DATASHEET

Apollo4 Blue SoC

Ultra-low Power Apollo SoC Family

Doc. ID: DS-A4B-1p3p0

Doc. Revision: 1.3.0, February 2023

Features

Ultra-low supply current:

- 5 µA/MHz active mode current
- Low-power sleep and deep sleep modes with selectable levels of RAM/cache retention

High-performance Arm Cortex-M4 Processor with FPU:

- 96/192 MHz operating modes
- Floating Point Unit, Memory Protection Unit
- Secure boot

Bluetooth Low Energy 5.1[1]:

- Data rate: 1 Mbps and 2 Mbps
- Extended advertising packets
- Angle of Arrival (AoA) and Angle of Departure (AoD)
- Tx output power: -10 dBm to +6 dBm
- Rx sensitivity: -95.5 dBm at 1 Mbps, -92 dBm at 2 Mbps

Ultra-low-power memory:

- Up to 2 MB of non-volatile memory (NVM) for code/data
- Up to 1.8 MB of low power RAM for code/data

Ultra-low-power interface for off-chip sensors:

- 8-bit, 10-bit and 12-bit ADC modes, 11 selectable channels
- Up to 2.8 MS/s sampling rate (8-bit mode)
- Temperature sensor with ±3ºC accuracy

Ultra-low-power flexible serial peripherals:

- 2x 2/4/8-bit SPI master interface
- 7x I^{2}C / SPI masters for peripheral communication
- I^{2}C/SPI slave for host communications
- 4x UART modules with flow control
- USB 2.0 HS/FS device controller
- SDIO (SD3.0) / eMMC (v4.51)

Display:

- MIPI DSI 1.2 with single data lane up to 500 Mbps
- Up to 500 x 500 resolution
- 4 layers with full alpha blending
- Frame buffer decompression

Graphics:

- 2D/2.5D graphics accelerator
- Texture and frame buffer compression

1. The *Bluetooth*® word mark and logos are registered trademarks owned by the Bluetooth SIG, Inc. and any use of such marks is under license. Other trademarks and trade names are those of their respective owners.

Audio Processing:

- Stereo low-power analog microphones
- 4x stereo digital microphones
- 2x full duplex I^{2}S ports with ASRC

Rich set of clock sources:

- 32 MHz and 32.768 kHz crystal oscillators
- 1 kHz low-frequency RC oscillator
- 2x high-frequency RC oscillators - 192/384 MHz

Power Management:

- Operating Voltage: 1.71 - 2.2 V
- Temp Range: -20°C to 60°C
- SIMO buck / BLE buck
- Multiple I/O voltages supported

Applications

- Smart watches/bands
- Activity and fitness monitors
- Motion and tracking devices
- Security and alarm systems
- Wireless sensors and IoT
- Consumer electronics
- Home automation
- Consumer medical devices

Package

- 4.7 mm x 4.7 mm, 12 x 12 BGA (131 pins / 81 GPIO)

Description

Ambiq®, the leader in low-power System-on-Chip (SoC) solutions, has once again raised the bar with the Apollo4 Blue SoC. With the lowest dynamic and sleep mode power on the market, the Apollo4 Blue allows designers of next generation wearables and smart devices to take their innovative products to the next level.

The Apollo4 Blue SoC is the 4th generation system processor solution built upon Ambiq's proprietary Subthreshold Power-Optimized Technology (SPOT®) platform. Its complete hardware and software solution enables the battery-powered endpoint devices of tomorrow to achieve a higher level of intelligence without sacrificing battery life.

The Apollo4 Blue with Bluetooth® LE is built on the 32-bit Arm® Cortex®-M4 core with Floating Point Unit (FPU) and is available in BGA packaging. With up to 2 MB of MRAM and 1.8 MB of SRAM, the Apollo4 Blue has more than enough compute power and storage to handle complex algorithms and neural networks while displaying vibrant, crystal clear, and smooth graphics.

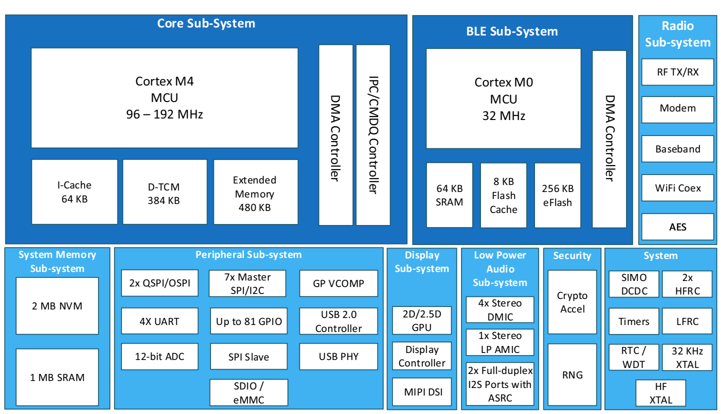


Table of Contents

List of Figures

List of Tables

1. Apollo4 Blue SoC Package Pins

1.1 Pin Configuration

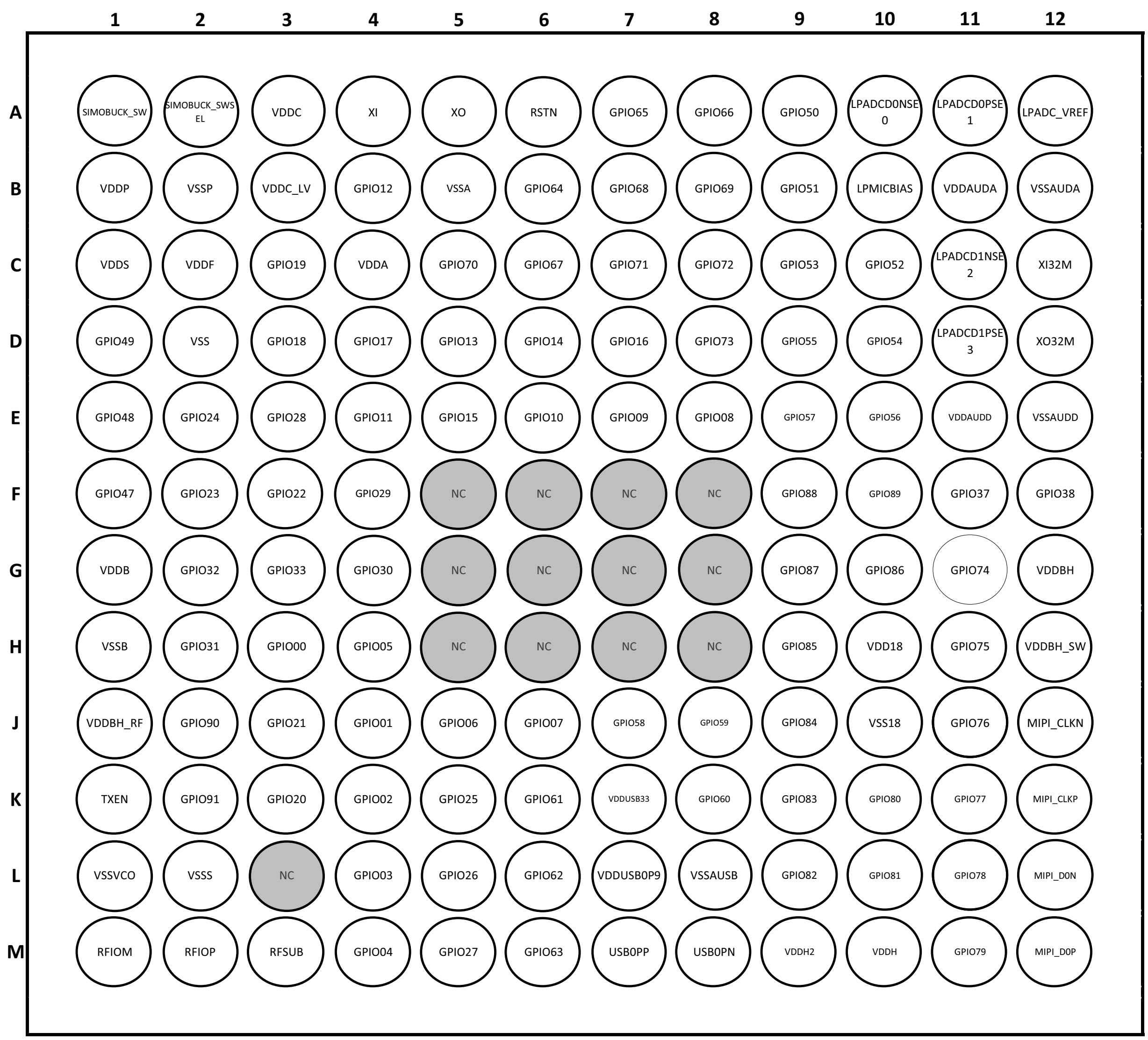


Figure 1. Apollo4 Blue SoC BGA Pin Configuration Diagram - Top View

1.2 Pin Connections

The following table lists the external pins of the Apollo4 Blue SoC and their available functions.

NOTE

Use of the DPI-2 interface, represented in the following table by pad functions DISP_D0 - DISP_D23, DISP_VS, DISP_HS, DISP_DE, DISP_PCLK, DISP_SD and DISP_CM, is not recommended or supported.

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
B5	-	-	VSSA	Analog Ground	Ground
H1	-	-	VSSB	BLE unregulated ground connection	Power
B2	-	-	VSSP	Ground Connection for buck regs	Ground
A12	-	-	LPADC_VREF	LP ADC Reference Decap	Analog
A10	-	-	LPADCD0NSE0	LP Analog to Digital Converter SE0/DiffN IN0	Input
A11	-	-	LPADCD0PSE1	LP Analog to Digital Converter SE1/DiffP IN0	Input
C11	-	-	LPADCD1NSE2	LP Analog to Digital Converter SE2/DiffN IN1	Input
D11	-	-	LPADCD1PSE3	LP Analog to Digital Converter SE3/DiffP IN1	Input
B10	-	-	LPMICBIAS	LP Microphone Bias	Output
J12	-	-	MIPI_CLKN	MIPI DPHY Clock Lane N	I/O
K12	-	-	MIPI_CLKP	MIPI DPHY Clock Lane P	I/O
L12	-	-	MIPI_D0N	MIPI DPHY Data Lane 0N	I/O
M12	-	-	MIPI_D0P	MIPI DPHY Data Lane 0P	I/O
M1	-	-	RFIOM	RF IO negative	Analog
M2	-	-	RFIOP	RF IO positive	Analog
M3	-	-	RFSUB	RF Substrate ground	Ground
A6	-	-	RSTN	External reset input (aka nRST)	Input
A1	-	-	SIMOBUCK_SW	SIMO Buck converter inductor switch output	Power
A2	-	-	SIMOBUCK_SWSEL	SIMO Buck converter inductor switch input	Power
K1	-	-	TXEN	Transmitter enable	Output
M8	-	-	USB0PN	The differential input/output signals of the PHY that support multiple modes. Depending on mode of operation they are either signaling 3.3V or 800mV differential.	Power
M7	-	-	USB0PP	The differential input/output signals of the PHY that support multiple modes. Depending on mode of operation they are either signaling 3.3V or 800mV differential.	Power
H10	-	-	VDD18	VDD supply for MIPI PHY	Power
C4	-	-	VDDA	Analog voltage supply	Power
B11	-	-	VDDAUDA	Analog Audio Voltage supply	Power

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
E11	-	-	VDDAUDD	Digital Audio Voltage supply	Power
G1	-	-	VDDB	BLE voltage supply (VCC)	Power
G12	-	-	VDDBH	BLE Buck converter VOUT	Power
J1	-	-	VDDBH_RF	BLE Buck converter RF VOUT	Power
H12	-	-	VDDBH_SW	BLE Buck converter inductor switch	Power
A3	-	-	VDDC	Core Buck converter VOUT	Power
B3	-	-	VDDC_LV	Core_LV Buck converter VOUT	Power
C2	-	-	VDDF	Mem Buck converter VOUT	Power
M10	-	-	VDDH	High voltage domain power supply	Power
M9	-	-	VDDH2	High voltage domain2 power supply	Power
B1	-	-	VDDP	VDD supply to I/O pads (Core)	Power
C1	-	-	VDDS	SRAM high voltage supply	Power
L7	-	-	VDDUSB0P9	USB 0.9v analog voltage supply	Power
K7	-	-	VDDUSB33	USB 3.3v voltage supply	Power
D2	-	-	VSS	Digital Ground for VDDF and PADS	Ground
J10	-	-	VSS18	MIPI PHY Analog Ground	Ground
B12	-	-	VSSAUDA	Analog Audio Ground	Ground
E12	-	-	VSSAUDD	Digital Audio Ground	Ground
L8	-	-	VSSAUSB	USB PHY Analog Ground	Ground
L2	-	-	VSSS	RF substrate Ground for BLE Core	Ground
L1	-	-	VSSVCO	BLECORE Signals - Ground for VCO	Ground
A4	-	-	XI	32.768kHz crystal input	XT
C12	-	-	XI32M	32MHz crystal input	XT24M
A5	-	-	XO	32.768kHz crystal output	XT
D12	-	-	XO32M	32MHz crystal output	XT24M

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
H3	0	0	SWTRACECLK	Serial Wire Debug Trace Clock	Output
		1	SLSCL	I^2C Slave clock	Input
		2	SLSCK	SPI Slave clock	Input
		3	GPIO00	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5	UART1TX	UART1 transmit output	Output
		6	CT0	Timer/counter 0	Output
		7	NCE0	IOMSTR/MSPI N Chip Select 0	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO00	Fast PIO	
J4	1	0	SWTRACE0	Serial Wire Debug Trace Output 0	Output
		1	SLSDAWIR3	I^2C Slave I/O data (I^2C) 3 Wire Data (SPI)	Bidirectional Open Drain
		2	SLMOSI	SPI Slave input data	Input
		3	GPIO01	General purpose I/O	I/O
		4	UART2TX	UART2 transmit output	Output
		5	UART3TX	UART3 transmit output	Output
		6	CT1	Timer/counter 1	Output
		7	NCE1	IOMSTR/MSPI N Chip Select 1	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO01	Fast PIO	
K4	2	0	SWTRACE1	Serial Wire Debug Trace Output 1	Output
		1	SLMISO	SPI Slave output data	Output
		2	TRIG1	ADC trigger input	Input
		3	GPIO02	General purpose I/O	I/O
		4	UART0RX	UART0 receive input	Input
		5	UART1RX	UART1 receive input	Input
		6	CT2	Timer/counter 2	Output
		7	NCE2	IOMSTR/MSPI N Chip Select 2	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO02	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
L4	3	0	SWTRACE2	Serial Wire Debug Trace Output 2	Output
		1	SLnCE	SPI Slave chip enable	Input
		2	SWO	Serial Wire Debug	Output
		3	GPIO03	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	UART3RX	UART3 receive input	Input
		6	CT3	Timer/counter 3	Output
		7	NCE3	IOMSTR/MSPI N Chip Select 3	Output
		8			
		9			
		10			
		11	FPIO03	Fast PIO	
M4	4	0	SWTRACE3	Serial Wire Debug Trace Output 3	Output
		1	SLINT	Configurable Slave Interrupt	Output
		2	32kHzXT	32kHZ from analog	Output
		3	GPIO04	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	UART1RTS	UART1 Request to Send (RTS)	Output
		6	CT4	Timer/counter 4	Output
		7	NCE4	IOMSTR/MSPI N Chip Select 4	Output
		8			
		9	I2S0_SDIN	I2S0 Data input	Input
		10	I2S1_SDIN	I2S1 Data input	Input
		11	FPIO04	Fast PIO	
H4	5	0	M0SCL	I^2C Master 0 clock	Open Drain Output
		1	M0SCK	SPI Master 0 clock	Output
		2	I2S0_CLK	I2S0 Bit clock	Input
		3	GPIO05	General purpose I/O	I/O
		4	UART2RTS	UART2 Request to Send (RTS)	Output
		5	UART3RTS	UART3 Request to Send (RTS)	Output
		6	CT5	Timer/counter 5	Output
		7	NCE5	IOMSTR/MSPI N Chip Select 5	Output
		8			
		9			
		10	I2S1_CLK	I2S1 Bit clock	Input
		11	FPIO05	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
J5	6	0	M0SDAWIR3	I^2C Master 0 I/O data (I^2C) 3 Wire data (SPI)	Bidirectional Open Drain
		1	M0MOSI	SPI Master 0 output data	Output
		2	I2S0_DATA	I2S0 Data	Bidirectional
		3	GPIO06	General purpose I/O	I/O
		4	UART0CTS	UART0 Clear to Send (CTS)	Input
		5	UART1CTS	UART1 Clear to Send (CTS) input	Input
		6	CT6	Timer/counter 6	Output
		7	NCE6	IOMSTR/MSPI N Chip Select 6	Output
		8			
		9	I2S0_SDOUT	I2S0 Data output	Output
		10	I2S1_SDOUT	I2S1 Data output	Output
		11	FPIO06	Fast PIO	
J6	7	0	M0MISO	SPI Master 0 input data	Input
		1	TRIG0	ADC trigger input	Input
		2	I2S0_WS	I2S0 L/R clock	Input
		3	GPIO07	General purpose I/O	I/O
		4	UART2CTS	UART2 Clear to Send (CTS) input	Input
		5	UART3CTS	UART3 Clear to Send (CTS) input	Input
		6	CT7	Timer/counter 7	Output
		7	NCE7	IOMSTR/MSPI N Chip Select 7	Output
		8			
		9			
		10	I2S1_WS	I2S1 L/R clock	Input
		11	FPIO07	Fast PIO	
E8	8	0	CMPRF1	Comparator reference 1	Input
		1	TRIG1	ADC trigger input	Input
		2			
		3	GPIO08	General purpose I/O	I/O
		4	M1SCL	I^2C Master 1 clock	Open Drain Output
		5	M1SCK	SPI Master 1 clock	Output
		6	CT8	Timer/counter 8	Output
		7	NCE8	IOMSTR/MSPI N Chip Select 8	Output
		8			
		9			
		10			
		11	FPIO08	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
E7	9	0	CMPRF0	Comparator reference 0	Input
		1	TRIG2	ADC trigger input	Input
		2			
		3	GPIO09	General purpose I/O	I/O
		4	M1SDAWIR3	I^2C Master 1 I/O data (I^2C) 3 Wire data (SPI)	Bidirectional Open Drain
		5	M1MOSI	SPI Master 1 output data	Output
		6	CT9	Timer/counter 9	Output
		7	NCE9	IOMSTR/MSPI N Chip Select 9	Output
		8			
		9			
		10			
		11	FPIO09	Fast PIO	
E6	10	0	CMPIN0	Voltage comparator input 0	Input
		1	TRIG3	ADC trigger input	Input
		2			
		3	GPIO10	General purpose I/O	I/O
		4	M1MISO	SPI Master 1 input data	Input
		5			
		6	CT10	Timer/counter 10	Output
		7	NCE10	IOMSTR/MSPI N Chip Select 10	Output
		8			
		9	DISP_TE	Display TE input	Input
		10			
		11	FPIO10	Fast PIO	
E4	11	0	CMPIN1	Voltage comparator input 1	Input
		1	TRIG0	ADC trigger input	Input
		2	I2S0_CLK	I2S0 Bit clock	Input
		3	GPIO11	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	UART3RX	UART3 receive input	Input
		6	CT11	Timer/counter 11	Output
		7	NCE11	IOMSTR/MSPI N Chip Select 11	Output
		8			
		9			
		10			
		11	FPIO11	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
B4	12	0	ADCSE7	Analog to Digital Converter SE IN7	Input
		1	TRIG1	ADC trigger input	Input
		2	I2S0_DATA	I2S0 Data	Bidirectional
		3	GPIO12	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5	UART1TX	UART1 transmit output	Output
		6	CT12	Timer/counter 12	Output
		7	NCE12	IOMSTR/MSPI N Chip Select 12	Output
		8			
		9	CMPRF2	Comparator reference 2	Input
		10	I2S0_SDOUT	I2S0 Data output	Output
		11	FPIO12	Fast PIO	
D5	13	0	ADCSE6	Analog to Digital Converter SE IN6	Input
		1	TRIG2	ADC trigger input	Input
		2	I2S0_WS	I2S0 L/R clock	Input
		3	GPIO13	General purpose I/O	I/O
		4	UART2TX	UART2 transmit output	Output
		5	UART3TX	UART3 transmit output	Output
		6	CT13	Timer/counter 13	Output
		7	NCE13	IOMSTR/MSPI N Chip Select 13	Output
		8			
		9			
		10			
		11	FPIO13	Fast PIO	
D6	14	0	ADCSE5	Analog to Digital Converter SE IN5	Input
		1	TRIG3	ADC trigger input	Input
		2			
		3	GPIO14	General purpose I/O	I/O
		4			
		5	UART1RX	UART1 receive input	Input
		6	CT14	Timer/counter 14	Output
		7	NCE14	IOMSTR/MSPI N Chip Select 14	Output
		8			
		9			
		10	I2S0_SDIN	I2S0 Data input	Input
		11	FPIO14	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
E5	15	0	ADCSE4	Analog to Digital Converter SE IN4	Input
		1	TRIG0	ADC trigger input	Input
		2			
		3	GPIO15	General purpose I/O	I/O
		4			
		5	UART3RX	UART3 receive input	Input
		6	CT15	Timer/counter 15	Output
		7	NCE15	IOMSTR/MSPI N Chip Select 15	Output
		8			
		9			
		10	REFCLK_EXT	External Reference Clock	Input
		11	FPIO15	Fast PIO	
D7	16	0	ADCSE3	Analog to Digital Converter SE IN3	Input
		1	TRIG1	ADC trigger input	Input
		2	I2S1_CLK	I2S1 Bit clock	Input
		3	GPIO16	General purpose I/O	I/O
		4			
		5	UART1RTS	UART1 Request to Send (RTS)	Output
		6	CT16	Timer/counter 16	Output
		7	NCE16	IOMSTR/MSPI N Chip Select 16	Output
		8			
		9			
		10			
		11	FPIO16	Fast PIO	
D4	17	0	ADCSE2	Analog to Digital Converter SE IN2	Input
		1	TRIG2	ADC trigger input	Input
		2	I2S1_DATA	I2S1 Data	Bidirectional
		3	GPIO17	General purpose I/O	I/O
		4			
		5	UART3RTS	UART3 Request to Send (RTS)	Output
		6	CT17	Timer/counter 17	Output
		7	NCE17	IOMSTR/MSPI N Chip Select 17	Output
		8			
		9	I2S1_SDOUT	I2S1 Data output	Output
		10			
		11	FPIO17	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
D3	18	0	ADCSE1	Analog to Digital Converter SE IN1	Input
		1			
		2	I2S1_WS	I2S1 L/R clock	Input
		3	GPIO18	General purpose I/O	I/O
		4	UART0CTS	UART0 Clear to Send (CTS)	Input
		5	UART1CTS	UART1 Clear to Send (CTS) input	Input
		6	CT18	Timer/counter 18	Output
		7	NCE18	IOMSTR/MSPI N Chip Select 18	Output
		8			
		9			
		10			
		11	FPIO18	Fast PIO	
C3	19	0	ADCSE0	Analog to Digital Converter SE IN0	Input
		1			
		2			
		3	GPIO19	General purpose I/O	I/O
		4	UART2CTS	UART2 Clear to Send (CTS) input	Input
		5	UART3CTS	UART3 Clear to Send (CTS) input	Input
		6	CT19	Timer/counter 19	Output
		7	NCE19	IOMSTR/MSPI N Chip Select 19	Output
		8			
		9	I2S1_SDIN	I2S1 Data input	Input
		10			
		11	FPIO19	Fast PIO	
K3	20	0	SWDCK	Software debug clock Input	Input
		1	TRIG1	ADC trigger input	Input
		2			
		3	GPIO20	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5	UART1TX	UART1 transmit output	Output
		6	CT20	Timer/counter 20	Output
		7	NCE20	IOMSTR/MSPI N Chip Select 20	Output
		8			
		9			
		10			
		11	FPIO20	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
J3	21	0	SWDIO	Software data I/O	Bidirectional 3-state
		1	TRIG2	ADC trigger input	Input
		2			
		3	GPIO21	General purpose I/O	I/O
		4	UART2TX	UART2 transmit output	Output
		5	UART3TX	UART3 transmit output	Output
		6	CT21	Timer/counter 21	Output
		7	NCE21	IOMSTR/MSPI N Chip Select 21	Output
		8			
		9			
		10			
		11	FPIO21	Fast PIO	
F3	22	0	M7SCL	I^2C Master 7 Clk	Bidirectional Open Drain
		1	M7SCK	SPI Master 7 Clk	Output
		2	SWO	Serial Wire Debug	Output
		3	GPIO22	General purpose I/O	I/O
		4	UART0RX	UART0 receive input	Input
		5	UART1RX	UART1 receive input	Input
		6	CT22	Timer/counter 22	Output
		7	NCE22	IOMSTR/MSPI N Chip Select 22	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO22	Fast PIO	
F2	23	0	M7SDAWIR3	I^2C Master 7 I/O data (I^2C) 3 Wire data (SPI)	Bidirectional Open Drain
		1	M7MOSI	SPI Master 7 data out	Output
		2	SWO	Serial Wire Debug	Output
		3	GPIO23	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	UART3RX	UART3 receive input	Input
		6	CT23	Timer/counter 23	Output
		7	NCE23	IOMSTR/MSPI N Chip Select 23	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO23	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
E2	24	0	M7MISO	SPI Master 7 data in	Input
		1	TRIG3	ADC trigger input	Input
		2	SWO	Serial Wire Debug	Output
		3	GPIO24	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	UART1RTS	UART1 Request to Send (RTS)	Output
		6	CT24	Timer/counter 24	Output
		7	NCE24	IOMSTR/MSPI N Chip Select 24	Output
		8			
		9			
		10			
		11	FPIO24	Fast PIO	
K5	25	0	M2SCL	I^2C Master 2 clock	Open Drain Output
		1	M2SCK	SPI Master 2 clock	Output
		2			
		3	GPIO25	General purpose I/O	I/O
		4			
		5			
		6	CT25	Timer/counter 25	Output
		7	NCE25	IOMSTR/MSPI N Chip Select 25	Output
		8			
		9			
		10			
		11	FPIO25	Fast PIO	
L5	26	0	M2SDAWIR3	I^2C Master 2 I/O data (I^2C) 3 Wire data (SPI)	Bidirectional Open Drain
		1	M2MOSI	SPI Master 2 output data	Output
		2			
		3	GPIO26	General purpose I/O	I/O
		4			
		5			
		6	CT26	Timer/counter 26	Output
		7	NCE26	IOMSTR/MSPI N Chip Select 26	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO26	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
M5	27	0	M2MISO	SPI Master 2 input data	Input
		1	TRIG0	ADC trigger input	Input
		2			
		3	GPIO27	General purpose I/O	I/O
		4			
		5			
		6	CT27	Timer/counter 27	Output
		7	NCE27	IOMSTR/MSPI N Chip Select 27	Output
		8			
		9	I2S0_SDIN	I2S0 Data input	Input
		10			
		11	FPIO27	Fast PIO	
E3	28	0	SWO	Serial Wire Debug	Output
		1	VCMPO	Output of the voltage comparator signal	
		2	I2S0_CLK	I2S0 Bit clock	Input
		3	GPIO28	General purpose I/O	I/O
		4	UART2CTS	UART2 Clear to Send (CTS) input	Input
		5			
		6	CT28	Timer/counter 28	Output
		7	NCE28	IOMSTR/MSPI N Chip Select 28	Output
		8			
		9			
		10			
		11	FPIO28	Fast PIO	
F4	29	0	TRIG0	ADC trigger input	Input
		1	VCMPO	Output of the voltage comparator signal	
		2	I2S0_DATA	I2S0 Data	Bidirectional
		3	GPIO29	General purpose I/O	I/O
		4	UART1CTS	UART1 Clear to Send (CTS) input	Input
		5			
		6	CT29	Timer/counter 29	Output
		7	NCE29	IOMSTR/MSPI N Chip Select 29	Output
		8			
		9	I2S0_SDOUT	I2S0 Data output	Output
		10			
		11	FPIO29	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G4	30	0	TRIG1	ADC trigger input	Input
		1	VCMPO	Output of the voltage comparator signal	
		2	I2S0_WS	I2S0 L/R clock	Input
		3	GPIO30	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5			
		6	CT30	Timer/counter 30	Output
		7	NCE30	IOMSTR/MSPI N Chip Select 30	Output
		8			
		9			
		10			
		11	FPIO30	Fast PIO	
H2	31	0	M3SCL	I^2C Master 3 clock	Open Drain Output
		1	M3SCK	SPI Master 3 clock	Output
		2			
		3	GPIO31	General purpose I/O	I/O
		4	UART2TX	UART2 transmit output	Output
		5			
		6	CT31	Timer/counter 31	Output
		7	NCE31	IOMSTR/MSPI N Chip Select 31	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO31	Fast PIO	
G2	32	0	M3SDAWIR3	I^2C Master 3 I/O data (I^2C) 3 Wire data (SPI)	Bidirectional Open Drain
		1	M3MOSI	SPI Master 3 output data	Output
		2			
		3	GPIO32	General purpose I/O	I/O
		4	UART0RX	UART0 receive input	Input
		5			
		6	CT32	Timer/counter 32	Output
		7	NCE32	IOMSTR/MSPI N Chip Select 32	Output
		8			
		9			
		10			
		11	FPIO32	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G3	33	0	M3MISO	SPI Master 3 input data	Input
		1	CLKOUT	Oscillator output clock	Output
		2			
		3	GPIO33	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5			
		6	CT33	Timer/counter 33	Output
		7	NCE33	IOMSTR/MSPI N Chip Select 33	Output
		8			
		9	DISP_TE	Display TE input	Input
		10			
		11	FPIO33	Fast PIO	
F11	37	0	MSPI1_0	MSPI Master 1 Interface Signal	I/O
		1	TRIG1	ADC trigger input	Input
		2	32kHzXT	32kHZ from analog	Output
		3	GPIO37	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	DISP_D15	Display Data 15	Output
		6	CT37	Timer/counter 37	Output
		7	NCE37	IOMSTR/MSPI N Chip Select 37	Output
		8			
		9			
		10			
		11	FPIO37	Fast PIO	
F12	38	0	MSPI1_1	MSPI Master 1 Interface Signal	I/O
		1	TRIG2	ADC trigger input	Input
		2	SWTRACECLK	Serial Wire Debug Trace Clock	Output
		3	GPIO38	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	DISP_D16	Display Data 16	Output
		6	CT38	Timer/counter 38	Output
		7	NCE38	IOMSTR/MSPI N Chip Select 38	Output
		8			
		9			
		10			
		11	FPIO38	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
F1	47	0	M5SCL	I^2C Master 5 Clk	Bidirectional Open Drain
		1	M5SCK	SPI Master 5 Clk	Output
		2	I2S1_CLK	I2S1 Bit clock	Input
		3	GPIO47	General purpose I/O	I/O
		4	UART0RX	UART0 receive input	Input
		5	UART1RX	UART1 receive input	Input
		6	CT47	Timer/counter 47	Output
		7	NCE47	IOMSTR/MSPI N Chip Select 47	Output
		8			
		9			
		10	I2S0_CLK	I2S0 Bit clock	Input
		11	FPIO47	Fast PIO	
E1	48	0	M5SDAWIR3	I^2C Master 5 I/O data (I^2C) 3 Wire data (SPI)	Bidirectional Open Drain
		1	M5MOSI	SPI Master 5 data out	Output
		2	I2S1_DATA	I2S1 Data	Bidirectional
		3	GPIO48	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	UART3RX	UART3 receive input	Input
		6	CT48	Timer/counter 48	Output
		7	NCE48	IOMSTR/MSPI N Chip Select 48	Output
		8			
		9	I2S1_SDOUT	I2S1 Data output	Output
		10	I2S0_SDOUT	I2S0 Data output	Output
		11	FPIO48	Fast PIO	
D1	49	0	M5MISO	SPI Master 5 data in	Input
		1	TRIG0	ADC trigger input	Input
		2	I2S1_WS	I2S1 L/R clock	Input
		3	GPIO49	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	UART1RTS	UART1 Request to Send (RTS)	Output
		6	CT49	Timer/counter 49	Output
		7	NCE49	IOMSTR/MSPI N Chip Select 49	Output
		8			
		9			
		10	I2S0_WS	I2S0 L/R clock	Input
		11	FPIO49	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
A9	50	0	PDM0_CLK	PDM0 Clock output	Output
		1	TRIG0	ADC trigger input	Input
		2	SWTRACECLK	Serial Wire Debug Trace Clock	Output
		3	GPIO50	General purpose I/O	I/O
		4	UART2RTS	UART2 Request to Send (RTS)	Output
		5	UART3RTS	UART3 Request to Send (RTS)	Output
		6	CT50	Timer/counter 50	Output
		7	NCE50	IOMSTR/MSPI N Chip Select 50	Output
		8			
		9	DISP_TE	Display TE input	Input
		10			
		11	FPIO50	Fast PIO	
B9	51	0	PDM0_DATA	PDM0 audio data input to chip	Input
		1	TRIG1	ADC trigger input	Input
		2	SWTRACE0	Serial Wire Debug Trace Output 0	Output
		3	GPIO51	General purpose I/O	I/O
		4	UART0CTS	UART0 Clear to Send (CTS)	Input
		5	UART1CTS	UART1 Clear to Send (CTS) input	Input
		6	CT51	Timer/counter 51	Output
		7	NCE51	IOMSTR/MSPI N Chip Select 51	Output
		8			
		9			
		10			
		11	FPIO51	Fast PIO	
C10	52	0	PDM1_CLK	PDM1 Clock output	Output
		1	TRIG2	ADC trigger input	Input
		2	SWTRACE1	Serial Wire Debug Trace Output 1	Output
		3	GPIO52	General purpose I/O	I/O
		4	UART2CTS	UART2 Clear to Send (CTS) input	Input
		5	UART3CTS	UART3 Clear to Send (CTS) input	Input
		6	CT52	Timer/counter 52	Output
		7	NCE52	IOMSTR/MSPI N Chip Select 52	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO52	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
C9	53	0	PDM1_DATA	PDM1 audio data input to chip	Input
		1	TRIG3	ADC trigger input	Input
		2	SWTRACE2	Serial Wire Debug Trace Output 2	Output
		3	GPIO53	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5	UART1TX	UART1 transmit output	Output
		6	CT53	Timer/counter 53	Output
		7	NCE53	IOMSTR/MSPI N Chip Select 53	Output
		8			
		9			
		10			
		11	FPIO53	Fast PIO	
D10	54	0	PDM2_CLK	PDM2 Clock output	Output
		1	TRIG0	ADC trigger input	Input
		2	SWTRACE3	Serial Wire Debug Trace Output 3	Output
		3	GPIO54	General purpose I/O	I/O
		4	UART2TX	UART2 transmit output	Output
		5	UART3TX	UART3 transmit output	Output
		6	CT54	Timer/counter 54	Output
		7	NCE54	IOMSTR/MSPI N Chip Select 54	Output
		8			
		9			
		10			
		11	FPIO54	Fast PIO	
D9	55	0	PDM2_DATA	PDM2 audio data input to chip	Input
		1	TRIG1	ADC trigger input	Input
		2	SWTRACECTL	Serial Wire Debug Trace Control	Output
		3	GPIO55	General purpose I/O	I/O
		4	UART0RX	UART0 receive input	Input
		5	UART1RX	UART1 receive input	Input
		6	CT55	Timer/counter 55	Output
		7	NCE55	IOMSTR/MSPI N Chip Select 55	Output
		8			
		9			
		10			
		11	FPIO55	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
E10	56	0	PDM3_CLK	PDM3 Clock output	Output
		1	TRIG2	ADC trigger input	Input
		2	SWO	Serial Wire Debug	Output
		3	GPIO56	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	UART3RX	UART3 receive input	Input
		6	CT56	Timer/counter 56	Output
		7	NCE56	IOMSTR/MSPI N Chip Select 56	Output
		8			
		9			
		10			
		11	FPIO56	Fast PIO	
E9	57	0	PDM3_DATA	PDM3 audio data input to chip	Input
		1	TRIG3	ADC trigger input	Input
		2	SWO	Serial Wire Debug	Output
		3	GPIO57	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	UART1RTS	UART1 Request to Send (RTS)	Output
		6	CT57	Timer/counter 57	Output
		7	NCE57	IOMSTR/MSPI N Chip Select 57	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO57	Fast PIO	
J7	58	0			
		1			
		2			
		3	GPIO58	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	UART3RTS	UART3 Request to Send (RTS)	Output
		6	CT58	Timer/counter 58	Output
		7	NCE58	IOMSTR/MSPI N Chip Select 58	Output
		8			
		9			
		10			
		11	FPIO58	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
J8	59	0			
		1	TRIG0	ADC trigger input	Input
		2			
		3	GPIO59	General purpose I/O	I/O
		4	UART0CTS	UART0 Clear to Send (CTS)	Input
		5	UART1CTS	UART1 Clear to Send (CTS) input	Input
		6	CT59	Timer/counter 59	Output
		7	NCE59	IOMSTR/MSPI N Chip Select 59	Output
		8			
		9			
		10			
		11	FPIO59	Fast PIO	
K8	60	0			
		1	TRIG1	ADC trigger input	Input
		2			
		3	GPIO60	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5	UART3CTS	UART3 Clear to Send (CTS) input	Input
		6	CT60	Timer/counter 60	Output
		7	NCE60	IOMSTR/MSPI N Chip Select 60	Output
		8			
		9			
		10			
		11	FPIO60	Fast PIO	
K6	61	0	M6SCL	I^2C Master 6 Clk	Bidirectional Open Drain
		1	M6SCK	SPI Master 6 Clk	Output
		2	I2S1_CLK	I2S1 Bit clock	Input
		3	GPIO61	General purpose I/O	I/O
		4	UART2TX	UART2 transmit output	Output
		5	UART3TX	UART3 transmit output	Output
		6	CT61	Timer/counter 61	Output
		7	NCE61	IOMSTR/MSPI N Chip Select 61	Output
		8			
		9			
		10			
		11	FPIO61	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
L6	62	0	M6SDAWIR3	I^2C Master 6 I/O data (I^2C) 3 Wire data (SPI)	Bidirectional Open Drain
		1	M6MOSI	SPI Master 6 data out	Output
		2	I2S1_DATA	I2S1 Data	Bidirectional
		3	GPIO62	General purpose I/O	I/O
		4	UART0RX	UART0 receive input	Input
		5	UART1RX	UART1 receive input	Input
		6	CT62	Timer/counter 62	Output
		7	NCE62	IOMSTR/MSPI N Chip Select 62	Output
		8			
		9	I2S1_SDOUT	I2S1 Data output	Output
		10			
		11	FPIO62	Fast PIO	
M6	63	0	M6MISO	SPI Master 6 data in	Input
		1	CLKOUT	Oscillator output clock	Output
		2	I2S1_WS	I2S1 L/R clock	Input
		3	GPIO63	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	UART3RX	UART3 receive input	Input
		6	CT63	Timer/counter 63	Output
		7	NCE63	IOMSTR/MSPI N Chip Select 63	Output
		8			
		9	DISP_TE	Display TE input	Input
		10			
		11	FPIO63	Fast PIO	
B6	64	0	MSPI0_0	MSPI Master 0 Interface Signal	I/O
		1	32kHzXT	32kHZ from analog	Output
		2	SWO	Serial Wire Debug	Output
		3	GPIO64	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	DISP_D0	Display Data 0	Output
		6	CT64	Timer/counter 64	Output
		7	NCE64	IOMSTR/MSPI N Chip Select 64	Output
		8			
		9	I2S1_SDIN	I2S1 Data input	Input
		10			
		11	FPIO64	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
A7	65	0	MSPI0_1	MSPI Master 0 Interface Signal	I/O
		1	32kHzXT	32kHZ from analog	Output
		2	SWO	Serial Wire Debug	Output
		3	GPIO65	General purpose I/O	I/O
		4	UART0CTS	UART0 Clear to Send (CTS)	Input
		5	DISP_D1	Display Data 1	Output
		6	CT65	Timer/counter 65	Output
		7	NCE65	IOMSTR/MSPI N Chip Select 65	Output
		8			
		9			
		10			
		11	FPIO65	Fast PIO	
A8	66	0	MSPI0_2	MSPI Master 0 Interface Signal	I/O
		1	CLKOUT	Oscillator output clock	Output
		2	SWO	Serial Wire Debug	Output
		3	GPIO66	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5	DISP_D2	Display Data 2	Output
		6	CT66	Timer/counter 66	Output
		7	NCE66	IOMSTR/MSPI N Chip Select 66	Output
		8			
		9			
		10			
		11	FPIO66	Fast PIO	
C6	67	0	MSPI0_3	MSPI Master 0 Interface Signal	I/O
		1	CLKOUT	Oscillator output clock	Output
		2	SWO	Serial Wire Debug	Output
		3	GPIO67	General purpose I/O	I/O
		4	UART2TX	UART2 transmit output	Output
		5	DISP_D3	Display Data 3	Output
		6	CT67	Timer/counter 67	Output
		7	NCE67	IOMSTR/MSPI N Chip Select 67	Output
		8			
		9			
		10			
		11	FPIO67	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
B7	68	0	MSPI0_4	MSPI Master 0 Interface Signal	I/O
		1	SWO	Serial Wire Debug	Output
		2			
		3	GPIO68	General purpose I/O	I/O
		4	UART0RX	UART0 receive input	Input
		5	DISP_D4	Display Data 4	Output
		6	CT68	Timer/counter 68	Output
		7	NCE68	IOMSTR/MSPI N Chip Select 68	Output
		8			
		9			
		10			
		11	FPIO68	Fast PIO	
B8	69	0	MSPI0_5	MSPI Master 0 Interface Signal	I/O
		1	32kHzXT	32kHZ from analog	Output
		2	SWO	Serial Wire Debug	Output
		3	GPIO69	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	DISP_D5	Display Data 5	Output
		6	CT69	Timer/counter 69	Output
		7	NCE69	IOMSTR/MSPI N Chip Select 69	Output
		8			
		9			
		10			
		11	FPIO69	Fast PIO	
C5	70	0	MSPI0_6	MSPI Master 0 Interface Signal	I/O
		1	32kHzXT	32kHZ from analog	Output
		2	SWTRACE0	Serial Wire Debug Trace Output 0	Output
		3	GPIO70	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	DISP_D6	Display Data 6	Output
		6	CT70	Timer/counter 70	Output
		7	NCE70	IOMSTR/MSPI N Chip Select 70	Output
		8			
		9			
		10			
		11	FPIO70	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
C7	71	0	MSPI0_7	MSPI Master 0 Interface Signal	I/O
		1	CLKOUT	Oscillator output clock	Output
		2	SWTRACE1	Serial Wire Debug Trace Output 1	Output
		3	GPIO71	General purpose I/O	I/O
		4	UART0CTS	UART0 Clear to Send (CTS)	Input
		5	DISP_D7	Display Data 7	Output
		6	CT71	Timer/counter 71	Output
		7	NCE71	IOMSTR/MSPI N Chip Select 71	Output
		8			
		9			
		10			
		11	FPIO71	Fast PIO	
C8	72	0	MSPI0_8	MSPI Master 0 Interface Signal	I/O
		1	CLKOUT	Oscillator output clock	Output
		2	SWTRACE2	Serial Wire Debug Trace Output 2	Output
		3	GPIO72	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5	DISP_D8	Display Data 8	Output
		6	CT72	Timer/counter 72	Output
		7	NCE72	IOMSTR/MSPI N Chip Select 72	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO72	Fast PIO	
D8	73	0	MSPI0_9	MSPI Master 0 Interface Signal	I/O
		1			
		2	SWTRACE3	Serial Wire Debug Trace Output 3	Output
		3	GPIO73	General purpose I/O	I/O
		4	UART2TX	UART2 transmit output	Output
		5	DISP_D9	Display Data 9	Output
		6	CT73	Timer/counter 73	Output
		7	NCE73	IOMSTR/MSPI N Chip Select 73	Output
		8			
		9			
		10			
		11	FPIO73	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G11	74	0	MSPI2_0	MSPI Master 2 Interface Signal	I/O
		1	DISP_QSPI_D0_OUT	Display SPI Data0	Output
		2	DISP_QSPI_D0	Display SPI Data0	Bidirectional
		3	GPIO74	General purpose I/O	I/O
		4	UART0RX	UART0 receive input	Input
		5	DISP_D10	Display Data 10	Output
		6	CT74	Timer/counter 74	Output
		7	NCE74	IOMSTR/MSPI N Chip Select 74	Output
		8			
		9	DISP_SPI_SD	Display SPI Data In/Out	Bidirectional
		10	DISP_SPI_SDO	Display SPI Data Out	Output
		11	FPIO74	Fast PIO	
H11	75	0	MSPI2_1	MSPI Master 2 Interface Signal	I/O
		1	32kHzXT	32kHZ from analog	Output
		2	DISP_QSPI_D1	Display SPI Data1	Output
		3	GPIO75	General purpose I/O	I/O
		4	UART2RX	UART2 receive input	Input
		5	DISP_D11	Display Data 11	Output
		6	CT75	Timer/counter 75	Output
		7	NCE75	IOMSTR/MSPI N Chip Select 75	Output
		8			
		9	DISP_SPI_DCX	Display SPI DCx	Output
		10			
		11	FPIO75	Fast PIO	
J11	76	0	MSPI2_2	MSPI Master 2 Interface Signal	I/O
		1	32kHzXT	32kHZ from analog	Output
		2	DISP_QSPI_D2	Display SPI Data2	Output
		3	GPIO76	General purpose I/O	I/O
		4	UART0RTS	UART0 Request to Send (RTS)	Output
		5	DISP_D12	Display Data 12	Output
		6	CT76	Timer/counter 76	Output
		7	NCE76	IOMSTR/MSPI N Chip Select 76	Output
		8			
		9			
		10			
		11	FPIO76	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
K11	77	0	MSPI2_3	MSPI Master 2 Interface Signal	I/O
		1			
		2	DISP_QSPI_D3	Display SPI Data3	Output
		3	GPIO77	General purpose I/O	I/O
		4	UART0CTS	UART0 Clear to Send (CTS)	Input
		5	DISP_D13	Display Data 13	Output
		6	CT77	Timer/counter 77	Output
		7	NCE77	IOMSTR/MSPI N Chip Select 77	Output
		8			
		9			
		10			
		11	FPIO77	Fast PIO	
L11	78	0	MSPI2_4	MSPI Master 2 Interface Signal	I/O
		1			
		2	DISP_QSPI_SCK	Display SPI CLK	Output
		3	GPIO78	General purpose I/O	I/O
		4	UART0TX	UART0 transmit output	Output
		5	DISP_D14	Display Data 14	Output
		6	CT78	Timer/counter 78	Output
		7	NCE78	IOMSTR/MSPI N Chip Select 78	Output
		8			
		9	DISP_SPI_SCK	Display SPI Clock	Output
		10			
		11	FPIO78	Fast PIO	
M11	79	0	MSPI2_5	MSPI Master 2 Interface Signal	I/O
		1			
		2	SDIF_DAT4	SD/SDIO/MMC Data4 pin	I/O
		3	GPIO79	General purpose I/O	I/O
		4	SWO	Serial Wire Debug	Output
		5	DISP_VS	Display RGB VSYNC	Output
		6	CT79	Timer/counter 79	Output
		7	NCE79	IOMSTR/MSPI N Chip Select 79	Output
		8			
		9	DISP_SPI_SDI	Display SPI Data IN	input
		10			
		11	FPIO79	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
K10	80	0	MSPI2_6	MSPI Master 2 Interface Signal	I/O
		1	CLKOUT	Oscillator output clock	Output
		2	SDIF_DAT5	SD/SDIO/MMC Data5 pin	I/O
		3	GPIO80	General purpose I/O	I/O
		4	SWTRACE0	Serial Wire Debug Trace Output 0	Output
		5	DISP_HS	Display RGB HSYNC	Output
		6	CT80	Timer/counter 80	Output
		7	NCE80	IOMSTR/MSPI N Chip Select 80	Output
		8			
		9			
		10			
		11	FPIO80	Fast PIO	
L10	81	0	MSPI2_7	MSPI Master 2 Interface Signal	I/O
		1	CLKOUT	Oscillator output clock	Output
		2	SDIF_DAT6	SD/SDIO/MMC Data6 pin	I/O
		3	GPIO81	General purpose I/O	I/O
		4	SWTRACE1	Serial Wire Debug Trace Output 1	Output
		5	DISP_DE	Display RGB Data Enable	Output
		6	CT81	Timer/counter 81	Output
		7	NCE81	IOMSTR/MSPI N Chip Select 81	Output
		8			
		9			
		10			
		11	FPIO81	Fast PIO	
L9	82	0	MSPI2_8	MSPI Master 2 Interface Signal	I/O
		1	32kHzXT	32kHZ from analog	Output
		2	SDIF_DAT7	SD/SDIO/MMC Data7 pin	I/O
		3	GPIO82	General purpose I/O	I/O
		4	SWTRACE2	Serial Wire Debug Trace Output 2	Output
		5	DISP_PCLK	Display RGB Pixel Clock	Output
		6	CT82	Timer/counter 82	Output
		7	NCE82	IOMSTR/MSPI N Chip Select 82	Output
		8			
		9			
		10			
		11	FPIO82	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
K9	83	0	MSPI2_9	MSPI Master 2 Interface Signal	I/O
		1	32kHzXT	32kHZ from analog	Output
		2	SDIF_CMD	SD1/SD4/MMC Command pin	I/O
		3	GPIO83	General purpose I/O	I/O
		4	SWTRACE3	Serial Wire Debug Trace Output 3	Output
		5	DISP_SD	Display RGB Shutdown	Output
		6	CT83	Timer/counter 83	Output
		7	NCE83	IOMSTR/MSPI N Chip Select 83	Output
		8			
		9			
		10			
		11	FPIO83	Fast PIO	
J9	84	0			
		1			
		2	SDIF_DAT0	SD/SDIO/MMC Data0 pin	I/O
		3	GPIO84	General purpose I/O	I/O
		4			
		5			
		6	CT84	Timer/counter 84	Output
		7	NCE84	IOMSTR/MSPI N Chip Select 84	Output
		8			
		9			
		10			
		11	FPIO84	Fast PIO	
H9	85	0			
		1			
		2	SDIF_DAT1	SD/SDIO/MMC Data1 pin	I/O
		3	GPIO85	General purpose I/O	I/O
		4			
		5			
		6	CT85	Timer/counter 85	Output
		7	NCE85	IOMSTR/MSPI N Chip Select 85	Output
		8			
		9			
		10			
		11	FPIO85	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
G10	86	0			
		1			
		2	SDIF_DAT2	SD/SDIO/MMC Data2 pin	I/O
		3	GPIO86	General purpose I/O	I/O
		4			
		5			
		6	CT86	Timer/counter 86	Output
		7	NCE86	IOMSTR/MSPI N Chip Select 86	Output
		8			
		9			
		10			
		11	FPIO86	Fast PIO	
G9	87	0			
		1			
		2	SDIF_DAT3	SD/SDIO/MMC Data3 pin	I/O
		3	GPIO87	General purpose I/O	I/O
		4			
		5			
		6	CT87	Timer/counter 87	Output
		7	NCE87	IOMSTR/MSPI N Chip Select 87	Output
		8			
		9	DISP_TE	Display TE input	Input
		10			
		11	FPIO87	Fast PIO	
F9	88	0			
		1			
		2	SDIF_CLKOUT	SD/SDIO/MMC Clock to Card (CLK)	Output
		3	GPIO88	General purpose I/O	I/O
		4			
		5			
		6	CT88	Timer/counter 88	Output
		7	NCE88	IOMSTR/MSPI N Chip Select 88	Output
		8			
		9			
		10			
		11	FPIO88	Fast PIO	

Table 1: Pin List and Function Table

Blue BGA Pin	GPIO Pad Number	Function Select Number	Pad Function Name	Description	Pin Type
F10	89	0			
		1			
		2			
		3	GPIO89	General purpose I/O	I/O
		4			
		5	DISP_CM	Display RGB Color Mode	Output
		6	CT89	Timer/counter 89	Output
		7	NCE89	IOMSTR/MSPI N Chip Select 89	Output
		8			
		9			
		10			
		11	FPIO89	Fast PIO	
J2	90	0			
		1			
		2			
		3	GPIO90	General purpose I/O	I/O
		4			
		5			
		6	CT90	Timer/counter 90	Output
		7	NCE90	IOMSTR/MSPI N Chip Select 90	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO90	Fast PIO	
K2	91	0			
		1			
		2			
		3	GPIO91	General purpose I/O	I/O
		4			
		5			
		6	CT91	Timer/counter 91	Output
		7	NCE91	IOMSTR/MSPI N Chip Select 91	Output
		8			
		9	VCMPO	Output of the voltage comparator signal	
		10			
		11	FPIO91	Fast PIO	

Blank Page

2. SoC Product Introduction

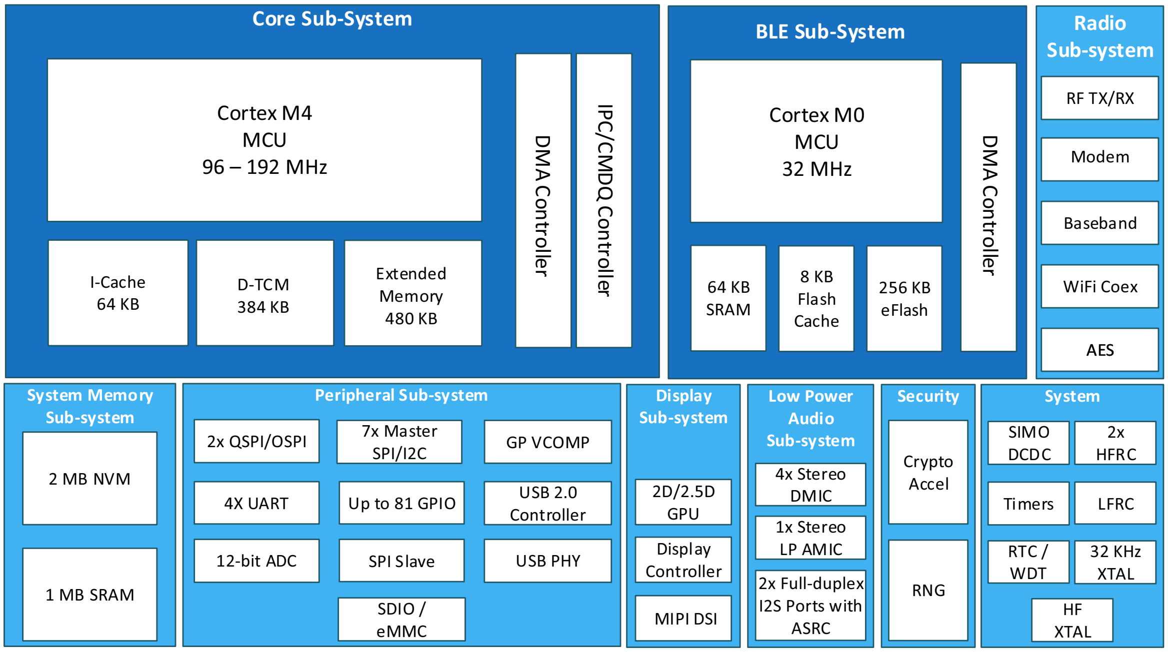


Figure 2. Apollo4 Blue SoC Block Diagram

The Apollo4 Blue SoC is an ultra-low power, highly integrated mixed-signal SoC designed for battery-powered devices. The SoC provides a significant enhancement in processing capability and highly integrated power management and audio capabilities to the Apollo SoC product family. The Apollo4 Blue SoC brings the powerful Arm Cortex-M4 processor with Floating Point Unit coupled with the world's lowest power audio and communications processing. The Apollo4 Blue SoC takes Ambiq's patented Subthreshold Power Optimized Technology (SPOT) Platform to a whole new level of compute power efficiency, setting new industry benchmarks in low power design and high efficiency portable computing.

