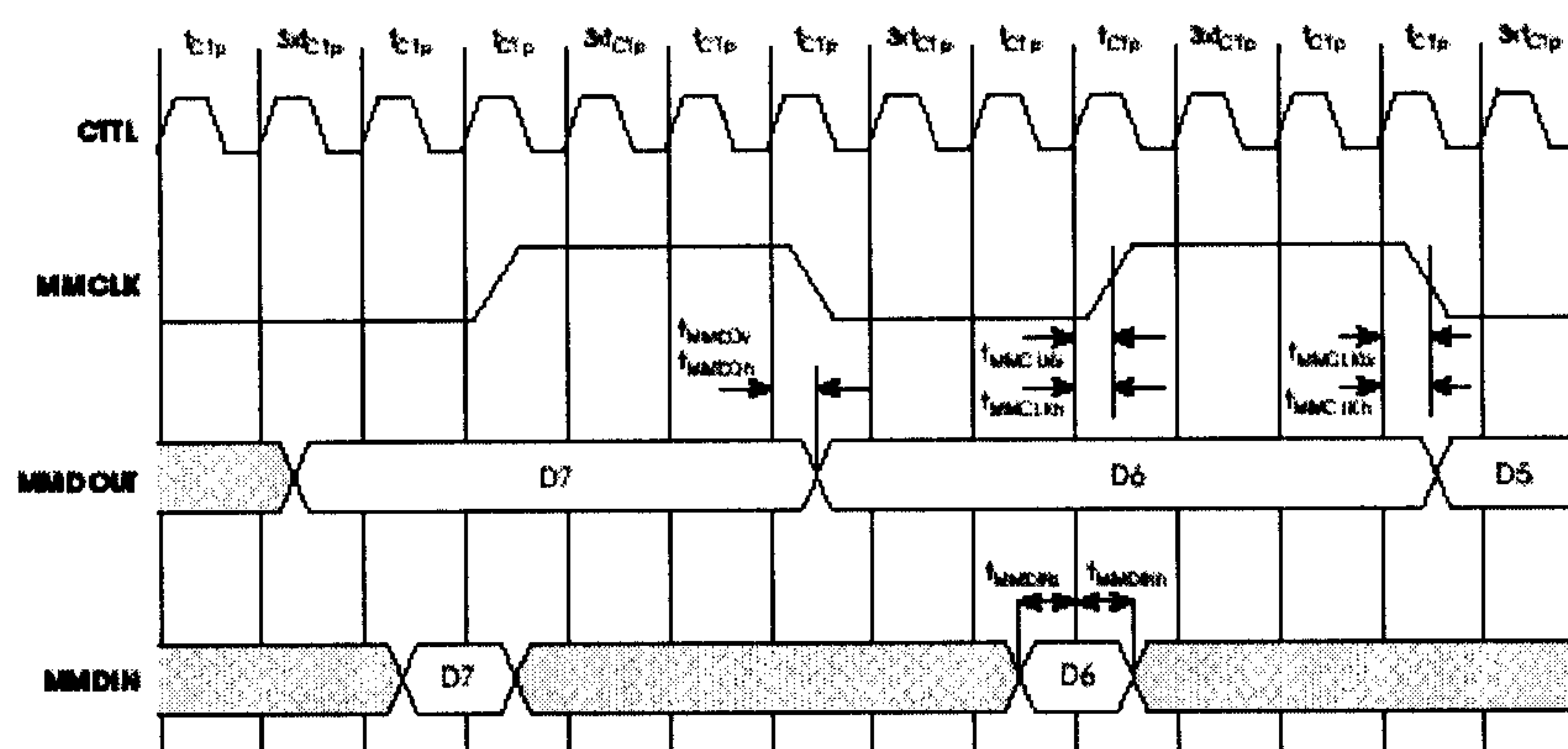
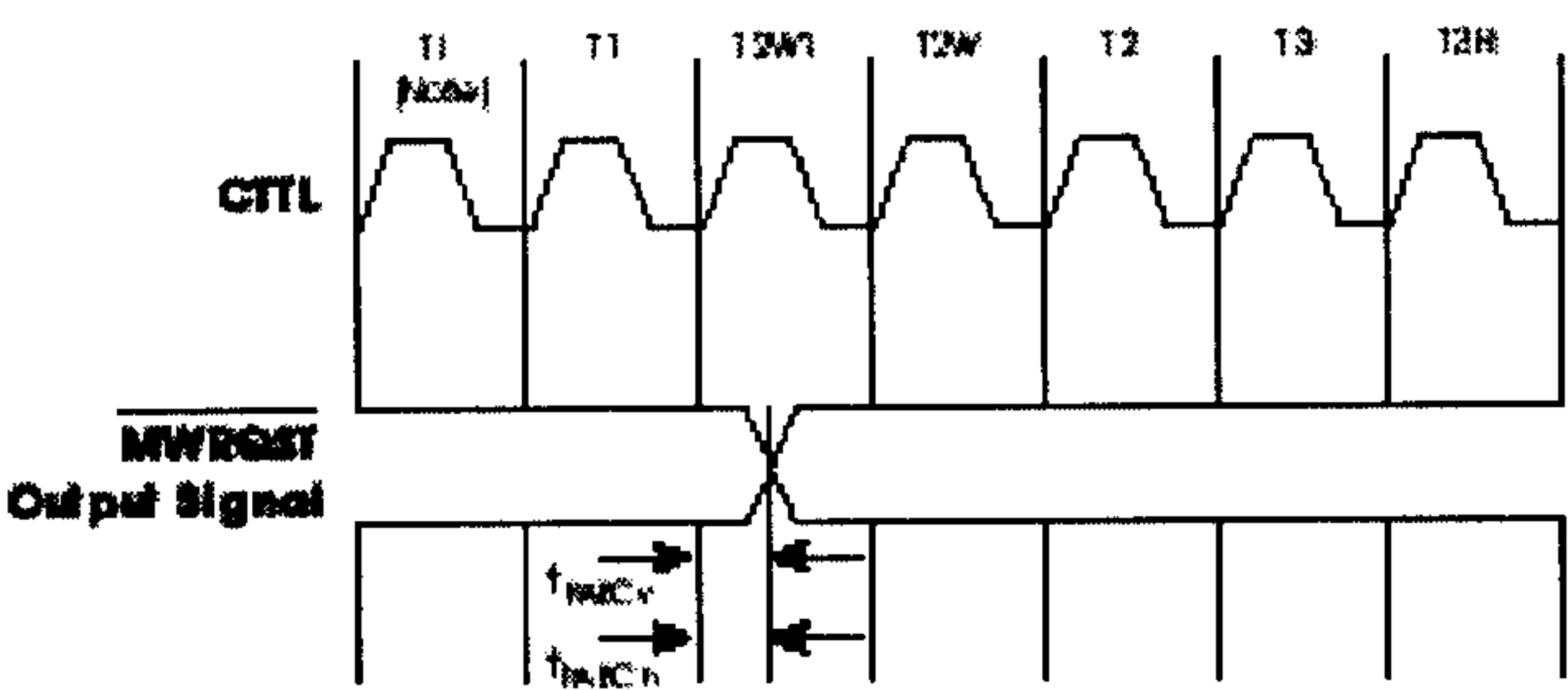
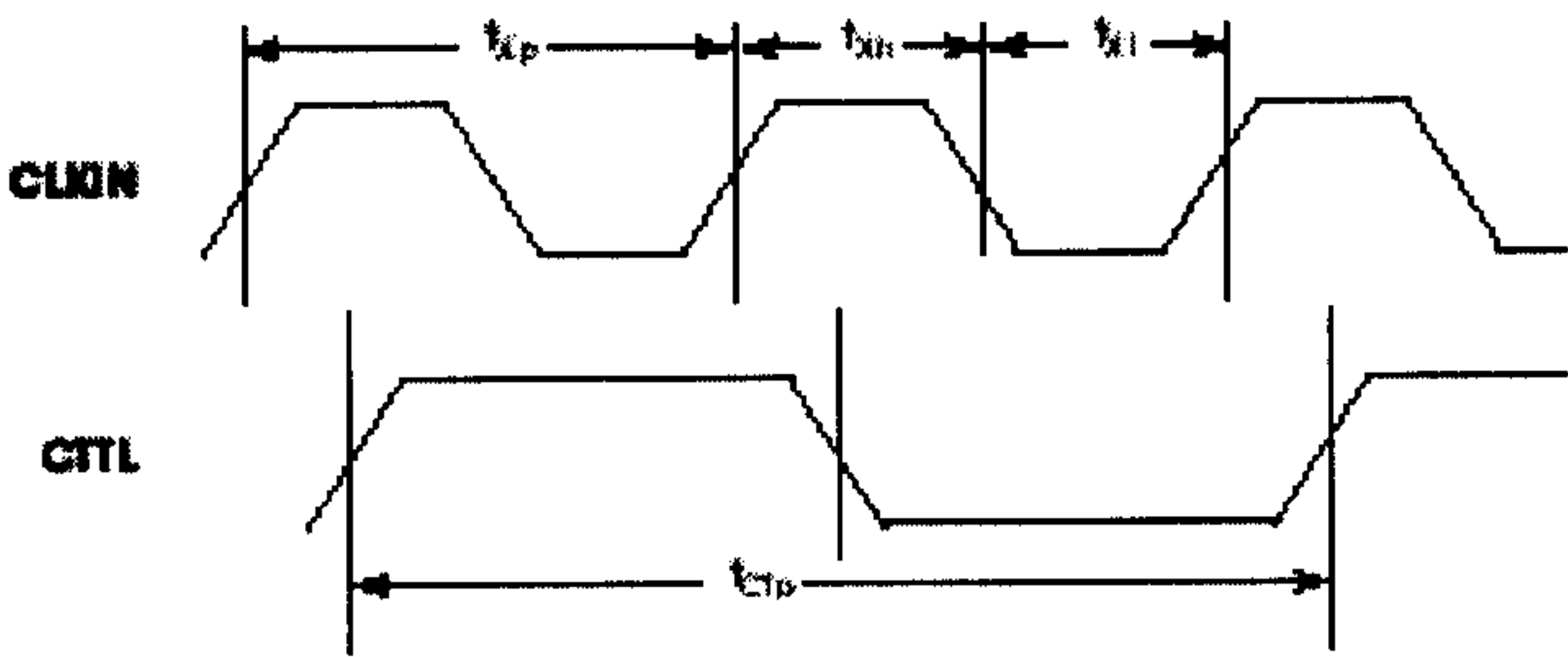


Figure 1-20: Output Signal Timing for Port PB and $\overline{\text{MWRQST}}$



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

Figure 1-21: CTTL and CLKIN Timing



ISD CONFIDENTIAL INFORMATION

Figure 1-22: Reset Timing When Reset is not at Power-Up

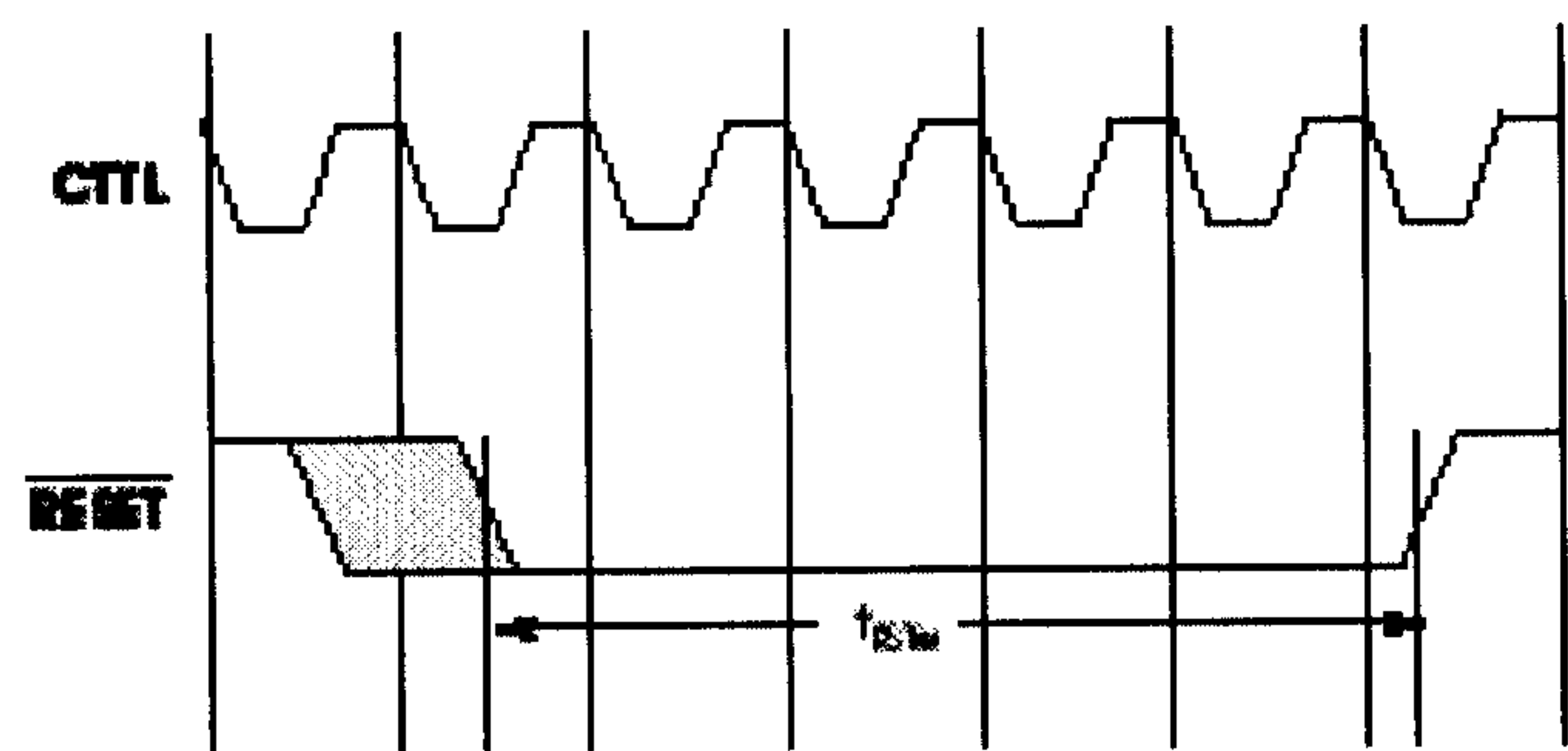
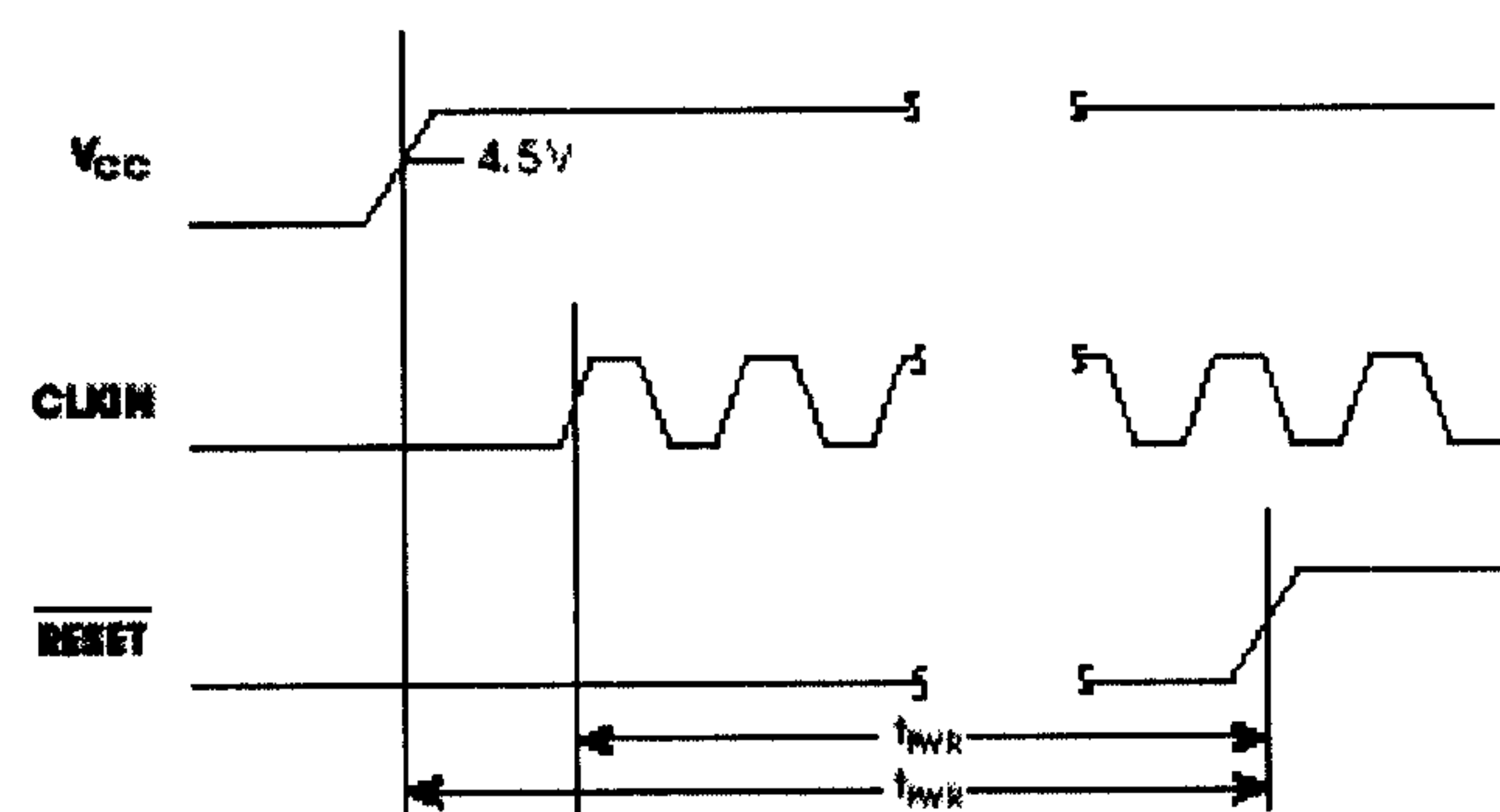


Figure 1-23: Reset Timing When Reset Is at Power-Up



Chapter 2 SOFTWARE

OVERVIEW

The ISD-SR3000 software resides in the on-chip ROM. It includes voice recognition algorithms, system support functions and a software interface to hardware peripherals. The following sections describe the ISD-SR3000 software in detail.

RECOGNITION PROCESSOR

ISD-SR3000 uses a segmented sub-phoneme recognition process. The sampled speech utterance is split into distinct phonetic sounds, the smallest units of speech. Because these phonemes vary in both sound and duration, the processor must be able to determine boundaries between the sounds. The ISD-SR3000 uses Hidden Markov Models to hypothesize boundaries between sounds and to form probabilistic models on each possible combination.

The outputs are then classified by determining matches between the phonetic sounds and the stored phoneme models. The acoustic models for the phonemes are gathered from a large sample of speakers, allowing for a wide variation across accents, dialect, and gender. This allows the recognizer to associate the sound segments with a number of possible phonemes, enabling recognition when words are pronounced differently.

The phonemes are then matched to vocabulary words or phrases using a search routine. The set of phonemes is compared to the vocabulary models for the active topics, and the recognized word is returned. If the phonemes do not match any of the active vocabulary words, a token is returned indicating the word is not in the vocabulary. This token can be used by the Voice User Interface to return a help prompt to the user. The ISD-SR3000 does not return a score with the word; like a digital system, it either recognizes a word, or it doesn't.

