Application Report
SLAA377–December 2007

CEC-to-I²C Bridge With the MSP430

Chris Sterzik — *MSP430 Applications*

ABSTRACT

This application report shows how to use the MSP430 to bridge data from an I²C™ bus to the Consumer Electronics Control (CEC) bus described in the High-Definition Media Interface (HDMI) specification 1.3. For this CEC application example, the MSP430 resides within the television. The scope of this application requires the MSP430 to handle the physical layer and data link layer (medium access control) responsibilities.

Contents

List of Figures

1 Introduction

High-Definition Media Interface (HDMI) incorporates a single-wire interface called Consumer Electronics Control (CEC) in order to support 'one touch' control features in newer audio/visual systems. CEC is not a conventional bus and is not readily found on microcontrollers like the more common protocols I^2C or SPI. One means of supporting CEC is the use of an external microcontroller dedicated to handling the CEC protocol and communicating the information to the host processor with the more common protocol I^2C. The MSP430F20x2 and MSP430F20x3 devices from Texas Instruments provide both the traditional I^2C support and the integrated peripherals needed for CEC. This document describes using the MSP430F2013 to bridge between I^2C and the CEC protocol within a television or monitor.

2 Context of CEC Within HDMI

HDMI is known for its provision of high-speed lanes for transmitting high-definition video content. HDMI also provides a networking structure for audio/visual components to communicate with each other. Figure 1 shows a typical HDMI network where each device has a physical and logical address. The HDMI standard defines the logical address, while a parent/child relationship defines the physical address. The parent device passes the physical address as part of the Extended Display Identification Data (EDID) to the child device. For example, the A/V receiver in Figure 1 receives a physical address of 2.0.0.0 from the TV and, in turn, passes 2.1.0.0, 2.2.0.0, and 2.3.0.0 to the DVD, D-VHS, and STB respectively. When the DVD player in Figure 1 is connected, the DVD broadcasts the physical and logical address on the CEC bus. This alerts all devices, including the TV, to the DVD's presence. The TV can then, in turn, address the DVD player via the CEC bus and send commands like Play or Stop.

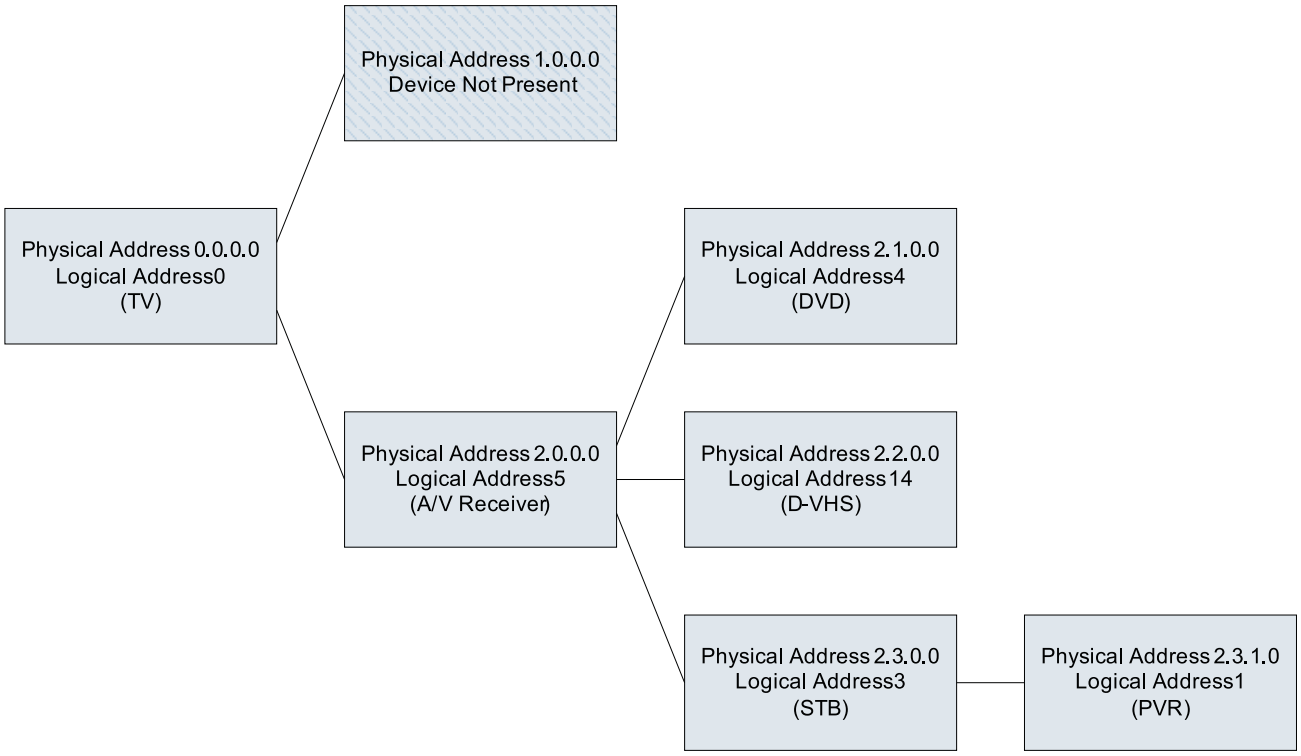

Figure 1. Physical Addressing

3 Using the MSP430

This application report is written from the perspective of the television, the component with logical address 0 in Figure 1. Within the television (see Figure 2), the MSP430 manages the physical-layer translation from I^2C to CEC. Included with this translation task is the screening of destination addresses in the CEC message. The MSP430 should only pass along messages with the TV address, 0x0h, or the broadcast message, 0xFh. Finally, MSP430 must also communicate with a host processor, which manages the application-layer functions of CEC.

While the MSP430 supports both the I^2C and CEC buses, the communication is half-duplex. That is, data flows in both directions, but only one direction at a time. The end application is not intended to receive or transmit data simultaneously on both the CEC and I^2C buses.