2.1 Features

Ultra-low supply current:

- 5 µA/MHz active mode current
- Low-power sleep and deep sleep modes with selectable levels of RAM/cache retention

High-performance Arm Cortex-M4F processor:

- 96/192 MHz operating modes
- Floating Point Unit
- Memory Protection Unit
- Wake-up interrupt controller with 32 interrupts
- Secure boot

Bluetooth 5.1 Low Energy

- Data rate: 1 Mbps and 2 Mbps
- Extended advertising packets
- Angle of Arrival (AoA) and Angle of Departure (AoD)
- Tx output power: -10 dBm to +6 dBm
- Rx sensitivity: -95.5 dBm at 1 Mbps, -92 dBm at 2 Mbps

Ultra-low power Memory:

- Up to 1.8 MB of low leakage / low power RAM for code/data
- 64 kB 2-way Associative/Direct-Mapped Cache per core
- 384 kB (Arm M4) Tightly Coupled RAM
- 480 kB Extended RAM
- Up to 2 MB of non-volatile memory (NVM) for code/data

Ultra-low power interface for off-chip sensors:

- 8-bit, 10-bit and 12-bit ADC modes
- Up to 2.8 MS/s sampling rate (8-bit mode)
- 11 selectable input channels available
- Voltage Comparator
- Temperature sensor with ±3ºC accuracy

Flexible serial peripherals:

- 2x 2/4/8-bit SPI master interface
- 7x I2C/SPI masters for peripheral communication
- I^2C / SPI slave for host communications
- 4x UART modules with 32-location TX and RX FIFOs
- USB 2.0 HS/FS device controller
- SDIO (SD3.0) / eMMC (v4.51)

Display:

- LCD Controller
- MIPI DSI 1.2 with single data lane up to 500 Mbps
- Up to 500 x 500 resolution
- 4 layers with full alpha blending
- Frame buffer decompression

Graphics:

- 2D/2.5D graphics accelerator
- Rasterizer
- Full Alpha Blending
- Texture Mapping
- Texture and Frame Buffer Compression

Audio processing:

- Stereo low-power analog microphones
- 4x stereo digital microphones
- 2x full duplex I^2S ports with ASRC
- Digital filtering
- Ultra low power voice and keyword detect

Rich set of clock sources:

- 32.768 kHz XTAL oscillator
- 32 MHz XTAL oscillator
- Low frequency RC oscillator – 1.024 kHz
- 2x high frequency RC oscillators – 192/384 MHz
- RTC based on Ambiq's AM08X5/18X5 families

Power Management:

- Wide operating range: 1.71 - 2.2 V
- SIMO DC-DC buck converter
- Dedicated DC-DC buck converter for BLE radio
- Multiple I/O voltages supported

Operating Temperature Range:

- -20°C to 60°C

2.2 Functional Overview

The ultra-low power Apollo4 Blue SoC, shown in Figure 2, is an ideal solution for battery-powered applications supporting mid-tier to high-end wearables and IoT products. In a typical system, the device serves as an applications processor with a fully integrated audio subsystem and interface to BT/Bluetooth Low Energy 5/Wi-Fi radios. The SoC includes an extensive set of digital and analog peripheral interfaces with integrated ADCs and digital sensor processing using the integrated serial master ports. The Cortex-M4 core with Floating Point Unit (referred to throughout this document as “M4”, “M4 Core” or “Cortex-M4”) integrated in the Apollo4 Blue SoC is capable of running complex data analysis, sensor fusion algorithms to process the sensor data and orchestrate complex audio processing signal flows. The Cortex-M4 core leverages the broad development and support ecosystem to accelerate time-to-market for application and product deployment.

In other configurations, a host processor can communicate with the Apollo4 Blue SoC over its serial slave port using the SPI or I^2C protocol. With unprecedented energy efficiency for sensor conversion, audio processing and data analysis, the SoC enables months and years of battery life for products only achieving days or months of battery life today. Similarly, the SoC enables the use of significantly complex algorithmic processing due to its industry leading low active mode power. By using the Apollo4 Blue SoC, uncompromised user experience with truly always on sensor and audio processing is brought to life.

The Apollo4 Blue SoC provides support for various operating modes to maximize energy efficiency depending on the workload demand. For extremely power sensitive workloads, the SoC supports low power operating modes leveraging Ambiq's patented SPOT technology to achieve industry leading energy efficiency. For timing critical or higher MIPS workloads, the SoC supports high performance operating modes through Ambiq's TurboSPOT™ technology. The TurboSPOT technology enables high performance while still maintaining extremely high energy efficiency operation. The SoC also supports secure boot using Ambiq's SecureSPOT™ technology enabling applications to establish and maintain a root of trust from boot to execution.

A rich set of sensor peripherals enable the monitoring of several sensors. An integrated temperature sensor enables the measurement of ambient temperature. A scalable ultra-low power Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) monitors the temperature sensor, several internal voltages, and up to eight external sensor signals. The General Purpose ADC is uniquely tuned for minimum power with a configurable measurement mode that does not require MCU intervention. In addition to integrated analog sensor peripherals, I^2C/SPI master ports and/or UART ports enables the SoC to communicate with external sensors that have digital outputs.

The Apollo4 Blue SoC integrates an audio subsystem supporting four stereo PDM microphones, a pair of stereo Low Power Analog microphones, two I^2S master/slave ports and ASRC support.

For higher bandwidth peripherals, the SoC supports two Multi-bit SPI (MSPI) controllers for 1-bit, 2-bit, 4-bit (QuadSPI) and 8-bit (OctalSPI) data.

The SoC also includes a set of timing peripherals and an RTC which is based on Ambiq's AM08XX and AM18XX Real-Time Clock (RTC) families. The general purpose Timer/Counter Module (TIMER), 32-bit System Timer (STIMER), and the RTC may be driven independently by one of three different clock sources: a low frequency RC oscillator, a high frequency RC oscillator, a high frequency crystal (XTAL) oscillator and a 32.768 kHz crystal (XTAL) oscillator. These clock sources use the proprietary advanced calibration techniques developed for the AM08XX and AM18XX products that achieve XTAL-like accuracy with RC-like power.

Additionally, the Apollo4 Blue SoC includes clock reliability functions first offered in the AM08XX and AM18XX products. For example, the RTC can automatically switch from an XTAL source to an RC source in the event of an XTAL failure. the SoC supports highly optimized PWM pattern generation for complex, efficient stepper motor control operation. Up to 8 independent motors can be controlled from the SoC supporting several different operating modes.

To facilitate development and debug, the Apollo4 Blue SoC is supported by a complete suite of standard software development tools. Ambiq provides drivers for all peripherals along with basic application code to shorten development time. The debug functions are accessible via Serial Wire Debugger (SWD).

3. MCU Core

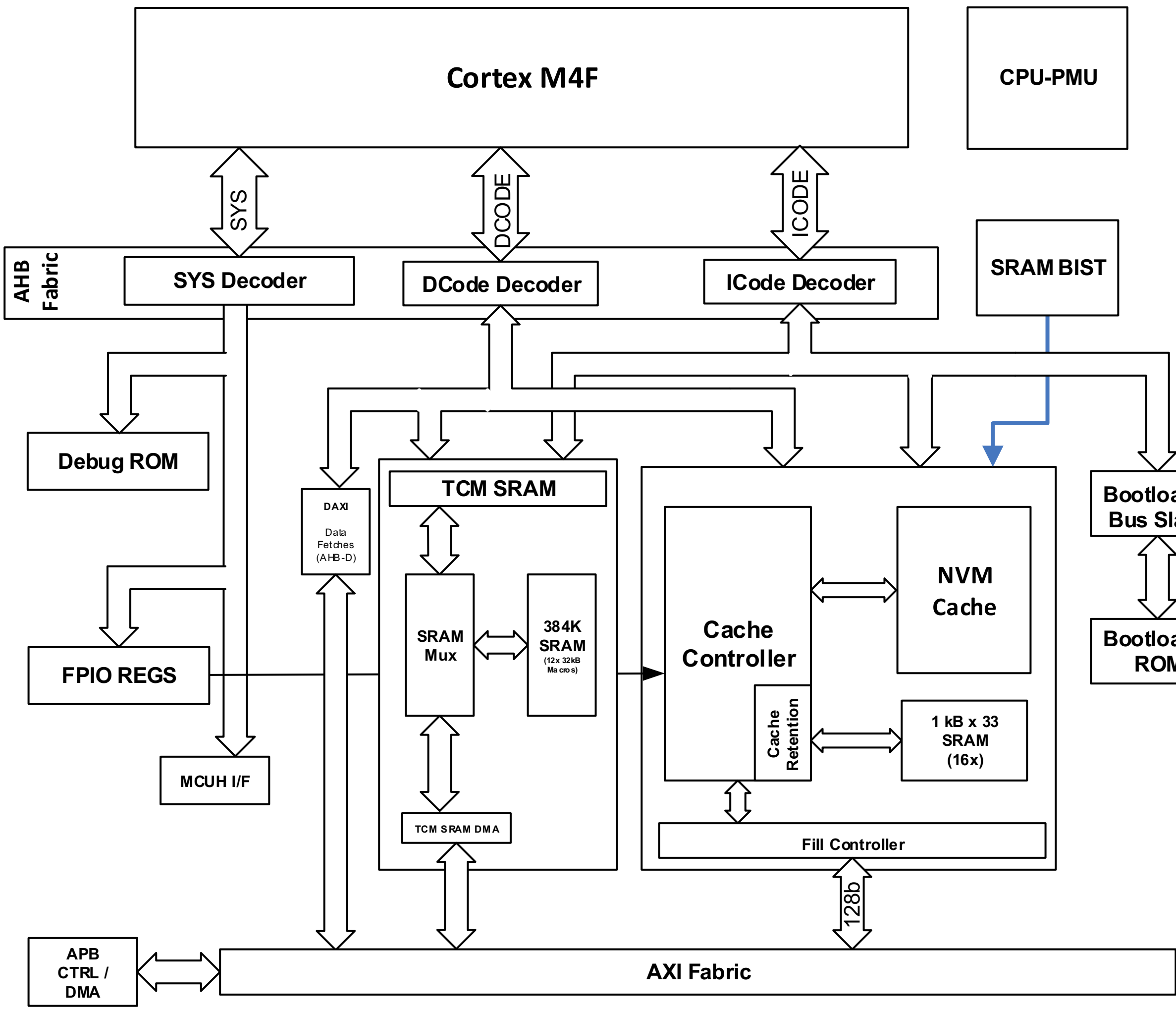


Figure 3. Apollo4 Blue SoC Core Block Diagram

Please refer to the MCUCTRL, PWRCTRL, CPU, ITM and other registers applicable to this chapter in the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

3.1 Functional Overview

As can be seen in Figure 3, at the center of the Apollo4 Blue SoC is a 32-bit Arm Cortex-M4 (CM4) core with the floating point option. This 3-stage pipeline implementation of the Arm v7-M architecture offers highly efficient processing in a very low power design. The Arm M DAP enables debugging access via a Serial Wire Interface from outside of the SoC which allows access to all of the memory and peripheral devices of the SoC.

The M4 core offers some other advantages including:

- Single 4 GB memory architecture with all Peripherals being memory-mapped
- Low-Power Consumption Modes:
  - Active

  - Sleep
  - Deep-Sleep
  - Power-Off
- Interrupts and Events
  - NVIC – interrupt controller
  - WIC – Wake-Up Interrupt Controller
  - Sleep-on-Exit (reduces interrupt overhead, used in an ISR SW structure)
  - WFI (enter sleep modes, wait for interrupts)

The Cortex-M4 processor supports the ARMv7 Protected Memory System Architecture (PMSA) that provides programmable support for memory protection using a number of software controllable regions. Memory regions can be programmed to generate faults when accessed inappropriately by unprivileged software reducing the scope of incorrectly written application code. The architecture includes fault status registers to allow an exception handler to determine the source of the fault and to apply corrective action or notify the system.

Reference the “Arm Cortex-M4 Processor Technical Reference Manual” for more details.

The following sections provide behavioral and performance details about each of the peripherals controlled by the MCU core. Where multiple instances of a peripheral exist on Apollo4 Blue SoC (e.g., the seven I^{2}C/SPI master modules), base memory addresses for the registers are provided for each and noted as INSTANCE 0, INSTANCE 1, etc.

3.2 CPU Subsystem

The CPU subsystem (or CPU complex) is composed of an Arm Cortex-M4 CPU, NVM cache, tightly coupled data memory, interrupt and debug logic as well as the associated power management control for the subsystem. The subsystem has the following features:

- Cortex-M4 with Floating Point Unit
- ARMv7 ISA
- Operating Modes
  - 96 MHz Low Power Mode
  - 192 MHz TurboSPOT Enhanced Mode
- WIC supported
- MPU: 8 regions
- Debug
  - Embedded Trace Macrocell (ETM) supported
  - 4x data watchpoint comparators and 8x breakpoint comparators
  - ITM/DWT supported
  - Multi-core break support
- CPU Power Management block
- 64 kB NVM Cache
- 384 kB Data TCM

The CPU complex has 64 kB of Non-Volatile Memory (NVM) caching (instruction and data accesses issued to the NVM space) as well as 384 kB of local data Tightly Coupled Memory (TCM). In addition, the CPU has access to 1024 kB of shared system SRAM, 2 MB of internal NVM and up to 480 kB of extended SRAM. All of the memory is memory mapped and accessible to the CPU. All of the memory accesses are qualified based on the memory protection attributes (enforced within the M4) and the system memory protection attributes (enforced within the system memory controllers).

3.3 Interrupts

Within the SoC, multiple peripherals can generate interrupts. In some cases, a single peripheral may be able to generate multiple different interrupts. Each interrupt signal generated by a peripheral is connected back to the M4 core in two places. First, the interrupts are connected to the Nested Vectored Interrupt Controller, NVIC, in the core. This connection provides the standard changes to program flow associated with interrupt processing. Additionally, they are connected to the WIC outside of the core, allowing the interrupt sources to wake the M4 core when it is in a deep sleep (SRPG) mode.

For details on the Interrupt model of the M4, please see the “**Cortex-M4 Devices Generic User Guide,**” document number DUI0553A. Note that the M4 NMI type interrupts are not supported.

The Cortex-M4 allows the user to assign various interrupts to different priority levels based on the requirements of the application. In this SoC implementation, 8 different priority levels are available.

One additional feature of the M4 interrupt architecture is the ability to relocate the Vector Table to a different address. This could be useful if the application requires a different set of interrupt service routines for a particular mode of an application. The software could move the Vector Table into SRAM and reassign the interrupt service routine entry addresses as needed.

3.4 Memory Map

Arm has a well-defined memory map for devices based on the Arm v7-M Architecture. The M4 further refines this map in the area of the Peripheral and System address ranges. Below is the system memory map as defined by Arm:

Table 2: Arm Cortex-M4 Memory Map

Address	Name	Executable	Description
0x00000000 – 0x1FFFFFFF	Code	Y	Internal NVM (NVM)
0x20000000 – 0x3FFFFFFF	Reserved	N	Reserved
0x40000000 – 0x5FFFFFFF	Peripheral	N	On-chip peripheral address space
0x60000000 – 0x9FFFFFFF	External RAM	Y	External / Off-chip Memory
0xA0000000 – 0xDFFFFFFF	External Device	N	External device memory
0xE0000000 – 0xE00FFFFF	Private Peripheral Bus	N	NVIC, System timers, System Control Block Reserved for system control and debug. Cannot be used for exception vector tables. Data accesses are either performed internally or on EPPB. Accesses in the range: 0xE0000000-0xE0043FFF Are handled within the processor. 0xE0044000-0xE00FFFFF Appear as APB transactions on the EPPB interface of the processor. Any attempt to execute instructions from the region results in a MemManage fault.
0xE0100000 – 0xFFFFFFFF	Vendor_SYS	N	Vendor Defined Partly reserved for future processor feature expansion. Any attempt to execute instructions from the region results in a MemManage fault. Data accesses are performed on S-AHB

3.5 Memory Protection Unit (MPU)

The Apollo4 Blue SoC includes an MPU which is a core component for memory protection. The M4 processor supports the standard ARMv7 *Protected Memory System Architecture* model. The MPU provides full support for:

- Protection regions.
- Access permissions
- Exporting memory attributes to the system.

MPU mismatches and permission violations invoke the MemManagehandler. See the Arm®v7-M Architecture Reference Manual for more information.

You can use the MPU to:

- Enforce privilege rules.
- Separate processes.
- Enforce access rules.

The Apollo4 Blue SoC supports up to 8 memory regions.

3.6 System Buses

The Arm Cortex-M4 utilizes 3 instances of the AMBA AHB bus for communication with memory and peripherals. The ICode bus is designed for instruction fetches from the ‘Code’ memory space while the DCode bus is designed for data and debug accesses in that same region. The System bus is designed for fetches to the SRAM and other peripheral devices of the SoC.

The Apollo4 Blue SoC maps the available SRAM memory onto an address space within the ‘Code’ memory space. This gives the user the opportunity to perform instruction and data fetches from the lower-power SRAM to effectively lower the power consumption of the SoC.

The peripherals of the Apollo4 Blue SoC which are infrequently accessed are located on an AMBA APB bus. A bridge exists which translates the accesses from the System AHB to the APB. Accesses to these peripherals will inject a single wait-state on the AHB during any access cycle.

More information about system buses can be found in “Memory Subsystem” on page 61.

3.7 Power Management

The power management is partitioned into several components across Apollo4 Blue SoC. For the CPU complex, a dedicated finite-state machine controls the transitions of the CPU between power modes. When moving from Active Mode to Deep Sleep Mode, the CPU-PMU manages the state-retention capability of the registers within the Cortex-M4 core and also handshakes with the central power management controller to appropriately handle the voltage rails to the CPU complex. Once in the Deep Sleep Mode, the CPU-PMU, in conjunction with the Wake-Up Interrupt Controller, waits for a wakeup event. When the event is observed, the CPU-PMU begins the power restoration process by handshaking with the central power management controller to adjust the voltage rails to the CPU complex and initiate the restoration of the CPU register state. The M4 is then returned to active mode once the state is ready.

3.7.1 Cortex-M4 Power Modes - Overview

The Arm Cortex-M4 supports the following 4 power modes:

- High Performance Active (not a differentiated power mode for the Cortex-M4)
- Active
- Sleep
- Deep Sleep

In addition to these Arm-defined modes, the Apollo4 Blue SoC supports system level power modes which are defined in subsequent sections.

3.7.1.1 High Performance Mode

The Apollo4 Blue SoC supports the Ambiq TurboSPOT^{TM} which enables a higher frequency, high performance operating mode (HP Mode). In this mode, the M4 and all closely coupled memory run at an elevated frequency. All of the non-debug Arm clocks (FCLK, HCLK) also operate at the elevated frequency level. All peripherals are maintained at the nominal frequency level during burst. This mode is entered and exited under software direction but transitions are completely handled in hardware.

This is not a standard Arm-defined power mode. From the Arm core, this mode is treated similarly to “Active Mode”.

3.7.1.2 Active Mode

In the Active Mode, the M4 core is powered up, clocks are active, and instructions are being executed. In this mode, the M4 expects all (enabled) devices attached to the interfaces to be powered and clocked for normal access. All of the non-debug Arm clocks (FCLK, HCLK) are active in this state.

To transition from the Active Mode to any of the lower-power modes, a specific sequence of instructions is executed on the M4 core. First, specific bits in the ARMv7-M System Control Register must be set to determine the mode to enter. See the ARMv7-M Architecture Reference Manual for more details.

After the SCR is setup, code can enter the low-power states using one of the 3 following methods:

- Execute a Wait-For-Interrupt (WFI) instruction.
- Execute a Wait-For-Event (WFE) instruction.
- Set the SLEEPONEXIT bit of the SCR such that the exit from an ISR will automatically return to a sleep state.

The M4 will enter a low-power mode after one of these are performed (assuming all conditions are met) and remain there until some event causes the core to return to Active Mode. The possible reasons to return to Active Mode are:

- A reset
- An enabled Interrupt is received by the NVIC
- An event is received by the NVIC

- A Debug Event is received from the DAP

> NOTE
>
> Various aspects of the Arm Cortex-M4 core, including operation and interrupt handling of the system counter, SysTick, are SoC specific. For Apollo4 family SoCs during Deep Sleep Mode, SysTick is completely powered down, and in normal Sleep Mode the clocks are gated off. In either case, the SysTick counter is not running and therefore cannot be used to wake the CPU.
>
> When the CPU enters a sleep mode, the CPU clock is gated off. Normally there is a free-run clock to keep the Wake-Up Interrupt Controller (WIC) and/or the Nested Vectored Interrupt Controller (NVIC) running. But for power saving purposes, this “free-running” clock is gated off in both sleep modes, and can be resumed when external interrupts are received.
>
> In the case of SysTick, its interrupt is generated by the SysTick counter, which is clocked by this “free-running” clock. Since this clock is gated off, the counter stops and cannot generate an interrupt.
>
> However, when a debugger is attached, the clock is running and the SysTick interrupt can be generated.

3.7.1.3 Sleep Mode

In the Sleep Mode, the M4 is powered up, but the clocks (HCLK, FCLK) are gated. The power supply is still applied to the M4 logic such that it can immediately become active on a wakeup event and begin executing instructions.

3.7.1.4 Deep Sleep Mode

In the Deep Sleep Mode, the M4 enters SRPG mode where the main power is removed, but the flops retain their state. The clocks are not active, and the SoC clock sources for HCLK and FCLK can be deactivated. To facilitate the removal of the source supply and entry into SRPG mode, the M4 will handshake with the Wake-up Interrupt Controller and Power Management Unit and set up the possible wakeup conditions.

3.7.2 CPU Power Management

Power Management on the Apollo4 Blue SoC is handled through a combination of hardware and software. The hardware handles the interfacing and control sequencing between the regulators and the individual power domains within the SoC. The software initiates transitions through power states by processor instructions and system-level power control commands.

The Power Management system is composed of a central power management controller and various power management units (PMU) for each primary subsystem/domain. The primary PMUs are listed below:

- CPU-PMU: responsible for power sequencing for the CPU subsystem
- IO-PMU: responsible for power sequencing at each I/O subsystem

3.7.2.1 CPU-PMU

When moving from Active Mode to Deep Sleep Mode, the CPU-PMU manages the state-retention capability of the registers within the Cortex-M4 core and also controls the interface to the voltage regulators as needed to support the various operating modes of the CPU. Once in the Deep Sleep Mode, the CPU-PMU, in conjunction with the Wake-Up Interrupt Controller, waits for a wakeup event. When the event is observed, the CPU-PMU begins the power restoration process by re-enabling the on-chip voltage regulators and restoring the CPU register state. The M4 is then returned to active mode once all state is ready.

The CPU-PMU enables support for the following Arm Cortex-M4 defined power modes:

- OFF
- Deep Sleep
- Sleep
- Active
  - Low Power / High Efficiency: 96 MHz
  - High Performance: 192 MHz

3.7.2.2 IO-PMU

The IO-PMU's manage power state for I/O subsystems. Each I/O subsystem supports the following power modes. Note that each I/O subsystem may have a different implementation that defines each specific power state. This is implementation-specific to each I/O controller. Also, not all power modes may be supported by each IO-PMU (typical configuration may support only OFF and Active LP).

- OFF
- Sleep (device is enabled but no active transactions)
- Active

3.7.2.3 Power Management Controller

The power management controller provides control functions for each supply regulator as well as the primary power gates under digital logic control. The power management controller (PWRCTL) receives input from all PMUs indicating requested power levels and also controls from software (via power management control registers). A power management mapping configuration is also provided (sourced from INFO1 shadowed to PWRCTL) which dictates the functional operation at the regulator interface based on the input power requests. This mapping configuration allows the power management functionality to be programmatic, enabling characterization, tuning and/or bug fixes.

Following are the supply regulator interfaces:

- SIMO Buck
- Core LDO
- Mem LDO

PWRCTL is also responsible for controlling power gate enables for all digital power domains. The power gate enables are controlled based on the power level requests. When an “OFF” level is requested from the respective requester PMU or a software override is asserted to force a requester “OFF” or, for I/O requesters, when the corresponding I/O device enable is de-asserted, the respective power gate enable is asserted to power off the domain. For all other power level requests, the power gate is disabled powering up the respective domain.

SRAM and NVM power domains are controlled based on the dependent requester domains. For NVM, if all CPU PMU requesters are “OFF” and DMA requesters are “OFF” or “Sleep”, the NVM power domain is powered OFF. For SRAM, each SRAM is powered OFF either based on the SKU memory configuration or if all CPU PMU requesters are “OFF” and DMA requesters are “OFF” or “Sleep” and the SRAM is enabled to power off based on the power control MEMPWDINSLEEP configuration.

3.7.2.4 System Power States

At the SoC level, various power states are supported to enable key workloads and ensure maximum power efficiency. System power states are defined in the sub-sections below.

If any memory is retained through one of the deep sleep modes described below, a portion of the 384 kB of local data Tightly Coupled Memory (TCM) is assumed to be used for any necessary memory retention. In addition to retaining a portion of the TCM, half or all of the shared SRAM can be retained as well (via the PWRCTRL_SSRAMRETCFG_SSRAMPWDSLP field). Power draw is increased with increased TCM or SRAM retained in these low power modes.

The amount of TCM memory retained in deep sleep is user settable by selecting from several options in the PWRCTRL_MEMRETCFG_DTCMPWDSLP field. Selectable TCM memory sizes to be retained include none, 8, 120, 128, 256, 376 and all 384 kB.

SRAM cache memory is powered down in deep sleep mode by setting the PWRCTRL_MEMRETCFG_CACHEPWDSLP bit. This would *not* be done if the SRAM bank is used as the target for DMA transfer while the CPU in deep sleep.

NVM is powered down by setting the PWRCTRL_MEMRETCFG_NVM0PWDSLP bit.

3.7.2.4.1 SYS Active (S_{ACT})

The CPU is in one of the Active Modes and executing instructions. All respective memory and I/O devices are powered ON and available as needed.

3.7.2.4.2 SYS Sleep Mode 0 (S_{S0})

SYS Sleep Mode 0 can be entered if all processor cores are in sleep mode or deeper sleep state. In SYS Sleep Mode 0, this is a low power state for the SoC. In this mode, all *enabled* TCM, and SRAM and extended memory is retained (up to 1.8 MB), NVM is in standby, HFRC is on, main core clock domain(s) is gated but peripheral clock domains can be on.

This state can be entered if a peripheral device (such as SPI, UART, I^2C or MSPI) is actively transferring data and the time window is sufficient for the processor(s) to enter Sleep Mode but is not long enough to go into a Deep Sleep Mode.

3.7.2.4.3 SYS Sleep Mode 1 (S_{S1})

SYS Sleep Mode 1 can be entered if all processor cores are in sleep mode or deeper sleep state and all peripheral devices are idle. In this mode, all *enabled* TCM, and SRAM and extended memory is retained (up to 1.8 MB), NVM is in standby, HFRC is on, all functional clocks are gated.

This state can be entered if a no peripheral device (such as SPI, UART, I^2C, or MSPI) is actively transferring data, however, communication may occur within a short time window which will prevent the processor(s) from entering Deep Sleep Mode (and subsequently the system from entering a lower power state).

This state is also referred to as “Active Idle”. In other words, all power domains can be powered on, but all clocks are gated. This state is a good power baseline for the system as it represents the active mode DC power level. Typically, the power in this state is dominated by leakage and always-on functional blocks.

3.7.2.4.4 SYS Deep Sleep Mode 0 (S_{DS0})

In SYS Deep Sleep Mode 0, this is a deep low power state for the SoC. In this mode, all processors are in Deep Sleep mode or are powered OFF. All SRAM is in retention (capacity controlled by software), cache memory is in retention, NVM is in power down, HFRC is on, main processor power domains are off but peripheral power domains can be on.

This state can be entered if a peripheral device (such as SPI, UART, I^2C or MSPI) is actively (or intermittently) transferring data but the window of acquisition is long enough to allow the processor to go into a deeper low power state.

3.7.2.4.5 SYS Deep Sleep Mode 1 (S_{DS1})

In SYS Deep Sleep Mode 1, this is a deep low power state for the SoC. In this mode, all processors are in Deep Sleep mode or are powered OFF. All SRAM is in retention (capacity controlled by software), cache memory is powered OFF, NVM is in power down, HFRC is on, main processor power domains are off but peripheral power domains can be on.

This state can be entered if the latency to warm up the cache can be tolerated. This could be an extended wait for peripheral communication event.

3.7.2.4.6 SYS Deep Sleep Mode 2 (S_{DS2})

In SYS Deep Sleep Mode 2, this is the minimum power state that the processor(s) can resume normal operation. In this mode, minimal SRAM memory is retained as needed for software to resume (note that SRAM can have configurable amount of instances in retention depending on the software/system functional and latency requirements), Cache is powered off (no retention), NVM is in power down, HFRC is off, XTAL is ON, all internal switched power domains are off/gated. Processors are in Deep Sleep or OFF. processor logic state is retained.

This state can be entered when all activity has suspended for a duration of time sufficient to sustain the longer exit latencies to resume. This could be a state where periodic data samples are taken and the data is locally processed/accumulated/transferred at long time intervals. This state can only be entered (vs S_{DS1}) if the peripheral devices are either not enabled/active or if the application can afford to save/restore the state of the controller(s) on entry/exit of this mode.

3.7.2.4.7 SYS Deep Sleep Mode 3 (S_{DS3})

In SYS Deep Sleep Mode 3, this is a deep sleep power state for the SoC. In this mode, no memory is in retention, all memory is powered down, LFRC is on (HFRC and XTAL are off), all internal switched power domains are off/gated. Processors are in deep sleep or OFF. Processor logic state is retained. Single timer is running. This state can be entered on long inactivity periods. Also can be used for very low power ADC sampling without CPU interaction.

3.7.2.4.8 SYS OFF Mode (S_{OFF})

In SYS OFF Mode, SoC is completely powered down with no power supplied. processors are in shutdown mode with no state retention. Only NVM is retained. This mode is controlled external to the SoC by removing power to the device.

3.8 Debug Interfaces

The Apollo4 Blue SoC supports the following debug features:

- Embedded Trace Macrocell (ETM)
- Intruction Trace Macrocell (ITM)
- Trace Port Interface Unit (TPIU)
- Multi-core break support

An external debugger can be connected to the SoC using the Arm Serial Wire Debug (SWD) interface or the JTAG interface. The SWD interface is a 2-wire interface that is supported by a variety of off-the-shelf commercial debuggers, enabling customers to utilize their development environment of choice. JTAG is an industry standard interface and adheres to the IEEE 1149.1 specification.

3.8.1 Embedded Trace Macrocell (ETM)

The Apollo4 Blue SoC supports hardware instruction tracing via an Embedded Trace Macrocell (ETM). The ETM stream is accessible via APB or TPIU. An Embedded Trace Buffer provides 32 kB of trace buffering.

3.8.2 Instrumentation Trace Macrocell (ITM)

For system trace the processor integrates an Instrumentation Trace Macrocell (ITM) alongside data watchpoints and a profiling unit. To enable simple and cost-effective profiling of the system events these generate, a Serial Wire Viewer (SWV) can export a stream of software-generated messages, data trace, and profiling information through a single pin.

3.8.3 Trace Port Interface Unit (TPIU)

The Apollo4 Blue SoC includes two Cortex-M4 Trace Port Interface Units (TPIU). One TPIU is a legacy TPIU which operates exactly as the one found on the Apollo3 SoCs. The other TPIU can only be configured through the debugger to operate the ETM, ETB, and Trace Funnel. It is an Arm IP component that acts as a bridge between the on-chip trace data from the ITM and the single pin supporting the Serial Wire Viewer Protocol. The TPIUs include a Trace Output Serializer that can format and send the SWV protocol in either a Manchester encoded form or as a standard UART stream.

3.8.4 Faulting Address Trapping Hardware

The Apollo4 Blue SoC offers an optional facility for trapping the address associated with bus faults occurring on any of the three AMBA AHB buses on the chip. This facility must be specifically enabled so that energy is not wasted when one is not actively debugging.

4. Memory Subsystem

The Apollo4 Blue SoC integrates four kinds of memory:

- SRAM
- Integrated NVM (MRAM) / External Memory via MSPI (with cache)
- Boot Loader ROM
- One Time Programmable (OTP) memory

Key features include:

- 1024 kB Shared SRAM
- 384 kB TCM
- 480 kB Extended SRAM
- 2 MB NVM
- 64 kB NVM cache (2-way set-associative/Direct Mapped 128-bit line size)
- 16 kB OTP
  - 2 kB for customer use, including NVM protection fields
- NVM Protection specified in 16 kB Chunks
  - 128 OTP bits specify Write Protected Chunks
  - 128 OTP bits specify Read Protected Chunks
  - A Chunk is Execute Only if Both Corresponding Protection Bits Specified
  - OTP bits Specify Debugger Lock Out State
  - OTP bits Can Protect SRAM Contents From Debugger Inspection
- External flash with XiP (via MSPI) and cache support (up to 64 MB)

4.1 Functional Overview

The Apollo4 Blue SoC Integrates up to 2048 kB of on-board NVM and 16 kB of one-time programmable memory. These memories are managed by the APB NVM controller for write operations.

During normal MCU code execution, the NVM Cache Controller translates requests from the CPU core to the NVM Memory instance for instruction and data fetches. The controller is designed to return data in zero wait-states when accesses hit into the cache and can operate up to the maximum operating frequency of the CPU core. On cache misses, the Cache Controller issues miss requests to the NVM Memory Controller.

The Memory Controller facilitates NVM erase and programming operations. When erase or programming operations are active, instructions cannot be fetched for execution from the NVM memory, so the on-chip SRAM would have to be used for code execution. The cache controller ensures these operations are synchronized. To facilitate the management of NVM updates and OTP programming, a number of helper functions are provided in the boot loader ROM.

The boot loader ROM contains instructions that are executed upon power up of the processor. Once a valid reset vector is establish at offset zero in the NVM, the boot loader transfers control to users application by issuing a POR type reset which causes the core to enter the reset vector in NVM.

The Apollo4 Blue SoC supports secure boot leveraging the SecureSPOT technology. The root of trust for the secure boot is the boot ROM and the Ambiq secure boot loader. Secure boot, if enabled, will be invoked on each boot and reset cycle. Some secure boot functionality is conditionally supported on reset leveraging the SECBOOTONRST configuration in OTP.

The CPU subsystem includes a boot ROM which is the initial boot memory for the system. The boot ROM initiates the secure boot flow (if enabled) as well as other primitive/critical helper functions to facilitate accesses such as NVM programming.

4.2 Memory Controller

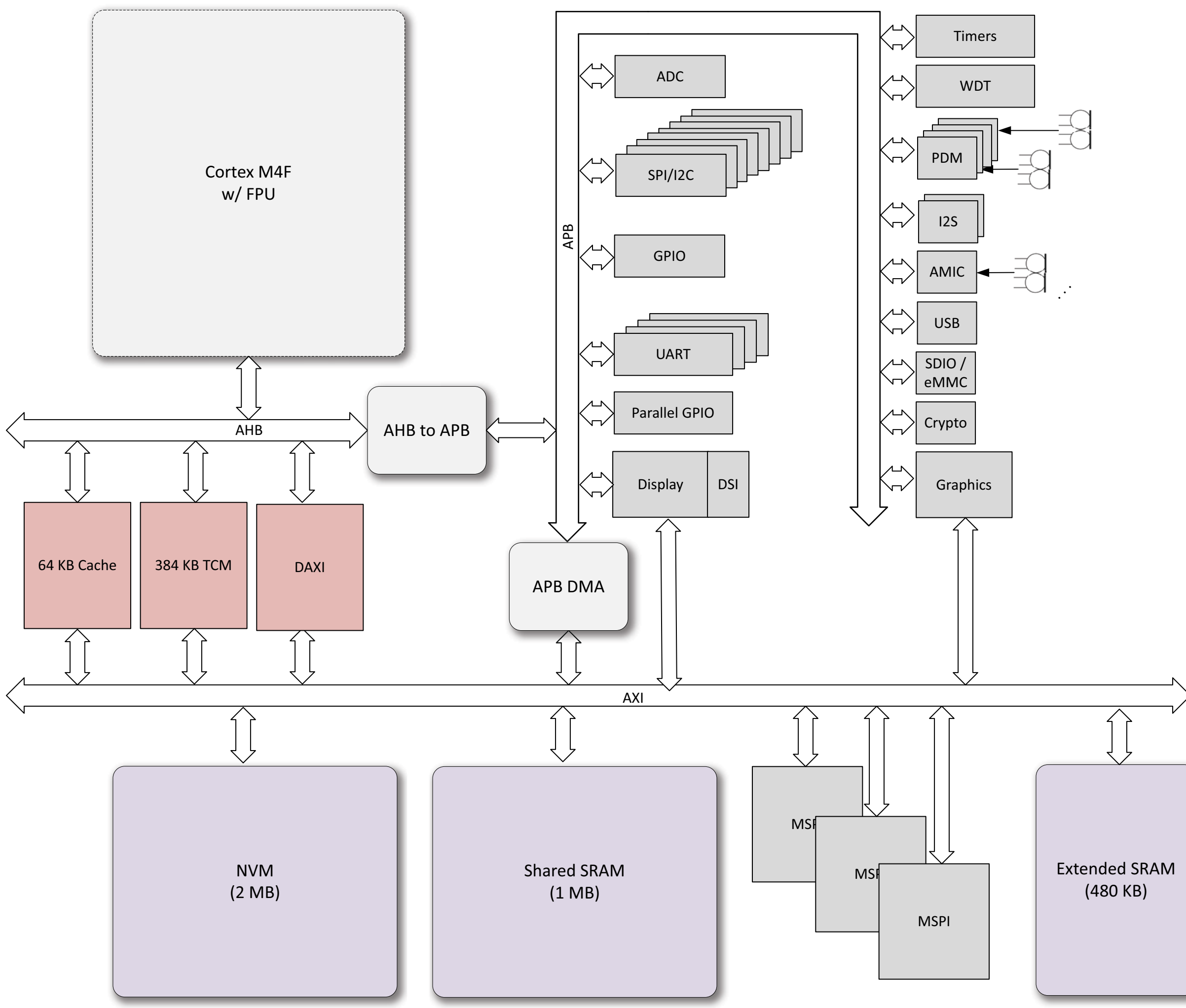


Figure 4. Apollo4 Blue SoC Peripherals, Memory and Buses

Interfaces to the different memory types consist of the following:

- Code cache
- Tightly Coupled Memory (TCM)
- NVM Interface
- Shared SRAM Interface

The NVM cache for the CPU subsystem caches all instruction and data accesses to the code region of the memory map. On the Apollo4 Blue SoC, this memory includes internal NVM and non-volatile external memory.

The NVM cache has the following features:

- 64 kB (additional lower power modes supported)
- Direct mapped or two-way set-associative
- 128-bit line size

On a cache miss, the cache controller will request the cache line from NVM (internal or external).

The TCM is accessible via the SRAM region. The TCM is a low-power, low-latency memory with configurable power enablement settings (PWRCTRL_MEMPWREN_ PWRENDTCM) for 8 kB, 128 kB and 384 kB configurations.

The Shared SRAM (SSRAM) is accessible via the SRAM region. The SSRAM is 1024 kB of shared system memory.

The CPU also has access to all of the 480 kB extended memory space. This allows the CPU to utilize the full memory space as additional data/instruction store if needed.

A Data-AXI (DAXI) module bridges the CM4's local AHB-D interface to the primary AXI crossbar for accesses that are outside of the NVM and the TCM interfaces. DAXI-enabled accesses are generally write buffer and light-duty caching with just a few 128-bit cache lines worth of data.

Specifically, the types of accesses that pass through the cache, DAXI, or neither can be summarized as follows:

Neither cached or DAXI access (single-cycle access):

- TCM
- Boot ROM

NVM Cache access:

- Instruction Fetches (AHB-I bus)
  - Internal NVM
  - SSRAM
  - Extended SRAM
  - MSPI XIP (Execute-In-Place) – off-chip memory/NVM
- Data (AHB-D bus)
  - Internal NVM

DAXI access:

- Data (AHB-D bus)
  - SSRAM
  - Extended SRAM
  - MSPI XIPMM and MSPI XIP (data) accesses to memory-mapped MSPIn apertures

NOTE
Cache must be placed in one of the 64 kB modes.

NOTE
No instruction accesses pass through the DAXI.

There are two other supported methods to access data via MSPI which use neither cache or the DAXI, and they are as follows:

- PIO access. This utilizes the MSPI’s 16-entry, 32-bit read/write FIFOs accessed through the APB bus (not AXI and therefore not DAXI).

- DMA access. This utilizes DMA and the MSPI's read/write FIFOs to efficiently transfer data between the MSPI device and another memory region.

4.2.1 DAXI

The DAXI contains 128-bit line buffers (4 x 32-bit), which buffers both reads and writes. Programmable read/write aging and replacement policy is also supported.

As mentioned in Section 4.2 on page 63, data accesses over the AHB-D bus to various memory types/ regions using the DAXI are the following:

- SSRAM
- Extended SRAM
- MSPI XIPMM and MSPI XIP (data) accesses to memory-mapped MSPIn apertures

These data accesses are located in the memory map at 0x10060000 – 0x1FFFFFFF.

TCM, NVM and MSPI XIP (instruction) accesses do not go through the DAXI.

NOTE

Due to an existing limitation on Apollo4 SoC, the DAXI must be used with either 2 buffers or 1 buffer enabled and, in either case, the flush level must be set to 0.

NOTE

There is no non-cached MSPI aperture nor can DAXI caching be configured based on address. The method to make something non-cached is to enable non-cacheable regions in the cache.

NOTE

Software coherency is a concern with the DAXI in that the DAXI may contain data from a stale DMA'd buffer. It should be invalidated before reading a reused DMA buffer region in SSRAM, for example.

Software coherency may also be of concern when writing out data to a buffer in SSRAM, Extended SRAM, or MSPI which will be read by a peripheral using DMA, or GPU or Display Controller.

The DAXI must be flushed before entering deepsleep and potentially before entering sleep if software coherency is of concern.

Please see the Apollo4 Family Programmer's Guide for additional, detailed information about DAXI use and operation.

4.2.1.1 Line Buffers

The DAXI module utilizes 128-bit line buffers which provide read caching and write buffering to the system memory interfaces (SSRAM, MSPI, etc). Each line is in, and is marked as, one of four “MWSI” states which have the following characteristics:

- MODIFIED (M): Read buffer that has local write modifications which need to be flushed. Modified lines contain cached read data, as well as hold partial (or full) updates from write operations. MODIFIED lines must be written back to system memory before re-allocation.
- WRITE (W): Buffer is allocated as a write buffer. Writes will accumulate in the buffer until the line is flushed. Byte enables indicate which bytes have been written for write-back.Lines in the WRITE state have write data but do not have valid read data. They are simply buffered writes and allow the DAXI to accumulate as many writes as possible before committing them to the AXI subsystem. WRITE lines can be converted to MODIFIED on a read operation by the CPU to the line's address.
- SHARED (S): Data read from system memory that will be cached for subsequent reads. The line contains data read from the AXI bus and reads will be serviced from the line. SHARED lines are converted to MODIFIED lines on a write to its address. SHARED lines are allocated only if no INVALID lines are available.
- INVALID (I): Buffer is invalidated.The invalid state basically indicates the line buffer is not in use and its data is invalid; the buffer is available for allocation. INVALID lines are the first buffers used for re-allocation.

From the invalid state, there are three common paths that a line buffer’s state might take:

- On a read, the DAXI would issue a read to the primary AXI bus and the buffer would be allocated (SHARED state) to hold the returned 128-bit data. Data would remain cached until a flush (in which case the line is invalidated) or until it is reallocated for a different read or write operation.
- On a read followed by a write operation from the Arm, a buffer would start out as SHARED (from the read) and then transition to MODIFIED after the write operation. It would then be flushed before being reallocated.
- On a write, the DAXI would allocate the line buffer to hold write data (WRITE state). Byte enables would indicate which bytes have been written. In the case of adjacent or subsequent writes to that line, the DAXI will continue updating bytes as the Arm issues them (think of the case of the Arm sequentially writing bytes or words of data). If the Arm reads any data from this line, the DAXI will issue a read operation on the AXI bus and merge the incoming read data with existing write data and convert the buffer to the MODIFIED state.

In addition to the three common paths listed above, lines are converted from WRITE to MODIFIED when all bytes are written.

4.2.1.1.1 Buffer Allocation

Generally, the DAXI will allocate buffers as follows:

1. The DAXI selects a buffer in the INVALID state if one is available.
2. The DAXI examines the LRU of each active buffer and flush/invalidate the buffer for re-allocation. Shared buffers also have preference for re-allocation since no write-back is required.
3. When two or more buffers are marked as W or M, the DAXI may flush these to free buffers. Buffers marked as M will be flushed and marked as S (to allow additional reads) which also allows them to be quickly reallocated.

4.2.1.1.2 Hardware Flush Level

The DAXI has a FLUSHLEVEL configuration that indicates how aggressively the DAXI flushes buffers in order to ensure that resources are always available for the CPU. When the FLUSHLEVEL is set to 0, the DAXI will attempt to maintain two free buffers (maximum of two buffers in the M/W states) and when the

FLUSHLEVEL is set to 1 it will allow up to 3 buffers to remain in the M/W state. This bit should be set to 1 only if the number of supported buffers is 3 or more.

Additionally, the DAXI configuration register allows each of the buffers to be independently enabled or disabled. It is expected that the full set of available buffers be enabled, but this option provides the ability to disable buffers to work around bugs or determine the performance impact of fewer (or more) buffers.

4.2.1.2 Aging Counter

An LRU/Aging counter manages line allocation and temporal flushes, including flushing on sleep and sideband flush controls. An aging counter is a gray-coded counter that increments every N CPU clock cycles, where N is programmable from 1 to 255 and the default is 2 aging counter time steps before flush. Every time a buffer is allocated or a read/write transaction utilized a buffer, it records the time on the aging counter. Lines will automatically self-flush after the aging counter reaches a count of two more than the originally stored counter value. This ensures that data is routinely flushed from the DAXI as it is no longer used and becoming stale and assists in maintaining data coherency. However, software must use the DAXI sync controls in some situations (like writing to a buffer used for DMA) to ensure coherency when interacting with other agents in the hardware.

The Aging Counter (AC) setting essentially sets a temporal window in which any buffer that is unused during the time period will be invalidated or flushed. Each buffer will be allowed to remain idle for a minimum of time of T_{AC} and a maximum time of $2*T_{AC}$.Figure 5 demonstrates the concept by showing the progression of the aging counter through its four states in each of 5 time steps as indicated by different colored blocks at the top, and operations on each of the four buffers as colored circles.

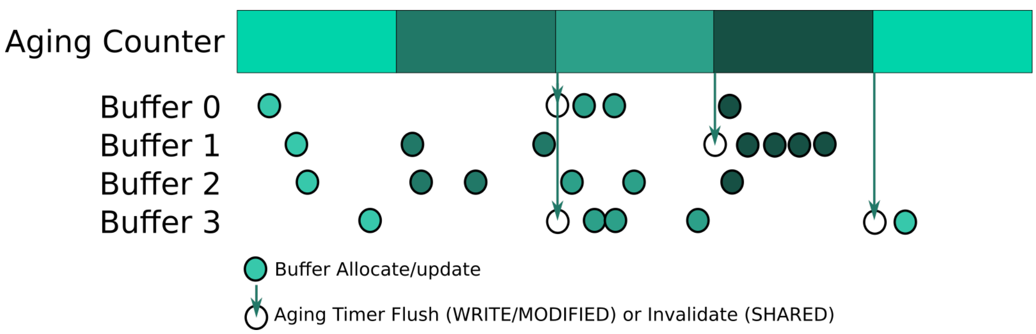


Figure 5. Aging Counter Operation

- Time step 1: All four buffers are allocated (for read or write) and the time stamp associated with each buffer is set to the current aging counter state.
- Time step 2: Buffers 1 and 2 are accessed by the CPU. These could be read hits to a buffer marked Shared (S), writes to a Write (W) or Modified (M) buffer, or a reallocation of a buffer.
- Time step 3: Buffers 0 and 3 are flushed by the aging counter. An S buffer is simply invalidated (to prevent stale read data from being cached too long) and a W/M buffer is flushed to ensure that stale write data doesn't remain cached too long. These buffers are the first to be reallocated. Buffer 2 is updated to the current time stamp with an additional access by the CPU.
- Time step 4: Buffer 1 is flushed by the aging timer, while buffers 0 and 2 are updated to the new time stamp with additional accesses from the CPU.
- Time step 5: Buffer 3 is invalidated and subsequently re-allocated.

In additional to being flushed by the aging counter, the DAXI control logic will dynamically flush/invalidate buffers before the timer ages in cases where the DAXI needs to re-allocate the buffer to support the current CPU operations.

4.2.1.3 Least Recently Used Counter (LRU)

Each line maintains an LRU counter that is reset to zero on every read, write, or allocation event to that line. The counter is then incremented each time one of the other buffers has an event. The LRU as well as buffer state are used by the DAXI to determine which lines to flush/invalidate when a new buffer allocation is required.

4.2.1.4 Write FIFO

In addition to the line buffers, the Reduced-AXI (RAXI) interface provides a 2-entry deep write FIFO (like the address FIFO) as part of the asynchronous AXI interface. On the AXI, the address transactions are decoupled from the read data/write data transactions, but are issued in order, with address FIFO transaction push either simultaneous with, or leading, write FIFO transaction. In other words, there are 2 DAXI-to-RAXI outbound FIFOs of 2-entries deep for address and write data, and 2 RAXI-to-DAXI inbound FIFOs of 1-entry deep for read data and write response/acknowledgment.

Flushes from line buffers are immediately accepted by the RAXI (unless it is full), allowing WRITE (W) / MODIFIED (M) lines to be reallocated quickly.

On the AXI, the address transactions are decoupled from the read data / write data transactions. But for this implementation they will be issued in order, with an address FIFO transaction push either simultaneous with, or leading, the write FIFO transaction. In other words, there are 2 DAXI-to-RAXI outbound FIFOs of 4 entries deep for address and write data and 2 RAXI-to-DAXI inbound FIFOs of 2 entries deep for read data and write response/acknowledgment.

In order to empty the DAXI completely, it is necessary to transition all buffers in Write or Modified states to Invalid or Shared states, respectively, by doing a flush operation. Then an invalidate operation can take the remaining buffers in the Shared state to the Invalid state. The end result is that all line buffers are in the Invalid state. Therefore, the sequence for a DAXI “barrier” operation should be to flush (transitions W->I and M->S) and then invalidate (transitions S->I). See definitions of buffer states in “Line Buffers” on page 66.

4.2.2 NVM Cache

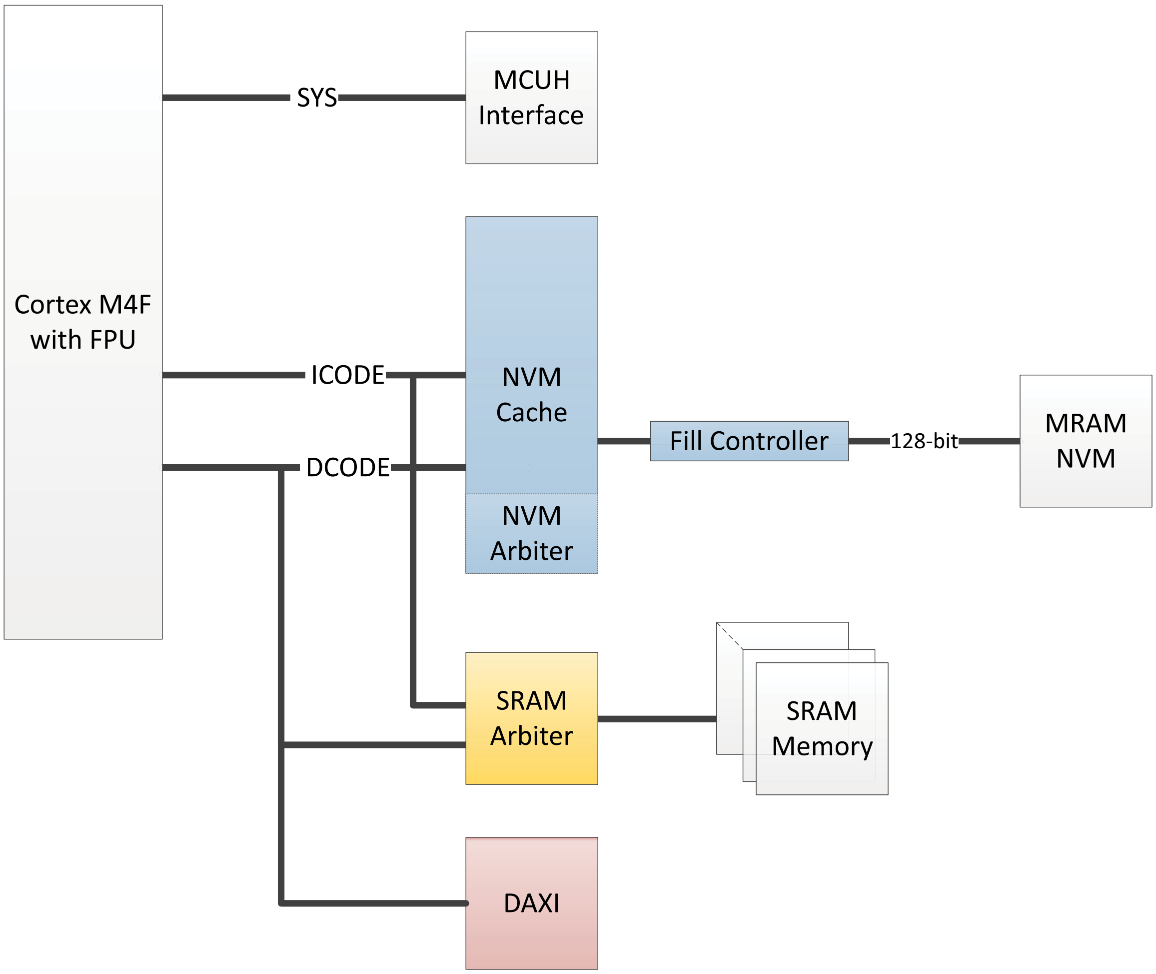


Figure 6. Block Diagram for Apollo4 Blue SoC NVM Cache

The Apollo4 Blue SoC incorporates a NVM cache to the I-Code and D-Code path from the microcontroller. This controller is intended to provide single cycle read access to NVM and reduce overall accesses to the NVM to reduce power. The controller is a unified I-Code and D-Code cache controller. The cache fill path is arbitrated between cache misses as well as the other NVM read agents, e.g., Info, Reg, Built-in Self Test (BIST).Caching is supported for the entire 2 MB internal NVM and all MSPI apertures. The cache is configurable 2-way set associative or direct mapped, 128-bit line size.

- Both I_BUS and D_BUS *reads* from internal NVM are cached by the NVM Cache.
  - All NVM accesses go through the NVM cache. They do not go through DAXI.
- For all other memory ranges (except TCM), Instruction fetches (I_BUS reads) are cached. This includes Instruction fetch from SSRAM, Extended RAM, and MSPI XIP.
  - The NVM does not cache D_BUS data reads from any memory other than MRAM.
  - TCM is accessed directly, and does not go through NVM cache.
- NVM cache fetch/miss accesses go directly onto AXI. They do not go through DAXI. The cache has its own set of line buffers from which it can hit.
- The NVM cache does not cache any *writes*. The cache's line buffers cannot be disabled. They are always active even if the cache is disabled.

- On a cache configuration change or a NVM program cycle, the cache should self-invalidate. However, the cache is unaware of any SSRAM or external XIP PSRAM update, so if paging in and out instruction overlays from some external memory to SSRAM for example, it is important that software invalidates the cache before accessing that region again.

4.2.2.1 Data fetches from XIP via NVM cache

In the Apollo3 memory sub-system architecture, there are two XIP mappings available, each with their own address range:

- XIP to external "NVM" which supported caching of instruction/data, but does not support direct writes by instructions as it is mapped as NVM.
- XIPMM to external "SRAM" which was not cached, but supports direct reads/writes as it is mapped as RAM.

Both of these separate XIP/XIPMM memory ranges can be mapped to the same external XIP device, such as a PSRAM, allowing one XIP device to support a mix of cached instruction/data fetches and non-cached data reads and writes. Overlapping these ranges is allowed but care should be taken when doing so. Since these XIP and XIPMM accesses use completely different addresses, it is easy to determine when an MSPI access is treated as a cacheable NVM access and when it is treated as non-cacheable SRAM access.

On the Apollo4 family SoCs, the XIP and XIPMM have been merged so that the two types of access utilize the same address ranges. Reads from areas used as read/write are not cacheable. Figure 7 below shows the SoC Bus Architecture and access to external memory on the AXI bus.

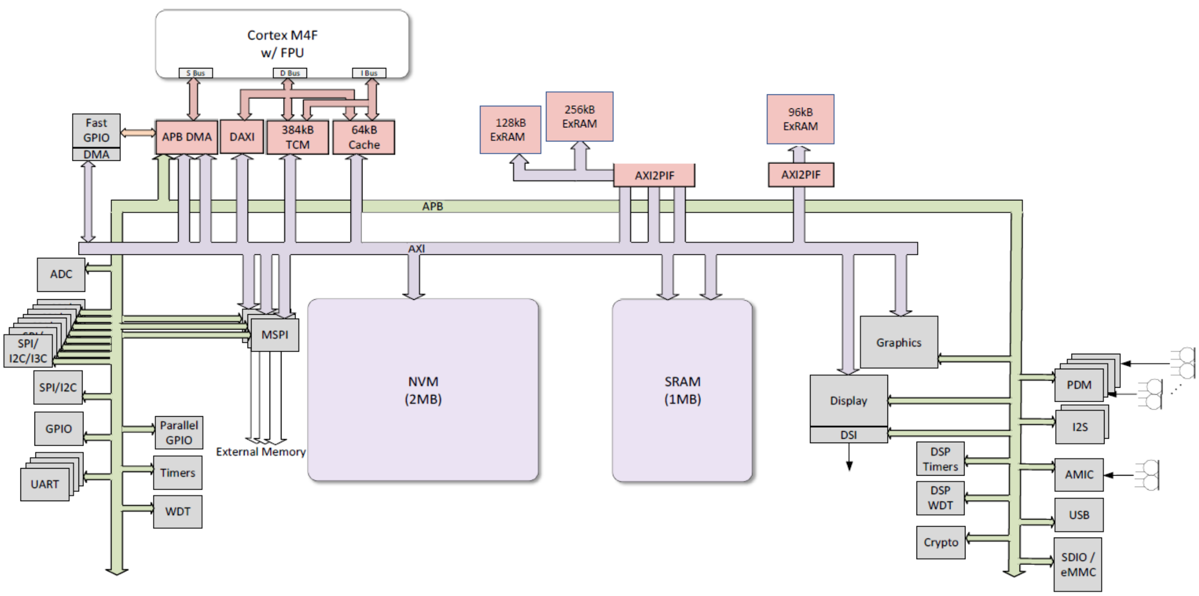


Figure 7. Apollo4 Blue SoC Bus Architecture Block Diagram

4.3 Wait States for Accesses to Memory Types

As mentioned, accesses to memory fall into one of four channels:

- Code cache
- Tightly Coupled Memory
- NVM Interface
- Shared SRAM Interface

There are six supported cache modes:

- W1_128B_512E = 0x4 - Direct mapped, 128-bit line size, 512 entries (4 SRAMs active)
- W2_128B_512E = 0x5 - Two-way set associative, 128-bit line size, 512 entries (8 SRAMs active)
- W1_128B_1024E = 0x8 - Direct mapped, 128-bit line size, 1024 entries (8 SRAMs active)
- W1_128B_2048E = 0xC - Direct mapped, 128-bit line size, 2048 entries (4 SRAMs active)
- W2_128B_2048E = 0xD - Two-way set associative, 128-bit line size, 2048 entries (8 SRAMs active)
- W1_128B_4096E = 0xE - Direct mapped, 128-bit line size, 4096 entries (8 SRAMs active)

64 kB mode can be configured as W2_128B_2048E or W1_128B_4096E. For MSPI memory-mapped XIP access, W1_128B_4096E mode must be used. All other modes are 16 kB.

