TSX-1001 Datasheet

APPROVALS	
Revision Date	April 20, 2012
Document No.	001-00012
Revision	A
Triad Semiconductor, Inc.	William Farlow Signature William Farlow Printed
Triad Semiconductor, Inc.	John Seibel Signature John Seibel Printed

Revision History

Revision	Modifications	Originator	Modification Date
	Initial draft	Jeff Mueller	10 March 2010
	Incorporation of HSS material, UART and SPI memory maps. Initial edits from C. David entered.	Jeff Mueller/ William Farlow	12 March 2010
	EEPROM, RAM and WDT sections added. Additional writing to NVIC section. Cleaned-up/formatted/renamed some registers and bit settings. First release to Keil/ARM.	Jeff Mueller	23 March 2010
	UART baud and Op-Amp tables added. Major update to Section 6. Relocation of Interrupt section to Section 6.	Jeff Mueller/ William Farlow/ D.avid Ihme	5 April 2010
	LOCK_OK -> SWCHK_LOCK	William Farlow	28 April 2010
	Merging info from Rev C docs on ADC, DAC, PWM, TMR	William Farlow C David Huntchins	Various
	Added marketing name CLK_IN for pin 98 internal engineering name was RESERVED/(BYPASS_CLK)	William Farlow	1 Dec 2011
A	Changed footer information to be Rev A, and added date	Bonny Bodle	20 Apr 2012

Table of Contents

Contents

1 TSX-1001 Overview

1.1 Introduction

The TSX-1001 is a general-purpose microcontroller that implements the low-power, 32-bit *Cortex*™-M0 microcontroller from ARM. Developed on Triad Semiconductor's Mocha-1 configurable ASIC platform, the IC also contains a variety of analog, digital, and mixed-signal peripherals that demonstrate the flexibility achieved through via-configurable design. These peripherals consume only a fraction of the configurable platform resources, leaving the remaining resources for additional (or alternative) circuit implementations.

Using Triad Semiconductor's proprietary via-configurable technology, new configurable via layers (CVLs) can be created, merely by remapping a single via layer of the Mocha-1 platform-- the TSX-1001 can be easily adapted to suit specific application requirements. For further information about Mocha-1 and a partial list of other candidate peripherals, see http://www.triadsemi.com/services/ipcatalog/.

1.2 TSX-1001 Features

Mircoprocessor

- Single-Cycle Multiplier (32 bits x 32 bits)
- 32 kbytes EEPROM
- 24 kbytes Static RAM
- Processor-Integrated Watchdog Timer
- Processor-Integrated SerialWire™ Debugger (SWD)
- 8 Operating Modes

Mixed-Signal

- 12-bit, High-Speed, Current-Steering DAC with Clock Scaling
- 256 DAC Sample Buffer with Circular and FIFO Capabilities
- 16-bit, Sigma-Delta ADC
- PLL Clock Generator with External Clocking Capabilities

Digital

- 8-Bits of Push-Pull, Bi-Directional GPIO
- 8-Bits of Open-Drain, Bi-Directional GPIO
- High-Speed UART
- SPI Interface (master/slave)
- 30-Bit PWM with External and Processor Enables
- 30-Bit General-Purpose Timer

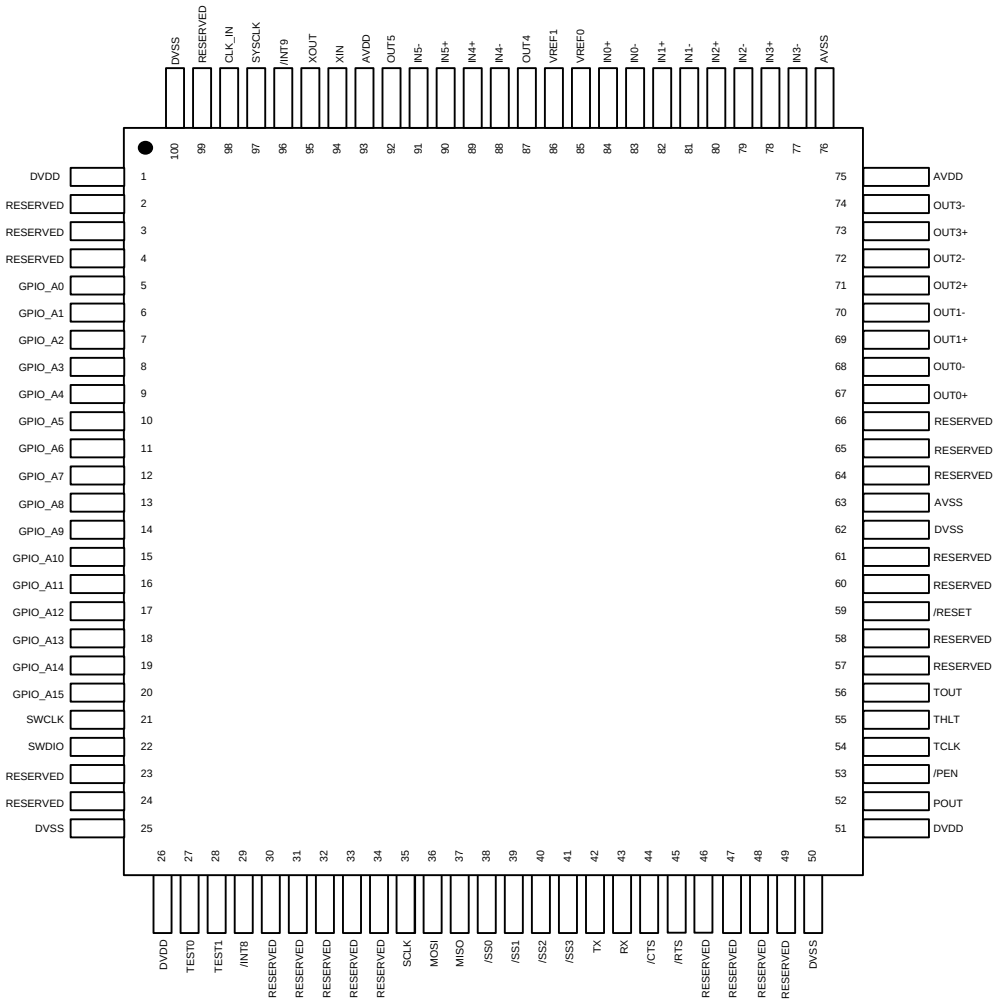
Analog

- On-Chip Bandgap and Vdd/2 Buffered Voltage References
- Two Fully Differential, 50 MHz Op Amps
- Two Low-Noise, 330 kHz Op Amps
- One Single-Ended, 50 MHz Op Amp

1.3 Applications

- Bluetooth Low-Energy Sensors
- Automatic Meter Reading
- Intelligent Lighting
- Low-Power Fitness Devices
- Arbitrary Waveform Synthesis
- Portable Medical Devices
- Motor Control
- Smart Grid Controllers
- Smart Inverters for Solar Panels

1.4 Pin Diagram



1.5 Block Diagram

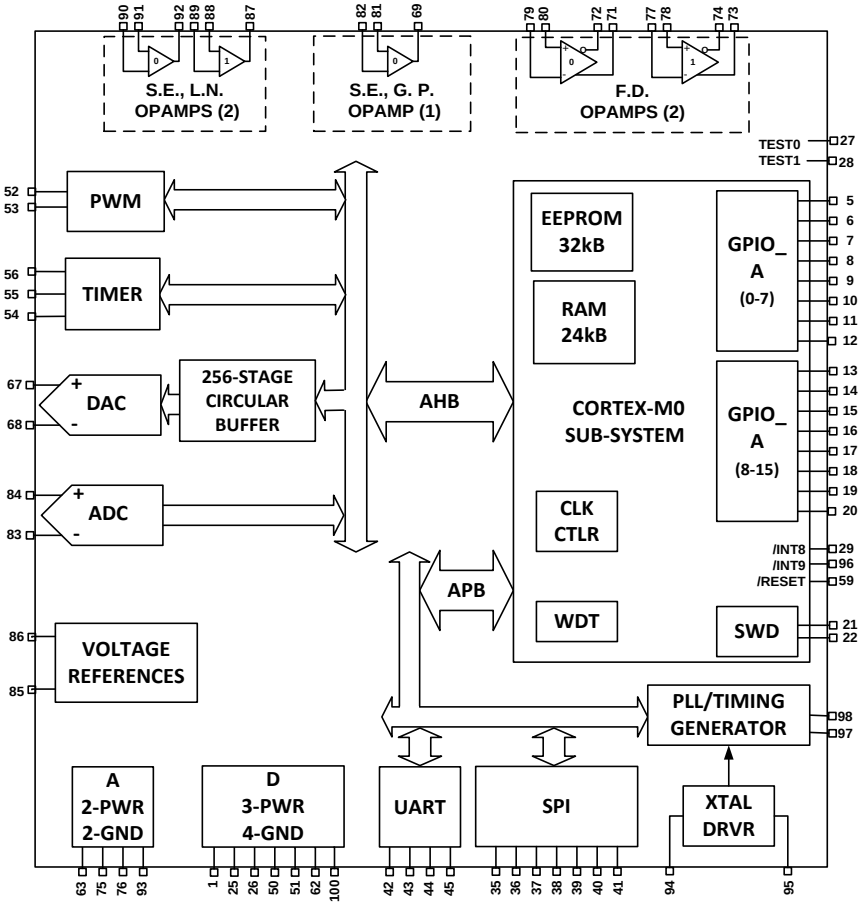


Figure 1: TSX-1001 block diagram

1.6 Pin Descriptions

PIN	NAME	TYPE	DESCRIPTION
1	DVDD	Digital Power	Positive supply for digital circuits
2	*Reserved*	Digital I/O	*Reserved*
3	*Reserved*	Digital I/O	*Reserved*
4	*Reserved*	Digital I/O	*Reserved*
5	GPIO_A0	Digital I/O	GPIO from ARM core. 8 bi-direction, push-pull pins (1)
6	GPIO_A1	Digital I/O	
7	GPIO_A2	Digital I/O	
8	GPIO_A3	Digital I/O	
9	GPIO_A4	Digital I/O	
10	GPIO_A5	Digital I/O	
11	GPIO_A6	Digital I/O	
12	GPIO_A7	Digital I/O	
13	GPIO_A8	Digital I/O	GPIO from ARM core. 8 bi-direction, open-drain pins (external 47k, nominal, pull-up resistor required).
14	GPIO_A9	Digital I/O	
15	GPIO_A10	Digital I/O	
16	GPIO_A11	Digital I/O	
17	GPIO_A12	Digital I/O	
18	GPIO_A13	Digital I/O	
19	GPIO_A14	Digital I/O	
20	GPIO_A15	Digital I/O	
21	SWCLK	Digital I	Serial Wire Debug access port clock (1)
22	SWDIO	Digital I/O	Serial Wire Debug data line (2)
23	*Reserved*	Digital I/O	*Reserved*
24	*Reserved*	Digital I/O	*Reserved*
25	DVSS	Digital Power	Negative supply for digital circuits
26	DVDD	Digital Power	Positive supply for digital circuits
27	TEST0	Digital I	TEST 0 input pin (1) (Active High PLL Bypass mode to use CLK_IN)
28	TEST1	Digital I	TEST 1 input pin (1) (Active High Manufacturing Test Mode)
29	/INT8	Digital I	External interrupt input, 8 (2)
30	*Reserved*	Digital I/O	*Reserved*
31	*Reserved*	Digital I/O	*Reserved*
32	*Reserved*	Digital I/O	*Reserved*
33	*Reserved*	Digital I/O	*Reserved*
34	*Reserved*	Digital I/O	*Reserved*
35	SCLK	Digital I/O	SPI clock output (Master mode) or input (Slave mode)
36	MOSI	Digital I/O	SPI Master Out Slave In data line
37	MISO	Digital I/O	SPI Master In Slave Out data line
38	/SS0	Digital I/O	SPI Slave Select 0 output (Master mode) or input (Slave mode)
39	/SS1	Digital O	SPI Slave Select 1 output (Master mode only)
40	/SS2	Digital O	SPI Slave Select 2 output (Master mode only)
41	/SS3	Digital O	SPI Slave Select 3 output (Master mode only)
42	TX	Digital O	UART TX data line
43	RX	Digital I	UART RX data line (2)
44	/CTS	Digital I	UART clear-to-send input (2)
45	/RTS	Digital O	UART ready-to-send output
46	*Reserved*	Digital I/O	*Reserved*
47	*Reserved*	Digital I/O	*Reserved*
48	*Reserved*	Digital I/O	*Reserved*
49	*Reserved*	Digital I/O	*Reserved*

50	DVSS	Digital Power	Negative supply for digital circuits
51	DVDD	Digital Power	Positive supply for digital circuits
52	POUT	Digital O	Pulse-Width-Modulator PWM0 output
53	/PEN	Digital I	Pulse-Width-Modulator PWM0 external enable input (1)
54	TCLK	Digital I	Timer TIMER0 clock (1)
55	THLT	Digital I	Timer TIMER0 halt (1)
56	TOUT	Digital O	Timer TIMER0 output
57	*Reserved*	Digital I/O	*Reserved*
58	*Reserved*	Digital I/O	*Reserved*
59	/RESET	Digital I/O	System reset
60	*Reserved*	Digital I/O	*Reserved*
61	*Reserved*	Digital I/O	*Reserved*
62	DVSS	Digital Power	Negative supply for digital circuits
63	AVSS	Analog Power	Negative supply for analog circuits
64	*Reserved*	Analog I/O	*Reserved*
65	*Reserved*	Analog I/O	*Reserved*
66	*Reserved*	Analog I/O	*Reserved*
67	OUT0+	Analog O	DAC DAC0 positive output
68	OUT0-	Analog O	DAC DAC0 negative output
69	OUT1+	Analog O	Single-ended amplifier output
70	OUT1-	Analog O	Internally connected to AVSS
71	OUT2+	Analog O	Fully differential amplifier, FDOA0, positive output
72	OUT2-	Analog O	Fully differential amplifier,FDOA0, negative output
73	OUT3+	Analog O	Fully differential amplifier, FDOA1, positive output
74	OUT3-	Analog O	Fully differential amplifier, FDOA1, negative output
75	AVDD	Analog Power	Positive supply for analog circuits
76	AVSS	Analog Power	Negative supply for analog circuits
77	IN3-	Analog I	Fully differential amplifier, FDOA1, negative input
78	IN3+	Analog I	Fully differential amplifier, FDOA1, positive input
79	IN2-	Analog I	Fully differential amplifier, FDOA0, negative input
80	IN2+	Analog I	Fully differential amplifier, FDOA0, positive input
81	IN1-	Analog I	Single-ended amplifier, SEOA0, negative input
82	IN1+	Analog I	Single-ended amplifier, SEOA0, positive input
83	IN0-	Analog I	ADC0 negative input
84	IN0+	Analog I	ADC0 positive input
85	VREF0	Analog O	1.2V bandgap voltage reference (buffered)
86	VREF1	Analog O	AVDD/2 voltage reference (buffered)
87	OUT4	Analog O	Low-noise amplifier LNA0 output
88	IN4-	Analog I	Low-noise amplifier LNA0 negative input
89	IN4+	Analog I	Low-noise amplifier LNA0 positive input
90	IN5+	Analog I	Low-noise amplifier LNA1 positive input
91	IN5-	Analog I	Low-noise amplifier LNA1 negative input
92	OUT5	Analog O	Low-noise amplifier LNA1 output
93	AVDD	Analog Power	Positive supply for analog circuits
94	XIN	Analog I	Crystal input pin
95	XOUT	Analog O	Crystal output pin
96	/INT9	Digital I/O	External interrupt input, 9 (2)
97	SYSCLK	Digital O	System clock output
98	CLK_IN	Digital I/O	System clock input (Selected with TEST1 = 0V and TEST0 = DVDD
99	*Reserved*	Digital I/O	*Reserved*
100	DVSS	Digital Power	Negative supply for digital circuits

Note 1: Integrated 100k Pull-Down
Note 2: Integrated 100k Pull-Up

Table 1: TSX-1001 pin list

1.7 General Device Characteristics/ Absolute Maximum Ratings

The following table provides nominal-operating-condition and absolute-maximum-ratings information for the TSX-1001. Electrical characteristics and configuration information for specific peripherals/sub-circuits are contained within their respective sections of this document.

Specification	Note	Min	Typ	Max	Unit
Operating V_{DD}		2.7	3.3	5.0	V
Maximum V_{DD}				7.0	V
Lifetime		2	5	10	years
Operating Temperature		-40	25	85	°C
Max Analog input levels		AV_{SS} - 0.7V		AV_{DD} + 0.7V	V
Max Digital I/O levels		DV_{SS} - 0.7V		DV_{DD} + 0.7V	V
Digital signaling levels (V_{IL}, V_{IH}, V_{OL}, V_{OH})					
Ground voltage differences	Use common ground plane for AV_{SS} DV_{SS}		TBD		
EEPROM write cycles			100,000		cycles
Internal (continuous) power dissipation			TBD		
Maximum junction temp			125		°C
Lead temp			TBD		

Table 2: General device characteristics

1.8 Memory Map Overview

A memory map showing the relative organization of the TSX-1001's major register blocks is shown in Table 3. Detailed descriptions of each block are discussed in subsequent sections of this datasheet.