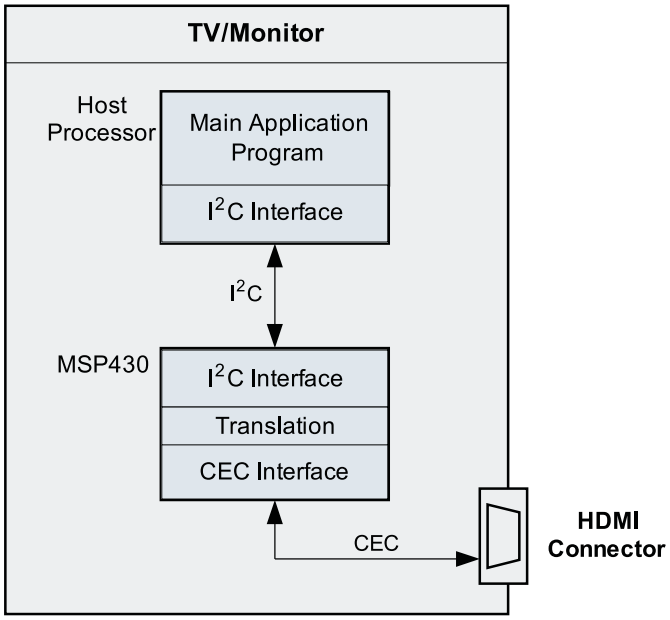

Figure 2. CEC Functional Partition Between Host and MSP430

The I^2C communication between the MSP430 and the host processor is a multi-master; both the MSP430 and the host processor are masters. In order to prevent a loss of data, the MSP430 has the higher I^2C bus priority (i.e., the lower numerical address). The USI module on the MSP430 provides the I^2C communication.

The I/O port, P1.0, of the MSP430 provides the CEC communication. The interrupt associated with P1.0 captures a rising or falling edge, while Timer_A is used to monitor the received bit times. Timer_A also establishes the timing of P1.0 during data transmission.

4 Consumer Electronics Control (CEC)

CEC is a feature defined within supplement 1 of the HDMI 1.3 standard. CEC, like I^2C, is a bidirectional multi-access bus. CEC is different from I^2C in that CEC is a single-wire interface. The CEC communication is made up of a start bit, an 8-bit header, 8-bit data, an end-of-message (EOM) bit, and an acknowledge bit.

4.1 Start Bit

As shown in Figure 3, the start bit is simply a single bit that meets the timing requirements for the low and total duration. When the start bit meets both timing requirements, the CEC receiving device needs to process the incoming data. If the start bit does not meet both requirements, then the receiver ignores the data. This prevents the receiver from processing noise as data.

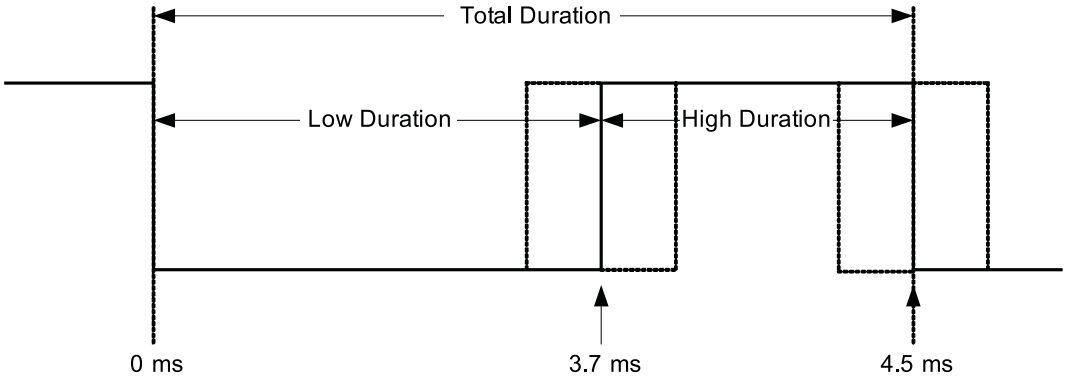

Figure 3. CEC Start Bit

4.2 Data Bits

Figure 4 shows the timing of the CEC data bit. The example code checks the timing requirements of both the rising and falling edges received as well as determines the sampling time. Since there is no clock, the duty cycle or low duration time of the bit determines logic 1 or 0. The total bit time is 2.4 ms, which provides a data rate of about 416 bits/second.