On a cache miss, the cache controller will request the cache line from NVM (internal or external) or internal/external RAM.

The Tightly Coupled Memory (TCM) is accessible via the SRAM region. The TCM is a low power / low latency memory with configurable power settings for 8 kB, 128 kB and 384 kB configurations.

The Shared SRAM (SSRAM) is accessible via the SRAM region. The SSRAM is 1024 kB of shared system memory. Access to this memory region incurs 7 or 13 cycle wait states (depending on the CPU operating mode).

The CPU also has access to 480 kB of extended SRAM memory. The extended memory region incurs 10 or 19 cycle wait states (depending on the CPU operating mode).

The CPU subsystem includes a boot ROM which is the initial boot memory for the system. The boot ROM initiates the secure boot flow (if enabled) as well as other primitive/critical helper functions to facilitate accesses such as NVM programming.

Figure 8 below shows the bus interconnections between the Cortex M4F CPU and the various memory/ storage elements and peripherals on the Apolllo4. Aside from the direct access to TCM, accesses between the CPU and a memory or peripheral are through the instruction or data cache, and then over the AXI bus to one of the memory or storage elements. Access to an external storage device is through one of the MSPI instances.

In addition to CPU accesses to each memory type, the Apollo4 Blue SoC supports direct memory access (DMA) to the various memories by APB peripherals. These direct accesses are enabled through multiple DMA controllers as follows:

- Graphics Controller
  - Has dedicated DMA
  - DMA has access to all memory
  - Supports multiple threads and dedicated FIFOs to accommodate memory latencies
- Display Controller
  - Has dedicated DMA
  - DMA has access to all memory
- Other DMA supported peripherals
  - DMA is handled through APB DMA controller
  - Each device has high/low priority
  - Supports up to 16 beats of data per arbitration cycle
  - DMA source/target can be any RAM or external memory (NVM only as source for limited devices)

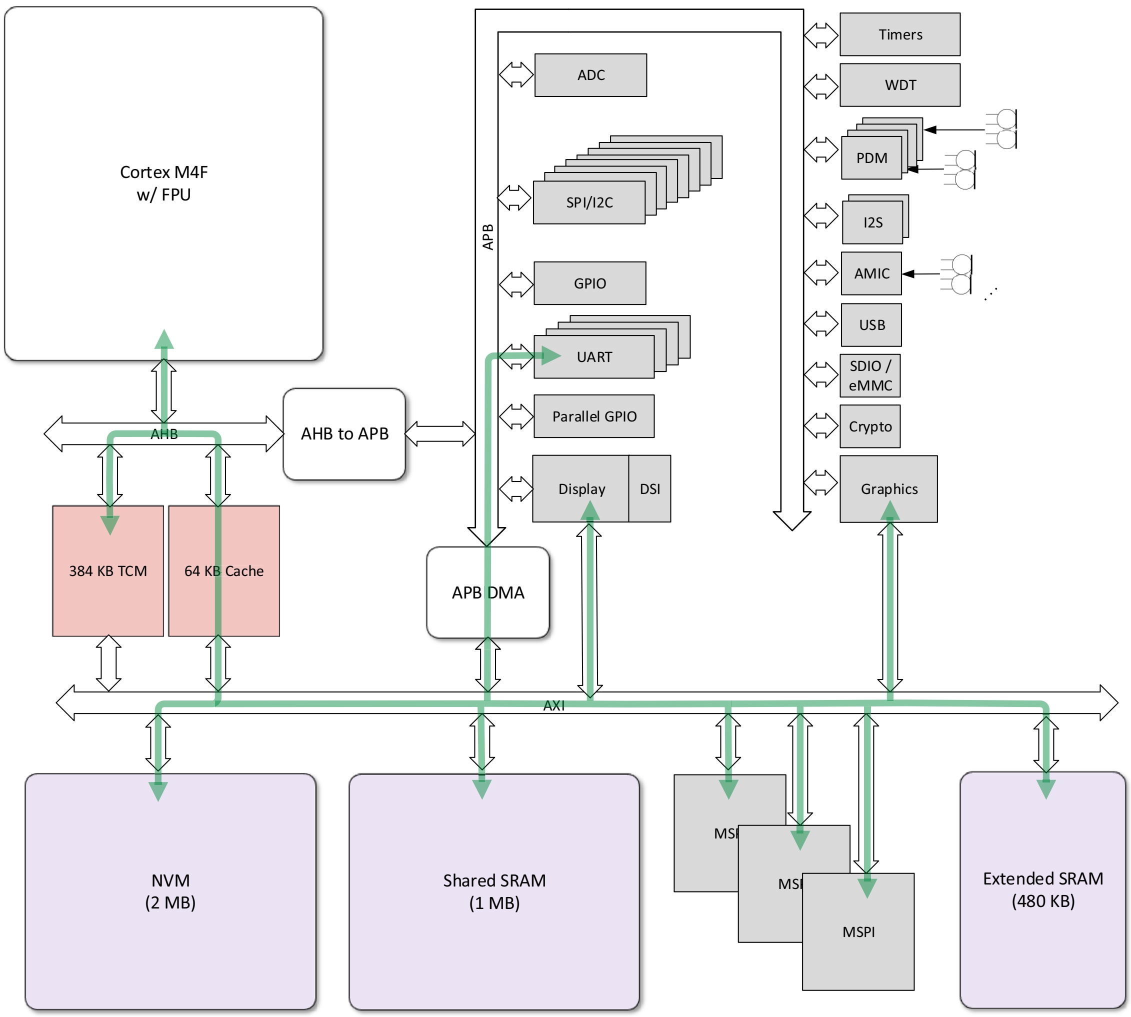


Figure 8. Apollo4 Blue SoC System Diagram

The efficiency of the interoperability of the CPU, cache, AXI bus, DMAs and memories can be seen in Table 3, which summarizes the number of wait states based on cache hits or misses for memory accesses to each type of memory supported on the Apollo4 Blue SoC.

Table 3: Wait States for Accesses to/from the CPU and Memory/Storage Elements

Memory / Storage Element	Cache Access	Accessed Memory	Access Type	DMA Bank Conflict	Cache State[1]	Wait States	Notes
TCM	No	Data	Read/ Write	No	-	0	
TCM	No	Data	Read/ Write	Yes	-	N_{TCM}	- Dependent on DMA activity and conflict to same 32KB bank of memory - Max transfer size is 16 before arbitration
TCM	No	Instruction	Read	No	-	0	
TCM	No	Instruction	Read	Yes	-	N_{TCM}	- Dependent on DMA activity and conflict to same 32KB bank of memory - Max transfer size is 16 before arbitration
SSRAM	Yes	Data	Write	-	Available	0	
SSRAM	Yes	Data	Write	-	Unavail-able	N_{WR}	- 4-entry, 128b line buffer (LRU allocated) - Sub-128b writes are accumulated into buffer - Writes are flushed based on time, evic-tion or flush command - Dependent on bus load - Unloaded is 0 cycles
SSRAM	Yes	Data	Read	-	Hit	0	
SSRAM	Yes	Data	Read	-	Miss	N_{RD}	- Critical word (miss) - Hits serviced from 4 entry 128b line buf-fer - Dependent on bus load - Unloaded is 7 cycles (LP) and 13 cycles (HP)
SSRAM	Yes	Instruction	Read	-	Hit	0	
SSRAM	Yes	Instruction	Read	-	Miss	N_{MISS}	- Critical word (miss) - Dependent on bus load - Unloaded is 7 cycles (LP) and 13 cycles (HP)

Table 3: Wait States for Accesses to/from the CPU and Memory/Storage Elements

Memory / Storage Element	Cache Access	Accessed Memory	Access Type	DMA Bank Conflict	Cache State[1]	Wait States	Notes
Extended SRAM	Yes	Data	Write	-	Available	0	
				-	Unavail-able	N_{WR}	- 4-entry, 128b line buffer (LRU allocated) - Sub-128b writes are accumulated into buffer - Writes are flushed based on time, evic-tion or flush command - Dependent on bus load - Unloaded is 0 cycles
			Read	-	Hit	0	
				-	Miss	N_{RD}	- Critical word (miss) - Hits serviced from 4-entry 128b line buf-fer - Dependent on bus load - Unloaded is 10 cycles (LP) and 19 cycles (HP)
	Yes	Instruction	Read	-	Hit	0	
				-	Miss	N_{MISS}	- Critical word (miss) - Dependent on bus load - Unloaded is 10 cycles (LP) and 19 cycles (HP)
MSPI	Yes	Data[2]	Write	-	Available	0	
				-	Unavail-able	N_{WR}	- 4-entry, 128b line buffer (LRU allocated) - Sub-128b writes are accumulated into buffer - Writes are flushed based on time, evic-tion or flush command - Dependent on bus load - Unloaded is 0 cycles
			Read	-	Hit	0	
				-	Miss	N_{RD}	- Critical word (miss) - Hits serviced from 4-entry 128b line buf-fer - Dependent on bus load - Unloaded is 39-45 cycles (LP) and 77-89 cycles (HP)[3]
	Yes	Instruction (XiP)	Read	-	Hit	0	
				-	Miss	N_{MISS}	- Critical word (miss) - Dependent on bus load - Unloaded is 39-45 cycles (LP) and 77-89 cycles (HP) + MSPI bus latency (varies by device)[3]
NVM	Yes	Instruction/ Data	Read	-	Hit	0	
				-	Miss	N_{MISS}	- Critical word (miss) - Dependent on bus load - Unloaded is 11 cycles (LP) and 21 cycles (HP)

1. Cache slot available/unavailable on a write, or hit/miss on a read
2. Memory-mapped MSPI data accesses incur the same latency concerns as instruction (XIP) accesses, which is dependent on MSPI bus width, frequency and turnaround time.
3. MSPI wait states are affected by DQS mode and the TURNAROUND settings.

4.4 One-Time Programmable (OTP) Memory

There is up to 16 kB of OTP memory on the Apollo4 Blue SoC. It is partitioned into 3 primary sections - Ambiq trim, Customer trim and Secure OTP. Each partition is OTP protected via hardware.

The Ambiq Trim OTP partition is used to store trims required for the per-chip functionality and optimization. These trims are provisioned at manufacturing. The Customer Trim OTP partition is used to store trims specific to customer implementation (e.g. buck configuration). These trims can be provisioned at customer manufacturing or as part of a trim update in the field. The Secure OTP partition is exclusively controlled by the crypto hardware. This partition is provisioned at different stages of the device life cycle using secure provisioning utilities.

5. Reset Generator (RSTGEN)

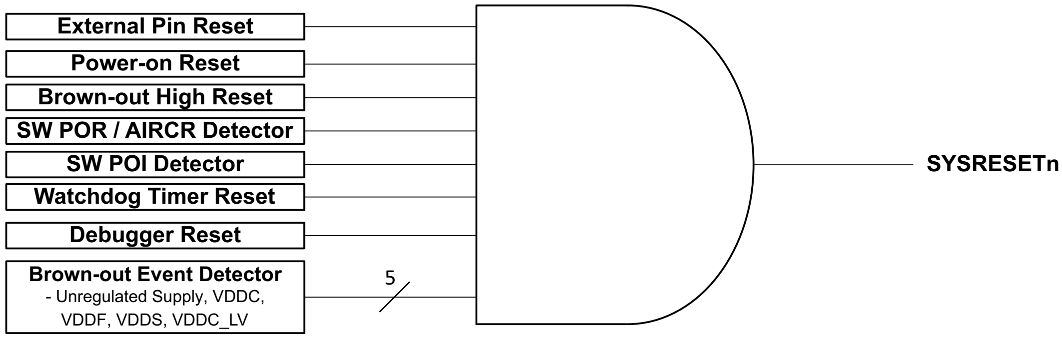


Figure 9. Block diagram for the Reset Generator Module

5.1 Functional Overview

The Reset Generator Module (RSTGEN) monitors a variety of reset signals and asserts the active low system reset (SYSRESETn) accordingly. A reset causes the entire system to be re-initialized, and the cause of the most recent reset is indicated by the STAT register.

Reset sources are described in the subsequent sections and include:

- External reset pin (RSTn)
- Power-on event
- Brown-out events
- Software request (SYSRESETREQn)
- Watchdog expiration

Please refer to the RSTGEN registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

5.2 External Reset Pin

The active-low RSTn pin can be used to generate a reset using an off-chip component (e.g., a push-button). An internal pull-up resistor in the RSTn pad enables optional floating of the RSTn pin, and a de-bounce circuit ensures that bounce glitches on RSTn does not cause unintentional resets. The RSTn pin is not maskable. An internal pull-down device will be active during a brownout event pulling the RSTn pin low. See Figure 10.

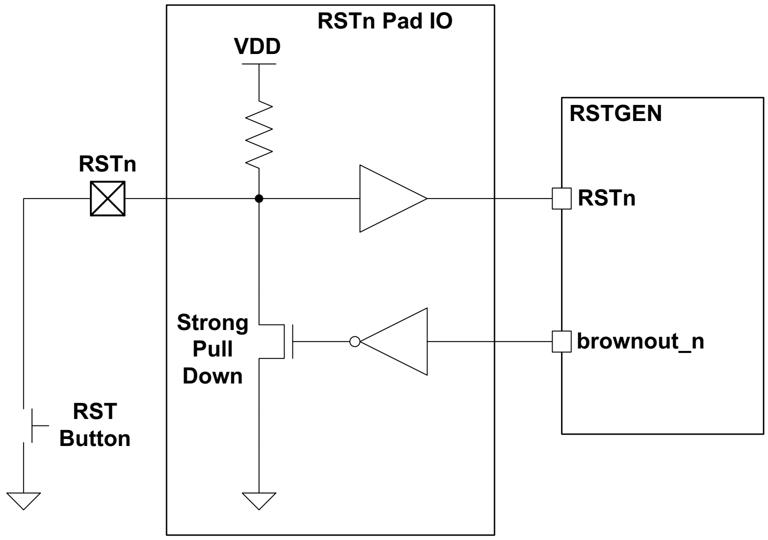


Figure 10. Block diagram of circuitry for Reset pin

5.3 Power-on Event

An integrated power-on detector monitors the supply voltage and keeps SYSRESETn asserted while VDD is below the rising power-on voltage, V_{POR+}. When VDD rises above V_{POR} at initial power on, the reset module will initialize the low power analog circuitry followed by de-assertion of SYSRESETn, and normal operation proceeds. SYSRESETn is re-asserted as soon as VDD falls below the falling power-on voltage, V_{POR-} . The power-on reset signal, PORn, is not maskable.

5.4 Brown-out Events

There are multiple brownout detectors in the Apollo4 Blue SoC. An integrated brown-out detector monitors the primary supply voltage and causes an automatic and non-configurable reset when the voltage has fallen below the low brownout threshold (BODL). An optional reset or interrupt can be enabled when the brown-out detector indicates the supply voltage has fallen below the high brownout threshold (BODH).

NOTE
WARNING: The brown out high reset should not be enabled if the supply voltage is lower than the BODH reset level (2.1V). Enabling this reset (RSTGEN_CFG_BODHREN = 1) in this situation causes repeated resets.

In addition, there are individual brownout detector monitors integrated within the core/memory and Bluetooth Low Energy supply regulators which cause separate/maskable reset assertions when the voltage falls below critical level for the respective voltage rails - VDDC, VDDC_LV, VDDS or VDDF. In the event the primary supply voltage falls below its high brownout threshold (BODH), or any of the other supplies fall below its corresponding core/memory/Bluetooth Low Energy threshold if enabled, the reset module will initiate a system reset, enabling the RSTn pull-down and driving the reset pin low. The occurrence of a BODH reset will be reflected by the setting of the BODH bit in the RSTGEN's INTSTAT Register after reset, and similarly for the other four selectable brownout resets.

In the event of a brownout detection, the following functionality is maintained until a power down detection occurs.

- All RTC registers retain state
- RTC and STIMER counters continue operation from 32 kHz XTAL or from LFRC (if below BODL). If clock sources stop oscillating at very low voltage, the RTC and STIMER will continue to maintain state.
- Clock configuration registers retain state

5.5 Software Reset

A reset may be generated via software using the Application Interrupt and Reset Control Register (AIRCR) defined in the Cortex-M4. For additional information on the AIRCR, see the Arm document titled “Cortex-M4 Devices Generic User Guide.” The software reset request is not maskable. A second source for the identical software reset functionality is made available through the SWPOR register in the RSTGEN peripheral module.

5.6 Watchdog Reset

The Watchdog Timer sub-module generates an interrupt if it has not been properly managed by software within a pre-defined time. The watchdog reset is maskable.

6. Clock Generator (CLKGEN)

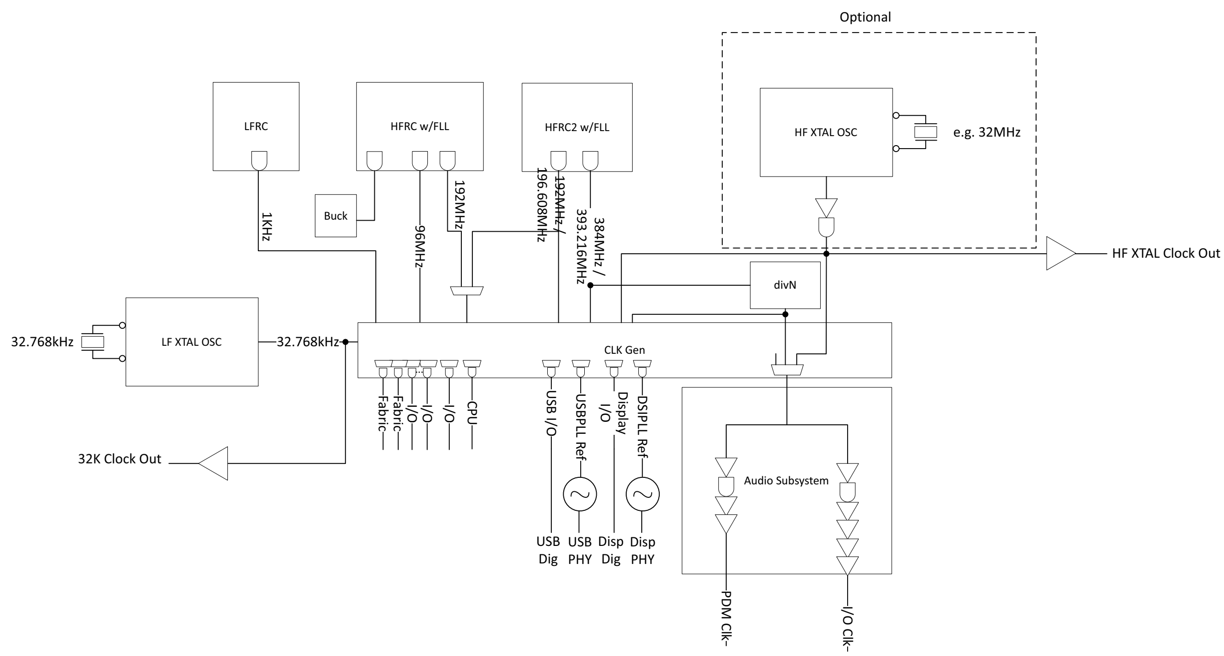


Figure 11. Block diagram for the Clock Generator

6.1 Features

The Apollo4 Blue SoC clock generation subsystem is responsible for generating all of the primary and derived clocks in the SoC.

- Independent frequency scaling for various SoC subsystems
- Ultra low power, low frequency clock generation with XTAL calibration
- Programmable I/O clock dividers
- High precision audio clock generation

> NOTE
>
> When enabling a module which automatically starts clocking with a default clock source, or when changing the clock source for any enabled module, there is a required 30 µs settling time for the selected clock.

Please refer to the CLKGEN registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

6.2 Functional Overview

A high-level view of the Clock Generator Module, which supplies all clocks required by the Apollo4 Blue SoC, is shown in Figure 11. Note that the output clock frequencies from the clock sources are nominal values. Consult the Electrical Characteristics section for specified values.

The clock generation subsystem consists of the following sub-modules:

- High frequency XTAL oscillator circuit
- Low frequency XTAL oscillator circuit
- 2x High frequency RC oscillator
- Low frequency RC oscillator
- High frequency PLL circuits
- SoC clock generation logic
- Audio subsystem clock generation logic

The high frequency RC block (HFRC) provides all the primary clocks for the high frequency digital processing blocks in the SoC except for audio, radio and HP mode clocks. These clocks are gated/ selected based on performance requirements. The digital clocks are isolated to avoid noise injection into the critical clocks for audio and radio communications. Additionally, the high frequency digital clock is programmatically divided to generate the various I/O clocks in the system. All high frequency clocks can be gated if not needed. The HFRC also supports a frequency-locked loop (FLL) circuit to ensure the HFRC oscillator locks to a specific frequency range to ensure high quality/low ppm output reference as needed to meet audio clock quality requirements. Although the HFRC can be calibrated to a variety of input reference clocks, the primary clock reference is the 32 kHz XTAL input.

The PLLs are used to create the clocks needed for the USB and Display subsystem.

The second high frequency RC block (HFRC2) provides a high accuracy clock required for audio applications. The HFRC2 supports a FLL circuit to ensure the HFRC2 oscillator locks to a specific frequency range to ensure high quality/low ppm output reference. Although the HFRC2 can be calibrated to a variety of input reference clocks, the primary clock reference is the HF XTAL input.

The low frequency RC block (LFRC) provides the low frequency clocks for timers and other logic within the SoC. The LFRC takes either the high frequency or low frequency XTAL as source.

The high frequency XTAL sets the primary clock input for the audio and radio subsystems. For configurations using Ambiq BlueSPOT, the XTAL is 32 MHz. However, additional XTAL frequencies need to be supported for different configurations. An external clock request is sourced to indicate the XTAL is needed by the external device. This is used in coordination with internal clock request logic to determine if the XTAL can be powered down.

The divN functionality provides integer divide ratio to generate the appropriate frequencies (e.g., 24.576 MHz or a 22.579 MHz to support 44.1 kHz sampling) needed for audio use cases. A bypass option also allows the HF XTAL to be used directly (in the case a 24.576 MHz or 22.579 MHz XTAL can be used for audio only).

An output clock is generated from the HF XTAL circuit to support external radio clocking. This output clock is intended to be high quality to ensure radio requirements can be met.

6.3 Low Frequency RC Oscillator (LFRC)

The low power LFRC, with a nominal frequency of 1024 Hz, is used when short term frequency accuracy is not important. It also supplies clocks for SIMO buck regulator in low power mode (32 kHz) as well as some basic state machines and is always enabled.

6.4 High Precision XT Oscillator (XT)

The high accuracy XT Oscillator is tuned to an external 32.768 kHz crystal, and has a nominal frequency of 32.768 kHz. It is used when frequency accuracy is critically important. Because a crystal oscillator uses a significant amount of power, the XT is only enabled when an internal module is using it.

It should be noted that the XT oscillator is also optional if the requirements of the design can tolerate the internal LFRC/HFRC oscillator specifications. It should also be noted that external capacitors are not required to tune an internal divided clock of the crystal input to achieve a precise scaling of 32.768 kHz. This is handled within the Apollo4 Blue SoC.

> **NOTE**
>
> **The XTAL is highly sensitive to external leakage on the XI pin. Therefore it is recommended to minimize the components on XI and to use extremely low leakage load capacitors.**

The RTC clock source, either the LFRC Oscillator or the XT Oscillator, is selected via the REG_CLKGEN_OCTRL_OSEL bit. If the XT Oscillator experiences a temporary failure and subsequently restarts, the Apollo4 Blue SoC will switch back to the XT Oscillator.

6.5 High Frequency RC Oscillator (HFRC)

The high frequency HFRC Oscillator, with a nominal frequency of 96 MHz, is used to supply all high frequency clocks in the Apollo4 Blue SoC such as the processor clock for the Arm core, memories and many peripheral modules. Digital calibration is not supported for the HFRC, but its frequency may be automatically adjusted by the Auto-adjustment function which is a combination of analog and digital operations.

The HFRC is enabled only when it is required by an internal module. When the Arm core goes into a sleep mode, the HFRC will be disabled unless another module is using it. If the Arm core goes into deep sleep mode, the HFRC will be powered down when it is not needed. When the HFRC is powered up, it will take a few microseconds for it to begin oscillating, and a few more microseconds before the output is completely stable. In order to prevent erroneous internal clocks from occurring, the internal clocks are gated until the HFRC is stable.

The Apollo4 Blue SoC supports high frequency TurboSPOT™ burst mode. The HFRC supplies both the 192 MHz as well as the 96 MHz clocks to support the high frequency core/memory domains and the stable 96 MHz clock for the remaining logic/IO controllers.

7. Real Time Clock (RTC)

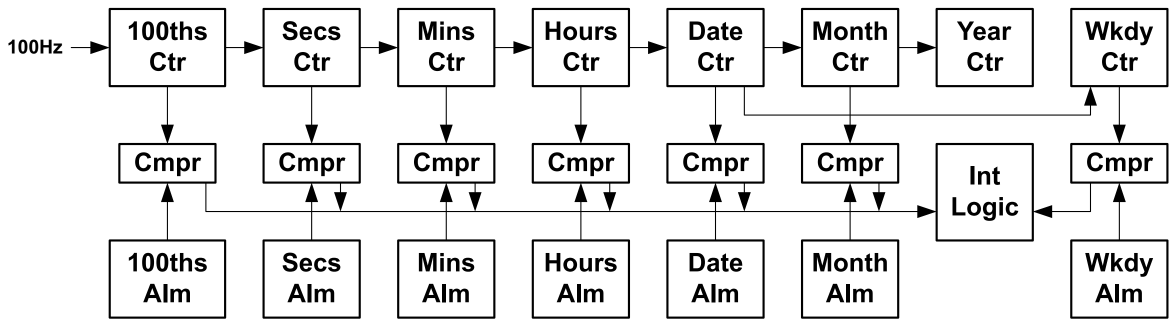


Figure 12. Block diagram for the Real Time Clock Module

7.1 Functional Overview

The Real Time Clock (RTC) Module, shown in Figure 12, provides an accurate real time measurement. Key features are:

- 100^{th} of a second resolution
- Time is measured for the years between 1900 and 2199
- Automatic leap year calculation
- Hours are specified in 24 hour mode
- Alarm precise to 1/100 second
- Alarm interval every 100^{th} second, 10^{th} second, second, minute, hour, day, week, month or year.
- 100 Hz input clock taken from either the high accuracy XT Oscillator or the low power LFRC Oscillator.

7.2 Additional Information

Please refer to the RTC registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

Please consult the Apollo4 Family Programmer's Guide for additional information about CLKGEN and RTC Module operations.

8. Security

8.1 Functional Overview

The Apollo4 Blue SoC supports the Arm Platform Security Architecture (PSA) and is Level 1 compliant. It provides robust system level security leveraging Ambiq's SecureSPOT™ technology.

The Apollo4 Blue SoC supports the following security features:

- Secure Boot
- Secure Over-the-Air (OTA) Updates
- Secure Wired Updates
- Secure Key Storage
- Secure Debug
- Key Revocation
- Crypto Acceleration
- True Random Number Generator (TRNG)
- CRC32
- External Storage Inline Encryption/Decryption

The following cryptographic features are supported:

- AES (128, 192, 256b)
  - ECB, CBC, CTR, OFB
  - CMAC, CBC-MAC, AES-CCM, AES_GCM
- AES Key Wrapping
- Diffie-Hellman (1024, 2048, 3072b)
  - ANSI X9.42-2003: Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography.
  - Public-Key Cryptography Standards (PKCS) #3: Diffie-Hellman Key Agreement Standard.
- ECC Key Generation (NIST and 25519 curves)
- ECIES
- ECDSA
- ECDH
- SHA1/SHA224/SHA256
- HKDF
- KDF
  - NIST SP 800-108: Recommendation for Key Derivation Using Pseudorandom Functions
- RSA PKCS#1 (2048, 3072, 4096b)
  - Public-Key Cryptography Standards (PKCS) #1 v2.1: RSA Cryptography Specifications
  - Public-Key Cryptography Standards (PKCS) #1 v1.5: RSA Encryption
- RSA Key Generation
- TRNG
  - BSI AIS-31: Functionality Classes and Evaluation Methodology for True Random Number Generators.
  - NIST SP 800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation.

The Apollo4 Blue SoC adheres to the Arm Platform Security Architecture (PSA). This establishes a secure processing environment that isolates security critical functionality and data from application software. The basis of the secure processing environment is a secure boot. This leverages the immutable Root-of-Trust (RoT) based on a set of hardware primitives which ensure trusted boot of the device. Maintaining the chain of trust is critical. Apollo4 provides robust security services to support Over-the-Air (OTA) updates, wired updates and secure debug sessions.

8.2 Secure Boot

The Secure Boot feature on the Apollo4 Blue SoC provides a secure foundation for customer firmware/ services. The secure boot loader provides authentication, decryption and integrity validation for all firmware upon installation and boot/reset. Secure boot loader provides firmware recovery and OTA update support.

Secure Boot policy can be used to direct the secure boot loader based on the customer security requirements.

A high level flow diagram of the Secure Boot process is illustrated in Figure 13. See separate Security document(s) for more details.

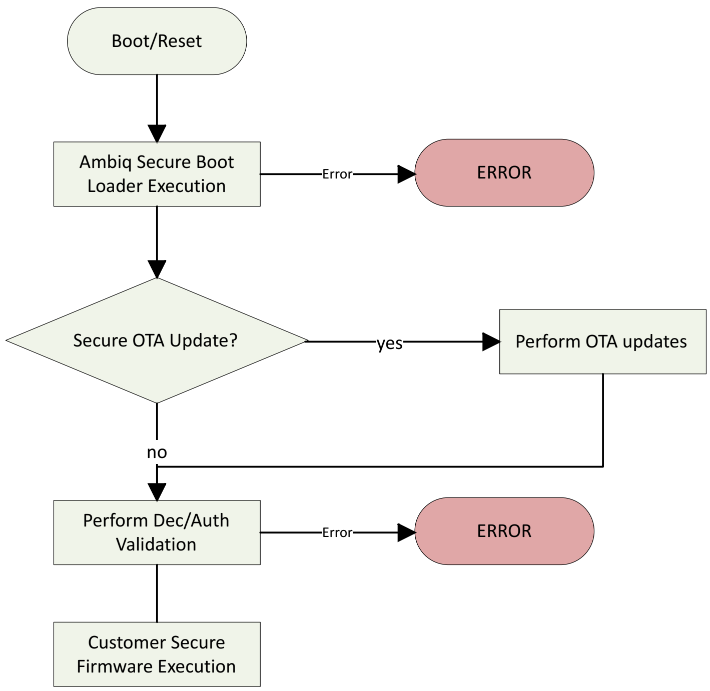


Figure 13. Secure Boot Flow

8.3 Secure OTA

The Apollo4 Blue SoC supports secure OTA leveraging the Ambiq secure boot loader. Customers can update any firmware component securely as directed via the security policy configuration in OTP.

The basic flow is shown in Figure 14.

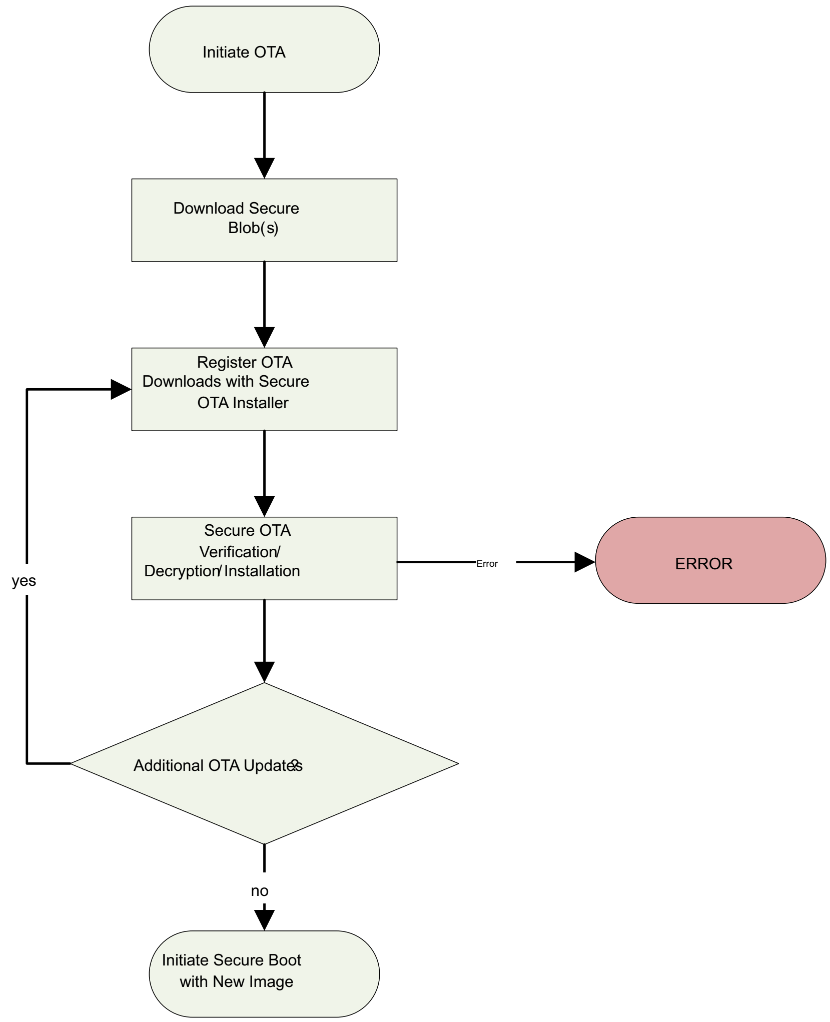


Figure 14. Secure OTA Flow

8.4 Secure Key Storage

Key material is managed by hardware and exposed to software via security APIs on the Apollo4 Blue SoC. The keys are stored securely in OTP memory and are never directly accessible to software. Certain key material is used/accessible only during certain life cycle state of the device. These are mainly used for provisioning of the device. See the Apollo4 Security Whitepaper for more details.

8.5 External Flash In-line Encrypt/Decrypt

External flash is supported on Apollo4 Blue SoC via the MSPI controller interface. The MSPI controller supports in-line encrypt/decrypt to enable customers to securely store firmware or any other secure image data in external flash without concern of the firmware/data confidentiality being compromised.

The Ambiq secure in-line encrypt/decrypt provides robust, high performance and extremely low power protection for external flash contents. Ambiq's in-line encrypt/decrypt enables truly in-line capability that does not degrade performance when asking external flash.

For more details on the in-line support, See "MSPI Master Module" on page 226.

8.6 Secure Life Cycle States

The Apollo4 Blue SoC supports the following life cycle states:

- Chip Manufacturer (CM)
- Device Manufacturer (DM)
- Secure
- RMA

The life cycles are managed by hardware and OTP. See the Apollo4 Security Whitepaper for more details.

8.7 Crypto Subsystem

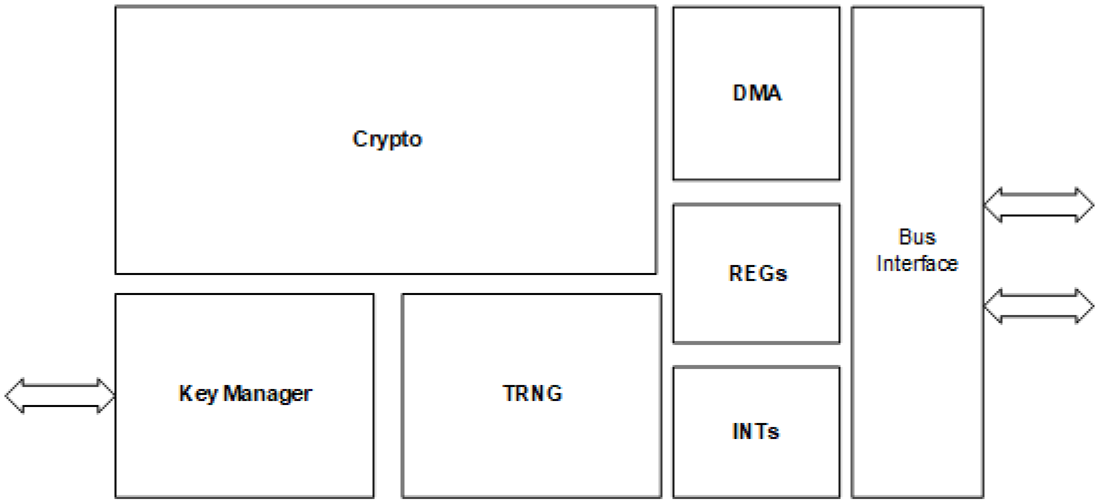


Figure 15. Crypto Subsystem

The crypto subsystem provides the following features:

- Cryptographic acceleration for the protection of data-in-transit and data-at-rest.
- Protection of various assets belonging to the chip manufacturer (ICV) or device manufacturer (OEM). Service operators provide services over the target device and the end product. These asset protection features include:
  - Image verification at boot/during runtime
  - Authenticated debug
  - Random number generation
  - Security life cycle state management
  - Asset Provisioning

The following standard specifications are supported:

- FIPS Publication 186-4: Digital Signature Standard (DSS), July 2013, compliant with sections 5.1, 6.2, 6.3, 6.4, B.1.2, B.2.2, B.3.6, B.4.2, C.3.1, C.3.3, C.3.5, C.9, and D.1.2.
- FIPS Publication 197: Advanced Encryption Standard, support only 128-bit and 256-bit keys.
- NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation: Methods and Techniques, compliant with sections 6.1, 6.2, 6.4, and 6.5.
- NIST SP 800-38B: Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication
- NIST SP 800-108: Recommendation for Key Derivation Using Pseudorandom Functions, compliant with section 5.1.
- Standards for Efficient Cryptography Group (SECG): SEC1 Elliptic Curve Cryptography, 2000, compliant with sections 2.1.1, 2.2.1, 3.1.1, 3.2, 3.3.1, 3.6.1, 4, and 6.1.

The crypto subsystem provides the cryptographic acceleration and isolation required to support the Apollo4 security model. These services are managed by software to support private and public-side cryptographic functions.

9. Bluetooth Low Energy Controller

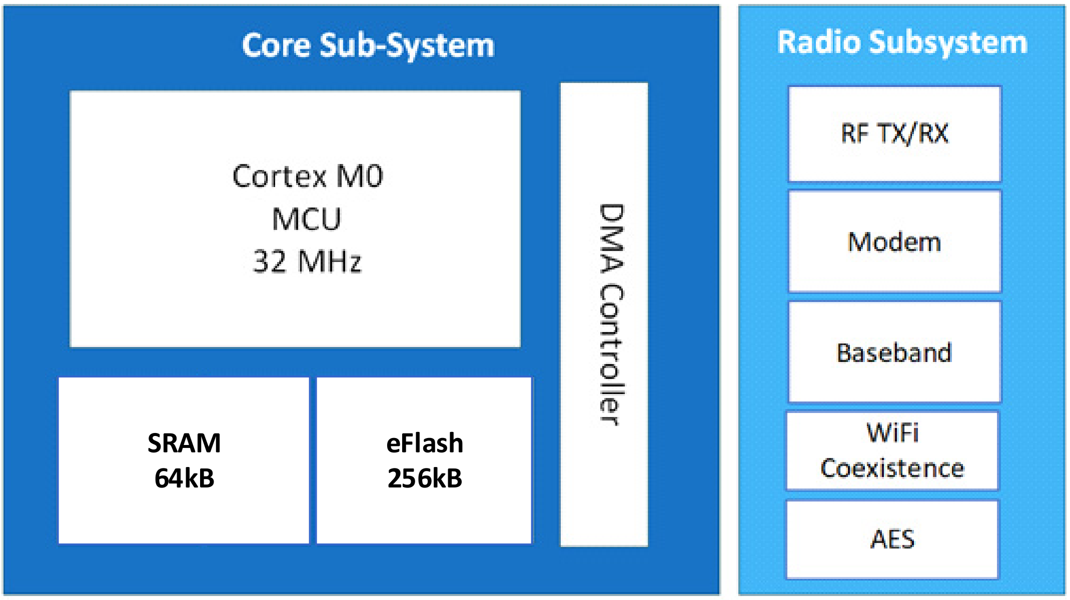


Figure 16. Apollo4 Blue SoC Bluetooth Low Energy Controller Block Diagram

incorporates a dedicated Arm Cortex-M0 processor, Bluetooth Low Energy baseband, modem and 2.4 GHz RF transceiver. It also provides corresponding PMU, clocking, I/O, 64 kB of SRAM and 256 kB of non-volatile memory.

Communication with the Bluetooth Low Energy Controller is supported through an internal high-speed SPI interface. Dedicated data movement hardware enables efficient interface for HCI packet transfers.

9.1 Feature Set

Energy-efficient Arm Cortex-M0 MCU

- 256 kB Flash including link layer stack
- 64 kB SRAM
- 32 MHz MCU and Flash speed

Bluetooth 5.1 Low Energy Technology

- Bluetooth Low Energy 5.1 Compliant Controller Subsystem
- High Data Rate (2 Mbps)
- Advertising extensions
- Coexistence
- Angle of Arrival/Departure Support
- Up to 10 simultaneous links supported
- AES-128 hardware acceleration

Integrated Power Management

- Buck regulator
- Retention LDO for low power sleep mode

High Performance RF

- -95.5 dBm Bluetooth Low Energy RX sensitivity @ 1 Mbps

- -92 dBm Bluetooth Low Energy RX sensitivity @ 2 Mbps
- -10 dBm to +6 dBm TX output power

9.2 Functional Overview

The block diagram of the Bluetooth Low Energy Module is as shown in Figure 16. The controller is designed to provide low power Bluetooth Low Energy 5.1 connectivity.

The controller can operate internally at different clock frequencies as required for the communication workload. The maximum clock input frequency supported is 32 MHz. The Bluetooth Low Energy system incorporates a PLL to generate the necessary clocking. The reference clock for the PLL can be sourced from either a dedicated external crystal or an external oscillator. Power regulation is supported internally via a buck DCDC regulator and supporting LDO regulators needed to generate all internal voltages for the radio and digital subsystems.

9.3 Clocking

The Apollo4 Blue has inputs for a 32 MHz crystal (HF_XTAL) for, among other uses, the Bluetooth Low Energy Controller's clock which uses the XO32M and XI32M pins. See the Pin List and Function Table in chapter 1. The clocking configuration must be set in the MCUCTRL_XTALHSCTRL register accordingly. At reset, the internal RC oscillators are enabled. Software must change the clock configuration as well as set the appropriate mux settings in the GPIOCONFIG register.

If using an externally-sourced clock for the Bluetooth Low Energy Controller, it must be set up for input on the XO32M pin as a single-ended signal.

The Bluetooth Low Energy Controller has an integrated PLL for generating the 2.4 GHz signal that serves as the modulated carrier during transmission, or as the local oscillator (LO) during reception.

9.4 Power Management

The Bluetooth Low Energy Controller has an integrated DCDC switching regulator as well as LDOs to provide all voltage rails for the Bluetooth Low Energy functionality. The 5 power states are as follows:

- Active
- Idle
- Standby
- Sleep
- Shutdown

9.5 Hardware Reference

9.5.1 Power Delivery

The integration diagram in Figure 17 gives a reference implementation for a buck enabled configuration.

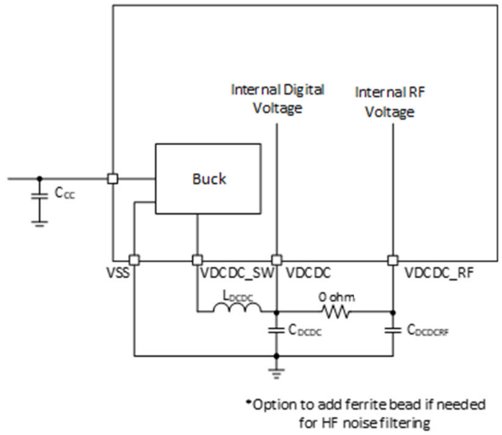


Figure 17. Integration Diagram for Buck Enabled Configuration

9.6 Antenna

The antenna filter shown in Figure 18 is recommended for the Bluetooth Low Energy Controller. It is intended to suppress higher harmonics and meet regulatory limits even at the highest output power levels offered by the Bluetooth Low Energy controller, while also ensuring good impedance matching with the antenna.

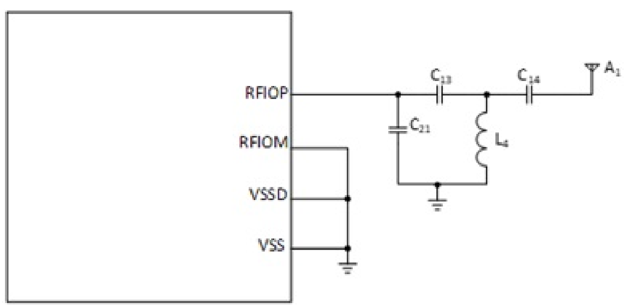


Figure 18. Recommended Antenna Filter

10. Counter/Timer Module (TIMER)

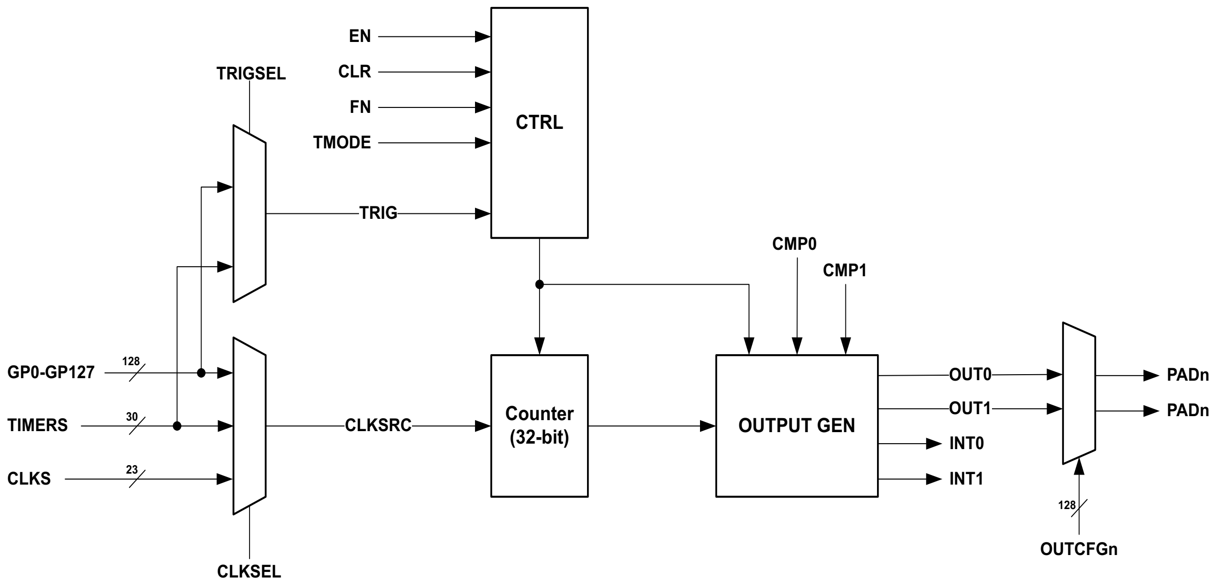


Figure 19. Block Diagram for One Counter/Timer

10.1 Functional Overview

The Apollo4 Blue SoC Timer/Counter module includes sixteen Timer/Counters, one of which is shown in Figure 19. This is in addition to a system timer as described in the System Timer chapter. Each Timer/ Counter includes a very low power asynchronous 32-bit counter. Each Timer/Counter has external pin connections using any GPIO pads as outputs for each of the two comparators. As well, any GPIO can be selected as the clock or trigger source for any of the timers.

NOTE
The Timer/Counter module no longer offers the HCLK_DIV4 as a timer clock option.

Please refer to the TIMER registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

The features of the Timer Module are as follows:

- Sixteen 32-bit binary counters used for simple waveform generation, timed interrupt sources, and counting applications.
- Six selectable timer modes - EDGE, UPCOUNT, PWM, DOWNCOUNT, SINGLEPATTERN and REPEATPATTERN.
- Each timer (TIMERn, where n = 0 to 15) has 2 interrupts, TMRn0INT and TMRn1INT.
- Each timer has two outputs (OUT0 and OUT1) which are controlled by the CMP0 and CMP1 registers based on timer mode and each can be inverted independently.
- Each timer can interface to any GPIO, allowing any GPIO to be driven by any timer output and any GPIO to be used as a timer's clock or trigger.
- All timers are fully independent but can be linked by clocking one timer from another's output.
- Clock sources include several sources from CLKGEN, another timer's output, or any GPIO input.
- Each timer supports an optional trigger condition which starts the timer.

- Counter value may be written directly; otherwise CTRLn_CLR bit initializes the counter for the selected mode.
- CTRLn_TMRnLMT field can be set to generate 1-255 repetitions of a waveform (0=unlimited).
- For most modes, timers are up-counters and CMP0 defines the end of a counter cycle; timer either stops or repeats. CMP1 is a secondary comparator.
- A down-counting mode, where TIMERn count = 0 defines the end of a counter cycle, allows software to update with a new count between interrupts (in repeat mode) enabling variable periods, or allows the counter to be reloaded automatically with a predefined start count.

NOTE
The CTIMER module used on Apollo3 Blue and earlier SoCs is not used on the Apollo4 Blue SoC. Although similar in design and use, the upgraded timer module on the Apollo4 Blue SoC has differences which should be understood if migrating from an earlier Apollo device.

10.2 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about TIMER Module operations.

11. System Timer (STIMER)

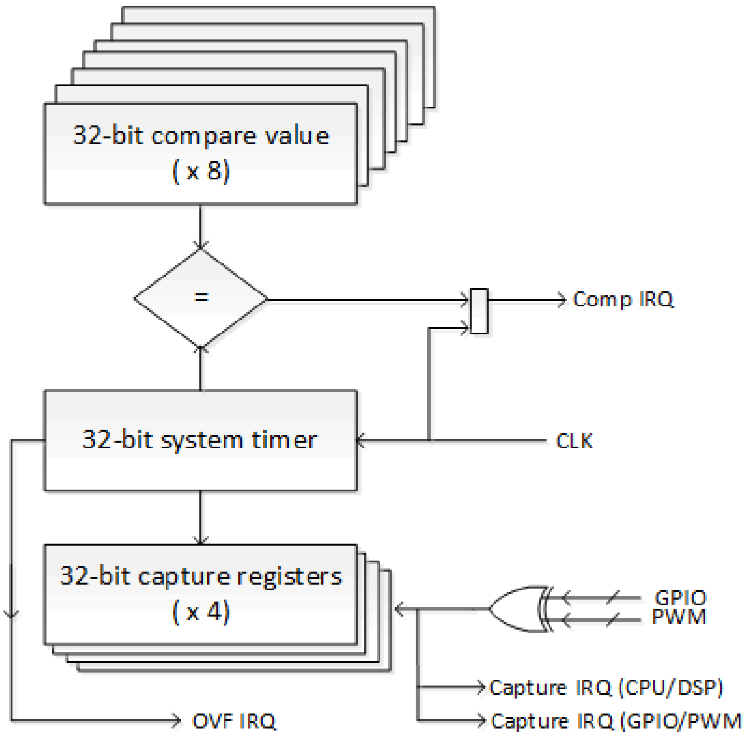


Figure 20. Block Diagram for the System Timer

11.1 Functional Overview

The Apollo4 Blue SoC System Timer (STIMER), shown above in Figure 20, tracks the global synchronized counter. It can be used for RTOS scheduling and real-time system tracking. This timer is provided in addition to the other timer peripherals to enable software/firmware to have a simple, globally synchronized timer source.

Please refer to the STIMER registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

The System Timer (STIMER) Module provides real time measurement for all task scheduling, sensor sample rate calibration, and tracking of real time and calendar maintenance. Key features are:

- 32-bit binary counter used for RTOS scheduling decisions.
- Eight 32-bit compare and interrupt registers to facilitate light weight scheduling (designs without RTOS).
- Accurate scheduling of comparator interrupts
- Only offsets from “NOW” are written to comparator registers.
- Maintains real time epoch for applications.
- Overflow interrupt to allow firmware to keep the extended part (more than 32-bits) of real time epoch.
- Time stamping hardware for multiple sensor streams (4 capture registers).
- Firmware handling of odd calculations such as Leap Second. It also handles things like surprise/legislated changes to the daylight savings time transition dates.
- Firmware handling of 1024 versus 1000 scaling of real time conversions.

- Only reset by POA (Power On Analog - system cold reset) so that it retains time across all POI and POR (system warm reset) events except full power cycles.
- Contains three 32-bit NVRAM registers that are only reset by POA to maintain real time offset from epoch.
- Programmable external GPIO trigger and/or PWM trigger on capture (required for sensor synchronization)

The heart of the STIMER is a single 32-bit counter that keeps track of current time for the application running on the Apollo4 Blue SoC. This counter is reset at the actual power cycle reset of the SoC. It is generally never reset or changed again. Up to eight 32-bit comparator registers can be loaded each of which can generate an interrupt signal to the NVIC. Comparators A through H generate interrupt A through H while capture registers A through D and the overflow event generate interrupt I, all the way to the NVIC. Thus the scheduler can run these 9 interrupts at different priorities in the NVIC.

The comparator interrupts are each used to schedule a function (task) to run for the application. Thus these tasks run on interrupt levels at priorities lower than the I/O interrupts.

The overflow interrupt allows firmware to keep track of real time beyond that maintained in the 32-bit timer.

11.2 Additional Information

Please consult the Apollo4 Family Programmer’s Guide for additional information about CLKGEN and RTC Module operations.

12. Watchdog Timer (WDT)

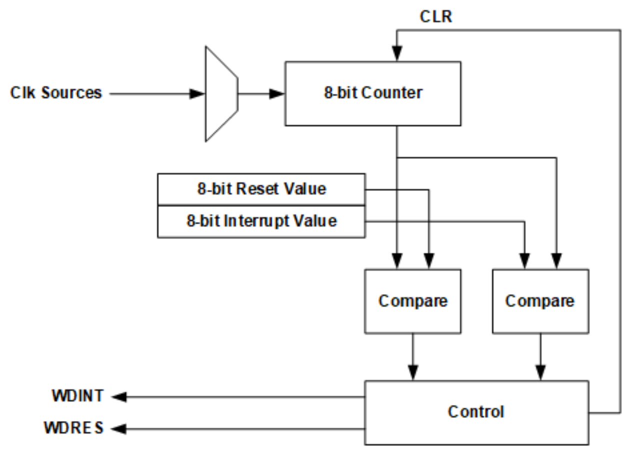


Figure 21. Block Diagram for the Watchdog Timer Module

12.1 Functional Overview

The Watchdog Timer (WDT), shown in Figure 21, is used to ensure that software is operational, by resetting the Apollo4 Blue SoC if the WDT reaches a configurable value before being cleared by software. The WDT can be clocked by one of four selectable prescalers of the always active low-power LFRC clock, but is nominally clocked at 128 Hz. The WDT may be locked to ensure that software cannot disable its functionality, in which case the WDTCFG register cannot be accidentally reprogrammed. An interrupt can also be generated at a different counter value to implement an early warning function.

NOTE
The RESEN bit in the WDT_CFG register must be set and the WDREN bit in the RSTGEN_CFG register must be set to enable a watchdog timer reset condition.

Please refer to the WDT registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

12.2 Additional Information

Please consult the Apollo4 Blue SoC Programmer's Guide for additional information about WDT Module operations.