Device Type	Address Range/ Base Address	Description	Origin
AHB Memory	0xF0002000 - 0xFFFFFFFF	*Not Populated ROM Table Space*	Mocha-1 Platform
	0xF0000000 - 0xF0001FFF	Mocha1 HSS and VCA ROM tables	
ARM APB Device	0xE0000000 - 0xEFFFFFFF	ARM Private Peripheral bus	ARM *Cortex™*-M0
	0xE000EDF0	Core Debug Base Address (CoreDebug_BASE)	
	0xE000ED00	System Control Block Base Address (SCB_BASE)	
	0xE000E100	Nested Vectored Interrupt Control (NVIC_BASE)	
	0xE000E010	System Tick Timer (SysTick_BASE)	
AHB Peripheral	0xC0008000 - 0xDFFFFFFF	*Not Populated (AHB default slave)*	Mocha-1 Platform
	0xC0000000 - 0xC0007FFF	EEPROM Mapped to Device Space for EEPROM Writing	
APB Peripheral	0xB0000020 - 0xBFFFFFFF	*Not Populated HSS APB Device Space*	
	0xB0000010 - 0xB000001F	Watch Dog Timer	
	0xB000000C - 0xB000000F	Reserved	
	0xB0000000 - 0xB000000B	System Control and WIC status	

	0xA0000080 - 0xAFFFFFFF	*Not Populated Configurable Device Space*	TSX-1001(1)
	0xA0000040 - 0xA000007F	UART	
	0xA0000000 - 0xA000003F	SPI	
AHB Peripheral	0x50001800 - 0x9FFFFFFF	*Not Populated (configured as AHB default slave)*	Mocha-1 Platform
	0x50001000 - 0x500017FF	GPIO_2 (CVL Analog enables and UART setup)	
	0x50000800 - 0x50000FFF	GPIO_1 (CVL PLL and Clock Control)	
	0x50000000 - 0x500007FF	GPIO_0 (CVL External GPIO, SYSCLK output control)	
	0x40002800 - 0x4FFFFFFF	*Not Populated (configured as AHB default slave)*	TSX-1001(1)
	0x40002000 - 0x400027FF	Timer	
	0x40001800 - 0x40001FFF	Pulse Width Modulator	
	0x40001000 - 0x400017FF	ADC	
	0x40000800 - 0x40000FFF	DAC	
	0x40000400 - 0x400007FF	*Not Populated (configured as AHB default slave)*	
	0x40000000 - 0x400003FF	DAC Memory Buffer	
AHB Memory	0x20003000 - 0x3FFFFFFF	*Not Populated (configured as AHB default slave)*	
	0x20000000 - 0x20002FFF	HSS Block RAM	Mocha-1 Platform
	0x00008000 - 0x1FFFFFFF	*Not Populated (configured as AHB default slave)*	
	0x00000F00 - 0x00007FFF	EEPROM	
	0x00000000 - 0x00000EFF	EEPROM or RAM overlay from RAM space 0x20000000-0x20000EFF overlay size controlled by RAM_OVLY	

Note 1: Addresses determined by the TSX-1001 CVL. Other via configurations can alter these addresses, whereas Mocha-1 and ARM *Cortex™*-M0 addresses are fixed by the platform.

Table 3: ARM *Cortex*™-M0 memory map summary.

2 Hardened Subsystem (HSS)

Shown in the block diagram of Figure 1, the Hardened Subsystem (HSS) of the TSX-1001/Mocha-1 platform consists of the ARM *Cortex*™-M0 processor, EEPROM and RAM, as well as processor-support peripherals that include a watch dog timer, clock controller module, GPIO, processor-interface bridge, and memory interfaces. These elements define the core microprocessor subsystem that is common to all configurations of the Mocha-1 platform. The peripherals shown outside the *Cortex*™-M0 Processor, though part of the HSS, are unique to the Mocha-1 platform, and hence, the TSX-1001 as well.

The HSS is described as “hardened”, because unlike the configurable nature of the analog and digital tiles of the Mocha-1 platform, the HSS stays fixed across all CVL designs, including the TSX-1001.

The majority of communication within the HSS, between its processor and peripherals, occurs across the AHB-Lite bus. Exceptions are the Wake-up from Interrupt Controller (WIC) and Watch Dog Timer (WDT) that use the APB interface.

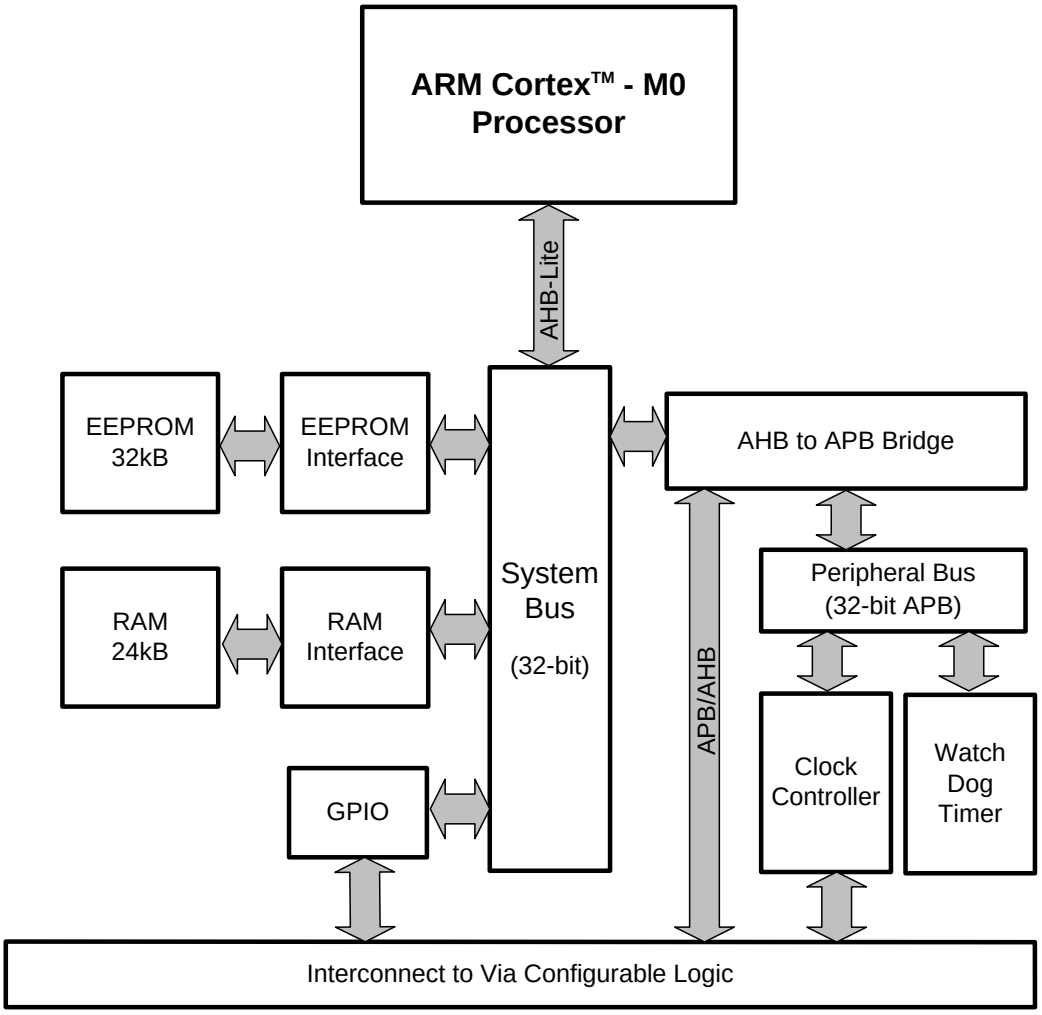


Figure 2 Simplified block diagram of the TSX-1001/Mocha-1 Hardened Subsystem (HSS).

2.1 ARM *Cortex*™-M0 Processor

Optimized for low-power code execution, the ARM *Cortex*™-M0 Processor is the heart of the HSS. As defined by ARM, the *Cortex*™-M0 Processor contains a processor core, a Nested Vectored Interrupt Controller (NVIC), and debugger with SerialWire™ access port (DAP). The HSS also implements the optional system "tick" timer (STT), providing the IC with real-time programming capabilities. For further information on these circuit blocks, the reader is encouraged to consult ARM's documentation (ARM DUI 0497A: "Cortex-M0 Devices: Generic User Guide", 2009).

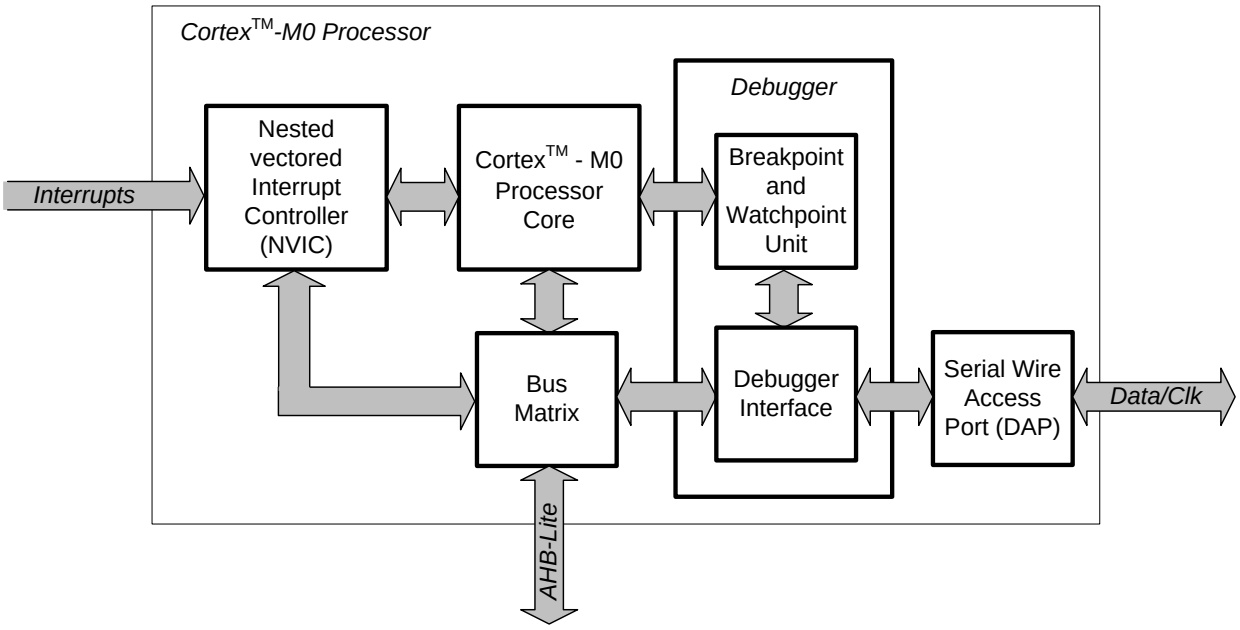


Figure 3: Simplified block diagram of the TSX-1001 ARM *Cortex*™-M0 Processor

2.2 Hardened Subsystem Internal Peripherals

For discussion purposes, the HSS peripherals are grouped according to interface type: AHB and APB.

2.2.1 Hardened Subsystem AHB GPIO Peripherals

Three GPIO registers banks are integrated into the hardened subsystem of the TSX-1001. In addition to monitoring and controlling the external GPIO pins (GPIO_A0 through GPIO_A15), the GPIO registers are also responsible for configuring and/or monitoring certain aspects of some TSX-1001 peripherals.

The lowest register in each of these register banks follows an expanded addressing scheme. Individual control of specific bits is achieved by writing to a specific register offset for its base address. Addressing in this manner reduces the number of instructions required to configure a register. For further information see section entitled “2.2.2 Expanded Bit Addressing.”

2.2.1.1 GPIO0 Register Bank

The GPIO0 Register Bank is a collection of registers whose primary role is to control the behavior of the 16 GPIO pins of the TSX-1001. The bank also controls clock selection for the SYSCLOCK pin.

Pin data is accessed through the **GPIO_DATA** register, while **GPIO_DIR** sets both the output enables for push-pull pins GPIO_A0 through GPIO_A7, as well as enabling GPIO0 interrupts for individual bits of the **GPIO0_DATA** register. Corresponding bits of **GPIO0_IE_RE** and **GPIO0_IE_FE** must also be set for rising or falling edge detection, respectively.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
GPIO0_DATA(1)	0x50000000-0x500003FF	[- - - X]	R/W	GPIO_A7	GPIO_A6	GPIO_A5	GPIO_A4	GPIO_A3	GPIO_A2	GPIO_A1	GPIO_A0
		[- - X -]	R/W	GPIO_A15	GPIO_A14	GPIO_A13	GPIO_A12	GPIO_A11	GPIO_A10	GPIO_A9	GPIO_A8
		[- X - -]	R/W	*Unused*							
		[X - - -]	R/W	SYSCLK_SEL		*Unused*					

GPIO0_DIR	0x50000400-0x50000403	[- - - X]	R/W	OE_A7	OE_A6	OE_A5	OE _A4	OE _A3	OE _A2	OE _A1	OE _A0
		[- - X -]	R/W	GPIO0_EN_IRQ1 [7:0]							
		[- X - -]	R/W	GPIO0_EN_IRQ2 [7:0]							
		[X - - -]	R/W	GPIO0_EN_IRQ3 [7:0]							
--	0x50000404-0x5000040F	ALL	R/W	*Reserved*							
GPIO0_IE_RE	0x50000410-0x50000413	[- - - X]	R/W	GPIO0_IRQ_RE0 [7:0]							
		[- - X -]	R/W	GPIO0_IRQ_RE1 [7:0]							
		[- X - -]	R/W	GPIO0_IRQ_RE2 [7:0]							
		[X - - -]	R/W	GPIO0_IRQ_RE3 [7:0]							
--	0x50000414-0x5000041F	ALL	R/W	*Reserved*							
GPIO0_IE_FE	0x50000420-0x50000423	[- - - X]	R/W	GPIO0_IRQ_FE0 [7:0]							
		[- - X -]	R/W	GPIO0_IRQ_FE1 [7:0]							
		[- X - -]	R/W	GPIO0_IRQ_FE2 [7:0]							
		[X - - -]	R/W	GPIO0_IRQ_FE3 [7:0]							
--	0x50000424 - 0x500007FF	ALL	R/W	*Reserved*							

Note 1: Open drain pads do not have an output enable ability (GPIO0_DIR is not used by output pad) so respective GPIO_ bit written to 0 will drive the I/O pin low regardless of the corresponding OE_ bit, however these bits effect interrupt behavior and corresponding GPIO0_DIR bit must be configured for input to allow interrupts to be generated.

GPIO_A[0 –7]**:** Controls/monitors corresponding general purpose CMOS I/O pins.

GPIO_A[8 –15]: Controls/monitors corresponding general purpose open-drain I/O pins. When low, pin is driven low. When high, pin is released.

SYSCLK_SEL: SYSCLK pin clock select. [00] selects HCLK, [01] selects FCLK, [10] selects PCLK, [11] no clock is presented (SYSCLK pin low. Default).

OE_A[0 – 7]: Connected to output enable of respective general purpose CMOS I/O pins. When high, pin defined as output. When low, pin defined as input (default).

GPIO0_EN_IRQ*n* [7:0]: Enable IRQ generation from corresponding bit in **GPIO0_DATA** register. Valid for bytes [X X X -] only. Active when low. Byte default is 0x00.

GPIO0_IRQ_RE*n* [7:0]: IRQ Enable for Rising Edge in corresponding bit of **GPIO0_DATA** when corresponding **GPIO0_DIR** bit is low.

GPIO0_IRQ_FE*n* [7:0]: IRQ Enable for Falling Edge in corresponding bit of **GPIO0_DATA** when corresponding **GPIO0_DIR** bit is low.

Table 4: GPIO0 register definitions.

2.2.1.2 GPIO1 Register Bank

The GPIO1 Register bank serves as the interface to the TSX-1001's PLL, monitoring and controlling PLL status and behavior (**GPIO1_DATA** register). Like the GPIO0 register bank, independent control over data-generated interrupt polarity is configured through accompanying registers (**GPIO1_IE_RE** and **GPIO1_IE_FE**), though no **GPIO0_DIR** register enabling applies to the GPIO01 register bank.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
GPIO1_DATA	0x50000800-0x50000BFF	[- - - X]	R/W	LOCK_IGNORE	PLL_BYPASS	PLL_LOCKED	PLL_CT_SEL		PLL_CP_SEL		PLL_POL
		[- - X -]	R/W	*Unused*							SWCHK_LOCK
		[- X - -]	R/W	*Unused*							
		[X - - -]	R/W	*Unused*							
--	0x50000C00-0x50000C0F	ALL	R/W	*Reserved*							
GPIO1_IE_RE	0x50000C10-0x50000C13	[- - - X]	R/W	GPIO0_IRQ_RE0 [7:0]							
		[- - X -]	R/W	GPIO0_IRQ_RE1 [7:0]							
		[- X - -]	R/W	GPIO0_IRQ_RE2 [7:0]							
		[X - - -]	R/W	GPIO0_IRQ_RE3 [7:0]							
--	0x50000C14-0x50000C1F	ALL	R/W	*Reserved*							

GPIO1_IE_FE	0x50000C20-0x50000C23	[- - - X]	R/W	GPIO0_IRQ_FE0 [7:0]
		[- - X -]	R/W	GPIO0_IRQ_FE1 [7:0]
		[- X - -]	R/W	GPIO0_IRQ_FE2 [7:0]
		[X - - -]	R/W	GPIO0_IRQ_FE3 [7:0]
--	0x50000C24 - 0x50000FFF	ALL	R/W	*Reserved*

PLL_POL: Invert the polarity of the phase detector. Inverted when high. Non-inverted when low (default). Should be left as low at all times.

PLL_CP_SEL: Charge pump current control (phase detector gain and PLL loop bandwidth control). [00] => 10uA , 165kHz (default); [01] => 5uA , 90kHz; [11] => 15uA, 235kHz; [10] => 20uA , 300kHz. Should be left as [00] at all times.

PLL_CT_SEL: Should be programmed to [00], these bits have the effect of VCO coarse tuning affecting the VCO center frequency. For vtune = 0.9V, [00] => 45 MHz (default); [01] => 40 MHz; [11] => 50MHz; [10] => 55MHz. Should be left as [00] at all times.

PLL_LOCKED: Read-only bit indicating locked state of PLL. High indicates locked. Low indicates unlocked.

PLL_BYPASS: Read-only bit indicating bypass state of PLL. High indicates bypassed. Low indicates running and not bypassed.

LOCK_IGNORE: Indicates to system clock controller that it should ignore the PLL_LOCKED indicator and automatically switch to fast clock when waking from WICDEEPSLEEP modes

SWCHK_LOCK: Indicates to system clock controller HW that it should ignore PLL_LOCKED when waking from WICDEEPSLEEP and system software must perform lock check before toggling SWCHK_LOCK back low or setting LOCK_IGNORE bit.

GPIO0_IRQ_RE*n* [7:0]: IRQ Enable for Rising Edge in corresponding bit of **GPIO1_DATA.**

GPIO0_IRQ_FE*n* [7:0]: IRQ Enable for Falling Edge in corresponding bit of **GPIO1_DATA.**

Table 5: GPIO1 register definitions.

