CML Microcircuits
COMMUNICATION SEMICONDUCTORS

Application Note

CMX608/CMX618/CMX638 Voice Storage and Retrieval

AN/General/CMX6x8/2 July 2008

Additional Resources	Firmware listing – Download from the CML website using the link that accompanied this on the Application Notes download page

1 Introduction

The ability to store and retrieve voice is used in a wide range of applications. Good voice quality and reasonable memory requirements are key to the successful implementation of this function. Unfortunately, these two requirements are often mutually exclusive.

The CMX608/CMX618/CMX638 RALCWI Vocoder IC family from CML Microcircuits provides near toll-quality speech from bit rates as low as 2400bps. These innovative devices allow longer recording times for a given sized memory device than other coding schemes, all while providing good voice quality.

The purpose of this application note is to illustrate how the CMX6x8 family can be configured to perform voice storage and retrieval. This information is not intended to be a reference design, but is instead intended to act as a “seed” to accelerate development of CMX6x8-based designs.

Forward error correction, FEC, has been enabled for demonstration purposes but its use but is unnecessary where flash memory is used for local storage. This is because modern flash memory uses its own error correction on retrieval. An approximate 33% gain is achieved by disabling FEC, increasing the available recording time in this application to nearly 1 hour on a low-cost flash memory device.

The CMX608/CMX618/CMX638 datasheet and EV6180 Evaluation Kit User Manual should be consulted during review of this application note.

TABLE OF CONTENTS

2 History

Version	Changes	Date
1	New Issue	29-10-07
2	Use of ACK bit in SVCACK clarified. Note added on increased recording length if FEC is disabled.	11-07-08 17-07-08

3 CMX6x8 RALCWI Vocoder Family

CML Microcircuits offers three low data rate RALCWI vocoders; the CMX608, CMX618, and CMX638. These devices are pin-compatible and software-compatible. The notable differences between these products are summarized in the following chart:

Product	Half-duplex or Full-duplex?	Internal or External Codec?	Evaluation Kit
CMX608	Half-duplex	External codec only	EV6180
CMX618	Half-duplex	Internal or External Codec	EV6180
CMX638	Full-duplex	Internal or External Codec	EV6380

Table 1: CMX6x8 RALCWI Vocoder Differences

The CMX618 vocoder IC was chosen for the development of this application note, but either the CMX608 or CMX638 can be used in place of the CMX618 if desired.

4 General Description

This application note describes how the CMX618 can be easily configured to:

- Transform a microphone input signal to a low-rate digital data stream.
- Store this digital data stream into flash memory.
- Retrieve this data when desired.
- Convert the data back into the original voice signal.

The EV6180 and its on-board CMX618 RALCWI Vocoder IC performed all voice processing functions in this application. A Winbond W25X80 8Mbit flash chip was used for voice storage and an Atmel AT89C2051 8051-class processor supervised all functions. The following block diagram depicts the test platform used for this application note:

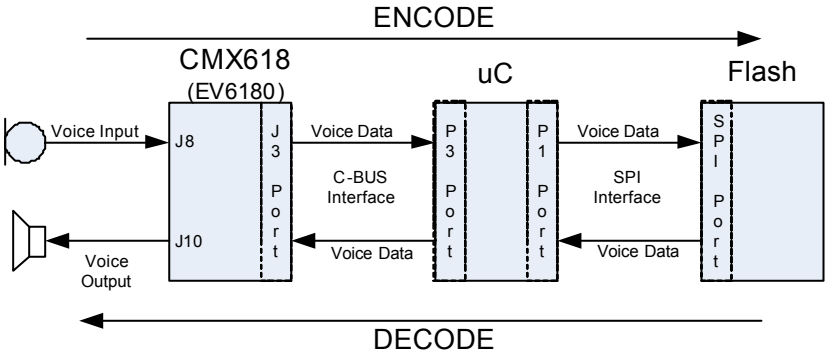

Figure 1: System Block Diagram

Once the system is powered up, the host processor and CMX618 are initialized and the flash is erased in preparation for data writes. The circuitry waits until a switch is pressed, at which time the CMX618 is placed into encode mode and begins to provide 27-byte packets of encoded voice data to the processor every 60ms. The processor writes this data to the flash chip. This process continues until a second switch is pressed, at which time the CMX618 is placed into decode mode and the voice data is extracted from flash and passed to the CMX618 for voice playback.

