Differences between ADuCM360/361 and ADuCM362/363

Introduction

This document presents the differences between ADuCM360 and ADuCM362.

Summary

The ADuCM362 and ADuCM363 are based on the ADuCM360 and ADuCM361. The flash memory was doubled from 128kB to 256kB and the SRAM was tripled to 24kB. There are also modifications to the clock tree and some of the digital peripherals were updated to the latest version available within ADI.

Hardware Differences

Significant Changes

1. The CLKSYSDIV[0] register used to change the main system clock (UCLK) from 16MHz to 8MHz. This has now been expanded to CLKSYSDIV[1:0]

Table 1. CLKSYSDIV Register Bit Descriptions

Bits	Name	Description
12:2	Reserved	
1:0	UCLKCD	Clock divide bits for UCLK system clock 00: UCLK = 16 MHz 01: UCLK = 8 MHz 10: UCLK = 4 MHz 11: UCLK = 2 MHz

2. Added new register CLCKCON2 at 0x4000203C to control the clock of UART1

Table 2. CLKCON1 Register Bit Description

Bits	Name	Description
15:11	Reserved	
10:8	UART1CD	Clock divide bits for UART1 system clock 000: UCLK/1 = 16 MHz 001: UCLK/2 = 8 MHz 010: UCLK/4 = 4 MHz 011: UCLK/8 = 2 MHz 100: UCLK/16 = 1 MHz 101: UCLK/32 = 500 kHz 110: UCLK/64 = 250 kHz 111: UCLK/128 = 125 kHz
7:1	Reserved	
0	DISUART1CLK	0: Enable UART1 system clock. 1: Disable UART1 clock.

3. On the ADuCM360 the N_MINUS_1 field in the DMA structure was updated on every DMA transaction. This is no longer the case in the ADuCM362. This means that the N_MINUS_1 field

can no longer be used to poll the status of the DMA transaction. Two new registers have been provided that replace this functionality.

Table 3. Bit Descriptions for DMAGETNMINUS1 (Address: 0x40010910)

Bits	Name	Description
31:16	RESERVED	Reserved. Reserved, reads back 0
15:0	CHREQNMINUS1	Set the appropriate bit to request an update of the N_MINUS_1 field of the corresponding DMA channel. This can then be read in the DMANMINUS1 register Bit 0 corresponds to DMA channel 0, bit M-1 corresponds to DMA channel M-1 When Written: Bit [C] = 0, No effect. Bit [C] = 1, Request an update of the DMANMINUS1 register for the corresponding channel

Table 4. Bit Descriptions for DMANMINUS1 (Address: 0x40010914)

Bits	Name	Description
31:12	RESERVED	
11	VALIDCHAN	0: DMA structure updated in SRAM. Read N_MINUS_1 from corresponding DMA structure 1: DMA structure is stored in internal buffers. N_MINUS_1 should be read from DMANMINUS1
10	VALIDCNT	0: N_MINUS_1 value not yet valid. Repeat reading the register until valid. 1: N_MINUS_1 value is valid.
9:0	NMINUS1	Current N_MINUS_1 value for the channel requested

4. There are two UART peripherals available on the ADuCM362/363. UART0 is connected to P0.1 – P0.2 and UART1 is connected to P0.6 – P0.7. UART0 is the same UART as on the ADuCM360, and the MMRs are at the same address. UART1 also has the same registers as UART0 and they are located at 0x40005400
5. The order of the interrupts has been modified to accommodate extra DMA interrupts. This can be addressed with the use of a new header file and startup file or changing the CMSIS device
6. On the ADuCM360 the alternate structures had to be located at an address 0x100 bytes away from the primary structures. Due to the extra DMA channels on the ADuCM362 the alternate structures have to be located 0x200 bytes away. This can be implemented by modifying CCD_SIZE in the DMA library from 16 to 32.
7. A new flash controller had to be used in order to support the 256kB of flash memory. The flash controller commands have been made compatible with the ADuCM360, however the page size has been increased from 512 bytes to 2kB and the verify function uses a new 32 bit CRC instead of a 24 bit one.

Fixes or Enhancements

1. SPI0 has been updated to include DMA functionality.

2. Both SPI blocks have been updated to the latest version. This is backwards compatible with the previous version and new functionality was added. In continuous mode, it is now possible to enable CS interrupts, which will trigger an interrupt when the CS line is goes from HIGH to LOW and when it goes from LOW to HIGH. This is useful in multi slave systems.

