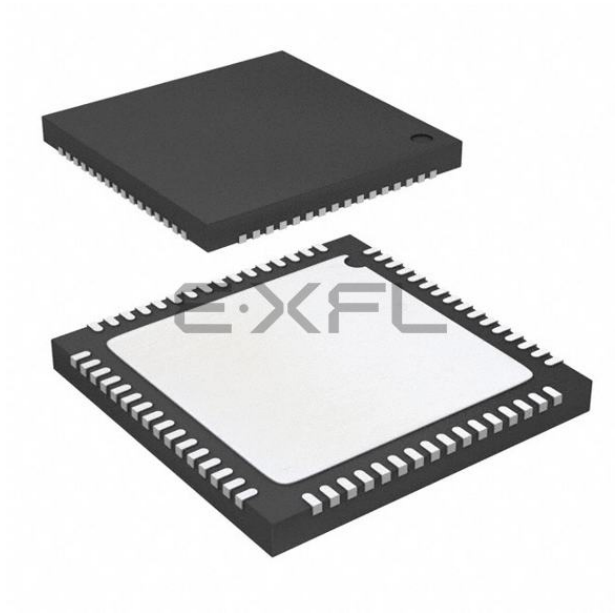

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM7®
Core Size	16/32-Bit
Speed	41.78MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	PLA, POR, PWM, PSM, Temp Sensor, WDT
Number of I/O	28
Program Memory Size	126KB (63K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 32
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 10x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad, CSP
Supplier Device Package	64-LFCSP-VQ (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7128bcpz126-rl

TABLE OF CONTENTS

Features ... 1
Functional Block Diagram ... 1
Revision History ... 2
General Description ... 3
Specifications ... 4
Timing Specifications ... 8
Absolute Maximum Ratings ... 15
ESD Caution ... 15
Pin Configuration and Function Descriptions ... 16
Typical Performance Characteristics ... 21
Terminology ... 24
ADC Specifications ... 24
DAC Specifications ... 24
Overview of the ARM7TDMI Core ... 25
Thumb Mode (T) ... 25
Long Multiply (M) ... 25
EmbeddedICE (I) ... 25
Exceptions ... 25
ARM Registers ... 25
Interrupt Latency ... 26
Memory Organization ... 27
Flash/EE Memory ... 27
SRAM ... 27
Memory Mapped Registers ... 27
Complete MMR Listing ... 28
ADC Circuit Overview ... 32
ADC Transfer Function ... 32
Typical Operation ... 33
Converter Operation ... 36
Driving the Analog Inputs ... 37
Temperature Sensor ... 37
Band Gap Reference ... 38
Nonvolatile Flash/EE Memory ... 39
Flash/EE Memory Overview ... 39
Flash/EE Memory ... 39
Flash/EE Memory Security ... 40
Flash/EE Control Interface ... 40
Execution Time from SRAM and FLASH/EE ... 43
Reset and Remap ... 44
Other Analog Peripherals ... 45
DAC ... 45
DDS ... 46
Power Supply Monitor ... 47
Comparator ... 47
Oscillator and PLL—Power Control ... 49
Digital Peripherals ... 51
PWM General Overview ... 51
PWM Convert Start Control ... 52
General-Purpose I/O ... 55
Serial Port Mux ... 57
UART Serial Interface ... 57
Serial Peripheral Interface ... 63
I²C-Compatible Interfaces ... 65
I²C Registers ... 65
Programmable Logic Array (PLA) ... 69
Processor Reference Peripherals ... 72
Interrupt System ... 72
Timers ... 73
Timer0—Lifetime Timer ... 73
Timer1—General-Purpose Timer ... 75
Timer2—Wake-Up Timer ... 77
Timer3—Watchdog Timer ... 79
Timer4—General-Purpose Timer ... 81
External Memory Interfacing ... 83
Timing Diagrams ... 84
Hardware Design Considerations ... 87
Power Supplies ... 87
Grounding and Board Layout Recommendations ... 87
Clock Oscillator ... 88
Power-On Reset Operation ... 89
Development Tools ... 90
In-Circuit Serial Downloader ... 90
Outline Dimensions ... 91
Ordering Guide ... 92

REVISION HISTORY

4/07—Revision 0: Initial Version

Table 8. SPI Slave Mode Timing (PHASE Mode = 0)

Parameter	Description	Min	Typ	Max	Unit
t_{CS}	CS to SCLOCK edge[1]	$2 \times t_{UCLK}$			ns
t_{SL}	SCLOCK low pulse width[2]		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{SH}	SCLOCK high pulse width[2]		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{DAV}	Data output valid after SCLOCK edge			$2 \times t_{HCLK} + 2 \times t_{UCLK}$	ns
t_{DSU}	Data input setup time before SCLOCK edge[1]	$1 \times t_{UCLK}$			ns
t_{DHD}	Data input hold time after SCLOCK edge[1]	$2 \times t_{UCLK}$			ns
t_{DF}	Data output fall time		5	12.5	ns
t_{DR}	Data output rise time		5	12.5	ns
t_{SR}	SCLOCK rise time		5	12.5	ns
t_{SF}	SCLOCK fall time		5	12.5	ns
t_{DOCS}	Data output valid after CS edge			25	ns
t_{SFS}	CS high after SCLOCK edge	0			ns

[1] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.
[2] t_{HCLK} depends on the clock divider or CD bits in the PLLCON MMR, $t_{HCLK} = t_{UCLK}/2^{CD}$.

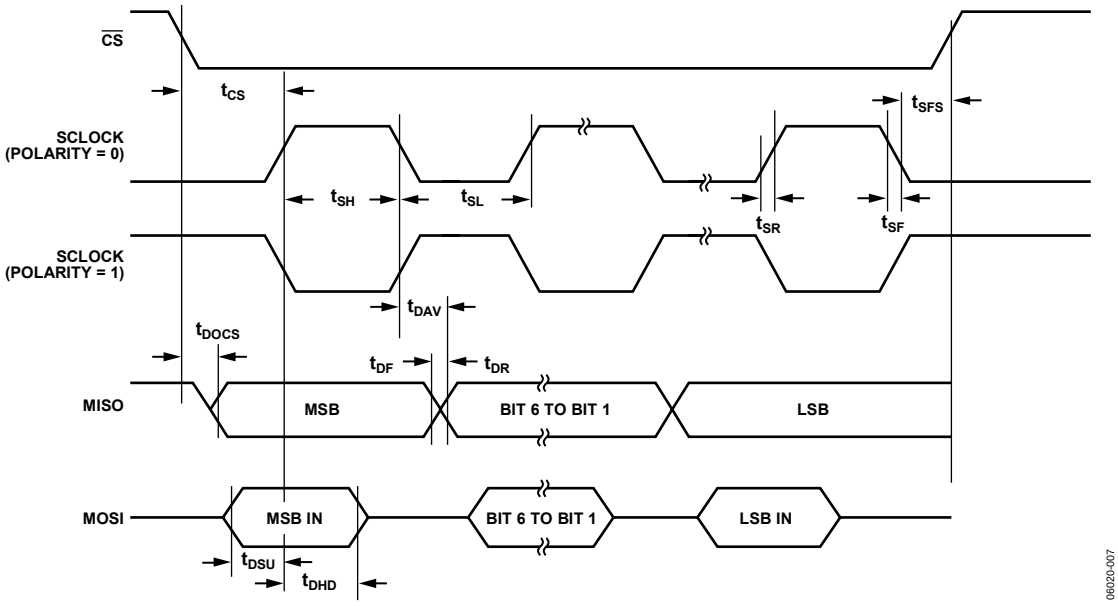

Figure 9. SPI Slave Mode Timing (PHASE Mode = 0)

TERMINOLOGY

ADC SPECIFICATIONS

Integral Nonlinearity
The maximum deviation of any code from a straight line passing through the endpoints of the ADC transfer function. The endpoints of the transfer function are zero scale, a point ½ LSB below the first code transition and full scale, a point ½ LSB above the last code transition.

Differential Nonlinearity
The difference between the measured and the ideal 1 LSB change between any two adjacent codes in the ADC.