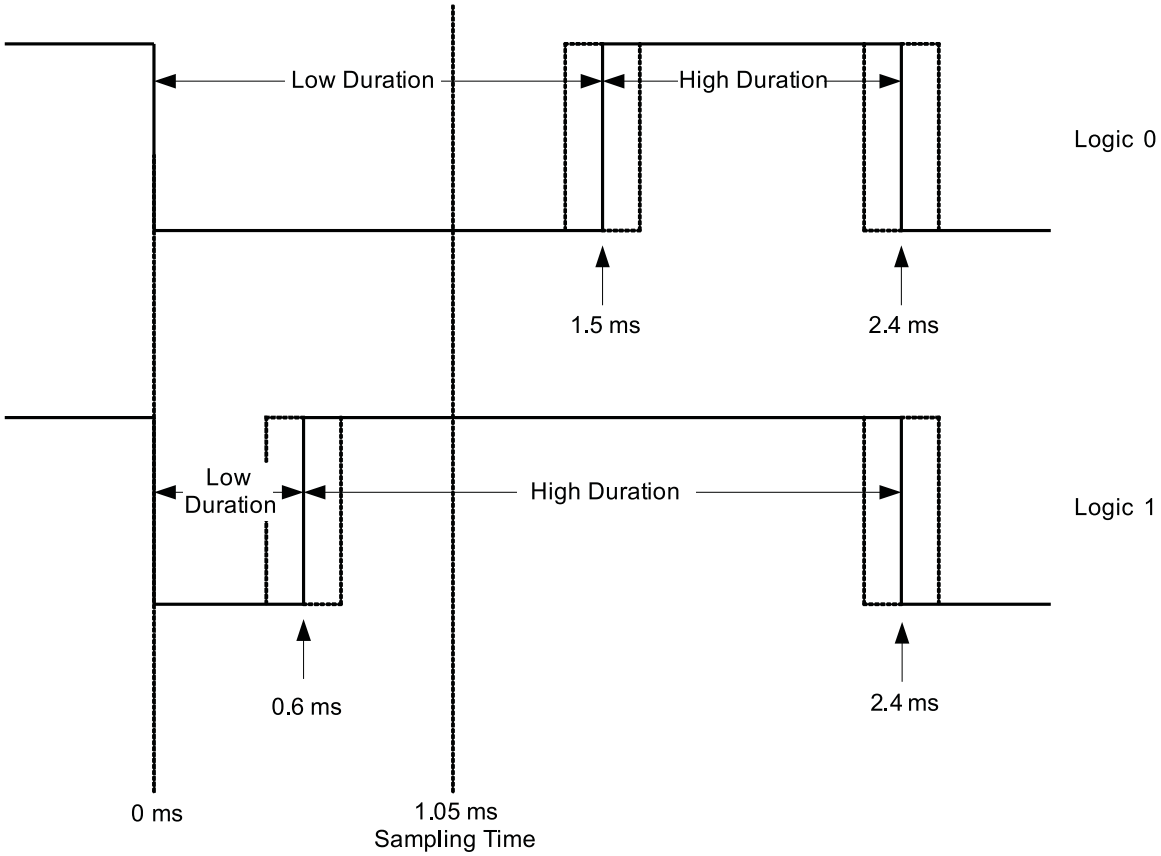

Figure 4. CEC Data Bit

4.3 Header/Data Block

The header and data blocks within CEC are composed of 10 bits. The first 8 bits contain the addressing, operation code (opcode), or operand. The last two bits are the End of Message (EOM) bit and the Acknowledge bit.

4.3.1 Addressing

The header block is the only block with addressing and is the first block of the CEC transmission. The first four bits indicate the source or "initiator" of the transmission. The second four bits identify the destination or intended recipient. CEC allows for both direct messaging and message broadcasts. The TV is looking for messages that have destination addresses 0x0h or 0xFh. These addresses represent the TV logical address and a broadcast message, respectively.

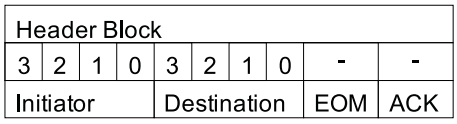

Figure 5. Header Block

4.3.2 Opcodes and Operands

The 8-bit field of the data block represents the opcode or the operand. The HDMI standard defines the opcodes and the possible number of operands for each message. The number of operands depends upon the message. Since the standard provides for manufacturer-defined messages, the number of operands can vary. Disregarding the manufacturer-defined opcodes, the maximum number of possible operands is 15. Figure 6 shows an example CEC message. For this example, a DVD player is broadcasting to all devices on the CEC bus the DVD physical address (Figure 1), the logical address (via the header), and the device type.

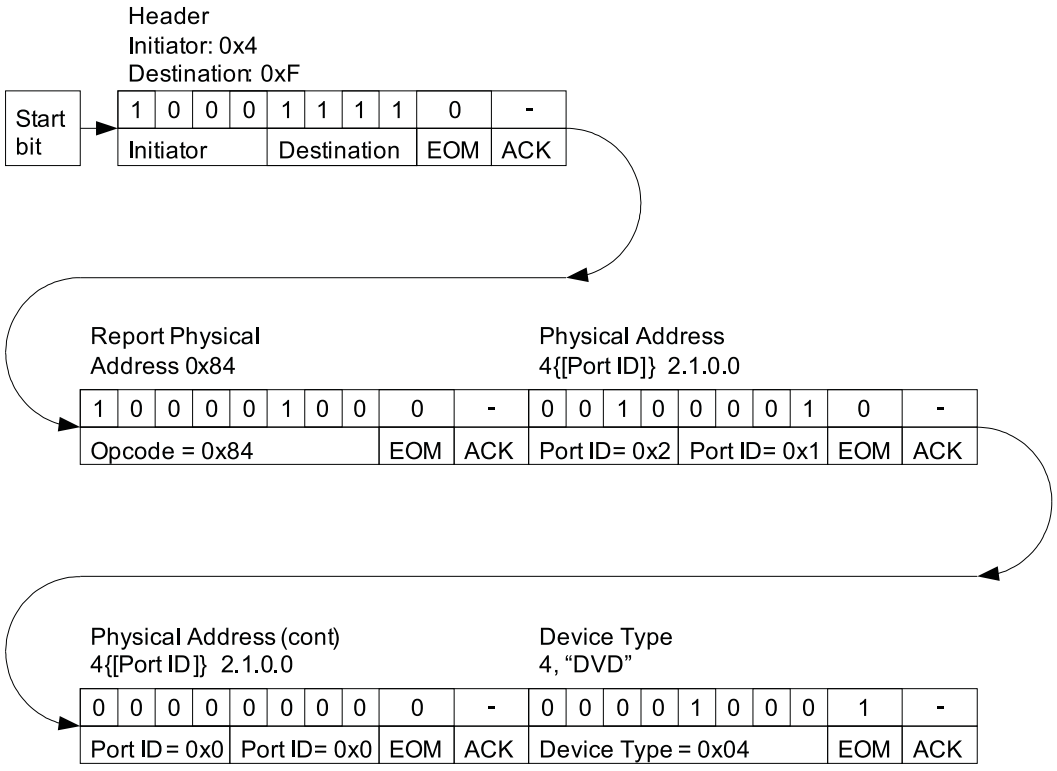

Figure 6. Example CEC Message

4.3.3 End of Message (EOM)

The EOM bit is set only in the last transmitted data block. In the example shown in Figure 6, only the last data block, Device Type, contains a set EOM bit. Since the number of operands is unknown to the MSP430, the MSP430 depends upon the transmission of the I²C stop bit or the CEC EOM bit to identify the end of the message.

4.3.4 Acknowledge

The acknowledge bit is the means for the receiver to acknowledge that it has successfully received a transmission. This is similar to I²C where the transmitter releases the bus and samples the bus (as defined in Figure 4) to determine if the receiver has pulled the bus low.

5 I²C

One can find a thorough description of I²C in the I²C specification. The MSP430 user's guide and the MSP430F20xx I²C library also provide information about implementing I²C with the MSP430 USI module. See References for more information.

6 I²C-to-CEC Translation

When the host processor wants to send data to another node on the CEC bus, it sends the data via I²C to the MSP430. The MSP430 translates the data to CEC and transmits it. The MSP430 repackages the data in the CEC format, as shown in Figure 7. When the USI module detects a stop bit on the I²C bus, the MSP430 then transmits over the CEC bus by emulating the protocol with digital I/O. This section describes the events involved in this process.