5 Hardware Description

5.1 CMX618 RALCWI Vocoder IC

The CMX618 is a high-performance, half-duplex vocoder that utilizes the RALCWI (Robust Advanced Low Complexity Waveform Interpolation) algorithm to create near toll-quality speech at very low bit rates. Highlights of the CMX618's feature set are provided below:

- Bit rates = 2400bps or 2750bps
- FEC encoding and decoding (bit rate = 3600bps with FEC enabled)
- Integrated audio codec
- No licensing or royalty payments
- Ancillary audio functions include:
    - Voice Activity Detector
    - Comfort Noise Generator
    - DTMF and Single Tone Regeneration

The CMX618 provides considerable flexibility in its configuration, and the following configuration selections were made for this application:

- Bit rate = 2400bps
- FEC enabled for encoding and decoding
- CMX618 data rate = 3600bps (2400bps voice data + 1200bps FEC data)
- Three 20ms voice frames are assembled into a 60ms voice packet for presentation to the processor.
- CMX618 issues an interrupt when a 60ms voice packet is available during encoding.
- CMX618 issues an interrupt when a 60ms voice packet is required during decoding.

The CMX618 communicates with the host processor over C-BUS, which is a simple serial interface that is very similar to the SPI protocol. The C-BUS interface consists of the following five lines:

- SCLK (serial clock)
- CSN (chip select, active low)
- IRQN (interrupt request, active low)
- RDATA (reply data)
- CDATA (command data)

C-BUS transactions can be easily performed with either a "bit-banging" routine or with a processor's SPI port. The processor in this application did not have an SPI port, so the C-BUS communications were performed with bit-banged routines.

In a typical C-BUS transaction, the host processor will take the following actions:

- Assert CSN.
- Write the address of the desired register on CDATA.
- Host processor can now either:
    - Write the desired configuration data on CDATA, or…
    - Read the desired register contents on RDATA.
- De-assert CSN.

The CMX618 also offers streaming C-BUS registers that allow multiple bytes of data to be exchanged with the processor with just a single register address. More information about the C-BUS interface can be found in the CMX618 datasheet and in application notes on the CML Microcircuits' website.
The EV6180 Evaluation Kit provides an easy to use platform for CMX618-based project development. Audio input and output circuitry, accessible microcontroller interface connections and CMX618 signal test points made the EV6180 a good choice for the development of this application note. The software

package supplied with the EV6180 was not used in this project. Instead, the CMX618 is controlled by the Atmel AT89C2051 microcontroller using code that is described later in this document.