Offset Error
The deviation of the first code transition (0000 . . . 000) to (0000 . . . 001) from the ideal, that is, +½ LSB.

Gain Error
The deviation of the last code transition from the ideal AIN voltage (full scale − 1.5 LSB) after the offset error has been adjusted out.

Signal to (Noise + Distortion) Ratio
The measured ratio of signal to (noise + distortion) at the output of the ADC. The signal is the rms amplitude of the fundamental. Noise is the rms sum of all nonfundamental signals up to half the sampling frequency ($f_S/2$), excluding dc. The ratio is dependent on the number of quantization levels in the digitization process; the more levels, the smaller the quantization noise.

The theoretical signal to (noise + distortion) ratio for an ideal N-bit converter with a sine wave input is given by

Signal to (*Noise* + *Distortion*) = (6.02 N + 1.76) dB

Thus, for a 12-bit converter, this is 74 dB.

Total Harmonic Distortion
The ratio of the rms sum of the harmonics to the fundamental.

DAC SPECIFICATIONS

Relative Accuracy
Otherwise known as endpoint linearity, relative accuracy is a measure of the maximum deviation from a straight line passing through the endpoints of the DAC transfer function. It is measured after adjusting for zero error and full-scale error.

Voltage Output Settling Time
The amount of time it takes for the output to settle to within a 1 LSB level for a full-scale input change.

OVERVIEW OF THE ARM7TDMI CORE

The ARM7 core is a 32-bit reduced instruction set computer (RISC). It uses a single 32-bit bus for instruction and data. The length of the data can be 8 bits, 16 bits, or 32 bits. The length of the instruction word is 32 bits.

The ARM7TDMI is an ARM7 core with the following four additional features:

- T, support for the Thumb® (16-bit) instruction set
- D, support for debug
- M, support for long multiplications
- I, includes the embedded ICE module to support embedded system debugging

THUMB MODE (T)

An ARM® instruction is 32-bits long. The ARM7TDMI processor supports a second instruction set that has been compressed into 16-bits, called the Thumb instruction set. Faster execution from 16-bit memory and greater code density can usually be achieved by using the Thumb instruction set instead of the ARM instruction set, which makes the ARM7TDMI core particularly suitable for embedded applications.

However, the Thumb mode has two limitations:

- Thumb code typically requires more instructions for the same job. As a result, ARM code is usually best for maximizing the performance of the time-critical code.
- The Thumb instruction set does not include some of the instructions needed for exception handling, which automatically switches the core to ARM code for exception handling.

See the ARM7TDMI user guide for details on the core architecture, the programming model, and both the ARM and Thumb instruction sets.

LONG MULTIPLY (M)

The ARM7TDMI instruction set includes four extra instructions that perform 32-bit by 32-bit multiplication with 64-bit result, and 32-bit by 32-bit multiplication-accumulation (MAC) with 64-bit result. This result is achieved in fewer cycles than required on a standard ARM7 core.

EMBEDDEDICE (I)

EmbeddedICE provides integrated on-chip support for the core. The EmbeddedICE module contains the breakpoint and watchpoint registers that allow code to be halted for debugging purposes. These registers are controlled through the JTAG test port.

When a breakpoint or watchpoint is encountered, the processor halts and enters debug state. Once in a debug state, the processor registers can be inspected, as well as the Flash/EE, the SRAM, and the memory mapped registers.

EXCEPTIONS

ARM supports five types of exceptions and a privileged processing mode for each type. The five types of exceptions are

- Normal interrupt or IRQ. This is provided to service general-purpose interrupt handling of internal and external events.
- Fast interrupt or FIQ. This is provided to service data transfer or communication channel with low latency. FIQ has priority over IRQ.
- Memory abort.
- Attempted execution of an undefined instruction.
- Software interrupt instruction (SWI). This can be used to make a call to an operating system.

Typically, the programmer defines interrupt as IRQ, but for higher priority interrupt, that is, faster response time, the programmer can define interrupt as FIQ.

ARM REGISTERS

ARM7TDMI has a total of 37 registers: 31 general-purpose registers and six status registers. Each operating mode has dedicated banked registers.

When writing user-level programs, 15 general-purpose, 32-bit registers (R0 to R14), the program counter (R15), and the current program status register (CPSR) are usable. The remaining registers are used only for system-level programming and exception handling.

When an exception occurs, some of the standard registers are replaced with registers specific to the exception mode. All exception modes have replacement banked registers for the stack pointer (R13) and the link register (R14), as represented in Figure 28. The fast interrupt mode has more registers (R8 to R12) for fast interrupt processing. Interrupt processing can begin without the need to save or restore these registers and, thus, saves critical time in the interrupt handling process.

More information relative to the programmer's model and the ARM7TDMI core architecture can be found in the following ARM7TDMI technical and ARM architecture manuals available directly from ARM Ltd.:

- DDI0029G, *ARM7TDMI Technical Reference Manual*
- DDI-0100, *ARM Architecture Reference Manual*

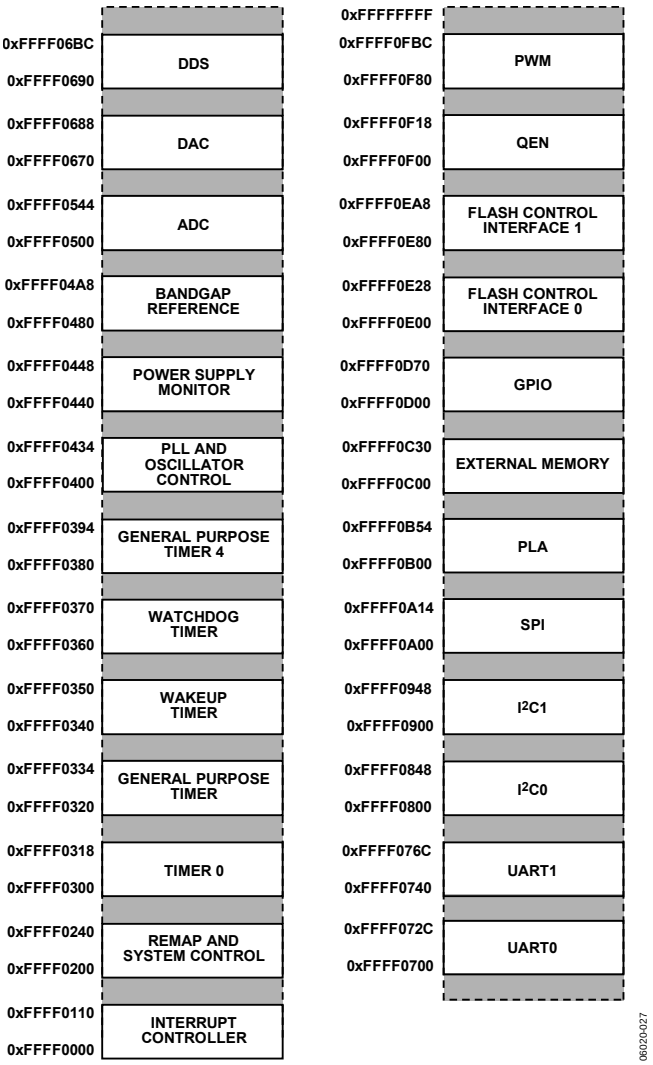

Figure 31. Memory Mapped Registers

COMPLETE MMR LISTING

Note that the Access Type column corresponds to the access time reading or writing an MMR. It depends on the AMBA bus used to access the peripheral. The processor has two AMBA buses: the AHB (advanced high performance bus) used for system modules and the APB (advanced peripheral bus) used for lower performance peripherals.