13. General Purpose Input/Output (GPIO)

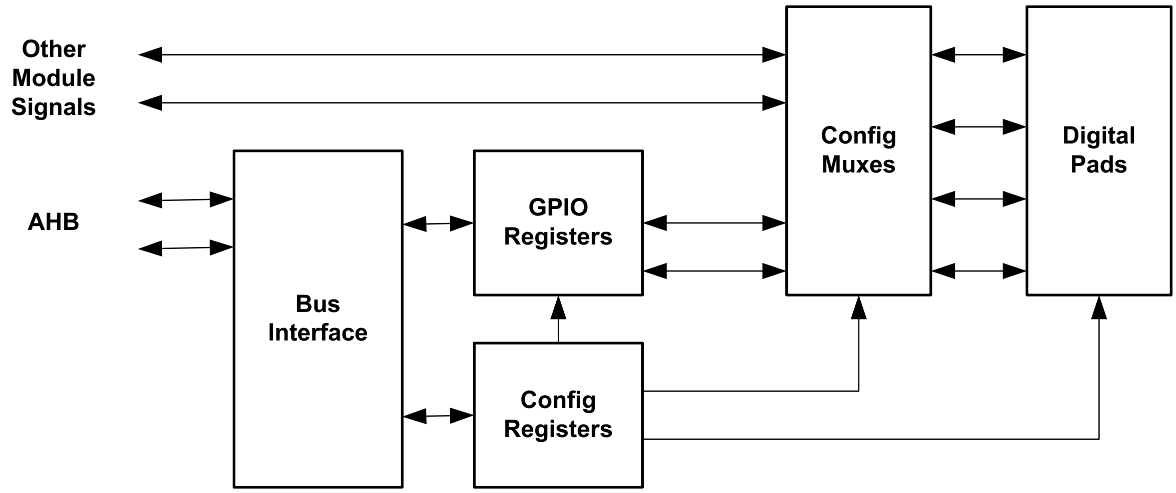


Figure 22. Block diagram for the General Purpose I/O (GPIO) Module

13.1 Functional Overview

The General Purpose I/O and Pad Configuration (GPIO) Module, shown in Figure 22, controls connections to up to 105 digital/analog pads[1]. Each pad may be connected to a variety of module interface signals, with all pad input and output selection and control managed by the GPIO module. In addition, any pad may function as a general purpose input and/or output pad which may be configured for a variety of external functions. Each GPIO may be configured to generate an interrupt when a transition occurs on the input. In addition, any GPIO pad brought out to an external pin may be configured as any available chip enable for any IOM or MSPI instance, or the Display Controller.Please refer to the GPIO and FPIO registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

> NOTE
>
> Once the PADKEY is written, it should be explicitly cleared (with a non-key value) after GPIO configuration register updates are complete.

13.2 Pad Configuration Functions

Each GPIO on the Apollo4 Blue SoC can be configured as one of several functions according to the Pin Mapping tables starting with Table 5.

The REG_GPIO_PINCFGn (n = 0 to 104) registers are used to control the function of each pad. Note that the REG_GPIO_PADKEY Register must be set to the value 0x73 in order to write the PINCFGn registers. The REG_GPIO_PINCFGn_FNCSELn (n = 0 to 104) field selects one of up to twelve signals to be used for each pad. Functions are grouped by module per the color coding shown in Table 7. There are several

1. GPIO105-127 are reserved and unavailable for use. Not all 105 GPIO are available on all members of the Apollo4 family.

special pad types which are used upon selection of specific pad functions, and these Special Pad Types are defined in Table 8.

Note that the column labeled “SIP PKG” indicates which GPIO pins come out on the package used by the Apollo4 Blue SoC. Any GPIO without an “X” in this column is not brought out to a pin and might be being used for dedicated, internal functionality, e.g., Bluetooth Low Energy Controller interface. Function selections should not be used or set for these GPIO on this package unless explicitly instructed to do so.

NOTE

Although there are 128 GPIO_PINCFGn registers present on the chip, the number of pads brought out to pins is limited by the package used. For any pad not brought out to an external pin, it is advised not to change the default settings for any of its configuration registers, as it may result in a state which has not been validated, and unintended device operation may occur.

The REG_GPIO_PINCFGn_DSn field controls the drive strength of each pad. A drive strength of either 0.1 of full strength (0P1X) or half strength (0P5X) may be selected for any GPIO pin, and 3/4 full strength (0P75X) and full strength (1P0X) are additionally offered on select GPIO pins.

For all pads associated with external pins (non-virtual), the REG_GPIO_PINCFG_PULLCFGn field enables a weak pull-down (50K) or a selection of 6 values of pull-up resistor settings - 1.5K, 6K, 12K, 24K, 50K or 100K.

The I/O voltage source reference for each pad as listed in the right-most column of Table 5 and Table 6 equates to the corresponding voltage supply shown in Table 4.

Table 4: I/O Pin Voltage Source

I/O Reference	Voltage Supply
0	VDDAUDD
1	VDDH
2	VDDH2

NOTE

Use of the DPI-2 interface, which includes pad functions DISP_D0 - DISP_D23, DISP_VS, DISP_HS, DISP_DE, DISP_PCLK, DISP_SD and DISP_CM, is not recommended or supported.

Table 5: Apollo4 Blue SoC Pin Mapping (Pg 1)

Pad	PADnFNCSEL											SIP PKG	I/O Voltage
	0	1	2	3	4	5	6	7	9	10	11		
0	SWTRACECLK	SLSCL	SLSCK	GPIO00	UART0TX	UART1TX	CT0	NCE0	VCMPO	-	FPIO00	X	1
1	SWTRACE0	SLSDAWIR3	SLMOSI	GPIO01	UART2TX	UART3TX	CT1	NCE1	VCMPO	-	FPIO01	X	1
2	SWTRACE1	SLMISO	TRIG1	GPIO02	UART0RX	UART1RX	CT2	NCE2	VCMPO	-	FPIO02	X	1
3	SWTRACE2	SLnCE	SWO	GPIO03	UART2RX	UART3RX	CT3	NCE3	-	-	FPIO03	X	1
4	SWTRACE3	SLINT	32KHzXT	GPIO04	UART0RTS	UART1RTS	CT4	NCE4	I2S0_SDIN	I2S1_SDIN	FPIO04	X	1
5	M0SCL	M0SCK	I2S0_CLK	GPIO05	UART2RTS	UART3RTS	CT5	NCE5	-	I2S1_CLK	FPIO05	X	1
6	M0SDAWIR3	M0MOSI	I2S0_DATA	GPIO06	UART0CTS	UART1CTS	CT6	NCE6	I2S0_SDOUT	I2S1_SDOUT	FPIO06	X	1
7	M0MISO	TRIG0	I2S0_WS	GPIO07	UART2CTS	UART3CTS	CT7	NCE7	-	I2S1_WS	FPIO07	X	1
8	CMPRF1	TRIG1	-	GPIO08	M1SCL	M1SCK	CT8	NCE8	-	-	FPIO08	X	2
9	CMPRF0	TRIG2	-	GPIO09	M1SDAWIR3	M1MOSI	CT9	NCE9	-	-	FPIO09	X	2
10	CMPIN0	TRIG3	-	GPIO10	M1MISO	-	CT10	NCE10	DISP_TE	-	FPIO10	X	2
11	CMPIN1	TRIG0	I2S0_CLK	GPIO11	UART2RX	UART3RX	CT11	NCE11	-	-	FPIO11	X	2
12	ADCSE7	TRIG1	I2S0_DATA	GPIO12	UART0TX	UART1TX	CT12	NCE12	CMPRF2	I2S0_SDOUT	FPIO12	X	2
13	ADCSE6	TRIG2	I2S0_WS	GPIO13	UART2TX	UART3TX	CT13	NCE13	-	-	FPIO13	X	2
14	ADCSE5	TRIG3	-	GPIO14	-	UART1RX	CT14	NCE14	-	I2S0_SDIN	FPIO14	X	2
15	ADCSE4	TRIG0	-	GPIO15	-	UART3RX	CT15	NCE15	-	REFCLK_EXT	FPIO15	X	2
16	ADCSE3	TRIG1	I2S1_CLK	GPIO16	-	UART1RTS	CT16	NCE16	-	-	FPIO16	X	2
17	ADCSE2	TRIG2	I2S1_DATA	GPIO17	-	UART3RTS	CT17	NCE17	I2S1_SDOUT	-	FPIO17	X	2
18	ADCSE1	-	I2S1_WS	GPIO18	UART0CTS	UART1CTS	CT18	NCE18	-	-	FPIO18	X	2
19	ADCSE0	-	-	GPIO19	UART2CTS	UART3CTS	CT19	NCE19	I2S1_SDIN	-	FPIO19	X	2
20	SWDCK	TRIG1	-	GPIO20	UART0TX	UART1TX	CT20	NCE20	-	-	FPIO20	X	1
21	SWDIO	TRIG2	-	GPIO21	UART2TX	UART3TX	CT21	NCE21	-	-	FPIO21	X	1
22	M7SCL	M7SCK	SWO	GPIO22	UART0RX	UART1RX	CT22	NCE22	VCMPO	-	FPIO22	X	2
23	M7SDAWIR3	M7MOSI	SWO	GPIO23	UART2RX	UART3RX	CT23	NCE23	VCMPO	-	FPIO23	X	2
24	M7MISO	TRIG3	SWO	GPIO24	UART0RTS	UART1RTS	CT24	NCE24	-	-	FPIO24	X	2
25	M2SCL	M2SCK	-	GPIO25	-	-	CT25	NCE25	-	-	FPIO25	X	1
26	M2SDAWIR3	M2MOSI	-	GPIO26	-	-	CT26	NCE26	VCMPO	-	FPIO26	X	1
27	M2MISO	TRIG0	-	GPIO27	-	-	CT27	NCE27	I2S0_SDIN	-	FPIO27	X	1
28	SWO	VCMPO	I2S0_CLK	GPIO28	UART2CTS	-	CT28	NCE28	-	-	FPIO28	X	1
29	TRIG0	VCMPO	I2S0_DATA	GPIO29	UART1CTS	-	CT29	NCE29	I2S0_SDOUT	-	FPIO29	X	1
30	TRIG1	VCMPO	I2S0_WS	GPIO30	UART0TX	-	CT30	NCE30	-	-	FPIO30	X	1
31	M3SCL	M3SCK	-	GPIO31	UART2TX	-	CT31	NCE31	VCMPO	-	FPIO31	X	1
32	M3SDAWIR3	M3MOSI	-	GPIO32	UART0RX	-	CT32	NCE32	-	-	FPIO32	X	1
33	M3MISO	CLKOUT	-	GPIO33	UART2RX	-	CT33	NCE33	DISP_TE	-	FPIO33	X	1
34	M4SCL	M4SCK	SWO	GPIO34	UART0TX	-	CT34	NCE34	VCMPO	-	FPIO34		1
35	M4SDAWIR3	M4MOSI	SWO	GPIO35	UART2TX	UART3TX	CT35	NCE35	VCMPO	-	FPIO35		1
36	M4MISO	TRIG0	SWO	GPIO36	UART0RX	UART1RX	CT36	NCE36	-	-	FPIO36		1
37	MSPI1_0	TRIG1	32KHzXT	GPIO37	UART2RX	DISP_D15	CT37	NCE37	-	-	FPIO37	X	1
38	MSPI1_1	TRIG2	SWTRACECLK	GPIO38	UART0RTS	DISP_D16	CT38	NCE38	-	-	FPIO38	X	1
39	MSPI1_2	TRIG3	SWTRACE0	GPIO39	UART2RTS	DISP_D17	CT39	NCE39	-	-	FPIO39		1
40	MSPI1_3	TRIG1	SWTRACE1	GPIO40	UART0CTS	DISP_D18	CT40	NCE40	-	-	FPIO40		1
41	MSPI1_4	TRIG0	SWTRACE2	GPIO41	UART0TX	DISP_D19	CT41	NCE41	SWO	-	FPIO41		1
42	MSPI1_5	TRIG2	SWTRACE3	GPIO42	UART2TX	DISP_D20	CT42	NCE42	-	-	FPIO42		1
43	MSPI1_6	TRIG3	SWTRACECTL	GPIO43	UART0RX	DISP_D21	CT43	NCE43	-	-	FPIO43		1
44	MSPI1_7	TRIG1	SWO	GPIO44	UART2RX	DISP_D22	CT44	NCE44	VCMPO	-	FPIO44		1
45	MSPI1_8	TRIG2	32KHzXT	GPIO45	UART0TX	DISP_D23	CT45	NCE45	-	-	FPIO45		1
46	MSPI1_9	TRIG3	CLKOUT_32M	GPIO46	UART2TX	UART3TX	CT46	NCE46	I2S1_SDIN	I2S0_SDIN	FPIO46		1
47	M5SCL	M5SCK	I2S1_CLK	GPIO47	UART0RX	UART1RX	CT47	NCE47	-	I2S0_CLK	FPIO47	X	1
48	M5SDAWIR3	M5MOSI	I2S1_DATA	GPIO48	UART2RX	UART3RX	CT48	NCE48	I2S1_SDOUT	I2S0_SDOUT	FPIO48	X	1
49	M5MISO	TRIG0	I2S1_WS	GPIO49	UART0RTS	UART1RTS	CT49	NCE49	-	I2S0_WS	FPIO49	X	1
50	PDM0_CLK	TRIG0	SWTRACECLK	GPIO50	UART2RTS	UART3RTS	CT50	NCE50	DISP_TE	-	FPIO50	X	0
51	PDM0_DATA	TRIG1	SWTRACE0	GPIO51	UART0CTS	UART1CTS	CT51	NCE51	-	-	FPIO51	X	0
52	PDM1_CLK	TRIG2	SWTRACE1	GPIO52	UART2CTS	UART3CTS	CT52	NCE52	VCMPO	-	FPIO52	X	0

Table 6: Apollo4 Blue SoC Pin Mapping (Pg 2)

Pad	PADnFNCSEL											SIP PKG	I/O Voltage
	0	1	2	3	4	5	6	7	9	10	11		
53	PDM1_DATA	TRIG3	SWTRACE2	GPIO53	UART0TX	UART1TX	CT53	NCE53	-	-	FPIO53	X	0
54	PDM2_CLK	TRIG0	SWTRACE3	GPIO54	UART2TX	UART3TX	CT54	NCE54	-	-	FPIO54	X	0
55	PDM2_DATA	TRIG1	SWTRACECTL	GPIO55	UART0RX	UART1RX	CT55	NCE55	-	-	FPIO55	X	0
56	PDM3_CLK	TRIG2	SWO	GPIO56	UART2RX	UART3RX	CT56	NCE56	-	-	FPIO56	X	0
57	PDM3_DATA	TRIG3	SWO	GPIO57	UART0RTS	UART1RTS	CT57	NCE57	VCMPO	-	FPIO57	X	0
58	-	-	-	GPIO58	UART0RTS	UART3RTS	CT58	NCE58	-	-	FPIO58	X	1
59	-	TRIG0	-	GPIO59	UART0CTS	UART1CTS	CT59	NCE59	-	-	FPIO59	X	1
60	-	TRIG1	-	GPIO60	UART0TX	UART3CTS	CT60	NCE60	-	-	FPIO60	X	1
61	M6SCL	M6SCK	I2S1_CLK	GPIO61	UART2TX	UART3TX	CT61	NCE61	-	-	FPIO61	X	1
62	M6SDAWIR3	M6MOSI	I2S1_DATA	GPIO62	UART0RX	UART1RX	CT62	NCE62	I2S1_SDOUT	-	FPIO62	X	1
63	M6MISO	CLKOUT	I2S1_WS	GPIO63	UART2RX	UART3RX	CT63	NCE63	DISP_TE	-	FPIO63	X	1
64	MSPI0_0	32KHzXT	SWO	GPIO64	UART0RTS	DISP_D0	CT64	NCE64	I2S1_SDIN	-	FPIO64	X	1
65	MSPI0_1	32KHzXT	SWO	GPIO65	UART0CTS	DISP_D1	CT65	NCE65	-	-	FPIO65	X	1
66	MSPI0_2	CLKOUT	SWO	GPIO66	UART0TX	DISP_D2	CT66	NCE66	-	-	FPIO66	X	1
67	MSPI0_3	CLKOUT	SWO	GPIO67	UART2TX	DISP_D3	CT67	NCE67	-	-	FPIO67	X	1
68	MSPI0_4	SWO	-	GPIO68	UART0RX	DISP_D4	CT68	NCE68	-	-	FPIO68	X	1
69	MSPI0_5	32KHzXT	SWO	GPIO69	UART2RX	DISP_D5	CT69	NCE69	-	-	FPIO69	X	1
70	MSPI0_6	32KHzXT	SWTRACE0	GPIO70	UART0RTS	DISP_D6	CT70	NCE70	-	-	FPIO70	X	1
71	MSPI0_7	CLKOUT	SWTRACE1	GPIO71	UART0CTS	DISP_D7	CT71	NCE71	-	-	FPIO71	X	1
72	MSPI0_8	CLKOUT	SWTRACE2	GPIO72	UART0TX	DISP_D8	CT72	NCE72	VCMPO	-	FPIO72	X	1
73	MSPI0_9	-	SWTRACE3	GPIO73	UART2TX	DISP_D9	CT73	NCE73	-	-	FPIO73	X	1
74	MSPI2_0	DISP_QSPI_D0_OUT	DISP_QSPI_D0	GPIO74	UART0RX	DISP_D10	CT74	NCE74	DISP_SPI_SD	DISP_SPI_SDO	FPIO74	X	1
75	MSPI2_1	32KHzXT	DISP_QSPI_D1	GPIO75	UART2RX	DISP_D11	CT75	NCE75	DISP_SPI_DCX	-	FPIO75	X	1
76	MSPI2_2	32KHzXT	DISP_QSPI_D2	GPIO76	UART0RTS	DISP_D12	CT76	NCE76	-	-	FPIO76	X	1
77	MSPI2_3	-	DISP_QSPI_D3	GPIO77	UART0CTS	DISP_D13	CT77	NCE77	-	-	FPIO77	X	1
78	MSPI2_4	-	DISP_QSPI_SCK	GPIO78	UART0TX	DISP_D14	CT78	NCE78	DISP_SPI_SCK	-	FPIO78	X	1
79	MSPI2_5	-	SDIF_DAT4	GPIO79	SWO	DISP_VS	CT79	NCE79	DISP_SPI_SDI	-	FPIO79	X	1
80	MSPI2_6	CLKOUT	SDIF_DAT5	GPIO80	SWTRACE0	DISP_HS	CT80	NCE80	-	-	FPIO80	X	1
81	MSPI2_7	CLKOUT	SDIF_DAT6	GPIO81	SWTRACE1	DISP_DE	CT81	NCE81	-	-	FPIO81	X	1
82	MSPI2_8	32KHzXT	SDIF_DAT7	GPIO82	SWTRACE2	DISP_PCLK	CT82	NCE82	-	-	FPIO82	X	1
83	MSPI2_9	32KHzXT	SDIF_CMD	GPIO83	SWTRACE3	DISP_SD	CT83	NCE83	-	-	FPIO83	X	1
84	-	-	SDIF_DAT0	GPIO84	-	-	CT84	NCE84	-	-	FPIO84	X	1
85	-	-	SDIF_DAT1	GPIO85	-	-	CT85	NCE85	-	-	FPIO85	X	1
86	-	-	SDIF_DAT2	GPIO86	-	-	CT86	NCE86	-	-	FPIO86	X	1
87	-	-	SDIF_DAT3	GPIO87	-	-	CT87	NCE87	DISP_TE	-	FPIO87	X	1
88	-	-	SDIF_CLKOUT	GPIO88	-	-	CT88	NCE88	-	-	FPIO88	X	1
89	-	-	-	GPIO89	-	DISP_CM	CT89	NCE89	-	-	FPIO89	X	1
90	-	-	-	GPIO90	-	-	CT90	NCE90	VCMPO	-	FPIO90	X	1
91	-	-	-	GPIO91	-	-	CT91	NCE91	VCMPO	-	FPIO91	X	1
92	-	-	-	GPIO92	-	-	CT92	NCE92	VCMPO	-	FPIO92		1
93	-	-	-	GPIO93	-	-	CT93	NCE93	VCMPO	-	FPIO93		1
94	-	-	-	GPIO94	-	-	CT94	NCE94	VCMPO	-	FPIO94		1
95	-	-	-	GPIO95	-	-	CT95	NCE95	-	-	FPIO95		1
96	-	-	-	GPIO96	-	-	CT96	NCE96	-	-	FPIO96		1
97	-	-	-	GPIO97	-	-	CT97	NCE97	-	-	FPIO97		1
98	-	-	-	GPIO98	-	-	CT98	NCE98	-	-	FPIO98		1
99	-	-	-	GPIO99	-	-	CT99	NCE99	-	-	FPIO99		1
100	-	-	-	GPIO100	-	-	CT100	NCE100	-	-	FPIO100		1
101	-	-	-	GPIO101	-	-	CT101	NCE101	-	-	FPIO101		1
102	-	-	-	GPIO102	-	-	CT102	NCE102	-	-	FPIO102		1
103	-	-	-	GPIO103	-	-	CT103	NCE103	-	-	FPIO103		1
104	-	-	-	GPIO104	-	-	CT104	NCE104	-	-	FPIO104		1

Table 7: Pad Function Color Code

Color/Symbol	Module
Analog	Analog Modules (ADC, VCOMP)
CLKOUT	Clock output
Debug	Debug/Special
DISPLAY	Display
Global IOM	Shared IOM/MSPI
GPIO	GPIO
I2S0	I2S 0
I2S1	I2S 1
IOM0	IO Master 0
IOM1	IO Master 1
IOM2	IO Master 2
IOM3	IO Master 3
IOM4	IO Master 4
IOM5	IO Master 5
IOM6	IO Master 6
IOM7	IO Master 7
IOS	IO Slave
MIPI	MIPI
MSPI0	MSPI0
MSPI1	MSPI1
MSPI2	MSPI2
PDM0	PDM 0
PDM1	PDM 1
PDM2	PDM 2
PDM3	PDM 3
SDIO	SDIO
TCT	Counter/Timers
UART0	UART 0
UART1	UART 1
UART2	UART 2
UART3	UART 3

NOTE

Not all derivatives and packages include all of the module instances shown in the Pad Function Color Table.

Table 8: Special Pad Types

GPIO Pad Number	Function Select Number	Pad Function Name	Functional Interface	Description	Pin Type
1	1	SLSDAWIR3	IO Slave I2C	I2C Slave I/O data (I2C) 3 Wire Data (SPI)	Bidirectional Open Drain
5	0	M0SCL	IO Master 0 I2C	I2C Master 0 clock	Open Drain Output
6	0	M0SDAWIR3	IO Master 0 I2C	I2C Master 0 I/O data (I2C) 3 Wire data (SPI)	Bidirectional Open Drain
8	4	M1SCL	IO Master 1 I2C	I2C Master 1 clock	Open Drain Output
9	4	M1SDAWIR3	IO Master 1 I2C	I2C Master 1 I/O data (I2C) 3 Wire data (SPI)	Bidirectional Open Drain
21	0	SWDIO	Debug	Software data I/O	Bidirectional 3-state
22	0	M7SCL	IO Master 7 I2C	I2C Master 7 Clk	Bidirectional Open Drain
23	0	M7SDAWIR3	IO Master 7 I2C	I2C Master 7 I/O data (I2C) 3 Wire data (SPI)	Bidirectional Open Drain
25	0	M2SCL	IO Master 2 I2C	I2C Master 2 clock	Open Drain Output
26	0	M2SDAWIR3	IO Master 2 I2C	I2C Master 2 I/O data (I2C) 3 Wire data (SPI)	Bidirectional Open Drain
31	0	M3SCL	IO Master 3 I2C	I2C Master 3 clock	Open Drain Output
32	0	M3SDAWIR3	IO Master 3 I2C	I2C Master 3 I/O data (I2C) 3 Wire data (SPI)	Bidirectional Open Drain
35	0	M4SDAWIR3	IO Master 4 I2C	I2C Master 4 I/O data (I2C) 3 Wire data (SPI)	Bidirectional Open Drain
47	0	M5SCL	IO Master 5 I2C	I2C Master 5 Clk	Bidirectional Open Drain
48	0	M5SDAWIR3	IO Master 5 I2C	I2C Master 5 I/O data (I2C) 3 Wire data (SPI)	Bidirectional Open Drain
61	0	M6SCL	IO Master 6 I2C	I2C Master 6 Clk	Bidirectional Open Drain
62	0	M6SDAWIR3	IO Master 6 I2C	I2C Master 6 I/O data (I2C) 3 Wire data (SPI)	Bidirectional Open Drain

NOTE

GPIO35 is not brought out to an external pin and is therefore not offered as a special pad type on the Apollo4 Blue package.

13.3 Fast GPIO (FPIO)

Access to GPIO pin registers on the Apollo4 Blue SoC can be multiple CPU cycles to complete. To support certain functions that require shorter latency access, a Fast GPIO (FPIO) interface is supported. The Fast GPIO is accessed via the FPIO registers.

NOTE

Retention of FPIO output pin state is not guaranteed through deep sleep.

Please refer to the FPIO registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

13.4 Additional Information

Please consult the Apollo4 Family Programmer’s Guide for additional information about GPIO Module operations.

14. General Purpose ADC and Temperature Sensor Module

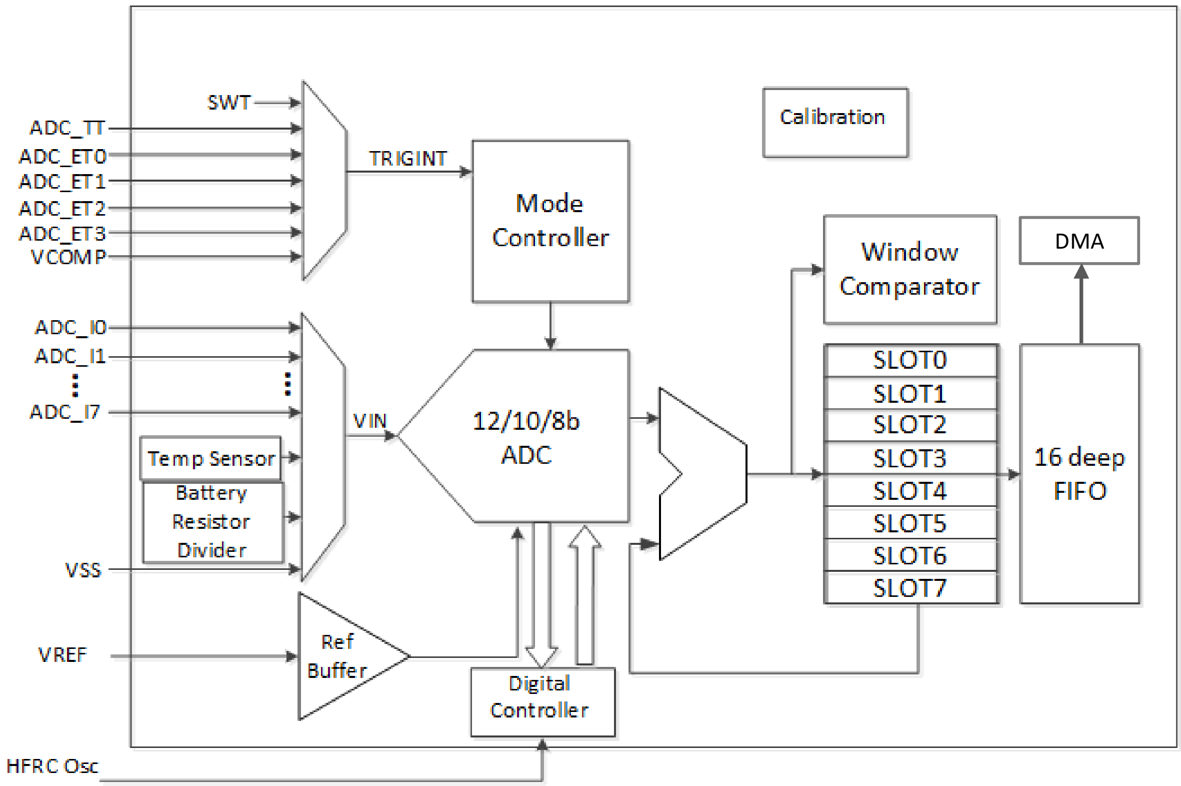


Figure 23. Block Diagram for ADC and Temperature Sensor

14.1 Features

The general purpose Analog-to-Digital Converter (ADC) and Temperature Sensor Module includes a single-ended 12-bit multi-channel Successive Approximation Register (SAR) ADC as shown in Figure 23.

Key features include:

- 11 user-selectable channels with sources including:
  - External pins
    - 8 single ended external pins
  - Internal voltage (VSS)
  - Voltage divider (battery)
  - Temperature sensor
- Configurable automatic low power control between scans
- Optional Battery load enable for voltage divider measurement
- Single shot, repeating single shot, scan, and repeating scan modes
- Variable sample tracking time, configurable on per-slot basis
- User-selectable clock source for variable sampling rates
- Automatically accumulate and scale module for hardware averaging of samples
- A 16-entry FIFO and DMA capability for storing measurement results and maximizing MCU sleep time
- Window comparator for monitoring voltages excursions into or out of user-selectable thresholds

- Support for up to 2.8 MS/s effective continuous, multi-slot sampling rate (8-bit mode) - see note in Functional Overview section below
- Interrupts for FIFO full, FIFO almost full, Scan Complete, Conversion Complete, Window Incursion Window Excursion, and various DMA-related notifications

Please refer to the ADC registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

14.2 Functional Overview

The Apollo4 Blue SoC integrates a sophisticated 12-bit successive approximation Analog to Digital Converter (ADC) block for sensing both internal and external voltages. The block provides eight separately managed conversion requests, called slots which are serially sequenced. The result of each conversion requests is delivered to a 16 deep FIFO. Firmware can utilize various interrupt notifications to determine when to collect the sampled data from the FIFO or from a buffer written by DMA. This block is extremely effective at automatically managing its power states and its clock sources.

When the ADC block is enabled and has an active scan in progress, it requests a clock source. There is an automatic hardware hand shake between the clock generator and the ADC. If the ADC is the only block requesting an HFRC based clock, then the HFRC will be automatically started. The ADC can be configured to completely power down the HFRC between scans if the startup latency is acceptable or it can leave the HFRC powered on between scans if the application requires low latency between successive conversions.

The ADC on all Apollo4 family SoCs offers four options for the reference clock source and frequency via the ADC's CFG_CLKSEL field:

1. HFRC at 48 MHz (default setting)
2. Inverted HFRC at 48 MHz
3. HFRC at 24 MHz
4. HFRC2 at 48 MHz

Sampling rate is calculated by dividing the ADC clock rate by the number of output data latency cycles, or FCLK / LCR. The output data latency, LCR, consists of the number of sampling cycles, NTRACK, plus a number of base latency cycles such that LCR = NTRACK + NBASE. NTRACK can be any number of sampling cycles from 5 to 69 cycles, while NBASE varies depending on the precision mode as: 19 cycles for 12-bit mode, 15 cycles for 10-bit mode, and 12 cycles for 8-bit mode.

> NOTE
>
> Due to two chip errata, ERR091 and ERR113, only the 24 MHz HFRC setting (CFG_CLKSEL = 0x2) is supported. The other three clock options should not be used. In addition, at least 37 sampling/tracking cycles (SLnCFG_TRKCYCn = 0x20) must be used to prevent the conversion data corruption described in the ERR113. With the above settings, the maximum sampling rate achievable in 8-bit precision mode is: 24 MHz / (37 cycles + 12 cycles) = 490 KS/s.

14.3 Voltage Reference Source

The Apollo4 Blue SoC ADC supports one internal reference source to be used for the analog to digital conversion step. The reference voltage is 1.19 V and is not user settable. ADC input voltages > 1.19 V exceed the ADC range and return full scale code, but will not damage ADC inputs.

14.4 Voltage Divider and Switchable Battery Load

The Apollo4 Blue SoC's ADC includes a switchable voltage divider that enables the ADC to measure the input voltage to the VDD rail. In most systems this will be the battery voltage applied to the SoC. The voltage divider is only switched on when one of the active slots is selecting analog mux channel 15. That is only when the mode controller is ultimately triggered and powers up the ADC block for a conversion scan of all active slots. Otherwise, the voltage divider is turned off.

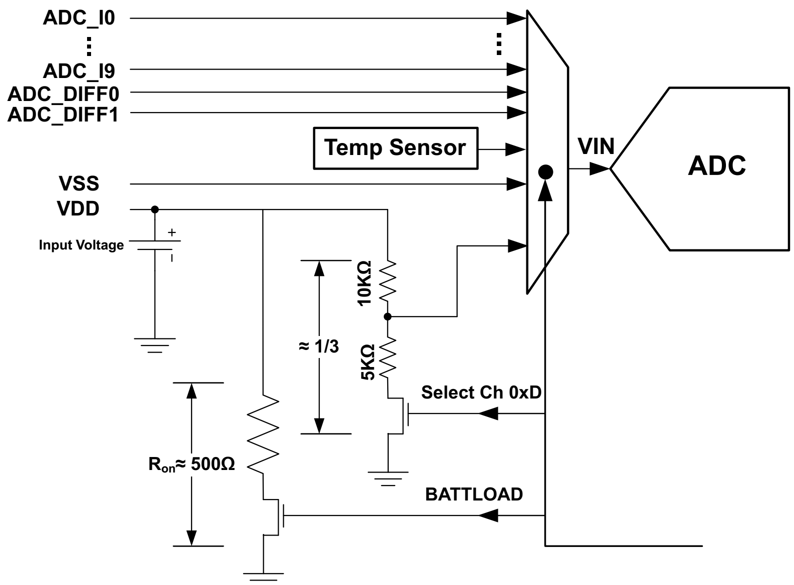


Figure 24. Switchable Battery Load

The switchable load resistor is enabled by the BATTLOAD bit as shown in the ADCBATTLOAD Register of the MCUCTRL Registers.

This feature is used to help estimate the health of the battery chemistry by estimating the internal resistance of the battery.

14.5 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about ADC Module operations, where the following topics are covered:

- Clock Source and Dividers
- Channel Analog Mux
- Triggering and Trigger Sources
- Voltage Reference Sources
- Eight Automatically Managed Conversion Slots
- Automatic Sample Accumulation and Scaling
- Sixteen Entry Result FIFO
- Window Comparator
- Operating Modes and the Mode Controller
- Interrupts

15. Multi-bit Serial Peripheral Interface (MSPI)

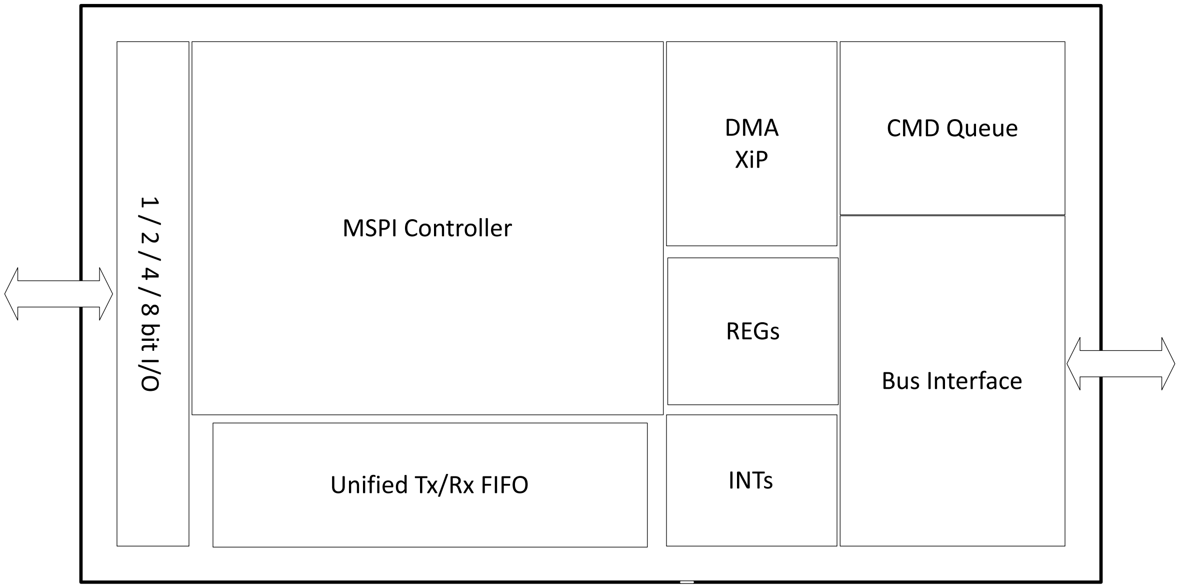


Figure 25. Block Diagram for the MSPI Master Module

15.1 Features

- 1/2/4/8-bit SPI interface
- Support for DCX signal for displays
- XiP supported
- DMA with peripheral-to-memory and peripheral-to-peripheral support
- Command Queue Support
- Up to 96 MHz clock in SDR mode; up to 48 MHz clock in DDR mode
- All four SPI CPOL/CPHA modes

15.2 Functional Overview

The Apollo4 Blue SoC includes two Multi-bit SPI (MSPI) modules, MSPI0 and MSPI2, which can be used to connect to external memory devices or displays. Each MSPI supports operation up to 96 MHz, depending on the MSPI mode or instance, and can transfer in serial, dual, quad, and octal data widths.

NOTE

- DQS mode is not supported on either MSPI instance.
- The maximum clock rate for MSPI0 for all data widths except octal is 96 MHz for SDR and 48 MHz for DDR; for octal it is 48 MHz for SDR and DDR.
- MSPI1 is not pinned out on the Apollo4 Blue.
- The maximum clock rate for MSPI2 for all data widths is 24 MHz for SDR and 12 MHz for DDR.

15.3 MSPI Transfers

The MSPI module has a unified 16-entry FIFO (32 bits wide) that is used for both transmit and receive data. To ensure that transactions are not dropped because of system or software latency, the MSPI controller will pause the clock (and thus the transfer on the bus) if the TX FIFO empties or the RX FIFO fills during an operation. It will automatically resume once the FIFO condition has cleared.

Please refer to the MSPI registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

MSPIn transfers generally consist of transmitting a 1 byte instruction, a 1-4 byte address (optional), and 1 byte to 64KB of write or read data (with an optional number of turnaround clock cycles between address and RX data, as well as an optional number of turnaround clock cycles between address and RX data). Most devices use the same number of pins to transmit instruction, address, and data (for example, all are quad or all are serial). However, some devices utilize mixed transfer modes to implement parallel data transfer on top of an inherently serial command structure. These devices are supported by the MSPIn by utilizing the XIPMIXEDn configuration, which forces the MSPIn to switch into dual or quad modes of operation for a portion of the transfer.

To utilize mixed mode transfers, the MSPI's normal configuration should be set to match the device's transfer characteristics for commands (usually serial), which allows the MSPIn to communicate with the device in its native mode. The XIPMIXED0 field in the DEV0XIP register should then be programmed to indicate whether the data phase (and optionally address phase) of the command should be performed in dual or quad mode. The MSPIn will automatically switch to the new mode after transmitting the command to the device for all DMA and XIP operations.

The two MSPI modules on the Apollo4 Blue SoC are directly attached to the system AXI bus and memory mapped (referred to as XIP for eXecute In Place) at address 0x14000000 (MSPI0) and 0x1C000000 (MSPI2).

Access to the MSPIn devices is as follows:

- MCU instruction accesses to XIP space are read-only and handled through the cache (which must be configured in a 64KB cache mode).
- MCU data accesses to XIP space are read/write and handled through the DAXI module (Data-AXI interface on the MCU). The DAXI incorporates write buffering and some caching to improve performance to MSPIn and SSRAM targets.
- PIO: The MCU can initiate PIO-based operations to manage basic device configuration and other low-level manual operations.
- DMA: MSPIn module can autonomously transfer data between the external device and internal memory or NVM.

Note that XIP and DMA do not enforce hardware coherency, so the cache should be disabled or invalidated when performing DMA or XIP operations to regions that contain code that may be cached. In each of these modes, the MSPIn module also supports data scrambling on accesses within a programmable address range having boundaries aligned to 64K address boundaries.

Once the external devices are configured, the MSPIn supports a simple DMA model, where software can program the internal (SRAM or NVM) address and external device address, transfer direction, and transfer size. Once enabled, the MSPIn DMA interface will move data between the system and external flash and interrupt when complete. The MSPIn also supports a higher-level command queuing (CQ) protocol, where software can construct a buffer of operations in SRAM (or internal NVM memory) and the MSPIn will execute the series of operations autonomously. The MSPIn can also power itself down at the end of DMA or CQ operations.

While each MSPIn module can be used as a generic SPI device (with either of its two chip enables), in addition to supporting Serial, Dual, and Quad displays, it is primarily designed to support serial NAND/NOR flash memory or PSRAM memory. It is intended to be used to initialize the external memory devices and then be configured with the parameters matching the flash access characteristics. Devices can then be accessed through DMA or XIP operations with minimal software overhead.

The DMA address range has been expanded to support the larger flash and SRAM sizes, and the MSPIn DMA/transfer length has been expanded to 24 bits to allow burst transactions of more than 64 kB.

The MSPIn module also contains:

- A DEV0BOUNDARY register which can be programmed to break a single long MSPIn DMA into smaller transfers at periodic intervals (DMATIMELIMIT0 bit field).
- Address boundaries (DMABOUND0 bit field) to provide breaks in DMA for XIP traffic and satisfy the page crossing and maximum refresh times of external PSRAM devices.

NOTE

The DMATIMELIMIT0 is approximate since the MSPI will continue transmitting to the next 32-bit word boundary before disengaging on the bus. For this reason, a device requiring an 8 µs maximum transmission time should be set to have about a 7.5 µs time limit.

NOTE

For DMABOUND0 to properly break at a page crossing, the DMADEVADDR for the transfer must be 4-byte aligned. If a non-aligned starting edge of the transfer is required, software should manually break the transaction into two parts, with the first transaction ending on the page boundary. Failure to observe this limitation will result in data loss as the MSPIn may write 1-3 additional bytes past the boundary which will either wrap within the device's page or be discarded by the device.

15.4 Pad Configuration and Enables

For the Apollo4 Blue SoC, the two supported MSPI modules support serial, dual, quad or octal mode and support the following external connections. The columns to the right indicate which bits are used in each configuration (S=serial, D=dual, Q=quad, O=octal with CE#). Within the table, O=output pin, I=input pin, and X=bidirectional.

Table 9: MSPI0 Pin Muxing (Serial, Dual, Quad, Octal)

Pin Name	Direction	GPIO	Description	S0	S1	D0	D1	Q0	Q1	O0	O1
MSPI0.0	Output	0-91	MSPI0 CE0	O		O		O		O	
MSPI0.1	Output		MSPI0 CE1		O		O		O		O
MSPI0_9	Input/Output	73	MSPI0 DM0/DQS0 (Octal)							X	X
MSPI0_8	Output	72	MSPI0 CLK	O	O	O	O	O	O	O	O
MSPI0_7	Input/Output	71	MSPI0 Data Bit 7							X	X
MSPI0_6	Input/Output	70	MSPI0 Data Bit 6							X	X
MSPI0_5	Input/Output	69	MSPI0 Data Bit 5							X	X
MSPI0_4	Input/Output	68	MSPI0 Data Bit 4							X	X
MSPI0_3	Input/Output	67	MSPI0 Data Bit 3					X	X	X	X
MSPI0_2	Input/Output	66	MSPI0 Data Bit 2					X	X	X	X
MSPI0_1	Input/Output	65	MSPI0 Data Bit 1	I	I	X	X	X	X	X	X

Pin Name	Direction	GPIO	Description	S0	S1	D0	D1	Q0	Q1	O0	O1
MSPI0_0	Input/Output	64	MSPI0 Data Bit 0	O	O	X	X	X	X	X	X

NOTE

MSPI1 is not available on the Apollo4 Blue SoC.

Table 10: MSPI2 Pin Muxing (Serial, Dual, Quad, Octal)

Pin Name	Direction	GPIO	Description	S0	S1	D0	D1	Q0	Q1	O0	O1
MSPI2.0	Output	0-91	MSPI2 CE0	O		O		O		O	
MSPI2.1	Output		MSPI2 CE1		O		O		O		O
MSPI2_9	Input/Output	83	MSPI2 DM/DQS							X	X
MSPI2_8	Output	82	MSPI2 CLK	O	O	O	O	O	O	O	O
MSPI2_7	Input/Output	81	MSPI2 Data Bit 7							X	X
MSPI2_6	Input/Output	80	MSPI2 Data Bit 6							X	X
MSPI2_5	Input/Output	79	MSPI2 Data Bit 5							X	X
MSPI2_4	Input/Output	78	MSPI2 Data Bit 4							X	X
MSPI2_3	Input/Output	77	MSPI2 Data Bit 3					X	X	X	X
MSPI2_2	Input/Output	76	MSPI2 Data Bit 2					X	X	X	X
MSPI2_1	Input/Output	75	MSPI2 Data Bit 1	I	I	X	X	X	X	X	X
MSPI2_0	Input/Output	74	MSPI2 Data Bit 0	O	O	X	X	X	X	X	X

The PADOUTEN register should be programmed to enable the proper pins for the selected mode.

NOTE

Due to timing issues, MSPIn_4 cannot be used as the clock line for any MSPI instance.

NOTE

For all MSPI instances, all MSPI interface pins other than the chip enable pin should be configured to have a drive strength setting of 1P0X.

Typically, most serial SPI devices use a separate MOSI and MISO when operating in serial mode. The SEPIO0 bit in the DEV0CFG register should be set when software needs to read data from devices in serial mode, since it redirects the MISO input from pin 1 down to input data pin 0 of the MSPI's RX logic.

Table 11 below shows the required field configurations for typical MSPI operating modes.

Table 11: Required Settings for Typical Configurations

Mode (Data Lines and CE)					DEV0CFG_ DEVCFG0	DEV0CFG_ SEPIO0	DEV0XIP_ XIPMIXED0	PADOUTEN_ OUTEN
Instruction	Address	Data	Separate IO	Chip Enable (CE)				
Serial	Serial	Serial	Yes	0	SERIAL0 (1)	1	NORMAL (0)	0x103
Serial	Serial	Serial	Yes	1	SERIAL1 (2)	1	NORMAL (0)	0x103
Serial	Serial	Serial	No	0	SERIAL0 (1)	0	NORMAL (0)	0x101
Serial	Serial	Serial	No	1	SERIAL1 (2)	0	NORMAL (0)	0x101
Serial	Serial	Dual	No	0	SERIAL0 (1)	0	D2 (1)	0x103
Serial	Serial	Dual	No	1	SERIAL1 (2)	0	D2 (1)	0x103
Serial	Dual	Dual	No	0	SERIAL0 (1)	0	AD2 (3)	0x103
Serial	Dual	Dual	No	1	SERIAL1 (2)	0	AD2 (3)	0x103
Serial	Serial	Quad	No	0	SERIAL0 (1)	0	D4 (5)	0x10F
Serial	Serial	Quad	No	1	SERIAL1 (2)	0	D4 (5)	0x10F
Serial	Quad	Quad	No	0	SERIAL0 (1)	0	AD4 (7)	0x10F
Serial	Quad	Quad	No	1	SERIAL1 (2)	0	AD4 (7)	0x10F
Dual	Dual	Dual	No	0	DUAL0 (5)	0	NORMAL (0)	0x103
Dual	Dual	Dual	No	1	DUAL1 (6)	0	NORMAL (0)	0x103
Quad	Quad	Quad	No	0	QUAD0 (9)	0	NORMAL (0)	0x10F
Quad	Quad	Quad	No	1	QUAD1 (0xA)	0	NORMAL (0)	0x1F0
Octal	Octal	Octal	No	0	OCTAL0 (0xD)	0	NORMAL (0)	0x3FF
Octal	Octal	Octal	No	1	OCTAL1 (0xE)	0	NORMAL (0)	0x3FF

15.5 Additional Information

Please consult the Apollo4 Family Programmer’s Guide for additional information about MSPI Module operations.

16. I^{2}C/SPI Master (IOM)

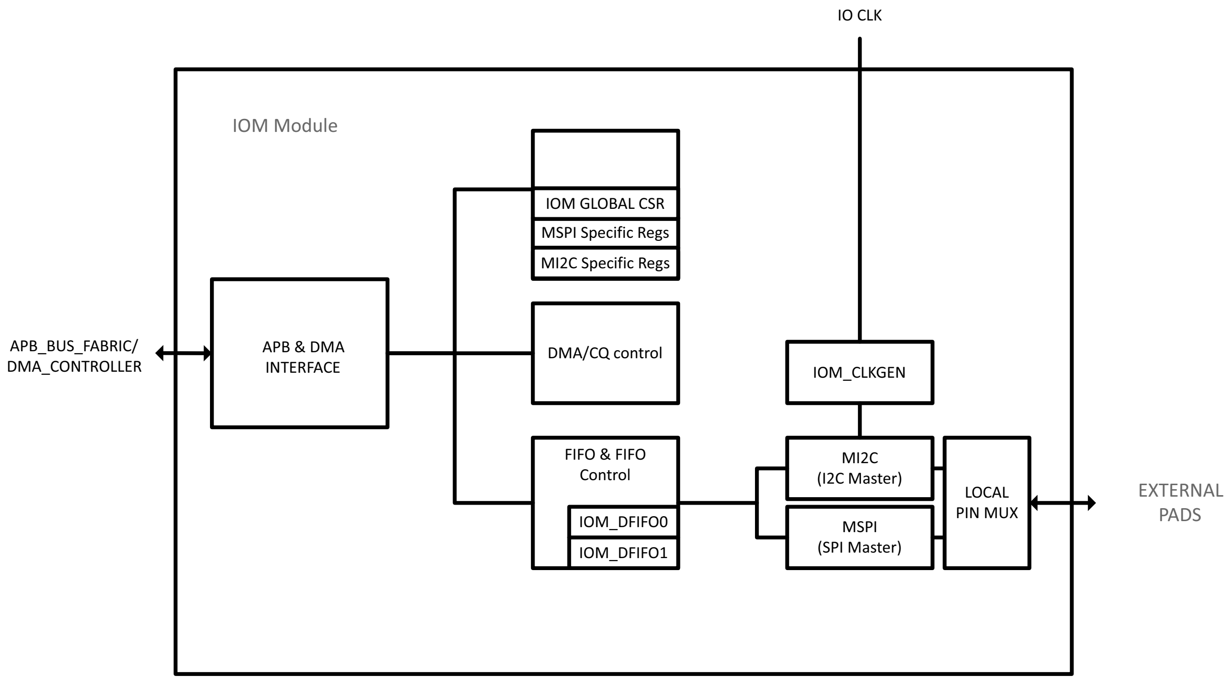


Figure 26. Block Diagram for the I^{2}C/SPI Master Module

16.1 Features

No resources are shared between IOM modules, but within a single IOM module, the submodules share a common set of FIFO and command resources.

16.1.1 Features common to all submodules

- 2 Independent 32-byte FIFOs, one dedicated each direction of data transfer
- Direct access of all FIFO data from MCU interface, including non-destructive reads.
- FIFO mode read/write access (push/pop mechanism)
- Direct command, direct data mode. (Command and data written to/read from the module registers directly)
- Direct command, DMA data mode. Commands are written directly to the module, but data is written to/read from the main SRAM array.
- Command queuing operations. Register write operations are read from main SRAM memory and fed to the register unit in series.
- Programmable interrupts
- Programmable threshold interrupt level
- Configurable clock selection
- Read data synchronized internally for MCU access
- Ability to send multi-byte offset addresses, with single command
- Ability to view FIFO data without causing pop operation
- Capability to store data for multiple commands in either FIFO
- Programmable number of byte offsets of 0-3

16.1.2 I²C Master features

- Support for standard mode (100 kHz), Fast mode (400 kHz), and Fast mode+ (1 MHz)
- Support for 7-bit and 10-bit addressing modes
- Transfer burst sizes of 0 to 512 bytes.
- Configurable LSB or MSB data transfer.
- Clock stretching support.

16.1.3 SPI Master features

- Support for transaction sizes up to 4095 bytes
- Programmable number of byte offsets of 0-3
- Programmable operation in all polarity modes
- 3-wire and 4-wire read and write support
- Flow control for reads or writes, based on MISO (write flow control), or external, selectable PIO.
- Full duplex operation

16.2 Functional Overview

The Apollo4 Blue SoC includes seven I²C/SPI high-speed Master Modules (IOM), shown in Figure 26, each of which functions as the master of an I²C or SPI interface as selected by the IOMn_SUBMODCTRL_SMODnEN bits. A 64-byte bidirectional FIFO and a sophisticated Command mechanism allow simple initiation of I/O operations without requiring software interaction.

In I²C mode the I²C/SPI Master supports 7- and 10-bit addressing, multi-master arbitration, interface frequencies from 1.2 kHz to 1.0 MHz and up to 512-byte burst operations. In SPI mode the I²C/SPI Master supports up to 4 slaves with automatic nCE selection, 3- and 4-wire implementation, all SPI polarity/phase combinations and up to 4095-byte burst operations, with both standard embedded address operations and raw read/write transfers. Interface timing limits are as specified in the Serial Peripheral Interface (SPI) Master Interface table of the Electrical Characteristics chapter.

The active interface is selected by enabling the module enable bit (SMODnEN) for the interface in the IOMn_SUBMODCTL register. Only one interface can be active at a time.

Each module contains a separate pair of 32-byte FIFOs, each of which is dedicated to data flow in a single direction (input or output). The modules support data transfer to or from the module through either direct or DMA paths. SRAM can be used as the source or the sink of data, and storage data can be used as source data for IOM transaction. Command Queue operations are also supported to allow commands to be placed in memory and fetched and executed in series. The Command Queue interface also includes inter-module flags which allows event communication between other IOM modules, MSPI modules and external pins through the GPIO interface.

Also supported in the design are test modes for use in setup and power measurements, and debug facilities to aid in software/hardware debug.

Please refer to the IO Master registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

16.3 Power Control

The 7 IOM modules must be enabled in the PWRCTRL_DEVPWREN register prior to access and operation. The power status of the IOM modules can be read in the PWRCTRL_DEVPWRSTATUS register. Note that the IOM modules are separated into 2 power domains, referred to as HCPB and HCPC.

IOM modules 0, 1, 2 and 3 are contained in HCPB, while IOM modules 5, 6 and 7 are contained in HCPC power domain. When one IOM is powered on, all other IOMs in the same group are powered on as well.

16.4 Clocking and Resets

The IOM design uses 2 main clocks, APB_CLK and IO_CLK. The APB_CLK is used for all register and DMA accesses. It runs at 96 MHz and is interfaced via the APB fabric synchronous interface. The IO_CLK is used as the source of the interface clock and has selectable frequencies. The overview of the clocking structure is shown below:

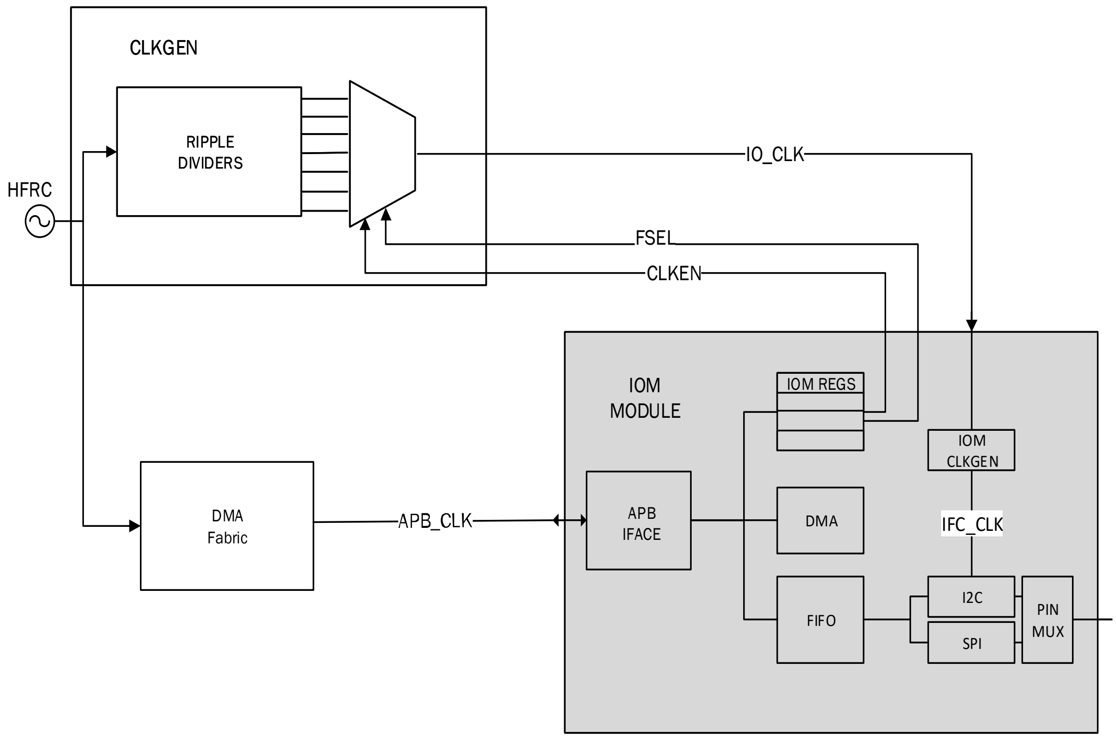


Figure 27. Clocking Structure for IOM Module

The APB_CLK is an internal clock sourced from the bus fabric and operates at a fixed 96 MHz frequency. It is used for internal communication and is heavily clock gated to reduce dynamic power.
The IO_CLK is generated within the central clocking module and enabled through the IOMn_CLKCFGIOCLKEN field. This clock must be enabled by software prior to module operation. The primary frequency of the IO_CLK is selected via the IOMn_CLKCFGFSEL field, and further divided by either or both of the internal divide by 3 divider (enabled via the IOMn_CLKCFGDIV3 field), or a programmable divider (enabled by IOMn_CLKCFGDIVEN and division set by IOMn_CLKCFGTOTPER and IOMn_IOMCLKCFGLOWPER fields) as shown below.

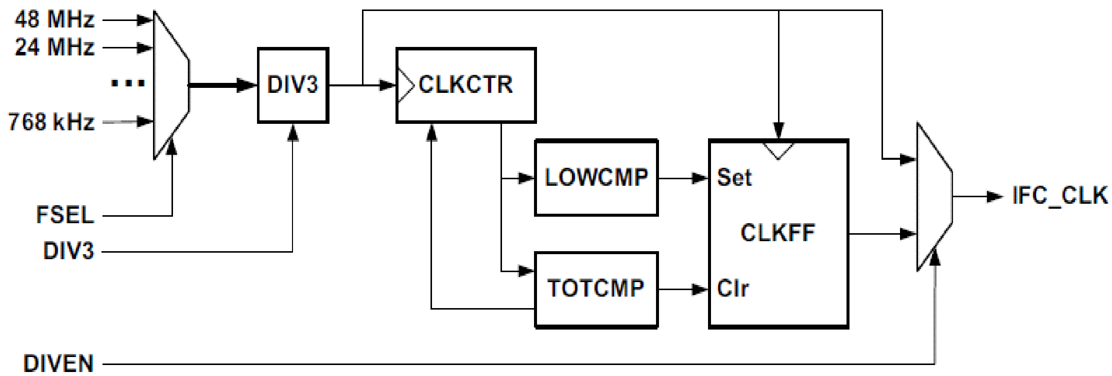


Figure 28. IO_CLK Generation

The divided by 3 divider is optional and will provide a 50% duty cycle divided by 3 clock. This divider is bypassed when the DIV3 field is set to 0.