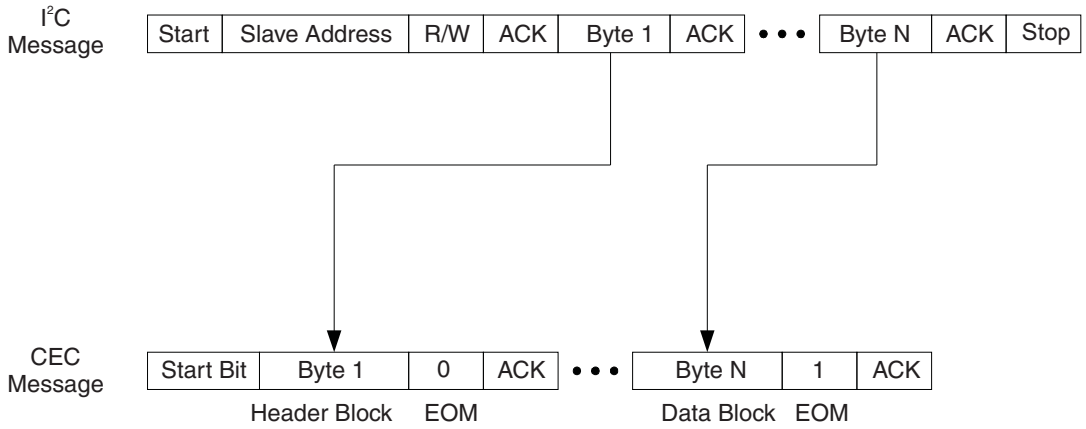

Figure 7. Mapping an I²C Message to a CEC Message

6.1 I²C Receive Mode

The MSP430 waits in LPM4 for the reception of either a CEC start bit or a valid I²C address byte. Upon reception of an I²C start bit, the MSP430 proceeds by verifying the 7-bit address. If the address does not match the address assigned to the MSP430, then the MSP430 ignores the I²C message and remains in LPM4. If the address does match, then the proceeding data bytes are stored in memory and retransmitted in the 8-bit payload of the CEC data transmission. The first byte following the I²C address is the CEC header block. The remaining bytes contain the order appropriate CEC data blocks. The stop bit identifies the end of the I²C data message and in this application the end of the CEC data message.

The USI module does not generate an interrupt when a stop bit occurs. Therefore, the following code, found in the main loop, polls for a stop bit and provides the transition to the CEC transmit portion of the code.

```
__bis_SR_register(LPM4_bits+GIE);       // Wait in LPM4 until interrupt for
                                        // I2C or CEC receive
__no_operation();                       // For debug purposes
while(Pldcnt)
{ /* If Pldcnt is true, then this is the I2C to CEC case */
  if (USICTL1 & USISTP)                 // Check for the Stop flag
  { /* Finished I2C data reception and moving on to CEC transmit */
    USICTL1 &= ~(USIIE+USISTTIE);       // Disable USI interrupts
    USICNT &= 0xE0;                     // Clear USI bit count
    P1DIR |= BIT0;                      // Set output Low to generate the
                                        // falling edge of the start bit
    TACTL |= TACLR;                     // Reset timerA
    TACTL = TASSEL_2 + ID_3 + MC0;      //SMCLK/8=62.5kHz, Up mode
    TACCR0 = Sbit_Nom_LD;
    Timer0_State = 2;
    Pldsize = Pldcnt;                   // Record the number of bytes rxd
    Pldcnt=0;                           // Initialize variables
    Retry = 0;
    TACCTL0 = CCIE;                     // Enable TimerA0 Interrupt
  }
}
/* Hold CPU here until the transmission of the message is completed. */
/* Upon completion, or failure to acknowledge reset interrupts and   */
/* return to LPM4                                                    */
__bis_SR_register(LPM0_bits);
__no_operation();                       // For Debug
```

Both the CEC receive portion of code and the I²C receive portion of code take the MSP430 out of LPM4. The difference is that the I²C receive portion remains in LPM4 until the after first data byte while the CEC receive portion leaves LPM4 at the beginning of the start bit. Therefore during an I²C receive, the Pldcnt variable has been incremented before exiting LPM4 and the MSP430 will execute the code contained within the while loop. Once within the while loop, the code checks for the USISTP flag to be set. The stop bit signals to the MSP430 that the I²C data reception is complete. At this time, the MSP430 disables the USI interrupt, stores in the variable Pldsize the number of bytes received, sets P1.0 low to begin the transmission of the CEC start bit, and sets TACCR0 to time the low duration of the start bit. Once the timer has expired for the low duration, Timer0_State 4 will set the start bit high duration and the next timer state.

Figure 8 shows how to configure the MSP430 Experimenter's Board to demonstrate the CEC to I²C and I²C to CEC translation. The MSP430F2013 is used for demonstration purposes, but the application code can work with other MSP430 devices that have a USI module and at least 2k of memory. The application code developed with IAR is provided with this report.

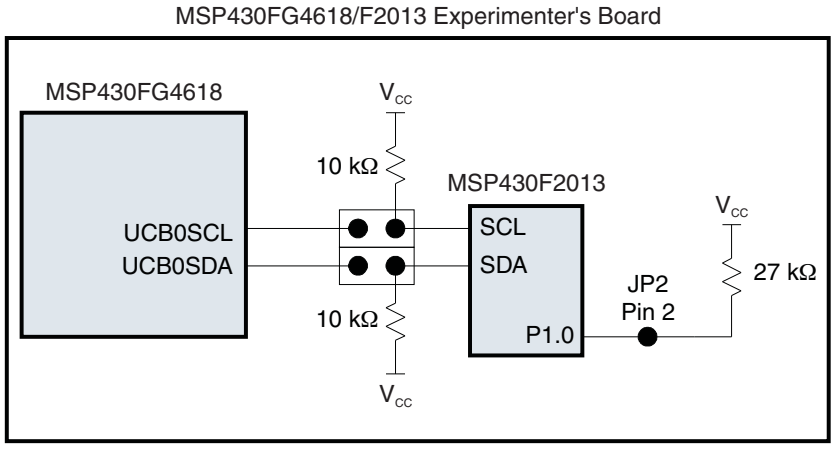

Figure 8. Test Implementation With the MSP430 Experimenter's Board

Figure 9 shows message traffic on the I²C lines SCL and SDA, channel 1 and channel 2, respectively, between the host MSP430FG4619 to the MSP430F2013. Channel 3 is the CEC message traffic between the MSP430F2013 and the CEC bus. The upper window in Figure 9 includes both the message transmitted to a DVD player and the DVD player's response. The lower window is an expansion of the I²C message, which includes the I²C received by the MSP430F2013 and the start of the CEC message transmitted by the MSP430F2013.