Table 12. IRQ Base Address = 0xFFFF0000

Address	Name	Byte	Access Type	Cycle
0x0000	IRQSTA	4	R	1
0x0004	IRQSIG	4	R	1
0x0008	IRQEN	4	R/W	1
0x000C	IRQCLR	4	W	1
0x0010	SWICFG	4	W	1
0x0100	FIQSTA	4	R	1
0x0104	FIQSIG	4	R	1
0x0108	FIQEN	4	R/W	1
0x010C	FIQCLR	4	W	1

Table 13. System Control Base Address = 0xFFFF0200

Address	Name	Byte	Access Type	Cycle
0x0220	REMAP	1	R/W	1
0x0230	RSTSTA	1	R	1
0x0234	RSTCLR	1	W	1

Table 14. Timer Base Address = 0xFFFF0300

Address	Name	Byte	Access Type	Cycle
0x0300	T0LD	2	R/W	2
0x0304	T0VAL0	2	R	2
0x0308	T0VAL1	4	R	2
0x030C	T0CON	4	R/W	2
0x0310	T0ICLR	1	W	2
0x0314	T0CAP	2	R	2
0x0320	T1LD	4	R/W	2
0x0324	T1VAL	4	R	2
0x0328	T1CON	4	R/W	2
0x032C	T1ICLR	1	W	2
0x0330	T1CAP	4	R	2
0x0340	T2LD	4	R/W	2
0x0344	T2VAL	4	R	2
0x0348	T2CON	4	R/W	2
0x034C	T2ICLR	1	W	2
0x0360	T3LD	2	R/W	2
0x0364	T3VAL	2	R	2
0x0368	T3CON	2	R/W	2
0x036C	T3ICLR	1	W	2
0x0380	T4LD	4	R/W	2
0x0384	T4VAL	4	R	2
0x0388	T4CON	4	R/W	2
0x038C	T4ICLR	1	W	2
0x0390	T4CAP	4	R	2

Table 15. PLL Base Address = 0xFFFF0400

Address	Name	Byte	Access Type	Cycle
0x0404	POWKEY1	2	W	2
0x0408	POWCON	2	R/W	2
0x040C	POWKEY2	2	W	2
0x0410	PLLKEY1	2	W	2
0x0414	PLLCON	2	R/W	2
0x0418	PLLKEY2	2	W	2

Table 16. PSM Base Address = 0xFFFF0440

Address	Name	Byte	Access Type	Cycle
0x0440	PSMCON	2	R/W	2
0x0444	CMPCON	2	R/W	2

Table 17. Reference Base Address = 0xFFFF0480

Address	Name	Byte	Access Type	Cycle
0x048C	REFCON	1	R/W	2

Table 25. PLA Base Address = 0xFFFF0B00

Address	Name	Byte	Access Type	Cycle
0x0B00	PLAELM0	2	R/W	2
0x0B04	PLAELM1	2	R/W	2
0x0B08	PLAELM2	2	R/W	2
0x0B0C	PLAELM3	2	R/W	2
0x0B10	PLAELM4	2	R/W	2
0x0B14	PLAELM5	2	R/W	2
0x0B18	PLAELM6	2	R/W	2
0x0B1C	PLAELM7	2	R/W	2
0x0B20	PLAELM8	2	R/W	2
0x0B24	PLAELM9	2	R/W	2
0x0B28	PLAELM10	2	R/W	2
0x0B2C	PLAELM11	2	R/W	2
0x0B30	PLAELM12	2	R/W	2
0x0B34	PLAELM13	2	R/W	2
0x0B38	PLAELM14	2	R/W	2
0x0B3C	PLAELM15	2	R/W	2
0x0B40	PLACLK	1	R/W	2
0x0B44	PLAIRQ	4	R/W	2
0x0B48	PLAADC	4	R/W	2
0x0B4C	PLADIN	4	R/W	2
0x0B50	PLAOUT	4	R	2

Table 26. External Memory Base Address = 0xFFFF0C00

Address	Name	Byte	Access Type	Cycle
0x0C00	XMCFG	1	R/W	2
0x0C10	XM0CON	1	R/W	2
0x0C14	XM1CON	1	R/W	2
0x0C18	XM2CON	1	R/W	2
0x0C1C	XM3CON	1	R/W	2
0x0C20	XM0PAR	2	R/W	2
0x0C24	XM1PAR	2	R/W	2
0x0C28	XM2PAR	2	R/W	2
0x0C2C	XM3PAR	2	R/W	2

Table 27. GPIO Base Address = 0xFFFF0D00

Address	Name	Byte	Access Type	Cycle
0x0D00	GP0CON	4	R/W	1
0x0D04	GP1CON	4	R/W	1
0x0D08	GP2CON	4	R/W	1
0x0D0C	GP3CON	4	R/W	1
0x0D10	GP4CON	4	R/W	1
0x0D20	GP0DAT	4	R/W	1
0x0D24	GP0SET	1	W	1
0x0D28	GP0CLR	1	W	1
0x0D2C	GP0PAR	4	R/W	1
0x0D30	GP1DAT	4	R/W	1
0x0D34	GP1SET	1	W	1
0x0D38	GP1CLR	1	W	1
0x0D3C	GP1PAR	4	R/W	1
0x0D40	GP2DAT	4	R/W	1
0x0D44	GP2SET	1	W	1
0x0D48	GP2CLR	1	W	1
0x0D50	GP3DAT	4	R/W	1
0x0D54	GP3SET	1	W	1
0x0D58	GP3CLR	1	W	1
0x0D5C	GP3PAR	4	R/W	1
0x0D60	GP4DAT	4	R/W	1
0x0D64	GP4SET	1	W	1
0x0D68	GP4CLR	1	W	1
0x0D6C	GP4PAR	1	W	1

Table 28. Flash/EE Block 0 Base Address = 0xFFFF0E00

Address	Name	Byte	Access Type	Cycle
0x0E00	FEE0STA	1	R	1
0x0E04	FEE0MOD	1	R/W	1
0x0E08	FEE0CON	1	R/W	1
0x0E0C	FEE0DAT	2	R/W	1
0x0E10	FEE0ADR	2	R/W	1
0x0E18	FEE0SGN	3	R	1
0x0E1C	FEE0PRO	4	R/W	1
0x0E20	FEE0HID	4	R/W	1

Table 29. Flash/EE Block 1 Base Address = 0xFFFF0E80

Address	Name	Byte	Access Type	Cycle
0x0E80	FEE1STA	1	R	1
0x0E84	FEE1MOD	1	R/W	1
0x0E88	FEE1CON	1	R/W	1
0x0E8C	FEE1DAT	2	R/W	1
0x0E90	FEE1ADR	2	R/W	1
0x0E98	FEE1SGN	3	R	1
0x0E9C	FEE1PRO	4	R/W	1
0x0EA0	FEE1HID	4	R/W	1

Table 44. FEE0PRO and FEE0HID MMR Bit Designations

Bit	Description
31	Read Protection. Cleared by user to protect Block 0. Set by user to allow reading Block 0.
30:0	Write Protection for Page 123 to Page 120, for Page 119 to Page 116, and for Page 3 to Page 0. Cleared by user to protect the pages in writing. Set by user to allow writing the pages.

Table 45. FEE1PRO and FEE1HID MMR Bit Designations

Bit	Description
31	Read Protection. Cleared by user to protect Block 1. Set by user to allow reading Block 1.
30	Write Protection for Page 127 to Page 120. Cleared by user to protect the pages in writing. Set by user to allow writing the pages.
31:0	Write Protection for Page 119 to Page 116 and for Page 3 to Page 0. Cleared by user to protect the pages in writing. Set by user to allow writing the pages.

EXECUTION TIME FROM SRAM AND FLASH/EE

This section describes SRAM and Flash/EE access times during execution for applications where execution time is critical.

Execution from SRAM

Fetching instructions from SRAM takes one clock cycle because the access time of the SRAM is 2 ns and a clock cycle is 22 ns minimum. However, if the instruction involves reading or writing data to memory, one extra cycle must be added if the data is in SRAM (or three cycles if the data is in Flash/EE), one cycle to execute the instruction and two cycles to get the 32-bit data from Flash/EE. A control flow instruction, such as a branch instruction, takes one cycle to fetch, but it also takes two cycles to fill the pipeline with the new instructions.

Execution from Flash/EE

Because the Flash/EE width is 16 bits and access time for 16-bit words is 23 ns, execution from Flash/EE cannot be done in one cycle (as can be done from SRAM when the CD bit = 0). In addition, some dead times are needed before accessing data for any value of CD bits.