2.2.1.3 GPIO2 Register Bank

Register Bank GPIO2 controls the enables for the analog peripherals of the TSX-1001 (for enables pertaining to the DAC, ADC, and PLL, see the respective sections in this datasheet). The register bank also controls certain UART signals that are derived from its hardware interface.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
GPIO2_DATA	0x50001000-0x500013FF	[- - - X]	R/W	FDA0_EN	SEA0_EN	VCM_EN	VREF1_EN	VREF0_EN	LNA1_EN	LNA0_EN	ANA_PD
		[- - X -]	R/W	*Unused*						CSDAC_CLK_POL	FDA1_EN
		[- X - -]	R/W	*Unused*		BSMSR2	BSMSR3	RIN_CTRL	DSRN_CTRL	DCDN_CTRL	BAUDCLK_EN
		[X - - -]	R/W	*Unused*							
--	0x50001400 - 0x500017FF	ALL	R/W	RESERVED							

ANA_PD: Disables all analog bias currents if programmed high. Disable when high (not recommended). Enabled when low (default).

LNA0_EN: Enable for low-noise amplifier zero. Enabled when high. Disabled when low (default).

LNA1_EN: Enable for low-noise amplifier one. Enabled when high. Disabled when low (default).

VREF0_EN: Enable for external reference VREF0. Enabled when high. Disabled when low (default).

VREF1_EN: Enable for external reference VREF(1). Enabled when high. Disabled when low (default).

VCM_EN: Enable for on-chip common mode generation. Enabled when high. Disabled when low (default).

SEA0_EN: Enable for single-ended amplifier. Enabled when high. Disabled when low (default).

FDA0_EN: Enable for fully differential amplifier zero. Enabled when high. Disabled when low (default).

FDA1_EN: Enable for fully differential amplifier one. Enabled when high. Disabled when low (default).

CSDAC_CLK_POL: invert CSDAC_CLK (i.e. change phase of clock w.r.t. data changes. Falling clock edge in middle of data when low (default).

BAUDCLK_EN: Enable for baud rate clock. Enabled when high. Disabled when low (default).

RIN_CTRL: Assert RI bit of UART **MSR** register. De-asserted when high. Asserted when low (default).

DSRN_CTRL: Assert DSR bit of UART **MSR** register. De-asserted when high. Asserted when low (default).

DCDN_CTRL: Assert DCD bit of UART **MSR** register. De-asserted when high. Asserted when low (default).

BSMCR3: Reserved for Modem Interrupt configuration. Set low (default).

BSMCR2: Reserved for Modem Ring Indicator configuration. Set low (default).

Table 6: GPIO2 register definitions.

2.2.2 Expanded Bit Addressing

The AHB GPIO peripheral provides bit addressability of corresponding GPIO_DATA register through an expanded address space. When performing byte access to write into the GPIO_DATA space, the written value is masked with the address bits [9:2] while preserving prior data for bits that are not enabled (the lower 2 bits define the byte index into a specific word location to which the mask is applied). The mask can be thought of as enables for the corresponding bits. In essence, enabling/disabling a specific bit or bits of a particular byte of a memory location can be performed with a single write instruction—there is no need to mask off the unaffected bits in a separate set of instructions. The most notable overhead of this feature is that a memory space of 2^10 or 1024 (0x3FF) locations must be reserved for such control over a 4-byte register.

With pointer arithmetic, individual bit or bit fields within a byte may be set as desired without a read modify write instruction sequence. Thus for full byte access the GPIO base address is offset by byte_index + 0xFF<<2 (the 0xFF indicates all bits are unmasked and the <<2 shifts the mask into bit positions [9:2]). For single bit access the GPIO base address can be offset by (byte_index + 0x01<<(2+bit_offset)).

In the memory map tables of this document, byte references are given by: (1) [- - - X], indicating the lowest byte of byte_index 0x0, (2) [- - X -] indicating byte_index = 0x1, (3) [- X - -] indicating byte_index=0x2, and [X - - -] indicating the highest byte of byte_index = 0x3. Again, the byte index corresponds to address bits [1:0].

A data structure defined in a high level programming language such as C or C++ corresponding to the expanded address space can abstract this bit addressability and simplify application programming. In general, GPIO base address can be offset with (byte_index + byte_data_mask<<2) allowing the hardware to perform the following masking operation for byte writes to GPIO_DATA:

DATA_byte = (new_value & byte_data_mask) | (DATA_byte & ~byte_data_mask),

Thereby writing the new value to the selected bits of DATA_byte, while leaving the unselected ones alone. This hardware feature can reduce the number of required instructions when bit banging since typically the compiler can optimize any required address arithmetic to constant values. When performing half-word and word writes to the GPIO register this expanded addressing mode of operation is disabled. Registers that are indicated with an address range from Base_Address to Base_Address + 0x3FF are implemented as AHB GPIO peripherals and thus provide this feature.

2.3 Hardened Subsystem APB Peripherals

Certain Mocha-1 peripherals (WIC, memory controllers, and Clock Control Module) are controlled by reads/writes to APB peripheral registers. These registers must be accessed as full 32-bit words.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
MOCHA1_CTRL0	0xB0000000	[- - - X]	R/W	*Unused*					CLK_RATE [2:0]		
		[- - X -]	R/W	*Unused*						APB_RATE [1:0]	
		[- X - -]	R/W	EE_NOWRSTALL	EE_SPEC	EE_WREN	EE_WORD	*Unused*		EE_WAIT [1:0]	
		[X - - -]	R/W	EE_READY	EE_SPECMASK [2:0]			RAM_OVERLAY [3:0]			
MOCHA1_CTRL1	0xB0000004	[- - - X]	R/W	IRQLATENCY [7:0]							
		[- - X -]	R/W	UNUSED							
		[- X - -]	R/W	*Unused*		KPA	STOPREQ	*Unused*			WIC_ENREQ
		[X - - -]	R/W	UNUSED							
MOCHA1_WIC	0xB0000008	[- - - X]	R	WIC_SENSE [7:0]							
		[- - X -]	R	WIC_SENSE [15:8]							
		[- X - -]	R	*Reserved*						WIC_SENSE [17:16]	
		[X - - -]	R	*Reserved*							
--	0xB000000C	[- - - X]	R	*Reserved*							
		[- - X -]	R								
		[- X - -]	R								
		[X - - -]	R								

CLK_RATE [2:0]: Clock rate control Mocha system clock relative to input clock. [000] sixteenth rate clock (default). [001] eighth rate clock. [010] quarter rate clock. [011] half rate clock. [1xx] full rate clock.

APB_RATE [1:0]: Controls the HCLK to PCLK ratio. [00] 1:1 (default). [01] 1:2. [10] 1:3. [11] 1:4.

EE_WAIT [1:0]: Selects the number wait states for EEPROM reads (and non LOAD triggered writes). [00] Zero Wait States. [01] One Wait State. [10] Two Wait States. [11] Three Wait States (default).

EE_WORD: Selects whether EEPROM load is triggered by all EEPROM writes or just writes to last address for each page. [0] Page mode (load triggered by EEPROM write when HADDR[5:2] = [1111] (default). [1] Word mode (load triggered by all EEPROM word writes.

EE_WREN: Selects whether EEPROM write/load is allowed. When low, EEPROM loading is prohibited (default). When high, EEPROM writing/loading is allowed.

EE_SPEC: Selects whether EEPROM speculative reads are allowed. When low, EEPROM speculative reads are disabled (default). When high, EEPROM speculative reads are allowed.

EE_NOWRSTALL: Selects whether EEPROM writing stalls the AHB bus. When low, EEPROM writes stall the AHB bus (default). When high, EEPROM writes allow code to continue running from RAM.

RAM_OVLY [3:0]: RAM_OVLY defines how much of the EEPROM starting at address 0 is overlaid with RAM. The overlay increment is 256 bytes allowing:

0 – no RAM overlay (default)
1 – lower 256 bytes of RAM aliases starting at address 0
2 – lower 512 bytes of RAM aliases starting at address 0
…
15 – lower 3840 bytes of RAM aliases starting at address 0

EE_SPECMASK [2:0]: EE_SPECMASK[2]. [0] Ignore branch hint when EESPEC=1 (default). [1] Suppress speculative reads when branch instruction are in decode.

EE_SPECMASK [1]: When low, ignore backward conditional branch hint when EE_SPEC=1 (default). When high, suppress speculative reads when backward unresolved conditional branch instructions are in decode.

EE_SPECMASK [0]: When low, ignore forward conditional branch hint when EE_SPEC=1 (default). When high, suppress speculative reads when forward unresolved conditional branch instructions are in decode.

EE_READY: Read only status bit indicating EEPROM ready status. When low, indicates EEPROM load is not yet complete. When high, indicates EEPROM is ready to use.

IRQLATENCY [7:0]: Sets the IRQ latency bits of the *Cortex™* M0. [00000000] (default for lowest latency IRQ servicing).

KPA: Control bit defining the behavior of device sleep modes. When high, keep peripherals awake during sleep modes (limits possible modes of operation to DEBUG, RUN, SLEEP_PA, DEEPSLEEP_PA, WICDEEPSLEEP_PA). When low, gate off APB and GPIO peripheral clock during sleep (default) (limits possible modes of operation to DEBUG, RUN, SLEEP, DEEPSLEEP, WICDEEPSLEEP, STOP).

STOPREQ: Control bit to request WICDEEPSLEEP fully stop the clocks for hardened subsystem. VCA/CVL must provide wakeup mechanism and trigger HSS WIC once clocks have restarted. [0] keep a clock source running in WICDEEPSLEEP for wakeup (default). [1] stop all clocks in WICDEEPSLEEP to enter STOP mode (TSX1001 does not implement this optional feature).

WIC_ENREQ: Control bit to request Deep Sleep be WIC based deep sleep. When high, enable WIC based deepsleep mode (default) (limits possible modes of operation to DEBUG, RUN, SLEEP_PA, WICDEEPSLEEP_PA, WICDEEPSLEEP, STOP). When low, disable WIC based deepsleep modes (limits possible modes of operation to DEBUG, RUN, SLEEP_PA , DEEPSLEEP_PA, SLEEP, DEEPSLEEP)

WIC_SENSE: Read-only bits indicating which WIC line woke up the processor
WIC_SENSE[0] = NVIC (Watch Dog IRQ)
WIC_SENSE[1] = RXEV (RX event)
WIC_SENSE[2] = IRQ[0] (GPIO 0 IRQ)
WIC_SENSE[3] = IRQ[1] (GPIO 1 IRQ)
…
WIC_SENSE[17] = IRQ[15]

Table 7: Mocha-1 register definitions.

2.3.1 Watchdog Timer

The Watchdog Timer (WDT) peripheral provides a dedicated timer that automatically resets the TSX-1001 system, should the processor become incapable of periodically resetting the timer. As shown in **Figure 4**, the WDT uses a 16-bit loadable down counter, clocked by a configurable prescaler divider to extend the range of the time out period (the system PCLK serves as the input to the prescaler). If enabled, the system software must periodically issue a special write to the kick or status register (**WD_KICK**, **WD_STATUS**, respectively) to reset the prescaler and reload the down counter to prevent the down counter from reaching the time out value of zero.

The WDT function is optional, through configuration of the WD_EN bit of the **WD_CTRL** register. If enabled, timer has the ability to send a processor NMI a configurable number of prescaler cycles before the time out. The control field NEAR_TO (near timeout) selects the counter value below which an interrupt is indicated. The watch- dog timer is implemented as an APB peripheral and all its registers must be accessed as full 32-bit words.

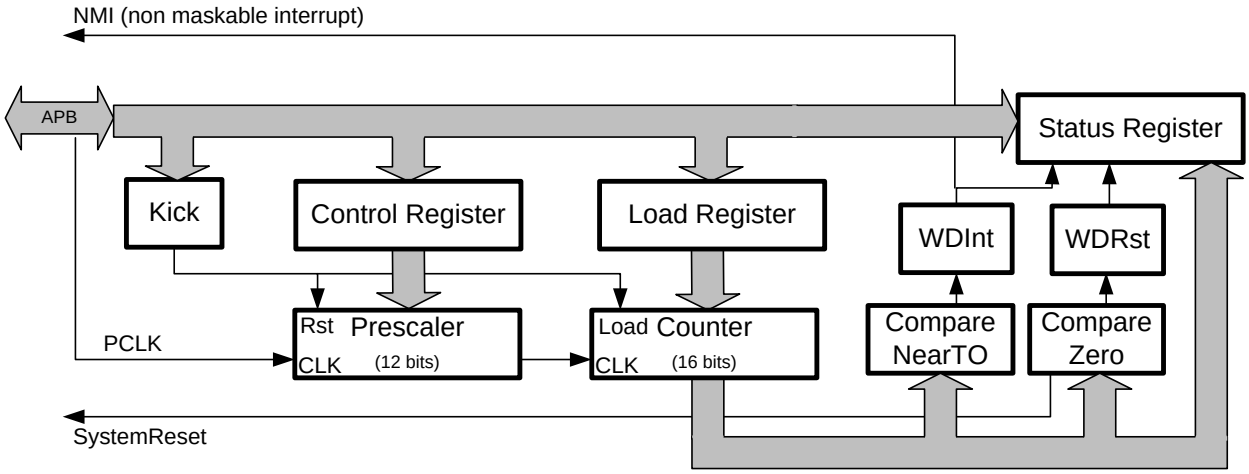


Figure 4: Watch Dog Timer

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
WD_LOAD	0xB0000010	[- - - X]	R/W	LOAD [7:0]							
		[- - X -]	R/W	LOAD [15:8]							
		[- X - -]	R/W	LOADKEY [7:0]							
		[X - - -]	R/W	LOADKEY [15:8]							
WD_CTRL	0xB0000014	[- - - X]	R/W	WD_EN	NEAR_TO [2:0]			PRESCALE_CTRL [3:0]			
		[- - X -]	R/W	RESERVED							
		[- X - -]	W	CTRLKEY [7:0]							
		[X - - -]	W	CTRLKEY [15:8]							
WD_KICK	0xB0000018	[- - - X]	R/W	KICKKEY [7:0]							
		[- - X -]	R/W	KICKKEY [15:8]							
		[- X - -]	R/W	KICKKEY [23:16]							
		[X - - -]	R/W	KICKKEY [31:24]							
WD_STATUS	0xB000001C	[- - - X]	W	STATUSKEY [7:0]							
		[- - X -]	W	STATUSKEY [15:8]							
		[- X - -]	W	STATUSKEY [23:16]							
		[X - - -]	W	STATUSKEY [31:24]							
WD_STATUS	0xB000001C	[- - - X]	R	WD_CNT [7:0]							
		[- - X -]	R	WD_CNT [15:8]							
		[- X - -]	R								
		[X - - -]	R	WD_INT	WD_RESET						

LOAD [15:0]: WDT counter preset value on load or kick

LOADKEY [15:0]: Write these bits as 0x5A5A with new LOAD to enable update

PRESCALE_CTRL [3:0]: Selects watch dog prescaler divide-by value

[0000] Prescaler bypass
[0001] Prescaler = PCLK /2
[0010] Prescaler = PCLK/4
[0011] Prescaler = PCLK/8
[0100] Prescaler = PCLK/16
…
[1011] Prescaler = PCLK/2048
[11xx] Prescaler = PCLK/4096

NEAR_TO [2:0]: Selects the count value below which the WDT will issue an interrupt.

[000] IRQ is not issued before reset
[001] IRQ is issued 256 preScaler clock cycles before reset
[010] IRQ 512 cycles prior to reset
[011] IRQ 1024 cycles prior to reset
[100] IRQ 2048 cycles prior to reset
[101] IRQ 4096 cycles prior to reset
[11x] IRQ 8192 cycles prior to reset

WD_EN: WDT enable. Enabled when high. Disabled when low (default)

KEYCTRL [15:0]: Write these bits as 0x5A5A to enable WDCTRL to allow update

KICKKEY [31:0]: Write 0x5A5A5A5A to initialize watch dog timer.

STATUSKEY [31:0]: Write 0x5A5A5A5A to initialize watch dog timer and clear WD_INT flags. Writing any value clears WD_RESET flag.

WD_CNT [15:0]: WDT counter value.

WD_RESET: WDT Reset occurred-flag. High when reset has occurred. Low when no reset has occurred. Status preserved through WD triggered reset.

WD_INT: WDT interrupt occurred-flag. High when interrupt has occurred. Low when no interrupt has occurred. Cleared after WD triggered reset.

Table 8: Watch Dog Timer (WDT) register definitions.

2.4 Program/Data Memory

The TSX-1001 includes 32 kbytes of EEPROM program/data memory organized as 8k x 32. Internally the memory is implemented with 4 blocks of 4k x 16. The TSX-1001 also includes 24 kbytes of RAM program/data memory organized as 6k x 32, implemented with 12 blocks of 2k x 8. Resident memory controllers translate AHB transactions from the ARM *Cortex*™-M0 to corresponding transactions over the native interfaces for the RAM and EEPROM blocks.

2.4.1 EEPROM

Two EEPROM memory controllers control the 4 EEPROM blocks located at the address range 0x00000000 – 0x00007FFF. The AHB slave controller for the EEPROM supports programmable wait states from 0-3. In register **MOCHA1_CTRL**, field EEWAIT[1:0] selects the wait states value and defaults to 3 wait states. The controller also provides an optional instruction prefetch feature to enhance system performance when operating with clock frequencies above the EEPROM max speed which requires wait states.
The two controllers use interleaved addressing which along with the instruction prefetch feature can minimize performance degradation when wait states are required. Thus bursts of sequential instruction fetches can effectively run without wait states. The EEPROM controller provides an optional mechanism to prevent debug access from allowing machine code images to be made from “released” products.

After Power-on-Reset, the device is configured to select page write mode but with EEPROM writing disabled. To enable EEPROM writing, field EEWR in **MOCHA1_CTRL** must be set high. Once enabled page write mode generates EEPROM LOAD signal when HADDR[6:2] = 0x1F for AHB word write operations to the EEPROM device space alias which starts at base address 0xC0000000. Field EEWORD, also in the **MOCHA1_CTRL**, register defaults to low for selecting page write mode. After power-up the field may be programmed high to allow single-word programming, rather than page mode programming. Writes must occur in 32-bit granularity in the EEPROM device space alias.