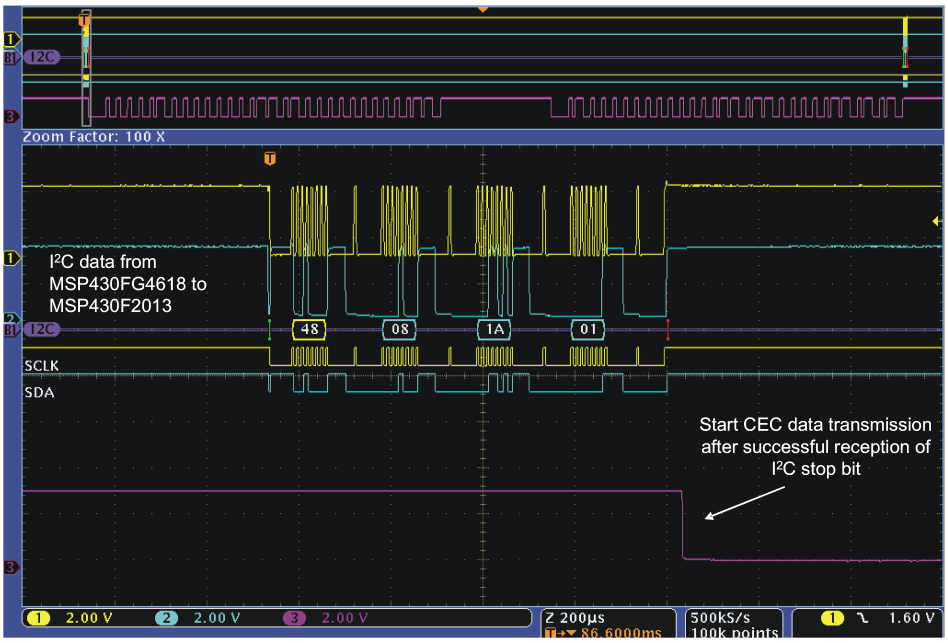

Figure 9. I²C Data Reception From Host

6.2 CEC Transmit Mode

The transmit mode is responsible for sending the start bit, the header block, the data blocks, the proper EOM flag, and for checking the acknowledge bit from the receiver. The initiator and destination information in the CEC block header are taken from the first I²C data byte. This is the first I²C byte after the I²C address. The flow chart in Figure 10 shows how to handle the EOM and retransmissions after a NACK. Once the CEC data transmission is complete, the MSP430 enters LPM4 and waits for either CEC or I²C bus activity.

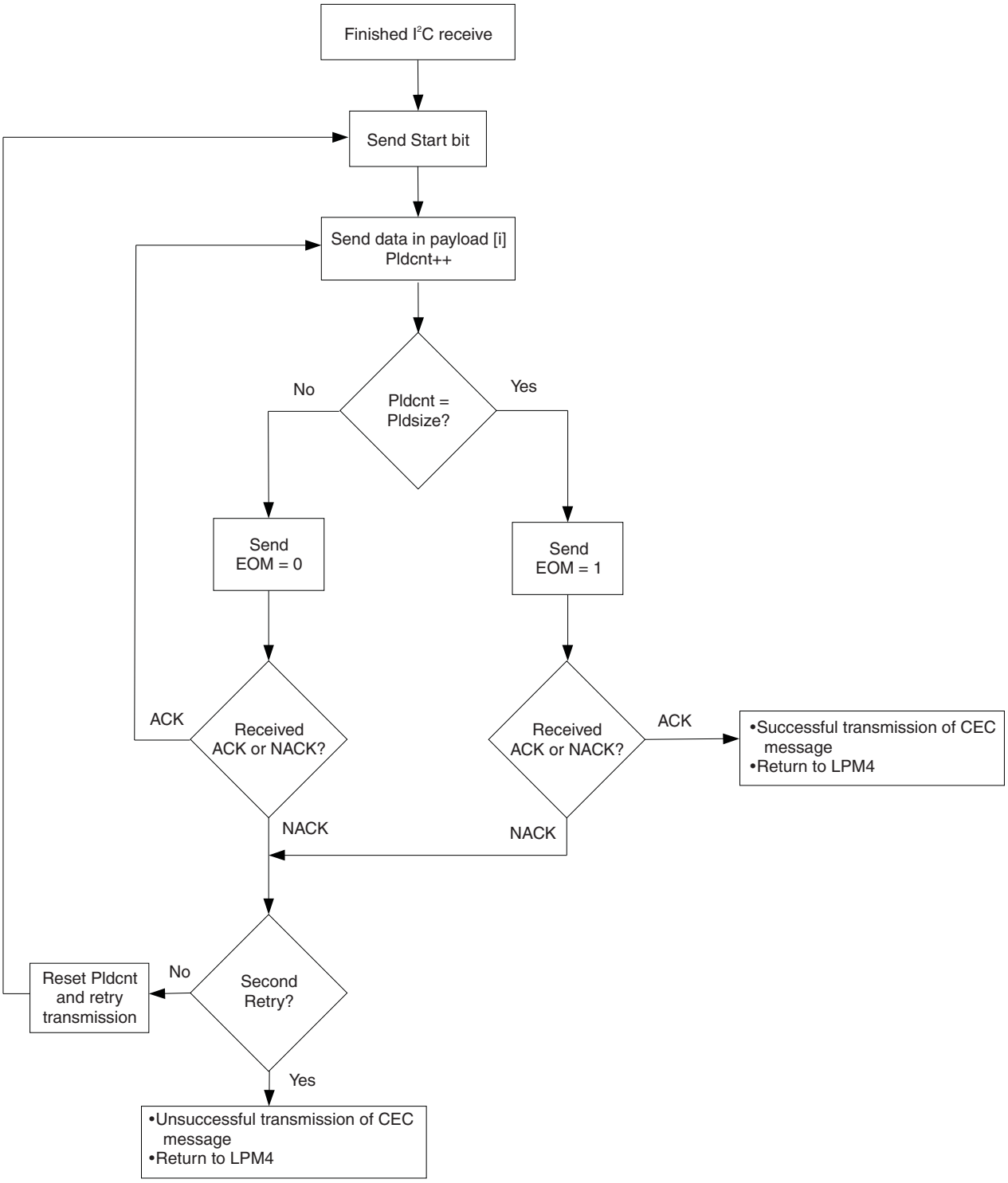

Figure 10. CEC Transmit Flow Chart

The bit banging of array Payload[i] described in Figure 10 is done with P1.0. The example code reads from the array and the appropriate low and high period duration is then set. Figure 11 shows the bit timings for a logic 0 or logic 1 CEC bit.