In ARM mode, where instructions are 32 bits, two cycles are needed to fetch any instruction when CD = 0. In Thumb mode, where instructions are 16 bits, one cycle is needed to fetch any instruction.

Timing is identical in both modes when executing instructions that involve using the Flash/EE for data memory. If the instruction to be executed is a control flow instruction, an extra cycle is needed to decode the new address of the program counter and then four cycles are needed to fill the pipeline. A data processing instruction involving only core registers doesn't require any extra clock cycles, but if it involves data in Flash/EE, an extra clock cycle is needed to decode the address of the data and two cycles to get the 32-bit data from Flash/EE. An extra cycle must also be added before fetching another instruction. Data transfer instructions are more complex and are summarized in Table 46.

Table 46. Execution Cycles in ARM/Thumb Mode

Instructions	Fetch Cycles	Dead Time	Data Access	Dead Time
LD	2/1	1	2	1
LDH	2/1	1	1	1
LDM/PUSH	2/1	N	$2 \times N$	N
STR	2/1	1	2×20 µs	1
STRH	2/1	1	20 µs	1
STRM/POP	2/1	N	$2 \times N \times 20$ µs	N

With $1 < N \leq 16$, N is the number of bytes of data to load or store in the multiple load/store instruction. The SWAP instruction combines an LD and STR instruction with only one fetch, giving a total of eight cycles plus 40 µs.

RESET AND REMAP

The ARM exception vectors are all situated at the bottom of the memory array, from Address 0x00000000 to Address 0x00000020, as shown in Figure 45.

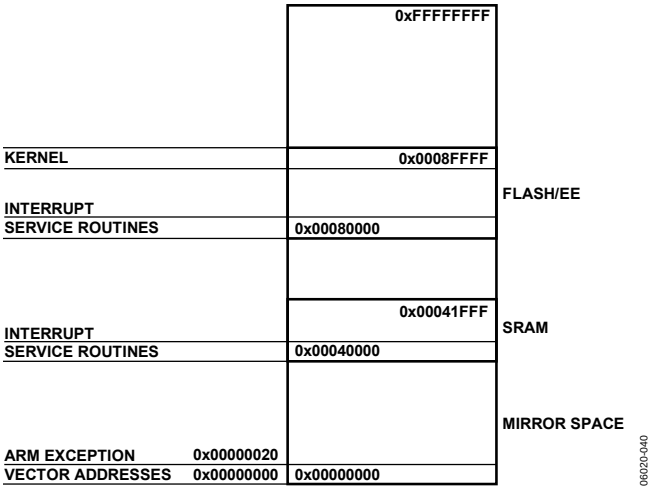

Figure 45. Remap for Exception Execution

By default and after any reset, the Flash/EE is mirrored at the bottom of the memory array. The remap function allows the programmer to mirror the SRAM at the bottom of the memory array, facilitating execution of exception routines from SRAM instead of from Flash/EE. This means exceptions are executed twice as fast, with the exception being executed in ARM mode (32 bits), and the SRAM being 32 bits wide instead of 16-bit wide Flash/EE memory.

Remap Operation

When a reset occurs on the ADuC7128/ADuC7129, execution starts automatically in factory-programmed internal configuration code. This kernel is hidden and cannot be accessed by user code. If the ADuC7128/ADuC7129 are in normal mode (the $\overline{BM}$ pin is high), they execute the power-on configuration routine of the kernel and then jump to the reset vector Address 0x00000000 to execute the user's reset exception routine. Because the Flash/EE is mirrored at the bottom of the memory array at reset, the reset interrupt routine must always be written in Flash/EE.

The remap is done from Flash/EE by setting Bit 0 of the REMAP register. Precautions must be taken to execute this command from Flash/EE, above Address 0x00080020, and not from the bottom of the array because this is replaced by the SRAM.

This operation is reversible: the Flash/EE can be remapped at Address 0x00000000 by clearing Bit 0 of the REMAP MMR. Precaution must again be taken to execute the remap function from outside the mirrored area. Any kind of reset remaps the Flash/EE memory at the bottom of the array.

Reset Operation

There are four kinds of reset: external reset, power-on reset, watchdog expiration, and software force. The RSTSTA register indicates the source of the last reset and RSTCLR clears the RSTSTA register. These registers can be used during a reset exception service routine to identify the source of the reset. If RSTSTA is null, the reset was external. Note that when clearing RSTSTA, all bits that are currently 1 must be cleared. Otherwise, a reset event occurs.

Table 47. REMAP MMR Bit Designations

Bit	Name	Description
0	Remap	Remap Bit. Set by user to remap the SRAM to Address 0x00000000. Cleared automatically after reset to remap the Flash/EE memory to Address 0x00000000.

Table 48. RSTSTA MMR Bit Designations

Bit	Description
7:3	Reserved.
2	Software Reset. Set by user to force a software reset. Cleared by setting the corresponding bit in RSTCLR.
1	Watchdog Timeout. Set automatically when a watchdog timeout occurs. Cleared by setting the corresponding bit in RSTCLR.
0	Power-On Reset. Set automatically when a power-on reset occurs. Cleared by setting the corresponding bit in RSTCLR.

OTHER ANALOG PERIPHERALS

DAC

The ADuC7128/ADuC7129 feature a 10-bit current DAC that can be used to generate user-defined waveforms or sine waves generated by the DDS. The DAC consists of a 10-bit IDAC followed by a current-to-voltage conversion.

The current output of the IDAC is passed through a resistor and capacitor network where it is both filtered and converted to a voltage. This voltage is then buffered by an op amp and passed to the line driver.

For the DAC to function, the internal 2.5 V voltage reference must be enabled and driven out onto an external capacitor, REFCON = 0x01.

Once the DAC is enabled, users see a 5 mV drop in the internal reference value. This is due to bias currents drawn from the reference used in the DAC circuitry. It is recommended that if using the DAC, it be left powered on to avoid seeing variations in ADC results.

Table 49. DACCON MMR Bit Designations

Bit	Value	Description
10:9		Reserved. These bits should be written to 0 by the user.
8		Reserved. This bit should be written to 0 by the user.
7		Reserved. This bit should be written to 0 by the user.
6		Reserved. This bit should be written to 0 by the user.
5		Output Enable. This bit operates in all modes. In Line Driver mode, this bit should be set. Set by user to enable the line driver output. Cleared by user to disable the line driver output. In this mode the line driver output is high impedance.
4		Single-Ended or Differential Output Control. Set by user to operate in differential mode, the output is the differential voltage between LD1TX and LD2TX. The voltage output range is $V_{REF}/2 \pm V_{REF}/2$. Cleared by user to reference the LD1TX output to AGND. The voltage output range is $AV_{DD}/2 \pm V_{REF}/2$.
3		Reserved. This bit should be set to 0 by the user.
2:1		Operation Mode Control. This bit selects the mode of operation of the DAC.
	00	Power-Down.
	01	Reserved.
	10	Reserved.
	11	DDS and DAC Mode. Selected by DACEN.
0		DAC Update Rate Control. This bit has no effect when in DDS mode. Set by user to update the DAC on the negative edge of Timer1. This allows the user to use any one of the core CLK, OSC CLK, baud CLK, or user CLK and divide these down by 1, 16, 256, or 32,768. A user can do waveform generation by writing to the DAC data register from RAM and updating the DAC at regular intervals via Timer1. Cleared by user to update the DAC on the negative edge of HCLK.

Comparator Interface

The comparator interface consists of a 16-bit MMR, CMPCON, described in Table 57.