With word programming the LOAD signal is generated by all word aligned writes to the EEPROM device alias. Each load operation requires up to 10ms to complete. During EEPROM writes the READY signal goes low and stalls the MCU until the operation is complete.

2.4.1.1 Initial Code Loading

For initial code loading and for time critical applications a mode is also provided to avoid AMBA BUS stalls. The field EENOWRSTALL may be set to 1 to prevent bus stalls when loading EEPROM. This mode requires that all software required to run during the EEPROM loading be loaded into RAM. Alternatively it may be used by the serial wire debug interface without code continuing to run so that the debugger can continue to access other parts of the system besides the EEPROM. The RAM_OVLY field in the **MOCHA1_CTRL0** register provides a means to alias a portion of RAM space down to EEPROM address space from address 0x00000000 up to [(RAM_OVLY[3:0] x 0x100) – 1]. When using the “no stall” feature, application code or serial wire debugger routines must check the EEREADY status before attempting to access to the EEPROM.

Initial code loading for a previously un-programmed part may require certain system registers be initialized. Certain register fields should be considered when developing initialization code for EEPROM writing routines (see Table 9). PLL-ready status is indicated by the **GPIO1_DATA** register. Desired clock speed of the HSS is achieved through configuration of the **MOCHA1_CTRL0** register.

Register	Bit	Status/Configuration
GPIO1_DATA	PLL_LOCKED	Should be read and checked to be “1” consistently over several reads
	SWCHK_LOCK	Should be set to 1 if PLL_LOCKED chatters when first locking. (Defaults low on HW reset or initial powerup (POR) to allow unprogrammed device acknowledge debugger power up request)

	LOCK_IGNORE	Should be set to 1 if PLL_LOCKED does not consistently read back "1" even after locking
MOCHA1_CTRL0	CLK_RATE [2:0]	Should be set to provide desired AHB HCLK frequency
	APB_RATE [1:0]	Should be set to provide desired APB PCLK frequency
	EE_WAIT[1:0]	Should be set to provide any required EEPROM wait states (0-3) is OK if HCLK is between [0:8) MHz (1-3) is OK if HCLK is between [8:16) MHz (2-3) is OK if HCLK is between [16:24) MHz
	EE_WORD	Should be set to 0 for fastest code loading
	EE_WREN	Must be set to 1 to allow EEPROM writing
	EE_SPEC	Default should be OK (recommend setting 1 for running code)
	EE_SPECMASK [2:0]	Default should be OK (recommend setting 4 or 7 for running code from EEPROM)
	RAM_OVLY [3:0]	Default should be OK

Table 9: Key register considerations for EEPROM writing.

2.4.1.2 EEPROM-Access Security

When coming out of reset, and before the processor starts to reboot, the EEPROM controller module fetches the last word in the EEPROM address space (0x7FFC) and compares it to the security key 0xA55A5AA5 to determine whether the Debugger should be locked-out until the EEPROM is fully erased. If security key is present at top of EEPROM address space then debugger has limited access. Essentially a locked out device must be fully erased to regain debugger access. This feature is provided to optionally prevent unauthorized access to an application code image.

2.4.2 RAM

The TSX-1001 implements 24 kBytes of SRAM starting at AHB bus address 0x20000000 and extending through 0x20002FFF. This memory operates with zero wait states and can be used for data and executable code. The SRAM memory supports word, half word and byte load/store operations.

The lower portion of this SRAM may be optionally aliased down to default code space starting at address 0x00000000. This aliasing can be used by developing a simple boot-strap initialization routine to perform a block copy from EEPROM space to SRAM space to then allow executing timing critical code from zero wait state SRAM memory after setting the SRAM_OVLY control bits. Routines that may need to run for some applications even during slow EEPROM updates can also be located in this overlay region so the processor can be initialized to not stall the system bus during EEPROM writing. Applications that need to avoid long bus stalls, up to 10ms for writing an EEPROM page, must load any needed code routines either into the RAM overlay or directly into SRAM space. The SRAM overlay size is configurable in increments of 256 bytes through the 4 bit field RAM_OVLY of register **MOCHA_CTRL0**. This 4-bit field thus allows selection of 0 – 3840 bytes.

3 Digital Peripherals

A variety of digital peripherals are implemented inside the TSX-1001 that include a PWM, timer, UART, and SPI. Although the TSX-1001 accesses these SPI and UART peripherals through the *Cortex*™-M0's APB interface, other peripherals, such as PWM, timer, and data converters, use the faster AHB interface.

3.1 Timing Peripherals

3.1.1 Pulse-Width Modulator (PWM)

The TSX-1001 contains a 30-bit pulse-width modulator (PWM) with independent register control of both period and pulse-width. All divisors are referenced to the GCLK system clock, nominally 22.1184 MHz. For example, to generate a 10 kHz square waveform with 25% duty cycle, the period count register, PWM_DIV, is loaded with a value of 2211 (dec), and the pulse-width register, PWM_DC, is loaded with 552 (decimal). A duty-cycle register value of 0 is always 0% duty cycle. When the duty-cycle register value equals the period count register value, the duty cycle of the output waveform is 100%.

Two register bits can enable/suspend the PWM timer operation. Setting PWM_HLD high halts the PWM counter. When returned to low, the PWM continues counting from where it left off. When PWM_EN is low, the counter also stops; however, when it is returned to high, counting re-initializes and a new divisor count is loaded.

The PWM block has one digital output pin for the pulse-width-modulated signal (POUT). An input pin, /PEN, is available that can mask the PWM output value through a logical AND operation, internal to the TSX-1001 (the counter can still run). Note that software enable inputs PWM_EN and PWM_HLD must be asserted to enable PWM operation, and output will not be presented to the POUT pin, unless the /PEN pin is asserted. Polarity bits, PEN_POL and POUT_POL, set the polarity of the PWM pins /PEN and POUT, with "1" indicating an asserted-when-high signal, and "0" (default) indicating an asserted-when-low signal.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
PWM_CFG0[0]	0x40001800-0x40001BFF	[- - - X]	W	PWM_DC [7:0]							
PWM_CFG0[1]		[- - X -]	W	PWM_DC [15:8]							
PWM_CFG0[2]		[- X - -]	W	PWM_DC [23:16]							
PWM_CFG0[3]		[X - - -]	W	PWM_EN	PWM_HLD	PWM_ DC [29:24]					
PWM_CFG0[0]	0x40001800-0x40001BFF	[- - - X]	R	PWM_CNT [7:0]							
PWM_CFG0[1]		[- - X -]	R	PWM_ CNT [15:8]							
PWM_CFG0[2]		[- X - -]	R	PWM_ CNT [23:16]							
PWM_CFG0[3]		[X - - -]	R	PWM_EN	PWM_HLD	PWM_ CNT [29:24]					
PWM_CFG1[0]	0x40001C00-0x40001C03	[- - - X]	R/W	PWM_DIV [7:0]							
PWM_CFG1[1]		[- - X -]	R/W	PWM_DIV [15:8]							
PWM_CFG1[2]		[- X - -]	R/W	PWM_DIV [23:16]							
PWM_CFG1[3]		[X - - -]	R/W	PEN_POL	POUT_POL	PWM_DIV [29:24]					
--	0x40001C04 - 0x40001FFF	ALL	R/W	RESERVED							

PWM_DC [29:0]: PWM duty cycle count, expressed as the number GCLK pulses. All bits default to 0.

PWM_CNT [29:0]: PWM counter value. All bits default to 0.

PWM_EN: Disable PWM counting and reset when low. Reload divisor count and enable counting when high. Reading reads status of bit. 1:PWM enabled. 0:PWM disabled (default).

PWM_HLD: PWM software enable. Writing to bit sets enable. Reading reads status. 1:PWM enabled. 0:PWM disabled (default).

PWM_DIV [29:0]: PWM clock divisor. Sets PWM frequency. All bits default to 0.

PEN_POL: Sets polarity of /PEN pin. 1: PWM enabled when /PEN high. 0: PWM enabled when /PEN is low (default).

POUT_POL: PWM output when PWM_EN is de-asserted. 1: POUT pin is high when PWM disabled. 0: POUT pin is low when PWM disabled (default).

Table 10: PWM register map

3.1.2 General-Purpose Timer/Counter

The TSX-1001 contains a 30-bit general-purpose timer that can assume three fundamental modes. In Mode 0, the timer count externals external pulses and issues an interrupt after a pre-programmed interval. In Mode 1, the timer counts internal clock cycles of the GCLK. If enabled (TMR_FREE set high), an interrupt is issued at the end of the counting window determined by the status of the gating pin, THLT pin. In Mode 2, the timer functions as a free-running PWM block. Note that timer counter does not saturate, but rather rolls-over after a divisor configurable number of clock cycles unless a halt condition occurs to stall counting. Doing so allows for timer to generate more than one hardware interrupt condition which could be used as control for a software based interrupt event counter.

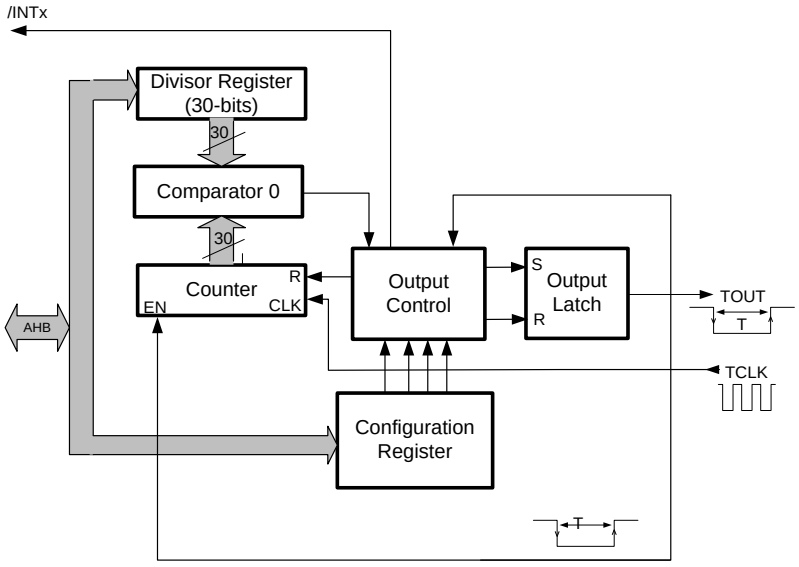


Figure 5: TIMER configuration in Mode 0 (event counting)

In Figure 5, the timer is in event counting mode (Mode 0), where pulses presented at the TCLK pin are counted during an event window (time width, “T” in the figure), determined by the GCLK divided by the number stored in TMR_DC. Because the event window is derived from a repetitive PWM waveform, a number larger than TMR_DC should be written to TMR_DIV to specify the overall period. Prior to counting, the TMR_EN bit should be cleared to reset the counter, if desired, and then set high to enable timer function. The counting interval begins when the TOUT pin goes low. Counting is complete once the TOUT pin returns to high, thereby generating a processor interrupt (IRQ[5]).

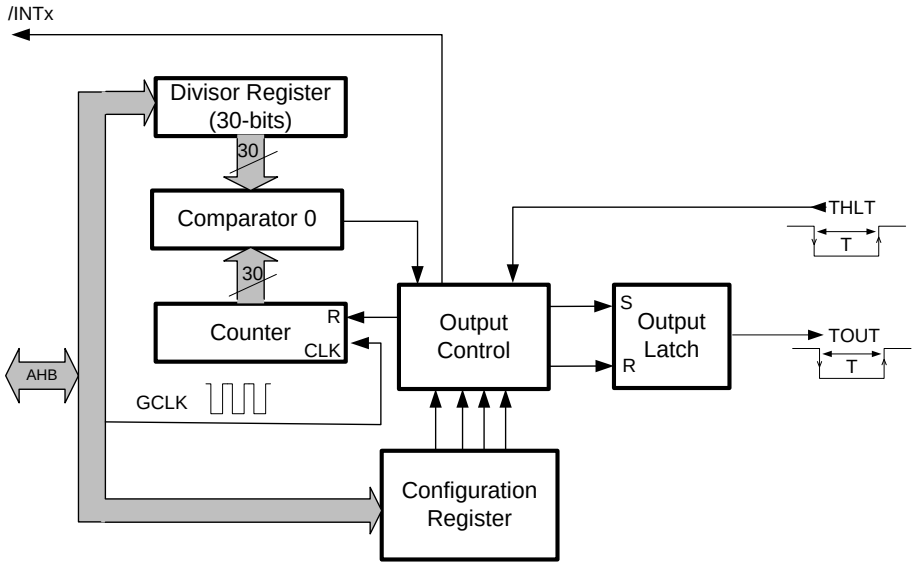


Figure 6: TIMER configuration in Mode 1 (interval timing)

As shown in Figure 6, when the timer is in interval counting mode (Mode 1), the THLT pin acts as a counter enable as well as an interrupt generator. When the THLT goes low, the event counter begins counting at a rate equal the GCLK, until the THLT pin returns to high and inhibits event counting. Throughout operation, TOUT mirrors the status of the THLT pin; hence, when THLT returns high, so does TOUT, thereby issuing a processor interrupt. The processor can then interrogate the event counter. Note that after reading the event counter, the processor must reset the event counter by pulsing the TMR_EN low, in order for the count to begin at 0 again.

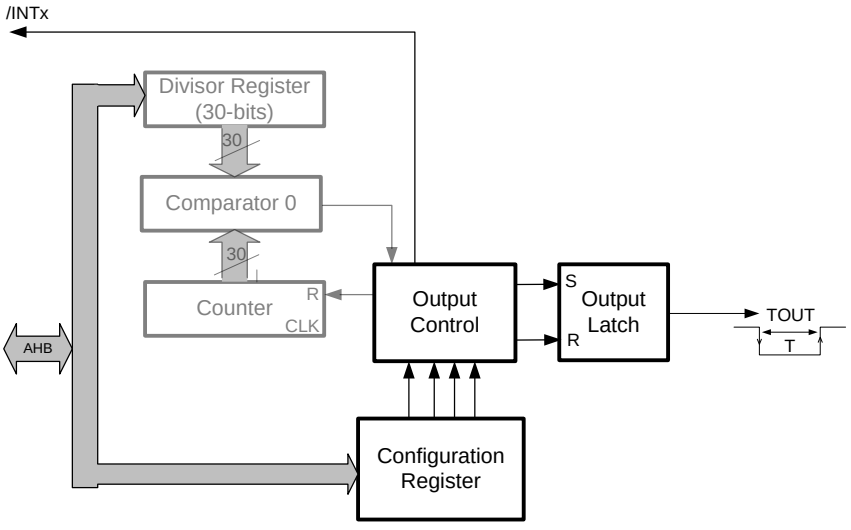


Figure 7: TIMER configuration in Mode 2 (PWM mode)

A third mode exists for operating the timer as a free-running PWM (not gated by THLT). When the TMR_HLT input signal is tied to non-halt state and TMR_MODE bits are set high, the THLT pin is disabled, leaving the timer to operate as a free-running PWM. In this mode, the clock period is determined by the GLCK divided by the value stored in TMR_DIV and the duty cycle is determined by value stored in TMR_DC. An interrupt is generated on the rising edge of the TOUT signal.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
TMR_CFG0[0]	0x40002000-0x400023FF	[- - - X]	W	TMR_DC [7:0]							
TMR_CFG0[1]		[- - X -]	W	TMR_ DC [15:8]							
TMR_CFG0[2]		[- X - -]	W	TMR_ DC [23:16]							
TMR_CFG0[3]		[X - - -]	W	TMR_EN	TMR_HLD	TMR_ DC [29:24]					
TMR_CFG0[0]	0x40002000-0x400023FF	[- - - X]	R	TMR_CNT [7:0]							
TMR_CFG0[1]		[- - X -]	R	TMR_ CNT [15:8]							
TMR_CFG0[2]		[- X - -]	R	TMR_ CNT [23:16]							
TMR_CFG0[3]		[X - - -]	R	TMR_EN	TMR_HLD	TMR_ CNT [29:24]					
TMR_CFG1[0]	0x40002400-0x40002403	[- - - X]	R/W	TMR_DIV [7:0]							
TMR_CFG1[1]		[- - X -]	R/W	TMR_DIV [15:8]							
TMR_CFG1[2]		[- X - -]	R/W	TMR_DIV [23:16]							
TMR_CFG1[3]		[X - - -]	R/W	HLT_POL	TMR_MODE	TMR_ DIV [29:24]					
--	0x40002404 - 0x400027FF	ALL	R/W	RESERVED							

TMR_DC [29:0]: THLT interval counter value. All bits default to 0.

TMR_CNT [29:0]: TMR event counter value. All bits default to 0.

TMR_HLD: Halts timer counter operation. Reading reads bit status. Default 0. 1: TMR halted. 0:TMR not halted (must also be enabled for timer functional operation).

TMR_EN: Enables and disables timer operation. Reading reads bit status. 1:TMR enabled. 0:TMR disabled (default).

TMR_DIV [29:0]: Sets TMR interval in Mode 0. All bits default to 0.

HLT_POL: When high, THLT gates counter when THLT is 1. When low, THLT gates counter when THLT is 0. Reading reads bit status. Default 0.

TMR_MODE: Selects event-counter clocking mode. When high, event counter input derived from GCLK. When low, from TCLK pin (default).

Table 11: TIMER register map

3.2 Interface Peripherals

3.2.1 Universal Asynchronous Receiver/Transmitter (UART)

Designed by SoC Solutions, the UART peripheral of the TSX-1001 is a complete implementation of the widely used 16550 UART. The UART registers are summarized in **Table 12**. Note that these registers are on are on 4-byte boundaries; hence, the 3 high bytes are not used in a given register (the 16550 UART was originally a 1-byte-register part).