5.2 Atmel AT89C2051 Microcontroller

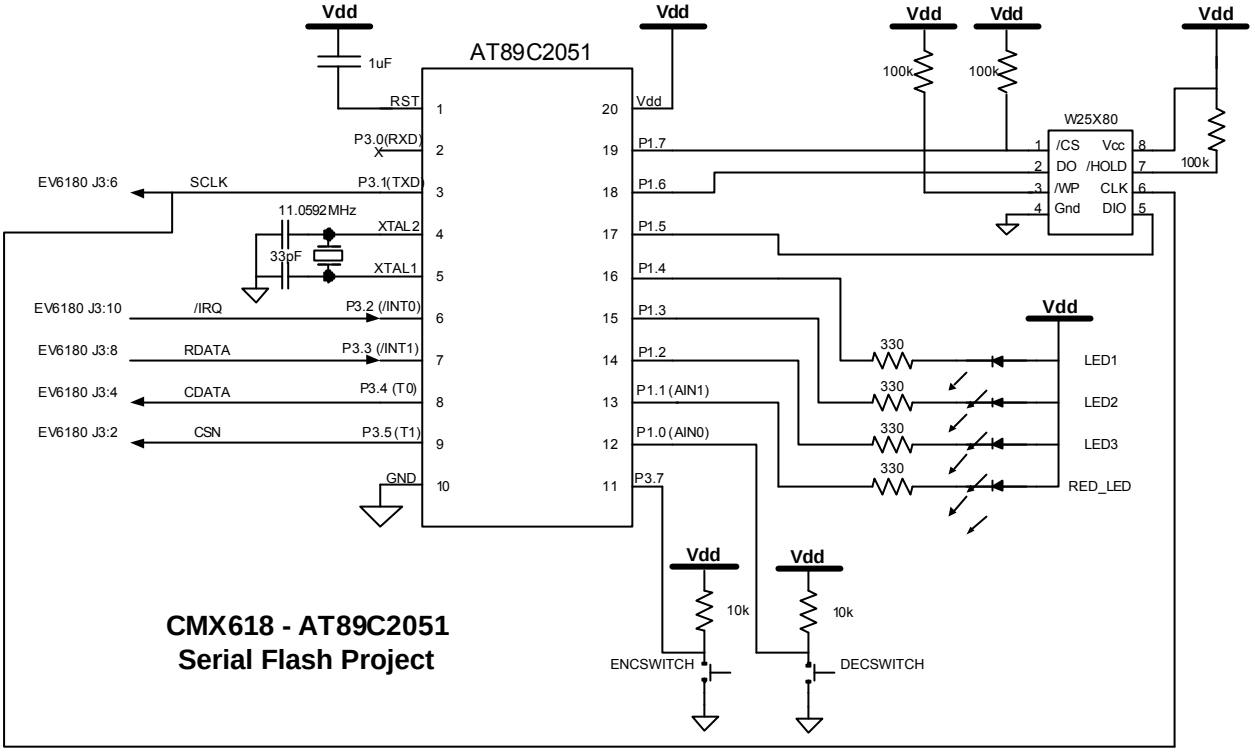

Figure 2: Voice Storage and Retrieval Circuit Schematic

The AT89C2051 microcontroller was selected for this project because of its flexibility, low cost and ample I/O capabilities. This "classic 12-cycle" 8051 processor includes many useful features, including:

- 2k of on-board flash memory for program storage
- 128 bytes of RAM
- 15 I/O lines
- Two 16-bit timer/counters
- Five vector two-level interrupt scheme
- On-chip oscillator
- V_{DD} from 2.7V to 6V

This project required little in the way of processor resources. The 2k of on-board reprogrammable flash memory in the AT89C2051 were more than sufficient for the code requirements of this project (1198 bytes). The C-BUS and SPI communications were performed with I/O lines controlled by bit-banged routines. LEDs were used to aid in debugging, but if the LEDs are removed, five of the processor's I/O lines become available for use in an end application. The AT89C2051 was operated with V_{DD}=3.3V in this project.

5.3 Winbond W25X80 Flash

The Winbond W25X80 8Mbit flash chip was chosen for this project because of its low cost and large storage capacity. With the 3600bps data rate used by the CMX618 in this project, it is possible to achieve 38.8minutes of voice storage with the W25X80! The CMX618 also supports lower data rates, and the corresponding storage time achievable with the W25X80 can be computed as follows:

CMX618 Data Rate	Approximate Storage Time with W25X80
2400bps	58.3 minutes
2750bps	50.8 minutes

Table 2: Voice Storage Times for Various Data Rates

The W25X80 uses an SPI port to communicate with the host processor. The AT89C2051 did not have an SPI port, so the SPI communications were performed with processor I/O lines via bit-banging.Since the W25X80 is a flash chip, writes to the device must be made at a page level. The W25X80 is divided into 4096 pages, and each page has 256 bytes of storage. The CMX618 is configured in this application to provide data in 27-byte packets but since 256 is not integer divisible by 27, some accommodation must be made. For this project, each flash page is written to 9 times, so (9 x 27) = 243 bytes written to each 256 byte page. This translates to 94.9% utilization of the flash. With 94.9% utilization and 3600bps data rate, the 8Mbit capacity of the flash provides 36.9 minutes of voice storage time.

6 Firmware Description

6.1 General Comments

The firmware for this project was written in C and compiled within the Keil uVision 3 Development Environment. This program was also verified with a different microcontroller (AT89S53) which utilized an SPI port. In that code, the SPI bit-banged routines were replaced with SPI-specific routines.