Table 57. CMPCON MMR Bit Designations

Bit	Value	Name	Description
15:11			Reserved.
10		CMPEN	Comparator Enable Bit. Set by user to enable the comparator. Cleared by user to disable the comparator. Note: A comparator interrupt is generated on the enable of the comparator. This should be cleared in the user software.
9:8		CMPIN	Comparator Negative Input Select Bits.
	00		AVDD/2.
	01		ADC3 input.
	10		$V_{REF} \times 0.6$.
	11		Reserved.
7:6		CMPOC	Comparator Output Configuration Bits.
	00		IRQ and PLA connections disabled.
	01		IRQ and PLA connections disabled.
	10		PLA connections enabled.
	11		IRQ connections enabled.
5		CMPOL	Comparator Output Logic State Bit. When low, the comparator output is high when the positive input (CMP0) is above the negative input (CMP1). When high, the comparator output is high when the positive input is below the negative input.
4:3		CMPRES	Response Time.
	00		5 µs response time typical for large signals (2.5 V differential). 17 µs response time typical for small signals (0.65 mV differential).
	01		Reserved.
	10		Reserved.
	11		3 µs response time typical for any signal type.
2		CMPHYST	Comparator Hysteresis Bit. Set by user to have a hysteresis of about 7.5 mV. Cleared by user to have no hysteresis.
1		CMPORI	Comparator Output Rising Edge Interrupt. Set automatically when a rising edge occurs on the monitored voltage (CMP0). Cleared by user by writing a 1 to this bit.
0		CMPOFI	Comparator Output Falling Edge Interrupt. Set automatically when a falling edge occurs on the monitored voltage (CMP0). Cleared by user.

OSCILLATOR AND PLL—POWER CONTROL

The ADuC7128/ADuC7129 integrate a 32.768 kHz oscillator, a clock divider, and a PLL. The PLL locks onto a multiple (1275) of the internal oscillator to provide a stable 41.78 MHz clock for the system. The core can operate at this frequency, or at binary submultiples of it, to allow power saving. The default core clock is the PLL clock divided by 8 (CD = 3) or 5.2 MHz. The core clock frequency can be output on the ECLK pin as described in Figure 48. Note that when the ECLK pin is used to output the core clock, the output signal is not buffered and is not suitable for use as a clock source to an external device without an external buffer.

A power-down mode is available on the ADuC7128/ADuC7129.

The operating mode, clocking mode, and programmable clock divider are controlled via two MMRs, PLLCON (see Table 61) and POWCON (see Table 62). PLLCON controls operating mode of the clock system, and POWCON controls the core clock frequency and the power-down mode.

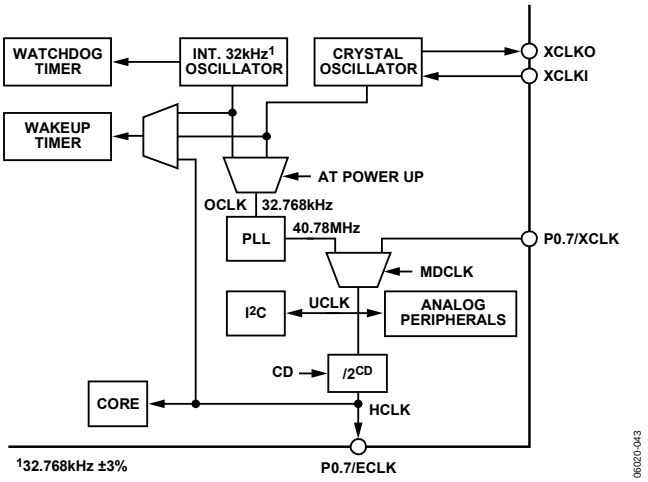

Figure 48. Clocking System

External Crystal Selection

To switch to an external crystal, use the following procedure:

1. Enable the Timer2 interrupt and configure it for a timeout period of >120 µs.
2. Follow the write sequence to the PLLCON register, setting the MDCLK bits to 01 and clearing the OSEL bit.
3. Force the part into nap mode by following the correct write sequence to the POWCON register.
4. When the part is interrupted from nap mode by the Timer2 interrupt source, the clock source has switched to the external clock.

Example Source Code

```
  T2LD = 5;
  TCON = 0x480;
  while ((T2VAL == t2val_old) || (T2VAL >
3)) //ensures timer value loaded
  IRQEN = 0x10;
//enable T2 interrupt
  PLLKEY1 = 0xAA;
  PLLCON = 0x01;
  PLLKEY2 = 0x55;

  POWKEY1 = 0x01;
  POWCON = 0x27;
// Set Core into Nap mode
  POWKEY2 = 0xF4;
```

In noisy environments, noise can couple to the external crystal pins, and PLL may lose lock momentarily. A PLL interrupt is provided in the interrupt controller. The core clock is immediately halted, and this interrupt is serviced only when the lock is restored.

In case of crystal loss, the watchdog timer should be used. During initialization, a test on the RSTSTA can determine if the reset came from the watchdog timer.

External Clock Selection

To switch to an external clock on P0.7, configure P0.7 in Mode 1. The external clock can be up to 44 MHz, providing the tolerance is 1%.

Example Source Code

```
    T2LD = 5;
    TCON = 0x480;
  while ((T2VAL == t2val_old) || (T2VAL >
3)) //ensures timer value loaded
  IRQEN = 0x10;
//enable T2 interrupt
  PLLKEY1 = 0xAA;
  PLLCON = 0x03; //Select external clock
  PLLKEY2 = 0x55;

  POWKEY1 = 0x01;
  POWCON = 0x27; // Set Core into Nap mode
  POWKEY2 = 0xF4;
```

Power Control System

A choice of operating modes is available on the ADuC7128/ADuC7129. Table 58 describes what part of the ADuC7128/ADuC7129 is powered on in the different modes and indicates the power-up time. Table 59 gives some typical values of the total current consumption (analog + digital supply currents) in the different modes, depending on the clock divider bits. The ADC is turned off.

Note that these values also include current consumption of the regulator and other parts on the test board on which these values were measured.

Table 70. GPIO Pin Function Designations

Port	Pin	Configuration			
		00	01	10	11
0	P0.0	GPIO	CMP	MS0	PLAI[7]
	P0.1[1]	GPIO		BLE	-
	P0.2[1]	GPIO		BHE	
	P0.3	GPIO	$\overline{TRST}$	A16	ADC_{BUSY}
	P0.4	GPIO/IRQ0	$\overline{CONVST}$	MS1	PLAO[1]
	P0.5	GPIO/IRQ1	ADC_{BUSY}	PLM_COMP	PLAO[2]
	P0.6	GPIO/T1	$\overline{MRST}$	AE	PLAO[3]
	P0.7	GPIO	ECLK/XCLK[2]	SIN0	PLAO[4]
1	P1.0	GPIO/T1	SIN0	SCL0	PLAI[0]
	P1.1	GPIO	SOUT0	SDA0	PLAI[1]
	P1.2	GPIO	RTS0	SCL1	PLAI[2]
	P1.3	GPIO	CTS0	SDA1	PLAI[3]
	P1.4	GPIO/IRQ2	RI0	CLK	PLAI[4]
	P1.5	GPIO/IRQ3	DCD0	MISO	PLAI[5]
	P1.6	GPIO	DSR0	MOSI	PLAI[6]
	P1.7	GPIO	DTR0	CSL	PLAO[0]
2	P2.0	GPIO	SYNC	SOUT	PLAO[5]
	P2.1[1]	GPIO		WS	PLAO[6]
	P2.2[1]	GPIO	RTS1	RS	PLAO[7]
	P2.3[1]	GPIO	CTS1	AE	
	P2.4[1]	GPIO	RI1	MS0	
	P2.5[1]	GPIO	DCD1	MS1	
	P2.6[1]	GPIO	DSR1	MS2	
	P2.7[1]	GPIO	DTR1	MS3	
3	P3.0	GPIO	PWM1	AD0	PLAI[8]
	P3.1	GPIO	PWM2	AD1	PLAI[9]
	P3.2	GPIO	PWM3	AD2	PLAI[10]
	P3.3	GPIO	PWM4	AD3	PLAI[11]
	P3.4	GPIO	PWM5	AD4	PLAI[12]
	P3.5	GPIO	PWM6	AD5	PLAI[13]
	P3.6[1]	GPIO	PWM1	AD6	PLAI[14]
	P3.7[1]	GPIO	PWM3	AD7	PLAI[15]
4	P4.0	GPIO	QENS1	AD8	PLAO[8]
	P4.1	GPIO	QENS2	AD9	PLAO[9]
	P4.2	GPIO	RSVD	AD10	PLAO[10]
	P4.3	GPIO	Trip (Shutdown)	AD11	PLAO[11]
	P4.4	GPIO	PLMIN	AD12	PLAO[12]
	P4.5	GPIO	PLMOUT	AD13	PLAO[13]
	P4.6	GPIO	SIN1	AD14	PLAO[14]
	P4.7	GPIO	SOUT1	AD15	PLAO[15]

[1] Available only on the 80-lead ADuC7129.
[2] When configured in Mode 1, PO.7 is ECLK by default, or core clock output. To configure it as a clock ouput, the MDCLK bits in PLLCON must be set to 11.