The output of the DIV3 module is then fed to the programmable divider. This divider can be bypassed or enabled via the DIVEN field in the CLKCFG. It will divide at a rate of TOTPER+1 (subtract 1 from actual value when writing TOTPER field), and will toggle at LOWPER+1 clock count of the base IO_CLK from the DIV3 module. This will generate the final IO_CLK used by the interface module.

The IO_CLK is used for the reference clock for the internal module state machine, and for the external output clock. The use in both areas is heavily gated and can also be overridden by setting register IOMn_IOMDBG.IOCLKON field to 1.

16.5 I^{2}C Clock Generation

The I^{2}C output clock (SCL) is derived from dividing the final IO_CLK by 2. For example, for 1 MHz I^{2}C operation, an IO_CLK frequency of 2 MHz is required. Because the state machine will operate at 2x the target frequency of the interface frequency, the nominal output clk (SCL) duty cycle will be 50%, regardless of the duty cycle of the IO_CLK. However, the timing specification of some I^{2}C modes require an asymmetrical duty cycle on the SCL output, with the high period of the clock less than the low period of the clock. The clocking module allows a programmable delay prior to propagating the rising edge of the SCL output. This delay is in units of the source IO_CLK period (prior to any enabled division). This delay is specified in the IOMn_MI2CCFGSCLENDLY register field. The recommended settings for this register for each mode are detailed below.

If clock stretching is done by the slave devices attached to the IOM interface, further restrictions must be observed during the setup of the clock controls. This is due to the possible clock stretch event done within a single cycle on the I^{2}C SCL. In this case, the minimum SCL high time must be maintained, regardless of the time the slave releases the SCL. To detect the event within the single I^{2}C cycle, the SCL signal needs to be sub-sampled. The source IO_CLK is used for this purpose also and allows for sampling of the SCL signal by a programmable number of source IO_CLK cycles. The sample granularity is determined by the ratio of the source IO_CLK to final IO_CLK frequency and must allow for synchronization time between the two domains. The recommended settings for each mode are below. Only speeds of 100 kHz, 400 kHz and 1 MHz are supported. Contact Ambiq for use of other frequencies.

Mode	FSEL	DIV3	DIV EN	TOT PER	LOW PER	SMP CNT	SDAEN DLY	SCLEN DLY
Standard Mode (100 kHz)	2	0	1	119	59	3	15	0
Fast Mode (400 kHz)	2	0	1	29	14	3	15	2
Fast Mode+ (1000 kHz)	3	0	1	6	3	33	3	0

Table 12: Recommended Mode Settings for Standard I^{2}C Clock Speeds

The full selection table is as follows:

Mode	FSEL	DIV3	DIV EN	TOT PER	LOW PER	SMP CNT	SDAEN DLY	SCLEN DLY	Notes
Standard (100 kHz)	1	0	1	243	159	12	15	0	Effective Freq 100 kHz
	2	0	1	121	79	6	15	0	Effective Freq 100 kHz
	3	0	1	60	39	3	15	0	Effective Freq 100 kHz
	4	0	1	30	19	1	15	0	Effective Freq 93.7 kHz
	5	0	1	16	9	1	6	0	Effective Freq 93.7 kHz, Low power
Fast Mode (400 kHz)	1	0	1	62	39	7	15	4	Effective Freq 400 kHz
	2	0	1	31	19	15	15	2	Effective Freq 400 kHz
	3	0	1	15	9	2	7	1	Effective Freq 375 kHz
	4	0	1	7	3	1	3	0	Effective Freq 375 kHz
	5	0	1	5	3	1	3	0	Effective Freq 375 kHz, Low power
Fast+ Mode (1 MHz)	1	0	1	24	12	1	7	0	Effective Freq 1 MHz
	2	0	1	12	6	1	6	0	Effective Freq 1 MHz
	3	0	1	6	3	1	3	0	Effective Freq 1 MHz

Table 13: Full Mode Settings for I^{2}C Clock Speeds

16.5.1 SPI Clock Generation

The final IO_CLK is used directly as the SPI clock output. No additional settings are needed.

16.6 FIFO

The IOM module contains 2 uni-directional FIFOs, each 32 bytes wide. These FIFOs are used only for data storage during IO transactions. The FIFO supports both single (half duplex) and duplex modes of operation.

During direct mode data transfer operations, IO data transfer between the IOM module and the MCU is done by accessing the IOMn_FIFOPOP and IOMn_FIFOPUSH registers. These registers allow read

(FIFOPOP) and write (FIFOPUSH) of data into and out of the FIFO, and automatic adjustment of pointers used by the submodules. Only word accesses are permitted to these registers and any unused byte locations will be ignored or filled with zero. If DMA is enabled during the IO command operation, data will automatically be read or written into the FIFO from the DMA address and the pointers updated. The FIFO pointers and data are NOT reset after each command, and care must be taken to not leave any extra data in the FIFO, as this will be used for subsequent transfers. If needed, there is a manual reset of the FIFO pointers that can be done using the IOMn_FIFOCTRLFIFORSTN field. Additional information on data alignment is covered in the later sections of this document.

The submodules will prevent overruns or underruns from the FIFO by pausing the active transaction, usually by stopping the output clock. Once data is available (write operations) or there is room in the FIFO (read operations), the transaction will continue.

For debug operations, the IOM module also allows direct access to the FIFO contents through the IOMn_FIFO aperture. Access via this path does not affect the pointers used by the submodules and cannot be used to send or receive data as part of the IO operation. The FIFO aperture allows read and write operations into the write FIFO and read access into the read FIFO. The current FIFO pointers are readable via the FIFOLOC register. For the write FIFO, this will point to the next location to be written, while the read FIFO pointer will indicate the next location to be read.

NOTE
When DMA operations are in progress, the FIFOPUSH and FIFOPOP registers should not be accessed, as this will interfere with the DMA data.

16.7 Data Alignment

All data accesses between the MCU and the IOM interface are word aligned. Since the transfer size is specified in bytes, unused bytes within the word will either be discarded (for write operations) or filled with zero (read operations) to align to the next word boundary. DMA operations support a byte starting address, and the programmed DMA address does not have to be word aligned. Direct mode write operations will start transferring the least significant byte of the word (little endian style) at the current write FIFO pointer. If any remaining bytes are unused in a word at the end of the write operation, they will be discarded, and the write pointer will be set to the next word location. Direct mode read operations will store the first received byte into the least significant byte of location specified by the read FIFO pointer, and will fill any unused byte locations with zero if the transaction size is not a word multiple. The FIFO read pointer will point to the next FIFO location in the read FIFO, which will be word aligned.

16.7.1 Direct Mode Data Transfers

Direct mode data is enabled when DMA is disabled via the IOMn_DMACFGDMAEN and the data transfer size (TSIZE) is greater than 0. In this mode, the MCU transfers data via direct writes or reads to registers in the IOM. The IOM maintains separate FIFO pointers for the read and write FIFOs, and updates these when a PUSH or POP register is accessed. Writing to the IOMn_FIFOPUSH register will perform a push event of the word into the FIFO and update the write pointer by 4 bytes. Only word accesses are supported to the IOM, and any unused bytes within a word will be discarded. An example of a 5 byte write transfer is shown below.

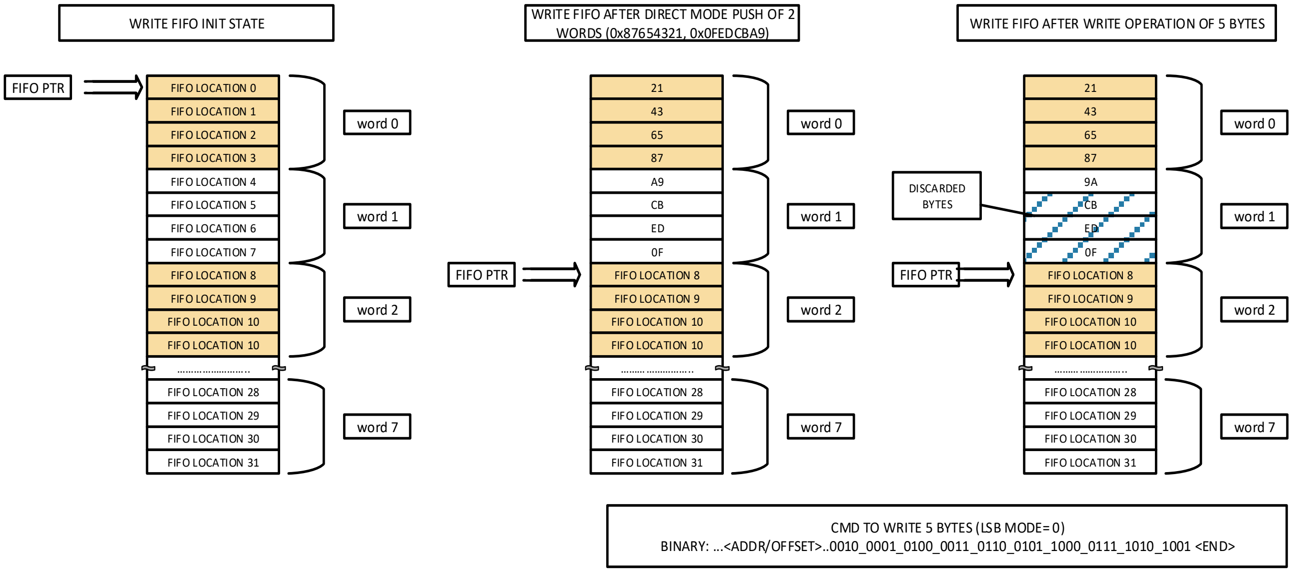


Figure 29. Direct Mode 5-byte Write Transfer

Reading from the IOMn_FIFOPOP register will perform a POP operation, return 4 bytes of data and advance the internal read FIFO pointer by 4 bytes. Any unused bytes within the read data will be filled with 0's and aligned to a word boundary at the end of the transaction. An example of a 5 byte read operation is shown below.

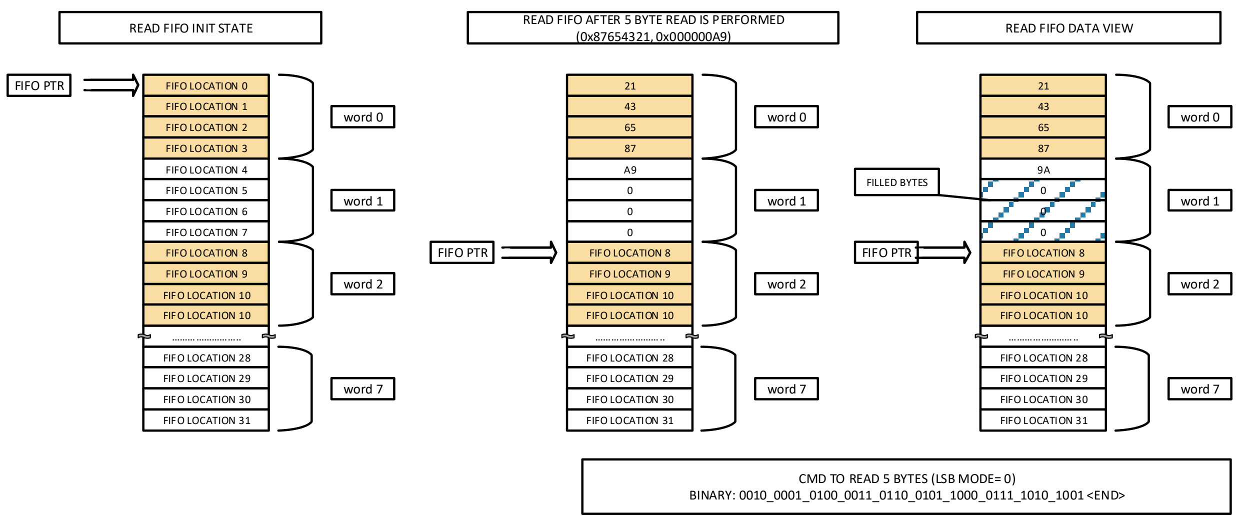


Figure 30. Direct Mode 5-byte Read

The IOM also supports a non-destructive POP mechanism to prevent unintended POP events from occurring. If the IOMn_FIFOCTRL_POPWR field is active (1), a write to the IOMn_FIFOPOP register will be required in order to complete the POP event. Reads will return the current data.
An active transaction will be paced by data availability and will hold the clock low if there is not enough data to continue write operations, or if the read FIFO is full during read operations. This wait condition is indicated when the IOMn_CMDSTAT_CMDSTAT field is 0x6. Once new data or FIFO locations are present, the command will continue operation automatically.

16.7.2 DMA Data transfers

DMA transfers are enabled by configuring the DMA related registers, enabling the DMA channel, and then issuing the command. The command will automatically fetch and store the data associated with the command without MCU intervention. The DMA channel is enabled via the IOMn_DMACFGDMAEN field. P2M DMA operations transfer data from peripheral to memory and are used in IOM READ operations. M2P DMA operations transfer data from memory to peripheral and are used in IOM write operations. DMA transfer size is programmed into the IOMn_DMATOTCOUNT register and supports up to 4095 bytes of data transfer. The DMA transfer size is independent from the transaction size, and allows a single DMA setting to be used across multiple commands. The direction of DMA data transfer must match the command. The IOMn_DMACFGDMAEN field enables/disables the DMA transfer capability and must be set last when configuring the DMA, generally prior to sending the command.

The DMA engine within the module will initiate a transfer of data when a trigger event occurs. There are 2 types of triggers available, threshold (THR) and command completion (CMDCMP). The THR trigger will activate when the threshold programmed into the FIFOWTHR or FIFORTHR in the IOMn_FIFOTHR register meets the data criteria. Because the MCU access to the interface is 32 bits wide, only the word count of the selected THR is used, and the low order bits of the FIFOWTHR or FIFORTHR are ignored.

During the transfer, the TOTCOUNT register is decremented to reflect the number of bytes transferred.

For IOM write operations (data written from IOM out to an external device), the THR trigger will activate when the write FIFO contains FIFOWTHR[5:2] free words. If the remaining DMA transfer size is less than this, only the needed number of words are transferred.

For IOM read operations (data read from external device), the THR trigger will activate when the read FIFO contains FIFORTHR[5:2] words of valid data. If the remaining DMA transfer size is less than the RTHR words, then the CMDCMP trigger can be enabled to transfer the remaining data. If the CMDCMP trigger is disabled, and the number of bytes in the read FIFO is greater to or equal to the current TOTCOUNT, a DMA transfer of TOTCOUNT will be done to complete the DMA operation. This mode requires that the THR trigger be enabled as well.

The CMDCMP trigger activates when the command is complete and will transfer the lesser of the TOTCOUNT or the number of bytes in the read FIFO. Note that this trigger is not needed for write operations, and the THR trigger should be used in this case. If a read operation is done, and the THR trigger is disabled, and only the CMDCMP trigger is enabled, and the transaction size is greater than the FIFO size (32 bytes), the module will hang, as there is no trigger to cause a DMA operation, and the logic will pause the interface until there is room within the read FIFO to store data.

If DMA transfer size is matched to the IOM transaction size, it is recommended to program both the FIFORTHR and FIFOWTHR to 0x10 (16 bytes) and only enable the THR trigger.

16.8 Transaction Initiation

To start a transaction, the IOM module must be powered up and the target external pins enabled via the GPIO module. For SPI transactions, this will generally require 4 pins to be enabled via the function select field of the PADREG registers in the GPIO module. The CEN pin for SPI transaction requires setting of the FNCSEL field of the appropriate pin, as well as the CFGREG of the corresponding pin. This also includes the setting of the default value of the CEN. This is needed to allow the IOM module to power down and not activate the CEN signal.

Once the IOM module is powered on, and the external pins configured, the IOM submodule must be enabled via the IOMn_SUBMODCTRL register. This will activate either the SPI or I^2C interface. Once this is complete, the submodule specific registers should be configured to set the desired mode and features. If DMA is desired, the DMA registers should also be set, with the IOMn_DMACTRLDMAEN field set last. The registers relating to DMA operations are as follows:

- IOMn_DMATRIGEN – Sets the trigger source for starting a DMA transfer

- IOMn_DMACFG – Sets the DMA direction and enable for DMA
- IOMn_DMATOTCOUNT – Sets the total count of bytes to be transferred via the DMA operation. Recommended to match the IOMn_CMD.TSIZE field for simplicity.
- IOMn_DMATARGADDR – The source or destination address of the DMA data. Sources can be either SRAM or storage. Destination address can only be SRAM. This is the memory mapped address of the DMA data as accessed by the MCU.

After the module setup is complete, the command register is written. This will start the IO transfer. The IOMn_CMD register contains the command itself, along with other fields used in the command, such as channel number, offset counts and transfer size. The IOM supports 2 main commands, read and write. A read command will write user selectable number of offset bytes (0 to 3), and then read IOMn_CMDTSIZE bytes, storing the data into the read FIFO. A write command will write the user selectable number of offset bytes (0 to 3), followed by a write of IOMn_CMDTSIZE bytes sourced from the write FIFO. Transfer sizes can be 0-4095 bytes for SPI operations and 0-512 bytes for I^2C operations. The number of offset bytes for each command is specified in the IOMn_CMDOFFSETCNT field.

16.9 Command Queue

The IOM module can also fetch register write data from SRAM or storage, and update the registers as if the write was performed via the MCU. Register data is stored as a doublet of 2 words. The first word is the module register address offset, word aligned. The second word is the write data value. Once enabled, the command queue (CQ) will fetch the address and perform a write to the register. If no command is started by the register write, the next doublet will be fetched by the CQ. If a command is started (write to IOMn_CMD register is done), the CQ processing will wait until the transaction is complete before fetching the next register write doublet. This is shown in the diagram below. No prefetching is done via the CQ, and the register write operations are performed in series with the transactions. This allows a predictable path for execution of commands. DMA enabled commands should be used during CQ operation, as there is no support to perform a direct mode read operation via the CQ.

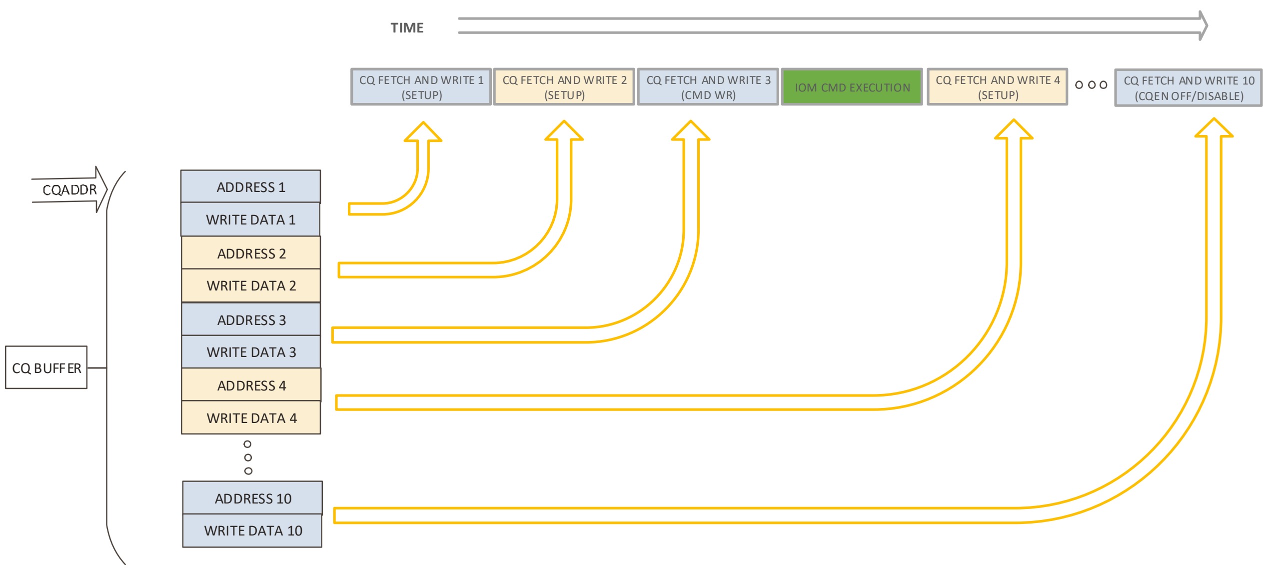


Figure 31. Register Write Data Fetches

The CQ starting fetch address is specified in the IOMn_CQADDR register. The CQ operation will start to fetch when the IOMn_CQCFG.CQEN field is set. This field should only be set when the IOM is idle and the FIFOs are empty. Once enabled, the CQ will continue to fetch sequentially until it encounters a pause event. A pause event can be caused by a CQ register write operation, or from external signals. This is shown in the sequence below.

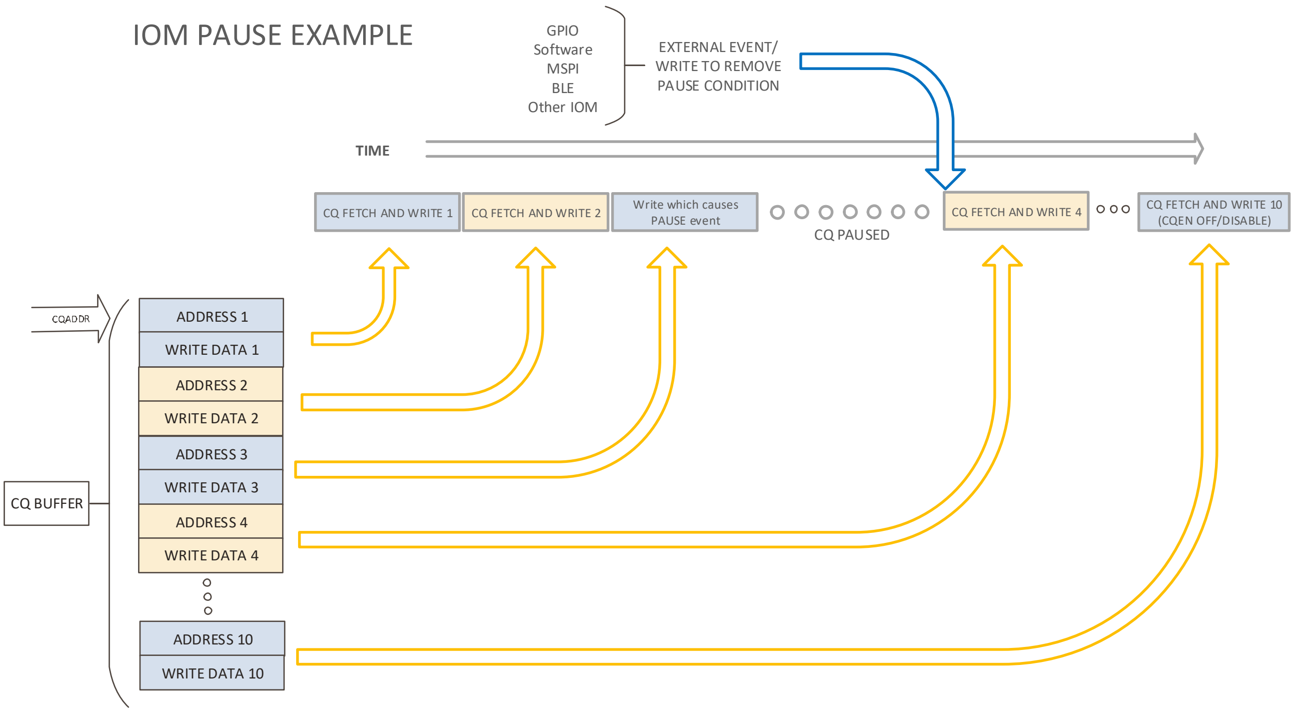


Figure 32. IOM Pause Example

Each pause source is independently enabled via the IOMn_PAUSEEN register. In addition to independent enable of the pause bits, there is also independent control of which pause event will signal a CQPAUSE interrupt. This is controlled through the IOMn_CQFLAGS.CQIRQMASK field.

There are 16 possible pause sources. When the value of the pause source is set, and the pause is enabled in the IOMn_PAUSEEN register, the CQ will stop fetching. The IOMn_CQADDR is updated after each fetch, and when paused, will point to the next doublet to be fetched when the pause condition is removed. The connection of the pause bits are shown below. The SW Flags are accessed via the IOMn_CQSETCLEAR register.

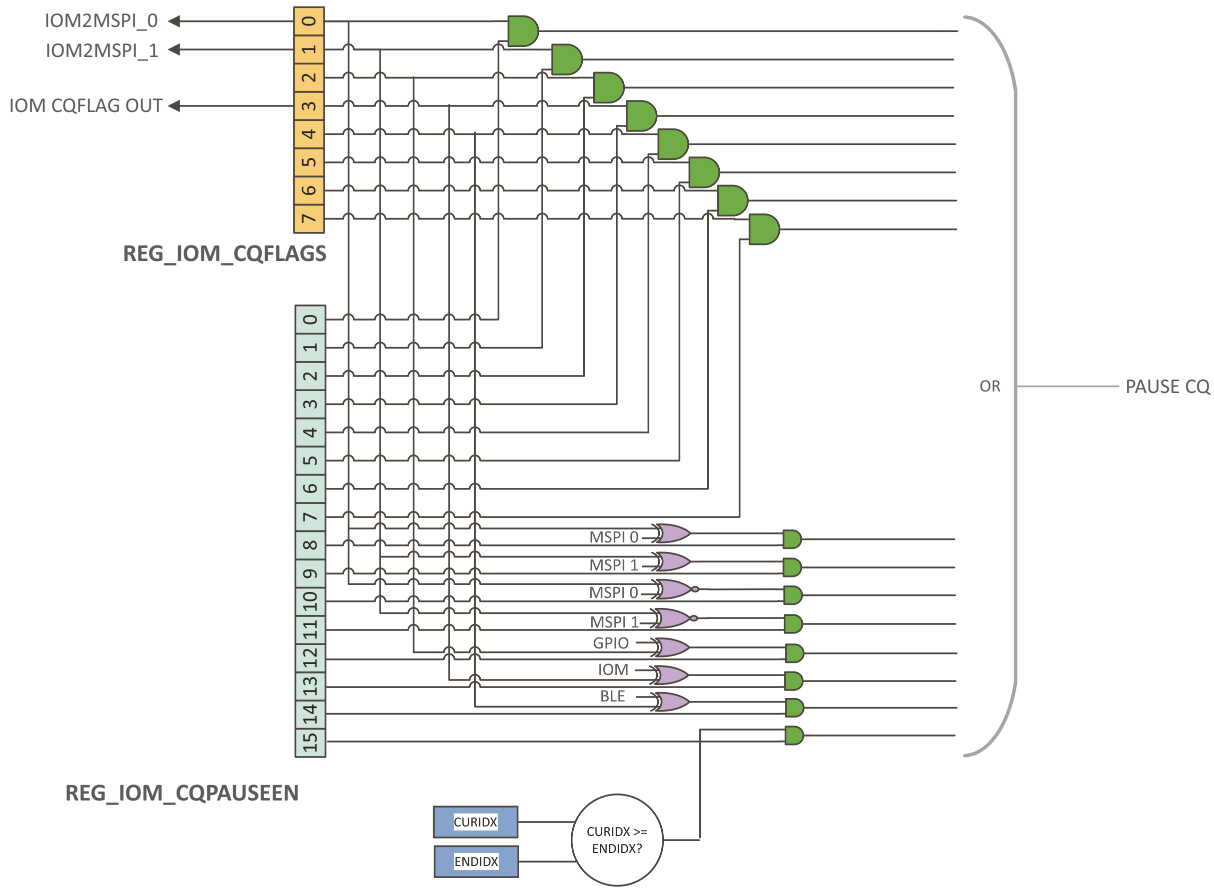


Figure 33. CQ Pause Bit Fetching

The first 8 pause sources (bits 7:0) are register bits which are directly writable via the MCU or through the CQ. These first 8 locations are called SW Flags. Because the CQ does not support a read-modify-write operation, special facilities are available to set, reset or toggle the SW Flags. This is accessed through the IOMn_CQSETCLEAR register. The 3 fields in this register allow a per bit set, reset or toggle of the SW Flag bits.

The next 7 pause sources (bits 14:8) use the SW Flags along with an external signal to set the pause event. The external signals are from the GPIO module, the MSPI module, or other IOM modules. On some cases, such as the MSPI interface, 4 of the SW Flags are used and combined with 2 similar signals from the MSPI module to facilitate a ping pong method of sharing 2 buffers and preventing overruns without MCU intervention.

The last pause source (bit 15) is used for index pausing. If this pause bit is enabled, the CQ will pause when the value of the IOMn_CURIDX matches the IOMn_ENDIDX. This is useful for software to be able to update the CQ buffer without causing a race condition between the CQ data buffer writes and the CQ fetches.

16.9.1 CQ Programming Notes

- Additional restrictions when using the CQ function is that the DMA must be disabled prior to writing the IOMn_CQADDR register, either from the MCU or from the CQ itself.
- For multiple commands using DMA, the DMAEN must be reset after the command is done and before the DMA registers are set for the next transaction.

- It is possible for the CQ to write the IOMn_CQADDR register during the CQ operation. The new address will take effect on the next fetch and allows the CQ to be relocated or looped.
- When starting the CQ operation, 1 doublet will be fetched regardless of the state of the pause status and bits. If any pause is active, it will take effect after the first fetch. For this reason, it is generally advisable to have a dummy register write as the first CQ doublet.

CQ write operations to SW flags used in combination with pause events 15:8 must first disable the pause enable, perform the SW flag write, then re-enable the pause enable register. SW flags 7:0 can be written without this restriction and will cause a pause immediately if activated.

NOTE
Due to the susceptibility of creating a clock glitch which could cause register corruption, changing SPHA and SPOL bits should be done in separate writes to the MSPICFG register.

16.10 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about IO Master Module operations, where the following topics are covered:

- Interface Clock Generation
- Command Operation
- FIFO
- I^2C Interface
- SPI Operations
- Bit Orientation
- SPI Flow Control
- Minimizing Power
- IOM Registers

17. I^{2}C/SPI Slave (IOS)

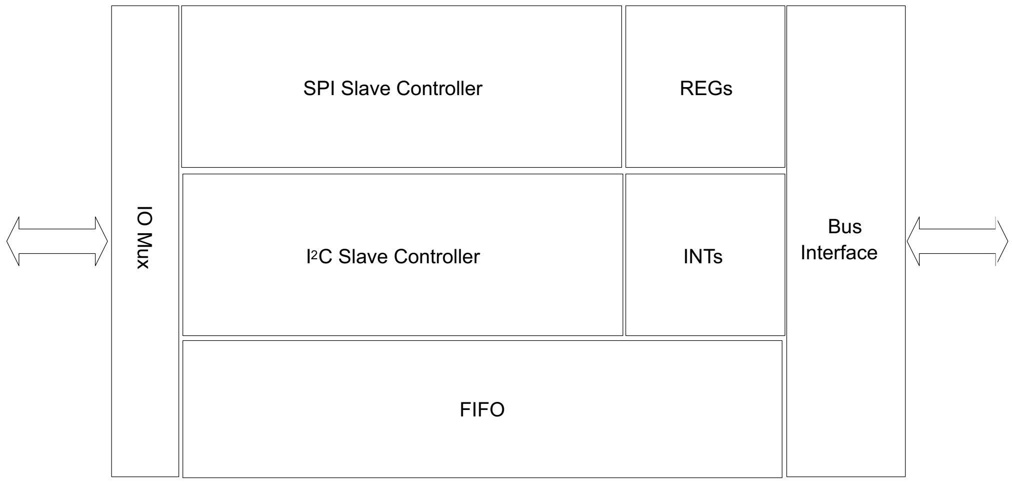


Figure 34. Block diagram for the I^{2}C/SPI Slave Module

17.1 Functional Overview

The I^{2}C/SPI Slave (IOS) Module, shown in Figure 34, allows the Apollo4 Blue SoC to function as a Slave in an I^{2}C or SPI system. The I^{2}C/SPI Slave operates in an independent fashion, so that the device may be placed in a sleep mode and still receive operations over the I/O interface. The Slave may be configured to generate an interrupt on specific references.

The I^{2}C/SPI Slave contains 256 bytes of RAM which is only accessible when the module is enabled. This RAM may be flexibly configured into three spaces: a block directly accessible via the I/O interface, a block which functions as a FIFO for read operations on the interface, and a block of generally accessible RAM used to store parameters during deep sleep mode.

In I^{2}C mode the Slave supports fully configurable 7 and 10-bit addressing with interface timing limits as specified in the Inter-Integrated Circuit (I^{2}C) Interface section of the Electricals chapter. In SPI mode, the Slave supports all polarity/phase combinations and interface frequencies as specified in the Serial Peripheral Interface (SPI) Slave Interface section.

Please refer to the IO Slave registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

17.2 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about IO Slave Module operations, where the following topics are covered:

- Local RAM Allocation
- Direct Area Functions
- Rearranging the FIFO
- Interface Interrupts
- Command Completion Interrupts

- Host Address Space and Registers
- I^2C Interface
- SPI Interface
- Bit Orientation
- Wakeup Using the I^2C/SPI Slave
- IOSLAVE Registers
- Host Side Address Space and Registers

18. Universal Asynchronous Receiver/Transmitter (UART)

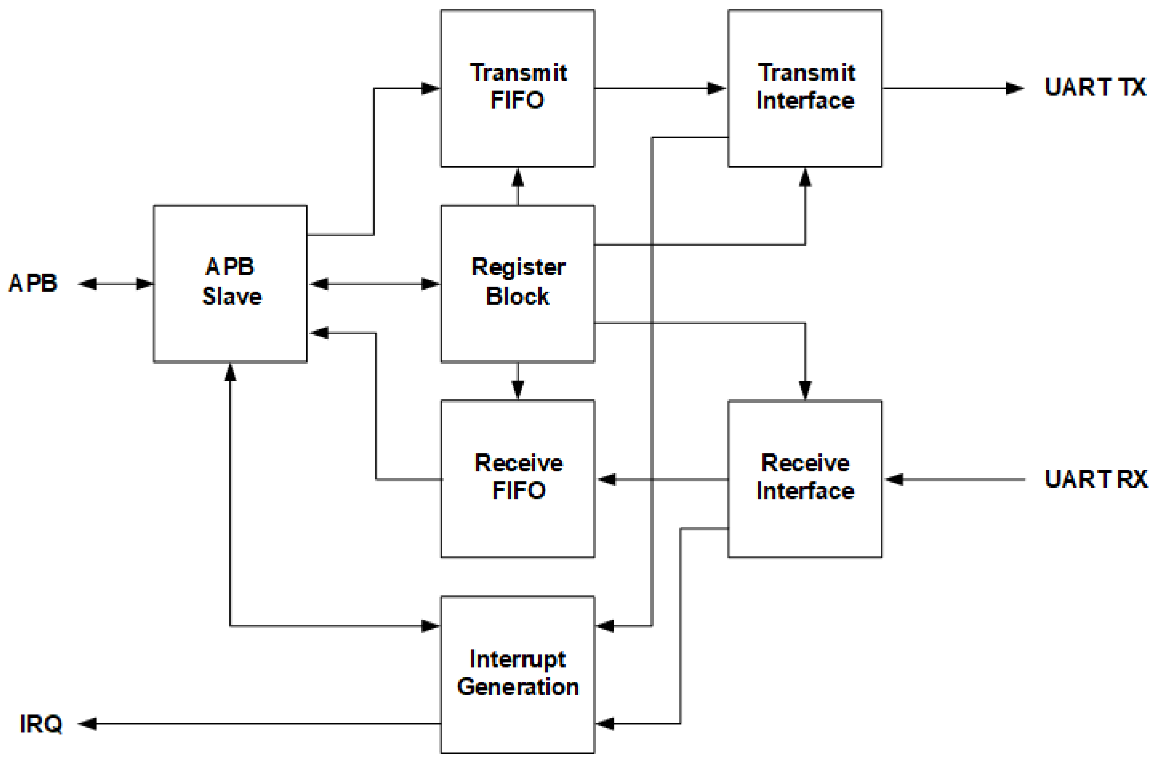


Figure 35. Block Diagram for the UART Module

18.1 Features

There are four (4) UART instances in the Apollo4 Blue SoC. The UART Module includes the following key features:

- Operates independently, allowing the SoC to enter a low power sleep mode during communication
- 32 x 8 transmit FIFO and 32 x 12 receive FIFO to reduce MCU computational load
- Programmable baud rate generator
- Fully programmable data size, parity, and stop bit length
- Programmable hardware flow control
- Support for full-duplex and half-duplex communication
- Loop back functionality for diagnostics and testing

18.2 Functional Overview

Shown in Figure 35, the UART Module converts parallel data written through the APB Slave port into serial data which is transmitted to an external device. It also receives serial data from an external device and converts it to parallel data, which is then stored in a buffer until the CPU reads the data.

The UART Module includes a programmable baud rate generator. An interrupt generator will optionally send interrupts to the CPU core for transmit, receive and error events.

Internally, the UART Module maintains two FIFOs. The transmit FIFO is 1-byte wide with 32 locations. The receive FIFO is 12-bits wide with 32 locations. The extra four bits in the receive FIFO are used to capture any error status information that the MCU needs to analyze.

18.3 Power Control

The 4 UART modules must be enabled in the PWRCTRL_DEVPWREN register prior to access and operation. The power status of the UART modules can be read in the PWRCTRL_DEVPWRSTATUS register. Note that the UART modules are in a single power domain, referred to as HCPA. When one UART is powered on, all other UARTs in this group are powered on as well.

Please refer to the UART registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

18.4 Additional Information

Please consult the Apollo4 Family Programmer’s Guide for additional information about UART Module operations.

19. Universal Serial Bus (USB)

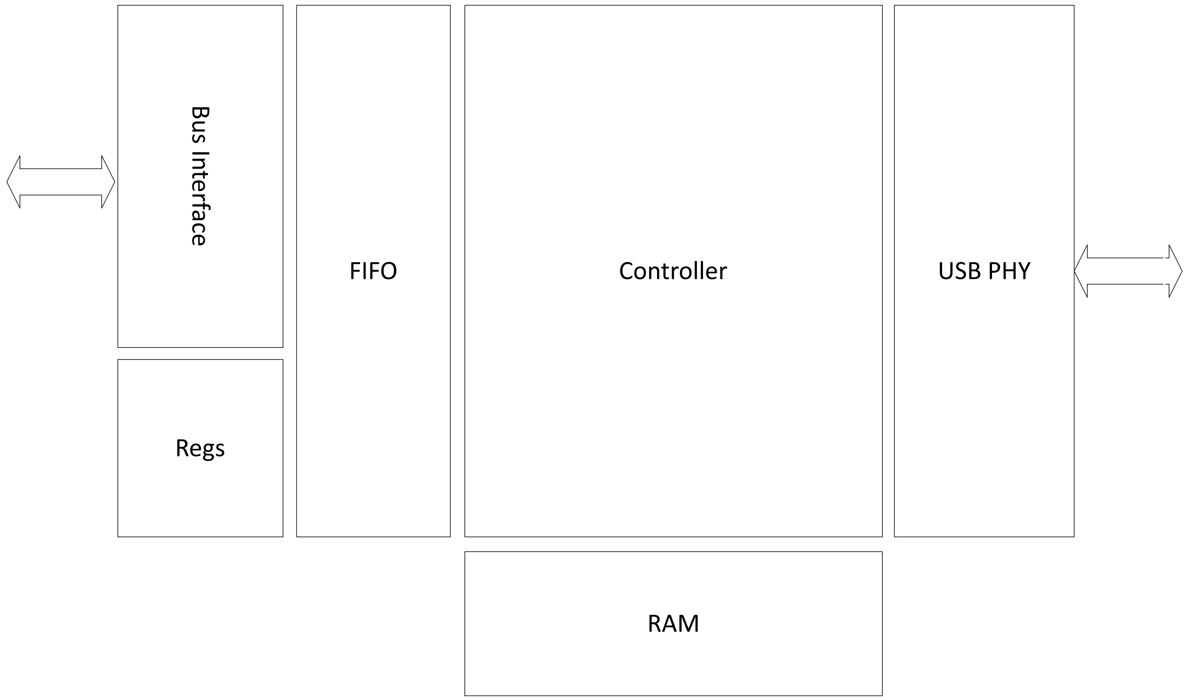


Figure 36. USB Block diagram

19.1 Features

The following features are supported in the Apollo4 Blue SoC:

- USB 2.0 FS/HS device with support for low-power mode
- Crystal-less operation
- Battery charging detection (BC1.2 and vendor-specific)
- On-die pull-ups/pull-downs and termination (no external calibration resistors, pull-ups or pull-downs required)
- Dynamic FIFO sizing: 4 kB total FIFO
- IN endpoints: 5
- OUT endpoints: 5
- IN bulk packet splitting
- OUT bulk packet combining
- Soft connect/disconnect
- Suspend mode

19.2 Functional Overview

The USB subsystem provides support for USB full speed (12 Mbps) and high speed (480 Mbps) interface. This interface is primarily used for bulk data transfer, firmware updates and charging detect.

The USB controller supports up to 5 IN / 5 OUT endpoints plus 1 control. The FIFO sizing for each endpoint is dynamically configurable up to 4 kB.

The Apollo4 Blue SoC has an integrated USB 2.0 PHY with support for suspend mode operation. Battery charger detection is supported within the PHY to enable battery charge algorithm execution and control of the external battery charge / power management IC. The charger detection supports Battery Charging Specification 1.2 (BC1.2) and also supports other non-BC1.2 standards such as Apple charger.

Please refer to the USB registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

19.3 Hardware Design Guidelines

The following sub-sections provide design guidelines for the use of the Apollo4's USB Controller and PHY. Also, please consult the USB section of the Electricals for voltage, power and timing requirements of the PHY.

19.3.1 Battery Charger Detection

Charger detection during USB connection follows the flow shown in Table 37.

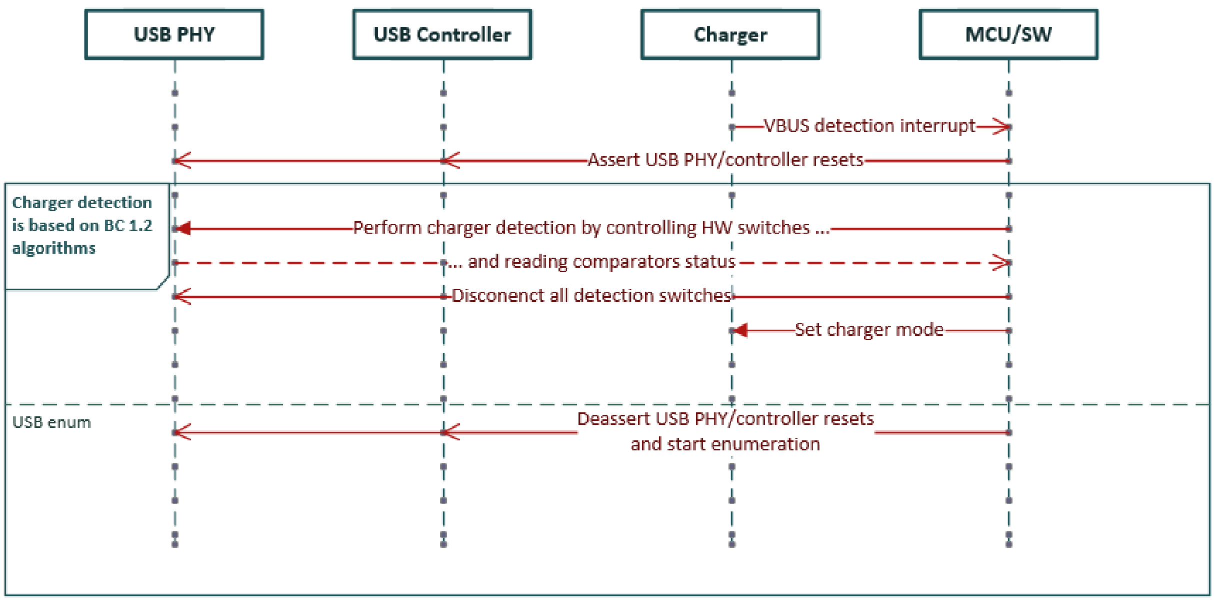


Figure 37. Charger Detection in USB Connection Flow

The charging detection algorithm in Figure 38 is based on the BC 1.2 specification. Weak Battery Algorithm (Figure 39) and Good Battery Algorithm (Figure 40) should be implemented in an interrupt-driven manner to take full advantage of Apollo4's power saving features. Note that debouncing software timers and their interrupts are not shown in the diagram for clarity, but must be implemented according to BC 1.2 specification.

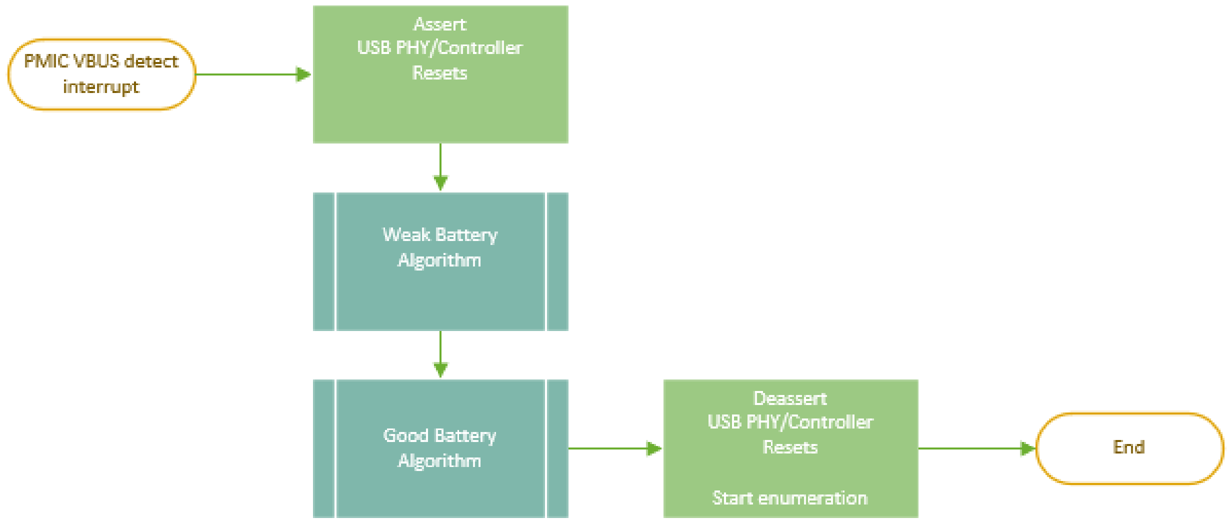


Figure 38. Charging Detection Algorithm

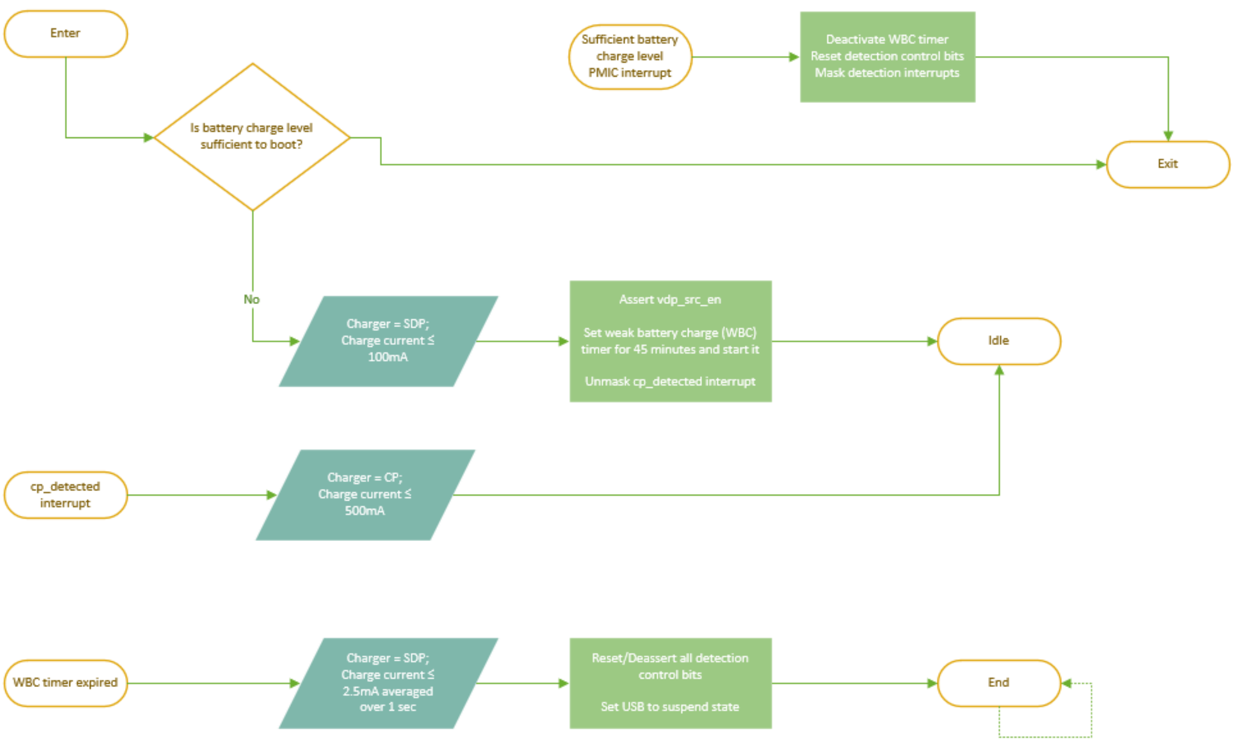


Figure 39. Interrupt-driven Weak Battery Algorithm

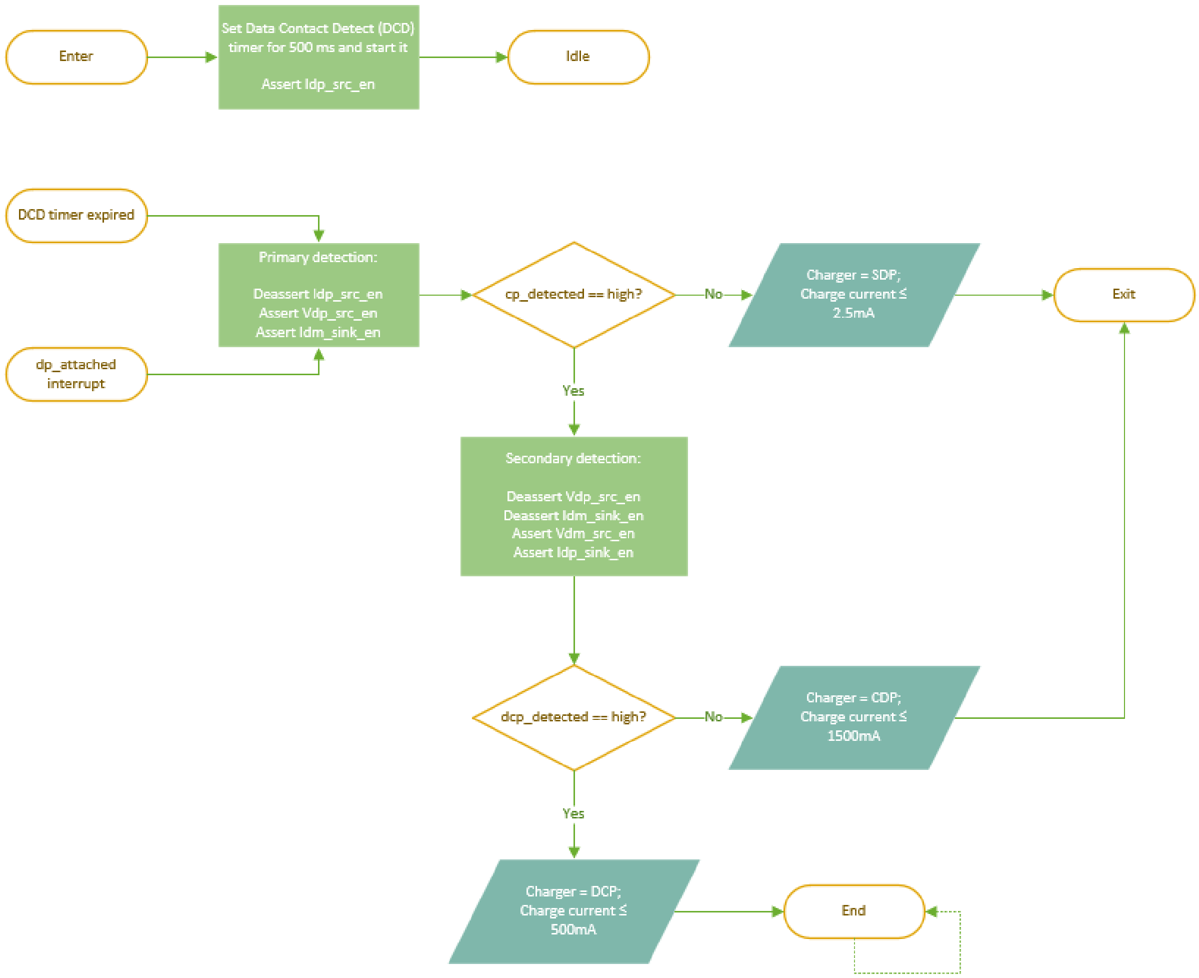


Figure 40. Interrupt-driven Good Battery Algorithm

Charger detection algorithms should take into account timings between assertion/deassertion control signal and status signal change as shown in Figure 41.

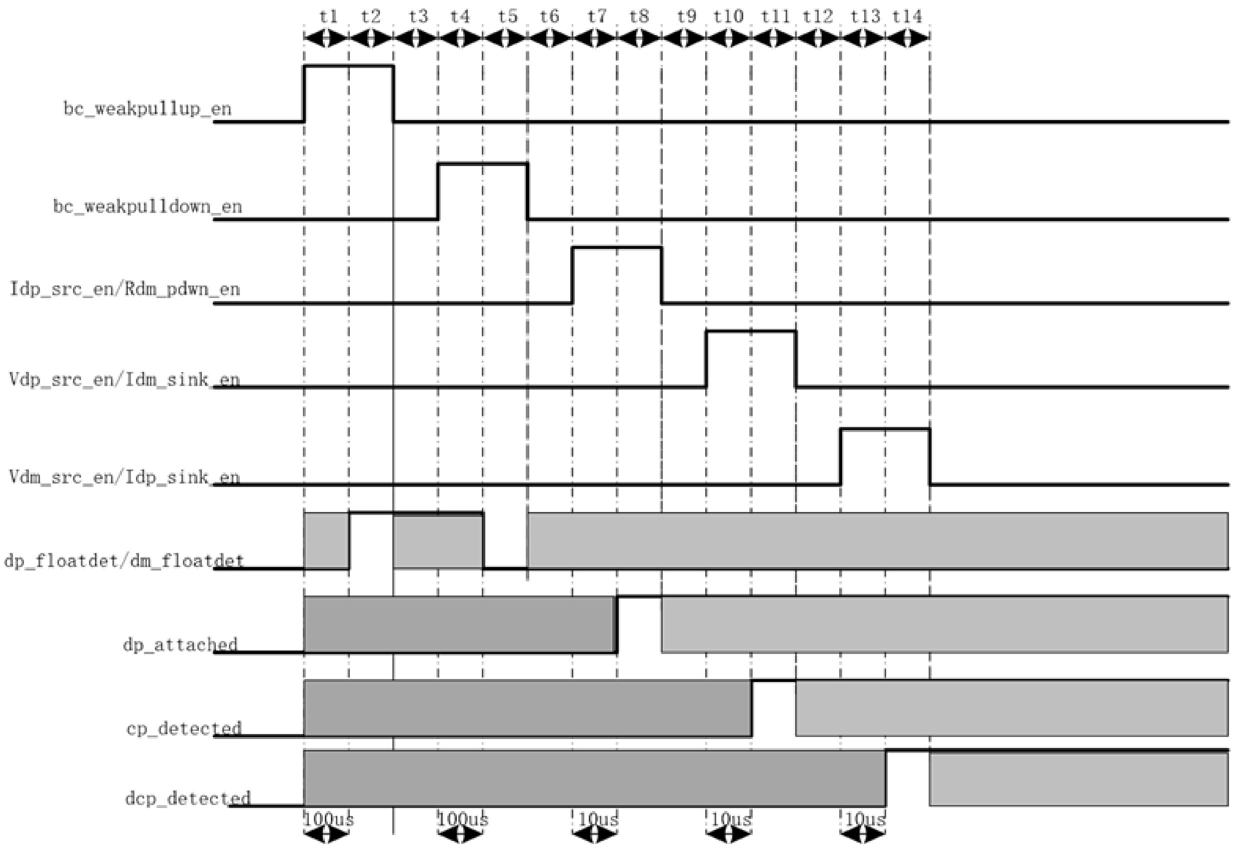


Figure 41. Battery Charging Sequence

19.3.2 Interface Timing

See USB PHY section in the Electricals for interface timing specifications.