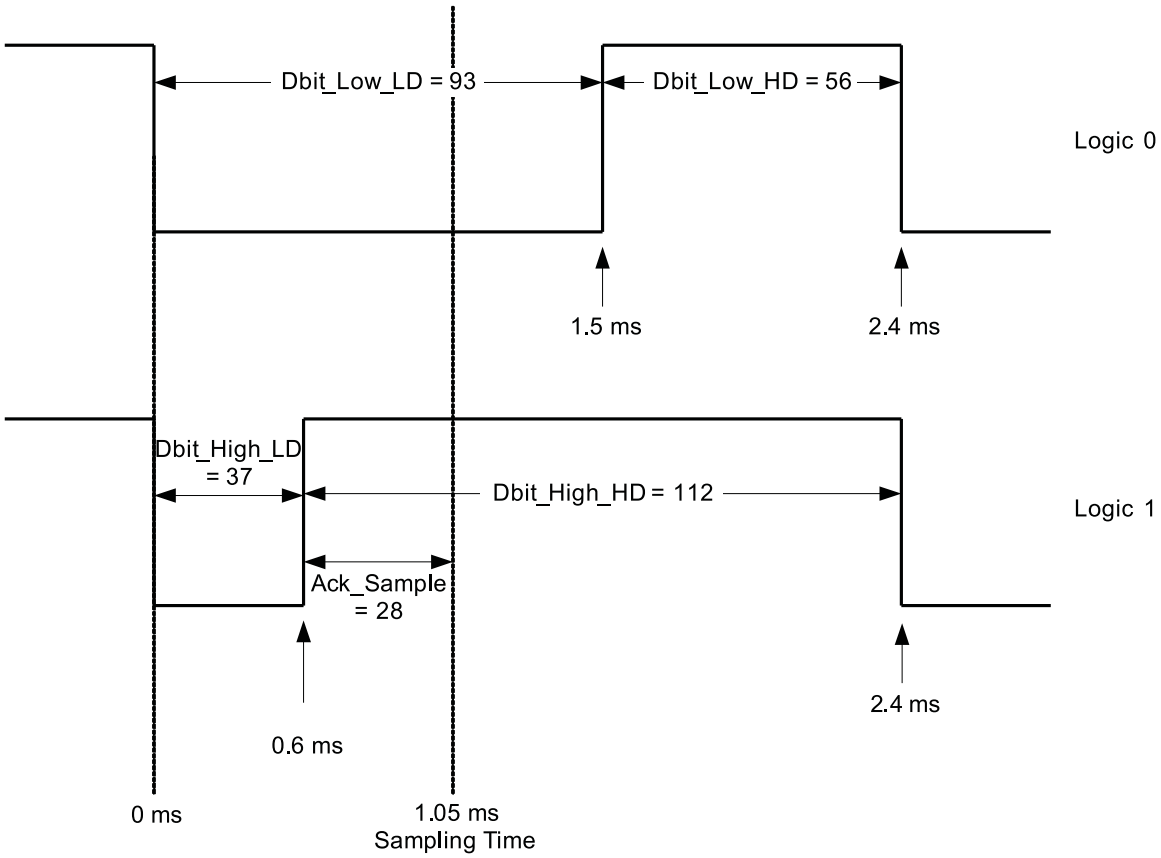

Figure 11. TACCRx Settings for Transmitting a CEC Data Bit

7 CEC-to-I²C Translation

When another device transmits a CEC message to the television, the MSP430 receives the message via I/O emulation, and the address is decoded. If the message was is intended for the television, meaning the logical destination address was either 0x0 for the television or 0xF for a broadcast address, then the MSP430 sends the message to the host processor via I²C. If the destination address is not 0x0 or 0xF, then the message is rejected and the host is not alerted. This section describes the events involved in this process.

7.1 CEC Receive Mode

A falling edge on P1.0 can be the first edge of a data message that another device is sending to the television. This edge triggers a PORT1 interrupt and starts the CEC to I²C translation. The MSP430 needs to time the low duration and high duration to confirm reception of a valid start bit. If the start bit violates a timing requirement, indicating that the trigger was noise, then the MSP430 returns to LPM4 and waits for bus activity on the I²C or for the occurrence of another CEC start bit. The use of timing requirements allows the MSP430 to reject false triggers that occur from noise on the CEC line. The flow chart in Figure 12 shows the bit timing checks throughout the process of receiving a CEC start bit.