Table 71. GPxCON MMR Bit Designations

Bit	Description
31:30	Reserved
29:28	Select function of Px.7 pin
27:26	Reserved
25:24	Select function of Px.6 pin
23:22	Reserved
21:20	Select function of Px.5 pin
19:18	Reserved
17:16	Select function of Px.4 pin
15:14	Reserved
13:12	Select function of Px.3 pin
11:10	Reserved
9:8	Select function of Px.2 pin
7:6	Reserved
5:4	Select function of Px.1 pin
3:2	Reserved
1:0	Select function of Px.0 pin

GPxPAR Register

Name	Address	Default Value	Access
GP0PAR	0xFFFF0D2C	0x20000000	R/W
GP1PAR	0xFFFF0D3C	0x00000000	R/W
GP3PAR	0xFFFF0D5C	0x00222222	R/W
GP4PAR	0xFFFF0D6C	0x00000000	R/W

GPxPAR programs the parameters for Port 0, Port 1, Port 3, and Port 4. Note that the GPxDAT MMR must always be written after changing the GPxPAR MMR.

Table 72. GPxPAR MMR Bit Designations

Bit	Description
31:29	Reserved
28	Pull-up disable Px.7 pin
27:25	Reserved
24	Pull-up disable Px.6 pin
23:21	Reserved
20	Pull-up disable Px.5 pin
19:17	Reserved
16	Pull-up disable Px.4 pin
15:13	Reserved
12	Pull-up disable Px.3 pin
11:9	Reserved
8	Pull-up disable Px.2 pin
7:5	Reserved
4	Pull-up disable Px.1 pin
3:1	Reserved
0	Pull-up disable Px.0 pin

Table 87. COMxIEN1 MMR Bit Designations

Bit	Name	Description
7	ENAM	Network Address Mode Enable Bit. Set by user to enable network address mode. Cleared by user to disable network address mode.
6	E9BT	9-Bit Transmit Enable Bit. Set by user to enable 9-bit transmit. ENAM must be set. Cleared by user to disable 9-bit transmit.
5	E9BR	9-Bit Receive Enable Bit. Set by user to enable 9-bit receive. ENAM must be set. Cleared by user to disable 9-bit receive.
4	ENI	Network Interrupt Enable Bit.
3	E9BD	Word Length. Set for 9-bit data. E9BT has to be cleared. Cleared for 8-bit data.
2	ETD	Transmitter Pin Driver Enable Bit. Set by user to enable SOUT as an output in slave mode or multimaster mode. Cleared by user; SOUT is three-state.
1	NABP	Network Address Bit, Interrupt Polarity Bit.
0	NAB	Network Address Bit. Set by user to transmit the slave's address. Cleared by user to transmit data.

Table 88. COMxIID1 MMR Bit Designations

Bit 3:1 Status Bits	Bit 0 NINT	Priority	Definition	Clearing Operation
000	1		No Interrupt.	
110	0	2	Matching Network Address.	Read COMxRX.
101	0	3	Address Transmitted, Buffer Empty.	Write data to COMxTX or read COMxIID0.
011	0	1	Receive Line Status Interrupt.	Read COMxSTA0.
010	0	2	Receive Buffer Full Interrupt.	Read COMxRX.
001	0	3	Transmit Buffer Empty Interrupt.	Write data to COMxTX or read COMxIID0.
000	0	4	Modem Status Interrupt.	Read COMxSTA1 register.

Note that to receive a network address interrupt, the slave must ensure that Bit 0 of COMxIEN0 (enable receive buffer full interrupt) is set to 1.

COMxADR is an 8-bit, read/write network address register that holds the address checked for by the network addressable UART. Upon receiving this address, the device interrupts the processor and/or sets the appropriate status bit in COMxIID1.

SPIRX Register

Name	Address	Default Value	Access
SPIRX	0xFFFF0A04	0x00	R

SPIRX is an 8-bit read-only receive register.

SPITX Register

Name	Address	Default Value	Access
SPITX	0xFFFF0A08	0x00	W

SPITX is an 8-bit write-only transmit register.

SPIDIV Register

Name	Address	Default Value	Access
SPIDIV	0xFFFF0A0C	0x1B	R/W

SPIDIV is an 8-bit serial clock divider register.

SPICON Register

Name	Address	Default Value	Access
SPICON	0xFFFF0A10	0x0000	R/W

SPICON is a 16-bit control register.

Table 91. SPICON MMR Bit Designations

Bit	Description
15:13	Reserved.
12	Continuous Transfer Enable. Set by user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the TX register. $\overline{CS}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until TX is empty. Cleared by user to disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPITX register, then a new transfer is initiated after a stall period.
11	Loopback Enable. Set by user to connect MISO to MOSI and test software. Cleared by user to be in normal mode.
10	Slave Output Enable. Set by user to enable the slave output. Cleared by user to disable slave output.
9	Slave Select Input Enable. Set by user in master mode to enable the output.
8	SPIRX Overflow Overwrite Enable. Set by user, the valid data in the RX register is overwritten by the new serial byte received. Cleared by user, the new serial byte received is discarded.
7	SPITX Underflow Mode. Set by user to transmit 0. Cleared by user to transmit the previous data.
6	Transfer and Interrupt Mode (Master Mode). Set by user to initiate transfer with a write to the SPITX register. Interrupt occurs when TX is empty. Cleared by user to initiate transfer with a read of the SPIRX register. Interrupt occurs when RX is full.
5	LSB First Transfer Enable Bit. Set by user, the LSB is transmitted first. Cleared by user, the MSB is transmitted first.
4	Reserved. Should be set to 0.
3	Serial Clock Polarity Mode Bit. Set by user, the serial clock idles high. Cleared by user, the serial clock idles low.
2	Serial Clock Phase Mode Bit. Set by user, the serial clock pulses at the beginning of each serial bit transfer. Cleared by user, the serial clock pulses at the end of each serial bit transfer.
1	Master Mode Enable Bit. Set by user to enable master mode. Cleared by user to enable slave mode.
0	SPI Enable Bit. Set by user to enable the SPI. Cleared to disable the SPI.

Bit	Description
7	Master Serial Clock Enable Bit. Set by user to enable generation of the serial clock in master mode. Cleared by user to disable serial clock in master mode.
6	Loop-Back Enable Bit. Set by user to internally connect the transition to the reception to test user software. Cleared by user to operate in normal mode.
5	Start Back-Off Disable Bit. Set by user in multimaster mode. If losing arbitration, the master immediately tries to retransmit. Cleared by user to enable start back-off. After losing arbitration, the master waits before trying to retransmit.
4	Hardware General Call Enable. When this bit and Bit 3 are set, and have received a general call (Address 0x00) and a data byte, the device checks the contents of the I2C0ALT against the receive register. If the contents match, the device has received a hardware general call. This is used if a device needs urgent attention from a master device without knowing which master it needs to turn to. This is a "to whom it may concern" call. The ADuC7128/ADuC7129 watch for these addresses. The device that requires attention embeds its own address into the message. All masters listen and the one that can handle the device contacts its slave and acts appropriately. The LSB of the I2C0ALT register should always be written to a 1, as per the I²C January 2000 specification.
3	General Call Enable Bit. Set this bit to enable the slave device to acknowledge an I²C general call, Address 0x00 (write). The device then recognizes a data bit. If it receives a 0x06 (reset and write programmable part of slave address by hardware) as the data byte, the I²C interface resets as per the I²C January 2000 specification. This command can be used to reset an entire I²C system. The general call interrupt status bit sets on any general call. The user must take corrective action by setting up the I²C interface after a reset. If it receives a 0x04 (write programmable part of slave address by hardware) as the data byte, the general call interrupt status bit sets on any general call. The user must take corrective action by reprogramming the device address.
2	Reserved.
1	Master Enable Bit. Set by user to enable the master I²C channel. Cleared by user to disable the master I²C channel.
0	Slave Enable Bit. Set by user to enable the slave I²C channel. A slave transfer sequence is monitored for the device address in I2C0ID0, I2C0ID1, I2C0ID2, and I2C0ID3. If the device address is recognized, the part participates in the slave transfer sequence. Cleared by user to disable the slave I²C channel.

