RUMPS401 SYSTEM SPECIFICATION
V 1.7

Change log:

Version	Date	Modifier	Description
1.0	2014.09.29	Dicky	Initial specification contains initial block diagrams and IO pin configuration
1.1	2014.10.31	Dicky	Update IO pin configuration: • Add OSC pins, remove CLK pin • Add Flash analog pins (Vpp, Vref, TM0) • Reformed the pins placement to set the flash test interface aligned in two parallel sides
1.2	2014.11.07	Dicky	Update IO pin configuration: • Add definition of pins alternate functions • Define the pins configuration that are bounded to the chip package (targeted for 80 pins package)
1.3	2014.11.16	Dicky	Reformed Specification Document Layout
1.4	2014.12.7	Dicky	Modify address map of the cores, convert one of the normal core to DSP core, add NoC section, update IO-Control core's GPIO spec, modify IO-Control core's mux ctrl, specify target package, add internal LDO, extend AHB peripheral address range (4k -> 64k)
1.5	2015.3.20	Dicky	Change the scheme of the watch dog timer, add watch dog timer section, add more information to timer and SPI sections, differentiate between the IO pads layout and IO pins package diagram, change the IO configuration, change the address mapping of the cores, update the diagram of the cores, add another normal mode with different remap address location for the IO-Control core, add information regarding the remap mechanism to section 4.1, reduce the number of GPIO interrupt of the IO-Control core to accommodate WDT interrupt, add information to MAC section, add more information to the hardware bootloading section, add more information in reset ctrl section.
1.6	2015.4.13	Dicky	Change the IO-Control core to IO-control core, add design flow and tools version sections, change the main architecture diagram, change system_ctrl diagram, add 32 kHz RC and power management section, add chip size section, add more information in GPIO sections, modify IO pins and pads sections, add register to AHB2NOC bridge test, modify GPIO section, add clock diagram, add exposed mode, change IO configuration at the north side, change the number of ss in SPI section, switch VPP and VSSP

			pins, switch XIN and XOUT pins.
1.7	2015.9.11	Dicky	Change pin package numbering

Table of Content

1 Document Information

1.1 Scope

This document contains the description of the RUMPS401 chip, the UTAR VLSI own Network-on-Chip-based Multiprocessor System-on-Chip design specifications.

1.2 Bibliography

- AMBA3 AHB-Lite Protocol v1.0
- AMBA3 APB Protocol v1.0
- Silterra Malaysia Sdn. Bhd. 0.18μm Process 1.8-Volt Low-Power SAGE-X^{TM} Standard Cell Library Databook (Provided by ARM)
- Artisan Standard Library 130nm-250nm SRAM Generator User Manual (Provided by ARM)
- ESF1-180_SST90WF2561_V12_MRD 180 SuperFlash 256K Bits IP (Provided by SST)

1.3 Abbreviations and Terminology

Symbol	**Definition**
AMBA	Advanced Microcontroller Bus Architecture
AHBA AHB	AMBA Advanced High-performance Bus
AMBA APB	AMBA Advanced Peripheral Bus
AIO	Analog input/output
BIST	Built-In Self-Test
DI, DIO, DO	Digital Input, Digital Input/Output, Digital Output
DFT	Design For Test
GPIO	General Purpose Input/Output
IP	Intellectual Property
LSB	Least Significant Bit
MSB	Most Significant Bit
NVM	Non Volatile Memory
POR	Power On Reset
SoC	The System On a Chip device
SRAM	Static Random-Access Memory
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
LDO	Low-Dropout Regulator
POR	Power On Reset
MAC	Multiply-Accumulate
NOC	Network-On-Chip
NMI	Non-Mask-able Interrupt
MAC	Multiply-Accumulate

2 Overview

2.1 Features

General:

- Operating voltage range:
  - IO: 2.97 V – 3.63 V.
  - Core: 1.62 V – 1.98 V.
- Internal 3.3V to 1.8V LDO to supply power for the core cells.
- Internal PoR circuit to generate PoR sequence.
- 16 MHz crystal oscillator circuit for normal operation.
- 32 kHz RC oscillator for sleep mode.
- Power management with clock-gating.
- Package: 100 pins QFP.

Processor Related:

- Four ARM Cortex-M0 processors. The cores are categorized into one IO-control core, one DSP core and two normal cores.
- Four local 32 Kbytes SST SuperFlash for program storage.
- Four local 8 Kbytes ARM low power SRAM.
- Peripheral interrupts and Wake-up For Interrupt mechanism.
- UTAR adaptive NoC inter-processor communication network.
- Software-controlled debug mechanism.
- Support for Hardware and Software bootloading mechanisms.
- Two AES accelerator modules local to the normal cores.
- Multiply-Accumulate (MAC) accelerator modules local to the DSP core.
- Four local Watch dog timers that able to independently reset the cores.