6.2 Initialization

The program flow is depicted in Figure 3:

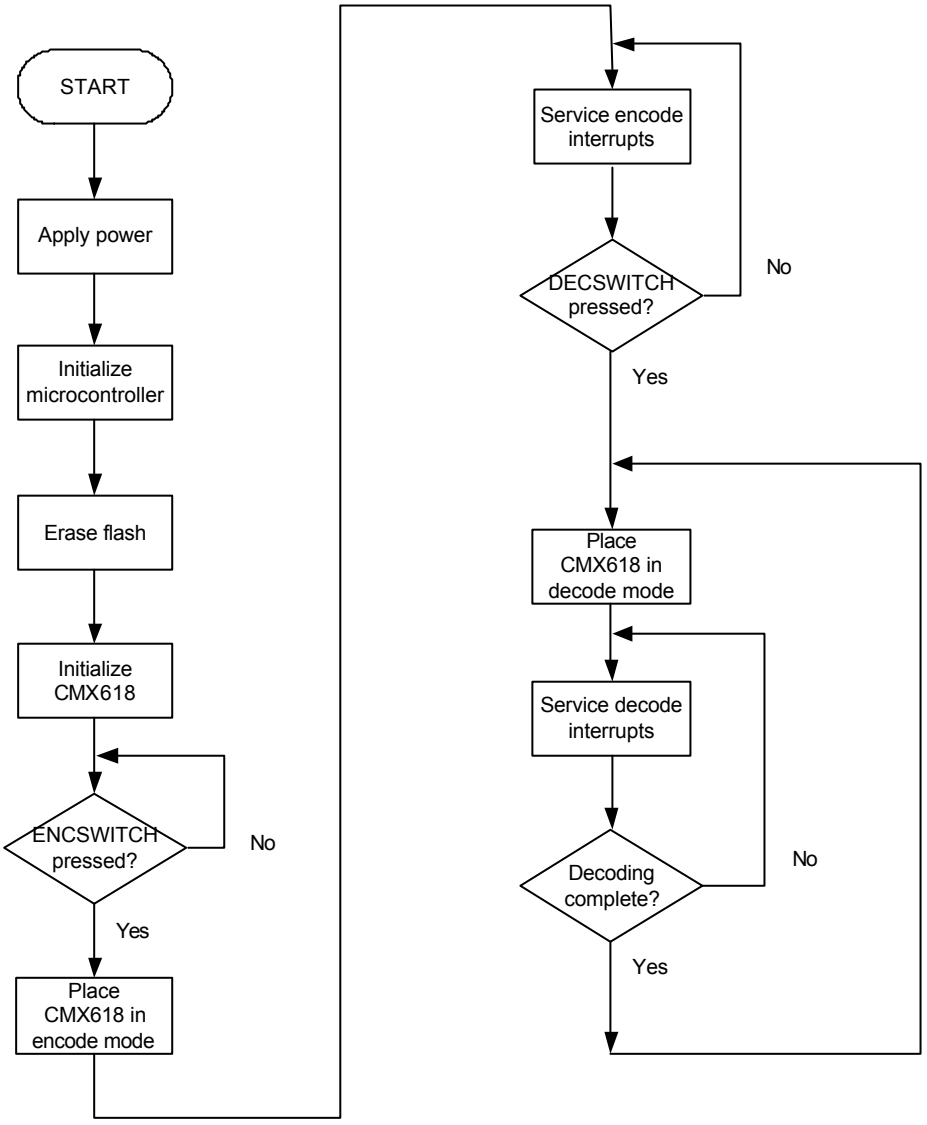

Figure 3: Program Flow

The firmware begins with initialization of the AT89C2051 processor. Within this initialization routine, Timer 1 and EXO interrupts are enabled, Timer 1 is configured for 16-bit timer operation, and input pins are configured as needed.

After the processor is configured, the “erase_flash” routine is called to erase the flash and prepare it for data writes. This process begins with a “write enable” command and a “chip erase” command passed to

the flash. The flash typically requires 10 seconds for a full chip erase, so a 20 second timer is started to prevent a flash problem from causing the routine to lock up. A red LED is lit if the 20 second timer expires. The W25X80 Status register is polled to monitor its "busy" bit. When the chip erase command is completed (busy bit=1), control reverts to the "main" routine.