TYPES OF RECOGNITION

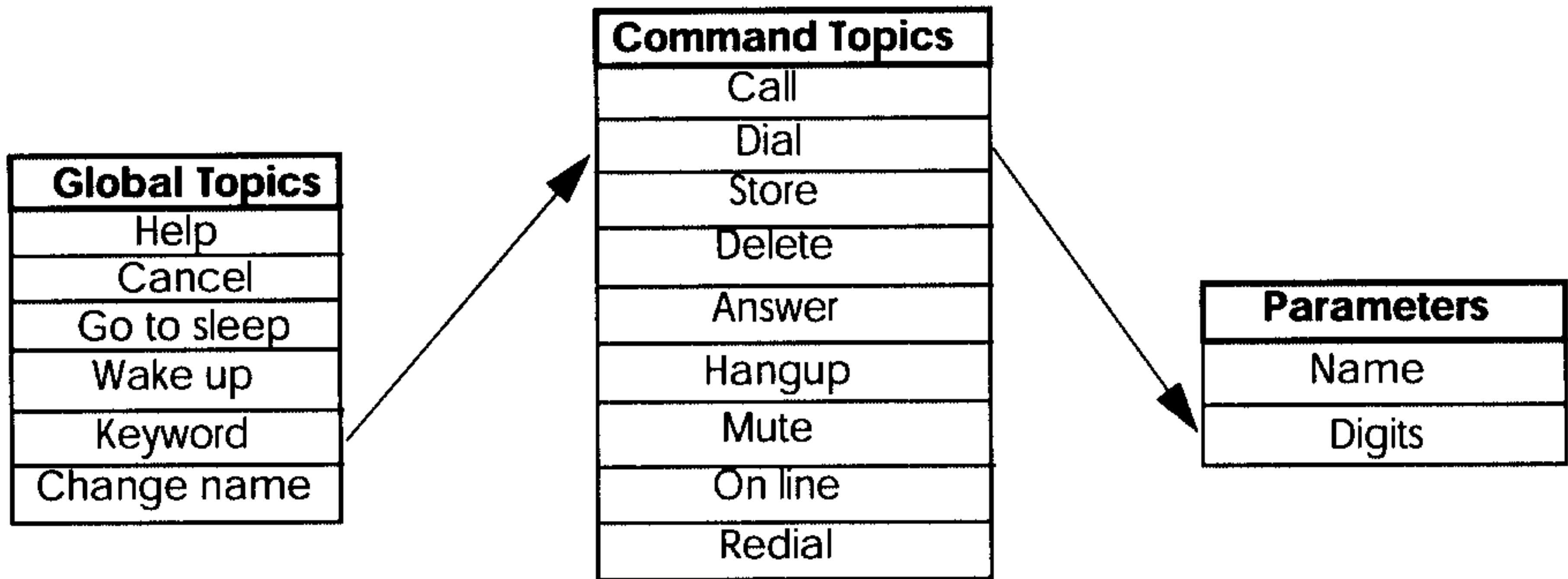
The ISD-SR3000 is capable of both speaker-independent and speaker dependent recognition. Speech input pattern is continuous for both commands and digits, allowing for a natural speech pattern. The commands and digits are speaker-independent, with models constructed from a large corpus of speakers. The user-provided voicetags for the phone book are partially speaker-dependent. However, they are constructed by creating acoustic models "on-the-fly" from the phoneme base. This means only two passes are required for entering the names, and recognition is possible with some variation in the way the name is spoken. The first pass is used to create the phoneme model, and the second pass is used for recognition confirmation.

GRAMMAR

Grammar is used to define the structure of the commands. ISD-SR3000 is designed to work with finite-state grammar. This type of grammar is designed to limit perplexity by pre-defining the number of allowable words at a given state. Perplexity is defined as the number of branches possible during recognition. For example, a prompt that requires a "yes" or "no" response has a perplexity of two. Greater perplexities increase the chances for substitution errors. During recognition, a limited number of topics are active. Topics are groups of words that are active at a given time. For example, in a voice dialing application, digit topics are active after the user issues the "dial" command. No other topics are open (except the global topics such as "cancel" or "help") so that the recognizer is only trying to recognize digits. This type of grammar and active topics inherently increases recognition accuracy.

ISD CONFIDENTIAL INFORMATION

Figure 2-1: Topic and Grammar Organization



From this example, it can be seen how topics are linked, and how only specific topics are active. This is a voice dialing command set furnished as ISD's standard VUI. Independent VUIs and vocabulary can be developed, but it is necessary to follow the grammar syntax as shown here.

VOCABULARY

A vocabulary defines the following characteristics of the ISD-SR3000:

- Speaker-independent command words and digits for which ISD-SR3000 responds
- Topics under which the commands and digits are organized
- Mapping of tokens to the vocabulary
- Strings returned by the CTW command
- Default keywords used for activation

ISD-SR3000 is designed to work with a specific vocabulary set. Up to 30 speaker-independent commands may be used. When the processor recognizes the commands, tokens (values) are returned to the host controller. Certain types of errors, such as spoke-too-soon, also return specific tokens. The host controller can use the tokens to accomplish tasks, such as generating DTMF for dialing a phone number. For detailed information

on the tokens, refer to the ISD-SR3000 Voice User Interface specification.

ISD supplies recommended vocabulary sets as part of the VUI for specific applications. The vocabulary sets have been carefully selected to ensure high recognition (avoiding confusable words) and effective user utility. The accuracy specifications for ISD-SR3000 are based on the ISD provided commands. Although use of these commands is highly recommended, it is possible to create custom vocabulary sets. Contact ISD for information about vocabulary development tools. The vocabulary can be stored either in external ROM or flash memory.

LANGUAGE

ISD-SR3000 uses a set of acoustic models designed to recognize American English. Additional languages require different acoustic models. Contact ISD for availability of additional languages.

SPEECH SYNTHESIS

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

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

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

INTERNATIONAL VOCABULARY SUPPORT (IVS)

IVS is a mechanism by which the ISD-SR3000 processor can use several vocabularies stored on an external storage device. IVS enables ISD-SR3000 processor to synthesize messages with the same meaning, but in different languages, from separate vocabularies.

Among IVS features:

- Multiple vocabularies are stored on a single storage device.
- Plug-and-play. The same microcontroller code is used for all languages.
- Synthesized and recorded messages use the same voice compression algorithm to achieve equal quality.
- Support for voicetag recognition confirmation:
 - Calling *name*.
 - Are you sure you want to delete *name*?
 - The number for *name* has been stored.

VOCABULARY DESIGN

There are several issues, sometimes conflicting, which must be addressed when designing an IVS-vocabulary.

Vocabulary Content

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

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

Vocabulary Recording

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

Another issue to consider is the difference in voice quality between synthesized and recorded prompts and voicetags. It is more pleasant to the human ear to hear them both in the same quality.

Vocabulary Access

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

When designing for a multi-lingual environment, there are more issues to consider. Each vocabulary should be able to handle language-specific structures and designed in a cooperative way with the other vocabularies so that the code to access each vocabulary is the same.

IVS VOCABULARY COMPONENTS

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

ISD CONFIDENTIAL INFORMATION

Basic Concepts

An IVS vocabulary consists of words, sentences, and special codes that control the behavior of the algorithm which the ISD-SR3000 processor uses to synthesize sentences.

Word Table

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

Number Tables

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

Sentence Table

The sentence table describes the predefined sentences in the vocabulary. The purpose of this table is to make the microcontroller that drives the ISD-SR3000 processor independent of the language being synthesized.

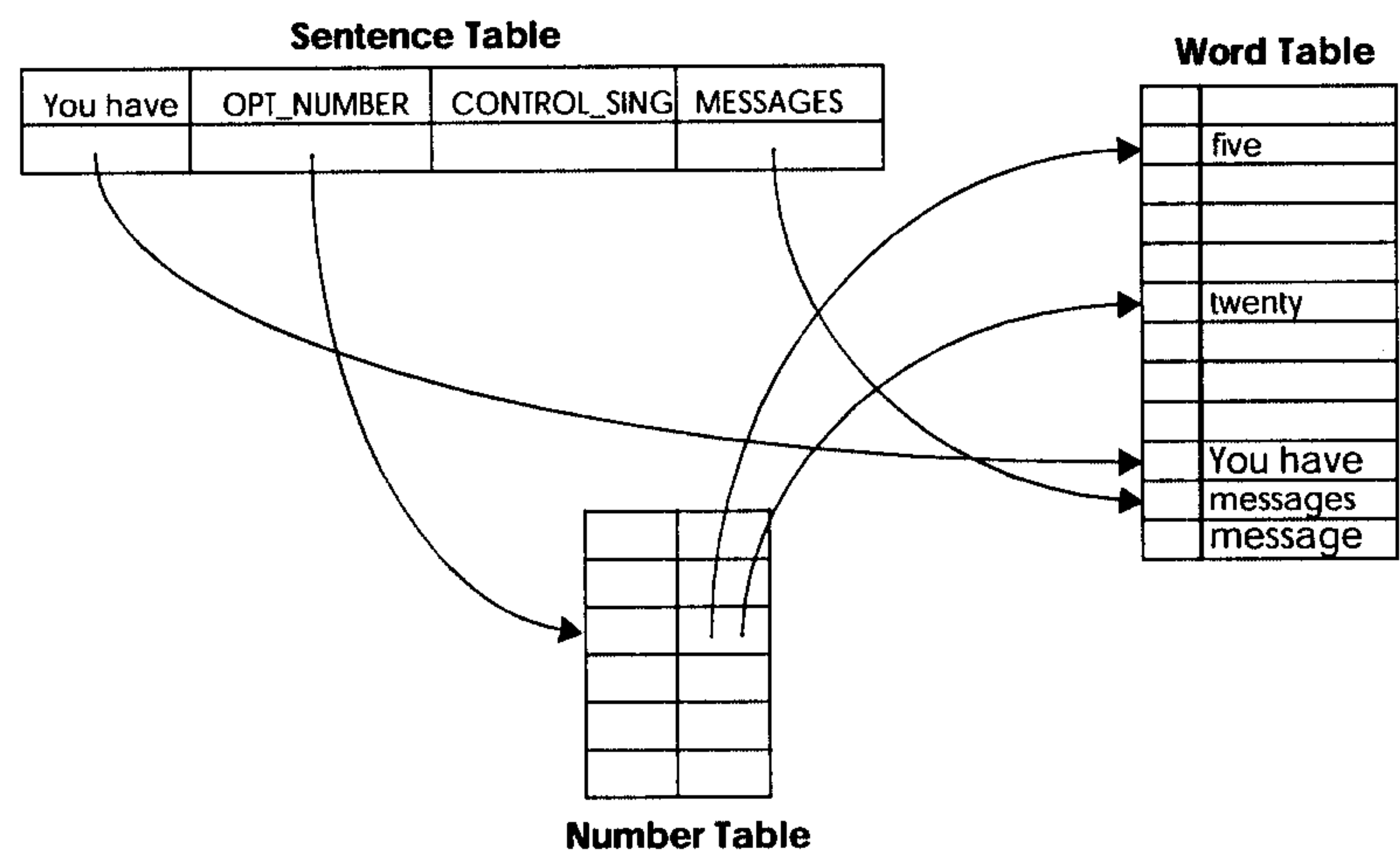
For example, if the Flash and/or ROM contain vocabularies in various languages, and the first sentence in each vocabulary means you have n messages, the microcontroller switches languages by issuing the following command to ISD-SR3000 processor:

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

The microcontroller software is thus independent of the grammar of the language in use.

The sentences consist of words, which are represented by their indices in the vocabulary.

Figure 2-2: The Interrelationship of a Word Table, a Sentence Table and a Number Table



Control and Option Codes

The list of word indices alone cannot provide the entire range of sentences that the ISD-SR3000 processor can synthesize. IVS control and option codes are used as special instructions that control the behavior of the speech synthesis algorithm in the ISD-SR3000 processor.

For example, if the sentence should announce the time of day, the ISD-SR3000 processor should be able to substitute the current day and time in the sentence. These control words do not represent recorded words, rather they instruct the ISD-SR3000 processor to take special actions.

THE IVS TOOL

The IVS tool includes two utilities:

The DOS-based IVS Compiler

IVSTOOL for Windows. A Windows 3.1 based utility

The tools allow you to create vocabularies for the ISD-SR3000 processor. They take you all the way from designing the vocabulary structure, through defining the vocabulary sentences, and recording the vocabulary words.

IVS Compiler

The IVS compiler runs on MS-DOS (version 5.0 or later). It allows you to insert your own vocabulary, i.e., basic words and data used to create numbers and sentences, as directories and files in MS-DOS.

The IVS compiler then outputs a binary file containing that vocabulary. This information can be burned into an EPROM or Flash memory for use by the ISD-SR3000 software.

Voice Compression

Each IVS vocabulary can be compiled with either the 5.2 Kbit/s or the 7.3 Kbit/s voice compression algorithm. You define the compression rate before compilation. The ISD-SR3000 processor automatically selects the required voice decompression algorithm when the SV command is used to select the active vocabulary.

Graphical User Interface (GUI)

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

HOW TO USE THE IVS TOOL WITH THE ISD-SR3000 PROCESSOR

The IVS tool creates IVS vocabularies, and stores them as a binary file. This file is burnt into a ROM device or programmed into a Flash memory device using the INJ command. The ISD-SR3000 processor SV command is used to select the required vocabulary. The SW, SO, SS and SAS commands are used to synthesize the required word or sentence. The typical vocabulary-creation process is as follows:

Design the Vocabulary.

Create the vocabulary files (as described in detail below). Use IVSTOOL for Windows 3.1 to simplify this process.

Record the words using any standard PC sound card and sound editing software, that can create .wav files.

Run the IVS compiler to compress the .wav files, and compile them and the vocabulary tables into an IVS vocabulary file.

Repeat steps 1 to 4 to create a separate IVS vocabulary for each language that you want to use.

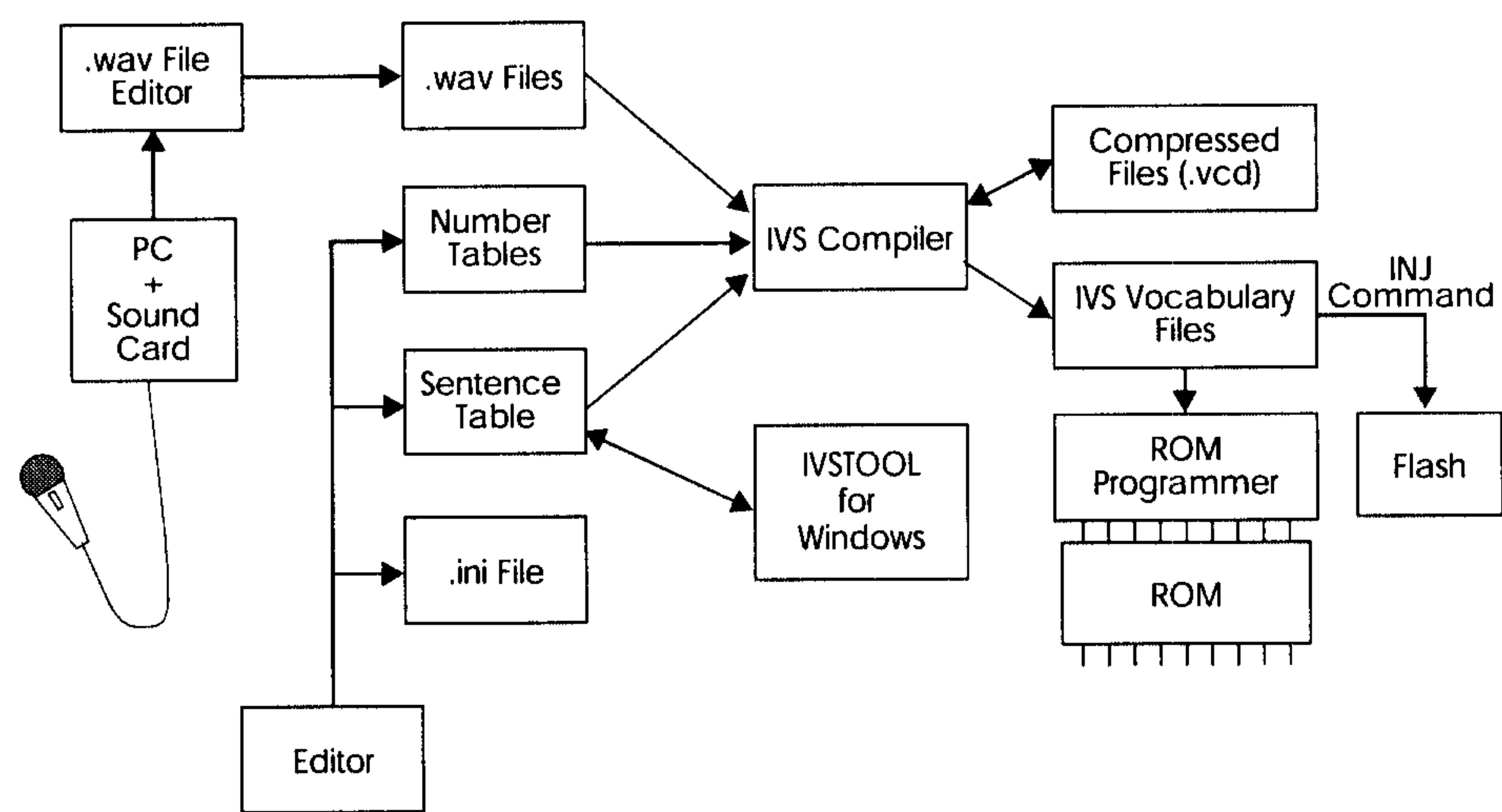
Burn the IVS vocabulary files into a ROM or Flash memory device. Use the INJ (Inject IVS) command to program the data into a Serial Flash device.

Once the vocabulary is in place, the speech synthesis commands of the ISD-SR3000 processor can be used to synthesize sentences.

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

ISD CONFIDENTIAL INFORMATION

Figure 2-3: IVS Components



INITIALIZATION

Use the following procedures to initialize the ISD-SR3000 processor:

NORMAL INITIALIZATION

Reset the ISD-SR3000 processor by activating the RESET signal. (See "RESETTING" on page 1-6.)

Issue a CFG (Configure ISD-SR3000) command to change the configuration according to your environment.

Issue an INIT (Initialize System) command to initialize the ISD-SR3000 firmwareISD-SR3000.

MICROWIRE Serial Interface

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

The ISD-SR3000 MICROWIRE interface implements the MICROWIRE/PLUS interface in slave mode, with an additional ready signal. It enables a microcontroller to interface efficiently with the ISD-SR3000 processor application.

The microcontroller is the protocol master and provides the clock for the protocol. The ISD-SR3000 processor supports clock rates of up to 400 KHz. This transfer rate refers to the bit transfer; the actual throughput is slower due to byte processing by the ISD-SR3000 processor and the microcontroller.

Communication is handled in bursts of eight bits (one byte). In each burst the ISD-SR3000 processor is able to receive and transmit eight bits of data. After eight bits have been transferred, an internal interrupt is issued for the ISD-SR3000 processor to process the byte, or to prepare another byte for sending. In parallel, the ISD-SR3000 processor sets $\overline{\text{MWRDY}}$ to 1, to signal the microcontroller that it is busy with the byte processing. Another byte can be transferred only when the $\overline{\text{MWRDY}}$ signal is cleared to 0 by the ISD-SR3000 processor. When the ISD-SR3000 processor transmits data, it expects to receive the value 0xAA before each transmitted byte. The ISD-SR3000 processor re-

ports any status change by clearing the $\overline{\text{MWRQST}}$ signal to 0.

If processor command's parameter is larger than one byte, the microcontroller transmits the Most Significant Byte (MSB) first. If a return value is larger than one byte, the ISD-SR3000 processor transmits the MSB first.

INTERFACE PROTOCOL TIME-OUTS

Depending on the ISD-SR3000 processor's state, if more than 100 milliseconds elapse between the assertion of the $\overline{\text{MWRDY}}$ signal and the transmission 8th bit of the next byte pertaining to the same command transaction, a time-out event occurs, and the ISD-SR3000 processor responds as follows:

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

ISD CONFIDENTIAL INFORMATION

MICROWIRE ECHO MECHANISM

The ISD-SR3000 echoes back to the microcontroller all the bits received by the ISD-SR3000. Upon detection of an error in the echo, the microcontroller should stop the protocol clock, which eventually causes a time-out error (i.e., ERR_TIMEOUT bit is set in the error word).

When a command has a return value, the ISD-SR3000 transmits bytes of the return value instead of the echo value.

The ISD-SR3000 transmits a byte as an echo when it receives the value 0xAA from the microprocessor. Upon detection of an error the ISD-SR3000 activates the MWRQST signal, and sets the ERR_COMM bit in the error word.

Master Mode

The ISD-SR3000 Master MICROWIRE controller implements the MICROWIRE/PLUS interface in master mode, thus enabling the processor to control the Flash memory devices. Several devices may share the Master MICROWIRE channel by connecting device selection signals to general purpose output ports.

Signals

The Master MICROWIRE controller's signals are the Master MICROWIRE Serial Clock (MMCLK), the Master MICROWIRE Serial Data Out (MMDOUT) signal, and the Master MICROWIRE Serial Data In (MMDIN) signal.

The Master MICROWIRE controller can handle up to four Flash memory devices. The processor uses the signals, CS0-CS3, relative to the number of devices used, as device chip-select signals.

Clock for Master MICROWIRE Data Transfer

Before data can be sent, the transfer rate must be determined and set. The MMCLK signal sets the data transfer rate on the Master MICROWIRE. This rate is the same as the CODEC Clock (CCLK) signal. As long as the Master MICROWIRE is transferring data, the CODEC interface must be enabled and its sampling rate should not be changed.

ROM Interface

IVS Vocabularies can be stored in either Flash memory and/or ROM. The ISD-SR3000 supports IVS ROM devices through an Expansion Memory mechanism. Up to 64 Kbytes (64Kx8) of Expansion Memory are directly supported. Nevertheless, the processor uses bits of the on-chip port (PB) to further extend the 64 Kbytes address space up to 0.5 Mbytes address space.