IO Peripherals:

- 32-bit GPIO for the IO-control core, and three 8-bit GPIOs for the DSP core and normal cores.
- Four timers, one for each core, that support capture and compare mechanism.
- Four SPIs, one for each core.
- 32-bit Parallel port controller for the IO-control core.

2.2 Chip Size

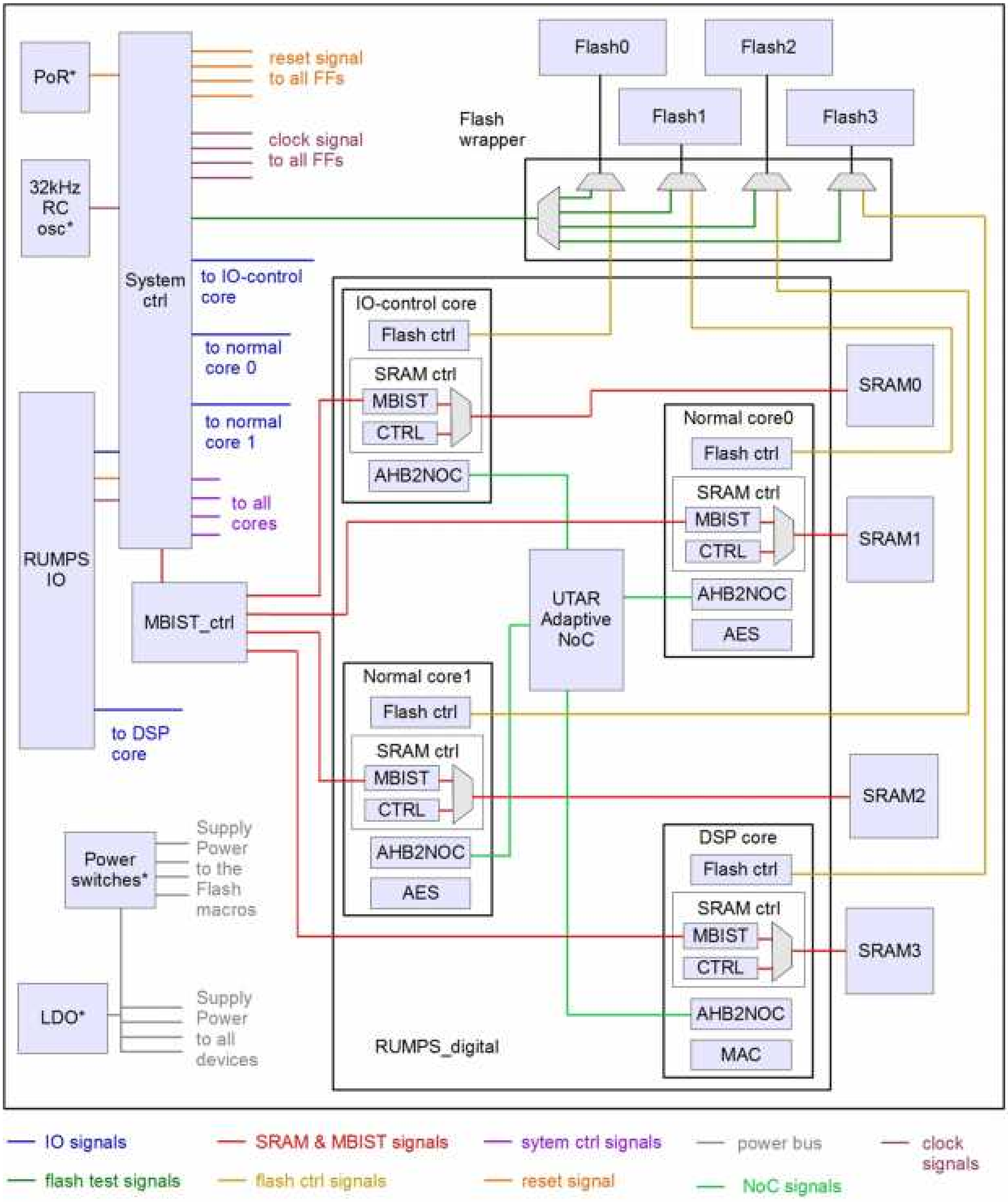
- The standard cell equivalent gate count:
  750 k gates
- The total chip equivalent gate count:
  1220 k gates
- The step and repeat size:
  Width: 5240.50 um
  Height: 5240.62 um

2.3 Packaging

The chip is designed for 100 pins QFP package.