I2CxDIV Register

Name	Address	Default Value	Access
I2C0DIV	0xFFFF0830	0x1F1F	R/W
I2C1DIV	0xFFFF0930	0x1F1F	R/W

I2CxDIV are the clock divider registers.

I2CxIDx Register

Name	Address	Default Value	Access
I2C0ID0	0xFFFF0838	0x00	R/W
I2C0ID1	0xFFFF083C	0x00	R/W
I2C0ID2	0xFFFF0840	0x00	R/W
I2C0ID3	0xFFFF0844	0x00	R/W
I2C1ID0	0xFFFF0938	0x00	R/W
I2C1ID1	0xFFFF093C	0x00	R/W
I2C1ID2	0xFFFF0940	0x00	R/W
I2C1ID3	0xFFFF0944	0x00	R/W

I2CxID0, I2CxID1, I2CxID2, and I2CxID3 are slave address device ID registers of I2Cx.

I2CxSSC Register

Name	Address	Default Value	Access
I2C0SSC	0xFFFF0848	0x01	R/W
I2C1SSC	0xFFFF0948	0x01	R/W

I2CxSSC is an 8-bit start/stop generation counter. It holds off SDA low for start and stop conditions.

Table 96. Element Input/Output

PLA Block 0			PLA Block 1		
Element	**Input**	**Output**	**Element**	**Input**	**Output**
0	P1.0	P1.7	8	P3.0	P4.0
1	P1.1	P0.4	9	P3.1	P4.1
2	P1.2	P0.5	10	P3.2	P4.2
3	P1.3	P0.6	11	P3.3	P4.3
4	P1.4	P0.7	12	P3.4	P4.4
5	P1.5	P2.0	13	P3.5	P4.5
6	P1.6	P2.1	14	P3.6	P4.6
7	P0.0	P2.2	15	P3.7	P4.7

PLA MMRs Interface

The PLA peripheral interface consists on 21 MMRs, as shown in Table 97.

Table 97. PLA MMRs

Name	Description
PLAELMx	Element 0 to Element 15 Control Registers. Configure the input and output mux of each element, select the function in the look-up table, and bypass/use the flip-flop.
PLACLK	Clock Selection for the Flip-Flops of Block 0 and Clock Selection for the Flip-Flops of Block 1.
PLAIRQ	Enable IRQ0 and/or IRQ1. Select the source of the IRQ.
PLAADC	PLA Source from ADC Start Conversion Signal.
PLADIN	Data Input MMR for PLA.
PLAOUT	Data Output MMR for PLA. This register is always updated.

A PLA tool is provided in the development system to easily configure the PLA.

Table 98. PLAELMx MMR Bit Designations

Bit	Value	PLAELM0	PLAELM1 to PLAELM7	PLAELM8	PLAELM9 to PLAELM15	Description
31:11						Reserved.
10:9	00	Element 15	Element 0	Element 7	Element 8	Mux (0) Control. Select feedback source.
	01	Element 2	Element 2	Element 10	Element 10	
	10	Element 4	Element 4	Element 12	Element 12	
	11	Element 6	Element 6	Element 14	Element 14	
8:7	00	Element 1	Element 1	Element 9	Element 9	Mux (1) Control. Select feedback source.
	01	Element 3	Element 3	Element 11	Element 11	
	10	Element 5	Element 5	Element 13	Element 13	
	11	Element 7	Element 7	Element 15	Element 15	
6						Mux (2) Control. Set by user to select the output of Mux (0). Cleared by user to select the bit value from PLADIN.
5						Mux (3) Control. Set by user to select the input pin of the particular element. Cleared by user to select the output of Mux (1).
4:1						Look-Up Table Control.
	0000					0
	0001					NOR
	0010					B AND NOT A
	0011					NOT A
	0100					A AND NOT B
	0101					NOT B
	0110					EXOR
	0111					NAND
	1000					AND
	1001					EXNOR
	1010					B
	1011					NOT A OR B
	1100					A
	1101					A OR NOT B
	1110					OR
	1111					1
0						Mux (4) Control. Set by user to bypass the flip-flop. Cleared by user to select the flip-flop. Cleared by default.

Table 111. T1CON MMR Bit Designations

Bit	Value	Description
31:18		Reserved. Should be set to 0 by the user.
17		Event Select Bit. Set by user to enable time capture of an event. Cleared by user to disable time capture of an event.
16:12		Event Select Range, 0 to 31. The events are as described in the introduction to the timers.
11:9		Clock Select.
	000	Core Clock (Default).
	001	32.768 kHz Oscillator.
	010	P1.0.
	011	P0.6.
8		Count Up. Set by user for Timer1 to count up. Cleared by user for Timer1 to count down (default).
7		Timer1 Enable Bit. Set by user to enable Timer1. Cleared by user to disable Timer1 (default).
6		Timer1 Mode. Set by user to operate in periodic mode. Cleared by user to operate in free-running mode (default).
5:4		Format.
	00	Binary (Default).
	01	Reserved.
	10	Hours:Minutes:Seconds:Hundredths: 23 Hours to 0 Hours.
	11	Hours:Minutes:Seconds:Hundredths: 255 Hours to 0 Hours.
3:0		Prescaler.
	0000	Source Clock/1 (Default).
	0100	Source Clock/16.
	1000	Source Clock/256.
	1111	Source Clock/32768.

The XMxPAR are registers that define the protocol used for accessing the external memory for each memory region.

Table 121. XMxPAR MMR Bit Designations

Bit	Description
15	Enable Byte Write Strobe. This bit is only used for two, 8-bit memory sharing the same memory region. Set by user to gate the AD0 output with the WS output. This allows byte write capability without using $\overline{BHE}$ and $\overline{BLE}$ signals. Cleared by user to use $\overline{BHE}$ and $\overline{BLE}$ signals.
14:12	Number of Wait States on the Address Latch Enable Strobe.
11	Reserved.
10	Extra Address Hold Time. Set by the user to disable extra hold time. Cleared by the user to enable one clock cycle of hold on the address in read and write.
9	Extra Bus Transition Time on Read. Set by the user to disable extra bus transition time. Cleared by the user to enable one extra clock before and after the read select ($\overline{RS}$).
8	Extra Bus Transition Time on Write. Set by the user to disable extra bus transition time. Cleared by the user to enable one extra clock before and after the write select ($\overline{WS}$).
7:4	Number of Write Wait States. Select the number of wait states added to the length of the $\overline{WS}$ pulse. 0x0 is 1 clock cycle; 0xF is 16 clock cycles (default value).
3:0	Number of Read Wait States. Select the number of wait states added to the length of the $\overline{RS}$ pulse. 0x0 is 1 clock cycle; 0xF is 16 clock cycles (default value).

TIMING DIAGRAMS

Figure 62 through Figure 65 show the timing for a read cycle (see Figure 62), a read cycle with address hold and bus turn cycles (see Figure 63), a write cycle with address hold and write hold cycles (see Figure 64), and a write cycle with wait states (see Figure 65).