19.3.3 System Power Sequencing for USB and DSI PHYs

The power sequence for the relevant power supplies is contingent on the PHY(s) used in the system. The sections below address the proper sequences which include powering the VDD18, VDDUSB33 (3.3 V) and VDDUSB0P9 (0.9 V) supplies, and cover the cases for USB PHY only, and both USB and DSI PHYs in the system.

Refer to the USB PHY section in the Electricals for supply voltage specifications.

Throughout the power sequences are references to calls in the AmbiqSuite SDK. It is highly recommended to use the SDK and the referenced routines.

19.3.3.1 USB PHY Only

19.3.3.1.1 Recommended Termination of Unused Interface

1. USB data pads (USB0PP and USB0PN) left open
2. USB PHY power rails VDDUSB33 and VDDUSB0P9 connected to ground

19.3.3.1.2 Power Tree

The recommended power tree for this configuration is as follows:

1. VDDUSB33 and VDDUSB0P9 should be powered by an LDO with output discharge and ON/OFF control over I^{2}C or GPIO. It is recommended to source VDDUSB33 and VDDUSB0P9 power from USB VBUS to minimize system power consumption from the battery. Some examples of suitable standalone small form factor LDOs:
   - ST Micro LDBL20 in 0.47 x 0.47 x 0.22 mm STSTAMP™ package
   - ST Micro LD39130S in 0.69 x 0.69 x 0.5 mm CSP package
   - TI LP5910 0.74 x 0.74 x 0.4 mm DSBGA package
2. Required system power-on state: VDDUSB33 is OFF, VDDUSB0P9 is OFF

19.3.3.1.3 Recommended Interface ESD protection and Common Mode Filtering

ESD protection on the USB data lines, USB0PP and USB0PN, is required. An integrated CMF and TVS solution, such as the Nexperia PCMF1USB3B/C or Panasonic EXC-14CS900H, is recommended.

If the design achieves EMC compliance without CMF on the USB data lines, then a TVS-only solution, such as the TI ESD122DMXR, may be used.

19.3.3.1.4 USB VBUS Detection

Apollo4 has no 5V-tolerant pins. Therefore, it relies on the external circuit for getting VBUS power OK (VBUS connected) status. The recommended solution is to connect PMIC/charger VBUS_OK output to Apollo 4 interrupt-capable GPIO pin.

19.3.3.1.5 USB Handling in AmbiqSuite SDK

1. The Apollo4 USB initialization routine in the AmbiqSuite SDK should be used to enable Suspend Mode by setting the ENABL bit of the USB_CFG0 Register of the USB Controller.
2. To minimize power in the USB suspend state, the software should turn off the FS differential receiver in the Suspend ISR. This may be done by clearing bit 1 of the USB PHY register at offset 0x10. The Resume ISR requires powering on the FS receiver by setting this same bit.
3. An API for registering a given GPIO as the USB VBUS interrupt source is available in the SDK. It can be configured with various triggering options such as level/edge and positive/negative edge triggering).

19.3.3.1.6 USB PHY Initialization Sequence

The sequences below assume that the SoC is powered on and booted to the application. On USB VBUS connect interrupt, perform the following actions.

Silicon Start-up Sequence:

1. Enable Apollo4 internal power rail to USB interface (VDDF_USB_SW).
2. Enable power to both VDDUSB0P9 and VDDUSB33 simultaneously.
3. Enable USB PHY reset override:
   - Force USB PHY POR reset by clearing the MCUCTRL_USBPHYRESET_USBPHYPORRSTDIS bit.
   - Force USB PHY UTMI reset by clearing the MCUCTRL_USBPHYRESET_USBPHYUTMIRSTDIS bit.

Table 14: **Reset Bits in the USBPHYRESET Register**

MCUCTRL_USBPHYRESET Register Address: 0x40020418			
Bit No.	**Name**	**R/W**	**Description**
1	USBPHYUTMIRSTDIS	W	De-assert USB PHY UTMI reset
0	USBPHYPORRSTDIS	W	De-assert USB PHY POR reset

4. Perform charger detection (optional step).
5. Disable USB PHY reset override:
   - Remove force from USB PHY POR reset by setting the MCUCTRL_USBPHYRESET_USBPHY-PORRSTDIS bit.
   - Remove force from USB PHY UTMI reset by setting the MCUCTRL_USBPHYRESET_USBPHYUT-MIRSTDIS bit.
6. Initialize USB SW stack.
7. Enable host connect (call am_hal_usb_attach()).

NOTE

Regarding the selection of HFRC2 for the USB PHY reference clock in High-Speed Mode, there is the possibility of the asynchronous shutdown of the HFRC2 clock divider causing a glitch when the requesting peripheral stops requesting the HFRC2 and allows it to be shut down. The HFRC2 must be forced on when it is to be used such that it does not get powered down by the internal hardware.

The HFRC2 is forced on by setting the CLKGEN_MISC_FRCHFRC2 bit. The sequence for selecting the HFRC2 as the clock source is to first force FRCHFRC2 bit on, then switch clock sources, and finally engage the peripheral.

If HFRC2 is the clock source, then shutting the module down cleanly requires switching to HFRC, for example, and then disabling the HFRC2 by clearing the CLKGEN_MISC_FRCHFRC2 bit.

19.3.3.1.7 USB PHY Shut-down Sequence

On a USB VBUS disconnect interrupt, the SoC should follow the below shut-down sequence.

Silicon Shut-down Sequence:

1. Put USB device in SUSPEND state.
2. Disable host connect (call am_hal_usb_detach()).
3. Disable power to USB PHY external rail(s).
4. Enable USB PHY reset override:
   - Force USB PHY POR reset by clearing the MCUCTRL_USBPHYRESET_USBPHYPORRSTDIS bit.
   - Force USB PHY UTMI reset by clearing the MCUCTRL_USBPHYRESET_USBPHYUTMIRSTDIS bit.

5. Disable Apollo4 internal power rail to USB interface (VDDF_USB_SW).

19.3.3.2 USB and DSI PHYs

19.3.3.2.1 Power Tree

The recommended power tree for this configuration is as follows:

1. VDD18, VDDUSB33, and VDDUSB0P9 are powered by LDO rails with output discharge and an independent ON/OFF control over I^2C or GPIO. It is recommended to source VDDUSB33 and VDDUSB0P9 power from USB VBUS to minimize system power consumption from the battery.
2. The following power-on (default) states are allowed:

   - VDD18 is OFF, VDDUSB33 is OFF, VDDUSB0P9 is OFF

19.3.3.2.2 USB VBUS Detection

Designs utilizing USB should implement USB VBUS detection via GPIO interrupt asserted on USB VBUS connect event.

19.3.3.2.3 System Initialization Sequence

The sequence below assumes that the SoC is powered on and booted to the application. On USB VBUS connect interrupt, perform the following actions:

1. Perform DSI initialization sequence as described in section “DSI PHY Only” on page 150.
2. Perform USB initialization sequence as described in section “USB PHY Only” on page 130.

19.3.4 Suspend State Power Consumption

USB PHY power consumption in suspend state when both 3.3 V and 0.9 V are supplied is as specified in the Electricals.

19.3.5 USB Data Line Filtering

A TVS on the data lines, USB0PP and PSB0PN, is required. An integrated CMF and TVS solution, such as the Panasonic EXC-14CS900H, is recommended. If the design achieves EMC compliance without CMF on the USB data lines, then a TVS-only solution, such as the TI ESD122DMXR, may be used.

19.3.6 Charger Detection and USB Enumeration Requirements

Apollo4 has no 5V-tolerant pins. Therefore, it relies on the external circuit for getting VBUS power OK (VBUS connected) status. The recommended solution is connecting PMIC/charger VBUS_OK output to Apollo 4 interrupt-capable GPIO pin.

19.3.7 LDO for USB PHY Power

Suitable small form-factor LDOs which could be used for USB PHY power rails are as follows:

- ST Micro LDBL20 in 0.47 x 0.47 x 0.22 mm STSTAMP™ package: https://www.st.com/resource/en/datasheet/ldbl20.pdf
- ST Micro LD39130S in 0.69 x 0.69 x 0.5 mm CSP package: https://www.st.com/resource/en/datasheet/ld39130s.pdf
- TI LP5910 0.74 x 0.74 x 0.4 mm DSBGA package: http://www.ti.com/product/LP5910

19.3.8 Unused Interface Terminations

When the USB port is not used, the following supply and signal terminations should be followed.

- DP/DM pins should be left open.
- VCCA3P3 and VCCA0P9 should be connected to ground.

19.4 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about USB Module operations.

20. Secure Digital Input Output (SDIO)

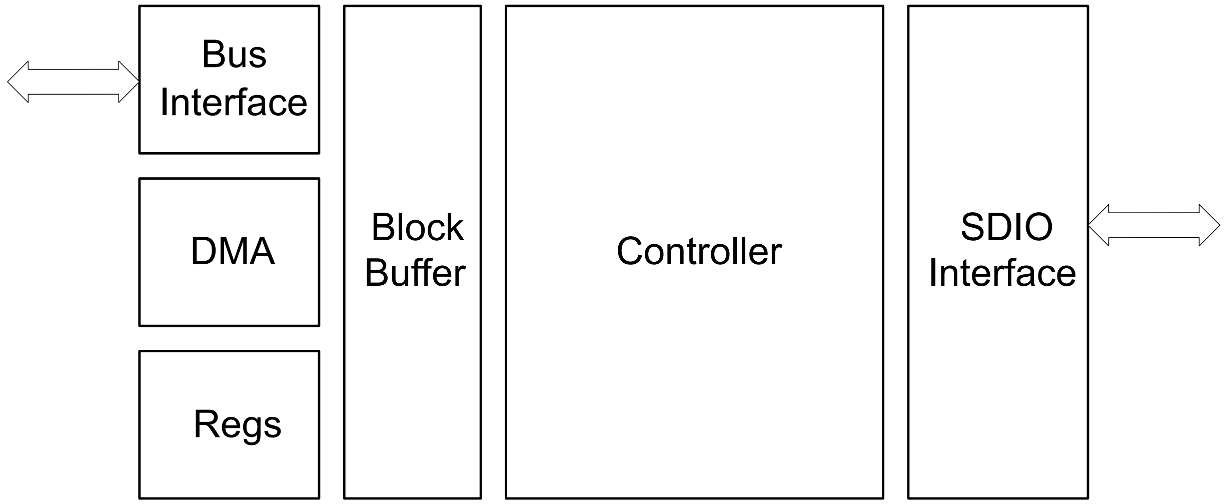


Figure 42. SDIO Block Diagram

20.1 Features

Features of the SDIO Module are as follows:

- SDIO card specification Version 3.0
- Host clock rate variable between 0 and 96 MHz
- Up to 50 MBytes per second data rate using 4 parallel data lines (SDR50/DDR50 mode)
- Transfers data in 1 bit and 4 bit SD modes
- Transfers data in SDR50 or DDR50 modes
- Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity
- Variable-length data transfers
- Performs Read wait Control, Suspend/Resume operation SDIO CARD
- Supports Read wait Control, Suspend/Resume operation

20.2 Functional Overview

The SDIO host controller provides support for higher bandwidth device transfer. Typical application is for IC connectivity. The SDIO controller supports up to 2 kB block buffering as well as dedicated DMA controller support to provide maximum host offload. The DMA algorithm supported is the Advanced DMA version 2 (ADMA2) which allows for flexibility in memory allocation. The controller interface supports a programmable DLL to allow for timing tuning for optimal windowing.

Please refer to the SDIO registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

20.3 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about SDIO Module operations.

21. Display Controller (DC)

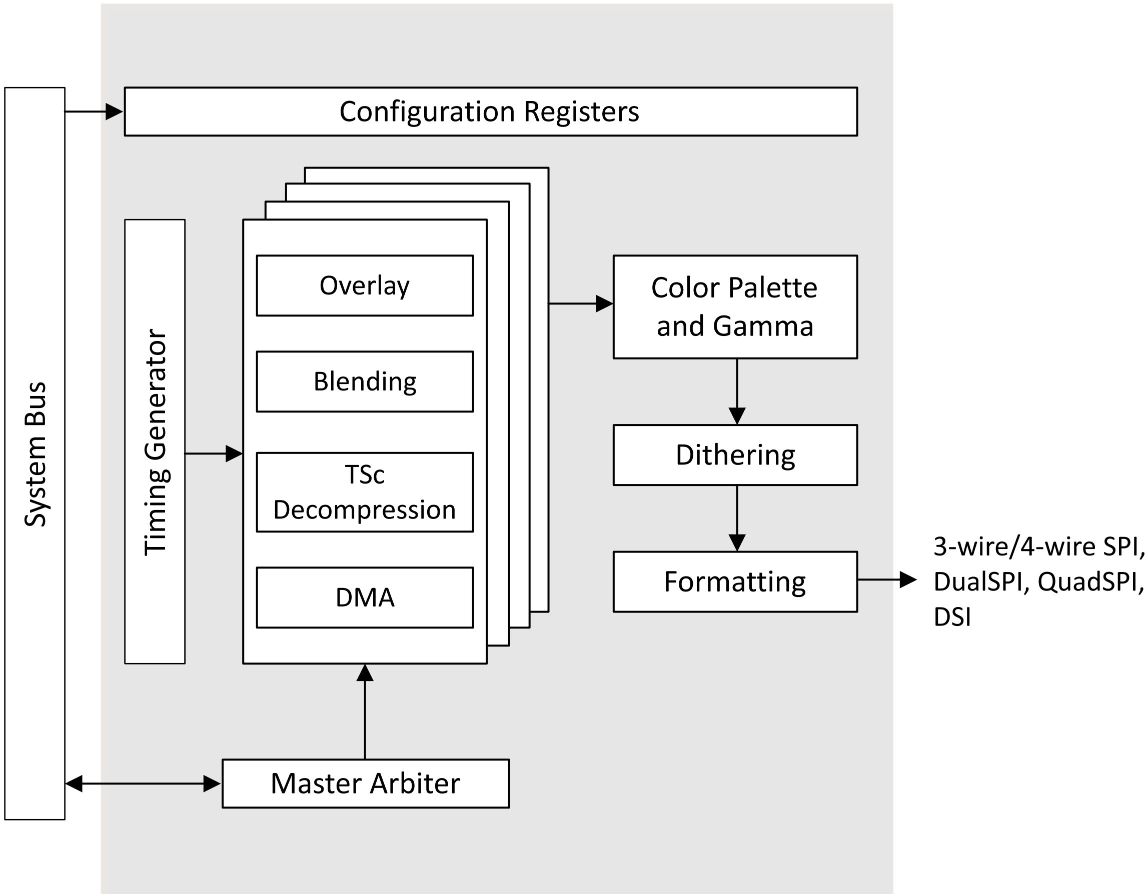


Figure 43. Apollo4 Blue SoC Display Controller Block Diagram

21.1 Features

- Programmable display resolutions up to 500x500
- Compressed framebuffer support
- 4 graphics layers
- Powerful composition
  - Alpha blending
  - Programmable size, offset and format per layer
  - Programmable stride/pitch enabling panning and clipping
- Per layer palette
- Global or Gamma correction (layers 3 and 4 only)
- Configurable dithering 15/16/18-bits for better results on displays

21.2 Functional Overview

The Display Controller (DC) contains several smart tools and functionalities to compose multiple graphics layers by improving image quality and contributing significantly to the reduction of the SoC power consumption.

The DC supports composition features, a wide range of display interfaces and advanced proprietary frame-buffer compression technology. The core is designed to lift the workload off the Graphics Processing Unit (GPU) or the host processor (CPU), in GPU-less systems, and minimize the memory bandwidth.

Multiple layers can be clipped, positioned and composed on the final display by overlaying graphics or application windows, with or without transparency. The Display Controller supports four layers.

Please refer to the Display Controller registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

21.2.1 Display Interfaces

- 3-wire/4-wire SPI
  - Pixel data writes at 50MHz
  - Command read at 10MHz (1-wire)
  - Command write at 50MHz (1-wire)
- DualSPI or QuadSPI interface mode
- DSI (Display Serial Interface) via on-chip DSI module
- Serial formats 2-beat, 3-beat and 4-beat RGB

NOTE
Use of RGBA4444 or ARGB4444 input color modes are supported, but the DC on the Apollo4 SoCs does not support any of RGB4444 interfaces, including NEMADC_RGBA4444 and NEMADC_ARGB4444. NEMADC_RGBA4444 interface is supported on Apollo4 Plus SoCs.

NOTE
Use of the DPI-2 interface is not recommended or supported, and is therefore not included in the list of available display interfaces.

21.2.2 Configuration Options

- DMA
  - Scanline based
  - Tile based (MIPI)
- Layer Overlay
  - 4 layers
- Screen formatting (output encoding)

21.3 Architecture

This section provides a high-level description of the DC's internal architecture, frame buffer compression/ decompression, the supported color formats and the display format interfaces.

21.3.1 Top Level Description

The Display Controller's configuration register file controls the operation of the display controller. Timing parameters are programmable and data are fetched for each layer by a dedicated DMA engine. Additional modules include:

- Gamma adjustment
- Dithering application

Depending on the target screen, output can be formatted to different types. Figure 43 depicts the main hardware components and the features of the Display Controller.depicts the main hardware components and the features of the Display Controller

21.3.2 Blending Modes

During the blending process, a translucent foreground color (current layer) with a background color (previous layer) are combined and a new blended color is produced. Foreground color's translucency may range from completely transparent to completely opaque. If the foreground color is completely transparent, the blended color will be the background color and if the foreground color is completely opaque, the blended color will be the foreground color. When the translucency ranges in between, the blended color is computed as a weighted average of the foreground and background colors.

21.3.3 Dithering

Dithering is the process of degrading the color image with a method that tries to produce better results than information truncation.

Dithering is used to create the illusion of "color depth" in images with a limited color palette. In a dithered image, colors that are not available in the palette are approximated by a diffusion of colored pixels from within the available palette. The human eye perceives the diffusion as a mixture of the colors within it.

21.4 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about Display Controller Module operations.

22. Graphics Processing Unit (GPU)

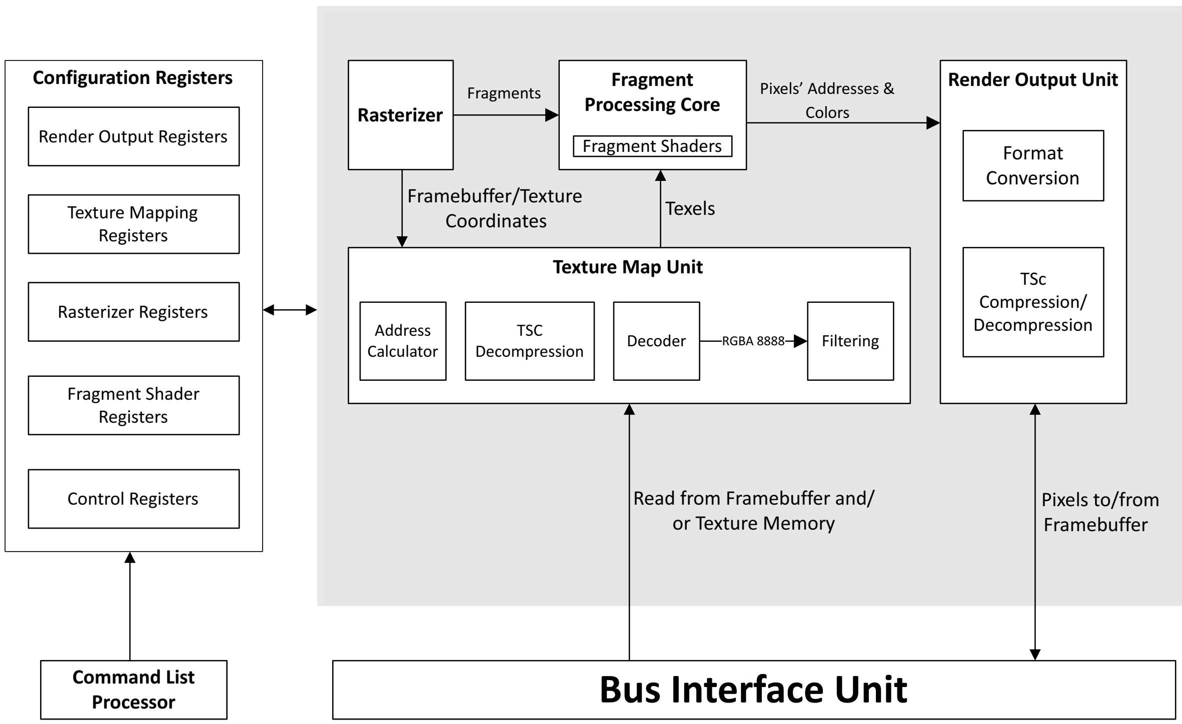


Figure 44. GPU Block Diagram

22.1 Features

- Hardware Components:
  - Programmable Shader engine
  - VLIW instruction set architecture supporting low-level vector graphics processing
  - Fixed point functional units
  - Command list-based DMAs to minimize CPU overhead
  - Primitive Rasterizer
  - Texture Mapping unit
- Drawing Primitives:
  - Pixel / Line drawing
  - Filled rectangles
  - Triangles (Gouraud Shaded)
  - Quadrilateral
- Color formats
  - 32-bit RGBA8888/BGRA8888/ABGR8888
  - 24-bit RGB
  - 16-bit RGBA5551/RGB565
  - 8-bit A8/L8/RGB332
  - 4-bit A4/L4
  - 2-bit A2/L2
  - 1-bit A1/L1
  - YUV (Read only)

  - TSC™ (Optional)
- Compression schemes
  - TSC™4 (4 bits per pixel)
  - TSC™6 / TSC™6a (6 bits per pixel with/out Alpha)
- Image transformation
  - Texture mapping
  - Point sampling
  - Bilinear filtering
  - Blit support
  - Rotation any angle
  - Mirroring
  - Stretch (independently on x and y axis)
  - Source and/or destination color keying
  - Format conversions on the fly
- 2.5D Perspective Correct Projections

22.2 Functional Overview

The GPU on the Apollo4 Blue SoC brings high quality graphics for user interfaces in a very small power budget. The GPU supports entry level IoT platforms, wearable and embedded devices with low cost and ultra-low power requirements and provides fluid graphics experience for a wide range of applications. Developers are able to create compelling Graphical User Interfaces (GUIs) and software applications with ultra-long battery life at a significantly lower cost for power-memory-area constrained IoT devices.

The GPU module's functions and interface are as shown in Figure 44.

Please refer to the GPU registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

22.3 Architecture

The GPU has been designed for graphics efficiency in ultra-compact silicon area. Its fixed-point data path and instruction set architecture (ISA) are tailored to GUIs acceleration and small display applications leading to substantial improvements in power consumption and silicon area. The GPU microarchitecture combines hardware-level support for multi-threading, VLIW and low-level vector processing in the most power efficient way.

22.3.1 I/O Interfaces

- Connected via an AXI bus that helps it communicate with the MCU, the configuration registers and system memory.
- Uses three AXI master ports 32/64-bit that access the main memory and fetch data from it (textures, frame data, etc.).
- The AXI version contains a separate command list AXI bus specifically for that purpose (CL Bus).

22.3.2 Graphics Pipeline

22.3.2.1 Configuration Register File

The GPU is programmed through a set of registers called the Configuration Register File (CRF) and each sub-module of the GPU is programmed through a subset of the CRF. The CRF can be memory mapped to the CPU address space, thus making it directly accessible. Writing the CRF directly is considered inefficient since it consumes a large volume of the CPU resources and ties the CPU execution to the GPU. For this reason, it can also be accessed indirectly through the Command List Processor (CLP).

22.3.2.2 Command List Processor

In order to decouple CPU and GPU execution and achieve both better performance and lower power consumption, the GPU incorporates an advanced Command List Processor (CLP), capable of reading entire lists of commands from the main memory and relay them to the Configuration Register File.

The CPU pre-assembles Command Lists (CL) prior to submitting them to the Command List Processor for execution, while a single Command List can be submitted multiple times. This approach alleviates the CPU from recalculating drawing operations for repetitive tasks, resulting in more efficient resource utilization.

The steps for writing commands to the Configuration Registers through the Command List Processor are the following:

1. The CPU assembles a Command List, through the GFX Library.
2. The CPU submits the Command List for execution. The Command List Processor is informed of a pending Command List.
3. The Command List Processor reads the Command List from the System Memory.
4. The Command List Processor relays the commands to the Configuration Register File.

22.3.2.3 Rasterizer

The GPU can draw a multitude of geometrical shapes called Geometric Primitives, such as lines, rectangles, triangles and quadrilaterals. The Rasterizer Unit reads the coordinates of the primitives' vertices and feeds the rest of the graphics pipeline with the fragments contained in the geometry. A fragment contains information concerning a single pixel. This information includes raster position (coordinates), texture coordinates, interpolated color and alpha values.

The Rasterizer can draw:

- Pixel Drawing
- Line Drawing (at any direction)
- Filled Rectangles
- Quadrilaterals
- Triangles

In addition, the Rasterizer handles clipping, that is dropping fragments that are outside the effective drawing area and back-face culling, that is dropping entire primitives that are considered to be non-visible, like the rear looking faces of a cube. If a pixel resides inside or outside of the geometry primitive, is determined by the value of E which is positive inside the geometry primitive and negative outside of it.

The pixel's edge function value is calculated for each line of the geometry primitive (e.g. 3 times for a triangle) using the following equation:

$$E = A \cdot x + B \cdot y + C.$$

Although the equation requires two multiplications and two additions, since the variation is always one pixel on the x or y axis, this is reduced to a single accumulator.

The color variances are also calculated in a similar way, where each of the RGBA components is linearly interpolated across the geometry of the primitive. The Rasterizer determines the coverage value of each pixel as a function of the pixel center to the closest edge distance.

Transformations are performed using matrix multiplication. The Vector Matrix Multiplier multiplies a 2x1 Vector (x, y) by a 3x3 homogeneous Matrix to produce a new 2x1 Vector (Tx, Ty). The following computation is required to calculate texel coordinates from screen coordinates:

$$Tx = \frac{t00 \cdot x + t01 \cdot y + t02}{t20 \cdot x + t21 \cdot y + t22} \qquad Ty = \frac{t10 \cdot x + t11 \cdot y + t12}{t20 \cdot x + t21 \cdot y + t22}$$

22.3.2.4 Texture Map Unit

The Texture Map Unit produces texels that sends to the Fragment Processing Core. It is fed with texture's attributes (base address, dimensions, color format) and the required coordinates. The Texture Map Unit performs some internal processing and outputs the corresponding texel. Generating a texture element requires a series of operations like wrapping (clamp, mirror, repeat e.t.c.), reading corresponding color values from memory, converting the color values to RGBA8888 format and performing filtering if necessary.

22.3.2.5 Fragment Processing Core

The Fragment Processing Core is the main processing unit of the GPU's architecture. It is a 64-bit VLIW processor which performs computations on the fragments coming from the Rasterizer Unit and on the texels coming from the Texture Map Unit and calculates the final color to a fragment. The Core is programmable through binary executables called Fragment Shaders.

22.3.2.6 Render Output Unit

The Render Output Unit (ROP) is the last stage of the Graphics Pipeline. The Fragment Processing Core feeds the Render Output Unit with the pixel's coordinates and color value. Before the color value is written to the memory, the color is converted to the Frame Buffer's format. When texture compression is used, decompression is performed while reading from the Frame Buffer and compression is performed while writing to the Frame Buffer.

Blending is performed strictly in software and requires a series of calculations between the source (foreground) and destination (background) color fragments for producing the final color, which will be written in memory. The following equations are used for the final color:

$$Fc = Sc \cdot Sf + Dc \cdot Df \qquad Fa = Sa \cdot Sf + Da \cdot Df$$

The Color and Alpha values range from 0 to 1, therefore each calculation result is also clamped to the same range. The available Blend Factors and the resulting RGBA values are listed in Table 18. Figure 48 shows the effect of the blending modes.

Table 15: Blend Factors

Blending mode Name	Blend Factors (Sf or Df)	RGBA Value
DSBF_ZERO	0	0, 0, 0, 0
DSBF_ONE	1	1, 1, 1, 1
DSBF_SRCCOLOR	Sc	$Rsrc, Gsrc, Bsrc, Asrc$
DSBF_INVSRCCOLOR	$(1 - Sc)$	$1 - Rsrc, 1 - Gsrc, 1 - Bsrc, 1 - Asrc$
DSBF_SRCALPHA	Sa	$Asrc, Asrc, Asrc, Asrc$
DSBF_INVSRCALPHA	$(1 - Sa)$	$1 - Asrc, 1 - Asrc, 1 - Asrc, 1 - Asrc$
DSBF_DESTALPHA	Da	$Adst, Adst, Adst, Adst$
DSBF_INVDESTALPHA	$(1 - Da)$	$1 - Adst, 1 - Adst, 1 - Adst, 1 - Adst$
DSBF_DESTCOLOR	Dc	$Rdst, Gdst, Bdst, Adst$
DSBF_INVDESTCOLOR	$(1 - Dc)$	$1 - Rdst, 1 - Gdst, 1 - Bdst, 1 - Adst$
DSBF_CONSTCOLOR	Cc	$Rconst, Gconst, Bconst, Aconst$
DSBF_CONSTALPHA	Ca	$Aconst, Aconst, Aconst, Aconst$
DSBF_UNKNOWN	0	0, 0, 0, 0

Sc:	**Source Color**	Dc:	**Destination Color**
Sa:	**Source Alpha**	Da:	**Destination Alpha**
Sf:	**Source Blend Factor (multiplier)**	Df:	**Destination Blend Factor (multiplier)**
Fc:	**Final Color**	Cc:	**Constant Color**

Sc:	**Source Color**	Dc:	**Destination Color**
Sa:	**Source Alpha**	Da:	**Destination Alpha**
Sf:	**Source Blend Factor (multiplier)**	Df:	**Destination Blend Factor (multiplier)**
Fc:	**Final Color**	Cc:	**Constant Color**

Fa:	**Final Alpha**	Ca:	**Constant Alpha**

and the RGBA values are noted as follows:

$Rsrc$:	**Source Red value**	$Rdst$:	**Destination Red value**
$ttsrc$:	**Source Green value**	$ttdst$:	**Destination Green value**
$Bsrc$:	**Source Blue value**	$Bdst$:	**Destination Blue value**
$Asrc$:	**Source Alpha value**	$Adst$:	**Destination Alpha value**

Rconst: **Constant Red value**
Gconst: **Constant Green value**
Bconst: **Constant Blue value**
Aconst: **Constant Alpha value**

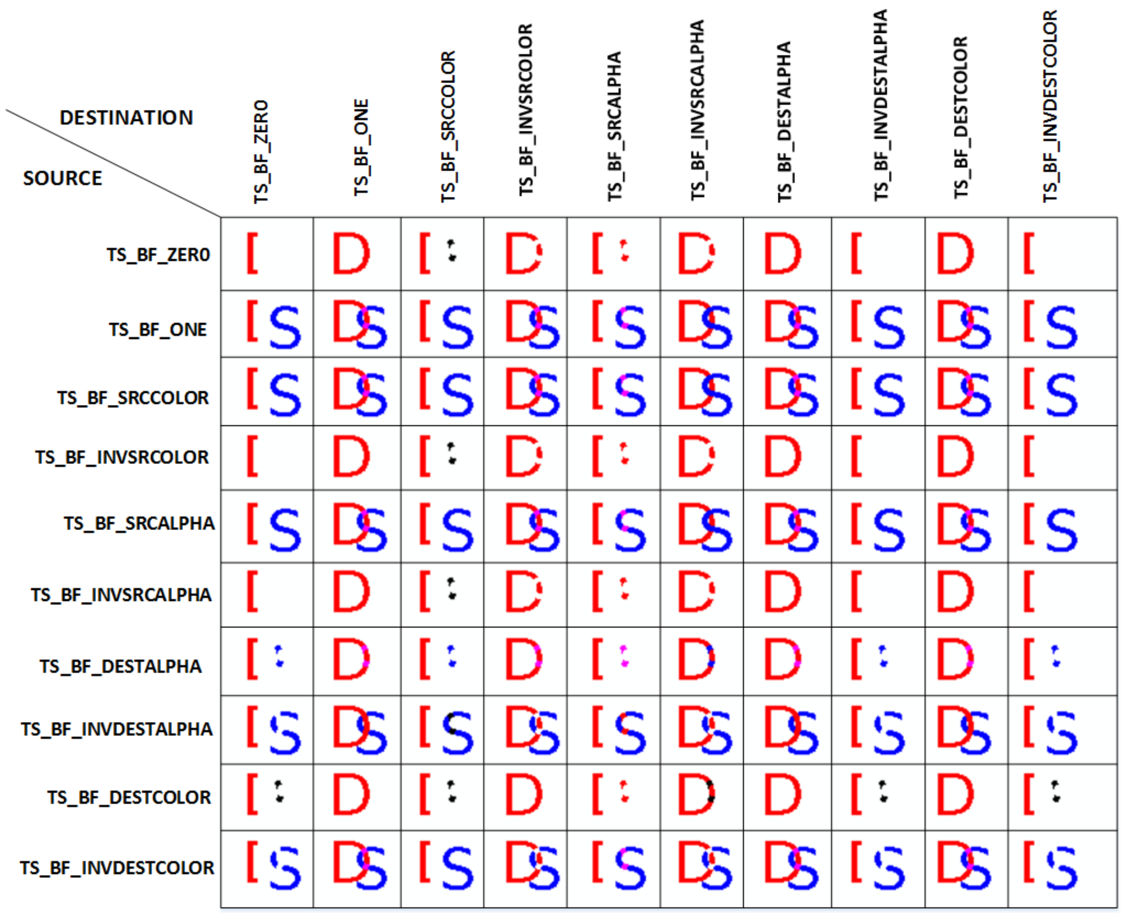


Figure 45. Blending Modes

22.3.3 Frame buffer Compression

- Framebuffer compression operates in screen blocks (4x4 pixel blocks) and, depending on the configuration, achieves TSC™4, TSC™6 and TSC™6a lossy, fixed-ratio compression.
- TSC™4 is a 6:1 compression (4 bpp)
- TSC™6 is a 4:1 compression (6 bpp)

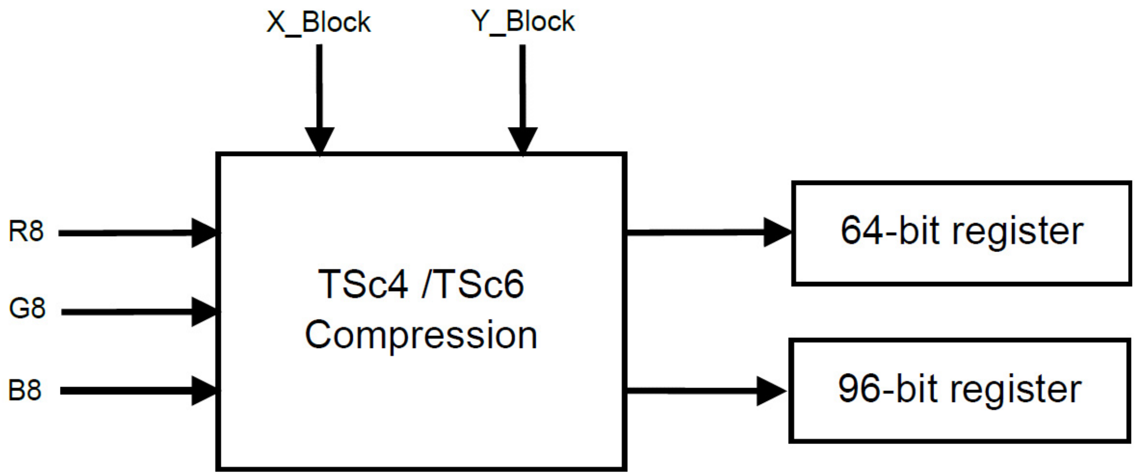


Figure 46. TSC™4 /TSC™6 Framebuffer Compression Module

22.3.4 Color Modes

The GPU supports multiple color formats. The most common color formats that are supported are the following.

22.3.4.1 Grayscale 8-bit

Values range from 0 (black) to 255 (white)

L7	L6	L5	L4	L3	L2	L1	L0

22.3.4.2 RGBA 8888 32-bit

R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0	A7	A6	A5	A4	A3	A2	A1	A0

22.3.4.3 ARGB 8888 32-bit

A7	A6	A5	A4	A3	A2	A1	A0	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

22.3.4.4 RGBA 5551 16-bit

R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	A0

22.3.4.5 RGB 565 16-bit

R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

22.3.4.6 YUYV 32-bit 2-pixels

Y07	Y06	Y05	Y04	Y03	Y02	Y01	Y00	U7	U6	U5	U4	U3	U2	U1	U0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	V7	V6	V5	V4	V3	V2	V1	V0

22.3.4.7 Custom Formats

Custom formats can be added by updating the ts_formats.v file.

22.3.4.8 Color Expansion

The internal format is always on RGBA8888 32-bit format. Therefore, lower order color formats are expanded to 8-bits per color channel. This is achieved by high-order bit replication. For example, a 5-bit color format is constructed as follows:

$$C[7:0] = \{C[4:0]; C[4:2]\}$$

22.4 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about graphics development and GPU Module operations.

23. Display Serial Interface (DSI)

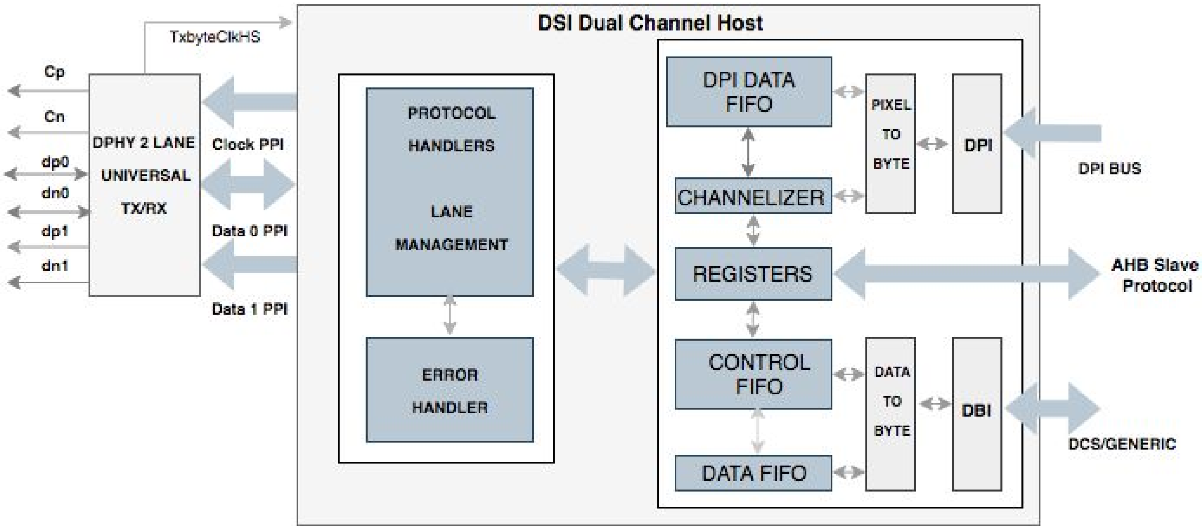


Figure 47. DSI Controller Block Diagram

23.1 Features

The DSI on the Apollo4 Blue SoC supports the features listed below.

- Standard D-PHY transceiver compliant to MIPI Specification
- Type1 display architecture in command mode
- Can be programmed to support command mode in single channel mode
- Generic read and write commands
- Low power data transfer for both DBI/generic
- Pixel formats of types:
  - 16bpp [RGB565]
  - 18bpp [RGB666] and [Loosely packed RGB666]
  - 24bpp [888RGB]
- Recovery from contention
- Timers and recovery schemes to come out of mode fault errors
- Watchdog timers to monitor D-PHY activity:
  - in low power mode
  - in high speed mode
  - during turn-around
- Interrupts to report protocol errors and expiry of timers
- Programmable device initialization timers
- Programmable maximum return packet size command
- One PHY data lane
- DBI interface for DCS commands and data transfer
- Data lane switching to low power mode during idle time
- Signals tearing effect
- Ultra low power mode switching
- Bus turn-around
- EOT disabling capacity to suit backward compatible displays
- Clock stop enabling feature during idle time
- Added support for deskew calibration
- Supported display resolutions:

- QCIF
- QVGA
- CIF
- VGA

The DSI is compliant with the following standards:

- DSI MIPI specification for Display Serial Interface (Version1.3.1)
- D-PHY standard MIPI specification (Version1.2)
- MIPI Alliance Standard for Display Bus Interface version2.0 0
- MIPI Alliance Standard for Display Command Set Version 1.02

23.2 Functional Overview

The Display Serial Interface bus (DSI) on the Apollo4 Blue SoC is a type of serial bus that enables transfer of data between a transmitter device and a receiver device. The DSI device has a point-to-point connection with DSI devices via D-PHYs as shown in Figure 48.

Please refer to the DSI registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

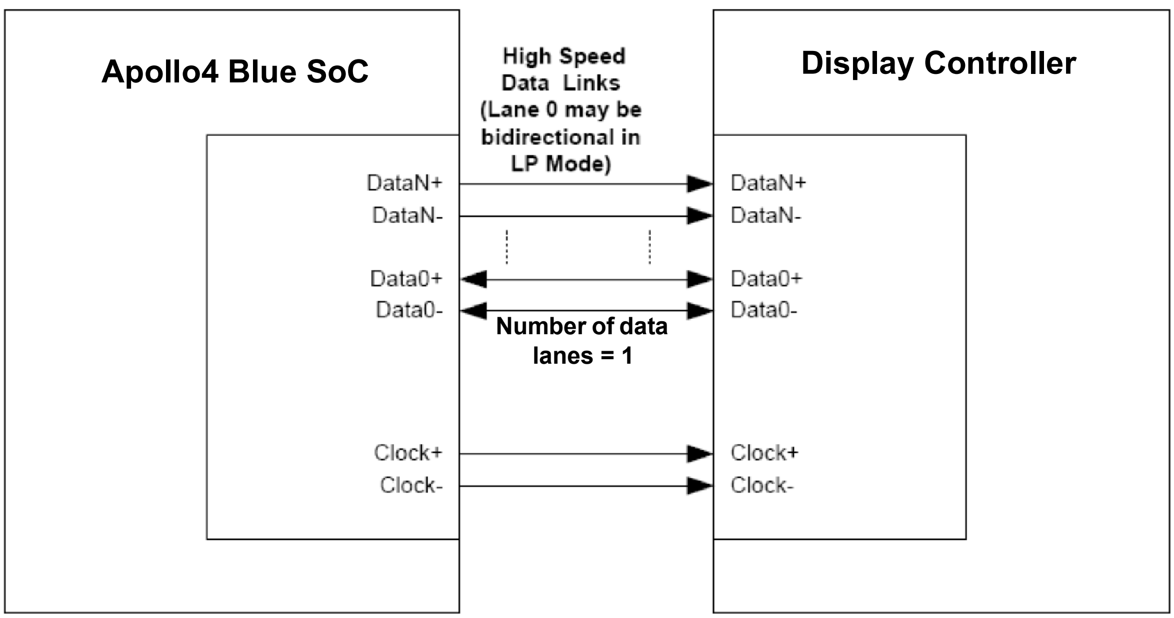


Figure 48. Display Serial Interface Bus with DSI Devices

The DSI module is configured to specify the interface and provide a connect between the MCU and a peripheral such as a display module. It is built on existing MIPI Alliance standards by adopting pixel formats, controlling pins and a command set specified in DBI-2 and DCS standards.

The D-PHY's data lane signals are transferred point-to-point as differential signals using one signal lane and a clock lane. There are two signaling modes: high speed mode that operates at a rate of 500 Mbps and a low power mode (LP) that operates at a lower transfer rate of 10 Mbps. The mode is set to a low power mode and a stop state at start up / power up. Depending on the desired data transfer type, the lanes switch between high and low power modes. High speed data transfer is unidirectional and data transfer at low speed can be unidirectional or bidirectional.

DSI devices operate in a layered fashion. There are 4 layers identified both at receiver and transmitter ends. Figure 49 shows the layers in the DSI data transfer model.

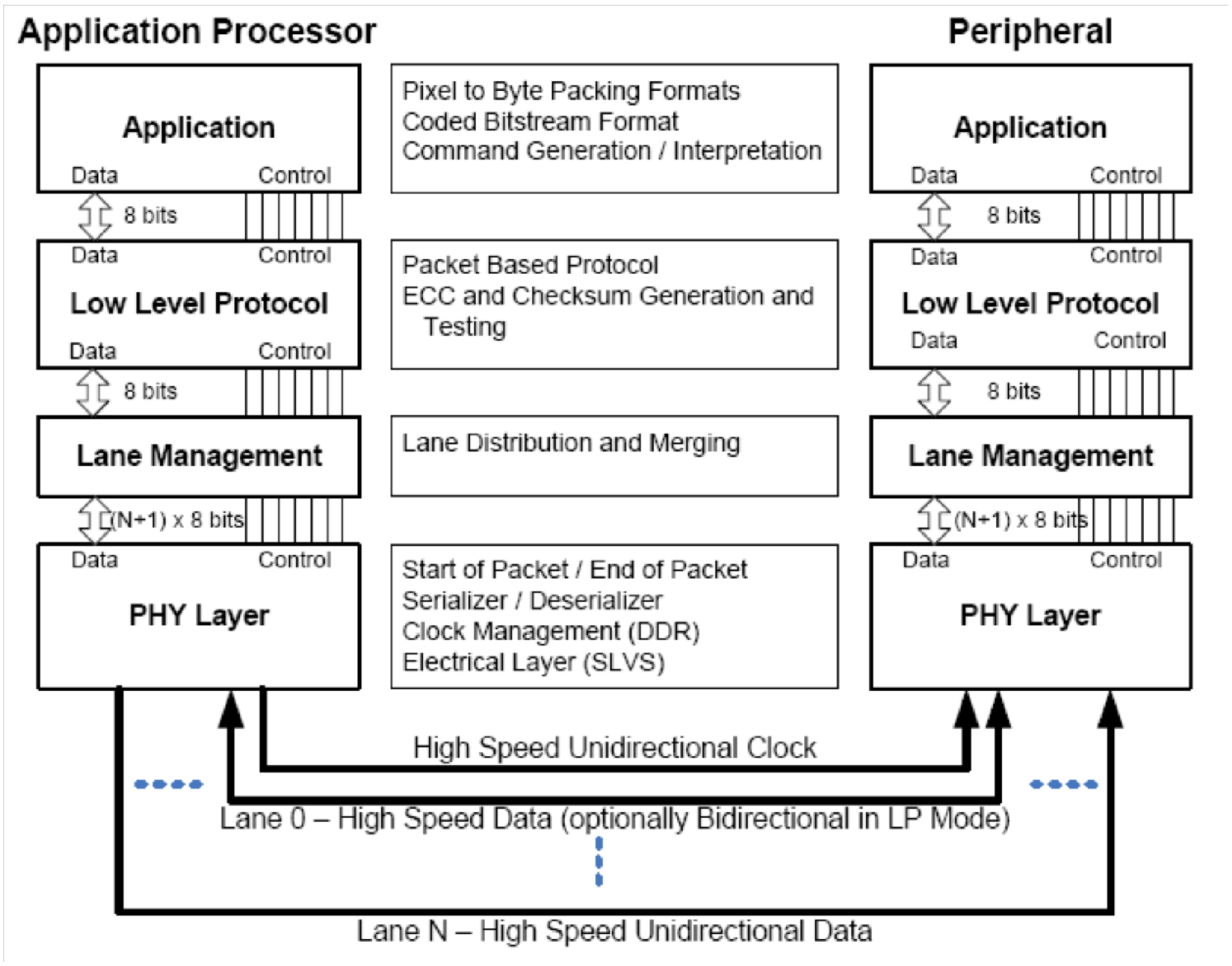


Figure 49. Layers in the DSI Data Transfer Model

PHY Layer: An embedded electrical layer that sends and detects start-of-packet and end-of-packet signaling on the data lanes. It has a serializer and de-serializer unit to dialogue with the PPI / lane management unit. It also has clock divider unit to source and receive clock during different modes of operation.

PPI / Lane Management Unit: This layer does the lane buffering and distributes the data in the lanes as programmed in a round robin manner and also merges them to stream line to the LLP/ PLI unit.

PLI / Low Level Protocol Unit: This layer packetizes as well as de-packetizes the data with respect to channels, frames, colors and line formats. There is an ECC generator and corrector unit to recover the data free from errors in the packet headers. It has a CRC checker or CRC generator unit to pack the payload data with CRC checksum bits for payload data protections.

Application: This layer describes higher-level encoding and interpretation of data contained in the data stream. Depending on the display subsystem architecture, it may consist of pixels having a prescribed format, or of commands that are interpreted by the display controller inside a display module.

23.3 Hardware Design Guidelines

The following sub-sections provide design guidelines for the use of the Apollo4's DSI PHY. Also, please consult the DSI section of the Electricals for voltage, power and timing requirements of the PHY.

23.3.1 System Power Sequencing for DSI TX Interface

The power sequence for the DSI TX interface is contingent on the PHY(s) used in the system. The sections below address the proper sequences which include powering VDD18 only or with VDDUSB33 (3.3 V) and VDDUSB0P9 (0.9 V) supplies, and cover the cases for DSI TX Interface (D-PHY) only, and both USB PHY and D-PHY in the system.

Refer to the DSI PHY section in the Electricals for supply voltage specifications.

23.3.1.1 DSI PHY Only

23.3.1.1.1 Power Tree

The recommended power tree for this configuration is as follows:

1. VDD18 is powered by LDO rails with output discharge and ON/OFF control over I^2C or GPIO.

The following power-on (default) state is allowed: VDD18 is OFF

NOTE
On Apollo4 and Apollo4 Blue, powering VDD18 without powering the DSI TX/D-PHY internal power rails results in uncontrolled current leakage to VDD18 and may lead to long-term reliability issues. This uncontrolled current leakage to VDD18 has been resolved in Apollo4 Plus and therefore turning power on and off to VDD18 in the following sequences is not needed.

23.3.1.1.2 DSI TX Initialization Sequence

The sequence below assumes that the SoC is powered on. During booting to the application, the following actions should be performed.

1. Enable power to DSI TX and D-PHY (VDDF_DSIPHY_SW).
2. Holding D-PHY in reset, enable power to the external VDD18 rail.
3. Configure DSI TX to desired mode.

For power saving put DSI TX to ULPS state during periods of inactivity, as D-PHY in LP STOP mode consumes high power from the VDD18 rail (~2.5mA). DPHY's bias, LDOs and PLL blocks can be disabled through trim registers upon ULPS PHY entry:

- DSI->AFETRIM3 |= 0x00038000; //
- DSI->AFETRIM2 |= 0x0000001C; //

23.3.1.1.3 DSI TX Shutdown Sequence

On terminating DSI TX / D-PHY operation, the SoC should follow the below shut-down sequence:

1. Disable power to the external VDD18 rail.
2. Disable power to D-PHY (VDDF_DISPHY_SW).

23.3.1.2 USB and DSI PHYs

Follow the system power sequencing recommendations for the devices utilizing both DSI and USB interfaces in section “USB and DSI PHYs” on page 133.

23.4 Additional Information

Please consult the Apollo4 Family Programmer’s Guide for additional information about DSI Module operations.

24. PDM-to-PCM Converter Module (PDM)

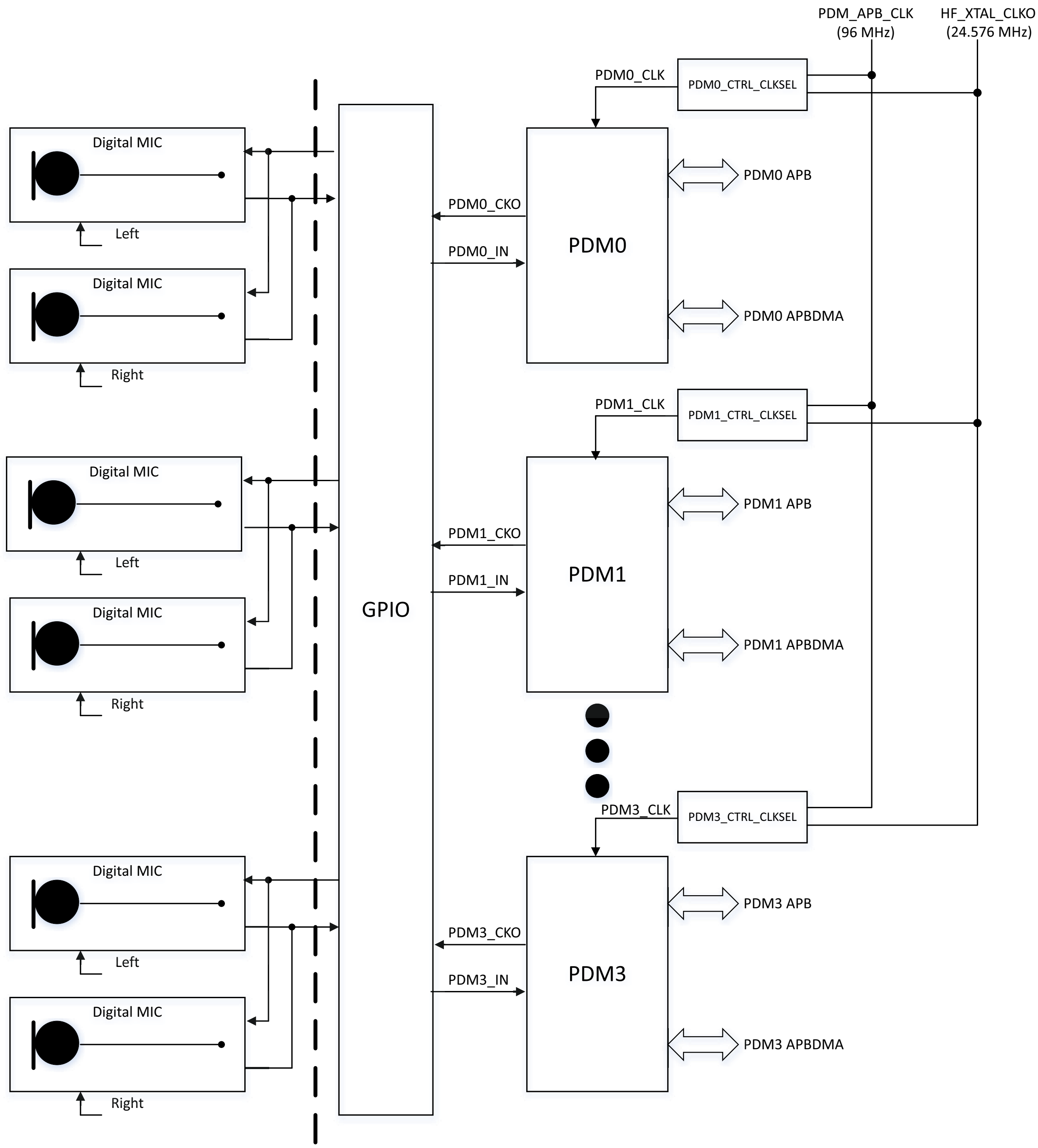


Figure 50. PDM Instances within Audio Subsystem (4 PDMs Shown)

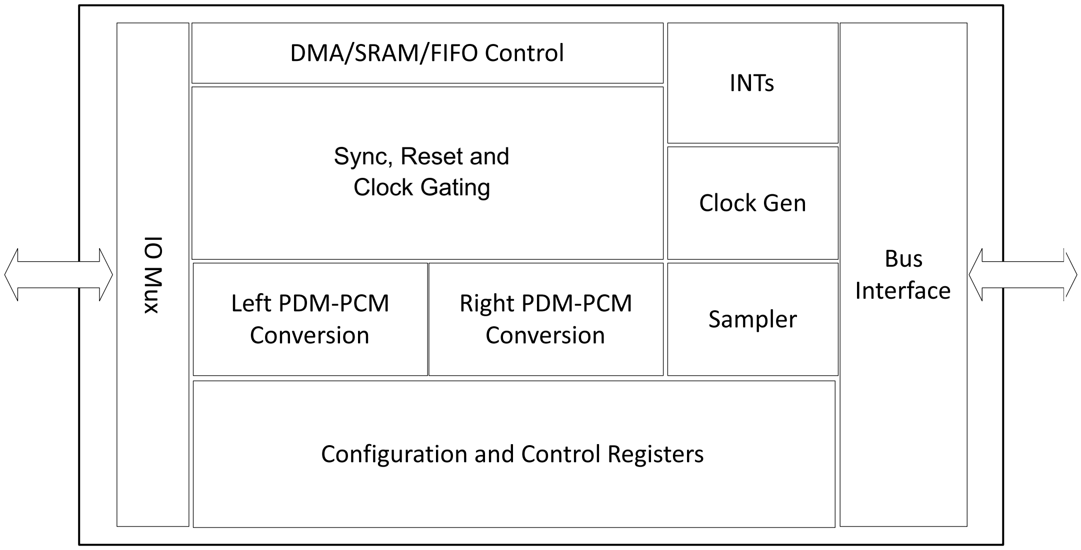


Figure 51. PDM Block Diagram

24.1 Features

The Pulse Density Modulation (PDM) to Pulse Code Modulation (PCM) Converter Module, referred to throughout as PDM, features a low power stereo/mono PDM-to-PCM converter with register programming. It is targeted for digital microphone voice/audio recording applications.

The module operates in dual mode (stereo or mono). In stereo mode, the PDM converts 1-bit stereo pulse-density modulated (PDM) bit stream data from external digital microphones into 24-bit pulse-code modulated (PCM) data for base-band processing. In default operation, the PDM data sampled on the rising-edge of digital microphone clock is assumed to be left channel input, while data on the falling-edge is assumed to be right channel input. Optional channel swap is available through register setting. In mono mode, only the left channel PCM output is valid while the right channel output is zero (no toggling).