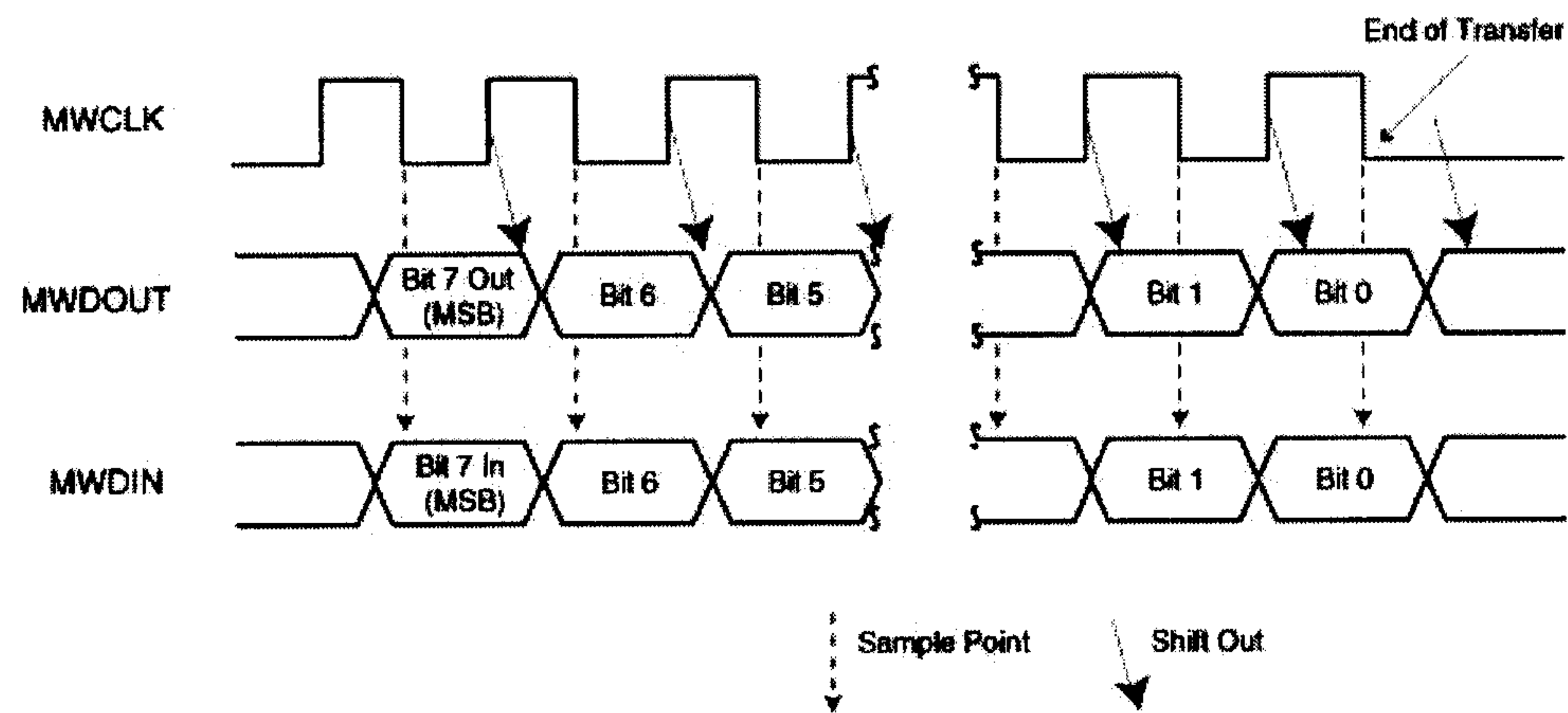
ROM is connected to the ISD-SR3000 using the data bus, D(0:7), the address bus, A(0:15), the extended address signals, EA(16:18), and Expansion Memory Chip Select, EMCS, controls. The number of extended address pins to use may vary, depending on the size and configuration of the ROM. ISD-SR3000 configured with Samsung Flash memory can not support extension ROM.

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. EMCS is asserted (cleared to 0) on a T2W1 cycle. This cycle is followed by three T2W cycles and one T2 cycle. The ISD-SR3000 processor samples data at the end of the T2 cycle.

The transaction is terminated at T3, when EMCS becomes inactive (set to 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.

Figure 2-4: Master MircroWIRE Data Transfer



ISD CONFIDENTIAL INFORMATION

Table 2-1: ISD-SR3000 Command Summary

Command	Mode	Description	Index	State	Reply	Command Parameters	Length	Reply Format	Reply Value
CCIO	S	Configure CODEC I/O	34	RESET, IDLE	No Change	config_value	1	None	—
CFG	S	Config ISD-SR3000	01	RESET	No Change	config_value	2	None	—
CKF	S	Check Flash	TBD	IDLE	No Change	None	—	TEST RESULT	1
CKV	S	Check Voicetags	TBD	IDLE	No Change	None	—	TEST RESULT	1
CTW	S	Convert Token to Word String	TBD	IDLE, RECO	No Change	token_number	2	STRING	VARIES
EVA	S	Erase All Voicetags	TBD	IDLE	No Change	user_numbers	1	None	—
EVS	S	Erase Selected Voicetags	TBD	IDLE	No Change	voicetag_number	1	None	—
FRQ	S	Flush Recognition Queue	TBD	IDLE	No Change	None	—	None	—
GCFG	S	Get Configuration Value	02	RESET, IDLE	No Change	None	—	Version, Config, Vocab	3
GEW	S	Get Error Word	1B	All States	No Change	None	—	Error Word	2
GI	S	Get Information Item	25	IDLE, PLAY, RECORD, SYNTHESIS	No Change	Item	1	Value	2
GNR	S	Get Next Recognition	TBD	IDLE, RECO	No Change	None	—	Count, Topic, Token	4
GSW	S	Get Status Word	14	All States	No Change	None	—	Status Word	2
INIT	S	Initialize System	13	RESET, IDLE	No Change	None	—	None	—
INJ	S	Inject IVS Data	29	RESET, IDLE	No Change	n, byte ₁ , byte _n ...	4 + n	None	—
KWNR	S	Keyword Not Required	TBD	IDLE, RECO	No Change	None	—	None	—
KWR	S	Keyword Required	TBD	IDLE, RECO	No Change	None	—	None	—
PDM	S	Go to Power-Down Mode	1A	IDLE	No Change	None	—	None	—
PRD	S	Disable Pause Reporting	TBD	IDLE, RECO	No Change	None	—	None	—

-125-

ISD CONFIDENTIAL INFORMATION

PRE	S	Enable Pause Reporting	TBD	IDLE, RECO	No Change	None	—	None	—
PV	A	Play Voicetag	TBD	IDLE	Play	voicetag_number	1	None	—
RES	S	Resume	1D	IDLE, PLAY, RECORD, SYNTHESIS	No change	None	—	None	—
RESK	S	Reset Keyword	TBD	IDLE	No change	user_numbers	1	None	—
RESR	S	Reset Recognition Engine	TBD	IDLE, RECO	IDLE	None	—	None	—
RD	S	Disable Recognition	TBD	RECO	IDLE	None	—	None	—
RE	A	Enable Recognition	TBD	IDLE	RECO	None	—	None	—
RKW	A	Record Keyword	TBD	IDLE	Record	user_id	1	None	—
RTAG	A	Record Voicetag	TBD	IDLE	Record	user_id, voicetag_number	1 + 1	None	—
SAS	A	Say Augmented Sentence	1E	IDLE	Synthesis	sentence_n, arg	1 + 1	None	—
SDET	S	Set Detectors Mask	10	IDLE	No change	detectors_mask	1	None	—
SO	A	Say One Word	07	IDLE	Synthesis	word_number	1	None	—
SS	A	Say Sentence	1F	IDLE	Synthesis	sentence_n	1	None	—
SUSP	S	Suspend	1C	IDLE, PLAY, RECORD, SYNTHESIS	No change	None	—	None	—
SW	A	Say Words	21	IDLE	Synthesis	n, word ₁ , word _n ...	1 + n	None	—
TAGC	S	Change Voicetag	TBD	IDLE	No change	user_id_info, voicetag_number	1 + 1	None	—

ISD CONFIDENTIAL INFORMATION

Command	Response	ISD	ISD	ISD	ISD	ISD	ISD	ISD	ISD
Command	Response	ISD	ISD	ISD	ISD	ISD	ISD	ISD	ISD
TACQ	S	Query Voicetag	TBD	IDLE	No change	voicetag_number	1	Voicetag info	1
TOPD	S	Disable Topics	TBD	IDLE	No change	topic_id	1	None	—
TOPE	S	Enable Topics	TBD	IDLE	No change	topic_id	1	None	—
TOPQ	S	Query Topics	TBD	IDLE	No change	topic_id	1	Topic info	1
VC	S	Volume Control	28	IDLE, PLAY, SYNTHESIS	No change	vol_level	1	None	—

NOTE: Note For the column labeled S/A, S = Synchronous command and A = Asynchronous command.

THE STATE MACHINE

The ISD-SR3000 processor functions as a state machine. It changes state either in response to a command sent by the host microcontroller, after execution of a command is completed, or as a result of an internal event (e.g. memory full or power failure). The ISD-SR3000 processor states are listed below.

RESET

The ISD-SR3000 processor is initialized to this state after a full hardware reset by the RESET signal.

IDLE

This is the state from which most commands are executed. As soon as a command and all its parameters are received, the ISD-SR3000 processor starts executing the command.

PLAY

In this state, a prompt is played.

SYNTHESIS

An individual word or sentence is synthesized from a vocabulary.

RECORD

In this state, a user's speech is recorded and stored.

RECO

In this state, speech recognition is active.

COMMAND EXECUTION

An ISD-SR3000 command is represented by an 8-bit opcode. Some commands have parameters, and some commands return values to the host microcontroller. Commands are either synchronous or asynchronous.

SYNCHRONOUS COMMANDS

A synchronous command must complete execution before the host microcontroller can send a new command.

A synchronous command sequence starts when the host microcontroller sends an 8-bit opcode to the ISD-SR3000 processor, followed by the command's parameters (if any).

The ISD-SR3000 processor executes the command and, if required, transmits a return value to the host microcontroller. Upon completion, the ISD-SR3000 processor notifies the host microcontroller that it is ready to accept a new command by asserting the MWRQST signal.

ASYNCHRONOUS COMMANDS

An asynchronous command runs in the background. During execution of an asynchronous command, other commands can be executed.

Status Word

The 16-bit Status Word indicates events that occur during normal operation. The ISD-SR3000 processor asserts the MWRQST signal to indicate a change in the Status Word. This signal remains asserted until the ISD-SR3000 processor receives a GSW command. The status word is cleared during reset, and upon successful execution of the GSW command.

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 MWRQST signal is asserted.

Error Handling

When the host microcontroller detects that the MWRQST signal has been asserted, the host should issue the GSW (Get Status Word) command, which de-asserts the MWRQST signal. Then the host should test the EV_ERROR bit in the data (the Status Word contents) returned by the GSW command. If the EV_ERROR bit is set, the host should issue the GEW (Get Error Word) command to read the Error Word for details of the error.

ISD CONFIDENTIAL INFORMATION

COMMAND DESCRIPTIONS

Commands are listed in alphabetical order, with their hex value in brackets after their mnemonic name. All command opcodes are one byte in length.

All opcodes, parameters and examples are shown using hex values for 8 bit and larger quantities, and binary values for bit values, unless otherwise noted.

Each command description includes an example application of the command. The examples show the opcode issued by the microcontroller, and the response returned by the ISD-SR3000 processor. For commands which require a return value from the ISD-SR3000 processor, the start of the return value is indicated by a thick vertical line. When a return value is required, the host microcontroller must pass value AA (hex) to the ISD-SR3000 engine as a placeholder for each byte to be returned.

CCIO (34 hex)

Configure Codec I/O *config_value*

Configures the voice sample paths in various states. It should be used to change the default ISD-SR3000 processor configuration.

The *config_value* parameter is a byte in size and is encoded as follows:

Bit 0-Loopback control.

- 0

Loopback disabled (default).
- 1

Loopback enabled. In the RECORD state, the input samples are echoed back unchanged (i.e., no volume control) to the codec. This is useful for debugging the analog and codec circuitry.

Bits 1-7-Reserved

- 0

These bits must be set to 0.

Example:

CCIO 01			
Byte Sequence:	Microcontroller	34	01
	ISD-SR3000	34	01
Description:	Configure the codec to have loopback on.		

CFG (01 hex)

Configure ISD-SR3000 *config_value*

Configures the ISD-SR3000 processor for various hardware environments. It should be used to change the default ISD-SR3000 processor configuration.

The *config_value* parameter is a 16-bit word and is encoded as follows:

Bit 0-Codec configuration.

- 0

Short-frame format (default).
- 1

Long-frame format. (Guaranteed by design, but not tested.)

ISD CONFIDENTIAL INFORMATION

Bit 1			Bits 10-8	
	0	This bit must be set to 0.		Number of installed Flash devices. The default is 001 for one flash device
Bit 2		Reserved-Do not use	000	Reserved - do not use.
			001	One flash device installed
Bit 3			010	Two flash devices installed
	0	This bit must be set to 0.	011	Three flash devices installed
Bits 5-4			100	Four flash devices installed
	00	Reserved - do not use.	101	Reserved - do not use.
	01	Reserved - do not use.	110	Reserved - do not use.
	10	Toshiba's TC58A040F Flash	111	Reserved - do not use.
	11	Samsung's KM29N040T Flash		
The default value is 10 for Toshiba's Serial Flash.			Bits 15-11	
Bits 7-6			00000	These bits must be set to 0.
	00	These bits must be set to 0.		

Example:

CFG 0124				
Byte Sequence:	Microcontroller	01	01	24
	ISD-SR3000	01	01	24
Description:	Configure the ISD-SR3000 processor to work with: - CODEC that supports short frame format. - One Toshiba TC58A040F flash device - Echo cancellation on			

CKF (TBD hex)Check Flash Checks (checksum) if the flash data is correctly programmed in the Flash devices. The flash checksum is stored in the first Flash device. This checksum checks all of the flash memory except for voicetags.
If the data is correct the return value is FF (hex). Otherwise the return value is 0.

Example:

CKF			
Byte Sequence:	Microcontroller	TBD	AA
	ISD-SR3000	TBD	FF
Description:	ISD-SR3000 is instructed to run a Checksum on the Flash memory, and it returns a code of FF, indicating the data is correct.		

ISD CONFIDENTIAL INFORMATION

CKV (TBD hex)

Check Voicetags

Checks (checksum) if the voicetag data is correctly programmed in the Flash devices. Each voicetag has its own checksum, which is stored with the voicetag. This command checks the voicetag checksums for all voicetags, reporting an error if any one is wrong.

If all the voicetag checksums are correct the return value is FF (hex). Otherwise the return value is the number of the first voicetag with a bad checksum. By clearing that voicetag number with command EVS, and then repeating command CKV, bad voicetag flash entries can be stepped through and erased.

Example:

CKV			
Byte Sequence:	Microcontroller	TBD	AA
	ISD-SR3000	TBD	03
Description:	ISD-SR3000 is instructed to run a Checksum on the Voicetag memory, and it returns a code of 03, indicating that an error was found in voicetag 03, and perhaps also others.		

CTW (TBD hex)

Convert Token to Word String *token_number*

Converts the token number indicated by *token_number* to an English (or other language) string. Parameter token-number is a 16-bit word.

This command is intended to allow language personalization to be entirely contained in ISD-SR3000's flash.

Note that this command assumes a token number can uniquely identify a recognition word string-with ISD-SR3000, a given token number occurs only once across all topics.

The CTW command returns a string which may be up to 256 bytes long, in following format:

Byte 1 - Number of bytes in string. A zero (00) value means the token was not found, in which case this is the only byte returned.

Bytes 2-N - String of bytes representing the word string, with the first character in byte 2 and successive characters in subsequent bytes.

The ISD-SR3000 engine makes no assumptions about the content of the string that is returned - interpretation of the contents is entirely up to the host microcontroller. Any byte values may be used, including the values for spaces, international characters, and upper and lower cases. 16-bit Unicode values or custom string representations can also be used, as long as they are represented by an integral number of bytes with total length 255 or less.

Example:

CTW							
Byte Sequence:	Microcontroller	TBD	37	AA	AA	AA	AA
	ISD-SR3000	TBD	37	03	42	79	65
Description:	Token 37 is passed to ISD-SR3000, and it responds with the 3 byte character string 42, 79, 65 (ASCII code for 'Bye'). This is an example only - the actual value of token 37 depends on the vocabulary programmed into ISD-SR3000's flash by the customer. See "PROGRAMMING DIFFERENT VOCABULARIES AND LANGUAGES" on page 2-1.						

EVA (TBD hex) Erase all Voicetags for user 1-4 *user_numbers*

Erases all voicetags for users 1 to 4, as selected by the bits in byte *user_numbers*

For confidentiality reasons, the voicetag is actually erased from flash - it is not just marked 'unused'.

If *user_numbers* is all zeros, then ERR_PARAM is set in the Error Word (See "GEW (1B hex) Get Error Word" on page 2-51).

If the erase operation can not be performed, then ERR_MEM is set in the Error Word (See "GEW (1B hex) Get Error Word" on page 2-51).

If the EVA command completes normally, then EV_NORMAL_END is set in the Status Word (See "GSW (14 hex) Get Status Word" on page 2-55).

Bit 0 - Reserved

0 This bit must be set to 0

Bit 1 - User 1

0 Do not erase any voicetags for user 1

1 Erase all voicetags for User 1

Bit 2 - User 2

0 Do not erase any voicetags for user 2

1 Erase all voicetags for User 2

Bit 3 - User 3

0 Do not erase any voicetags for user 3

1 Erase all voicetags for User 3

Bit 4 - User 4

0 Do not erase any voicetags for user 4

1 Erase all voicetags for User 4

Bits 5-7 - Reserved

000 These bits must be set to 0

ISD CONFIDENTIAL INFORMATION

Example:

EVA			
Byte Sequence:	Microcontroller	TBD	0C
	ISD-SR3000	TBD	0C
Description:	ISD-SR3000 is instructed to erase the voicetags for users 3 and 4.		

EVS (TBD hex)**Erase Selected Voicetag *voicetag_number***

Erases the voicetag indicated by byte *voicetag_number*. As part of the erasing procedure, the checksum for that entry is automatically updated (See "CKV (TBD hex) Check Voicetags" on page 2-16).

For confidentiality reasons, the voicetag is actually erased from flash - it is not just marked 'unused'.

If *voicetag_number* has a value greater than the number of voicetags (i.e. if *voicetag_number* is larger than 64 for a system with 65 voicetags, since voicetag numbering starts at 0), then ERR_PARAM is set in the Error Word (See "GEW (1B hex) Get Error Word" on page 2-51).

If the erase operation can not be performed, then ERR_MEM is set in the Error Word (see Command GEW).

If the EVS command completes normally, then EV_NORMAL_END is set in the Status Word (See "GSW (14 hex) Get Status Word" on page 2-55).

Example:

EVS			
Byte Sequence:	Microcontroller	TBD	18
	ISD-SR3000	TBD	18
Description:	ISD-SR3000 is instructed to erase voicetag number 18.		

FRQ (TBD hex)**Flush Recognition Queue**

Flushes all entries from the recognition queue. Following this command, the EV_RECO_QUEUE bit in the Status word is cleared to zero.

Example:

FRQ			
Byte Sequence:	Microcontroller	TBD	
	ISD-SR3000	TBD	
Description:	ISD-SR3000 is instructed to flush the recognition queue.		

GCFG (02 hex) Get Configuration Value

Returns a 24-bit word containing the following information:

Bits 0-7

Magic number, which specifies the ISD-SR3000 firmware version.

Bits 9-8

Memory type.

- 00 Reserved
- 01 Reserved
- 10 Toshiba's TC58A040F Flash
- 11 Samsung's KM29N040T Flash

Bits 10-23

Bits 10-23 indicate the vocabulary data set and version number in the ISD-SR3000.

This command should be used together with the CFG and INIT commands during ISD-SR3000 processor initialization.

Example:

GCFG				
Byte Sequence:	Microcontroller	02	AA	AA
	ISD-SR3000	02	02	03
Description:	ISD-SR3000's configuration value is requested, and it returns the code for Toshiba flash, and magic number (firmware revision number) 03 (hex).			

GEW (1B hex) Get Error Word

Returns the 16-bit error word.

Error Word

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

The GEW command reads the error word. All bits in the error word are cleared to zero during reset and after execution of each GEW command.

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

ISD CONFIDENTIAL INFORMATION

The bits of the error word are as follows:

Bits 0 - ERR_BARGE

- 0 No barge-in error
- 1 Barge-in. The user interrupted an ISD-SR3000 operation by 'barging in'. Barge-in occurs when the user issues the barge-in command while the ISD-SR3000 processor is executing a command. The precise wording of the barge-in command is set by the vocabulary - typically the command is something like "<keyword> Cancel" or "<keyword> nevermind". (See "PROGRAMMING DIFFERENT VOCABULARIES AND LANGUAGES" on page 2-1).

Bit 1 - ERR_OPCODE

- 0 No opcode errors
- 1 Illegal opcode. The ISD-SR3000 processor does not recognize the opcode.

Bit 2 - ERR_COMMAND

- 0 No command errors
- 1 Illegal command sequence. The command is not legal in the current state.

Bit 3 - ERR_PARAM

- 0 No parameter errors
- 1 Illegal parameter. The value of the parameter is out of range, or is not appropriate for the command.

Bit 4

- 0 or 1 Bit 4 is reserved and should be disregarded.

Bit 5 - ERR_COMM

- 0 No communications error
- 1 Microcontroller MICROWIRE communication error.

Bit 6 - ERR_TIMEOUT

- 0 No timeout error
- 1 Time-out error. Depending on the ISD-SR3000 processor's state, more than 100 milliseconds elapsed between the arrival of two consecutive bytes (for commands that have parameters).