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
RBR (Receive Buffer)	0xA0000040	[- - - X]	R	RX_DATA [7:0]							
THR (Transmit Holding)	0xA0000040	[- - - X]	W	TX_DATA [7:0]							
--	0xA0000041-0xA0000043	[X X X -]	R/W	*Unused*							
IER (Interrupt Enable)	0xA0000044	[- - - X]	R/W	*Unused*				STAT_INT	RXLN_INT	TXHLD_INT	RXDAT_INT
	0xA0000045-0xA0000047	[X X X -]	R/W	*Unused*							
IIR	0xA0000048	[- - - X]	R	FIFO_STAT [1:0]		*Unused (set to [00])*		INT_ID [2:0]			INT_PEND

(Interrupt ID)	0xA0000049-0xA000004B	[X X X -]	R	*Unused*							
FCR (FIFO Conrol)	0xA0000048	[- - - X]	W	RX_TRIG_LEV		*Reserved*		DMA_MODE	TX_FIFO_RST	RX_FIFO_RST	FIFO_EN
	0xA0000049-0xA000004B	[X X X -]	W	*Unused*							
LCR (Line Control)	0xA000004C	[- - - X]	R/W	DLAB	SET_BRK	STK_PAR	EVEN_PAR	PAR_EN	NUM_STBITS	WORD_LEN [1:0]	
	0xA000004D-0xA000004F	[X X X -]	R/W	*Unused*							
MCR (MODEM Control)	0xA0000050	[- - - X]	R/W	*Reserved*			LOOPBAK	AUX_OUT		RTS	DTR
	0xA0000051-0xA0000053	[X X X -]	R/W	*Unused*							
LSR (Line Status)	0xA0000054	[- - - X]	R	ERR_RX_FIFO	TX_EMP	TX_HLD_EMP	BREAK_INT	FRAME_ERR	PAR_ERR	OVRRN_ERR	DR
	0xA0000055-0xA0000057	[X X X -]	R/W	*Unused*							
MSR (Modem Status)	0xA0000058	[- - - X]	R	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
	0xA0000059-0xA000005B	[X X X -]	R/W	*Unused*							
SCR (Scratch)	0xA000005C	[- - - X]	R/W	SCRATCH_PAD [7:0]							
	0xA000005D-0xA000001F	[X X X -]	R/W	*Unused*							
DLL (Divisor Latch [LS])	0xA0000040	[- - - X]	R/W	LATCH_DIV [7:0]							
	0xA0000041-0xA0000043	[X X X -]	R/W	*Unused*							
DLM (Divisor Latch [MS])	0xA0000044	[- - - X]	R/W	LATCH_DIV [15:8]							
	0xA0000045-0xA0000047	[X X X -]	R/W	*Unused*							

Note: *All bits are active-high and low by default, unless otherwise noted.*

RX_DATA [7:0]: Received data.
TX_DATA [7:0]: Data to be transmitted.
STAT_INT: Enable MODEM status Interrupt.
RXLN_INT: Enable Receiver-Line-Status Interrupt.
TXHLD_INT: Enable Transmitter-Holding-Register interrupt.
RXDAT_INT: Enable Received-Data-Available interrupt
FIFO_STAT [1:0]: FIFOs enabled.
INT_ID [2:0]: Interrupt ID bits.
INT_PEND: Interrupt pending when low. Default high.
RX_TRIG_LEV: Receiver trigger level.
DMA_MODE: DMA Mode Select.
TX_FIFO_RST: Transmit FIFO reset (self-clearing).
RX_FIFO_RST: Receive FIFO reset (self-clearing).
FIFO_EN: FIFO enable.
DLAB: Divisor Latch Access.
SET_BRK: Set break.
STK_PAR: Stick parity.
EVEN_PAR: Even-Parity select.
PAR_EN: Parity enable.
NUM_STBITS: Number of stop bits.
WORD_LEN [1:0]: Word length select.
LOOPBAK: Loopback mode select.
AUX_OUT: Auxiliary user-designated output.
RST: Request to Send control bit.
DTR: Data Terminal Ready control bit.
ERR_RX_FIFO: Error in Receiver FIFO indicator.
TX_EMP: Transmitter empty indicator.
TX_HLD_EMP: Transmitter holding register empty indicator.
BREAK_INT: Break interrupt.
FRAME_ERR: Framing-Error indicator.
PAR_ERR: Parity-Error indicator.
OVRRN_ERR: Overrun error indicator.
DR: Data-Ready indicator.
DCD: Data-Carrier Detect indicator.
RI: Ring-Indicator indicator.
DSR: Data-Set-Ready indicator.
CTS: Clear-To-Send indicator.
DDCD: Delta-Data-Carrier-Detect indicator.
TERI: Trailing-Edge-Ring-Indicator indicator.
DDSR: Delta Data-Set-Ready indicator.
DCTS: Delta-Clear-To-Send indicator.
SCRATCH_PAD [7:0]: Scratchpad register.
LATCH_DIV [15:0]: Latch Divisor. Bits [7:0] default to 0xFF.

Table 12: UART register definitions.

The baud rate is given by the expression:

$$BaudRate = \frac{SystemClockFrequency}{16 \times DivisorLatchVal},$$

where SystemClockFrequency is the PCLK rate (typically 22.1184 MHz, given a 3.6864 MHz external crystal), and DivisorLatchVal is the corresponding divisor, given as high and low bytes in Table 13 (DLM and DLL, respectively). The divisor values are loaded into the corresponding **DLM** and **DLL** registers of **Table 12**.

Desired Baud Rate (baud)	DLM (dec [hex])	DLL (dec [hex])
300	18 (0x12)	0 (0x00)
600	9 (0x09)	0 (0x00)
1200	4 (0x04)	128 (0x80)
2400	2 (0x02)	64 (0x40)
4800	1 (0x01)	32 (0x20)
9600	0 (0x00)	144 (0x90)
14.4k	0 (0x00)	96 (0x60)
28.8k	0 (0x00)	48 (0x30)
56.0k	0 (0x00)	25 (0x19)
128.0k	0 (0x00)	11 (0x0B)

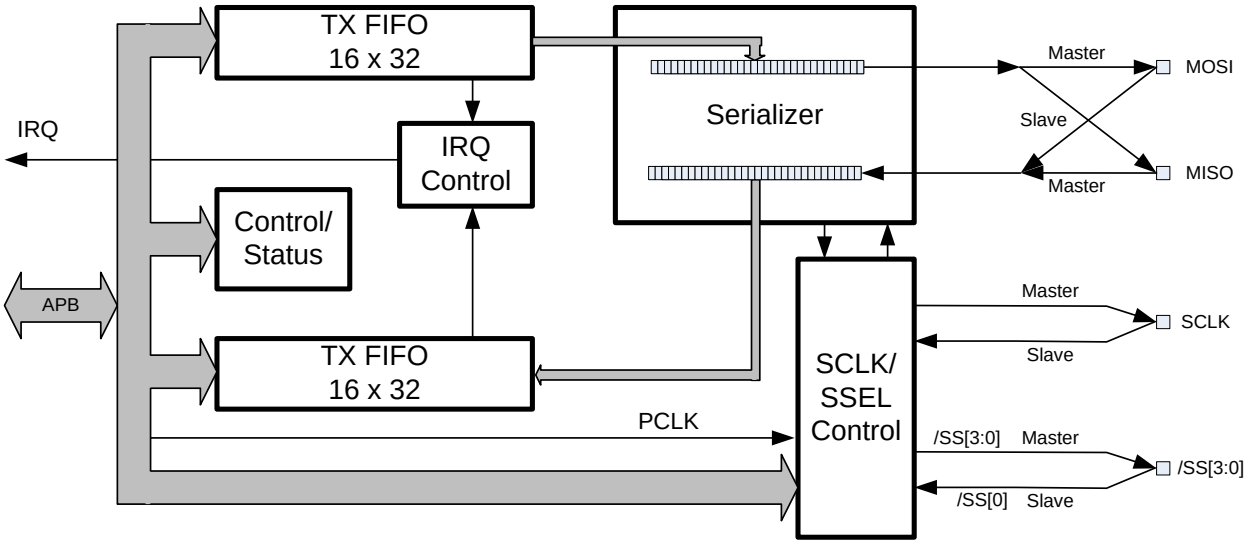
Table 13: Baud rate vs. DLM/DLL values for a PCLK frequency of 22.1184 MHz.

For additional information regarding the UART peripheral the user is referred to the SoC Solutions datasheet ("PiP-EC02: Universal Asynchronous Receiver/Transmitter: AMBA Compatible")

3.2.2 Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) Bus Controller is a synchronous serial data link controller. The design for this IP was provided by SoC Solutions. The SPI bus controller can be configured under software control to be a master or slave device, and configuration and data transfer is performed over the APB bus interface.

The SPI core is a 32-bit peripheral that can operate in 8-bit, 16-bit, and 32-bit data modes. The data is serialized and then transmitted from master to slave device using the standard 4-wire SPI bus interface. The data is transmitted synchronously with the MOSI (Master Out, Slave In) relative to the SCLK generated by the master device. The master also receives data on the MISO (Master In, Slave Out) signal in a full-duplex fashion. When the SPI is configured as a slave, the MISO signal is tri-stated to allow for multiple slaves to transmit data to the master. The slave MISO is active when the slave /SSx control is active (or selected).



Key register and bit definitions are given in **Figure 8**. Further information is available from SoC Solutions (IPC-SPI-APB v2.0: Serial Peripheral Interface (SPI) Controller").

Figure 8: APB SPI peripheral diagram.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
SPI_TXDATA	0xA0000000	[- - - X]	W	TXDATA[7:0]							
		[- - X -]	W	TXDATA [15:8]							
		[- X - -]	W	TXDATA [23:16]							
		[X - - -]	W	TXDATA [31:24]							
SPI_RXDATA	0xA0000004	[- - - X]	R	RXDATA [7:0]							
		[- - X -]	R	RXDATA [15:8]							
		[- X - -]	R	RXDATA [23:16]							
		[X - - -]	R	RXDATA [31:24]							
SPI_CLKDIV	0xA0000008	[- - - X]	R/W	CLKDIV[7:0]							
		[X X X -]	R/W	*Reserved*							
SPI_CONTROL	0xA000000C	[- - - X]	R/W	WORDWIDTH		MASTER	SCLKPOL	LD_EDGETX	MSBFIRST	SAMPLEDATA	ENABLE
		[X X X -]	R/W	*Reserved*							
SPI_STATUS	0xA0000010	[- - - X]	R	RXFULL	RXHALF	RXEMPTY	TXFULL	TXHALF	TXEMPTY	XFERERR	XFERIP
		[X X X -]	R	*Reserved*							
SPI_SSELECT	0xA0000014	[- - - X]	R/W	*Reserved*				SLV_SEL[3:0]			
		[X X X -]	R/W	*Reserved*							
SPI_SSELPOL	0xA0000018	[- - - X]	R/W	*Reserved*				SLV_SEL_POL[3:0]			
		[X X X -]	R/W	*Reserved*							
SPI_IRQENABLE	0xA000001C	[- - - X]	R/W	RXFULL_IE	RXHALF_IE	RXEMPTY_IE	TXFULL_IE	TXHALF_IE	TXEMPTY_IE	XFERERR_IE	SSINSYNC_IE
		[X X X -]	R/W	*Reserved*							
SPI_IRQSTATUS	0xA0000020	[- - - X]	R	RXFULL_IS	RXHALF_IS	RXEMPTY_IS	TXFULL_IS	TXHALF_IS	TXEMPTY_IS	XFERERR_IS	SSINSYNC_IS
		[X X X -]	R	*Reserved*							
SPI_IRQCLEAR	0xA0000024	[- - - X]	W	RXFULL_IC	RXHALF_IC	RXEMPTY_IC	TXFULL_IC	TXHALF_IC	TXEMPTY_IC	XFERERR_IC	SSINSYNC_IC
		[X X X -]	W	*Reserved*							
SPI_TXFIFOCTRL	0xA0000028	[- - - X]	R/W	*Reserved*			TXFIFO_THRESHOLD [4:0]				
		[X X X -]	R/W	*Reserved*							
SPI_RXFIFOCTRL	0xA000002C	[- - - X]	R/W	*Reserved*			RXFIFO_THRESHOLD [4:0]				
		[X X X -]	R/W	*Reserved*							
SPI_TXFIFOLVL	0xA0000030	[- - - X]	R	*Reserved*			TXFIFO_LEVEL [4:0]				
		[X X X -]	R	*Reserved*							
SPI_RXFIFOLVL	0xA0000034	[- - - X]	R	*Reserved*			RXFIFO_LEVEL [4:0]				
		[X X X -]	R	*Reserved*							

Note: All R/W bit fields default to 0 unless otherwise noted. Reading *Reserved* bits returns 0.

TXDATA [31:0]: This is the register interface to the SPI Transmitter. Data width can be 8 bits, 16 bits, or 32 bits. If less than 32 bits are being transmitted upper bits should be written with zeros. This is a write only register, reads are undefined.

RXDATA [31:0]: Data from the SPI receiver to be read by the processor is presented in this register. This is a read only register, write have no effect. Data width can be 8 bits, 16 bits, or 32 bits. If less than 32 bits are being received upper bits will be zeros.

CLKDIV [7:0]: The formula below gives the resulting SPI Master System Clock Frequency as a function of APB Clock Frequency and Divisor Value (from the Divisor Register) SPI_MASTERCLKFREQ = $\frac{\mathbf{PCLK}}{\mathbf{2}\cdot(\mathbf{1}+\mathbf{SPICLKDIV}[\mathbf{7:0}])}$

WORDWIDTH: SPI word width: [00] = 8 bits, [01] = 16 bits, [10]= 32 bits, [11] = Reserved.

MASTER: Master mode; 0 = slave mode, 1 = master mode.

SCLKPOL: SCLK Polarity: 0 = external SCLK is low when not active. 1 = external SCLK is high when not active

LD_EDGETX: Leading Edge Transmit. 0 = First transmit occurs before first edge of SCLK. 1 = First transmit is on first edge of SCLK

MSBFIRST: 0 = LSB First 1 = MSB First

SAMPLEDATA: 0 = Sample incoming data on opposite edge of SCLK from when outgoing data is driven. 1 = Sample incoming data on same edge of SCLK as when outgoing data is driven.

ENABLE: SPI Port Enable. Disabled when low (default). Enabled when high.

RXFULL: Receive FIFO/buffer full flag. Low when Receive FIFO not full. High when Receive FIFO full.

RXHALF: Receive FIFO "half full" flag. Half full is programmable in the watermark register. Low when Receive FIFO is less than half full. High when Receive FIFO is equal to or greater than half full.

RXEMPTY: (default 1) Receive FIFO/buffer empty flag. Low when Receive FIFO is not empty. High when Receive FIFO is empty.

TXFULL: Transmit FIFO/buffer full flag. Low when Transmit FIFO is not full. High when Transmit FIFO is full.

TXHALF: Transmit FIFO "half-full" flag. Half-full is programmable in the watermark register. Low when Transmit FIFO is less than half full. High when Transmit FIFO is equal to or greater than half full.

TXEMPTY: Transmit FIFO/buffer empty flag. Low when Transmit FIFO is not empty. High Transmit FIFO is empty (default).

XFERERR: Transfer Error. Low when no error. High when error. Bit is reset by disabling SPI in the Control Register.

XFERIP: Transfer In Progress Low when no transfer in progress. High when transfer In progress.

SLV_SEL [3:0]: Slave Select signals for up to 4 slaves. A single slave select from the serializer state machine is steered to all selected slaves (ssOut[3:0] outputs) when in Master mode. Note that the polarity of each ssOut signal is individually configured by the Slave Select Polarity register.

SLV_SEL_POL [3:0]:

Master Mode Operation: Slave Select Polarity signals for the ssOut[3:0] signals. Each of the four bits corresponds to a bit in the Slave Select Register.

1 = Corresponding Slave Select signal is active high;
0 = Corresponding Slave Select signal is active low.

Slave Mode Operation: Only the lowest bit (Slave Select Polarity [0]) governs how the ssIn signal is interpreted.

1 = ssIn is interpreted as an active high signal.
0 = ssIn is interpreted as an active low signal.

RXFULL_IE, RXHALF_IE, RXEMPTY_IE, TXFULL_IE, TXHALF_IE, TXEMPTY_IE, XFERERR_IE, SSINSYNC_IE:
Interrupt enables for the corresponding status flag IRQ sources. 1 = enabled interrupt. 0 = disabled interrupt. The SPI interrupt feeds into the M0's NVIC block which must be setup correctly in order to enable the interrupt signal to the processor. Interrupt source SSINSYNC comes from /SS[0] when SPI is in slave mode.

RXFULL_IS, RXHALF_IS, RXEMPTY_IS, TXFULL_IS, TXHALF_IS, TXEMPTY_IS, XFERERR_IS, SSINSYNC_IS:
Interrupt status for the corresponding IRQ flag.

RXFULL_IC, RXHALF_ IC, RXEMPTY_ IC, TXFULL_ IC, TXHALF_ IC, TXEMPTY_ IC, XFERERR_ IC, SSINSYNC_IC:
Interrupt clear for the corresponding interrupt flag. Writing 1 in corresponding bit in SPI_IRQCLEAR bit clears the IRQ flag.

TXFIFO_THRESHOLD [4:0]: Defines level of Half Full Flag in Transmit FIFO. Default is 8.

RXFIFO_THRESHOLD [4:0]: Defines level of Half Full Flag in Receive FIFO. Default is 8.

TXFIFO_LEVEL [4:0]: Receive FIFO Level – Indicates current fill level of Transmit FIFO.

RXFIFO_LEVEL [4:0]: Receive FIFO Level – Indicates current fill level of Receive FIFO.

Table 14: SPI register definitions.

4 Mixed-Signal Peripherals

Out of many possible mixed-signal peripherals that can be implemented on the Mocha-1 platform, the TSX-1001 implements three representative circuits: a precision 16-bit sigma-delta ADC, a high-speed 12-bit current-steering DAC, and a system clock generator PLL with associated reference oscillator. The ADC and DAC are available for general applications. While the PLL is programmable, it is dedicated for system clock generation purposes.

4.1 Sigma-Delta ADC

The TSX-1001 sigma-delta ADC is second-order, differential converter operating at a nominal 1 Msps sampling rate (921 ksps when using an external 3.6864 MHz crystal) and a decimation rate ranging from 64 to 1024. It is designed for accurate conversions between DC and 3.74 kHz. The actual bandwidth and ENOB depends on the clock and decimation rates specified.

4.1.1 ADC Configuration

By adjusting the decimation rate, different effective conversion resolutions can be achieved at the expense of conversion bandwidth (see Table 15). Decimation rate is selected by writing the desired decimation code to the DECI[2:0] bits of **ADC_CFG0[0]**.