The PDM-to-PCM converter supports data sampling rate at 16 kHz in default setting for voice application. It is capable of supporting output sampling rates (F_S) at 8, 16, 48, 96 kHz and up to 192 kHz at different master clock conditions. After input sampling, the PDM data bits are fed into digital filters for data conversion and gain amplification.

The PDM module provides the following features:

- Support Stereo/Mono Dual Mode PDM-to-PCM Conversion
- 1-bit PDM (pulse-density modulated) input for up to 4 pairs of microphone outputs
- 24-bit PCM conversion/output at up to 192 kHz sample rate
- Support Digital Microphone Clock at 512 kHz, 1.024 MHz, 2.048 MHz, 2.45 MHz, 3.072 MHz
- PCM Sampling Rate: 8 kHz, 16 kHz, 32 kHz, 48 kHz, 96 kHz, 192 kHz
- PGA Gain: -12dB - +34.5dB gain with 1.5dB/Step
- High Performance Mode
  - 110dB SNR, BW=20 kHz (A-weighted)
  - -105dB THD+N, BW=20 kHz (A-weighted)
- Mid Performance
  - 107dB SNR, BW=6.7 kHz (A-weighted)
  - -101dB THD+N, BW=6.7 kHz (A-weighted)
- Reduced Performance mode
  - 88dB SNR, BW=6.7 kHz (A-weighted)

  - -83dB THD+N, BW=6.7 kHz (A-weighted)
- Power Down Mode support

24.2 Functional Overview

The Apollo4 Blue SoC integrates a PDM-to-PCM Converter module designed for digital voice applications. The conversion path contains front-end left/right channel data sampling, digital filters and PGA gains for each channel. The digital filters convert single bit PDM data into 24-bit PCM data. The streamed data volume can be programmed through internal registers.

Please refer to the PDM registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

24.3 PDM-to-PCM Converter Clocking Mechanism

Table 16 below shows the PDM bit data sampling clock (PDMA_CKO) as a function of F_S and OSR for the various operating modes, at a PDM_CLK of 24.576 MHz.

Table 16: PDMA_CKO and OSR Settings for Different Sampling Frequencies

OPERATING MODE	F_{PDMA_CKO} (MHz)	F_S (kHz)	OSR	DIV_ MCLKQ [1:0]	MCLKDIV [3:0]	SINCRATE [6:0]	SINAD (dB)	DR (dB)
High Performance Mode	6.144	96	64	1	1	32	103	110.8
	3.072	48	64	1	3	32	105.5	108.7
	3.072	24	128	1	3	64	122.8	120.9
	3.072	16	192	1	3	96	116.1	120.4
	1.536	16	96	1	7	48	115.4	120.5
Reduced Performance Mode	3.072	96	32	1	3	16	86	87.9
	1.536	48	32	1	7	16	83.2	88.8
	0.768	16	48	1	15	24	89.7	97.2
Mid Performance Mode	1.536	24	64	1	7	32	101	107.6
	1.024	16	64	1	11	32	100	106.8

Notes:

1. Assumes PDM_CLK of 24.576 MHz.
2. The above frequency combinations are recommended values, where **DIV_MCLKQ = 2'b01**. User may determine other proper values according to actual master clock rate and digital microphones implemented in system design.
3. **SINAD** means ratio of signal to noise plus the first N harmonics of THD.
4. **DR** means dynamic range, which is measured as SINAD - (-60dB) in this table.

The module's master input clock (PDM_CLK) is generated by the SoC clock generator. The PDM bit data sampling clock for external digital microphones (PDMA_CKO) and the filters' internal operating clocks is generated internally by the module. The relationship between PDMA_CKO, the master clock, PDM_CLK, and the sampling frequency F_S is as shown below. The PDMA_CKO clock may be delayed by setting register PDMCKO_DLY for a clock phase shift during bit data sampling.

The PDM input clock is divided down by DIV_MCLKQ to generate the internal clock for the PGA and PDM-to_PCM converters, MCLKQ, as:

- $F_{MCLKQ} = F_{PDM\_CLK} / (DIV\_MCLKQ + 1)$

where the clock gating to left and right channels is:

- $F_{MCLK\_L} = F_{MCLK\_R} = F_{MCLKQ}$

The resulting PCMA_CKO frequency can be set as:

- $F_{PDMA\_CKO} = F_{MCLK\_L}$ / (MCLKDIV+1) = F_S x 2 x SINCRATE

NOTE
Sinc decimation rate, CORECFG0_SINCRATE, must be set to a value within the range of 16-64, or to 96.

and if the frame rate, also known as the baseband sampling frequency, F_S = 16 Ks/s and the decimation rate setting (SINCRATE) is set to 16, then:

- F_{PDMA_CKO} = 16 Ks/s x 2 x 16 = 512 kHz

The oversampling rate, OSR, or decimation rate, then becomes:

- OSR = F_{PDMA_CKO} / F_S = 2 x SINCRATE

and

- $F_{MCLK\_L} = F_{MCLK\_R} = F_S$ X 2 x SINCRATE x (MCLKDIV+1)

NOTE
Regarding the default selection of the HFRC2 to clock the PDM module, there is the possibility of an asynchronous shutdown of the HFRC2 clock divider by the internal hardware, causing a glitch when the requesting peripheral stops requesting the HFRC2. The HFRC2 must be forced on not only when HFRC2 is being selected and while being used as the clock source but also whenever the clock source is being changed regardless of the new clock source being selected. The HFRC2 is forced on by setting the CLKGEN_MISC_FRCHFRC2 bit. The sequence for changing the clock source regardless of clock selection is to first force HFRC2 on by setting the CLKGEN_MISC_FRCHFRC2 bit, select the clock source for the module, clear the CLKGEN_MISC_FRCHFRC2 bit only if HFRC2 is NOT selected, and then engage the peripheral. If HFRC2 is the clock source, then shutting the module down cleanly requires switching to HFRC, for example, and then disabling the HFRC2 by clearing the CLKGEN_MISC_FRCHFRC2 bit.

24.3.1 Clock Gating and Data Synchronization

For low power implementation, PDM_CLK is sourced from the MCU and divided down internally for each conversion channel with clock gating. Figure 52 shows the clock tree and internal synchronization from PDM_CLK clock domain to the MCLKQ clock domain. The DIVIDER block is user-controlled by the

DIVMCLKQ register field to generate MCLKQ, which in turn is divided by MCLKDIV and fed to the CLOCK GENERATOR block to generate PDMA_CKO.

MCLK_L is always on in both mono and stereo mode. That is, in mono right-channel operation, it actually uses the left channel of PDM-to-PCM core conversion for recording conversion process. LRSWAP must be set to “1” in mono right mode operation, and “0” in mono left mode operation. Please also refer to “Operating Mode” Section for more detail description.

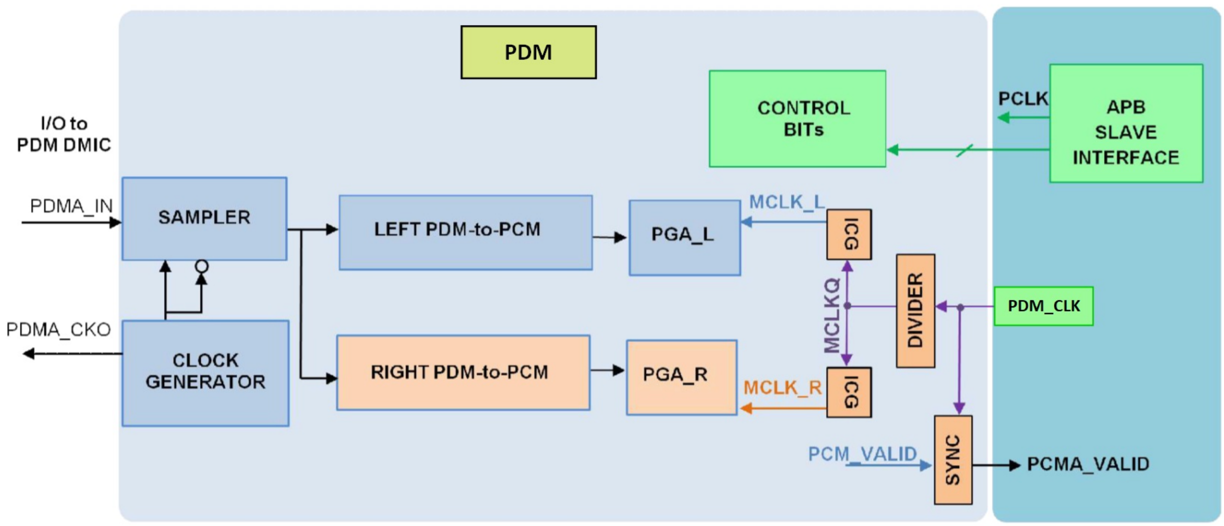


Figure 52. Clock Path and Data Synchronization Diagram

Figure 53 illustrates the clock gating scheme for MCLK_L and MCLK_R internal master clocks through the ICG (integrated clock gating) cells. PDM_LEFT_EN and PDM_STEREO_EN signals are controlled by the setting of the CORECFG1_PCMCHSET field.

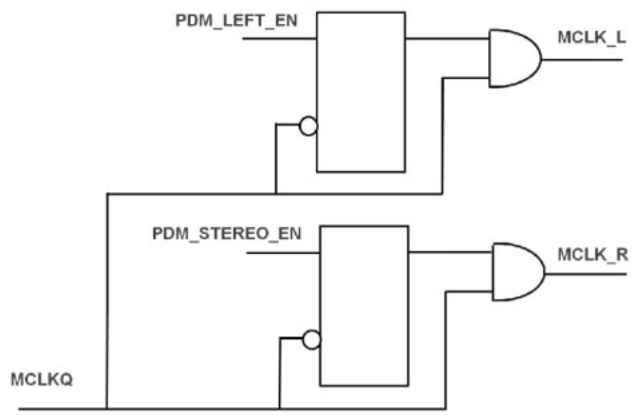


Figure 53. PDM Converter Core Local Clock Gating

24.4 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about PDM-to-PCM Converter Module operations, where the following topics are covered:

- Operating Modes
- Digital Volume Control and Soft Mute
- Low pass and high pass filters

25. Low Power Analog Audio Interface

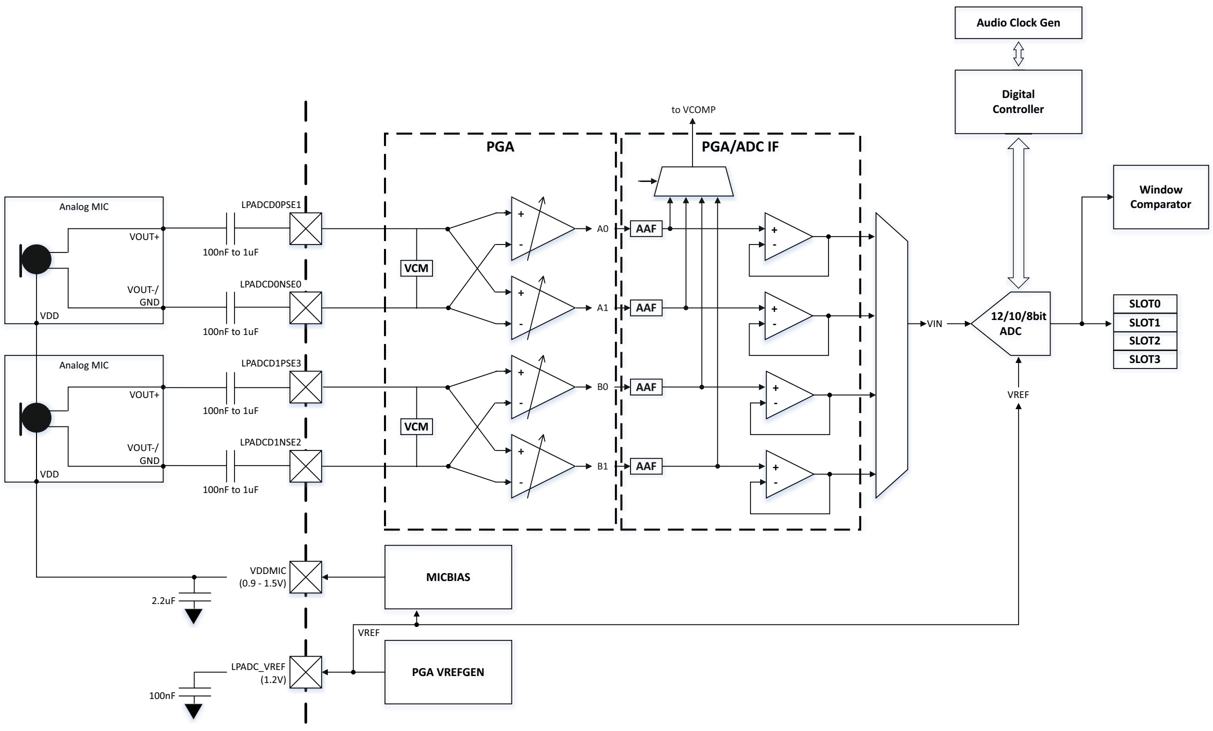


Figure 54. Low Power Analog Audio Block Diagram

25.1 Features

The Low Power Analog Audio Interface is comprised of 4 channels of Programmable Gain Amplifiers (PGAs), 12-bit 4-channel Audio ADC (AUDADC), and 1 low power microphone bias (MICBIAS) as shown in Figure 54.

Key features of the PGAs include:

- Programmable gain for AC-coupled audio inputs (20 Hz - 20 kHz) to drive AUDADC
- Audio inputs may be microphone or line inputs
  - Single Ended (SD)
  - Pseudo Differential (PD)
  - Fully Differential (FD)
- Full Scale Voltage
  - SE/PD: 0.5 Vrms
  - FD: 1 Vrms
- Gain steps supported: 0-24 dB in 0.5 dB increments
- Set input common-mode for active and sleep mode operation
- Implicit 2/3 attenuation to fit 1.2 V ADC full scale

Key features of the AUDADC include:

- Reconfigurable Successive Approximation Register (SAR) ADC
- 4 dedicated single-ended input channels from four PGA sources
- Input Range: 0 V to 1.2 V
- Configurable automatic low power control between scans
- Configurable for 12 / 10 / 8 bit ADC Precision Modes
- Configurable sampling time
- Uses 1.2 V external reference with internal buffer
- Single shot, repeating single shot, scan, and repeating scan modes
- Variable sample tracking time, configurable on per-slot basis
- User-selectable clock source for variable sampling rates
- Automatically accumulate and scale module for hardware averaging of samples
- 16-entry FIFO and DMA capability for storing measurement results and maximizing SoC sleep time
- Multiple Interrupt Support:
  - FIFO full
  - FIFO almost full
  - Scan Complete
  - Conversion Complete
  - Window Incursion
  - Window Excursion
  - Various DMA-related notifications
- Window comparator for monitoring voltages excursions into or out of user-selectable thresholds
  - Unsigned mode support ONLY
- Supports signed data mode by way of AUDADC_ADCCFG_DATAFMT
- Settable sampling/tracking time per-slot
- ADC-internal trigger timer providing low-jitter periodic repeated triggers
- Additional delays configurable via ADC registers

Key features of MICBIAS include:

- MICBIAS provides user-programmable regulated (0.9 V to 1.5 V) supply to analog MEMS microphones
- Performance Summary:
  - 200 µA max load current with 2.2 µF capacitor
  - 560 nA quiescent current
  - Typical PSR (from VDDAUD)
    - 34 dB @ 1 kHz
    - 15 dB @ 20 kHz
    - Startup < 1 ms

25.2 Functional Overview

The Apollo4 Blue SoC integrates a sophisticated 12-bit successive approximation Analog to Digital Converter (ADC) block for sensing both internal and external voltages. The Audio ADC provides four separately managed conversion requests, called slots which are serially sequenced. The result of each conversion request is delivered to a 16-deep FIFO. Firmware can utilize various interrupt notifications to determine when to collect the sampled data from the FIFO or from a buffer written by DMA. This block is extremely effective at automatically managing its power states and its clock sources.

Please refer to the Audio ADC registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

25.2.1 Clock Source and Dividers

The Audio ADC runs off of the HFRC, HFRC2 or a 24.576 MHz crystal clock source. When the Audio ADC block is enabled and has an active scan in progress, it requests a clock source. There is an automatic hardware hand shake between the clock generator and the Audio ADC. If the Audio ADC is the only block requesting an HFRC based clock, then the HFRC will be automatically started. The Audio ADC can be configured to completely power down the HFRC between scans if the startup latency is acceptable or it can leave the HFRC powered on between scans if the application requires low latency between successive conversions. The Audio ADC supports a HFRC clock frequency of 48 MHz, a HFRC2 clock frequencies of 48 MHz, or a nominal 24.567 MHz from a XTALHS crystal.

> NOTE
>
> Regarding the selection of HFRC2 to clock the Audio ADC module, there is the possibility of the asynchronous shutdown of the HFRC2 clock divider causing a glitch when the requesting peripheral stops requesting the HFRC2 and allows it to be shut down. The HFRC2 must be forced on when it is to be used such that it does not get powered down by the internal hardware.
>
> The HFRC2 is forced on by setting the CLKGEN_MISC_FRCHFRC2 bit. The sequence for selecting the HFRC2 as the clock source is to first force FRCHFRC2 bit on, then switch clock sources, and finally engage the peripheral.
>
> If HFRC2 is the clock source, then shutting the module down cleanly requires switching to HFRC, for example, and then disabling the HFRC2 by clearing the CLKGEN_MISC_FRCHFRC2 bit.
>
> Also, if HFRC2 is used for the Audio ADC clock source, low-jitter samples are needed which means using the Audio ADC’s internal trigger timer. This in turn requires the clock to be running while sampling is enabled.

25.2.2 4 Channel Analog Mux

As shown in Figure 54, the Audio ADC block contains a 4-channel analog multiplexer on the input port to the analog to digital converter. The analog mux channels are connected as follows:

1. Analog IN or Mic A PGA channel 0 (PGA_A0)
2. Analog IN or Mic A PGA channel 1 (PGA_A1)
3. Analog IN or Mic B PGA channel 0 (PGA_B0)
4. Analog IN or Mic B PGA channel 1 (PGA_B1)

Refer to the detailed register information below for the exact coding of the channel selection bit field for each slot.

25.2.3 Voltage Reference Source

The Apollo4 Blue SoC's Audio ADC 1.2 V voltage reference is internally generated and filtered from the LPADC_VREF pad using a 100 nF capacitor.

25.2.4 Four Automatically Managed Conversion Slots

The Audio ADC block contains four conversion slot control registers, one for each of the four slots. These can be thought of as time slots in the conversion process. When a slot is enabled, it participates in a conversion cycle. The Audio ADC’s mode controller cycles through up to four time slots each time it is triggered.

Table 17: One SLOT Configuration Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved					# Samples to Accum.			Reserved												CHANNEL SELECT				Reserved						WINDOW_COMP	SLOT_ENABLE

The channel select bit field specifies which one of the analog multiplexer channels will be used for the conversions requested for an individual slot.

Each of the four conversion slots can independently specify:

- Analog Multiplexer Channel Selection
- Participation in Window Comparisons
- Automatic Sample Accumulation

25.2.5 Sixteen Entry Result FIFO

All results written to the FIFO have exactly the same format as shown in Table 18. The properly scaled accumulation results are written the lower half word in 14.6 format. Since each slot can produce results at a different rate, the slot number generating the result is also written to the FIFO along with the total valid entry count within the FIFO.

Table 18: FIFO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RSV	Slot Number.			FIFO Count								FIFO DATA																			

25.2.6 DMA

When enabled, the Audio ADC can use DMA to keep its FIFO serviced and transfers samples to SRAM. Generally, DMA should be used when the desired use case is autonomous recording of samples to a pre-allocated buffer in SRAM. The buffer may be byte-aligned but must be a word-multiple in size.

An additional capability of the DMA is the ability to mask FIFOCNT and SLOTNUM data from FIFO data. The DMA engine can be configured to write only samples to SRAM without the FIFOCNT and SLOTNUM data. This allows the SoC to skip the manual process of masking the potentially undesirable upper bits of each data value written to SRAM.

25.2.7 Window Comparator

A window comparator is provided which can generate an interrupt whenever a sample is determined to be inside the window limits or outside the window limits. These are two separate interrupts with separate interrupt enables. Thus one can request an interrupt any time a specified slot makes an excursion outside the window comparator limits.

The window comparison function has an option for comparing the contents of the limits registers directly with the FIFO data (default) or for scaling the limits register depending on the precision mode selected for the slots.

NOTE
Currently the only supported ADC data format is unsigned binary format, which is the default setting for the AUDADC's CFG_DATAFMT field.

25.3 Interrupts

The Audio ADC has 8 interrupt status bits with corresponding interrupt enable bits, as follows:

1. Conversion Complete Interrupt
2. Scan Complete Interrupt
3. FIFO Overflow Level 1
4. FIFO Overflow Level 2
5. Window Comparator Excursion Interrupt (a.k.a. outside interrupt)
6. Window Comparator Incursion Interrupt (a.k.a. inside interrupt)
7. DMA Complete (DCMP)
8. DMA Error (DERR)
9. DMA transfer complete
10. DMA error condition

25.4 Microphone Biasing

The Apollo4 MICBIAS circuit can be trimmed with a 6-bit trim code programmed in MCUCTRL_AUDIO1_MICBIASVOLTAGETRIM and MCUCTRL_AUDIO2_MICBIASVOLTAGETRIM. Relationship between trim values and resulting VDDMIC are as shown in Figure 55. The relationship between VDDMIC voltage and trim code is approximately described by the following formula:

VDDMIC = (0.012481V * MICBIASVOLTAGETRIM[5:0]) + 0.827913V

NOTE
MICBIAS is powered by the VDDAUDA rail (1.8 V ± 10%). Microphones that need 1.62 V or greater (up to 1.98 V) can be supplied directly using VDDAUDA. A MICBIAS bypass mode is active when MICBIASVOLTAGETRIM[5:0] is set to 0x3F. This mode enables a bypass to MICBIAS so that it can output VDDAUDA voltage, which allows the MICBIAS circuit to act as a load-switch for analog mics requiring voltage > 1.5 V.

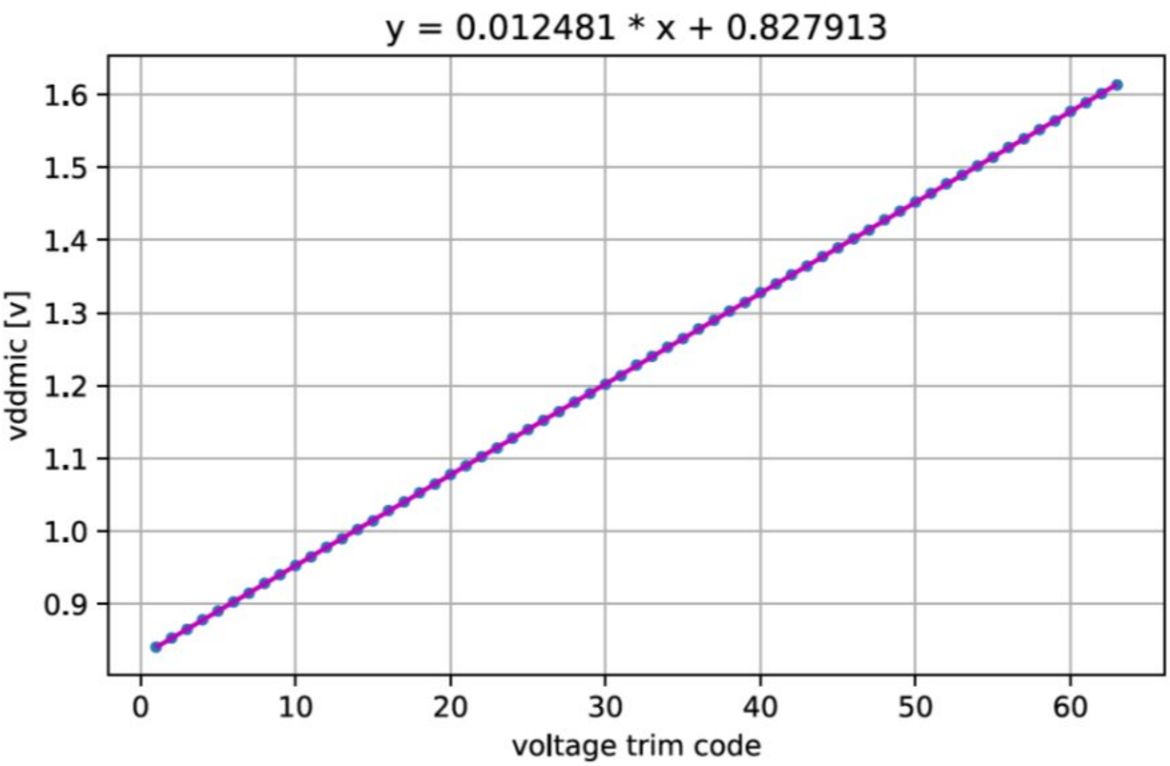


Figure 55. Mic Bias Trim Graph

25.5 Additional Information

Please consult the Apollo4 Family Programmer's Guide for additional information about Audio ADC Module operations.

26. Inter-IC Sound (I^2S)

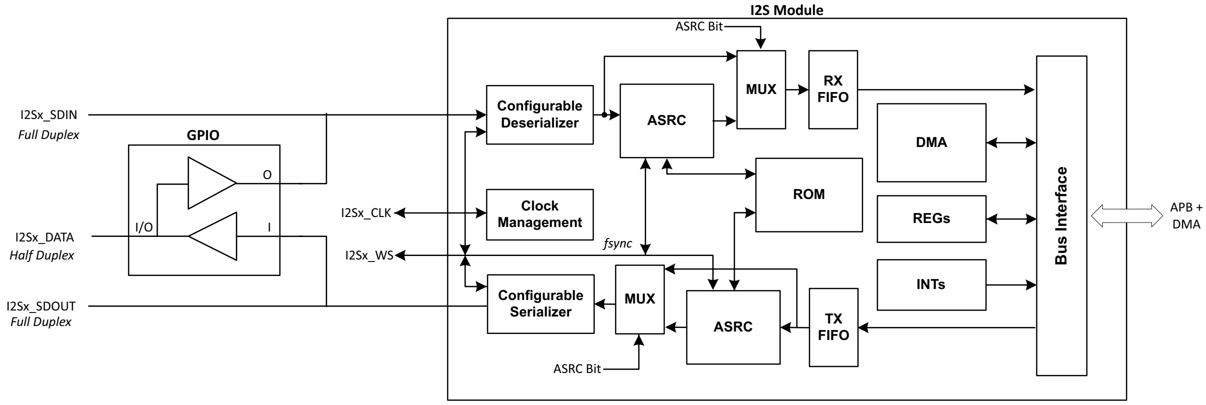


Figure 56. I^2S Block Diagram

26.1 Features

The I^2S module provides the following features:

- Inter-IC audio streaming interface
- Modes
  - I^2S Philips mode
  - I^2S right-justified and left-justified serial audio format modes
  - TDM mode
- Supported sample rates include 8, 11.025, 16, 22.05, 32, 44.1, 48, 96 and 192 kHz
- Audio sample sizes of 8,16, 24 and 32 bit
  - I^2S always sends 32 bits per channel
  - TDM has tremendous flexibility for framing and bit width
- 2 instances (IPB0/IPB1) of full-duplex I^2S (stereo TX + stereo RX) using shared CLK & FS
- Master and slave
- Optional Asynchronous Sample Rate Conversion (ASRC) on slave I^2S channels
  - 2-channel audio sample rate converter
  - Sample size: 24 bits (for internal processing, but accepts 8, 16, 24 or 32 bits)
  - Lower than -130 dB THD+N for common conversion ratios (when using 24-bit or 32-bit samples)
  - Minimum input clock frequency: FSin
  - Minimum output clock frequency: FSout
  - Extremely fast synchronization time with the input audio stream: 128 / FS_{in}
  - Latency: (FIFO_SIZE / ($2FS_{in}$)) + (2 / FS_{out})
  - Automatically adjusts to changes in both input and output sample rates
  - High input jitter tolerance: supports occasional bursts or skips of a couple of samples without sacrificing quality in practical terms
  - Input sample rate range: 8 kHz to 192 kHz
  - Output sample rate range: 8 kHz to 192 kHz
  - Maximum down conversion of 3.9:1
  - Maximum up conversion of 1:7

- Fixed FSYNC:SCLK ratio of 64:1 required
- 1 to 8 Channel TDM interface

NOTE
Sharing the TDM bus between/among multiple peripherals is not supported.

26.2 Functional Overview

The I^{2}S interface module provides the capture and transmit capability of I^{2}S digital audio data in a variety of formats and data rates, as well as an asynchronous sample rate conversion function for input or output data streams. It is a slave device which must be programmed by the MCU to function. There are 2 independent instances of the I^{2}S module on the Apollo4 Blue SoC. Each I^{2}S instance can support master or slave operation and full or half duplex operation.

Various modes and sample rates are supported to provide flexible audio data processing. DMA is supported to enable efficient transfer of data to/from SRAM. The I^{2}S modules are powered via VDDH or VDDH2, and the modules must be activated through the PWRENI2Sn field of the PWRCTRL_AUDSSPWREN power control register prior to accessing any registers within the modules. To access external devices, the GPIO module must also be programmed to allow the I^{2}S signals to propagate through the selected pins of the device. There is a variety of clock sources that can be used to control the transmit and receive ports, and the module can operate in clock master or slave mode, regardless of the direction of data flow.

Each module contains 2 major sections. One section consists of the interface logic control portion and second section contains the I^{2}S audio processing portion. The interface logic provides register and high level clock, data and interrupt control for the I^{2}S module. DMA capabilities are provided for TX, RX or full duplex RX/TX modes.

Please refer to the I^{2}S block diagram in Figure 56. Each controller supports configurable Asynchronous Sample Rate Converters, each capable of supporting stereo transmit and/or receive when configured as slave device. ASRC RX and ASRC TX perform 24-bit asynchronous sample rate conversions sharing the same ROM.

The Configurable Deserializer can be configured to convert the different possible formats of the incoming serial audio stream to a parallel interface. If the Receive FIFO is full, the newly arrived samples are dropped until there is space in the FIFO. The Configurable Deserializer should be reset before a stable serial audio signal is present at the input.

The Configurable Serializer reads the audio samples from the Transmit FIFO and converts the parallel audio stream interface to the desired output format. If the FIFO is empty, this module can be configured to repeat the last sample present in the FIFO or transmit zeros.

26.3 Additional Information

Please consult the Apollo4 Family Programmer’s Guide for additional information about I^{2}S Module operations.

27. Voltage Comparator (VCOMP)

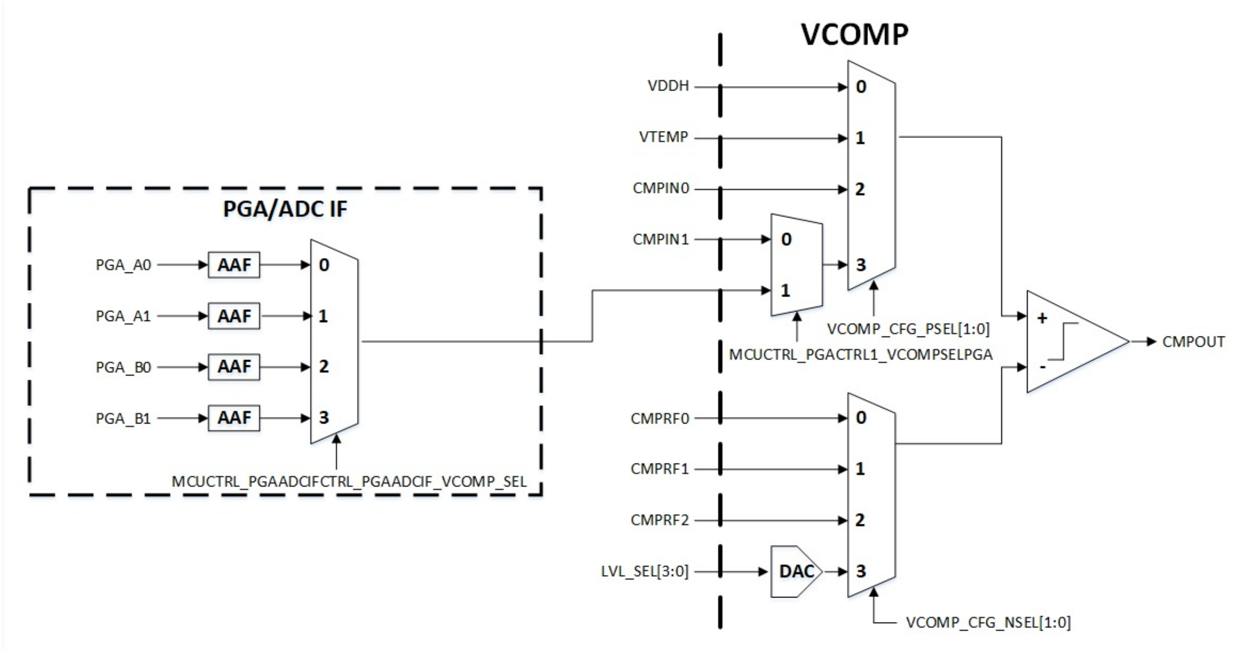


Figure 57. Block diagram for the Voltage Comparator Module

27.1 Functional Overview

The Voltage Comparator Module, shown in Figure 57, measures a user-selectable voltage at all times. It provides interrupt and software access to the comparator output with multiple options for input and reference voltages. It can be configured to generate an interrupt when the monitored voltage rises above a user-configurable threshold or when the monitored voltage drops below a user-configurable threshold.

The voltage to be monitored is selected by programming the comparator's positive terminal signal, PSEL[1:0], and may be any of:

1. The supply voltage (VDDH), or
2. The PTAT voltage from the temperature sensor (VTEMP), or
3. Two external voltage channels (CMPIN0 or CMPIN1), or
4. The filtered PGA outputs (PGA_A0, PGA_A1, PGA_B0, PGA_B1)

The reference voltage is selected by programming the comparator's negative terminal, NSEL[1:0] and may be any of:

1. Three external voltage channels (CMPRF0, CMPRF1 or CMPRF2), or
2. The internally generated reference (VREFINT)

The internal reference voltage is tuned using an on-chip DAC with level select signal LVLSEL[3:0]. When using external inputs or reference inputs, the associated pads must be configured using the GPIO function selects explained in the GPIO document section.

The Voltage Comparator CMPOUT output will remain high while the voltage at the positive input is above the voltage at reference input. The CMPOUT output will transition low when the voltage at the positive input to the comparator falls below the reference input taking into account hysteresis. The CMPOUT output is directly accessible by software by reading the CMPOUT field in the status register. The OUTHI interrupt will be set if enabled and the CMPOUT transitions high or if it is high at the time the interrupt is enabled. Similarly, the OUTLOW interrupt will be set if enabled and the CMPOUT output transitions low or if it is low at the time the interrupt is enabled.

Please refer to the VCOMP registers of the Apollo4 Blue SoC register set. The register set is delivered as part of the AmbiqSuite SDK.

28. Voltage Regulator Module

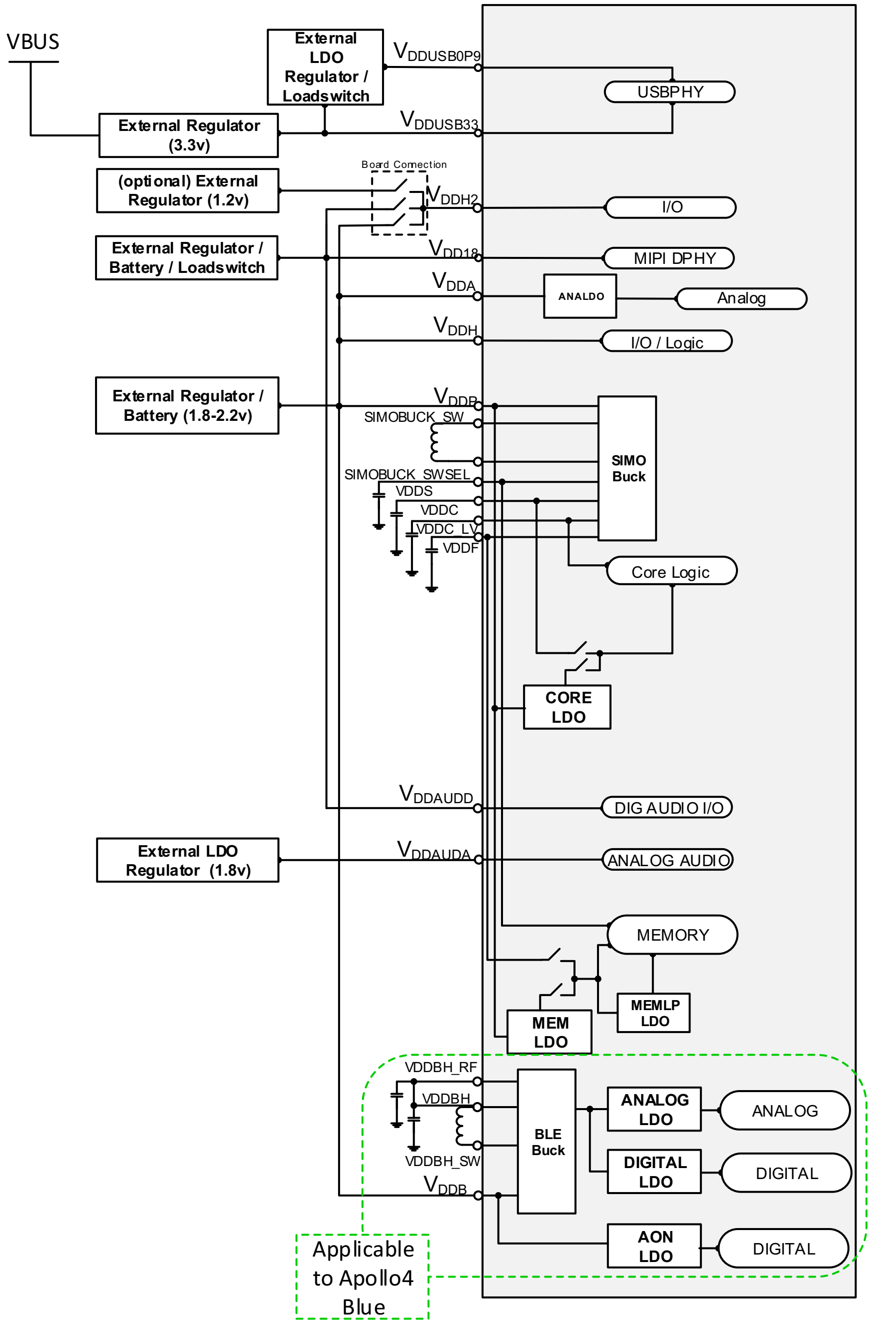


Figure 58. Block Diagram for Voltage Supplies and Regulation on Apollo4 Family

28.1 Functional Overview

The Voltage Regulator Module down-converts and regulates the supply voltage, VDD, with extremely high efficiency. A pair of Buck Converters enable down-conversion from the power supply input (e.g., a battery or external regulator) at efficiency of > 80%. With ultra-low quiescent current, the Buck Converters areoptimized for low power environments. There are also integrated low dropout linear regulators (LDOs) which are used in very low power modes and can also be utilized to provide a lower cost system solution by eliminating the need for the external inductors required in buck mode. The VDDC and VDDF capacitors are still required for the internal LDOs.

The Buck Converters and LDOs of the Voltage Regulator Module are tightly coupled to the various low power modes in the Apollo4 Blue SoC. When the device enters deep sleep mode, the Buck Converters switches into a low power mode to provide very high efficiency at low quiescent current.

28.2 SIMO Buck

The SIMO buck sources the primary supplies for the core and memory domains. This buck is a very high efficiency, single-inductor/multiple-output design. The SIMO buck must be enabled via software. This is done in the AmbiqSuite SDK using a HAL function:

am_hal_pwrctrl_control(AM_HAL_PWRCTRL_CONTROL_SIMOBUCK_INIT, 0).

Upon enabling the SIMO buck, it will be power up and stabilized through hardware control. The status of the SIMO buck can be queried via the PWRCTRL_VRSTATUS register. The SIMO buck has an efficient ultra-low power mode that is entered automatically via hardware control based on active load current of the system.

For cost/area constrained designs, the SIMO buck can be disabled and on-die LDO regulators can be used. In this configuration, the SIMO buck will remain powered down.

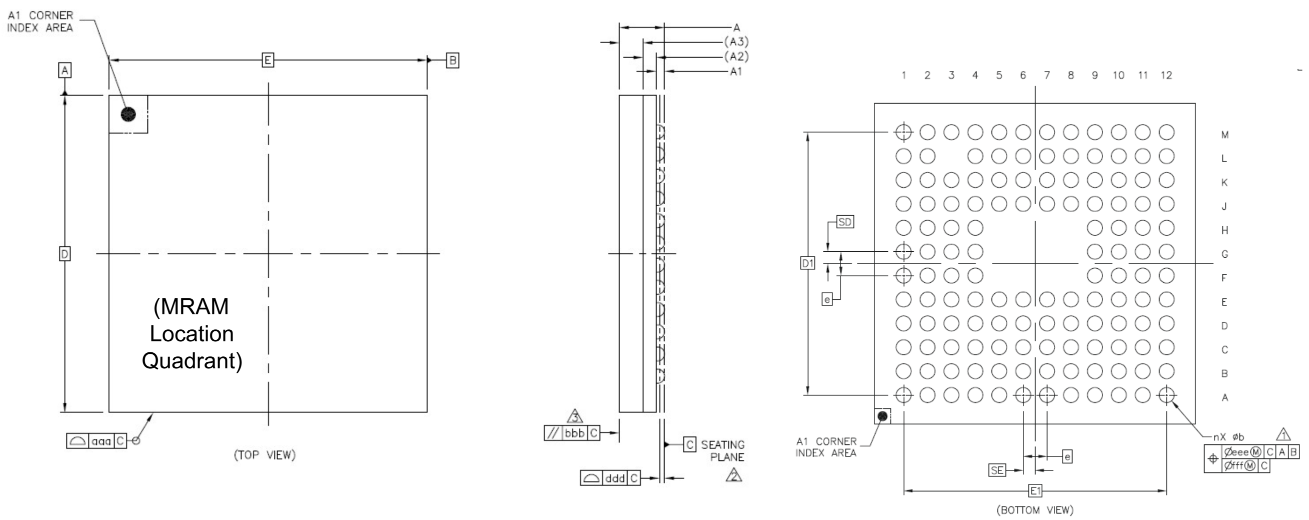
There is also a zero length detect circuit to ensure the regulated voltages from the SIMO buck do not drop out.

28.3 BLE Buck

The BLE buck sources the supplies to the Bluetooth Low Energy radio subsystem, and must be enabled prior to enabling the Bluetooth Low Energy features.

29. Package Mechanical Information[1]

29.1 Apollo4 Blue SoC BGA Package



	SYMBOL	COMMON DIMENSIONS		
		MIN.	NOR.	MAX.
TOTAL THICKNESS	A	0.715	0.79	0.9
STAND OFF	A1	0.08	---	0.18
SUBSTRATE THICKNESS	A2		0.21	REF
MOLD THICKNESS	A3		0.45	REF
BODY SIZE	D		4.7	BSC
	E		4.7	BSC
BALL DIAMETER			0.2	
BALL OPENING			0.2	
BALL WIDTH	b	0.17	---	0.27
BALL PITCH	e		0.35	BSC
BALL COUNT	n		131	
EDGE BALL CENTER TO CENTER	D1		3.85	BSC
	E1		3.85	BSC
BODY CENTER TO CONTACT BALL	SD		0.175	BSC
	SE		0.175	BSC
PACKAGE EDGE TOLERANCE	aaa		0.1	
MOLD FLATNESS	bbb		0.1	
COPLANARITY	ddd		0.08	
BALL OFFSET (PACKAGE)	eee		0.15	
BALL OFFSET (BALL)	fff		0.08	

NOTES:

1. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO DATUM PLANE C.

2. DATUM C (SEATING PLANE) IS DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.

3. PARALLELISM MEASUREMENT SHALL EXCLUDE ANY EFFECT OF MARK ON TOP SURFACE OF PACKAGE.

Figure 59. BGA Package Drawing for Apollo4 Blue SoC

1. All dimensions in mm unless otherwise noted.

29.2 Reflow Profile

Table 19 lists the reflow conditions for the lead-free package. Reference IR Reflow Profile for Moisture Sensitivity Test (J-STD-020).

Reflow times: 3 cycles

Table 19: Reflow Condition (260 °C) for Pb-free Package

Profile Features	Pb-Free Assembly
Average ramp-up rate (include 217 °C to Peak)	3 °C/second max.
Temperature maintained above 217 °C	60 to 150 seconds
Time within 5 °C of actual peak temperature	20 - 40 seconds
Peak temperature (minimum)	260 +0/-5 °C
Ramp-down rate	6 °C /second max.
Time 25 °C to peak temperature	8 minutes max.

Figure 60 illustrates the temperature profile for reflow soldering requirements.

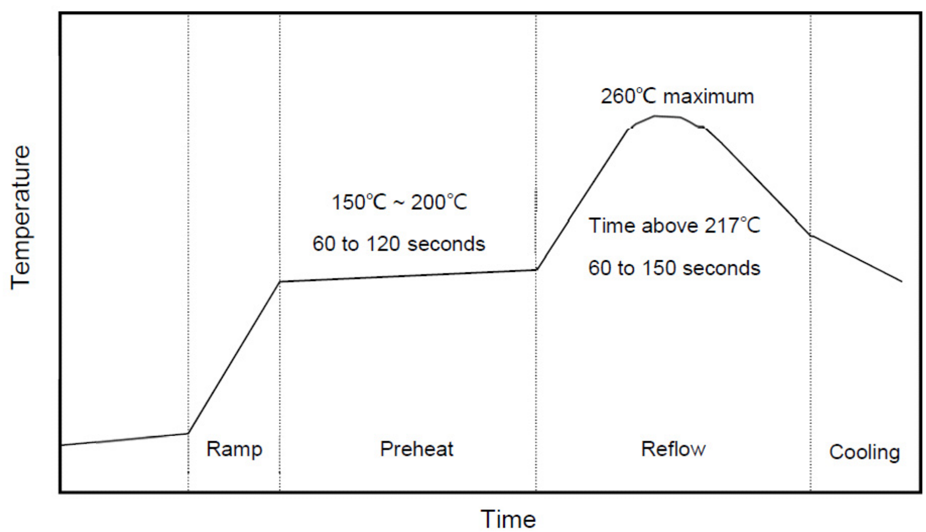


Figure 60. Reflow Profile

30. Electrical Characteristics

IMPORTANT NOTICE

Specifications and other information in this Apollo4 Blue SoC Datasheet are subject to change. Contact Ambiq sales with questions about specifications.

30.1 Absolute Maximum Ratings

The absolute maximum ratings are the limits to which the device can be subjected without permanently damaging the device and are stress ratings only. Device reliability may be adversely affected by exposure to absolute-maximum ratings for extended periods. Functional operation of the device at the absolute maximum ratings or any other conditions beyond the recommended operating conditions is not implied.

Table 20: Absolute Maximum Ratings

Symbol	Parameter	Test Conditions	Min	Max	Unit
VDDP	SIMO/LDO Buck Supply voltage		-	3.63	V
VDDA	Analog Supply voltage		-	3.63	V
VDDH	Primary IO Supply voltage		-	3.63	V
VDDH2	Secondary IO Supply voltage		-	3.63	V
VDD18	MIPI, DSI, DISPPLL Supply voltage		-	1.98	V
VDDB	BLE Buck / Controller Supply voltage		-	3.63	V
VDDAUDD	Digital Audio Supply voltage		-	3.63	V
VDDAUDA	Analog Audio Supply voltage		-	1.98	V
VDDUSB33	USB Analog 3.3 V Supply voltage		-	3.63	V
VDDUSB0P9	USB Analog 0.9V Supply voltage		-	0.99	V
V_{IO}	Voltage on all input and output pins		-0.30	VDDH+ 0.30	V
I_{SRC_STD}	Standard output pin source continuous current		-	16	mA
I_{SINK_STD}	Standard output pin sink continuous current		-	16	mA
$P_{IN\_MAX\_RF}$	Maximum RF input power and RFIO	Measured at VDDB = 1.9V		+6	dBm
T_{STORE}	Storage temperature		-55	125	ºC
T_{OP}	Operating temperature		-20	60	ºC
T_{REFLOW}	Reflow temperature	Reflow Profile per JEDEC J-STD-020D.1	-	260	ºC
I_{LU}	Latch-up current	EIA/JESD78, 25°C, ±100mA trigger current and Overvoltage at 1.5Vmax	-	100	mA
V_{ESDHBM}	ESD Human Body Model (HBM)	JS-001-2017	-	2000	V
V_{ESDCDM}	ESD Charged Device Model (CDM)	JS-002-2014	-	250	V

30.2 Recommended Operating Conditions

30.2.1 Voltage Supplies

Table 21: Voltage Supplies

Supply	Description	Source	Operating Range (V)			Comments
			Min	Typ	Max	
VDDA	Analog Supply	Battery / External Regulator	1.71	1.8 - 2.0	2.2	
VDDP	SIMO/LDO Buck Sup-ply	Battery / External Regulator	1.71	1.8 - 2.0	2.2	DCM buck will cause spikes of up to 100 mA.
VDDH	Primary I/O Supply	Battery / External Regulator	1.71	1.8 - 2.0	2.2	
VDDH2	Secondary I/O Supply	Battery / External Regulator	1.14	1.2 - 2.0	VDDA+ 0.5V	
VDDAUDA	Analog Audio and 24.576 MHz XTAL supply	External LDO, Low quiescent current, low noise preferred	1.62	1.75 - 1.85	1.98	See note below.[a] Refer to VDDAUDA Table for Noise/PSRR Requirements.
VDDAUDD	Digital Audio Supply	Battery / External Regulator	1.71	1.8 - 2.0	2.2	
$VDDB_{4dBm\_TX}$	BLE Buck / Radio Supply for +4dBm TX output	Battery / External Regulator	1.77	1.82 - 2.0	2.2	Must be same voltage as VDDH.
$VDDB_{6dBm\_TX}$	BLE Buck / Radio Supply for +6dBm TX output	Battery / External Regulator	1.80	1.85- 2.0	2.2	Must be same voltage as VDDH.
VDD18	MIPI DPHY LP LDO and transceivers	Battery / External Regulator	1.62	1.8	1.98	Powering VDD18 without powering DSI TX/D-PHY internal power rails results in uncontrolled current leakage to VDD18 and leads to long-term reliability issues. Noise/ripple: ± 2% (72mVpk-pk), freq. range 10 MHz - 3 GHz[b]
VDDUSB33	USB Analog 3.3 V Supply	Battery / External Regulator	3.0	3.3	3.63	For different USB power scenarios: See "Universal Serial Bus (USB)" on page 203.
VDDUSB0P9	USB Analog 0.9 V Supply	External Regulator	0.84	0.9	0.99	Noise/ripple < 3% (pk-pk)

a. Can be tied to GND when not using peripherals and GPIO powered by VDDAUDA.
b. This uncontrolled current leakage to VDD18 has been resolved in Apollo4 Plus and therefore turning power on and off to VDD18 in a particular sequence on Apollo4 Plus SoCs is not needed.

30.2.1.1 Using an External DC-DC Buck Supply for VDDA

In a configuration where an external DC-DC buck converter is used to supply VDDA, the maximum ripple voltage as a function of the regulator switching frequency is as shown in Table 22.

Table 22: Maximum Allowable Ripple as a Function of External Buck Switching Frequency

1.9 V External Buck Switching Frequency	Maximum Safe Peak-to-Peak Ripple
1 MHz	24 mV
800 kHz	24 mV
600 kHz	20 mV
500 kHz	20 mV
400 kHz	16 mV
300 kHz	24 mV
200 kHz	20 mV
100 kHz	12 mV

30.2.2 VDDAUDA Voltage Supply Requirements

The VDDAUDA voltage supply provides power to the AUDADC module and the Bluetooth Low Energy Module. Each module has its own set of power supply noise and ripple requirements. The sub-sections below specify these requirements when either module or both modules are intended to be used.

30.2.2.1 AUDADC and Bluetooth Low Energy Module Current Consumption

This section lists current required by the VDDAUDA supply when it is powering the AUDADC, audio microphones, and/or the Bluetooth Low Energy Module.

Table 23: AUDADC Power Supply (VDDAUDA)

Parameter	Condition	Min	Typ	Max	Units	Comments
VDDAUDA - Voltage		-	1.8	-	V	< 1 μA quiescent current > 40 dB PSRR in the range 1 kHz to 10 kHz
VDDAUDA - Leakage Current		-	10	-	nA	
VDDAUDA - Active Current	32 MHz Clock for Bluetooth Low Energy Active	-	580	680	μA	Transient peak current up to 2.2 mA (< 100 μs)
VDDAUDA - Active Current	AUDADC and PGA Active	-	60	120	μA	
VDDAUDA - Active Current	MICBIAS Active	-	20	400	μA	MICBIAS current dependent on external microphone power consumption
VDDAUDA - Active Current	All Active	-	660	1200	μA	

30.2.2.2 AUDADC Only Requirements

This section specifies the VDDAUDA requirement when it is powering only the AUDADC.

Table 24: VDDAUDA Phase Noise

Parameter	Condition	Min	Typ	Max	Units
Noise/Ripple on VDDAUDA	<20 kHz	-	0.396	1.187	mV_{pp}
	20 kHz-200 kHz	-	0.089	0.266	mV_{pp}

Table 25: PSRR Requirements for AUDADC + PGA to Achieve 80dB SNR

Parameter	Frequency	PSRR (20 mV_{pp})	PSRR (40 mV_{pp})	PSRR (60 mV_{pp})	PSRR (100 mV_{pp})	Unit
LDO PSRR Requirement by frequency dependent on ripple of voltage source to LDO	<2 kHz	-4	-10	-13	-18	dB
	10 kHz	-19	-25	-28	-33	dB
	20 kHz	-25	-31	-34	-39	dB
	200 kHz	-38	-44	-47	-52	dB

30.2.2.3 Bluetooth Low Energy Module Only Requirements

This section specifies the VDDAUDA requirement when it is powering the Bluetooth Low Energy Module but not the AUDADC.