Bit 7 - ERR_INVALID

- 0 No context error
- 1 Command can not be performed in current context.

Bits 15-8

- 0 or 1 Bits 15-8 are reserved and should be disregarded. These bits may return any mix of 0 and 1.

Example:

GEW				
Byte Sequence:	Microcontroller	1B	AA	AA
	ISD-SR3000	1B	00	20
Description:	ISD-SR3000's error word is requested, and it returns the code for a communications error.			

GI (25 hex) Get Information item

Returns the 16-bit value specified by byte value item from one of the internal registers of the ISD-SR3000 processor. Note that some values returned will never exceed 255 - the high order bits will always be zero in these cases.

item may be one of the following:

- 00 Returns the total number of voicetag memory locations
- 01 Returns the number of unused voicetags in the shared pool
- 02 Returns the voicetag number of the next voicetag that should be used from the shared pool. Any flash memory supports only a finite number of write cycles. ISD-SR3000 tracks the number of times each voicetag is written to and allocates voicetags from the shared pool using an algorithm that tries to distribute write cycles evenly throughout the flash, to maximize the flash memory lifetime.
- 03-0E Reserved. Do not use. Returns unpredictable values, but will not cause an ERR_PARAM in the Error Word.
- 0F Returns the number of Topics in the ISD-SR3000 vocabulary. (Note - topics may not be included in ISD-SR3000, in which case this is a reserved value for item and should not be used.)
- 10 Returns the number of voicetags used by user 1
- 11 Returns the number of voicetags used by user 2
- 12 Returns the number of voicetags used by user 3
- 13 Returns the number of voicetags used by user 4
- 14-FF Reserved. Do not use. Returns unpredictable values, but will not cause an ERR_PARAM in the Error Word.

ISD CONFIDENTIAL INFORMATION

Example:

G1 00					
Byte Sequence:	Microcontroller	25	00	AA	AA
	ISD-SR3000	25	00	00	03
Description:	Information item number 00 (hex) is requested from ISD-SR3000, and it returns the 16-bit value 0003 (hex).				

GNR (TBD hex)

Get Next Recognition

Gets the token for the next word recognized. This command should be issued by the microcontroller after it is interrupted by ISD-SR3000, following ISD-SR3000's recognition of a word (as evidenced by the EV_RECO bit being set in the Status Word—see the GSW command). If the GNR command is issued when the EV_RECO bit is not set, the GNR command will execute but the data returned will not be valid.

The GNR command returns 4 bytes, defined as follows:

Byte 1

The number of recognition events in the queue after this one was removed. Value FF (hex) means an overflow has occurred and does not necessarily mean there are precisely 255 left.

Byte 2

The Topic number that the recognition event occurred in. Topic FF (hex) indicates an error code and the following two bytes can be used for reporting error information.

The error information is TDB.

Bytes 3 & 4

The Token number that recognition event matches. The special token number zero hex (00h) means a pause was found. Token numbers that are used in one topic must not appear in another topic - thus by knowing only the token number, the microcontroller can uniquely identify the specific word found and the topic. See also "CTW (TBD hex) Convert Token to Word String token_number" on page 2-48.

Example:

GNR						
Byte Sequence:	Microcontroller	TBD	AA	AA	AA	AA
	ISD-SR3000	TBD	02	01	00	03
Description:	The next recognition event is requested from ISD-SR3000. ISD-SR3000's response indicates: • there are 02 (hex) events in the recognition queue, in addition to the event returned with this command • this recognition event is from Topic 01 (hex) • the Token recognized is Token number 0003 (hex).					

GSW (14 hex)**Get Status Word**

Returns the 16-bit status word.

Status Word

The ISD-SR3000 processor has a 16-bit status word to indicate events that occur during normal operation. The ISD-SR3000 processor asserts the MWRQST signal (driven low), to indicate a change in the status word. This signal remains active until the ISD-SR3000 processor receives a GSW command.

The status word is cleared during reset, and upon successful execution of the GSW command.

The bits in the status word are used as follows:

Bit 0 - EV_RECO_QUEUE

- 0 There are no events in the ISD-SR3000 recognition queue.
- 1 Reco-1 has one or more recognition events in its queue. Use command GNR to retrieve items from the queue.

Bit 1 - EV_NORMAL_END

- 0 When this bit is zero, it means either:
 - a) no command is underway, or
 - b) a command is being processed but has not yet completed, or
 - c) a command completed but had an error (as indicated by a '1' in the EV_ERROR bit)
- 1 Normal completion of an operation, e.g., end of playing of a prompt, or final token detected in a command sequence.

Bit 2 - EV_MEMFULL

- 0 Memory is not full
- 1 Memory is full.

Bit 3 - EV_ERROR

- 0 No error detected
- 1 Error detected in the last command. The host microcontroller must issue the GEW command to return the error code and clear the error condition.

Bit 4 - EV_RESET

- 0 Normally, this bit changes to 0 after performing the INIT command.
- 1 When the ISD-SR3000 processor completes its power-up sequence and enters the RESET state, this bit is set to 1, and the MWRQST signal is activated (driven low).

Normally, this bit changes to 0 after performing the INIT command. If this bit is set during normal operation of the ISD-SR3000 processor, it indicates an internal ISD-SR3000 processor error. The microcontroller can recover from such an error by re-initializing the system.

ISD CONFIDENTIAL INFORMATION

Bit 6,5 - EV_RECORD_STATUS

- 00 Displayed following commands other than RKW and RTAG, or if the RKW and RTAG commands cause an error.
- 01 This bit combination is only displayed when the RKW or RTAG commands complete successfully (as evidenced by a 1 in EV_NORMAL_END). When this code is displayed, it means the keyword or voicetag was successfully captured and analyzed, but the RKW or RTAG command must be repeated for verification. Normally, one repeat is needed for a verification cycle. However, if the recognition engine is having trouble analyzing the voicetag or keyword, it may require multiple repeats. It is up to the host microcontroller to limit the number of repeats (i.e. to avoid endless looping). The loop should be broken by erasing the same keyword or voicetag location (using the EKEY or EVS commands).
- 10 Never used.
- 11 This bit combination is only displayed when the RKW or RTAG commands complete successfully (as evidenced by a 1 in EV_NORMAL_END). When this code is displayed, it means the keyword or voicetag was successfully captured and analyzed, and a repeat of the command is not needed.

Bits 15-7

0 or 1 Bits 15-7 are reserved and should be disregarded. These bits may return any mix of 0 and 1.

Example:

GSW				
Byte Sequence:	Microcontroller	14	AA	AA
	ISD-SR3000	14	00	02
Description:	ISD-SR3000's error word is requested, and it returns code 0002 hex, indicating normal command completion (EV_NORMAL_END).			

INIT (13 hex) Initialize System

Execute this command after the ISD-SR3000 processor has been configured (see the CFG and GCFG commands).

Performs a soft reset of the ISD-SR3000 processor, which includes:

- TBD

Example:

INIT		
Byte Sequence:	Microcontroller	13
	ISD-SR3000	13
Description:	ISD-SR3000's is initialized.	

INJ (29 hex) Inject IVS data n byte₁ . . . byte_n

Injects vocabulary data of size n bytes (where n is expressed as a 32-bit value) to good Flash blocks.

This command programs Flash devices, on a production line, with vocabulary data. ("PROGRAMMING DIFFERENT VOCABULARIES AND LANGUAGES" on page 2-1) It is optimized for speed; all ISD-SR3000 processor detectors are suspended during execution of the command. Use the CKV command to check whether programming was successful.

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

Example:

INJ 000E0000							
Byte Sequence:	Microcontroller	29	00	0E	00	00	n bytes of vocabulary data
	ISD-SR3000	29	00	0E	00	00	Echo of n bytes of vocabulary data
Description:	Inject E0000 (hex) (917,504 decimal) bytes of data into the Flash.						

KWNR (TBD hex) Keyword Not Required

After this command is issued, spoken commands do not have to be prefaced with a valid keyword to be recognized. Note that although a keyword is not required after this command is issued, use of a keyword is still allowed - commands prefaced with a keyword will still be recognized.

Example:

KWNR		
Byte Sequence:	Microcontroller	TBD
	ISD-SR3000	TBD
Description:	This command tells ISD-SR3000 that a keyword is not required for recognition.	

KWR (TBD hex) Keyword Required

After this command is issued, all spoken commands must be prefaced with a valid keyword to be recognized.

ISD CONFIDENTIAL INFORMATION

Example:

KWR		
Byte Sequence:	Microcontroller	TBD
	ISD-SR3000	TBD
Description:	This command tells ISD-SR3000 to require a keyword.	

PDM (1A hex) Go To Power-down Mode

Switches the ISD-SR3000 processor to power-down mode. Sending any command while in power-down mode returns the ISD-SR3000 processor to normal operation mode.

If an event report is pending (i.e., MWRQST is active), and it is not processed by the microcontroller prior to issuing the PDM command, the event is lost.

Example:

PDM		
Byte Sequence:	Microcontroller	1A
	ISD-SR3000	1A
Description:	This command tells ISD-SR3000 to go into power-down mode.	

PKW (TBD hex) Play Keyword *keyword_number*

Plays the keyword indicated by byte *keyword_number*, where *keyword_number* is defined as follows:

Bits 2-0

- 000 Play the user-programmed keyword for user 1 (see RKW on page 2-62)
- 001 Play the user-programmed keyword for user 2 (see RKW on page 2-62)
- 010 Play the user-programmed keyword for user 3 (see RKW on page 2-62)
- 011 Play the user-programmed keyword for user 4 (see RKW on page 2-62)
- 100 Play the factory-default keyword for user 1
- 101 Play the factory-default keyword for user 2
- 110 Play the factory-default keyword for user 3
- 111 Play the factory-default keyword for user 4

Bits 7-3

- 0 These bits must be set to zero.

Example:

PKW			
Byte Sequence:	Microcontroller	TBD	12
	ISD-SR3000	TBD	12
Description:	ISD-SR3000 is instructed to play voicetag number 12 (hex).		

PRD (TBD hex) Disable Pause Reporting

Disables reporting of detection of pause events. When pause reporting is disabled, ISD-SR3000 does not generate a recognition event for each pause heard. Typically, if ISD-SR3000 is left in recognition mode, while waiting for a keyword, pause detection should be disabled to avoid accumulating recognition events on the recognition queue during each detection of a pause in conversation. PRD and PRE affect placement of pause events on the recognition queue - the contents already on the recognition queue do not change as a result of executing PRD or PRE.

Example:

PRD			
Byte Sequence:	Microcontroller	TBD	
	ISD-SR3000	TBD	
Description:	This command tells ISD-SR3000 to disable pause reporting.		

PRE (TBD hex) Enable Pause Reporting

Enables reporting of detection of pause events. See PRD on page 2-59.

Example:

PRE			
Byte Sequence:	Microcontroller	TBD	
	ISD-SR3000	TBD	
Description:	This command tells ISD-SR3000 to enable pause reporting.		

PV (TBD hex) Play Voicetag *voicetag_number*

Plays the voicetag indicated by byte *voicetag_number*.

ISD CONFIDENTIAL INFORMATION

Example:

PV			
Byte Sequence:	Microcontroller	TBD	12
	ISD-SR3000	TBD	12
Description:	ISD-SR3000 is instructed to play voicetag number 12 (hex).		

RES (1D hex)Resume

Resumes the activity that was suspended by the SUSP command.

Example:

RES			
Byte Sequence:	Microcontroller		1D
	ISD-SR3000		1D
Description:	This command tells ISD-SR3000 to resume.		

RESK (TBD hex)Reset Keyword *user_numbers*

Resets the keyword for users 1 to 4, as selected by the bits in byte *user_numbers*.

Bit 0 - User 1

- 0 Do not change keyword
- 1 Reset keyword to factory-programmed value for User 1

Bit 1 - User 2

- 0 Do not change keyword
- 1 Reset keyword to factory-programmed value for User 2

Bit 2 - User 3

- 0 Do not change keyword
- 1 Reset keyword to factory-programmed value for User 3

Bit 3 - User 4

- 0 Do not change keyword
- 1 Reset keyword to factory-programmed value for User 4

Bits 4-7 - Reserved

- 000 These bits must be set to 0

ISD CONFIDENTIAL INFORMATION

Example:

RES			
Byte Sequence:	Microcontroller	TBD	08
	ISD-SR3000	TBD	08
Description:	Resets the keyword for user 4.		

RESR (TBD hex) Reset Recognition Engine

Resets the recognition engine to the initial power up state with recognition disabled, the recognition queue is flushed (as if the FRQ command were issued), and all topics are disabled. The keywords and voicetag entries are not affected.

Example:

RESR			
Byte Sequence:	Microcontroller	TBD	
	ISD-SR3000	TBD	
Description:	This command tells ISD-SR3000 to reset the recognition engine.		

RD (TBD hex) Disable Recognition

Stops the recognition engine. Any recognition events on the recognition queue are preserved. The setting of the EV_RECO_QUEUE bit in the Status Register is not altered by this command. If the RD command is issued while the recognition engine is in the middle of recognizing a word or pause, that recognition event is discarded.

Example:

RD			
Byte Sequence:	Microcontroller	TBD	
	ISD-SR3000	TBD	
Description:	This command tells ISD-SR3000 to reset the recognition engine.		

RE (TBD hex) Enable Recognition

Turns the ISD-SR3000 recognition engine on, allowing it to start listening for a keyword or command (as determined by the KWR and KWNR commands).

ISD CONFIDENTIAL INFORMATION

Example:

RE			
Byte Sequence:	Microcontroller	TBD	
	ISD-SR3000	TBD	
Description:	This command tells ISD-SR3000 to start the recognition engine.		

RKW (TBD hex)Record Keyword *user_id*

Records a new keyword for the user number indicated in byte *user_id*. See the description for the EV_RECORD_STATUS bit under the GSW command. The definition of the *user_id* byte is as follows:

Bits 1,0

- 00 Record the keyword for user 1
- 01 Record the keyword for user 2
- 10 Record the keyword for user 3
- 11 Record the keyword for user 4

Bits 7-2

- 0 These bits must be set to zero.

Example:

RKW			
Byte Sequence:	Microcontroller	TBD	01
	ISD-SR3000	TBD	01
Description:	ISD-SR3000 is instructed to record the keyword for user 2.		

RTAG (TBD hex)Record Voicetag *user_id* *voicetag_number*

Records a new voicetag into the voicetag location given by byte *voicetag_number* and marks the voicetag to show it belongs to the user numbers flagged in byte *user_id*. Note that a voicetag can be used by more than one user (for example, this might be used for a voicetag of "Police".) See the description for the EV_RECORD_STATUS bit under the GSW command.

The byte *user_id* is coded as follows:

Bit 0

- 0 Do not assign this voicetag for user 1
- 1 Assign this voicetag to user 1

- Bit 1
- 0 Do not assign this voicetag for user 2
 - 1 Assign this voicetag to user 2

- Bit 2
- 0 Do not assign this voicetag for user 3
 - 1 Assign this voicetag to user 3

- Bit 3
- 0 Do not assign this voicetag for user 4
 - 1 Assign this voicetag to user 4

- Bits 7-4
- 0 These bits must be set to zero.

Example:

RTAG				
Byte Sequence:	Microcontroller	TBD	03	10
	ISD-SR3000	TBD	03	10
Description:	ISD-SR3000 is instructed to record a new voicetag into voicetag number 10 (hex) and mark the tag as belonging to user 4.			

SAS (1E hex) Say Argumented Sentence *sentence_n arg*

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 ISD-SR3000 processor sets the EV_NORMAL_END bit in the status word, and activates the MWRQST signal.

If the current vocabulary is undefined, ERR_INVALID is reported.

Example:

SAS				
Byte Sequence:	Microcontroller	1E	00	03
	ISD-SR3000	1E	00	03
Description:	Announce the first sentence in the sentence table of the currently selected vocabulary with '3' as the actual parameter.			

ISD CONFIDENTIAL INFORMATION

SDET (10 hex)

Set Detectors Mask *detectors_mask*

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.

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 2-3

Reserved. Must be cleared to 0.

Bit 4

Report detection of a constant energy.

Bit 5

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

Example:

SDET A3			
Byte Sequence:	Microcontroller	10	A3
	ISD-SR3000	10	A3
Description:	Set reporting of all ISD-SR3000 processor detectors, except for end-of-DTMF		

SO (07 hex)

Say One Word *word_number*

Plays the word number *word_number* in the current vocabulary. The 1 byte *word_number* may be any value from 0 through the index of the last word in the vocabulary.

When playback of the selected word has been completed, the ISD-SR3000 processor sets the EV_NORMAL_END bit in the status word, and activates the MWRQST signal.

ISD CONFIDENTIAL INFORMATION

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

If the current vocabulary is undefined, ERR_INVALID is reported.

Example:

SO 00			
Byte Sequence:	Microcontroller	07	00
	ISD-SR3000	07	00
Description:	Announce the first word in the word table of the currently selected vocabulary.		

SS (1F hex) Say Sentence *sentence_n*

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

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

When playing has been completed, the ISD-SR3000 processor sets the EV_NORMAL_END bit in the status word, and activates the MWRQST signal.

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

If the current vocabulary is undefined, ERR_INVALID is reported.

Example:

SS 00			
Byte Sequence:	Microcontroller	1F	00
	ISD-SR3000	1F	00
Description:	Announce the first sentence in the sentence table of the currently selected vocabulary.		

SUSP (1C hex) Suspend

Suspends the execution of the current PV, SAS, SO, SS, or SW command. The SUSP command does not change the state of the ISD-SR3000 processor; execution can be resumed with the RES command.

Example:

SUSP		
Byte Sequence:	Microcontroller	1C
	ISD-SR3000	1C
Description:	This command tells ISD-SR3000 to pause.	

ISD CONFIDENTIAL INFORMATION

SW (21 hex) Say Words n word₁ . . . word_n

Plays n words, indexed by word₁ to word_n. On completion, the EV_NORMAL_END bit in the status word is set, and the 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, or if n > 8, ERR_PARAM is reported.

If the current vocabulary is undefined, ERR_INVALID is reported.

Example:

SW 02 00 00					
Byte Sequence:	Microcontroller	21	02	00	00
	ISD-SR3000	21	02	00	00
Description:	Announce the first word, in the word table of the currently selected vocabulary, twice.				

TAGC (TBD hex) Change Voicetag user_id_info, voicetag_number

Changes the *user_id* info associated with a particular voicetag given in byte *voicetag_number*. See TAGQ also. Byte *user_id_info* is defined as shown below.

Bits 4,0

- 00, 01 This voicetag's data for user 1 is not changed
- 10 Voicetag data is updated to show that this voicetag is not used by user 1.
- 11 Voicetag data is updated to show that this voicetag is used by user 1.

Bits 5,1

- 00, 01 This voicetag's data for user 2 is not changed
- 10 Voicetag data is updated to show that this voicetag is not used by user 2.
- 11 Voicetag data is updated to show that this voicetag is used by user 2.

Bits 6,2

- 00, 01 This voicetag's data for user 3 is not changed
- 10 Voicetag data is updated to show that this voicetag is not used by user 3.
- 11 Voicetag data is updated to show that this voicetag is used by user 3.

Bits 7,3

- 00, 01 This voicetag's data for user 4 is not changed
- 10 Voicetag data is updated to show that this voicetag is not used by user 4.
- 11 Voicetag data is updated to show that this voicetag is used by user 4.

Example:

TAGC				
Byte Sequence:	Microcontroller	TBD	31	10
	ISD-SR3000	TBD	31	10
Description:	ISD-SR3000 is instructed to update voicetag 10's user data to show that it is used by user 1, not used by user 2, and user 3 and 4's information is left unchanged.			

TAGQ (TBD hex) Query Voicetag *voicetag_number*

Returns the user info for the voicetag given by byte *voicetag_number*.

The data returned is as follows:

Bit 0

- 0 This voicetag is not used by user 1.
- 1 This voicetag is used by user 1.

Bit 1

- 0 This voicetag is not used by user 2.
- 1This voicetag is used by user 2.

Bit 2

- 0 This voicetag is not used by user 3.
- 1 This voicetag is used by user 3.

Bit 3

- 0 This voicetag is not used by user 4.
- 1 This voicetag is used by user 4.

Bits 4-7

- x These bits are reserved and may unpredictable values.

Example:

TAGQ				
Byte Sequence:	Microcontroller	TBD	15	AA
	ISD-SR3000	TBD	15	05
Description:	ISD-SR3000 is instructed to return the user information for voicetag 15. The data returned indicates that the voicetag is used by user 1 and by user 3.			

ISD CONFIDENTIAL INFORMATION

TOPD (TBD hex) Disable Topics *topic_id*

Disables selected topics. If the topic number doesn't exist, this command sets bit ERR_PARAM in the Error Word (see the GEW command on page 2-51).

Byte value *topic_id* identifies the topic to be disabled. The special value FF (hex) disables all topics. (Note: topics are currently not planned for the ISD-SR3000 product, but we wanted to include this command description in case this decision is changed.)

Example:

TOPD			
Byte Sequence:	Microcontroller	TBD	10
	ISD-SR3000	TBD	10
Description:	ISD-SR3000 is instructed to disable Topic number 10 (hex).		