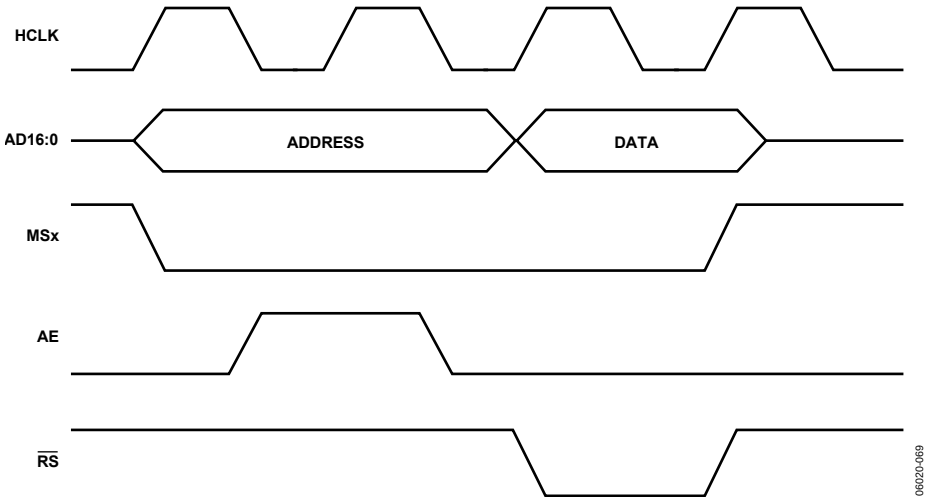

Figure 62. External Memory Read Cycle

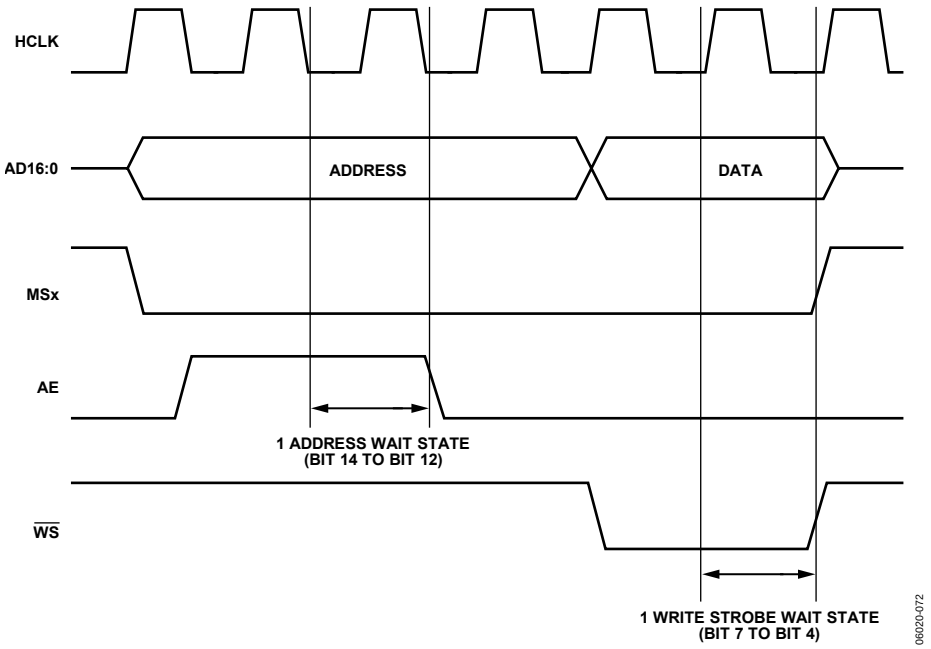

Figure 65. External Memory Write Cycle with Wait States

In these cases, tie the AGND pins and IOGND pins of the ADuC7128/ADuC7129 to the analog ground plane, as shown in Figure 69b. In systems with only one ground plane, ensure that the digital and analog components are physically separated onto separate halves of the board such that digital return currents do not flow near analog circuitry and vice versa. The ADuC7128/ADuC7129 can then be placed between the digital and analog sections, as illustrated in Figure 69c.

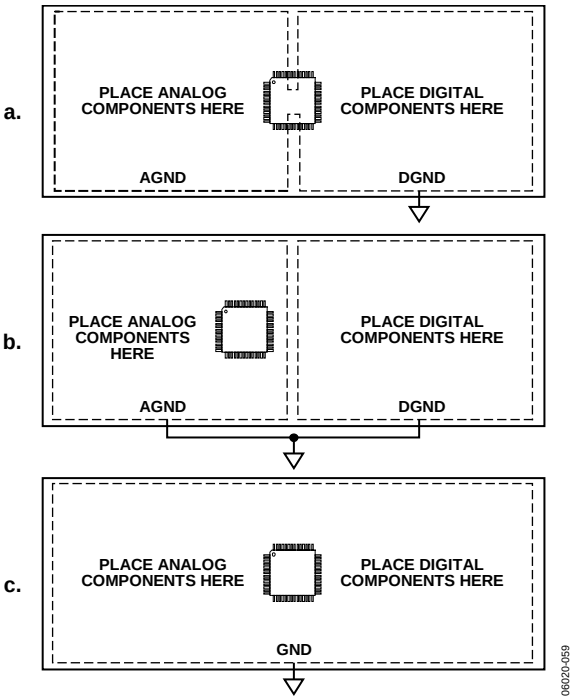

Figure 69. System Grounding Schemes

In all of these scenarios, and in more complicated real-life applications, keep in mind the flow of current from the supplies and back to ground. Make sure the return paths for all currents are as close as possible to the paths the currents took to reach their destinations. For example, do not power components on the analog side (see Figure 69b) with IOV_{DD} since that would force return currents from IOV_{DD} to flow through AGND. Avoid digital currents from flowing under analog circuitry, which could happen if the user places a noisy digital chip on the left half of the board (see Figure 69c). Whenever possible, avoid large discontinuities in the ground planes (such as are formed by a long trace on the same layer) because they force return signals to travel a longer path. Make all connections to the ground plane directly, with little or no trace separating the pin from its via to ground.

If a user plans to connect fast logic signals (rise/fall time < 5 ns) to any of the digital inputs of the ADuC7128/ADuC7129, add a series resistor to each relevant line to keep rise and fall times longer than 5 ns at the ADuC7128/ADuC7129 input pins.

A value of 100 Ω or 200 Ω is usually sufficient to prevent high speed signals from coupling capacitively into the ADuC7128/ADuC7129 and affecting the accuracy of ADC conversions.

CLOCK OSCILLATOR

The clock source for the ADuC7128/ADuC7129 can be generated by the internal PLL or by an external clock input. To use the internal PLL, connect a 32.768 kHz parallel resonant crystal between XCLKI and XCLKO as shown Figure 70. External capacitors should be connected as per the crystal manufacturer's recommendations. Note that the crystal pads already have an internal capacitance of typically 10 pF. Users should ensure that the total capacitance (10 pF internal + external capacitance) does not exceed the manufacturer rating.

The 32 kHz crystal allows the PLL to lock correctly to give a frequency of 41.78 MHz. If no external crystal is present, the internal oscillator is used to give a frequency of 41.78 MHz ± 3% typically.

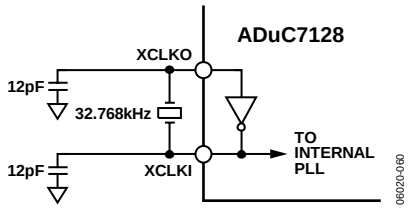

Figure 70. External Parallel Resonant Crystal Connections

To use an external source clock input instead of the PLL, Bit 1 and Bit 0 of PLLCON must be modified. The external clock uses the XCLK pin.

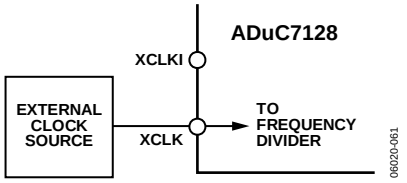

Figure 71. Connecting an External Clock Source

Whether using the internal PLL or an external clock source, the specified operational clock speed range of the ADuC7128/ADuC7129 is 50 kHz to 41.78 MHz to ensure correct operation of the analog peripherals and Flash/EE.