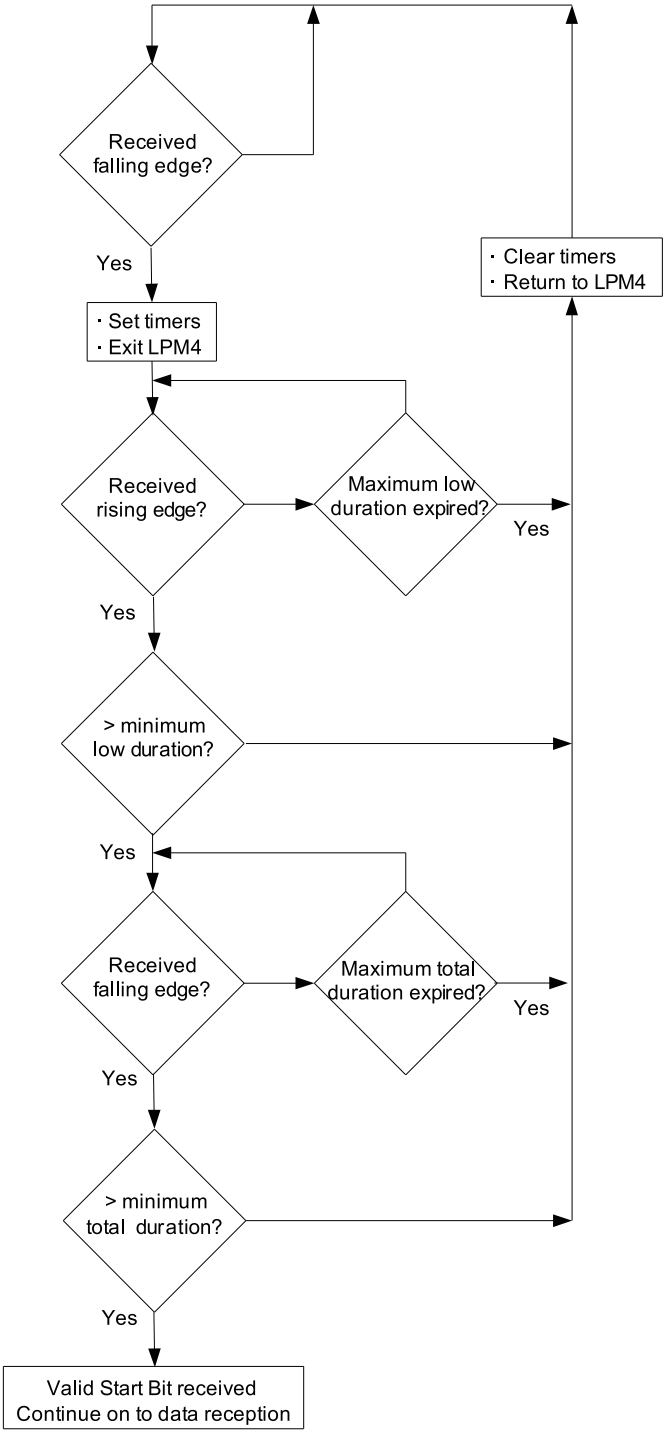

Figure 12. CEC Receive Start Bit Flow Chart

Upon receipt of a valid start bit, the MSP430 verifies that the destination address is valid. Valid destination addresses are 0x0 and 0xF, indicating a message specifically for the television or a broadcast message, respectively. If the address is not valid or if the timing of the start bit or any of the following bits violates the timing requirements, then the MSP430 will return to LPM4 and await start bits from either the I^2C or CEC busses. Returning to LPM4, the MSP430 does not acknowledging CEC data reception without the start of a new message. Failure to acknowledge the receipt of a byte during a message alerts the transmitter of an error.

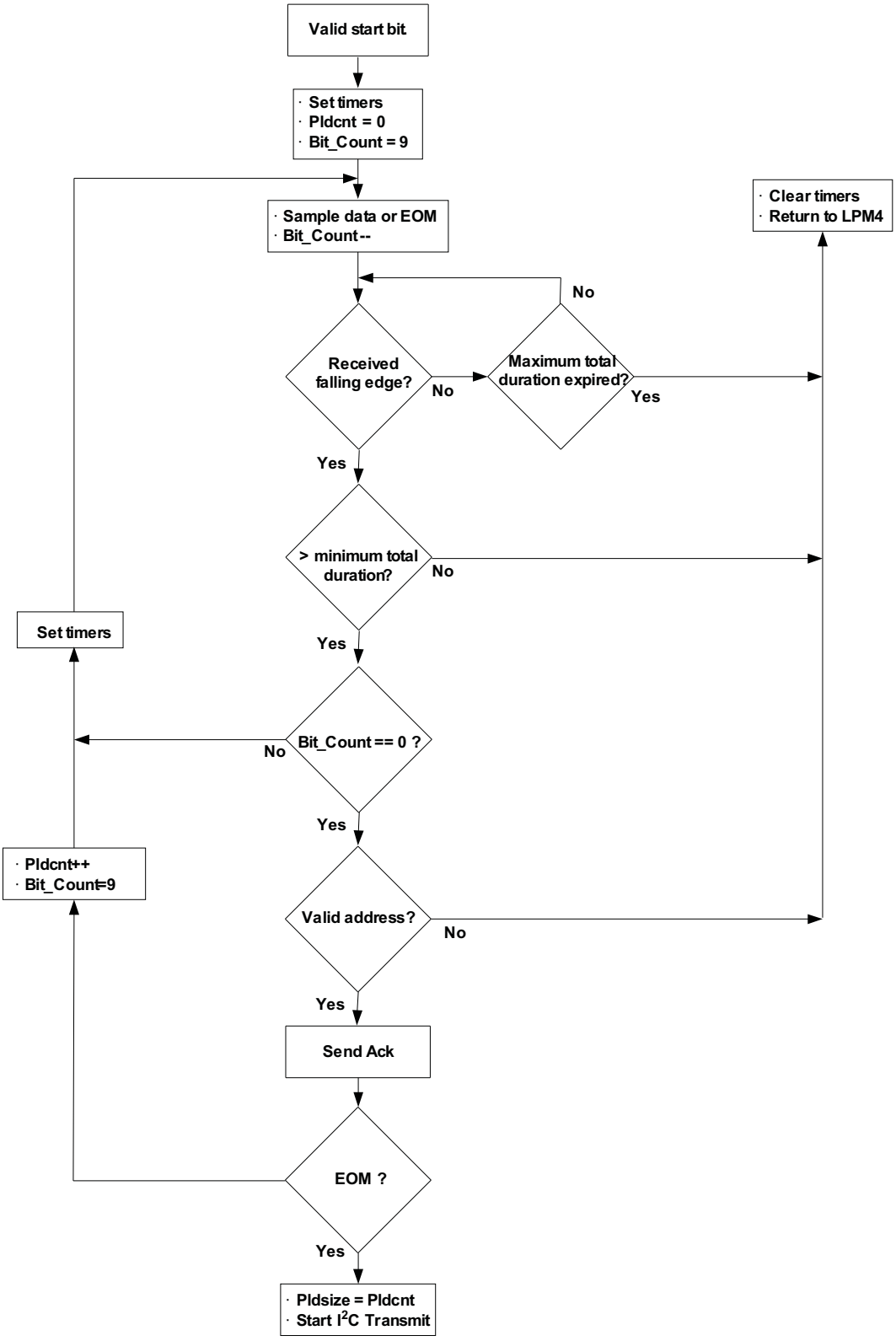

Figure 13. CEC Data Receive Flow Chart

7.2 I²C Transmit Mode

I²C transmission takes place after the reception of a CEC message. The USI I²C module enters the master transmit mode and operates at 100 kbps. The message data is loaded into the I²C transmit buffer from the variable Payload[i]. As with the CEC transmit code, when Pldcnt equals Pldsize, the last byte has been transmitted. At this time, the MSP430 generates a stop bit. Once the I²C data transmission is complete, the MSP430 enters LPM4 and waits for either CEC or I²C bus activity.