TOPE (TBD hex) Enable Topics *topic_id*

Enables selected topics. If the topic number doesn't exist, this command sets bit ERR_PARAM in the Error Word (see the GEW command on page 2-51).

Byte value *topic_id* identifies the topic to be enabled. The special value FF (hex) enables all topics. (Note: topics are currently not planned for the ISD-SR3000 product, but we wanted to include this command description in case this decision is changed.)

Example:

TOPE			
Byte Sequence:	Microcontroller	TBD	05
	ISD-SR3000	TBD	05
Description:	ISD-SR3000 is instructed to enable Topic number 05 (hex).		

TOPQ (TBD hex) Query Topics *topic_id*

Tests if a specific topic is enabled. If the topic number doesn't exist, this command sets bit ERR_PARAM in the Error Word (see the GEW command on page 2-51) and indicates the topic is invalid in a bit in the returned byte.

Byte value *topic_id* identifies the topic number to be queried. (Note: topics are currently not planned for the ISD-SR3000 product, but we wanted to include this command description in case this decision is changed.)

The TOPQ command returns a single byte with the following bit definitions:

Bit 0

- 0 Topic is not enabled.
- 1 Topic is enabled.

Bit 1

- 0 Topic number is valid
- 1 Topic number is invalid. (Bit ERR_PARAM in the Error Word is also set.)

Bits 7-2

0 or 1 Bits 7-2 are reserved and should be disregarded. These bits may return any mix of 0 and 1.

Example:

TOPQ				
Byte Sequence:	Microcontroller	TBD	10	AA
	ISD-SR3000	TBD	10	01
Description:	ISD-SR3000 is queried for the status of Topic 10 (hex). The response from ISD-SR3000 indicates the Topic is a valid number and is enabled.			

VC (28 hex) Volume Control *vol_level*

Controls the energy level of all the voice outputs. 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 24.

For example, if the tunable variable VCD_LEVEL is 6, and *vol_level* is -2, then the output level equals $VCD_LEVEL + vol_level = 4$.

Example:

VC			
Byte Sequence:	Microcontroller	28	05
	ISD-SR3000	28	05
Description:	ISD-SR3000 is instructed to set the <i>vol_level</i> to +5 (hex).		

-152-

CLAIMS

What is claimed is:

1. A voice controlled device, comprising:
a processor;
a processor readable storage medium;
code recorded in the processor readable storage medium to store at least one user assignable appliance name in the processor readable storage medium;
code recorded in the processor readable storage medium to recognize the at least one user assignable appliance name;
code recorded in the processor readable storage medium to recognize a command; and
code recorded in the processor readable storage medium to control the voice controlled device in response to recognizing the user assignable appliance name and the command.
2. The voice controlled device of claim 1, wherein,
the user assignable appliance name and the command are provided using audible speech.
3. The voice controlled device of claim 1, wherein,

-153-

the user assignable appliance name and the command are provided using non-audible speech.

4. The voice controlled device of claim 1 further comprising:

code recorded in the processor readable storage medium to store personal preferences of the voice controlled device associated with the at least one user assignable appliance name; and

code recorded in the processor readable storage medium to personalize the voice controlled device to the stored personal preferences associated with the at least one user assignable appliance name upon recognition of the at least one user assignable appliance name.

5. The voice controlled device of claim 1 further comprising:

code recorded in the processor readable storage medium to store a default appliance name associated with the voice controlled device;

code recorded in the processor readable storage medium to recognize the default appliance name associated with the voice controlled device; and wherein,

code recorded in the processor readable storage medium to control the voice controlled electronic device is further responsive to recognizing the default

-154-

appliance name and the command.

6. The voice controlled device of claim 5,
wherein,

the default appliance name associated with the
voice controlled device is factory assignable.

7. The voice controlled device of claim 5,
wherein,

the default appliance name associated with the
voice controlled device is factory and user assignable.

8. A method of controlling a voice controlled
device, the method comprising:

providing a voice controlled device having a speech
recognition system for recognizing speech;

storing at least one user assignable appliance name
into the voice controlled device;

communicating a communicated appliance name and a
command to the voice controlled device; and

controlling the voice controlled device if the
communicated appliance name is recognized as matching
the at least one user-assignable appliance name and the
command is recognized by the voice controlled device.

9. The method of claim 8 for activating a voice
controlled device, wherein,

-155-

the communicated appliance name and the command are communicated using audible speech.

10. The method of claim 8 for activating a voice controlled device, wherein,

the communicated appliance name and the command are communicated using non-audible speech.

11. A method of controlling a voice controlled device, the method comprising:

providing a voice controlled device having a speech recognition system for recognizing speech;

storing a default appliance name into the voice controlled device;

communicating a communicated name and a command to the voice controlled device; and

controlling the voice controlled device if the communicated name is recognized as matching the default appliance name and the command is recognized by the voice controlled device.

12. The method of claim 11 for activating a voice controlled device, wherein,

the communicated appliance name and the command are communicated using audible speech.

13. The method of claim 11 for activating a voice

-156-

controlled device, wherein,

the communicated appliance name and the command are communicated using non-audible speech.

14. A method for activating a voice controlled device, the method comprising:

providing a voice controlled device having a speech recognition system for recognizing speech;

storing a default appliance name into the voice controlled device;

storing at least one user assignable appliance name into the voice controlled device;

communicating a communicated name and a command to the voice controlled device; and

controlling the voice controlled device if the communicated name is recognized as matching the at least one user assignable appliance name or the default appliance name and the command is recognized by the voice controlled device.

15. A method of assigning a new name to a voice controlled device, the method comprising:

providing a voice controlled device having a speech recognition system for recognizing speech;

activating the voice controlled device; and

communicating a new name to the voice controlled device at least once.

-157-

16. The method of claim 15 for assigning a new name to a voice controlled device, wherein,
the voice controlled device is activated by communicating a current appliance name and a change name command.

17. The method of claim 15 for assigning a new name to a voice controlled device, wherein,
the new name is communicated using audible speech.

18. The method of claim 15 for assigning a new name to a voice controlled device, wherein,
the new name is communicated using non-audible speech.

19. The method of claim 15 for assigning a new name to a voice controlled device, wherein:
the voice controlled device includes prompting capability and the voice controlled device communicates audible prompts to a user in order to request communication from the user of the new name.

20. The method of claim 15 for assigning a new name to a voice controlled device, wherein:
the voice controlled device includes prompting capability and the voice controlled device communicates non-audible prompts to another voice controlled device

-158-

in order to request communication from the device of the new name.

21. A first voice controlled device capable of operating in a communication environment with at least one other voice controlled device, the first voice controlled device comprising:

a processor;

a processor readable storage medium;

code recorded in the processor readable storage medium to store a plurality of user assignable appliance names in the processor readable storage medium for activating the voice controlled device;

code recorded in the processor readable storage medium to recognize the plurality of user assignable appliance names associated with the one voice controlled device;

code recorded in the processor readable storage medium to recognize a command; and

code recorded in the processor readable storage medium to control the voice controlled electronic device in response to recognizing one of the plurality of user assignable appliance names and the command.

22. The first voice controlled device of claim 21 capable of operating in a communication environment with at least one other voice controlled device, wherein,

-159-

the user assignable appliance names and the command are provided using audible speech.

23. The first voice controlled device of claim 21 capable of operating in a communication environment with at least one other voice controlled device, wherein,

the user assignable appliance names and the command are provided using non-audible speech.

24. The first voice controlled device of claim 21 capable of operating in a communication environment with at least one other voice controlled device, the first voice controlled device further comprising:

code recorded in the processor readable storage medium to store personal preferences of the voice controlled device associated with the at least one user assignable appliance name; and

code recorded in the processor readable storage medium to personalize the voice controlled device to the stored personal preferences associated with the at least one user assignable appliance name upon recognition of the at least one user assignable appliance name.

25. The first voice controlled device of claim 21 capable of operating in a communication environment with at least one other voice controlled device, the first voice controlled device further comprising:

-160-

code recorded in the processor readable storage medium to store a default appliance name associated with the voice controlled device;

code recorded in the processor readable storage medium to recognize the default appliance name associated with the voice controlled device; and wherein,

code recorded in the processor readable storage medium to control the voice controlled electronic device is further responsive to recognizing the default appliance name and the command.

26. The first voice controlled device of claim 25 capable of operating in a communication environment with at least one other voice controlled device, wherein,

the default appliance name associated with each of the voice controlled devices is factory assignable.

27. The first voice controlled device of claim 25 capable of operating in a communication environment with at least one other voice controlled device, wherein

the default appliance name associated with each of the voice controlled devices is factory and user assignable.

28. The first voice controlled device of claim 21 capable of operating in a communication environment with

-161-

at least one other voice controlled device, the first voice controlled device further comprising:

a security means to protect each voice controlled device from unauthorized use.

29. The first voice controlled device of claim 24 capable of operating in a communication environment with at least one other voice controlled device, the first voice controlled device further comprising:

a security means to protect each voice controlled device from unauthorized use.

30. The first voice controlled device of claim 27 capable of operating in a communication environment with at least one other voice controlled device, the first voice controlled device further comprising:

a security means to protect each voice controlled device from unauthorized use.

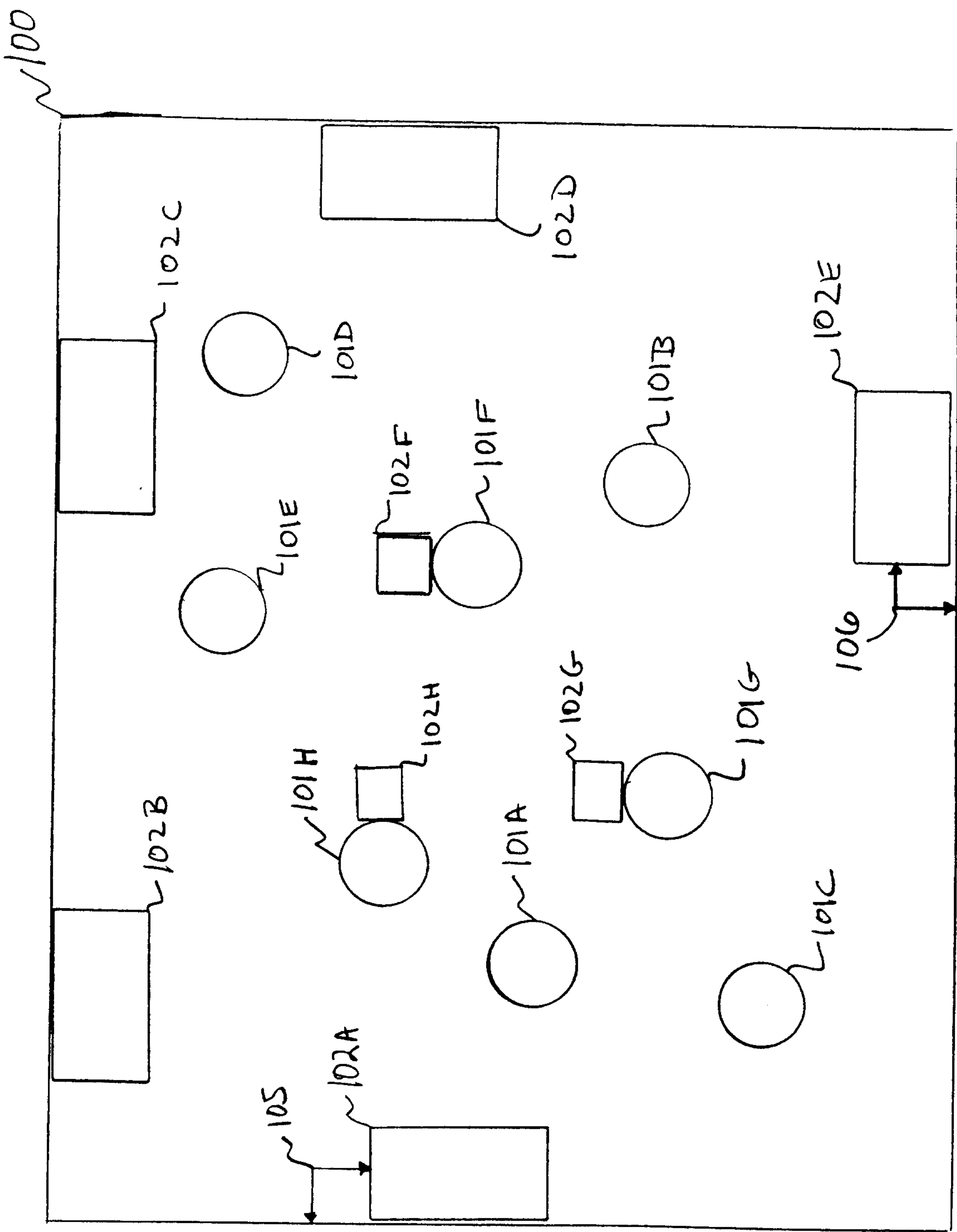
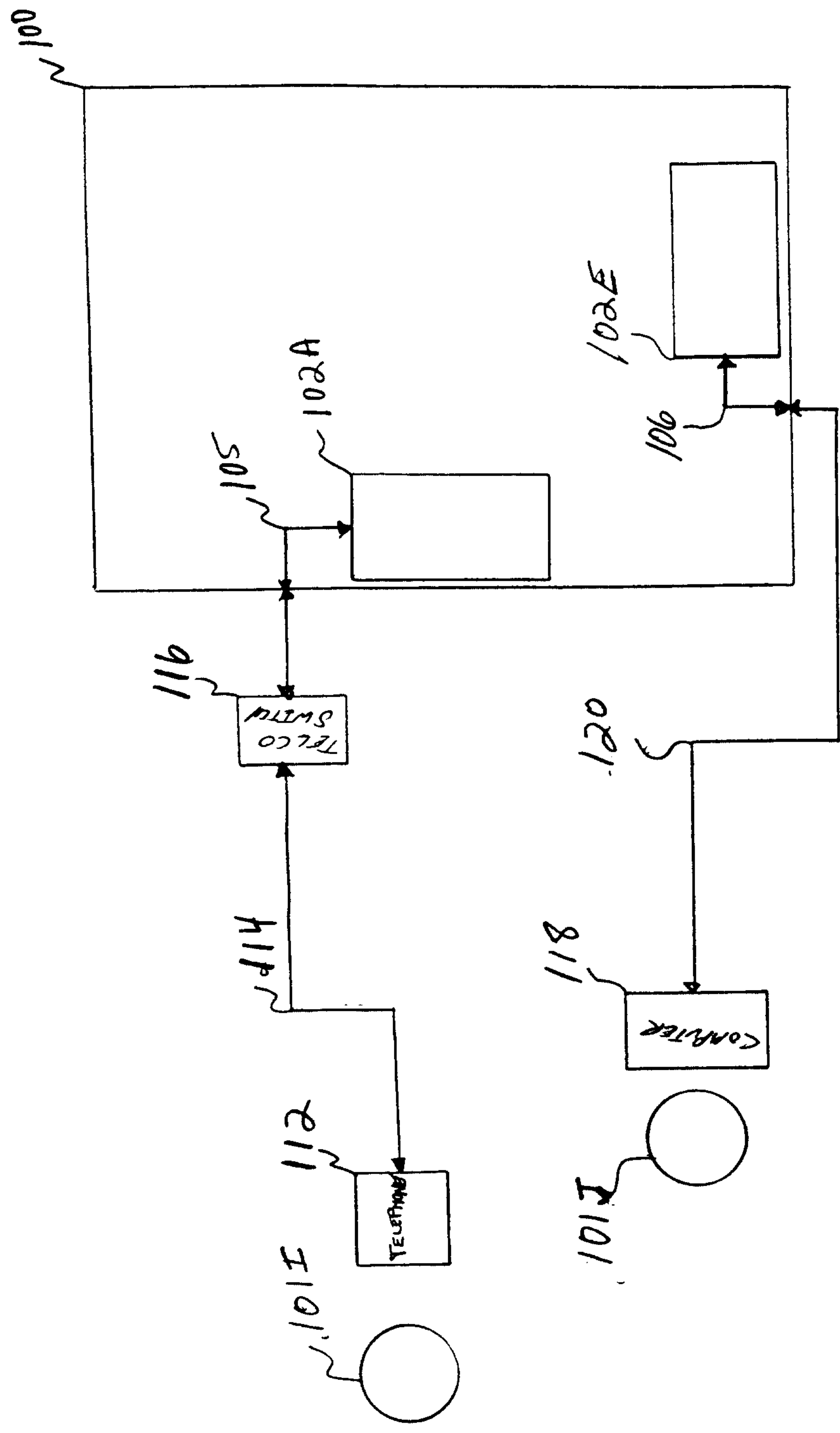


FIG. 1A



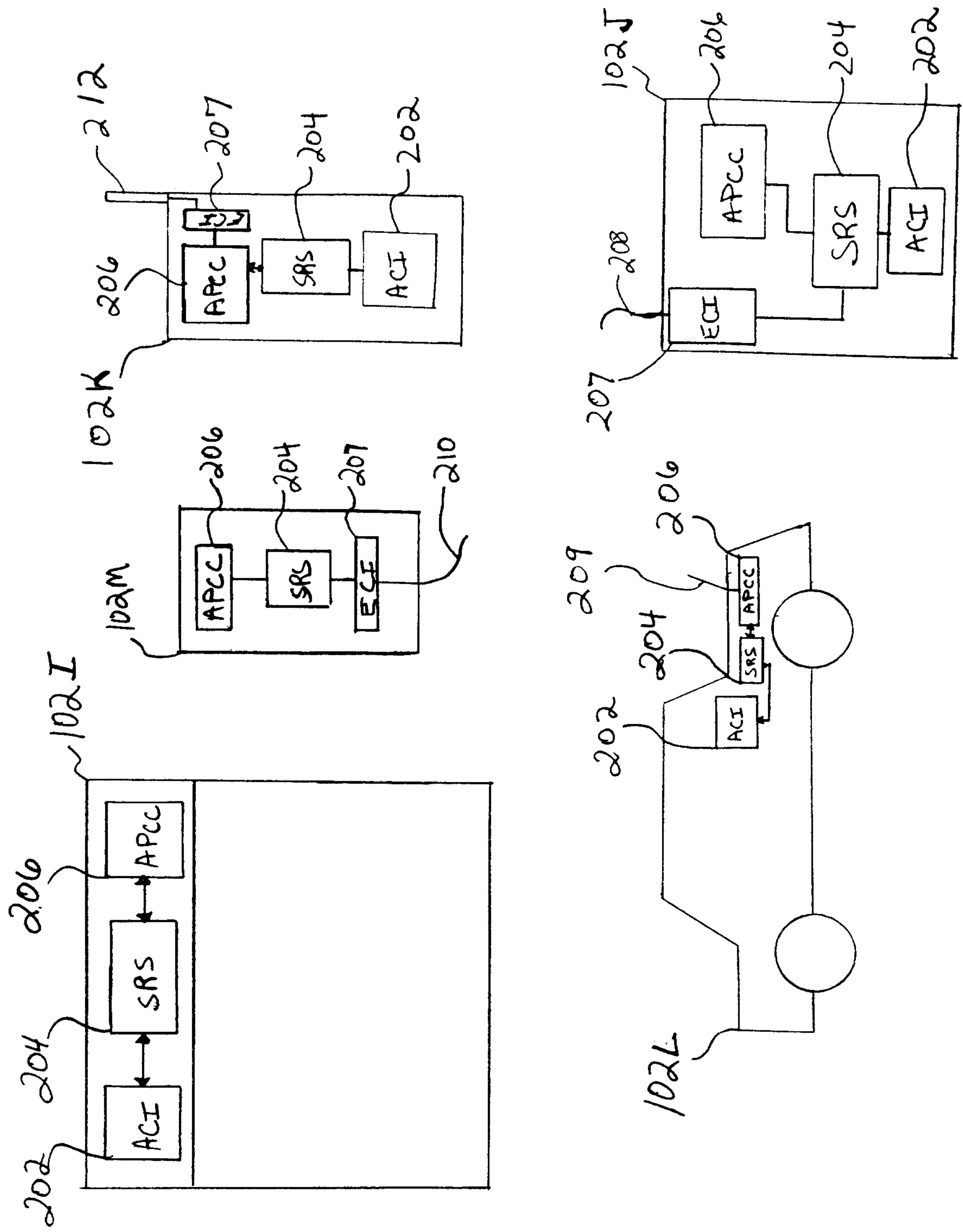
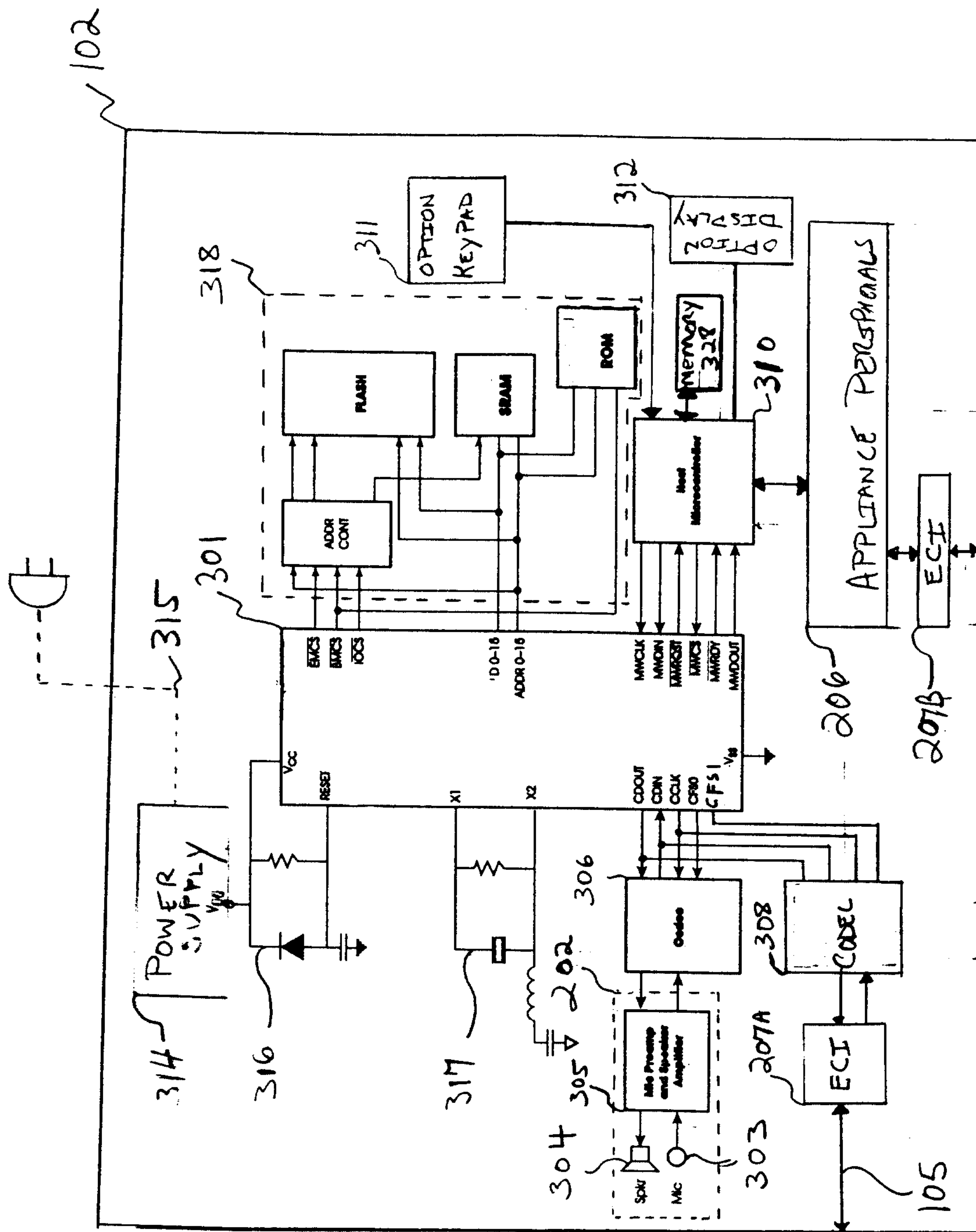