Next, the "main" routine sets a state variable (state=0) to indicate that the CMX618 is in configuration mode. (The state variable in this project is primarily used to indicate which CMX618 Status register bits need to be checked during the CMX618 interrupt service routine.)

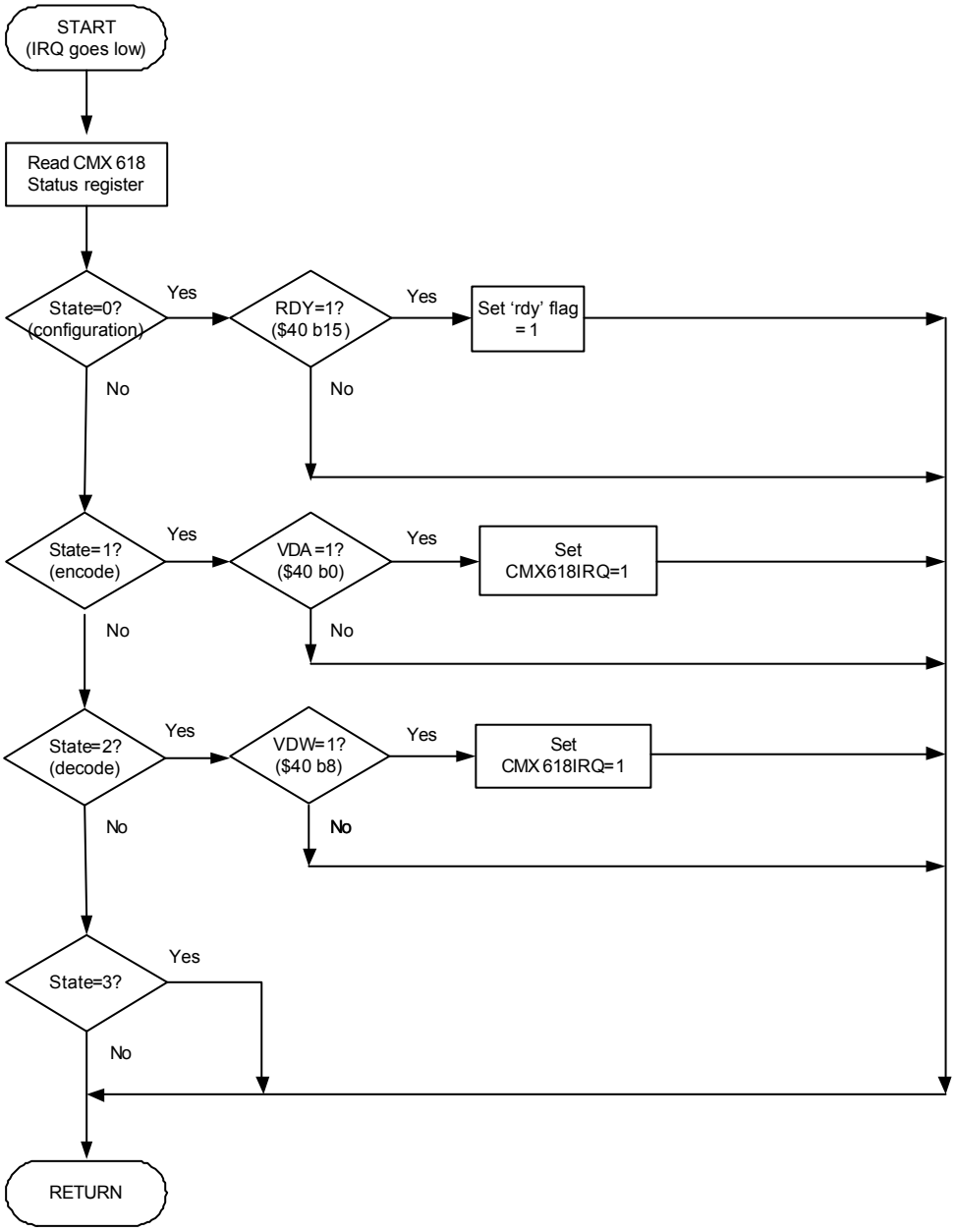

Figure 4: CMX618 Interrupt Service Routine

A call is made to the "initialize_618" function to configure the CMX618. Since the CMX618 issues a RDY interrupt (Status register, $40, b15) after power is applied, the CMX618 initialization routine begins by reading the Status register to clear that interrupt. (Note: The RDY interrupt is not masked when power is applied). A software reset is performed and the routine waits for the RDY bit to be set, indicating that the