Table 26: VDDAUDA Noise Spectral Density Specifications to Support XTALHS

Frequency	Min	Typ	Max	Units
10 Hz	-	5	10	µV/√Hz
100 Hz	-	2.5	5	µV/√Hz
1 kHz	-	2	4	µV/√Hz
10 kHz	-	0.125	0.25	µV/√Hz
100 kHz	-	0.05	0.1	µV/√Hz
1 MHz	-	0.05	0.1	µV/√Hz

Table 27: LDO PSRR Specifications to Support 32 MHz XTALHS

Parameter	Frequency	PSRR (20 mV_{pp})	PSRR (40 mV_{pp})	PSRR (60 mV_{pp})	PSRR (100 mV_{pp})	Unit
LDO PSRR Requirement by frequency dependent on ripple of voltage source to LDO	1kHz	-6	-12	-16	-22	dB
	10 kHz	-2	-6	-10	-15	dB
	100 kHz	0	-1	-4	-10	dB
	1 MHz	0	0	0	0	dB

30.2.3 Power Sequence

Table 28: Power Sequence

External Supply[a]	Conditions/Notes
VDDP/VDDH/VDDA	1. Must all ramp up together to the same voltage (i.e., VDDP = VDDH = VDDA). 2. Should generally be supplied before or at same time as other rails. 3. Some skew is acceptable.
VDDH2	1. May be powered up at the same time as VDDP/VDDH/VDDA but not before. 2. May be kept at 0V and powered up at any time after VDDP/VDDH/VDDA are powered. 3. May be kept at 0V (grounded) if not using any GPIO powered by this rail.
VDDAUDA	1. May be powered at the same time as VDDP/VDDH/VDDA but not before. 2. If powered up at same time as VDDP/VDDH/VDDA, then VDDAUDA must be lower than VDDP/VDDH/VDDA. 3. Can be tied to GND when not using peripherals and GPIO powered by VDDAUDA.Can be supplied appropriate voltage when needed at any time after VDDP/VDDH/VDDA are powered. 4. Must be a very clean supply when used. 5. Primarily needed for LP analog microphone or BLE, if present.
VDDAUDD	1. May be powered at the same time as VDDP/VDDH/VDDA but not before. 2. May be kept at 0V and powered up at any time after VDDP/VDDH/VDDA are powered. 3. May be kept at 0V (grounded) if not using the peripherals and GPIO powered by this rail.
VDDB	1. May be powered at the same time as VDDP/VDDH/VDDA but not before. 2. No other strict power-sequencing between the Bluetooth Low Energy Controller die and Apollo4 die. 3. Can safely be kept at 0V indefinitely, provided that it is powered before trying to run any code that boots or communicates with the radio. 4. Can be connected to ground if not using the Controller. 5. The Bluetooth Low Energy radio IC is a separate die co-packaged with Apollo4 in the Apollo4 Blue BGA.
VDD18	1. Should be tied to ground if not using MIPI DSI interface. 2. If using MIPI, then may be powered at the same time as VDDP/VDDH/VDDA but not before. 3. Preferably only enabled/powered up when the display is being used.
VDDUSB33/ VDDUSB0P9	1. For different USB power scenarios: See “System Power Sequencing for USB and DSI PHYs” on page 130. 2. Only required when using USB (i.e., when USB cable is plugged in). 3. Powering both supplies at the same time from VBUS source is recommended, with 0.9 V generated by LDO from the 3.3 V.

a.Recommended Termination of Unused Interface:

- USB data pads (USB0PP and USB0PN) left open
- USB PHY power rails VDDUSB33 and VDDUSB0P9 connected to ground
- DSI TX data and clock pads left open

30.2.4 Recommended External Components for the Buck Converters

Table 29: SIMO Buck Converter

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
L_{SBUCK}	SIMO Buck converter inductance (V_{SIMO})		-	2.2	-	μH
C_{BUCK}	SIMO Buck converter output capacitance (2) (V_{DDC}, V_{DDF})		-	2.2	-	μF

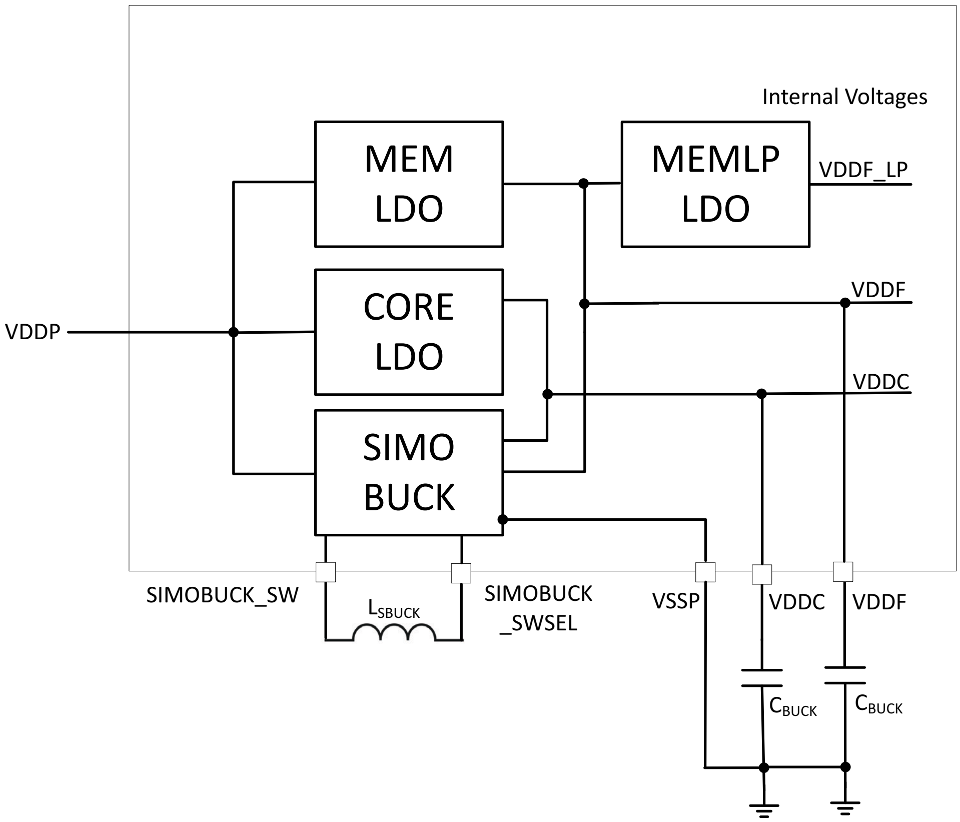


Figure 61. External Components for SIMO Buck

Table 30: BLE Buck Converter

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
$L_{BLEBUCK}$	BLE Buck converter (VDDBH) inductance		-	1.0	-	μH
$C_{BLEBUCK}$	Buck converter output (VDDBH) capacitance		-	4.7	-	μF
C_{VDDBH_RF}	Buck converter RF voltage (VDDBH_RF) capacitance		-	1.0	-	μF

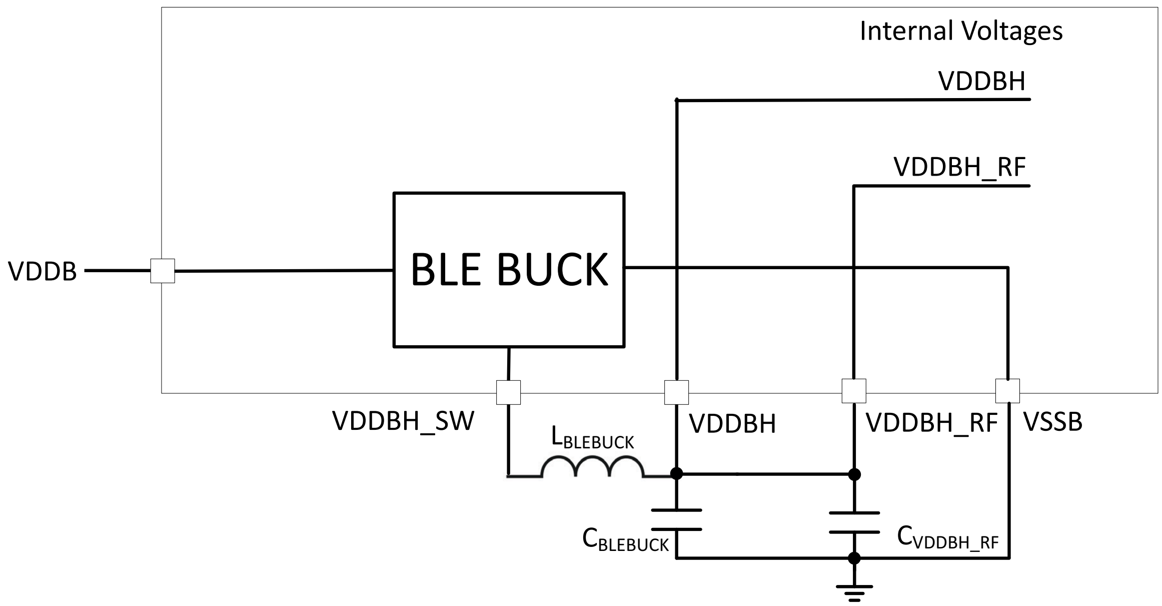


Figure 62. External Components for BLE Buck

30.2.5 Recommended External Components for Voltage Supplies

Table 31: Recommended Bypass Capacitors for Internal Supplies

Internal Supply	Bypass Capacitor
VDDC, VDDC_LV, VDDF, VDDS	2.2 µF cap to ground
VDDBH	4.7 µF Cap to ground
VDDBH_RF	1 µF Cap to ground
LPADC_VREF	100 nF cap to ground

NOTES:
1) 0201, 2.2 µF, 10 V, X5R caps are recommended for these internal rails
2) Murata GRM033R61A225KE47D are used on Ambiq validation boards for the 2.2 µF caps.
3) https://www.digikey.com/product-detail/en/murata-electronics/GRM033R61A225KE47D/490-13227-1-ND/5877435 [digikey.com]

Table 32: Recommended Bypass Capacitors for External Supplies

External Supply	Bypass Capacitor
VDDP, VDDH, VDDH2, VDDA	1 µF to Ground
VDDB	2.2 µF to Ground
VDDAUDD	2.2 µF to Ground
VDDAUDA	2.2 µF to Ground (Typ); Follow recommendations of LDO supplier. See note 4 below.
VDDUSB33	2.2 µF to ground
VDDUSB0P9	2.2 µF to Ground
VDD18	2.2 µF to Ground

NOTES:
1) Recommend use of 5 V or greater caps for 1.9 V rails
2) Recommend use of 10 V caps for 3.3 V rails
3) Do not float any supply inputs. If not powered, they should be grounded
4) Suitable standalone small form factor LDOs:
- Microchip MCP1811A in 1.0 x 1.0 x 0.50 mm UDFN package
- TI TPS7A02 in 1.0 x 1.0 x 0.40 mm X2SON package

Other Supplies:

- SIMO Buck Inductor (connected between SIMOBUCK_SW and SIMOBUCK_SWSEL):
  - 2.2 µH
  - Saturation current > 400 mA (> 500 mA recommended to achieve specified power consumption)
  - Maximum DC resistance < 0.55 ohms
  - Operating frequency range > 20 MHz
  - Recommended part: Murata DFE201610E-2R2M=P2 (0806) or Taiyo Yuden MBKK1608T2R2M (0603)
- BLE Buck Inductor[1] (Connected between VDDBH_SW and VDDBH):

1. BLE Buck and its external components are for the Apollo4 Blue SoC.

  - 1 µH
  - Saturation current > 800 mA, e.g., < 20% loss at 1 A current
  - Maximum DC resistance < 0.55 ohms
  - Operating frequency range > 20 MHz
  - Recommended part: Murata DFE18SAN1R0ME0 (0603)
- MICBIAS
  - MICBIAS can source up to 400 µA at 1.3 V (VDDMIC), with a minimum VDDAUDA supply of 1.62 V.
  - 2.2 µF cap is needed on VDDMIC to support 400 µA. Smaller acceptable for lower current.

30.3 Current Consumption

Table 33: Current Consumption in Active Mode and Sleep Modes

Symbol	Parameter	Test Conditions	VDD (V)	Min	Typ	Max	Unit	Notes
$I_{RUNLPFB}$	Coremark run current	Executed from internal NVM, cache enabled, buck enabled, 128 kB TCM, HFRC=96 MHz	1.9	-	15.8	-	µA/ MHz	1, 2, 3, 4
			3.3	-	9.1	-		
$I_{RUNHPFB}$	Coremark run current	Executed from internal NVM, cache enabled, buck enabled, 128 kB TCM, HFRC=192 MHz	1.9	-	21.3	-	µA/ MHz	1, 2, 3, 4
			3.3	-	12.3	-		
$I_{RUNWLPFB}$	While loop run current	Executed from internal NVM, cache enabled, buck enabled, 8 kB TCM	1.9	-	8.7	-	µA/ MHz	1, 2, 3, 4
			3.3	-	5.0	-		
I_{SS2}	System Sleep mode 2 current	WFI instruction with SLEEP=1, clocks gated, oscillators on, buck converters enabled, 8 kB TCM retained	1.9	-	180	-	µA	1, 2, 3, 4
			3.3	-		-		
$I_{SDS2-8RET}$	System Deep Sleep mode 2 current	WFI instruction with SLEEPDEEP=1, XTAL on, buck enabled, BLE off, 8 kB TCM retained	1.9	-	13.6	-	µA	2, 3, 4
			3.3	-	7.8	-		
$I_{SDS2-384RET}$	System Deep Sleep mode 2 current	WFI instruction with SLEEPDEEP=1, XTAL on, buck enabled, BLE off, 384 kB TCM retained	1.9	-	20.5	-	µA	2, 3, 4
			3.3	-	11.8	-		
$I_{SDS2-1864RET}$	System Deep Sleep mode 2 current	WFI instruction with SLEEPDEEP=1, XTAL on, buck enabled, BLE off, 1864 kB SRAM retained	1.9	-	47.1	-	µA	2, 3, 4
			3.3	-	27.1	-		
I_{SDS3}	System Deep Sleep mode 3 current	WFI instruction with SLEEPDEEP=1, XTAL off, buck enabled, BLE off, all SRAM off	1.9	-	13.4	-	µA	2, 3, 4
			3.3	-	7.7	-		

[1] Core clock (HCLK) is 96 MHz for each parameter unless otherwise noted.

[2] All values measured at 25°C.

[3] Current consumption is normalized to 3.3V and shown for comparison purposes. Efficiency of conversion not considered. Specifications at other VDD voltages available upon request.

[4] All I/O power domains and peripherals powered off.

Table 34: Bluetooth Low Energy Radio Operating Current

Symbol	Parameter	Test Conditions	VDD (V)	Min	Typ	Max	Unit	Notes
I_{ACT_RX}	Radio Rx Current	f_{RF} = 2440 MHz	3.3	-	5.9	-	mA	1, 2, 3, 4
I_{ACT_TX}	Radio Tx Current @ 0 dBm	f_{RF} = 2440 MHz	3.3	-	5.2	-	mA	1, 2, 3, 4
I_{SLP}	Sleep mode current	f_{RF} = 2440 MHz	3.3	-	2.5	-	µA	1, 2, 3, 4
I_{SD}	Shutdown mode current	f_{RF} = 2440 MHz	3.3	-	0.1	-	µA	1, 2, 3, 4

[1] All values measured at 25°C.

[2] Specifications at other VDD voltages available upon request

[3] All I/O power domains and peripherals powered off.

[4] Current consumption is normalized to 3.3V here for comparison purposes. Efficiency of conversion not considered. Specifications at other VDD voltages available upon request.

30.4 Non-volatile Memory (NVM)

Table 35: NVM

Symbol	Parameter	Min	Typ	Max	Unit
P_{CYC}	Program cycles before failure	100,000	-	-	cycles
$T_{DATARET}$	Data retention @125C	10	-	-	years
T_{BW}	Burst write time	-	-	1.5	kB/ms

30.5 Power-On RESET (POR) and Brown-Out Detector (BOD)

Table 36: Power-On Reset (POR) and Brown-Out Detector (BOD)

Symbol	Parameter	Min	Typ	Max	Unit
V_{POR_RISING}	POR rising threshold voltage	-	1.62-1.72	-	V
$V_{BODL\_FALLING}$	Brownout detection low falling threshold voltage	-	1.62-1.72	-	V

30.6 General Purpose Input/Output (GPIO)

Table 37: General Purpose Input/Output (GPIO)

Symbol	Parameter	Min	Typ	Max	Unit
ALL GPIOs[a]					
C_{GPI}	Input capacitance	-	3	6	pF

a. All GPIOs have Schmitt trigger inputs

30.7 Clocks/Oscillators

Table 38: Primary Internal Clocks

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
F_{HFRC_LP}	HFRC frequency - Low Power		-	96	-	MHz
F_{HFRC}	HFRC frequency - High Performance Burst Mode		-	192	-	MHz
DC_{HFRC}	HFRC duty cycle		45	50	55	%
F_{HFRC2_LP}	HFRC2 frequency - Low Power		-	196.608	-	MHz
F_{HFRC2}	HFRC2 frequency - High Performance Burst Mode		-	393.216	-	MHz
F_{LFRC}	LFRC frequency		-	1024	-	Hz
DC_{LFRC}	LFRC duty cycle	CLKGEN_CLK-OUT_CKSEL = LFRC_DIV2	45	50	55	%

Table 39: Low-frequency Crystal

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
F_{XT}	XT frequency		-	32.768	-	kHz
DC_{XT}	XT duty cycle		45	52	60	%
C_{INX}	Internal XI/XO pin capacitance		-	3.4	-	pF
$C_{EXT\_XT\_TOL}$	Allowed external XI/XO pin capacitance per pin		-	-	7	pF
F_{OF}	XT oscillator failure detection frequency		-	2.2	-	kHz
OA_{XT}	XT oscillation allowance	At 25°C using a 32.768 kHz tuning fork crystal	320	-	-	KΩ

Table 40: High-speed Crystal Oscillator

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
F_{XTAL}	Crystal frequency		-	32	-	MHz
ΔFXTAL	Frequency tolerance	Untrimmed; include initial tolerance/aging/ temperature drift	-40	-	40	ppm
C_L	Crystal load capacitance		-	6	-	pF
ESR	Equivalent serial resistance		-	-	100	Ω
T_{XTAL}	Startup time		-	1	-	ms
C_{INX}	Internal XI32M/XO32M pin capacitance		-	3.4	-	pF
OA_{XT}	XT oscillation allowance		320	-	-	KΩ
N_{PHASE}	Phase noise at 10 kHz		-	-135	-	dBc/Hz
	Phase noise at 100 kHz		-	-141.7	-	dBc/Hz
	Phase noise at 1 MHz		-	-146.6	-	dBc/Hz

Table 41: High-speed External Oscillator

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
F_{EXTCLK}	External clock frequency		5	32	48	MHz
$DC_{EXTCLKSQ}$	Square wave external clock duty cycle		35	50	65	%
$DC_{EXTCLKSN}$	Sine wave external clock duty cycle		-	50	-	%
ΔDC_{CLKOUT_32M}	Duty cycle tolerance (from 50%) on CLK-OUT_32M external pin when external clock applied		-	-	±5	%
V_{EXTCLK_SQ}	Square wave external clock voltage amplitude (DC offset not a concern)		0.3	1	VDDAUDA	V
V_{EXTCLK_SN}	Sine wave external clock peak-to-peak voltage (DC offset not a concern)		0.3	1	VDDAUDA	V_{P-P}
$C_{XO32M\_IN\_NOXT}$	Input capacitance at XO32M pin - no crystal[a]		-	-	10	pF

a. Care must be taken to ensure external clock can drive pin's internal capacitance ($C_{XO32M\_IN\_NOXT}$) while still meeting minimum amplitude requirement (V_{EXTCLK}).

30.8 Real Time Clock (RTC)

Table 42: Real Time Clock (RTC)

Symbol	Parameter	Min	Typ	Max	Unit
F_{RTC}	Clock frequency	-	100	-	Hz
T_{CLKRES}	Clock/Alarm Resolution	-	-	1/100	s

30.9 STIMER

Table 43: System Timer (STIMER)

Symbol	Parameter	Min	Typ	Max	Unit
F_{STIMER}	Input frequency	-	-	$F_{HFRC}/16$	MHz

30.10 Watchdog Timer (WDT)

Table 44: Watchdog Timer (WDT)

Symbol	Parameter	Min	Typ	Max	Unit
T_{WD}	Watchdog timer resolution	1/16	128	128	Hz

30.11 Bluetooth Low Energy Controller

NOTE

See “VDDAUDA Voltage Supply Requirements” on page 177 for the power supply requirements for the Bluetooth Low Energy Controller module.

Table 45: Bluetooth Low Energy Operating Characteristics

Symbol	Parameter[a]	Test Conditions	Min	Typ	Max	Unit
AC Characteristics - Rx						
R_{SENS}	Receiver sensitivity	1 Mbps Bluetooth Low Energy ideal transmitter, <= 37 bytes, PER < 30.8%, DCDC off, at 27C and VDDB = 1.85 V	-	-95.5	-	dBm
		2 Mbps Bluetooth Low Energy ideal transmitter, <= 37 bytes, PER < 30.8%, DCDC off, at 27C and VDDB = 1.85 V	-	-92	-	dBm
		1 Mbps BLE ideal transmitter, extended packet size = 251 bytes, PER < 30.8%, DCDC off, at 27C and VDDB = 1.85 V	-	-94.5	-	dBm
$R_{SENS,\ VAR}$	Rx sensitivity variance between channels		-	±0.5	-	dB
$P_{RX,\ MAX}$	Maximum receiver input power	PER < 30.8%	-	-	0	dBm
$C/I_{co\text{-}channel}$	Co-channel interference	Wanted signal at -67dBm, modulated interferer in channel, PER < 30.8%	7	-		dB
F_{ET}	Frequency error tolerance		-	-	125	kHz
AC Characteristics – Tx (across process, 27C, 1.85 V)						
P_{OUT_MAX}	Maximum output power	Maximum TX output power setting @ 1.9 V	6	6.5	-	dBm
P_{OUT_MIN}	Minimum output power	Minimum TX output power	-12	-10	-8	dBm
$P_{OUT,VAR}$	Maximum Tx output power variance between channels		-	-	1	dB
P_{OUT_STEP}	Tx output power step		-	0.5	-	dB
P_{OUT_HD2}	Second harmonic output power level	Radio TX at +4dBm	-	-	-30	dBm
P_{OUT_HD3}	Third harmonic output power level	Radio TX at +4dBm	-	-	-30	dBm
P_{OUT_HD4}	Fourth harmonic output power level	Radio TX at +4dBm	-	-	-30	dBm
RF_{IMP}	RFIO Impedance		-	50	-	Ω
F_{SCK}	SPI Clock Frequency		-	-	16	MHz

a. FCC and BQB test reports are available upon request.

30.12 Voltage Comparator (VCOMP)

Table 46: Voltage Comparator (VCOMP)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
V_{COMPIN}	Input voltage range		0	-	VDDA	V

30.13 General Purpose Analog-to-Digital Converter (ADC)

Table 47: General Purpose Analog to Digital Converter (ADC)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
ANALOG INPUT						
V_{ADCIN}	Input voltage range single-ended input		0	-	V_{ADCREF}	V
V_{ADCIN_DIFF}	Input voltage range in differential mode		$-V_{ADCREF}/2$	-	$+V_{ADCREF}/2$	V
V_{ADCINN} V_{ADCINP}	Absolute differential input voltage range		0	-	VDDH	V
V_{ADCREF}	Internal reference voltage range		-	1.19	-	V
C_{ADCIN}	Input source capacitance		-	4	-	pF
SAMPLING DYNAMICS						
RES	Resolution		8	-	12	bit
F_{ADCONV}	Conversion rate[a]		-	1.6 (12b) 2.0 (10b) 2.66 (8b)	-	MS/s
TSNGLSLOT_SCNCMP_PM12	Delay from scan start to scan complete, precision mode 12		-	25	-	cycles
TSNGLSLOT_SCNCMP_PM10	Delay from scan start to scan complete, precision mode 10		-	21	-	cycles
TSNGLSLOT_SCNCMP_PM8	Delay from scan start to scan complete, precision mode 8		-	18	-	cycles
T_{CAL}	Calibration Period		-	271	-	µs
INTERNAL TEMPERATURE SENSOR						
E_{TEMP}	Temperature sensor accuracy		-	± 3	-	°C
S_{TEMP}	Temperature sensor slope		-	3.38	-	mV/°C

a. Refer to Errata List for any known device issues which may impact the achievable conversion rate.

30.14 Display

30.14.1 Display Controller (DC)

Table 48: Display Controller Serial Peripheral Interface (SPI) Interface

Symbol	Parameter	Test Condition	VCC	Min	Typ	Max	Unit
T_{SCLK_LO}	Clock low time			$1/2F_{S\text{-}CLK(max)}$	-	-	s
T_{SCLK_HI}	Clock high time			$1/2F_{S\text{-}CLK(max)}$	-	-	s

30.15 Multi-bit Serial Peripheral Interface (MSPI)

Table 49: Multi-bit Serial Peripheral Interface (MSPI)

Symbol	Parameter	Test Conditions	VDD	Min	Typ	Max	Unit	Comments
F_{CLK}	MSPI Clock frequency in data transfer mode			-	48/96	96	MHz	See MSPI section for any instance-specific clock limitations.
T_r	Clock rise time	FCLK = 96 MHz, SDR;FCLK = 48 MHz, SDR or DDR		-	2.7	-	-	
T_f	Clock fall time	FCLK = 96 MHz, SDR;FCLK = 48 MHz, SDR or DDR		-	2.8	-	-	
$T_{W(CKL)}$	Clock low time	F_{CLK} = 48 MHz		-	9.7	-	ns	Based on TT corner gatesim
$T_{W(CKH)}$	Clock high time	F_{CLK} = 48 MHz		-	10.82	-	ns	Based on TT corner gatesim
T_{ISU}	Input setup time	F_{CLK} = 96 MHz -20C<TA<60C		4.3	-	-	ns	
T_{ISU}	Input setup time	F_{CLK} = 48 MHz -20C<TA<60C		2.4	-	-	ns	
T_{IH}	Input hold time	F_{CLK} = 96 MHz -20C<TA<60C		3.1	-	-	ns	
T_{IH}	Input hold time	F_{CLK} = 48 MHz -20C<TA<60C		2.9	-	-	ns	
T_{OV}	Output valid time	F_{CLK} = 96 MHz -20C<TA<60C		-	-	2.4	ns	
T_{OV}	Output valid time	F_{CLK} = 48 MHz -20C<TA<60C		-	-	2.2	ns	
T_{OH}	Output hold time	F_{CLK} = 96 MHz -20C<TA<60C		0.6	-	-	ns	
T_{OH}	Output hold time	F_{CLK} = 48 MHz -20C<TA<60C		-0.3	-	-	ns	

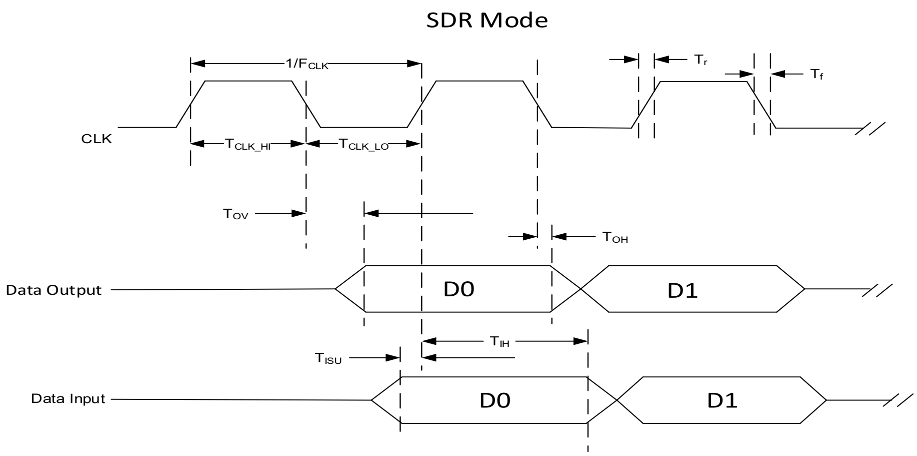


Figure 63. MSPI Timing Diagram - SDR Mode

30.16 *I²C/SPI Master* (IOM)

30.16.1 Serial Peripheral Interface (SPI) Master Interface

Table 50: Serial Peripheral Interface (SPI) Master Interface

Symbol	Parameter	Test Condition	VCC	Min	Typ	Max	Unit
F_{SCLK}	SCLK frequency range			-	-	24	MHz
B_{FIFO}	FIFO size			64 (32 for input, 32 for output)			Bytes
T_{SCLK_LO}	Clock low time			$1/2F_{S-CLK(max)}$	-	-	s
T_{SCLK_HI}	Clock high time			$1/2F_{S-CLK(max)}$	-	-	s
T_{SCLK_R}	Clock rise time	35pF load, max drive strength, GPIOx	2.0	-	2.7	-	ns
T_{SCLK_F}	Clock fall time	35pF load, max drive strength, GPIOx	2.0	-	2.8	-	ns

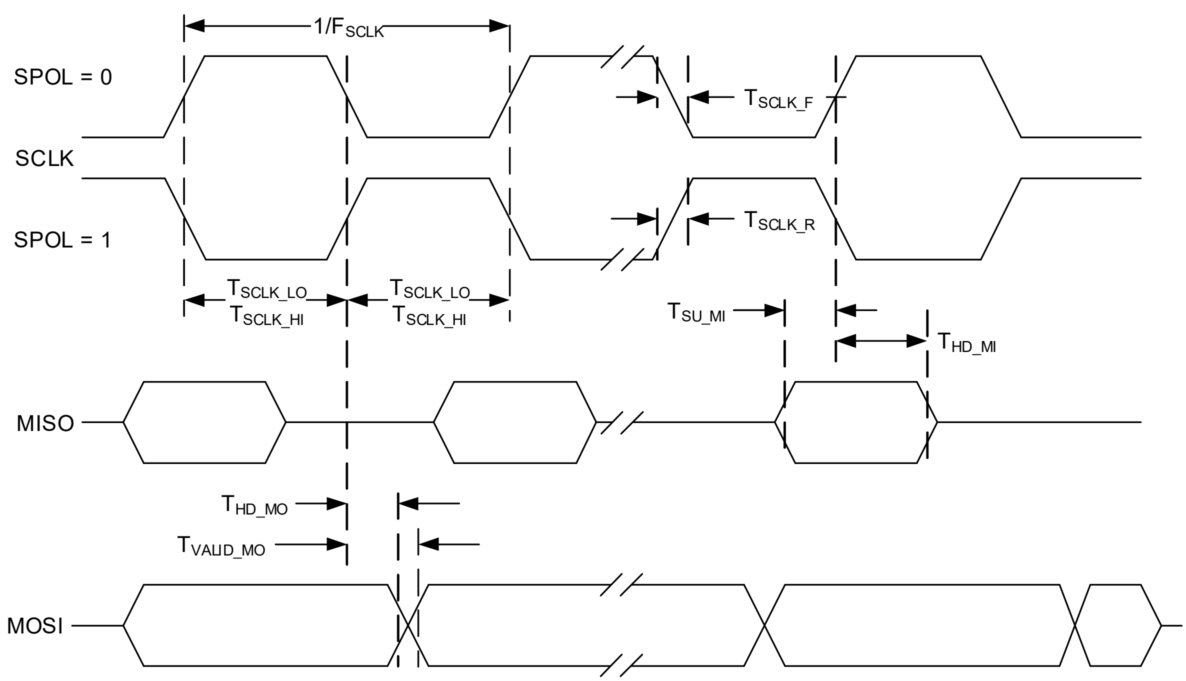


Figure 64. SPI Master Mode, Phase = 0

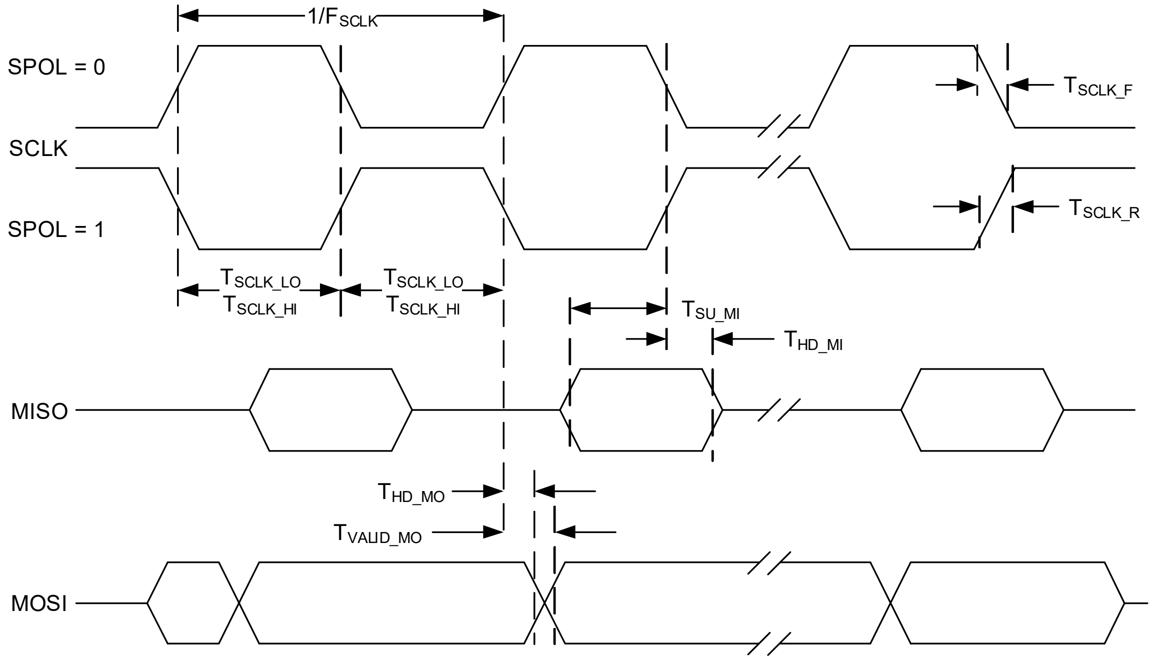


Figure 65. SPI Master Mode, Phase = 1

30.17 Universal Asynchronous Receiver/Transmitter (UART)

Table 51: Universal Asynchronous Receiver/Transmitter (UART)

Symbol	Parameter	Min	Typ	Max	Unit
F_{BAUD}	UART baud rate		-	2.6	Mbps

30.18 Universal Serial Bus (USB)

30.18.1 USB Power Gating and Leakage Current

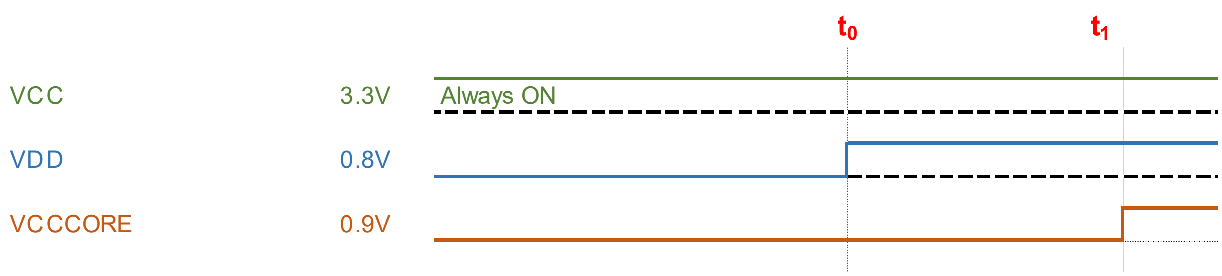
Table 52: USB Power Gating

VDDF_USB_SW	VDDUSB0P9	VDD33	Requirements
off	on	on	VDDF_USB_SW must be pulled to ground. A floating supply will cause leakage from VDDUSB0P9.
on	off	on	VDDUSB0P9 must be pulled to ground. A floating supply will cause leakage from VDDF_USB_SW.
on	off	off	VDDUSB0P9/VDD33 must be pulled to ground. A floating supply will cause leakage from VDDF_USB_SW.
off	off	on	VDDF_USB_SW /VDDUSB0P9 must be pulled to ground. A floating supply will cause leakage from VDD33.

Table 53: Leakage Current at Different External Supply Voltages

Operating conditions	VDD (0.8 V nom)	VCCCORE (0.9 V nom)	VCC (3.3 V nom)	Worst case leakage @VCC	Worst case leakage @ VCCCORE	Worst case leakage @VDD
USB came out of reset; PHY is in suspend mode	0.8 V	0.9 V	3.3 V	40 µA	4 µA	0
USB came out of reset; PHY is in suspend mode	0.8 V	0.9 V	3.4 V	42 µA	4 µA	0
PHY is not operational; All input signals are at their default values.	0	0	3.3 V	1.2 µA	0	0
PHY is not operational; All input signals are at their default values.	0	0	3.2 V	1.1 µA	0	0
PHY is not operational; All input signals are at their default values.	0	0	1.9 V	0.7 µA	0	0
PHY is not operational; All input signals are at their default values.	0	0.9 V	3.2 V	X	X	0
PHY is not operational; All input signals are at their default values.	0	0.9 V	1.9 V	X	X	0

NOTES:
1) 3.4 V nominal on 3.3 V rail is OK as long as the maximum voltage kept to <= 3.6 V.
2) All numbers given are with the assumption PHY is brought to suspend state, and the USB controller is held in reset before any rails are powered off.
3) Both DP/DM are in the high-Z state if the PHY is powered off when in suspend or reset state. Otherwise, the DP/DM state depends on the state USB controller/SW is in at the moment when PHY power goes off.
4) Leakage via external ESD protection brings DP/DM voltage to zero when PHY puts them in the high-Z state.
5) If 3.3 V is always powered, the software shall enable internal 0.8 V rail before the external 0.9V. Otherwise, leakage can't be controlled (last two rows of the table).
6) "X" indicates that the leakage current is out of control.
7) Assuming 3.3 V is always on, the power-on sequence depicted below is allowed with no limitations on the duration [t0...t1]. The leakage for the period from t0 to t1 is several microamps.



30.18.2 USB PHY Power and Interface Timing Requirements

Table 54: USB PHY Power Supply

Parameter	Condition	Min	Typ	Max	Units
VDDUSB33 - Voltage		3.0	-	3.6	V
VDDUSB33 - Max Load		-	-	6	mW
VDDUSB0P9 - Voltage		0.837	-	0.990	V
VDDUSB0P9 - Ripple		-	-	5	%
VDDUSB0P9 - Max Continuous Current		-	-	3.23	mA
VDDUSB0P9 - Peak Current		-	-	40	mA
VDDUSB33 - Power Consumption	Power consumption in suspend state when both 3.3 V and 0.9 V are supplied.	-	-	4	µA
VDDUSB0P9 - Power Consumption		-	-	5	µA

30.19 Secure Digital Input Output (SDIO)

Table 55: Secure Digital Input Output (SDIO)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
T_r	Clock rise time (SDR or DDR)		-	2.7	-	ns
T_f	Clock fall time (SDR or DDR)		-	2.8	-	ns

30.20 Audio Analog-to-Digital Converter (AUDADC)

NOTE

See “VDDAUDA Voltage Supply Requirements” on page 177 for the power supply requirements for the AUDADC module.

30.20.1 AUDADC Audio Specifications

In Table 56, the following conditions apply unless otherwise indicated.

- CLK source = XTALHS 24.576 MHz
- Input Frequency = 997 Hz
- Sample Rate = 48 kHz
- PGA GAIN = 0dB
- Pseudo-differential input
- DRE disabled
- 0.1uF X5R/X7R AC coupling capacitors used
- 100 nF low leakage capacitor on LPADC_VREF (part number 0201ZD104KAT2A)
- VDDAUDA driven by LDO except for PSRR measurements

Table 56: AUDADC Audio Specifications

Symbol	Parameter	Test Condition	Weighting	Min	Typ	Max	Units
A_{AUDADC_PGA}	PGA Gain	Minimum	-	-	-6	-	dB
		Maximum	-	-	24	-	dB
		Step Size	-	-	0.5	-	dB
IC_ISOL_{AUDADC}	Interchannel Isolation	PGA Gain = 0 dB	1 kHz	-	60	-	dB
		PGA Gain = 24 dB	1 kHz	-	60	-	dB
$FDIV\_FS_{AUDADC}$	Full Scale Input Voltage (Fully Differential Input)	PGA Gain = -6 dB	-	-	1000	-	mVrms
		PGA Gain = 0 dB	-	-	500	-	mVrms
		PGA Gain = 12 dB	-	-	125	-	mVrms
		PGA Gain = 24 dB	-	-	31.25	-	mVrms
		PGA Gain = 30 dB	-	-	15.63	-	mVrms
MOD_IDX_{AUDADC}	Modulation Index[a]		-	-	78.3	-	%FS
$V_{AUDADCIN}$	DC Voltage Range at Analog Input Pin	Pin floating	-	-	0.8	-	V
$V_{AUDADCREF}$	DC Voltage Range at LPADC_REF	See Note[b]	-	-	1.2	-	V
$ATOT_{AUDADC}$	Audio Turn-on Time	With BGTLP always on and PGA VREFGEN quick charge	-	-	10	-	ms

a. Modulation index specifies the percentage of the digital full scale obtained when a full scale analog input is driven on the PGA inputs for a given gain. Exceeding the full-scale analog voltage may result in a distorted/clipped signal.
b. LPADC_REF pin may not be loaded due to low current drive (even with a typical voltmeter). For voltage measurements, use high impedance buffer before connecting to voltmeter.

30.20.2 AUDADC Mic Bias Specifications

Table 57: AUDADC Mic Bias Specifications

Symbol	Parameter	Test Condition		Min	Typ	Max	Units
V_{AUDADC_OVNL}	Output Voltage (no load)	VOLTAGETRIM = 5		-	0.9	-	V
V_{AUDADC_OVNL}	Output Voltage (no load)	VOLTAGETRIM = 52		-	1.5	-	V
V_{AUDADC_OVBM}	Output Voltage (Bypass Mode)	VOLTAGETRIM = 63		-	VDDAUDA	-	V

30.20.3 ASRC Performance

30.20.3.1 Analysis by Test Cases

The figures in this subsection present the spectra using a sinusoidal test wave of amplitude -1 dB and different input and output frequencies. The test cases and the corresponding parameters are listed in Table 58.

Table 58: ASRC Test Cases

Test Case	Figure	Sample Size (Bits)	Signal Frequency (kHz)	FS_{in} (kHz)	FS_{out} (kHz)	Source Ratio (FS_{out} / FS_{in})
1	Figure 66	8	1	192	98.5	0.513
2	Figure 67	16	2	48	48.1	1.002
3	Figure 68	24	5	47.9	96	2.004
4	Figure 69	32	10	96	95.8	0.998

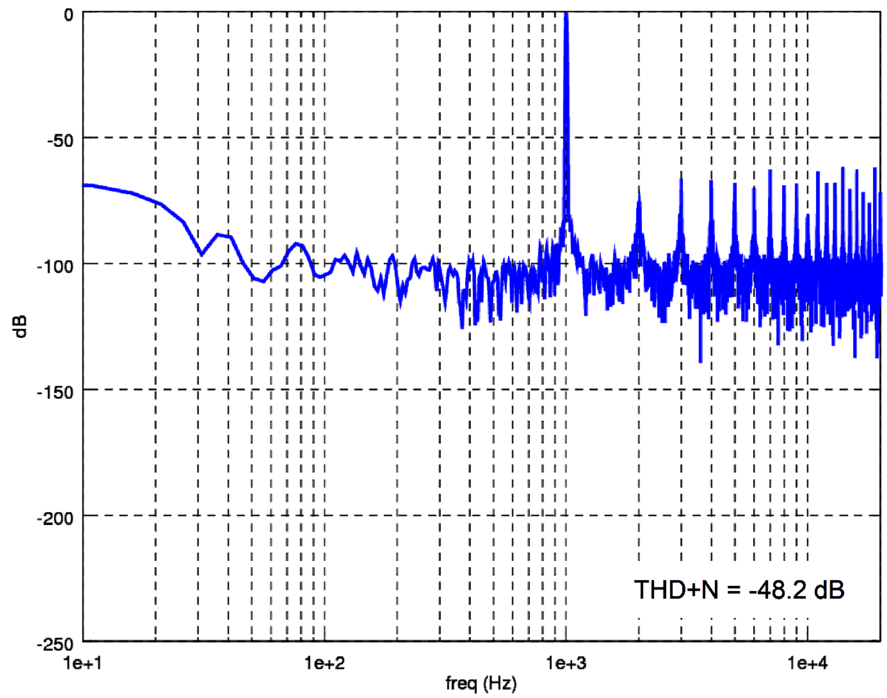


Figure 66. ASRC Performance Analysis - Test Case 1

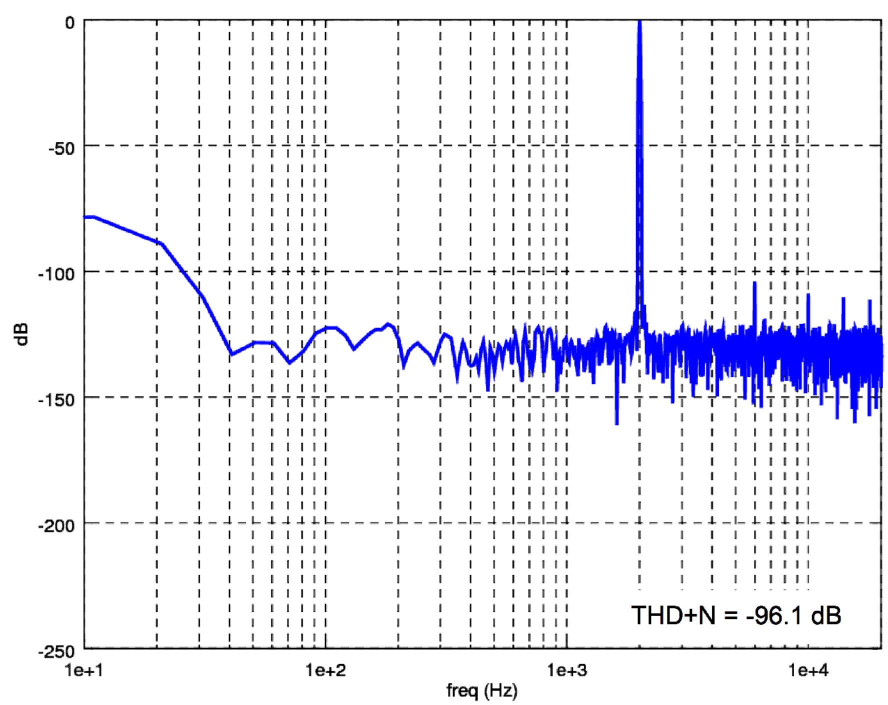


Figure 67. ASRC Performance Analysis - Test Case 2

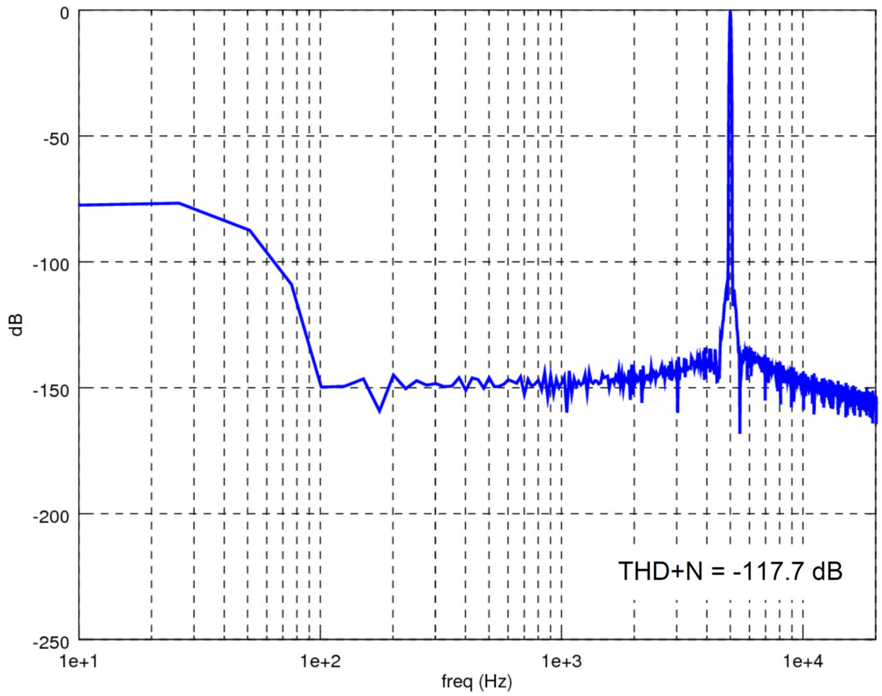


Figure 68. ASRC Performance Analysis - Test Case 3

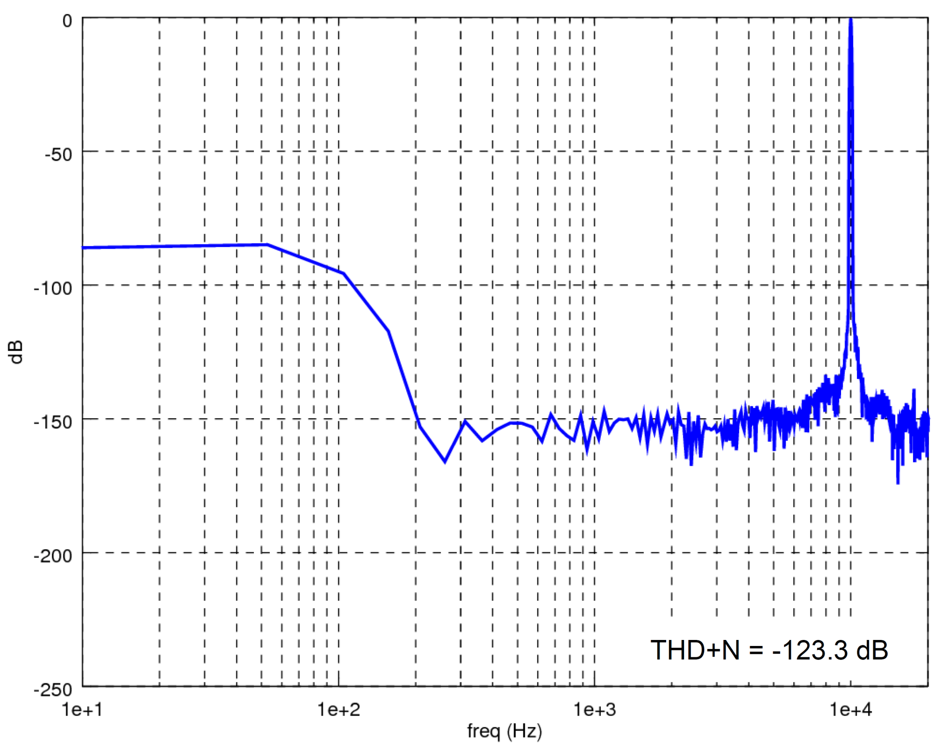


Figure 69. ASRC Performance Analysis - Test Case 4

30.20.3.2 Linearity Analysis

Figure 70 shows the THD+N measurement for different input signal frequencies and for different sample sizes.

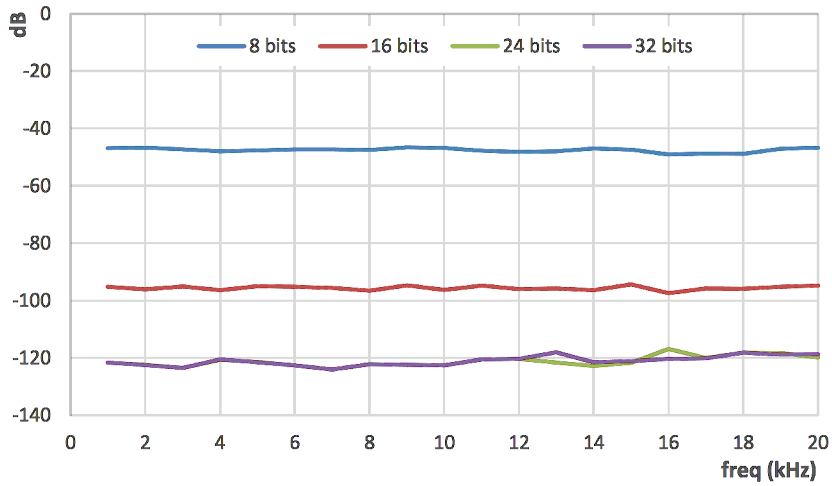


Figure 70. THD+N vs Input Frequency Using FS_{in} = 48 kHz and FS_{out} = 48.1 kHz

30.20.3.3 Analysis for Sine Wave with Frequency of 1 kHz

Table 59 shows the values of the Total Harmonic Distortion + Noise (THD+N) for some conversion ratios, using a 1 kHz input signal.

Table 59: THD+N for Some Conversion Ratios Using a 1 kHz Input Signal

Rate Conversion (kHz)	Ratio	Total Harmonic Distortion + Noise (dB)			
		8 bits	16 bits	24 bits	32 bits
48 → 48.1	1.0021	-47.5	-97.2	-127.0	-127.1
44.1 → 80	1.8141	-48.5	-95.8	-131.8	-131.8
96 → 96.1	1.0010	-47.2	-96.0	-131.5	-131.5
48 → 72	1.5000	-47.6	-97.1	-141.7	-141.9

Table 59: THD+N for Some Conversion Ratios Using a 1 kHz Input Signal

Rate Conversion (kHz)	Ratio	Total Harmonic Distortion + Noise (dB)			
		8 bits	16 bits	24 bits	32 bits
98.5 → 192	1.9492	-47.9	-95.5	-135.9	-135.5
180 → 192	1.0667	-48.7	-95.7	-125.7	-125.6
48.1 → 48	0.9979	-48.9	-96.7	-128.6	-128.6
80 → 44.1	0.5513	-49.6	-97.4	-120.9	-121.0
96.1 → 96	0.9989	-47.8	-96.1	-131.8	-131.7
72 → 48	0.6667	-50.7	-98.9	-141.4	-139.5
192 → 98.5	0.5130	-48.1	-96.8	-126.3	-126.2
192 → 180	0.9375	-47.2	-95.9	-139.9	-139.9

31. Ordering Information

Table 60: Ordering Information

Device Name	Orderable Part Number[1]	NVM	RAM	Package	Packing	Temperature Range
Apollo4 Blue SoC	AMA4B2KK-KBR-B2	2 MB	1.8 MB	131-pin BGA	Tape and Reel	–20 to 60°C

1. The silicon revision is identified by the first letter in the bottom row of the package's top marking: E = revision B1, F = revision B1+, G = revision B2, H = revision B2+.

32. Document Revision History

Table 61: Document Revision List

Revision	Date	Description
1.0.0	Sep 2021	Initial public release.
1.1.0	May 2022	Front Page Features List: - Active mode current updated - BLE RX sensitivity update (here and throughout) MCU Core: - Active mode current updated - Updated “System Power States” section. GPIO: - Removed notation about power switches on GPIO29 and GPIO30. Package Pins, GPIO, Display Controller: - Added note specifying that the DPI-2 interface is not supported. MSPI: - Removed remaining reference to unsupported DEVICE1. - Corrected set of MSPI CE pin options. IOM: - Maximum I2C transfer size updated to 512 bytes. I^2S: - Added note about limitation of TDM use to a single peripheral. Display Controller: - Removed sections on unsupported serial interfaces. Electrical Specifications: - Added Current Consumption table. - Added VDDAUDA Voltage Supply Requirements section - Added High-speed Crystal Oscillator table. - Added SDIO table. - Updated BLE specifications.
1.2.0	Aug 2022	MSPI: - Added maximum clock rate for MSPI0 for octal is 48 MHz for SDR. - Added maximum clock rate for MSPI2 for all data widths updated to 24MHz SDR and 12MHz DDR. - Added MSPIn_4 cannot be used as the MSPI’s clock line. - Noted that DQS mode not supported. Electrical Specifications: - Bluetooth Low Energy Module Only Requirements section updated. - Bluetooth Low Energy Operating Characteristics table updated. - PGA maximum gain updated in AUDADC Audio Specifications

Table 61: Document Revision List

Revision	Date	Description
1.3.0	Feb 2023	Front Page Features List, SoC Product Introduction and ADC: - Clarified that stated ADC max sampling rate applies to 8-bit mode. Front Page Features List, SoC Product Introduction and DSI: - Corrected number of MIPI DSI data lanes. ADC: - Clock source selection updated in Functional Overview. - Note added about effect of existing errata on achievable sample rate. MSPI: - Noted that all non-nCE MSPI interface pins should be configured for 1P0X drive strength. I2S: - Several “usage” sections moved to Apollo4 Family Programmer’s Guide, v7.0. DC: - Note added describing interface restrictions when using RGBA4444 or ARGB4444 input color modes. Electrical Specifications: - Absolute Maximum Ratings: Storage temp range added. - Recommended Operating Conditions: Updated $VDDB_{4dBm\_TX}$ and $VDDB_{6dBm\_TX}$ voltage. - Added section “Using an External DC-DC Buck Supply for VDDA” - Added BLE Buck inductor information in section 30.2.5.

Legal Information and Disclaimers

AMBIQ MICRO INTENDS FOR THE CONTENT CONTAINED IN THE DOCUMENT TO BE ACCURATE AND RELIABLE. THIS CONTENT MAY, HOWEVER, CONTAIN TECHNICAL INACCURACIES, TYPOGRAPHICAL ERRORS OR OTHER MISTAKES. AMBIQ MICRO MAY MAKE CORRECTIONS OR OTHER CHANGES TO THIS CONTENT AT ANY TIME. AMBIQ MICRO AND ITS SUPPLIERS RESERVE THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS AND OTHER CHANGES TO ITS PRODUCTS, PROGRAMS AND SERVICES AT ANY TIME OR TO DISCONTINUE ANY PRODUCTS, PROGRAMS, OR SERVICES WITHOUT NOTICE.

THE CONTENT IN THIS DOCUMENT IS PROVIDED "AS IS". AMBIQ MICRO AND ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS CONTENT FOR ANY PURPOSE AND DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS CONTENT, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHT.

AMBIQ MICRO DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO COVERING OR RELATING TO THIS CONTENT OR ANY COMBINATION, MACHINE, OR PROCESS TO WHICH THIS CONTENT RELATE OR WITH WHICH THIS CONTENT MAY BE USED.

USE OF THE INFORMATION IN THIS DOCUMENT MAY REQUIRE A LICENSE FROM A THIRD PARTY UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF THAT THIRD PARTY, OR A LICENSE FROM AMBIQ MICRO UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF AMBIQ MICRO.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE SYSTEM AND SOFTWARE IMPLEMENTERS TO USE AMBIQ MICRO PRODUCTS. THERE ARE NO EXPRESS OR IMPLIED COPYRIGHT LICENSES GRANTED HEREUNDER TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS OR INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN. AMBIQ MICRO MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES AMBIQ MICRO ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT, AND SPECIFICALLY DISCLAIMS ANY AND ALL LIABILITY, INCLUDING WITHOUT LIMITATION CONSEQUENTIAL OR INCIDENTAL DAMAGES. “TYPICAL” PARAMETERS WHICH MAY BE PROVIDED IN AMBIQ MICRO DATA SHEETS AND/OR SPECIFICATIONS CAN AND DO VARY IN DIFFERENT APPLICATIONS AND ACTUAL PERFORMANCE MAY VARY OVER TIME. ALL OPERATING PARAMETERS, INCLUDING “TYPICALS” MUST BE VALIDATED FOR EACH CUSTOMER APPLICATION BY CUSTOMER’S TECHNICAL EXPERTS. AMBIQ MICRO DOES NOT CONVEY ANY LICENSE UNDER NEITHER ITS PATENT RIGHTS NOR THE RIGHTS OF OTHERS. AMBIQ MICRO PRODUCTS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE AS COMPONENTS IN SYSTEMS INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE AMBIQ MICRO PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR. SHOULD BUYER PURCHASE OR USE AMBIQ MICRO PRODUCTS FOR ANY SUCH UNINTENDED OR UNAUTHORIZED APPLICATION, BUYER SHALL INDEMNIFY AND HOLD AMBIQ MICRO AND ITS OFFICERS, EMPLOYEES, SUBSIDIARIES, AFFILIATES, AND DISTRIBUTORS HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES, AND EXPENSES, AND REASONABLE ATTORNEY FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PERSONAL INJURY OR DEATH ASSOCIATED WITH SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT AMBIQ MICRO WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE PART.

Ambiq Micro, Inc.
6500 River Place Boulevard, Building 7,
Suite 200, Austin, TX 78730-1156
www.ambiq.com/
sales@ambiq.com
https://support.ambiq.com
+1 (512) 879-2850

DS-A4B-1p3p0
Version 1.3.0
February 2023