FIG. 2



M
b
LH

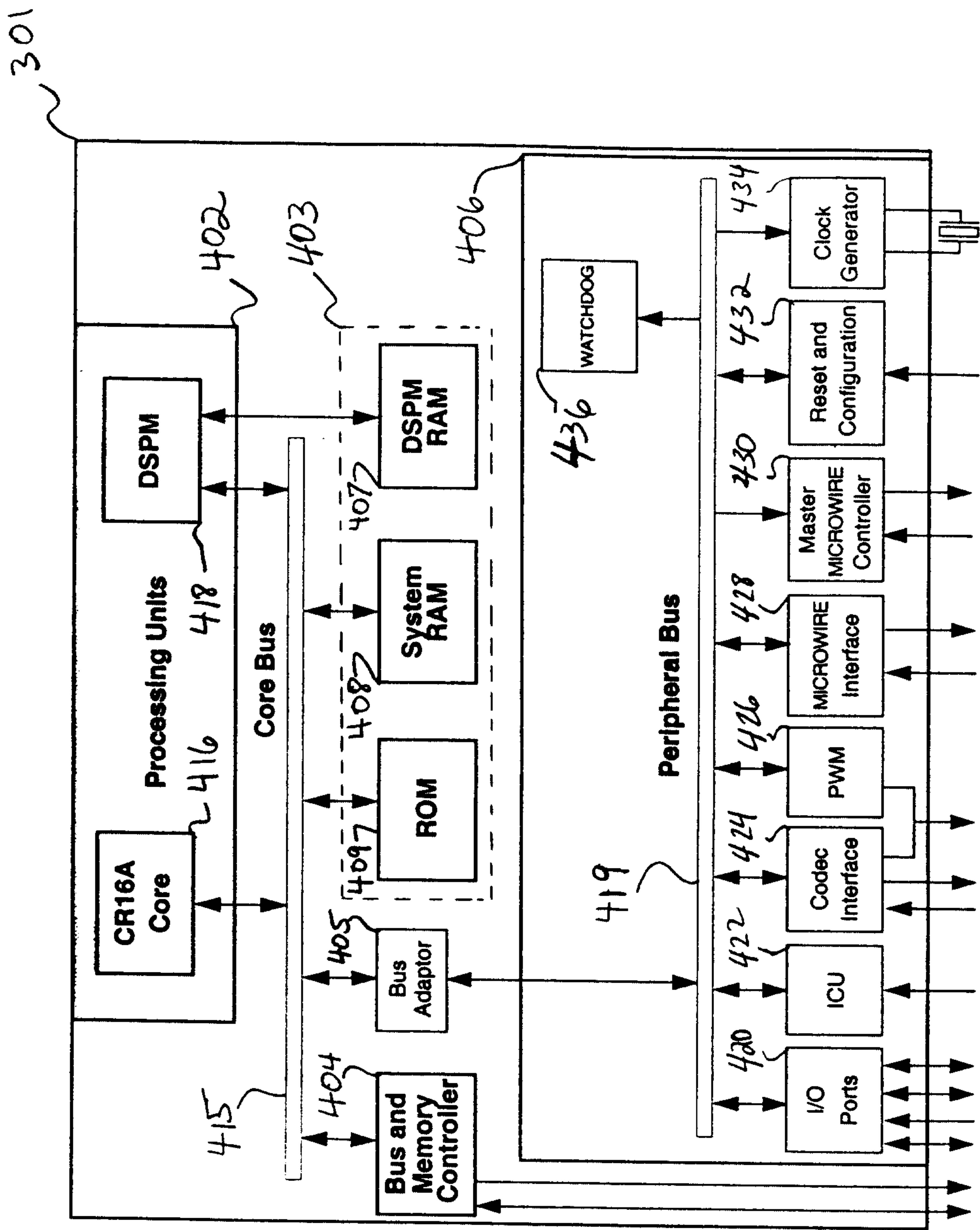


FIG. 4

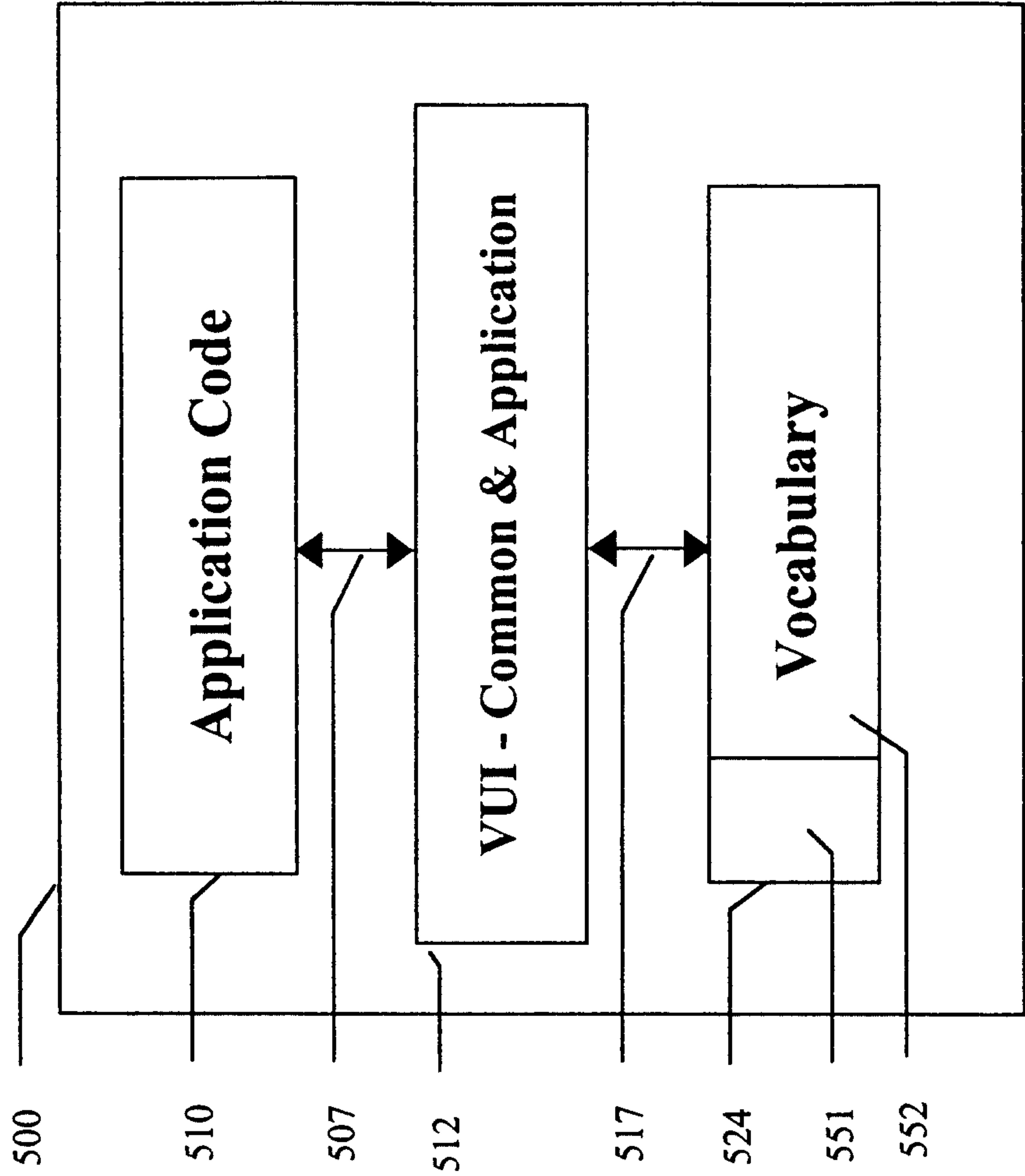
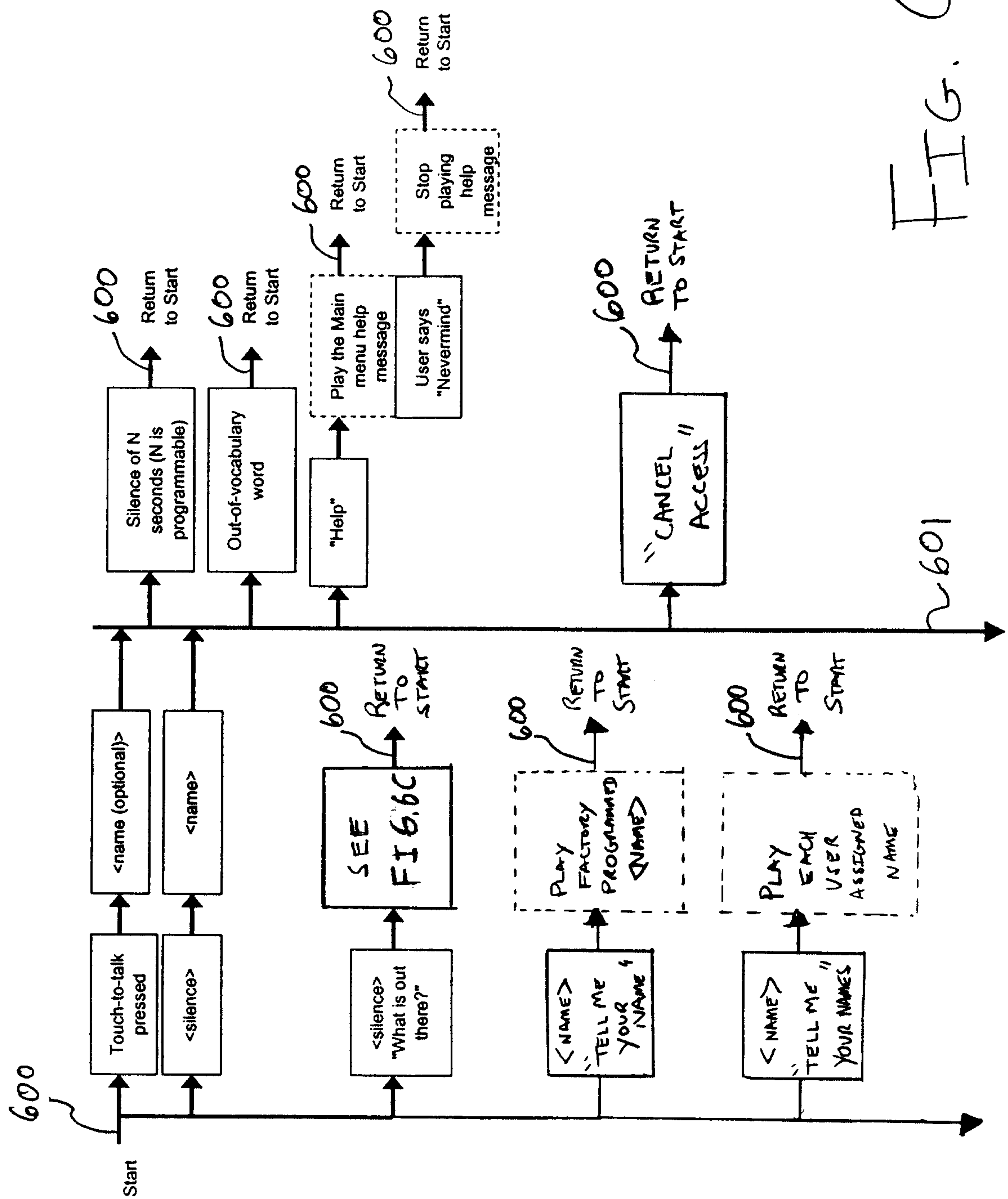


FIG. 5



LIG.