CMX618 is ready for new commands. When the RDY bit gets set in the Status register, the CMX618 issues an interrupt and program flow vectors to the CMX618 interrupt service routine (ISR). The ISR flow is illustrated in the Figure 4:

NOTE: The Status register's RDY bit is set to "1" to indicate that the device is ready to accept additional commands. Additionally, the SVCACK register's ACK bit ($2E b0) will be set when the following time-consuming operations have been completed:

- Writing to VCFG register.
- Writing to VCTRL register.
- Writing to EXCODECCONT register.

The ACK bit only has meaning in the following circumstances:

- A time consuming operation (from list above) has been performed.
- RDY bit has been set to 1 to indicate completion of that operation.

The ACK bit will get set to "1" **before** the RDY bit gets set to "1". This means that when one of these operations has completed and RDY=1, the ACK bit **must** also be set to 1, otherwise an error has occurred. If ACK=0 in this situation a General Reset should be issued and the device reconfigured. A check of the ACK bit can be useful during code development for debugging purposes but should not be required for final code deployment.

The CMX618 Status register is read to clear the interrupt condition upon entry into the ISR. Since the state variable is set to "configuration mode" (state=0), a check is made for the RDY bit, a flag is set if required, and the program exits from the ISR.

When control returns to the "initialize_618" routine, configuration data is passed to the POWERSAVE and VCFG registers. Configuration of the POWERSAVE register enables the CMX618 internal codec and bias circuitry. The VCFG write configures the CMX618 for:

- 2400bps coding
- 1200bps FEC
- 3x 20ms frames grouped into 60ms data packets
- "Hard bits" expected by the decoder.

A 100msec timer is started to ensure that the CMX618 V_{BIAS} has settled, and commands are sent to the AIG and AOG registers to configure input and output audio gain levels. High and low "watermarks", corresponding to CMX618 codec sample level thresholds for IRQ purposes, are programmed into the VDWHLWM register. The high watermark (160 samples) and low watermark (72 samples) values were taken from the CMX618 datasheet, "Event Driven/Method 1" example for multiple frame packet decoding. Control then reverts back to the "main" routine.

6.3 Encode Operation

When ENCSWITCH is pressed, a write is made to the CMX618 to enable the "RDY" interrupt, and the CMX618 encoder is enabled. After the RDY bit has been set to signify that the CMX618 can accept new commands, the IRQENAB register is adjusted so that the only condition that can cause a CMX618 interrupt is "vocoder data available" (VDA). The state variable is changed to the "encode mode" value (state=1), and the program waits for the DECSWITCH to be pressed.

The microphone input is applied to the CMX618 encoder and a 27-byte packet of voice data is made available to the processor every 60ms. When a voice packet is ready, the CMX618 IRQ line goes low and the CMX618 interrupt service routine (ISR) is entered. The program flow following an encode interrupt is illustrated in Figure 5:

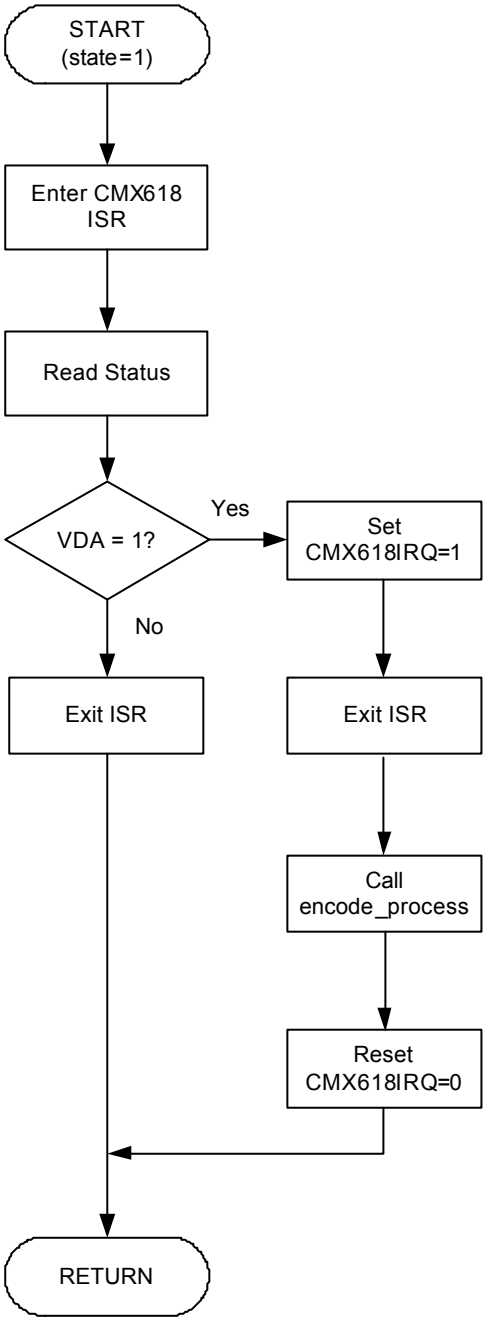

Figure 5: Program Flow after Encode Interrupt

The CMX618 Status register is read to clear the interrupt condition. The state variable determines which Status register bits are checked; during encode mode, a test is performed for the VDA bit. If the VDA bit is set, the CMX618IRQ flag is set to 1. Control reverts to the "main" routine after the VDA bit is checked.

The "main" routine checks the status of CMX618IRQ, and if that flag is set to 1, the "encode_process" function is called. The first task within "encode_process" is to read out the 27-byte voice packet from the CMX618's ENCFRAME register and store this information in a global array variable. The data is then written to flash by calling the "program_flash" routine. The flash uses a 3-byte address for write and read memory locations, so the desired flash address is passed to the "program_flash" function.

The “program_flash” function begins by sending a “write enable” command to the flash. A “page program” command is sent to the flash next along with the 3-byte address of the memory location where the write is to begin. The 27-bytes of voice data are sequentially written to flash, and the “program_flash” function is completed.

Control reverts to the “encode_process” routine where the flash memory address is incremented, and once that is complete, the program moves back to the “main” routine.

This process repeats itself continuously until DECSWITCH is pressed, at which time operation switches from encode mode to decode mode.

6.4 Decode Operation

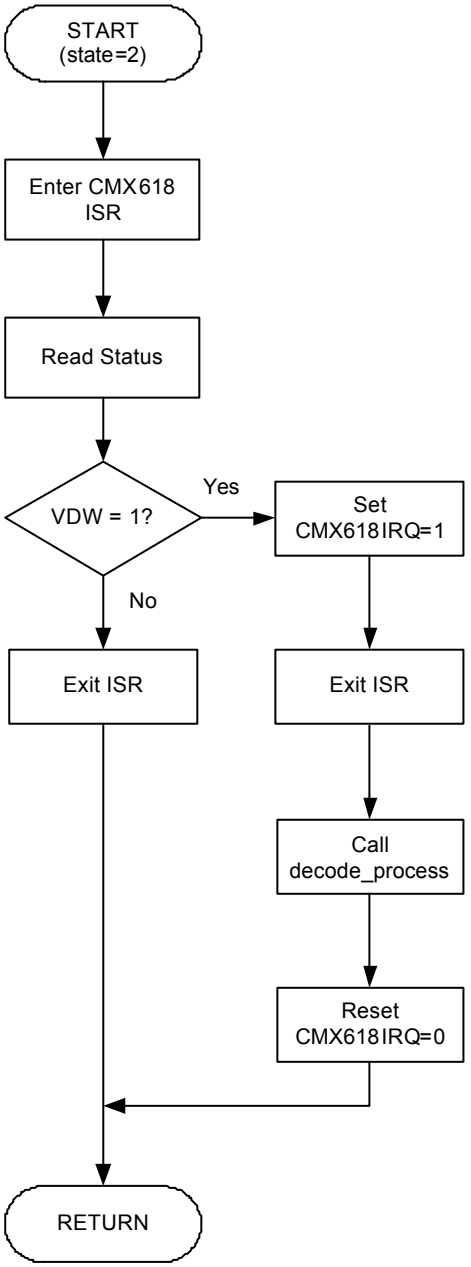

Figure 6: Program Flow after Decode Interrupt

After the DECSWITCH is pressed, an infinite loop containing the decode operation is entered. The "decode_complete" flag and decode flash addresses are reset. The state variable is changed to its "configuration" value (state=0) to ensure that the RDY bit will be checked within the CMX618 ISR. The CMX618 IRQENAB register is adjusted to allow only RDY interrupts to occur.