Figure 14 is the same message traffic shown in Figure 9, but the lower window focuses on the end of the message. The lower window is an expansion of the I²C message, which includes the end of the CEC message received by the MSP430F2013 and the I²C message transmitted by the MSP430F2013 and received by the host MSP430FG4619.

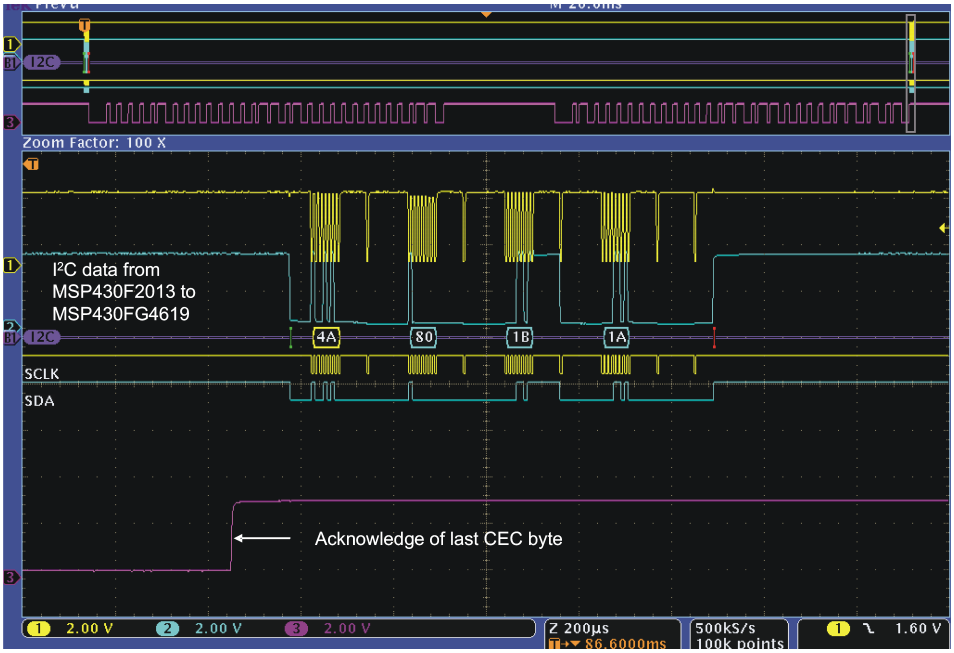

Figure 14. I²C Transmit Waveforms

8 Use of LPM4

As shown in Figure 15, the MSP430 waits in low power mode 4 (LPM4) until an interrupt is received from the digital IO (indicating the reception of a CEC transaction) or the USI module (indicating the reception of an I^2C data transmission).

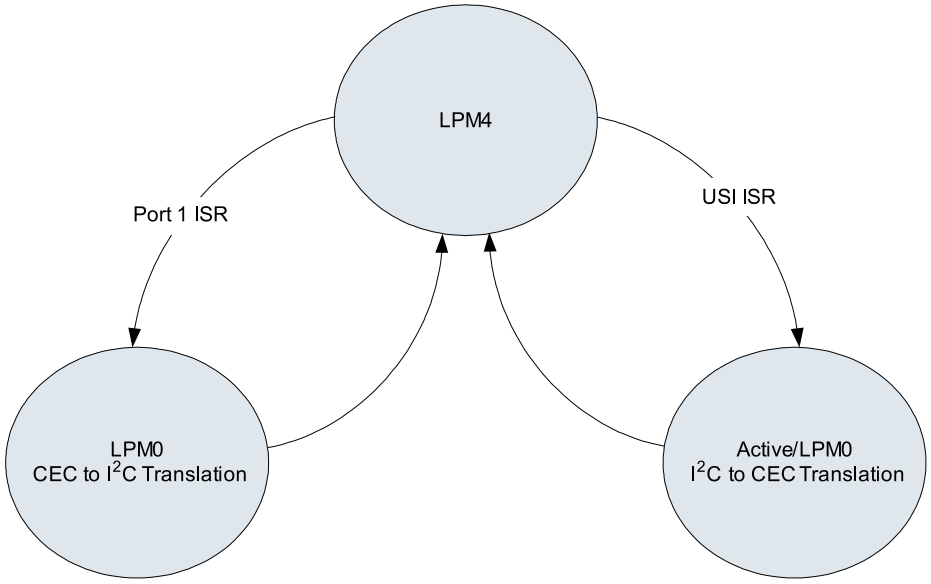

Figure 15. CEC-to-I^2C Bridge State Diagram

In LPM4, there is no clock running within the MSP430. This is not a problem in this application because activity is always initiated by receiving data from another device, whether it be from the host via I^2C or from another node on the CEC bus.

The USI module is able to handle the address check on its own, so any I^2C packets directed at other nodes on the bus do not wake the MSP430 CPU. The USI also allows data to be sent or received without CPU involvement, further enabling use of LPM4.

In the case of CEC, the same use of LPM4 cannot be utilized, because the protocol is being emulated. As a result, the CPU stays in active mode for the duration of the CEC transmission/reception. However, the existence of port interrupts allows the MSP430 to stay in LPM4 up until the first bit of a received CEC packet.

9 Summary

The USI module and the IO port pins on the MSP430 make it well suited for bridging data between I^2C and CEC data buses. Additional features of this application include data retransmission upon receipt of a NACK on both the I^2C or CEC busses and pulse width validation of CEC start and data bits. Both of these features improve the overall reliability of the data link.

Since most of the upper level protocol resides outside the MSP430, other applications can use this as a basis for an I^2C to single-wire interface.

10 References

1. *High-Definition Multimedia Interface Specification Version 1.3* (http://www.hdmi.org)

2. *I^2C-Bus Specification and User Manual* (http://www.nxp.com)

3. *MSP430x2xx Family User's Guide* (SLAU144)

4. *Using the USI I^2C Code Library* (SLAA368)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265