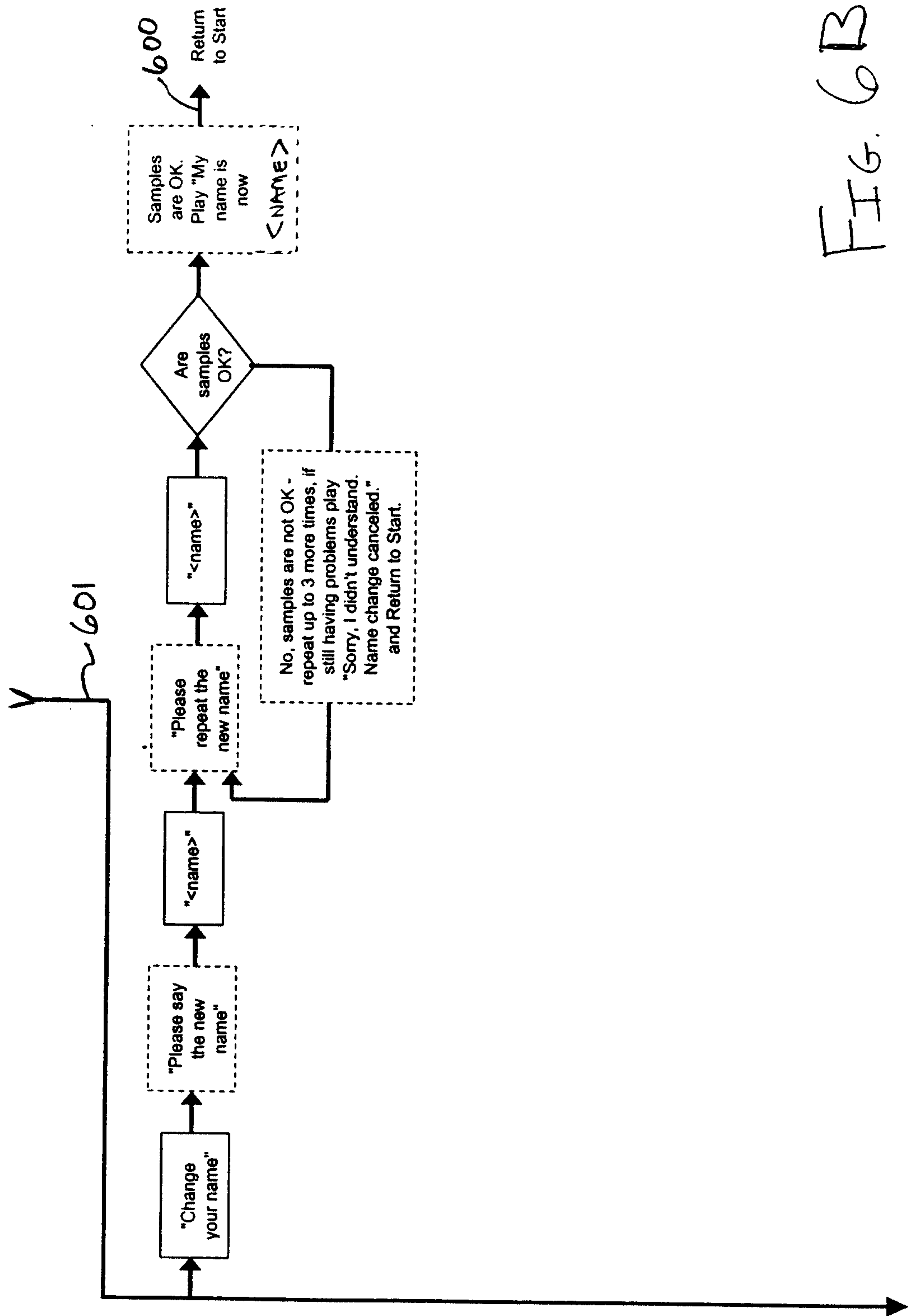


FIG. 6B

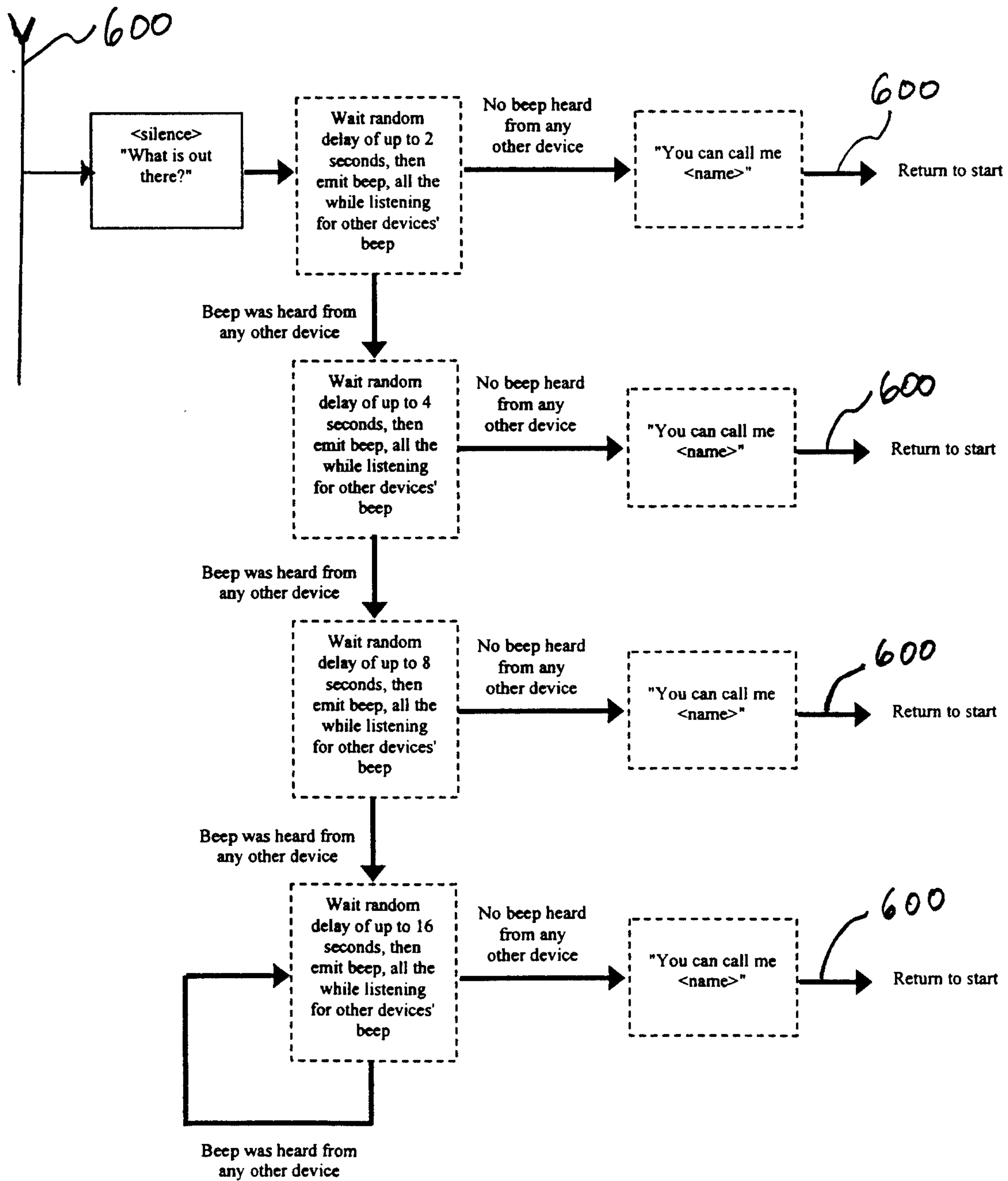


FIG. 6C

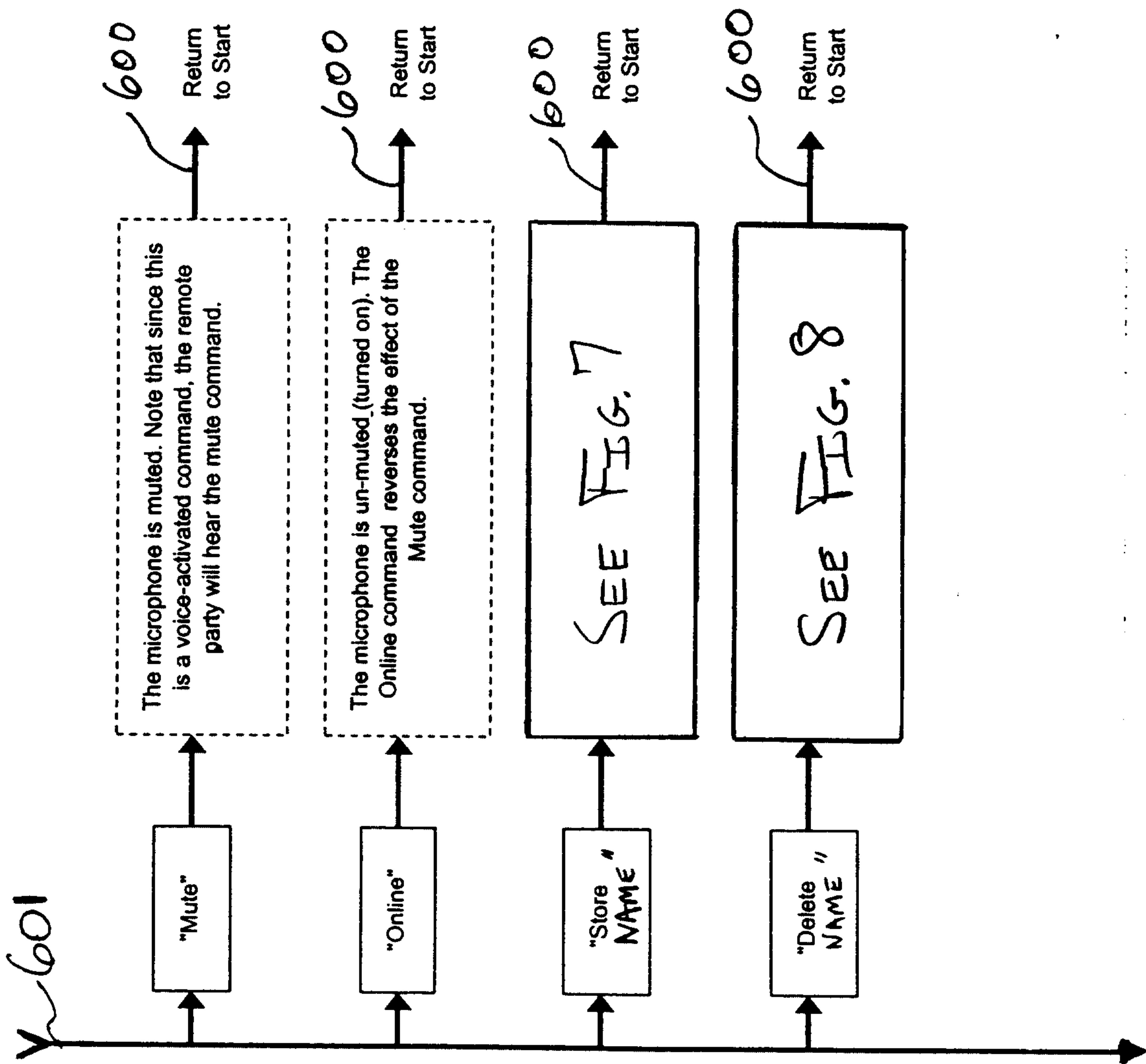
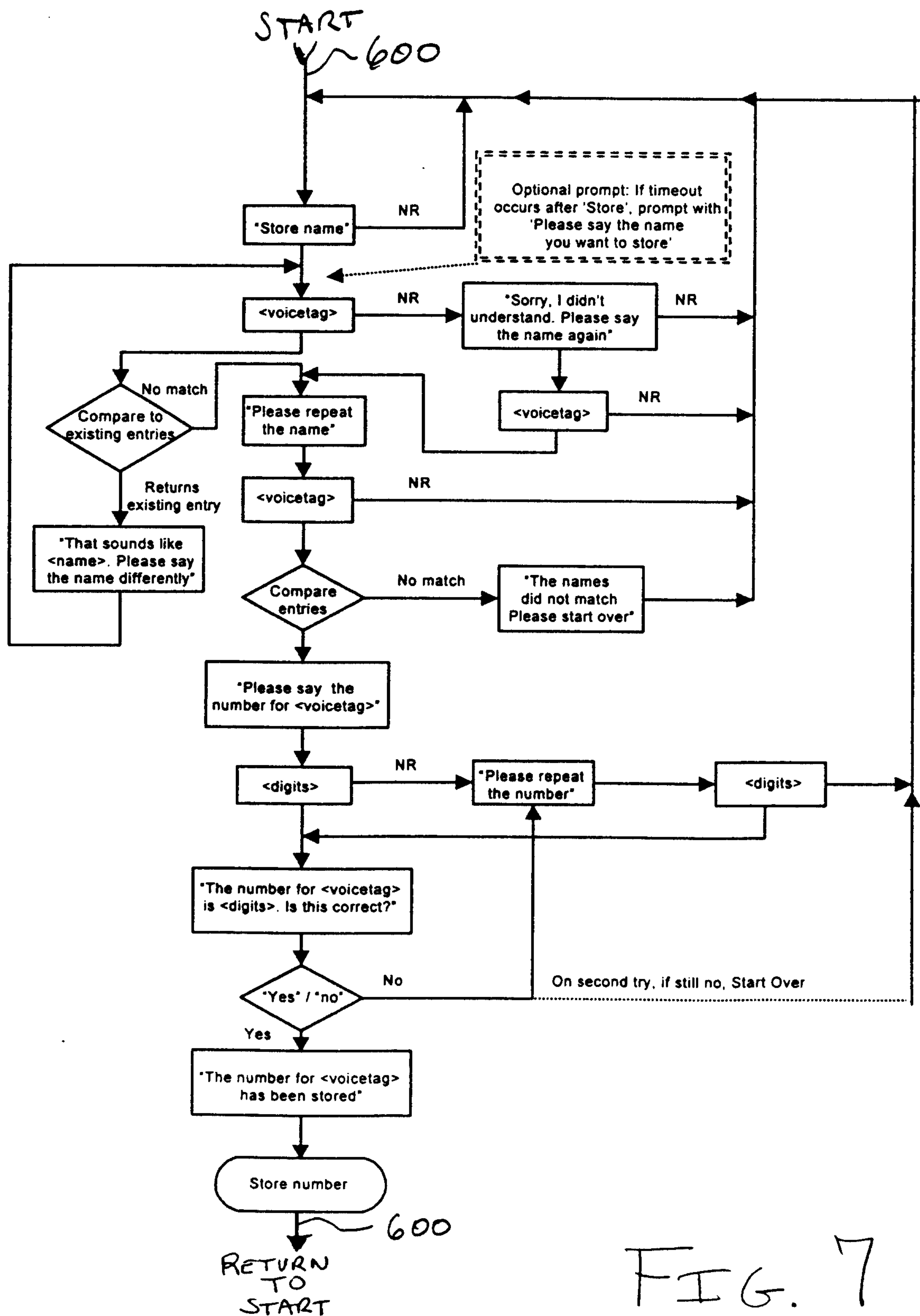