Table 5. Bit Descriptions for SPIxSTA

Bits	Name	Description
14	CSRSG	Detected a rising edge on CS, in CONT mode. This bit will cause an interrupt. This can be used to identify the end of an SPI data frame. 0: Cleared to 0 when the Status register is read. 1: Set to 1 when there was a rising edge in CS line, when the device was in master mode, continuous transfer, High Frequency mode and CSIRQ_EN was asserted
13	CSFLG	Detected a falling edge on CS, in CONT mode. This bit will cause an interrupt. This can be used to identify the start of an SPI data frame. 0: Cleared to 0 when the Status register is read. 1: Set to 1 when there was a falling edge in CS line, when the device was in master mode, continuous transfer, High Frequency mode and CSIRQ_EN was asserted

Table 6. Bit Descriptions for SPIxDIV

Bits	Name	Description
8	CSIRQ_EN	Enable interrupt on every CS edge in CONT mode. If this bit is set and the SPI module is in continuous mode, any edge on CS will generate an interrupt and the corresponding status bits (CSRSG, CSFLG) will be asserted. If this bit is clear, then no interrupt will be generated. This bit has no effect if the SPI is not in continuous mode.
6	HFM	High Frequency Mode. This bit is used for applications using high frequency where the pad introduces a significant delay on the SCL. This can cause a significant enough difference between the serial clock and the data being received on the Rx shift register. In this mode, the Rx shift register is clocked by SCLIN instead of UCLK.

3. The I2C has been updated to the latest version. This is backwards compatible and the new functionality added is the automatic clock-stretching feature.

Table 7. Bit Descriptions for I2CSSCL

Bits	Name	Description
15:10	RESERVED	
9	SSRTSTA	Stretch timeout status bit for slave. Set when slave automatic stretch mode has timed out. Cleared when this bit is read
8	MSRTSTA	Stretch timeout status bit for master. Set when master automatic stretch mode has timed out. Cleared when this bit is read.
7:4	SSTRCON	Automatic stretch mode control for slave. These bits control automatic stretch mode for slave operation. Allows slave to hold the SCL line LOW and gain more time to service an interrupt, load a FIFO or read a FIFO. The timeout feature should be used to avoid a bus lockup condition where the slave indefinitely holds SCL low.

		As a slave transmitter, SCL is automatically stretched from the negative edge of SCL if slave TX FIFO is empty before sending ACK/NACK for address byte, or before sending data for a data byte. Stretching will stop when slave TX FIFO is no longer empty or a timeout occurs. As a slave receiver, SCL clock is automatically stretched from the negative edge of SCL, when slave RX FIFO is full, before sending ACK/NAK. Stretching will stop when slave RX FIFO is no longer in an overflow condition or a timeout occurs. 0000: Automatic slave clock stretching disabled 0001 to 1110: Automatic slave clock stretching enabled. Time-out period defined by following equation: $\frac{I2C0DIV[15:8] + I2C0DIV[7:4] - 1}{UCLK/CLKCON[10:8]} \left(2^{I2C0ASSCL[7:4]}\right)$ Note: I²C bus baud rate has no influence on the slave stretch timeout period 1111: Automatic slave clock stretching enabled with indefinite time-out period
3:0	MSTRCON	Automatic stretch mode control for master. These bits control automatic stretch mode for master operation. Allows master to hold the SCL line LOW and gain more time to service an interrupt, load a FIFO or read a FIFO. The timeout feature should be used to avoid a bus lockup condition where the slave indefinitely holds SCL low. As a master transmitter, SCL is automatically stretched from the negative edge of SCL if master TX FIFO is empty before sending ACK/NACK for address byte, or before sending data for a data byte. Stretching will stop when master TX FIFO is no longer empty or, a timeout occurs. As a master receiver, SCL clock is automatically stretched from the negative edge of SCL, when master RX FIFO is full, before sending ACK/NAK. Stretching will stop when master RX FIFO is no longer in an overflow condition or, a timeout occurs. 0000: Automatic master clock stretching disabled 0001 to 1110: Automatic master clock stretching enabled. Time-out period defined by following equation: $\frac{I2C0DIV[15:8] + I2C0DIV[7:4] - 1}{UCLK/CLKCON[10:8]} \left(2^{I2C0ASSCL[3:0]}\right)$

4. EXTREF2IN can be used with both ADCs on the ADuCM362
5. ADCxCFG[1:0] has been expanded to ADCxCFG[2:0] to allow extra flexibility on the external reference buffers

Table 8. Bit Descriptions for ADCxCFG

Bits	Name	Description
2:0	EXTBUF	Enable external buffers. 000: Both external reference buffers are powered down and bypassed. 001: External buffers are enabled. The VREF+ and VREF− inputs are selected as buffer inputs. 010: External buffers are enabled. The VREF+ and EXTREF2IN+ inputs are selected as buffer inputs. 011: External buffer on VREF+ is enabled. External buffer on VREF− is powered down and bypassed. 100: External buffer on EXTREF2IN + is enabled. Other reference buffers are powered down and bypassed.