By changing the FREQ[1:0] bits of ADC_CFG1, the sigma-delta ADC has the ability to (over)sample at rates of 3.684 Msps (write [0,1] to FREQ[1:0]) and 11.052 Msps (write [1,0] to FREQ[1:0]). Increasing the sampling rate allows proportionately more bandwidth through the converter. Though increased sampling rates are possible, the ADC performance has not been evaluated under these conditions.

Modulator Clock Frequency (MHz)	Decimation Rate	Decimation Binary Code	Output Sample Rate (sps)	Resolution (ENOB)	Maximum Bandwidth (Hz)	OSR
0.921	64 (default)	000	14400	13.5	3744	123
	128	001	7200	14.5	1872	246
	256	010	3600	15	936	492
	512	011	1800	15.5	468	984
	1024	100	900	16	234	1969
	Reserved	101, 110, 111				
3.6864	64 (default)	000	57600	13.5	14976	123
	128	001	28800	14.5	7488	246
	256	010	14400	15	3744	492
	512	011	7200	15.5	1872	984
	1024	100	3600	16	936	1969
	Reserved	101, 110, 111				
11.0592	64 (default)	000	172800	13.5	44928	123
	128	001	86400	14.5	22464	246
	256	010	43200	15	11232	492
	512	011	21600	15.5	5616	984
	1024	100	10800	16	2808	1969
	Reserved	101, 110, 111				

Table 15: Experimental ADC Performance: Decimation rate vs. Resolution and Bandwidth

The sigma-delta ADC has a two enable bits in the ADC_CFG1 register. ADC_EN enables/disables operation of the converters, while ADC_MOD_EN further disables the converter by powering down the analog portion of the circuit.

Two internal references are available to the ADC by changing the ADC_REFSEL bit of the **ADC_CFG1** register. The former enables ratiometric measurements relative to the analog supply (ADC_REFSEL=1 [default]), while the other providing absolute measurements relative to the TSX-1001's bandgap's output voltage (ADC_REFSEL=0).

EDGE_SEL bit permits sampling on the opposite ADC clock phase. DATA_SEL and CLK_SEL allow capture flops in digital domain to clock on rising edge of GCLK (DATA_SEL / CLK_SEL = 0) or falling edge of GCLK (DATA_SEL / CLK_SEL = 1). These bits should remain at the default values.

Once the ADC is properly configured and enabled, conversions occur at the output sampling rate. The EOC bit of the **ADC_CFG1** register goes high once a conversion is complete and a processor interrupt is generated (IRQ 2). Reading the top 2 bytes of **ADC_CFG0** returns the conversion value. The EOC and interrupt is cleared by writing any value to **ADC_CFG1**.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
ADC_CFG0[0]	0x40001000-0x400013FF	[- - - X]	W	FREQ_SEL [0]	CLK_SEL	DATA_SEL	EDGE_SEL	DECI[2:0]			ADC_EN
ADC_CFG0[1]		[- - X -]	W						ADC_MOD_EN	ADC_REFSEL	FREQ_SEL [1]
ADC_CFG0[2]		[- X - -]	W								-
ADC_CFG0[3]		[X - - -]	W								-
ADC_CFG0[0]	0x40001000-0x400013FF	[- - - X]	R	FREQ_SEL [0]	CLK_SEL	DATA_SEL	EDGE_SEL	DECI[2:0]			ADC_EN
ADC_CFG0[1]		[- - X -]	R	EOC					ADC_MOD_EN	ADC_REFSEL	FREQ_SEL [1]
ADC_CFG0[2]		[- X - -]	R	ADC_VAL [7:0]							
ADC_CFG0[3]		[X - - -]	R	ADC_VAL [15:8]							
ADC_CFG1	0x40001400	ALL	R	RESERVED							

ADC_CFG1	0x40001400	ALL	W	ADC_CLR
--	0x40000404 - 0x400007FF	ALL	R/W	RESERVED

ADC_EN: ADC enable. Enabled:(1) Disabled: 0 (default).

DECI [2:0]: Decimation rate. See Table 15 for mapping and default.

EDGE_SEL: Selects ADC's comparator latch polarity. Positive: 1. Negative: 0 (default).

ADC_REFSEL: Selects the ADC reference source: AVDD/2: 1. VBG: 0 (default).

ADC_MOD_EN: Enables analog portion of the ADC. Enabled: 1. Disabled: 0 (default).

CLK_SEL: Inverts the ADC clock source capture polarity. Rising edge:0 (default). Falling edge: 1.

DATA_SEL: Inverts the ADC data source capture polarity. Rising edge:0 (default). Falling edge: 1.

FREQ_SEL: Selectes the ADC clock frequency: [00]: GCLK/4 (default), [01]: GCLK, [10]: GCLK*3, [11]: Reserved.

ADC_VAL [15:0]: ADC output value.

EOC: ADC End-of Conversion. Conversion complete: (1)

ADC_CLR: Writing to this address clears the IRQ and the EOC bit. Any value can be written.

Table 16: ADC register definitions.

4.1.2 ADC ELECTRICAL CHARACTERISTICS

T_A = 25°C, AVDD = VDD = 3.3V, fs = 921 ksps, V_{ref} = AVDD/2 = 1.65V. All values determined by design, unless otherwise noted.

Specification	Note	Min	Typ	Max	Unit
UGBW	Internal amplifiers/integrators		50		MHz
Opamp offset	Internal amplifiers/integrators		1		mV
Full Scale Input Voltage	No integrator overloading		3.2		Vpp differential
Resolution (ENOB)	14.4 ksps		13.5		bits
	7.20 ksps		14.5		bits
	3.60 ksps		15		bits
	1.80 ksps		15.5		bits
	0.90 ksps		16		bits
Decimation Rate (DR) control range	64, 128, 256, 512, 1024	64		1024	
SDM clock rate, f_s	Sys Clk ÷4, x1, x3	0.921		11.059	MHz
Output sample rate	0.921 MHz SDM clock rate	0.900		173	ksps
Decimation filter 3dB BW			0.26x		Output sample rate
Input sampling capacitor	Does not include pad capacitance.		1		pF

Table 17: ADC electrical characteristics.

4.1 Current-Steering DAC

The TSX-1000 has a 12-bit current-steering DAC with conversion rates up to 11 MSps. The DAC clock is derived from the system clock (GCLK) which is typically 22.1184 MHz during normal operation (when using the suggested 3.6864 MHz external crystal). Individual divisor registers can slow the DAC update time by a programmable 24-bit factor. Samples are converted by writing to a memory that is configurable to be used as a FIFO or circular buffer with up to 256 entries. An interrupt can be issued when the buffer contains less than a pre-set number of samples. The DAC buffer can also operate in a circular, free-running mode, where the buffer samples are converted in sequence at the programmed clock rate. This mode can be used in applications where relatively simple repetitive waveforms are desired.

With an LSB current of 0.500 uA, the maximum output current sourced from OUT0+ and OUT0- is 2.0475 mA. When the DAC converts a code of 0x000, OUT0+ sources 0uA, while OUT0- sources 2.0475 mA. When the DAC converts a code of 0xFFF, OUT0+ sources 2.0475 uA, and OUT0- sources 0uA.

Because a differential current steering DAC outputs at constant total current between its output terminals (in this case approximately 2.048mA), DAC power is consumed, even when converting low values. For this reason, applications should make use of the DAC enable (DAC_EN) when the DAC output is not required or when power savings is desired. Another option to save power is to convert the DAC output to a voltage using a trans-impedance amplifier, then perform a sample-and-hold on the resulting voltage. For maximum operating speed, pin capacitance should be less than 10 pF, though larger capacitances are allowable for slower conversion rates.

4.1.1 DAC Configuration

The DAC is initialized by setting the enable bits, DACWR_EN and DAC_EN, high. DAC memory access initialization is handled by setting MEMB_EN to 0 (default) and MEMA_EN high, which select whether AHB or DAC Interface Module is controlling A/B ports of DAC Buffer Memory. For functional mode of DAC MEMB_EN must be set to 0 to allow DAC access to buffer memory. DAC Buffer Memory shall be accessed by AHB address space 0x40000000 – 0x400003FF using address offsets of 0x4 (0x40000000, 0x40000004, …, 0x400003FC).

The DAC can be used in two modes: Direct (FIFO) and Circular-Buffer. In direct mode is entered by setting RDEMPTY to 0. In either mode, the circular buffer manages the read pointer for every data sample that is read and MCU must handle buffer write pointer for data written due the buffer memory implementation. To prevent buffer underflow, a processor interrupt can be created by setting the IRQ_THRESH to the minimum number of remaining sample data to be output—the processor can be made to respond by transferring another batch of data. For example, if IRQ_THRESH is set to 32 and the full 256-sample buffer size is used, the DAC buffer will issue an interrupt when less than 33 data samples remain in the buffer. The processor can then download another 256 – 32 = 224 samples without over-running the buffer. In buffer FIFO mode, MCU must handle buffer write pointer in software to put data in appropriate location to be read by DAC controller. If FIFO is set to a depth of one sample, the MCU may write to same memory for each sample, therefore off loading requirement of managing sample buffer write pointer.

In Circular-Buffer mode, a buffer of 256 entries is used to sequentially write a batch of pre-specified data samples. This mode can be used in applications where repetitive waveforms are required. The mode is entered by setting RDEMPTY to (1) The size of the buffer can be made smaller than 256 by setting the READ_ADDR_LIM bits to the maximum desired buffer size. In this way, data is restricted to being fetched from buffer addresses 0 through READ_ADDR_LIM. Note that the circular buffer is not initialized to a particular value at start-up, and hence, must be configured with appropriate data before using.

The DAC clock is determined by the GCLK frequency divided by one plus the 24-bit integer stored in the DAC_DIV[23:0] bits of the control register. A value of “1” divides the GCLK by 2, and a value of “2” divides the GCLK by 3, etc. Note that the minimum value of DAC_DIV is “1”, due to the 2-cycle access of the memory buffer. For proper DAC operation, the sample rate control register should only be modified while the DAC is disabled.

Reg Name	Address	Byte Ref	R/W	B7	B6	B5	B4	B3	B2	B1	B0
DAC_CFG0[0]	0x40000800-0x40000BFF	[- - - X]	W	IRQ_THRESH [3:0]					RDEMPTY	DACWR_EN	DAC_EN
DAC_CFG0[1]		[- - X -]	W	-	MEMB_EN	MEMA_EN	-	IRQ_THRESH [8:4]			
DAC_CFG0[2]		[- X - -]	W	-	-	-	-	-	-	-	-
DAC_CFG0[3]		[X - - -]	W	-	-	-	-	-	-	-	-
DAC_CFG0[0]	0x40000800-0x40000BFF	[- - - X]	R	IRQ_THRESH [3:0]					RDEMPTY	DACWR_EN	DAC_EN
DAC_CFG0[1]		[- - X -]	R	-	MEMB_EN	MEMA_EN	-	IRQ_THRESH [8:4]			
DAC_CFG0[2]		[- X - -]	R	DATA_CNT [3:0]				DAC_THRESH	-	-	-
DAC_CFG0[3]		[X - - -]	R	-	-	-	DATA_CNT [8:4]				
DAC_CFG1[0]	0x40000C00	[- - - X]	R/W	DAC_DIV [7:0]							
DAC_CFG1[1]		[- - X -]	R/W	DAC_DIV [15:8]							
DAC_CFG1[2]		[- X - -]	R/W	DAC_DIV [23:16]							
DAC_CFG1[3]		[X - - -]	R/W	READ_ADDR_LIM [7:0]							
--	0x40000C04 - 0x40000FFF	ALL	R/W	RESERVED							

DAC_EN: DAC enable. Enabled:1 (default). Disabled: 0.

DACWR_EN: Enables writing to DAC buffer. Enabled:1 (default). Disabled: 0.

RDEMPTY: Circular buffer enable. 0 in Normal mode. 1 in Circular-Buffer mode. Default: 0.

IRQ_THRESH [9:0]: Holds threshold number of samples. When <= IRQ_THRESH number of samples are in the buffer, an interrupt is issued. All bits default to 0.

MEMA_EN: DAC memory buffer Port A enable, 1 – enables AHB access. Default: 0.

MEMB_EN: DAC memory buffer Port B enable, 1 – enables AHB access. Default: 0.

DAC_THRESH: DAC threshold indicator. 0 when <= IRQ_THRESH values in buffer. 1 otherwise.

DATA_CNT[8:0]: Value of current number of samples in DAC sample buffer.

DAC_DIV[23:0]: Sample-rate divisor of GCLK sample clock. Default 0.

READ_ADDR_LIM [7:0]: Maximum buffer address before recycling. All bits default to 0.

Table 18: DAC register map summary

4.1.2 DAC ELECTRICAL CHARACTERISTICS

T_A = 25°C, AVDD = VDD = 3.3V, fs = 10 Msps, V_{ref} = AVDD/2 = 1.65V. All values determined by design, unless otherwise noted.

Specification	Note	Min	Typ	Max	Unit
Resolution	Fin = 1kHz		12		bits
Conversion Rate				11	Msps
Conversion Latency			2.5		Clock cycles
Full-Scale Differential Output Current			2.0475		mA
Integral Linearity (INL)			TBD		LSB
Differential Linearity (DNL)			TBD		LSB
Output capacitance	Does not include pad capacitance. Based on capacitor reset noise sqrt(kT/C)		5		pF

Table 19: DAC electrical characteristics.

5 Analog Peripherals

5.1 Operational Amplifiers

Three op-amp types are available on the generic Mocha-1 platform: general-purpose (12), low-noise (8), and fully differential (16). When implementing a CVL, these amplifiers can be independently configured to optimize selected characteristics, such as speed, power consumption, bandwidth, common-mode range, etc. Additional options allow for multiple configurations that can be configured dynamically. For the TSX-1001 CVL, the general-purpose and fully differential amplifiers have all been configured for maximum output drive, widest bandwidth, and widest common-mode range.

The TSX-1001 provides external access to 1 general-purpose, 2 low-noise, and 2 fully differential op-amps with characteristics given in Table 20 through 11. Two more fully differential and four more single-ended amplifiers are used to implement other peripherals (ADC and voltage references). In total, roughly 30% of the amplifiers resources were used to implement the TSX-1001 analog/mixed-signal peripherals, leaving 70% available for additional future CVL functions (amplifier usage is a good metric for inferring future CVL function expansion, as it is generally the most limiting factor).

T_A = 25°C, AVDD = VDD = 3.3V, AVSS = VSS = 0V and V_{CM} = AVDD/2 = 1.65V. Z_{LOAD} = 10 MΩ || 2pF, by default. All values determined by design, unless otherwise noted.

Paramter	Specification	Conditions		Min	Typ	Max	Unit
V_{DD}	Positive Supply Voltage			3.0	3.3	3.6	V
V_{SS}	Negative Supply Voltage			0	0	0	V
I_{DD}	Supply Current				785		µA
P_{DD}	Power Consumption				2.59		mW
V_{OS}	Input Offset Voltage			-3		3	mV
CMIRL	Common-Mode Input Voltage Low			0	0.05		V
CMIRH	Common-Mode Input Voltage High				3.25	3.3	V
VOUTRL	Output Voltage Low			0	0.05		V
VOUTRH	Output Voltage High				3.25	3.3	V
I_{SOURCE}	Output Source Current				13.5		mA
I_{SINK}	Output Sink Current				16.1		mA
A_0	Open Loop Voltage Gain				105		dB
GBW	Gain Bandwidth	C_{load} = 2 pF			56		MHz
		C_{load} = 30 pF			29		MHz
Φ_M	Phase Margin	C_{load} = 2 pF			85		degree
		C_{load} = 30 pF			35		degree
A_M	Gain Margin	C_{load} = 2 pF			17		dB
		C_{load} = 30 pF			21		dB
R_0	Output Resistance	f = 1 kHz			3.6		kΩ
CMRR	Common Mode Rejection Ratio	f = 1 kHz			128		dB
$PSRR_{VDD}$	Power Supply Rejection Ratio (V_{DD})	f = 1 kHz			88		dB
$PSRR_{VSS}$	Power Supply Rejection Ratio (V_{SS})	f = 1 kHz			88		dB
THD	Total Harmonic Distortion	V_{IN} = 1.1 V_{pk-pk}, f = 1 kHz, unity gain			0.001		%
En	Input Referred Noise	f = 10 Hz			1078		nV/√Hz
		f = 1 kHz			108		nV/√Hz
		f = 100 kHz			18		nV/√Hz
SR	Slew Rate (10% to 90%)	V_{IN} = 20 mV_{pk-pk}	Unity gain configuration C_{load} = 2 pF		2.47		V/µs
		V_{IN} = 1 V_{pk-pk}			64.4		V/µs
T_S	Settling Time to 1% of input step	V_{IN} = 20 mV_{pk-pk}			2.46		ns
		V_{IN} = 1 V_{pk-pk}			19.8		ns
OS	Overshoot	V_{IN} = 20 mV_{pk-pk}			0.001		%
		V_{IN} = 1 V_{pk-pk}			0.001		%

Table 20: General-Purpose Op-Amp electrical characteristics.

T_A = 25°C, AVDD = VDD = 3.3V, AVSS = VSS = 0V and V_{CM} = AVDD/2 = 1.65V. Z_{LOAD} = 10 MΩ || 30pF, by default. All values determined by design, unless otherwise noted.