A write is made to the CMX618 VCTRL register to enable the decoder and the "vocoder data wanted" (VDW) interrupt function. Once these commands have been processed (RDY=1), the IRQENAB register is again adjusted, this time to allow only the VDW interrupt to occur. The state variable is now changed to its "decode" value (state=2).
Data is read out of the flash using the "read_flash" function call to provide voice data to the CMX618 to begin the decoding process. The "read_flash" routine is very similar to the "program_flash" function. A "read data" instruction and the address of the memory location for the read, are written to the flash. A simple "for" loop is used to sequentially read the data out of memory for storage in a global array variable. Control reverts to the "main" routine and the memory address value is incremented in preparation for the next read. The voice data is then written to the CMX618 decoder's DECFRAME register.

The act of reading the flash can take a few milliseconds. To ensure that data is available for decoding when requested by the CMX618, another read of the flash is performed before the CMX618 issues its "vocoder data wanted" interrupt.

Once the decoded voice samples within the CMX618's internal buffer fall below the previously programmed "low watermark" value, the CMX618 will issue an interrupt with the Status register's VDW bit set. Program flow vectors to the CMX618's interrupt service routine as illustrated in the following Figure 6:

The ISR reads the CMX618 Status register, thus clearing the interrupt. Since the state variable is set to "decode mode" the VDW bit is checked, and if it is set, the CMX618IRQ flag is set to 1. Control is passed back to the "main" routine once the VDW bit has been checked.

The status of the CMX618IRQ flag is checked upon re-entry into the "main" routine, and if it is set to 1, the "decode_process" function is called. The voice data previously read out of the flash is written to the CMX618's DECFRAME register upon entry into "decode_process". Next, a new 27-byte voice data packet is read out of the flash with a call to the "read_flash" function. This data will be immediately available when the next CMX618 VDW interrupt occurs. The memory address is incremented and checked against the highest address used for the encode function. If the decode address equals the encode address, all data has been read out of flash and the "decode_complete" flag is set to 1. Control reverts to the "main" routine.

The program continues to service interrupts until all of the decoded data has been read out of the flash. Once the "decode_complete" flag is set to 1, the decode loop restarts and the encoded voice is again played out through the speaker.

7 Conclusion

Voice storage and retrieval (VSR) is a useful function that is required in many applications. The CMX618, a member of CML's family of RALCWI vocoders, can serve in a VSR application because of its low-bit rate and good voice quality. This document has illustrated one method of performing the VSR function with the CMX618. It is hoped that the information in this application note will help the designer with their CMX618-based designs.

CML Microcircuits (UK) Ltd COMMUNICATION SEMICONDUCTORS	CML Microcircuits (USA) Inc. COMMUNICATION SEMICONDUCTORS	CML Microcircuits (Singapore) Pte Ltd COMMUNICATION SEMICONDUCTORS	
		Singapore	China
Tel: +44 (0)1621 875500 **Fax:** +44 (0)1621 875600 **Sales:** sales@cmlmicro.com **Tech Support:** techsupport@cmlmicro.com	**Tel:** +1 336 744 5050 800 638 5577 **Fax:** +1 336 744 5054 **Sales:** us.sales@cmlmicro.com **Tech Support:** us.techsupport@cmlmicro.com	**Tel:** +65 67450426 **Fax:** +65 67452917 **Sales:** sg.sales@cmlmicro.com **Tech Support:** sg.techsupport@cmlmicro.com	**Tel:** +86 21 6317 4107 +86 21 6317 5910 **Fax:** +86 21 6317 0243 **Sales:** cn.sales@cmlmicro.com.cn **Tech Support:** sg.techsupport@cmlmicro.com
- www.cmlmicro.com -			