FIG. 6E



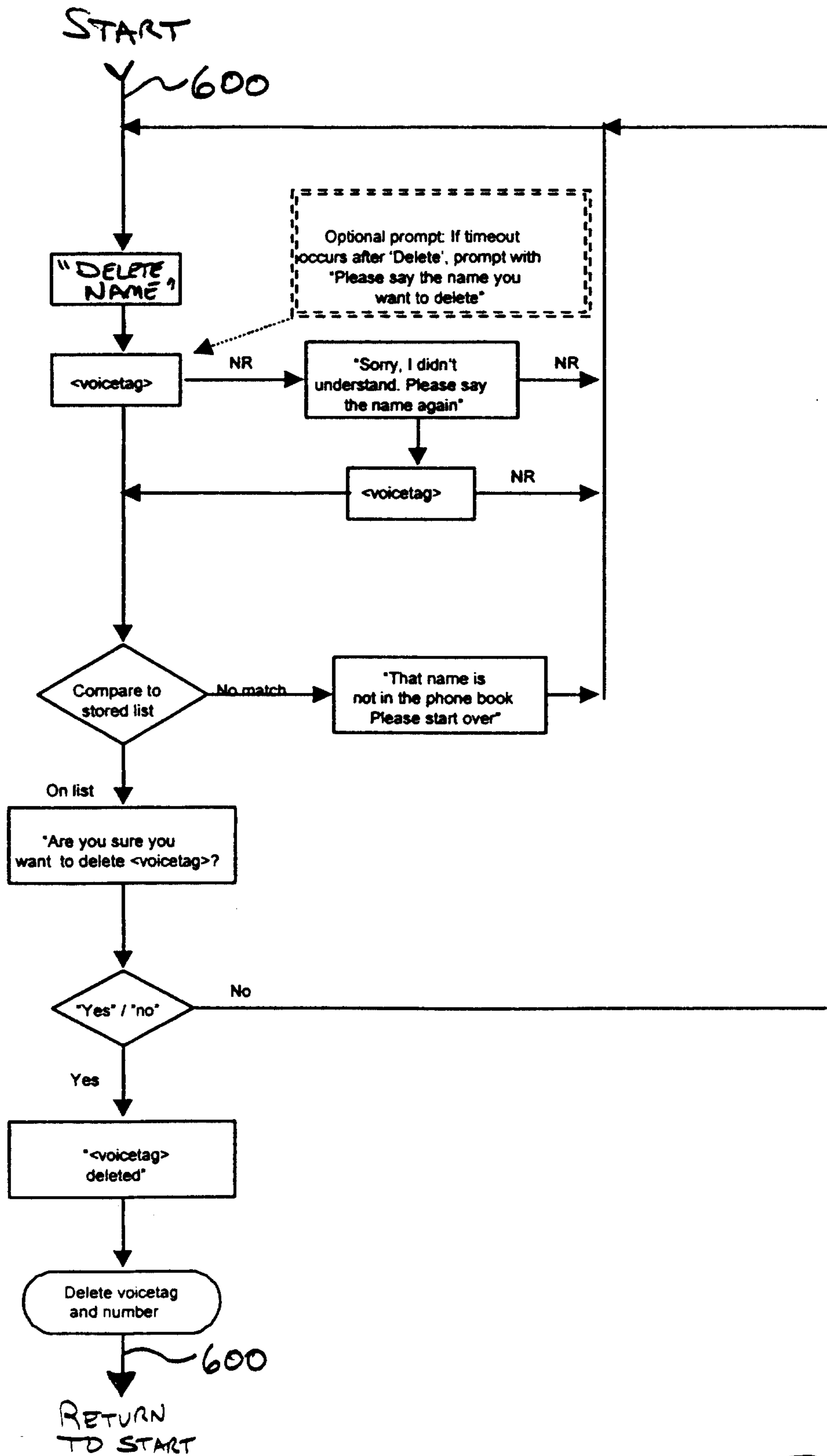


FIG. 8

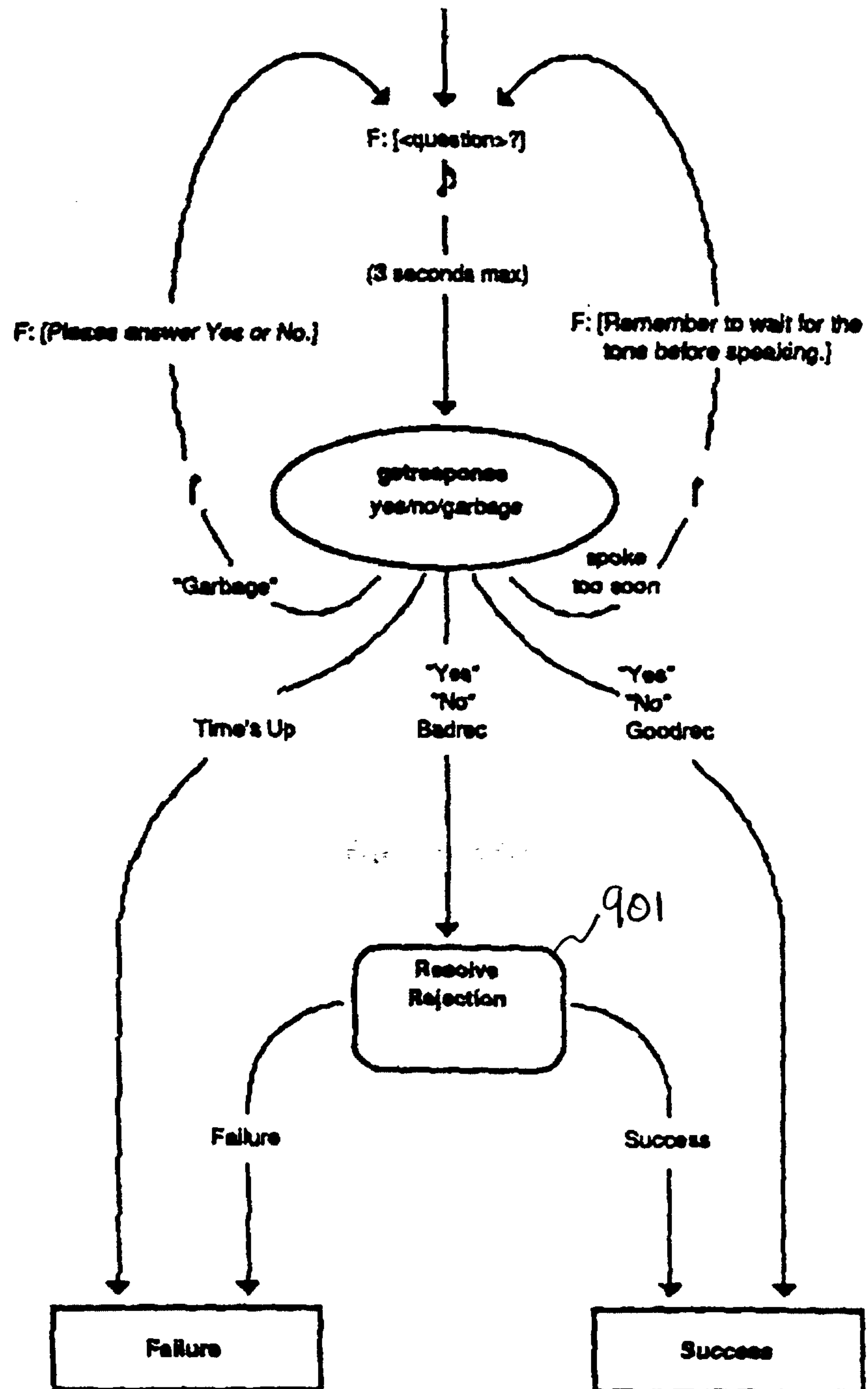


FIG. 9A

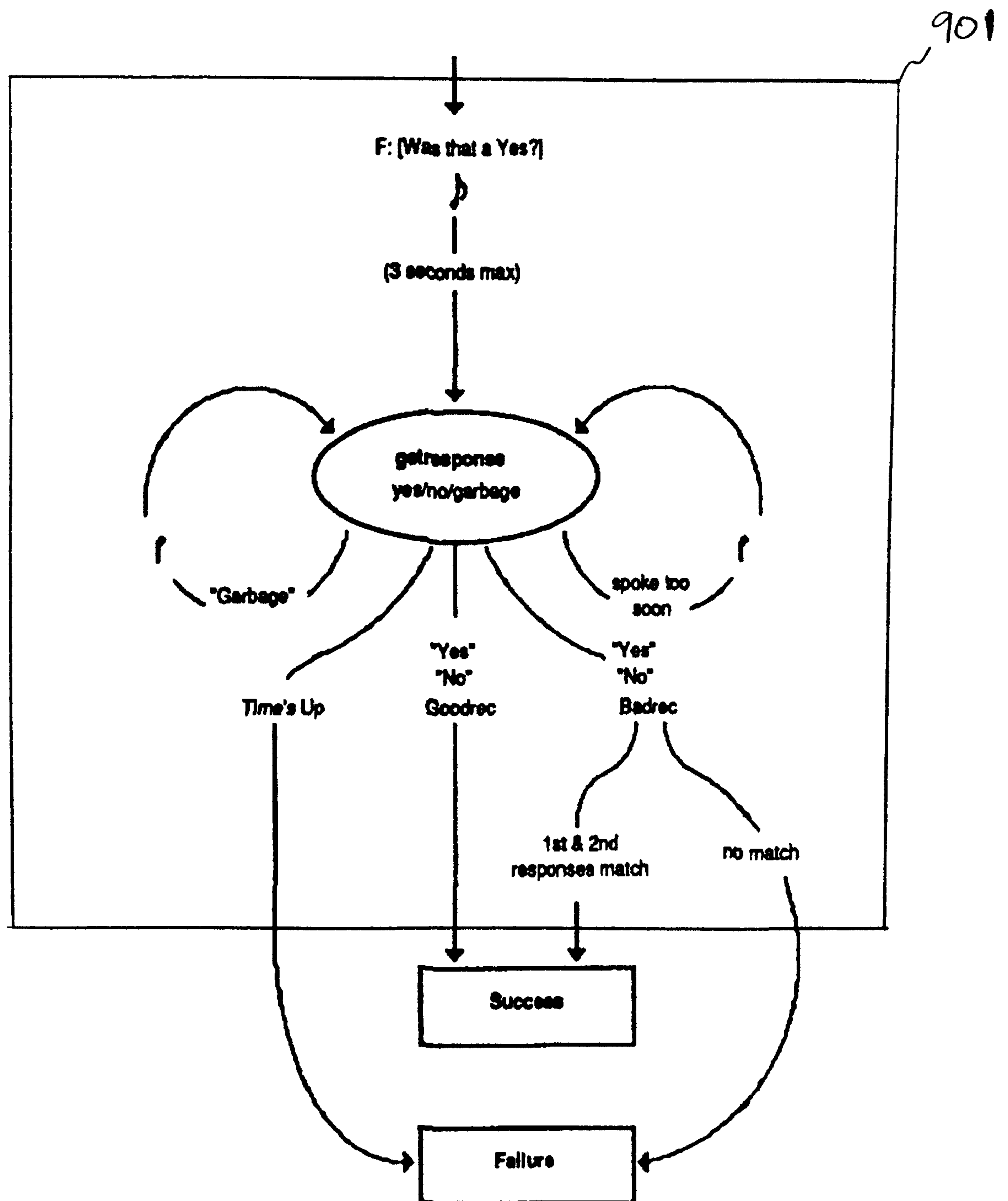


FIG. 9B

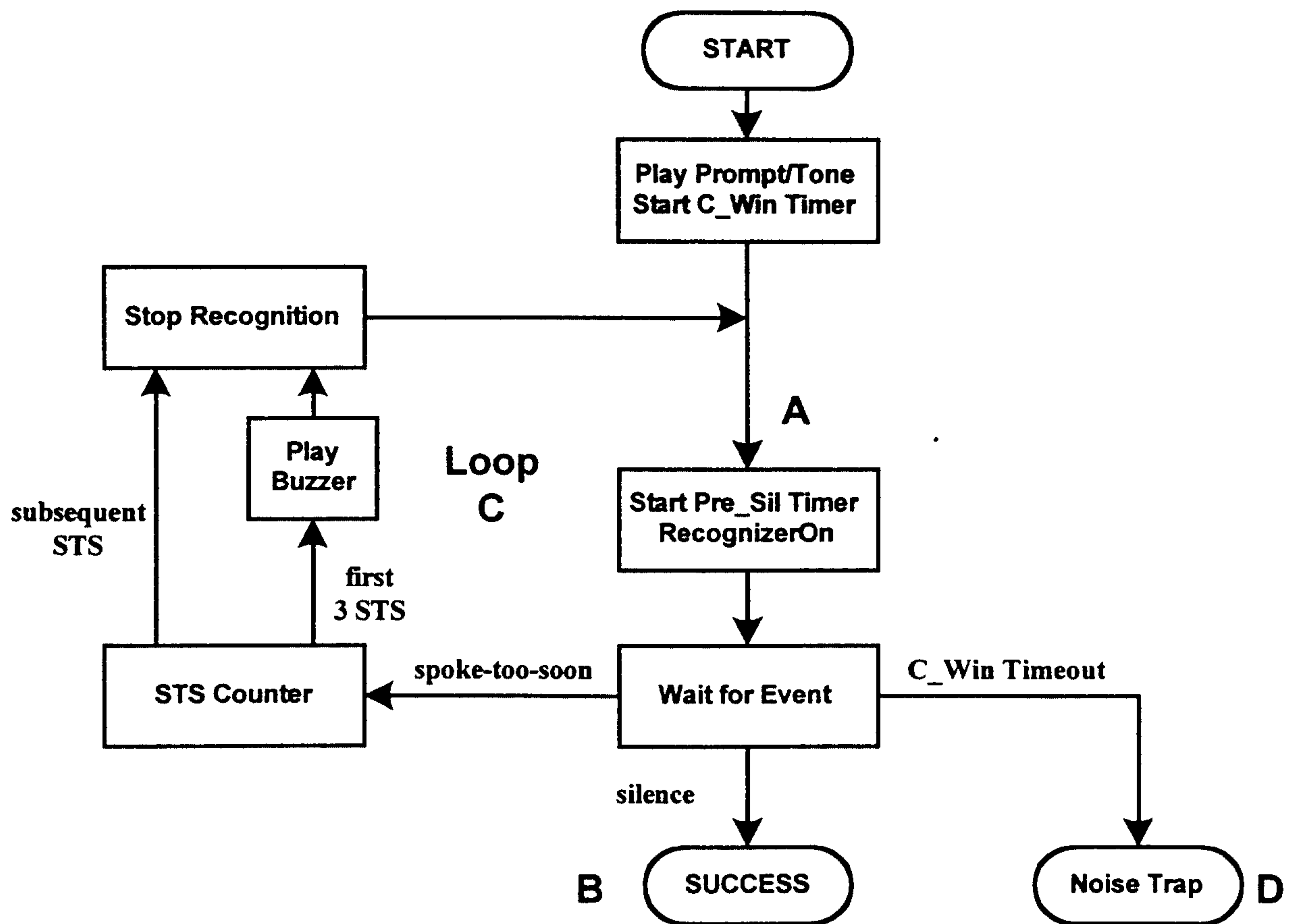


FIG. 10A

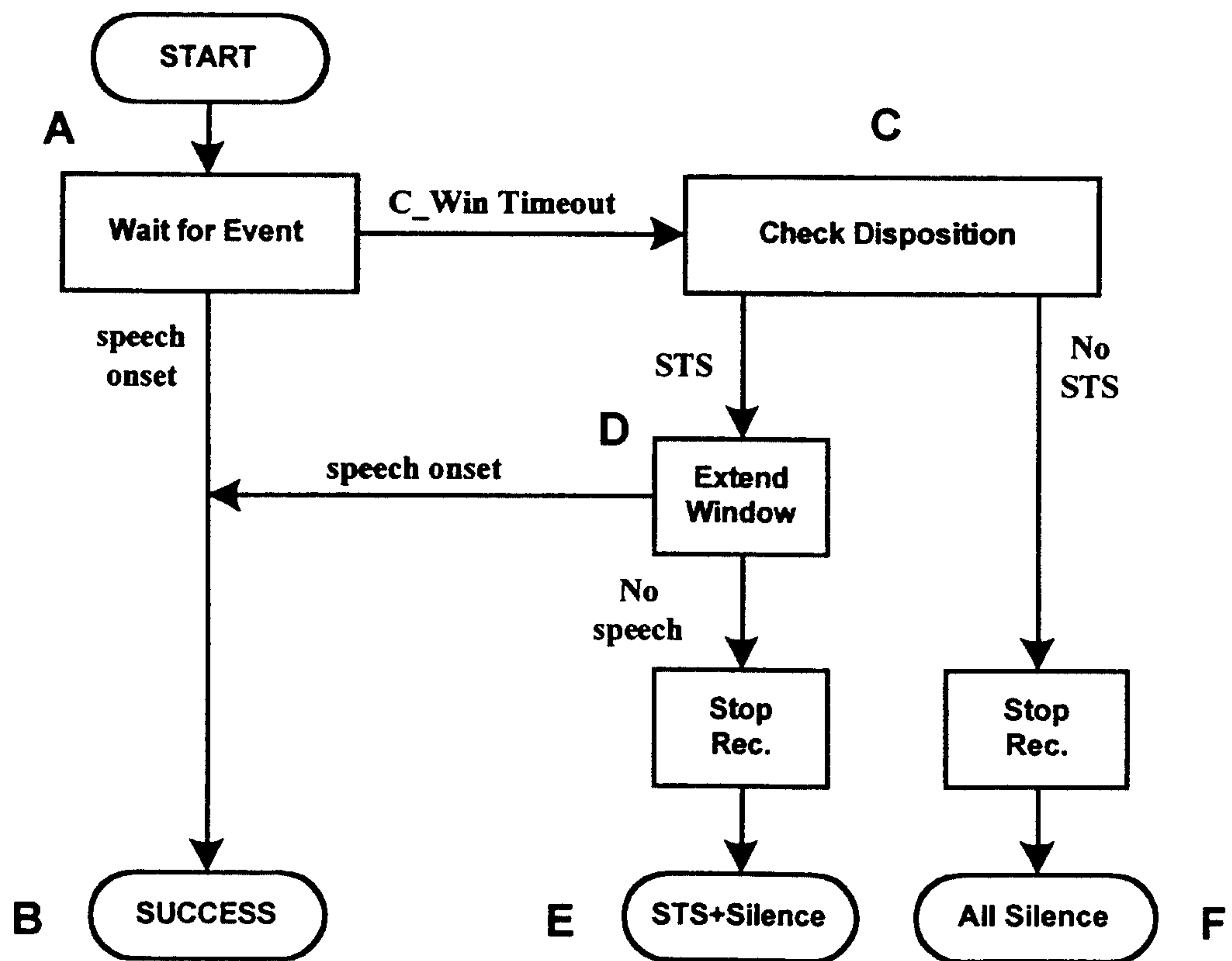


FIG. 10B

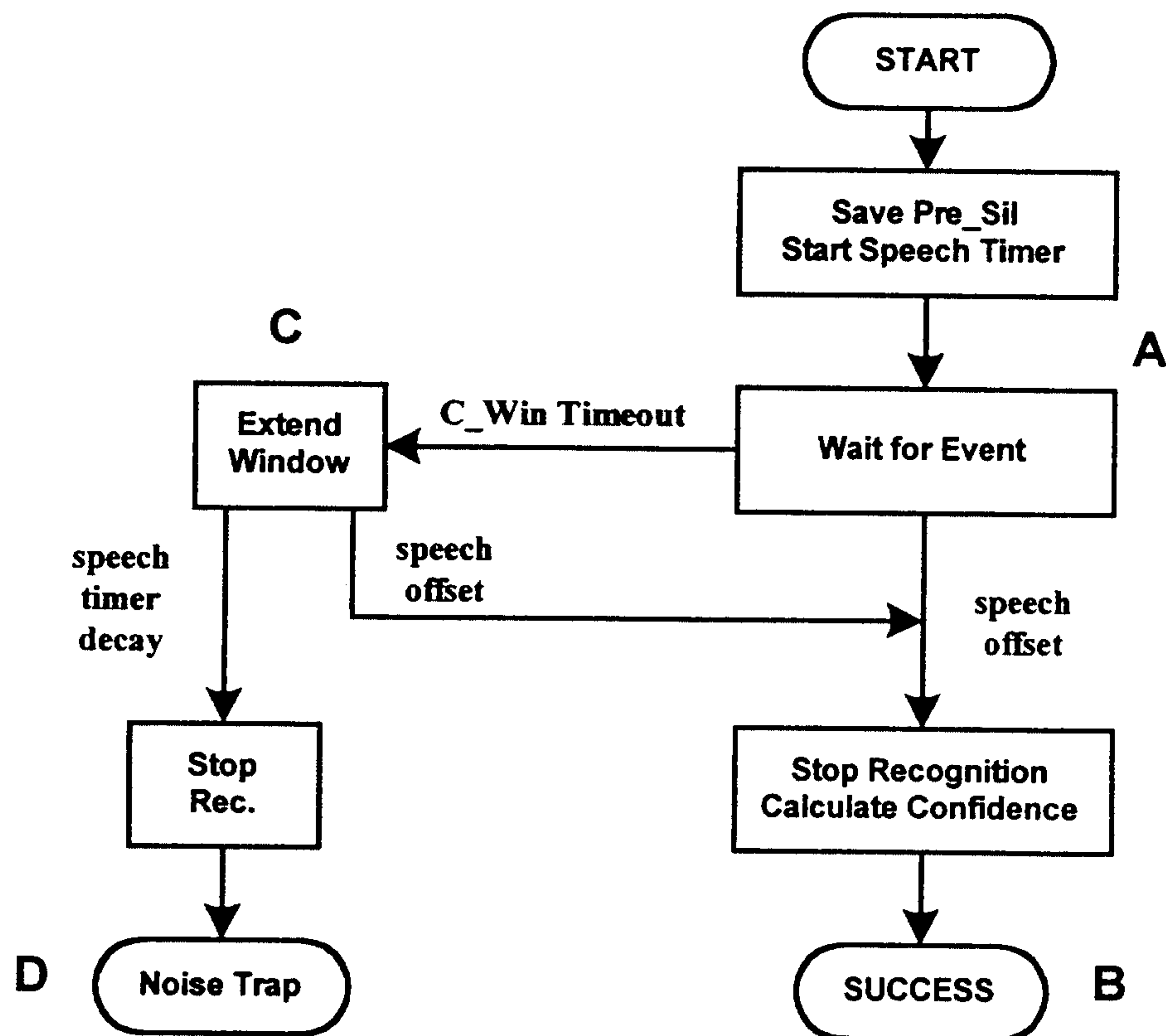


FIG. 10C

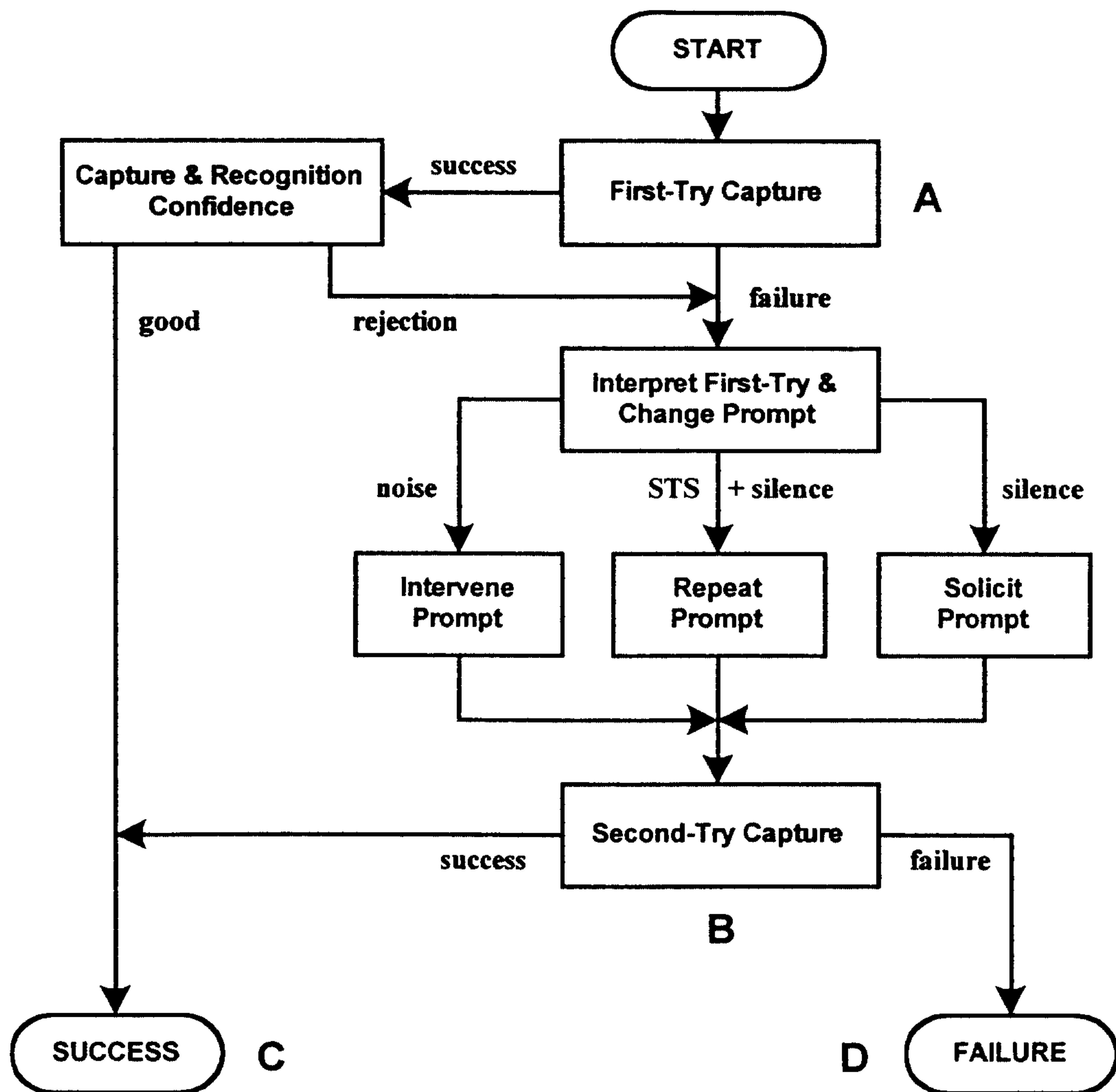


FIG. 11

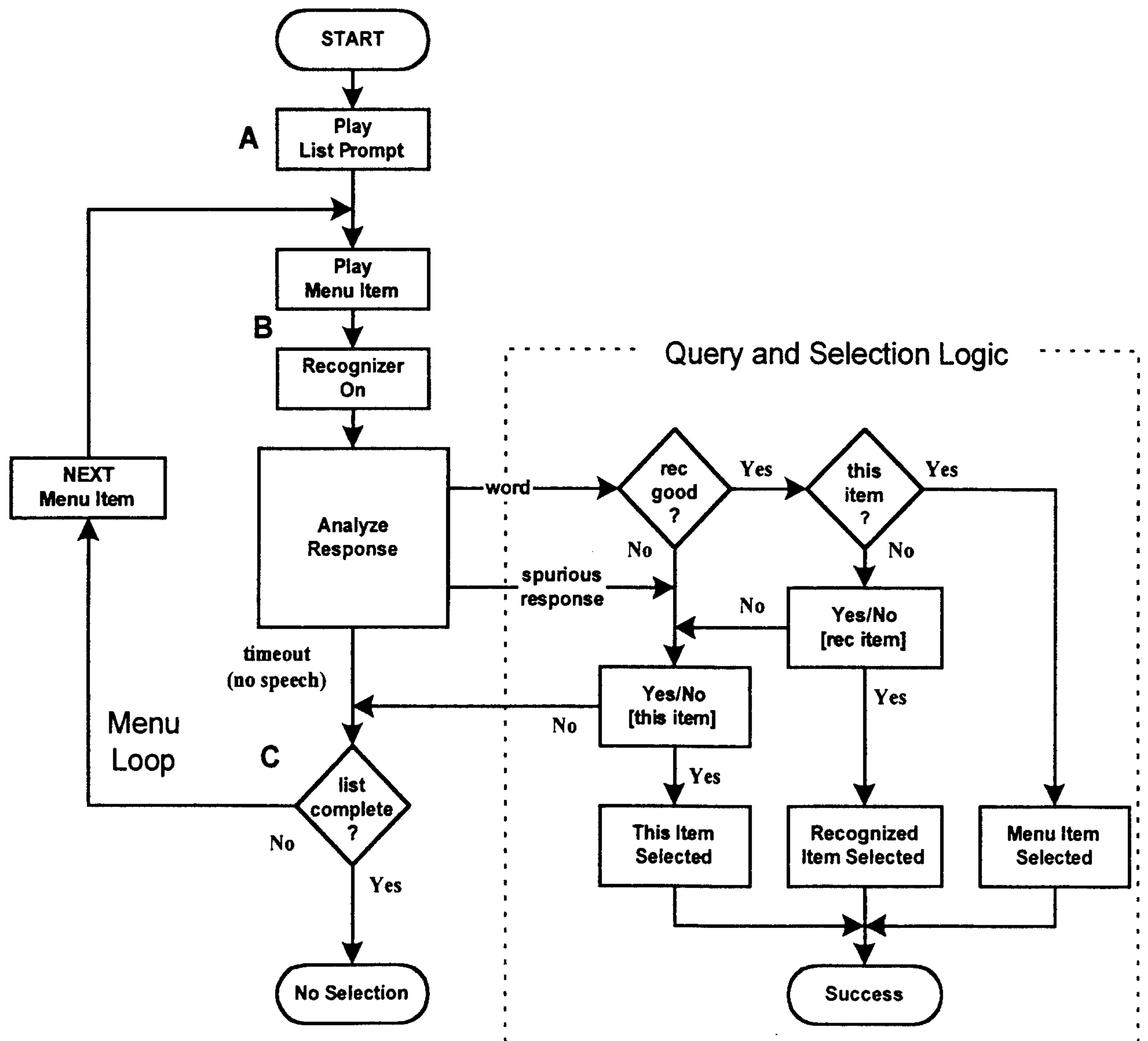


FIG. 12

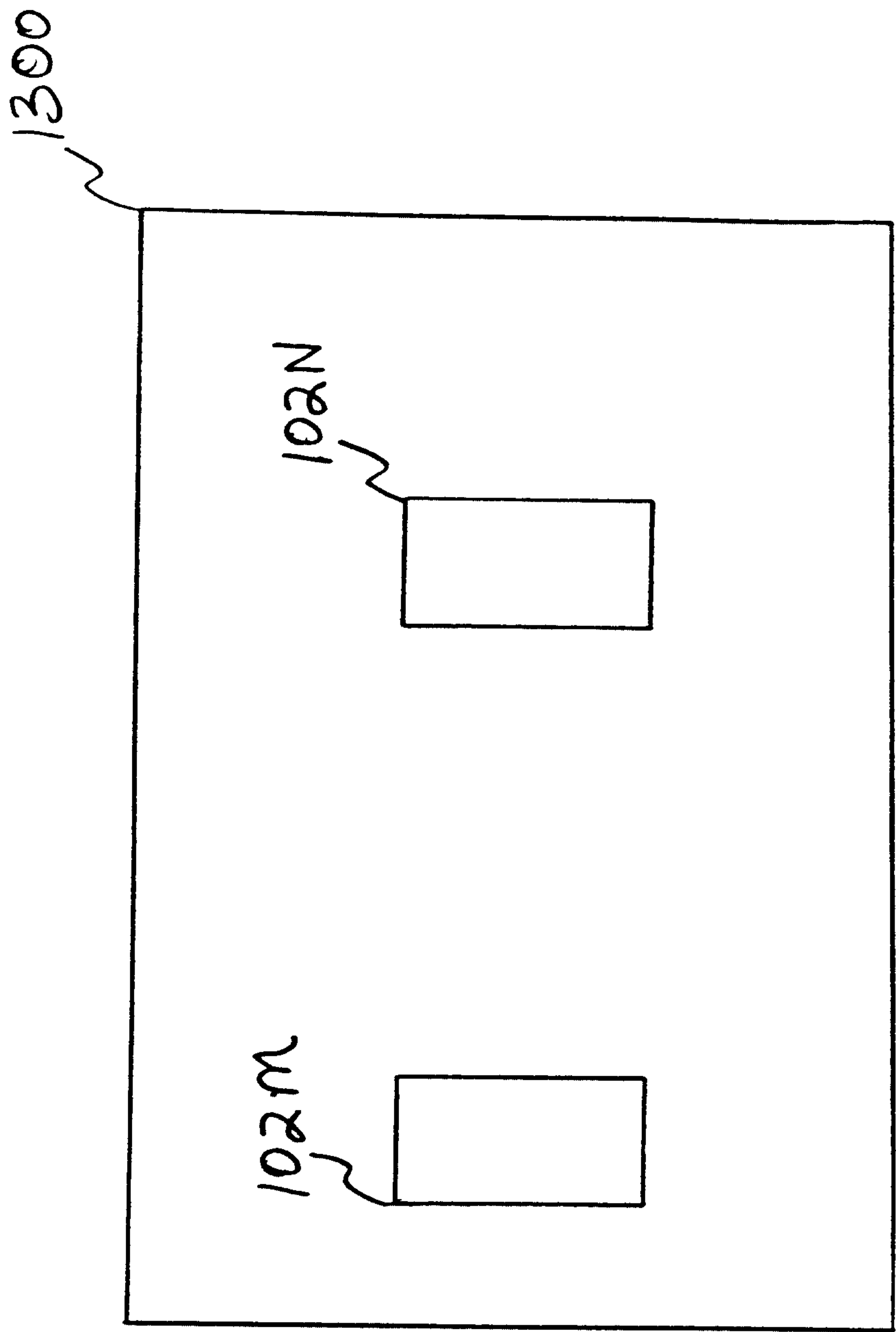


FIG. 13