Paramter	Specification	Conditions	Min	Typ	Max	Unit
V_{DD}	Positive Supply Voltage		3.0	3.3	3.6	V
V_{SS}	Negative Supply Voltage		0	0	0	V
I_{DD}	Supply Current			272		µA
P_{DD}	Power Consumption			897		µW
V_{OS}	Input Offset Voltage		-3		3	mV
CMIRL	Com.-Mode Input Voltage Low		0	0.05		V
CMIRH	Com.-Mode Input Voltage High			2.4	2.5	V
VOUTRL	Output Voltage Low		0	0.05		V
VOUTRH	Output Voltage High			3.25	3.3	V
I_{SOURCE}	Output Source Current			27		mA
I_{SINK}	Output Sink Current			24		mA
A_0	Open Loop Voltage Gain			100		dB
GBW	Gain Bandwidth	C_{load} = 2 pF		2.2		MHz

		C_{load} = 30 pF			2.2		MHz
Φ_M	Phase Margin	C_{load} = 2 pF			70		degree
		C_{load} = 30 pF			61		degree
A_M	Gain Margin	C_{load} = 2 pF			30		dB
		C_{load} = 30 pF			28		dB
R_0	Output Resistance	f = 1 kHz			1.2		kΩ
CMRR	Common Mode Rejection Ratio	f = 1 kHz			108		dB
$PSRR_{VDD}$	Power Supply Rejection Ratio (V_{DD})	f = 1 kHz			30		dB
$PSRR_{VSS}$	Power Supply Rejection Ratio (V_{SS})	f = 1 kHz			68		dB
THD	Total Harmonic Distortion	V_{IN} = 1.1 V_{pk-pk}, f = 1 kHz, unity gain			0.1		%
En	Input Referred Noise	f = 10 Hz			76		nV/√Hz
		f = 1 kHz			20		nV/√Hz
		f = 100 kHz			18		nV/√Hz
SR	Slew Rate (10% to 90%)	V_{IN} = 20 mV_{pk-pk}	Unity gain configuration C_{load} = 30 pF		0.12		V/µs
		V_{IN} = 1 V_{pk-pk}			2.0		V/µs
T_S	Settling Time to 1% of input step	V_{IN} = 20 mV_{pk-pk}			35		ns
		V_{IN} = 1 V_{pk-pk}			705		ns
OS	Overshoot	V_{IN} = 20 mV_{pk-pk}			8.2		%
		V_{IN} = 1 V_{pk-pk}			3.6		%

Table 21: Low-Noise Op-Amp electrical characteristics.

T_A = 25°C, AVDD = VDD = 3.3V, AVSS = VSS = 0V, and V_{CM} = AVDD/2 = 1.65V. Z_{LOAD} = 10 MΩ || 30pF, by default. All values determined by design, unless otherwise noted.

Paramter	Specification	Conditions	Min	Typ	Max	Unit
V_{DD}	Positive Supply Voltage		3.0	3.3	3.6	V
V_{SS}	Negative Supply Voltage		0	0	0	V
I_{DD}	Supply Current			782		µA
P_{DD}	Power Consumption			2.58		mW
V_{OS}	Input Offset Voltage		-3		3	mV
CMIRL	Com.-Mode Input Voltage Low		0	0.05		V
CMIRH	Com.-Mode Input Voltage High			3.25	3.3	V
VOUTRL	Output Voltage Low		0	0.05		V
VOUTRH	Output Voltage High			3.25	3.3	V
I_{SOURCE}	Output Source Current	Output short to V_{SS} with V_{IN} = V_{DD} in unity gain configuration.		13.2		mA
I_{SINK}	Output Sink Current	Output short to V_{DD} with V_{IN} = V_{SS} in unity gain configuration.		15.7		mA
A_0	Open Loop Voltage Gain			113		dB
GBW	Gain Bandwidth	C_{loadDM}= 1 pF, C_{loadCM}= 30 pF		52		MHz
Φ_M	Phase Margin	C_{loadDM}= 1 pF, C_{loadCM}= 30 pF		75		degrees
A_M	Gain Margin	C_{loadDM}= 1 pF, C_{loadCM}= 30 pF		15		dB
R_0	Output Resistance	C_{loadDM}= 1 pF, C_{loadCM}= 30 pF, f = 1 kHz		3.6		kΩ
CMRR	Common Mode Rejection Ratio	C_{loadDM}= 1 pF, C_{loadCM}= 30 pF, f = 1 kHz		80		dB
$PSRR_{VDD}$	Power Supply Rejection Ratio (V_{DD})	C_{loadDM}= 1 pF, C_{loadCM}= 30 pF, f = 1 kHz		43		dB
$PSRR_{VSS}$	Power Supply Rejection Ratio (V_{SS})	C_{loadDM}= 1 pF, C_{loadCM}= 30 pF, f = 1 kHz		64		dB
En10Hz	Input Referred Noise at 10 Hz			3410		nV/√Hz
En1KHZ	Input Referred Noise at 1 kHz			110		nV/√Hz
En100KHZ	Input Referred Noise at 100 kHz			18		nV/√Hz

SR	Slew Rate (10% to 90%)	V_{IN} = 20 mV_{pk-pk}	Unity gain configuration C_{loadDM} = 2 pF C_{loadCM} = 30 pF		1.48		V/µs
		V_{IN} = 1 V_{pk-pk}			73.5		V/µs
T_S	Settling Time to 1% of input step	V_{IN} = 20 mV_{pk-pk}			50		ns
		V_{IN} = 1 V_{pk-pk}			62		ns
OS	Overshoot	V_{IN} = 20 mV_{pk-pk}			9.5		%
		V_{IN} = 1 V_{pk-pk}			13		%

Table 22: Fully Differential Op-Amp electrical characteristics.

5.2 Current Biasing and Voltage References

All analog and mixed-signal components are powered by mirroring currents from the IBIAS circuit of the TSX-1001. These currents are generated by mirroring a PTAT current, created by placing a bandgap voltage across a polysilicon resistor. The IBIAS circuit has its own enable bit that can disable power to the analog section (see Section 2.2.1.3); however, this bit is rarely used during normal operation, as it stops PLL operation, and hence all IC functions.

Two positive, buffered voltage references are externally available on the TSX-1001. VREF0 provides a temperature-stabilized bandgap output voltage of 1.21V (nominal), while VREF1 provides a buffered AVDD/2 reference voltage. Both references are individually controllable through separate enable bits as controlled by the GPIO2 register (also Section 2.2.1.3). However, VREF1 has an additional enable bit to control propagation of the AVDD/2 voltage internally throughout the TSX-1001 (VREF0 does not have an internal enable, since it is an integral part of the IBIAS circuitry). Because VREF0 and VREF1 are implemented using the Mocha-1 single-ended op-amp, it is possible to reduce power consumption by shutting down VREF0 and VREF1, when they are not required by the application.

6 Applications Information

6.1 Clocks Domains

The Mocha1 platform's Cortex™-M0 Hardened Subsystem contains architecturally gated clock domains to provide various operating modes. These operating modes allow software to be optimized for low-power operation of the TSX-1001 according to each application's requirements.

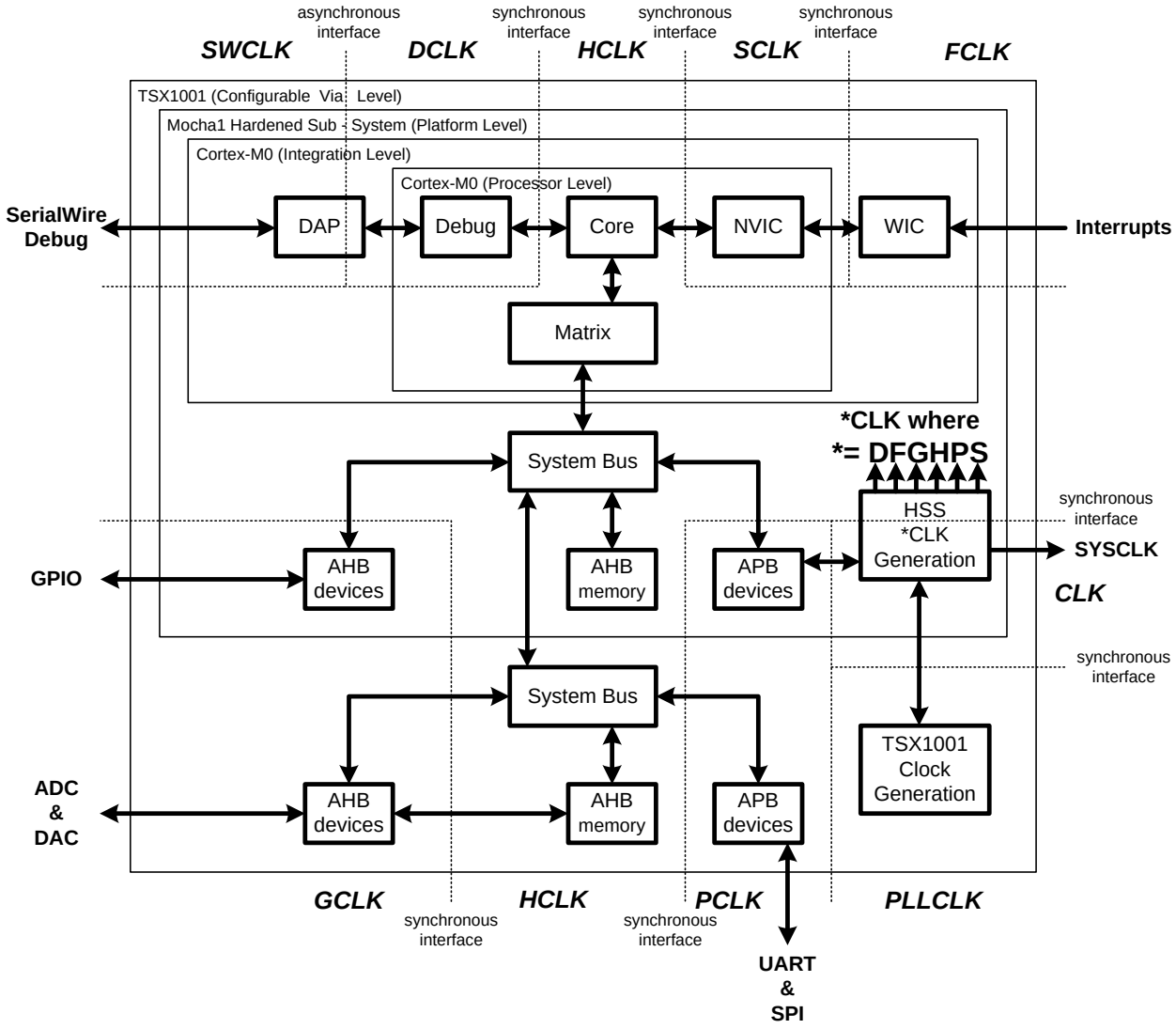


Figure 9: Clock domain block diagram

TSX-1001 has two primary input clock domains which are asynchronous: SWCLK and PLLCLK. SWCLK is input from an external SerialWire debug device and is used to clock the serial bidirectional data between the internal DAP (debug access port) and the external debugger. PLLCLK is the other primary input domain from which the rest of the clock domains are derived.

PLLCLK is generated from an on-chip integer PLL. When enabled, the PLL generates a clock 12x the crystal frequency. When disabled the PLL crystal reference is disabled and the PLL VCO runs at a slower frequency. The TSX-1001 Clock Generation Block divides the PLL output by either 128 or 2 to produce the CLK input to the hardened subsystem. To allow the PLL to lock during power-up, the TSX-1001 clock generation divider is initially set to 128. After the processor initially boots, software can initialize the system clock control bit, LOCKOK, to trigger automatic switching to divide-by-two mode once the PLL is locked. Subsequently when the device enters the lowest power mode (WICDEEPSLEEP mode), the system will reduce the TSX-1001 clock divider to divide-by-128 status, then automatically return to divide-by-two mode after a wakeup event.

The output clock from the TSX-1001 clock generation block is the primary clock input to the hardened subsystem's clock generation block (CLK domain). The hardened subsystem's clock generation block provides a controllable divider, with divide-by settings 1, 2, 4, 8, and 16, to produce the clock for the core processor. This divider is common to the remaining synchronous domains (FCLK, SCLK, HCLK, DCLK, GCLK, and PCLK domains) that are gated versions of the output of the hardened subsystem clock divider. They domains are used in both the hardened subsystem and throughout the TSX-1001's internal logic array. On power-up, the hardened subsystem divider is set to divide-by-16. The HSS clock generation block also provides a programmable divider for generating the APB (AMBA Peripheral Bus) clock. The APB divider has settings (1, 2, 3, and 4) to allow users to optimize power consumption and performance tradeoffs separately for APB devices.

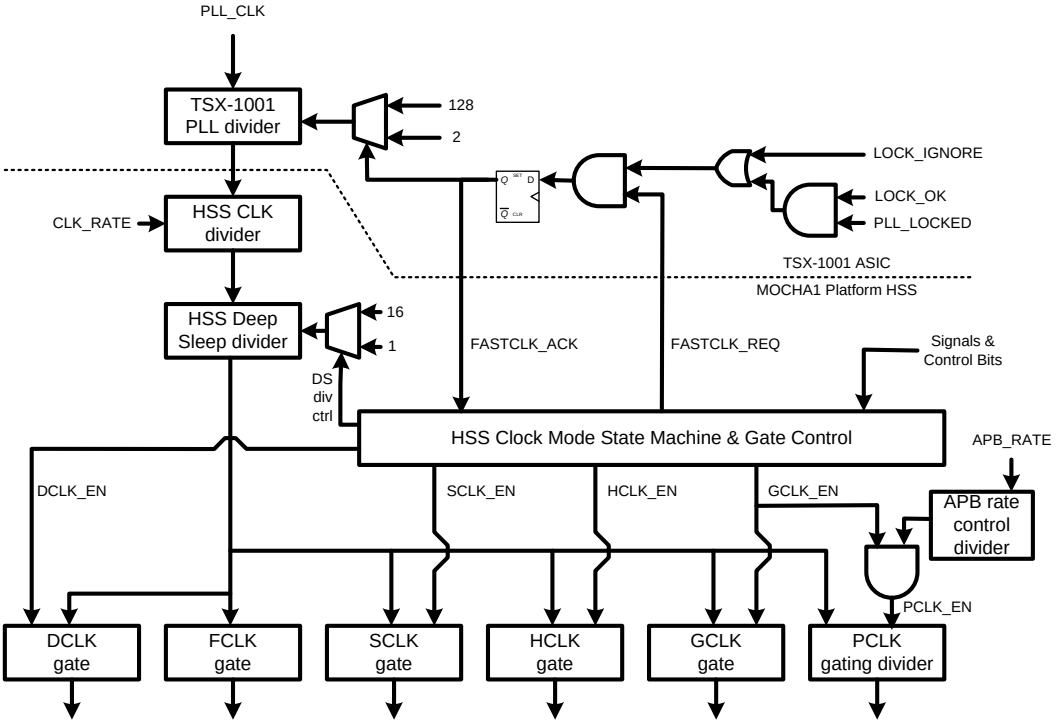


Figure 10: TSX1001 and MOCHA1 hardened subsystem clock generation

DCLK clocks the portion of the Cortex™-M0 which is used to communicate with the external debugger via the Debug Access Port (DAP). FCLK clocks a small amount of logic in the core for the Wakeup Interrupt Controller (WIC) to wake up the processor from its deepest sleep modes. SCLK is used by the processor core system tick timer and NVIC (Nested Vectored Interrupt Controller) to allow interrupts to wake up the processor from internal timer events and external interrupts. HCLK is the master clock for AHB (AMBA High-Performance Bus). Controlled by the KPA bit of **MOCHA1_CTRL1** (bit 21), GCLK is similar to HCLK, but can keep AHB devices awake while the processor sleeps. PCLK is the APB clock and also can be kept awake in certain sleep modes with the KPA bit. Within the hardened subsystem the GPIO uses GCLK, while the watchdog timer uses PCLK. In the TSX-1001, the timer, pulse-width modulator, and data converters use GCLK, while the UART and SPI blocks use PCLK.

Clock Domain	Description
SWCLK	SerialWire Debug clock asynchronous to core

Clock Domain	Description
DCLK	Debugger clock synchronous to core
FCLK	WIC clock
SCLK	System Tick and NVIC clock
HCLK	AHB peripheral clock without keep awake feature
GCLK	AHB peripheral clock with keep awake feature
PCLK	APB peripheral clock with keep awake feature
CLK	Primary clock input to hardened subsystem
PLLCLK	Primary clock input to digital section of the via configurable ASIC

Table 23: Clock domain summary

6.2 Operating Modes

As shown in **Note:** * pll runs when FASTCLK_REQ is true **Table 24**, the clock domains are enabled according to the device operating mode. Software and peripheral generated interrupts and events can trigger changes in operating modes according to a state machine within the Mocha1 platform hardened subsystem (see Figure 11).

Mode	FCLK	SCLK	HCLK	PCLK GCLK	DCLK	PLLCLK*	HSS DS div 16	TSX DS div 128
DEBUG	✓	✓	✓	✓	✓	✓		
RUN	✓	✓	✓	✓		✓		
SLEEP_PA	✓	✓		✓		✓		
DEEPSLEEP_PA	✓	✓		✓		✓	✓	
WICDEEPSLEEP_PA	✓			✓		✓	✓	
SLEEP	✓	✓				✓		
DEEPSLEEP	✓	✓				✓	✓	
WICDEEPSLEEP	✓						✓	✓
STOP								

Note: * pll runs when FASTCLK_REQ is true

Table 24: Hardened Subsystem clock modes for power savings

All clocks are on whenever the debugger is active to allow for debug access to the all sections of the device. After a Power-On Reset (POR), the TSX-1001 will enter into normal RUN mode with clock control bits set to divide by 16 (CLKRATE=[000]).

Other power modes exist that optimize power savings according to specific application requirements. The Cortex™-M0 SLEEP mode gates off the HCLK (the system bus clock). DEEPSLEEP mode is unique to the TSX-1001 HSS. It provides additional power savings by reducing the clock speed by a factor of 16. When the WIC is enabled prior to going into DEEPSLEEP, SCLK is gated for further power savings (WICDEEPSLEEP mode). Variations of SLEEP, DEEPSLEEP, and WICDEEPSLEEP allow peripherals to be turned on or off (the modes append _PA to their respective names). By leaving them on, the peripherals can operate normally while the processor sleeps, at the expense of additional power consumption. The designer is advised to turn off unused peripherals using respective enable bits accordingly, when using _PA modes.

The different power modes of the TSX-1001 are selected by altering specific bits in the **MOCHA1_CTRL1** register (see **Table 7**). The field WIC_ENREQ selects between WICDEEPSLEEP and normal

DEEPSLEEP mode. Bit KPA (MOCHA1_CTRL1[21]) allows the device peripherals to be optionally kept awake during the "_PA" sleep modes so that they can continue operating and potentially be used to wake the HSS Cortex™-M0 core.

The System Tick Timer (STT) and the NVIC are integrated in the Cortex™-M0 processor level and require SCLK to operate. Because the SCLK does not operate in DEEPSLEEP(_PA) and WICDEEPSLEEP(_PA), the STT is not available in these modes. Although the NVIC does not operate during these modes, the WIC module records pending wake-up interrupts. When the processor returns to RUN mode after sleeping, the NVIC can process the interrupts.

Though the Mocha-1 platform is fully capable of the functionality, the TSX-1001 does not implement a formal STOP mode. This mode can be handled outside the TSX-1001 using pass-FETs, in series with the IC's positive supply pins, and suitable GPIO signaling. As an alternative, the TSX-1001 provides a very low-power mode (WICDEEPSLEEP), where the crystal reference is disabled, allowing the PLL's VCO to run as a low-power on chip oscillator with additional clock frequency reduction provided by a divider in the TSX-1001 clock generation block. Future CVL designs on the Mocha-1 platform could provide a true on-chip STOP mode that would provide a wakeup mechanism to re-enable the HSS CLK (primary clock input to HSS) and then fully awaken the device from WICDEEPSLEEP mode. When FASTCLK_REQ (output from HSS) transitions low and STOPREQ (**MOCHA1_CTRL1** register) is high, the CVL design could gate-off the input HSS CLK signal (HSS clock input).

Table 25 lists the signals and register bits which define or affect the device operating modes, while Figure 11 shows how they cause transitions between these modes. ARM's Cortex™-M0 documentation has further details on power management events and modes and fully defines the Cortex™-M0 System Control Block registers including the System Control Register (SCR) (ARM DUI 0497A: "Cortex-M0 Devices: Generic User Guide", 2009).

Signal or Register Control Bit	S=Signal R=Register Bit	Description
DBG_CLK_REQ	S	DBG_CLK_REQ is generated at the Cortex™-M0 integration level when debugger is active.
WICENACK	S	WICENACK is generated by the WIC block from as acknowledgement of the WIC_ENREQ bit in the **MOCHA1_CTRL1** register.
GATEHCLK	S	GATEHCLK is asserted by the Cortex™-M0 integration level from sleep events: WFI (code hint) WFE (code hint) Returning from Interrupt Service Routine when SLEEPONEXIT in Cortex™-M0 **SCR**[1] is set. GATEHCLK is de-asserted by Cortex™-M0 due to wakeup events: Enabled interrupts All interrupts if SEVONPEND in **SCR**[4] is 1 Debug event or debugger active
KPA	R	KPA (Keep Peripherals Awake) bit in the **MOCHA1_CTRL1** register. 0 = Let Peripherals Sleep when processor sleeps

Signal or Register Control Bit	S=Signal R=Register Bit	Description
		1 = Keep Peripherals Awake
SLEEPDEEP	R	SLEEPDEEP is **SCR**[2] in the Cortex™-M0 0 = SLEEP mode 1 = DEEPSLEEP mode
SEVONPEND	R	SEVONPEND (sent event on pending bit) in **SCR**[4] in Cortex™-M0 affects GATEHCLK and WAKEUP signals 0 = Only enabled interrupts or events will wakeup 1 = Enabled events and all interrupts will wakeup
SLEEPONEXIT	R	SLEEPONEXIT in **SCR**[1] in Cortex™-M0 affects GATEHCLK signal 0 = Do not sleep when returning to thread mode from ISR (Interrupt Service Routine) 1 = Enter sleep or deep sleep or wic deep sleep on return from an ISR to thread mode Setting this bit to 1 enables an interrupt-driven application to avoid returning to an empty main application.
WAKEUP	S	WAKEUP is produced by WIC block for enabled WIC events.
FASTCLK_REQ	S	FASTCLK_REQ is asserted by HSS Clock Generation Block for the modes that expect the PLL-locked clock
FASTCLK_ACK	S	FASTCLK_ACK is asserted by the TSX-1001 Clock Generation Block to acknowledge to the HSS that the PLL clock output is available.

Table 25: Signals and control bits defining device operating mode.

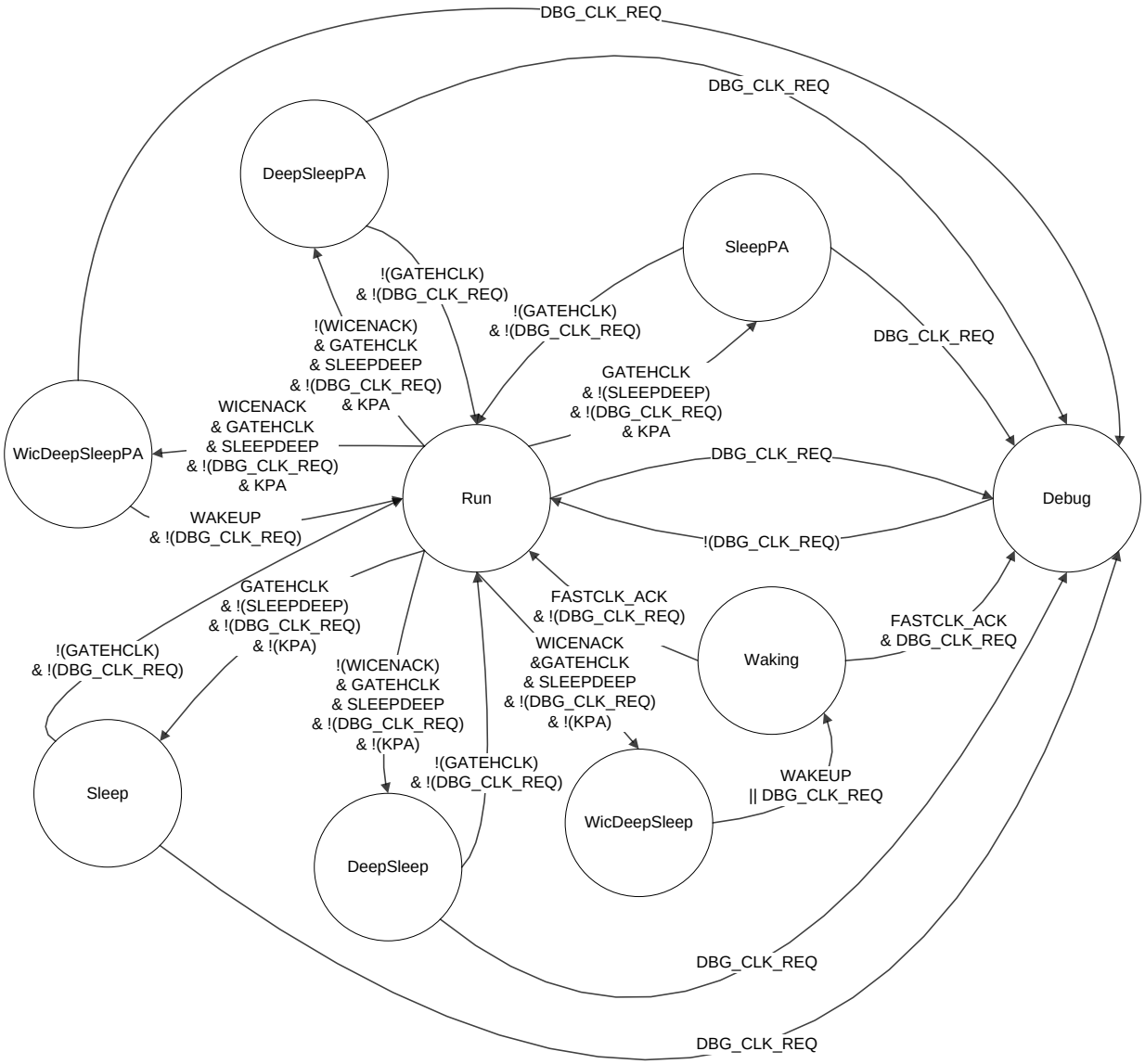


Figure 11: Operating mode state machine.

6.3 Interrupt Mapping

The TSX-1001 implements the 13 interrupts that are defined in Table 26. Three of these interrupts are generic and are available on the *Cortex*™-M0 by default: a non-maskable interrupt (NMI) that is derived from the Watchdog Timer circuit, and two maskable interrupts (Interrupts 0 and 1) that activate from activity occurring on the GPIO registers of the HSS (**GPIO0** and **GPIO1**, respectively). These interrupts follow ARM's "pulsed" format, where these interrupt lines are held high for at least one clock cycle of the FCLK, then are de-asserted. In this way, a single interrupt event cannot retrigger the same interrupt routine to occur twice.

Interrupts 2 through 7 are generated from peripherals specific to the TSX-1001. These interrupts follow ARM's "level-sensitive" format and must be cleared once detected to avoid accumulating additional interrupts of the same kind unnecessarily. Before the interrupt service routine has finished servicing the interrupt, it must write to the proper register in the corresponding peripheral address to clear the interrupt and prevent it from triggering additional interrupts.

Finally interrupts 8 through 11 indicate transitions occurring on the dedicated interrupt pins, /INT8 and /INT9. These interrupts are either "pulsed" or "level-sensitive", depending on the external implementation. Note: a "pulsed"-type interrupt would require external circuitry and access to the FCLK through the SYSCLOCK pin.

IRQ Assignments	
NMI	HSS Watchdog Timer non-maskable interrupt w.r.t. *Cortex™* M0's NVIC
IRQ[0]	HSS GPIO0 interrupt. Programmable/selectable IRQ sources within the peripheral. See Section "2.2.1.1 GPIO0 Register Bank" for details.
IRQ[1]	HSS GPIO1 interrupt. Programmable/selectable IRQ sources within the peripheral. See Section "2.2.1.2 GPIO1 Register" for details.
IRQ[2]	ADC1 interrupt
IRQ[3]	DAC1 FIFO interrupt
IRQ[4]	PWM1 interrupt
IRQ[5]	TMR1 interrupt
IRQ[6]	UART interrupt (programmable/selectable IRQ sources within the peripheral)
IRQ[7]	SPI interrupt (programmable/selectable IRQ sources within the peripheral)
IRQ[8]	External interrupt from /INT8 pin. Active low. (active low w.r.t. TSX1001 pin /INT8, inverted in VCA and retimed to HSS FCLK since *Cortex™* M0's NVIC inputs are active high) (may need masking of interrupt if active high signaling used)
IRQ[9]	External interrupt from /INT9 pin. Active low. (active low w.r.t. TSX1001 pin /INT9, inverted in VCA and retimed to HSS FCLK since *Cortex™* M0's NVIC inputs are active high) (may need masking of interrupt if active high signaling used)
IRQ[10]	From /INT8 external interrupt. Active high. (active high w.r.t. TSX1001 pin /INT8, retimed in VCA to HSS FCLK) (may need masking of interrupt if active low signaling used)
IRQ[11]	From /INT9 external interrupt (active high w.r.t. TSX1001 pin /INT9, retimed in VCA to HSS FCLK) (may need masking of interrupt if active low signaling used)
IRQ[15:12]	HSS IRQ inputs unused in TSX1001

Table 26: TSX-1001 interrupt assignments.

6.4 Cascaded System Rom Tables

TSX1001 has a 3 cascaded or linked System Rom Tables to indentify the device to external debugging tools following ARM CoreSite documentation: (CoreSite Architecture Specification) and (ARM DDI 0314H: "CoreSight Components Technical Reference Manual", 2009). Note that the Component ID of Rom Tables = 0xB105100D, and of Core Memory Components = 0xB105E00D

Register	CVL Rom Table Values[1] [2]	Mocha1 Rom Table Values[1]	Cortex-M0 Rom Table Values[1]	SCS Core Memory Component	DWT Core Memory Component	BPU Core Memory Component
Peripheral ID4	0x00000006	0x00000006	0x00000004	0x00000004	0x00000004	0x00000004
Peripheral ID3	0x00000030	0x00000010	0x00000000	0x00000000	0x00000000	0x00000000
Peripheral ID2	0x0000000F	0x0000000F	0x0000000B	0x0000000B	0x0000000B	0x0000000B
Peripheral ID1	0x000000E0	0x000000E0	0x000000B4	0x000000B0	0x000000B0	0x000000B0
Peripheral ID0	0x00000000	0x00000001	0x00000071	0x00000008	0x0000000A	0x0000000B
Component ID3	0x000000B1	0x000000B1	0x000000B1	0x000000B1	0x000000B1	0x000000B1
Component ID2	0x00000005	0x00000005	0x00000005	0x00000005	0x00000005	0x00000005
Component ID1	0x00000010	0x00000010	0x00000010	0x000000E0	0x000000E0	0x000000E0
Component ID0	0x0000000D	0x0000000D	0x0000000D	0x0000000D	0x0000000D	0x0000000D
Parsed Identification Information						
JEP106 identity code (7bits)	0x7E	0x7E	0x3B	0x3B	0x3B	0x3B
JEP106 continuation code (4 bits)	0x6	0x6	0x4	0x4	0x4	0x4
Part number (12 bits)	0x000	0x001[3]	0x471	0x008	0x00A	0x00B
Revision (4 bits)	0x0	0x0	0x0	0x0	0x0	0x0
Revand (4 bits)	0x3	0x1	0x0	0x0	0x0	0x0
Customer modified (4 bits)	0x0	0x0	0x0	0x0	0x0	0x0
Table Entries	Mocha 1 Rom Table	Cortex-M0 Rom Table[3]	SCS[4]			
	TABLE END	TABLE END	DWT[5]			
			BPU[6]			
			TABLE END			

Note 1: The Cortex-M0 ROM table only supports word size transactions.

Note 2: Future CVLs on Mocha 1 can use customer's JEP106 ID and their choice of part number or Triad Semiconductor's JEP ID 0x7E / 0x6 and a part number assigned by Triad. The CVL Rom Table contains a single entry pointing to the Mocha 1 Rom Table.

Note 3: Mocha Rom Table Part number 0x001 indicates Mocha 1 platform. If debugger access has been blocked with EEPROM code security word then presence indicator of single entry (link to) Cortex-M0 Rom Table will not be indicated and Access to Debugger will be disabled, also any access to memory will be blocked except for:

CVL and Mocha1 Rom Tables for identification purposes and
code security word to trigger a complete erasure of the EEPROM.

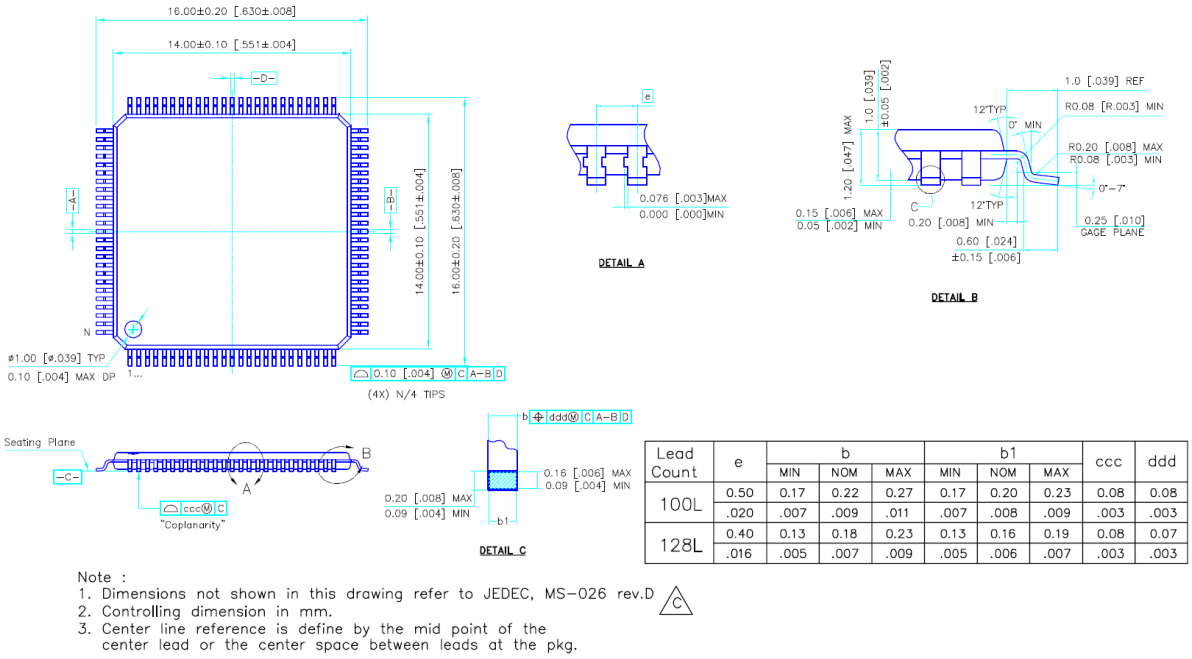
Note 4: ARM - v6M System Control Space.

Note 5: ARM - v6M Data Watchpoint and Trace.

Note 6: ARM - v6M Breakpoint Unit.

7 Packaging

The TSX-1001 is available in a 100-lead quad flat pack (QFP) package.



Lead Count	e	b			b1			ccc	ddd
		MIN	NOM	MAX	MIN	NOM	MAX		
100L	0.50	0.17	0.22	0.27	0.17	0.20	0.23	0.08	0.08
	.020	.007	.009	.011	.007	.008	.009	.003	.003
128L	0.40	0.13	0.18	0.23	0.13	0.16	0.19	0.08	0.07
	.016	.005	.007	.009	.005	.006	.007	.003	.003

Note :
1. Dimensions not shown in this drawing refer to JEDEC, MS-026 rev.D
2. Controlling dimension in mm.
3. Center line reference is define by the mid point of the center lead or the center space between leads at the pkg.

Figure 12: 100-Lead QFP detailed package drawing

8 References

"PiP-EC02: Universal Asynchronous Receiver/Transmitter: AMBA Compatible". (n.d.). Retrieved from Soc Solutions, LLC.

ARM DDI 0314H: "CoreSight Components Technical Reference Manual". (2009). Retrieved October 7, 2011, from http://infocenter.arm.com/help/topic/com.arm.doc.ddi0314h/DDI0314H_coresight_components_trm.pdf

ARM DDIO432A: "Cortex-M0 Technical Reference Manual (rev.:r0p0)". (2009, November 30). Retrieved from ARM, Ltd.: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0497a/index.html

ARM DUI 0497A: "Cortex-M0 Devices: Generic User Guide". (2009, November 21). Retrieved from ARM, Ltd.: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0497a/index.html

CoreSite Architecture Specification. (n.d.). Retrieved from ARM, Ltd.: https://silver.arm.com/download/Systems_IP/Debug_&_Trace/TM093-DC-70005-r0p1-00rel0/CoreSight_Architecture_Specification.pdf

IPC-SPI-APB v2.0: Serial Peripheral Interface (SPI) Controller". (n.d.). Retrieved from Soc Solutions, LLC.

END OF DOCUMENT