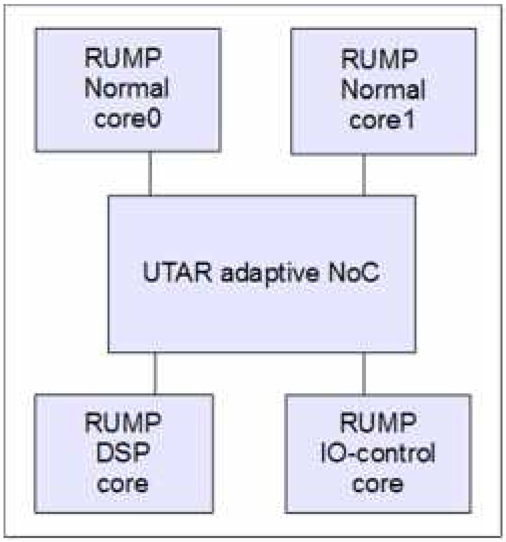
2.4 Architecture

2.4.1 RUMPS Chip

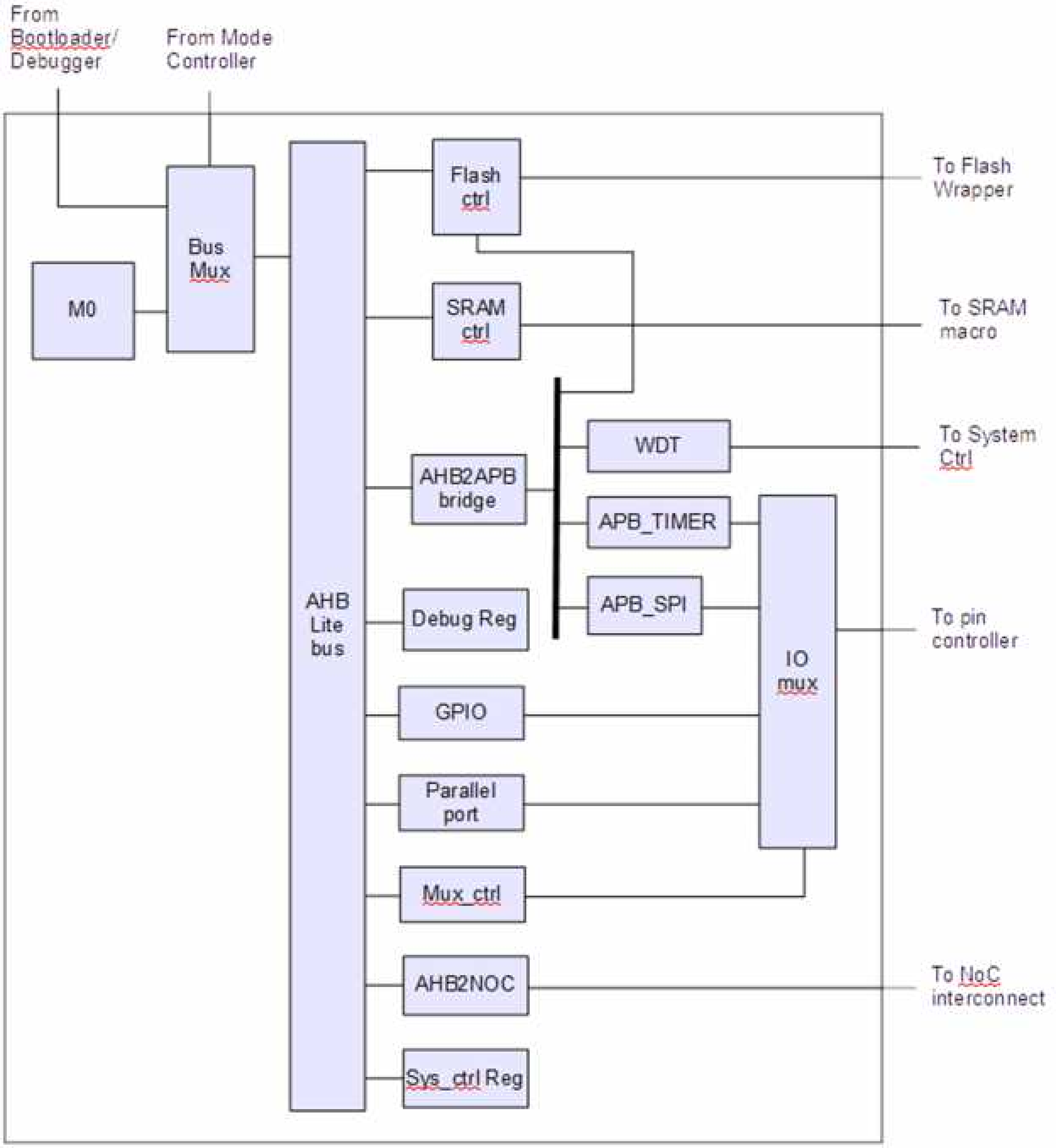


* the PoR, LDO, 32kHz RC oscillator and power switches are the IP courtesy of SZGC

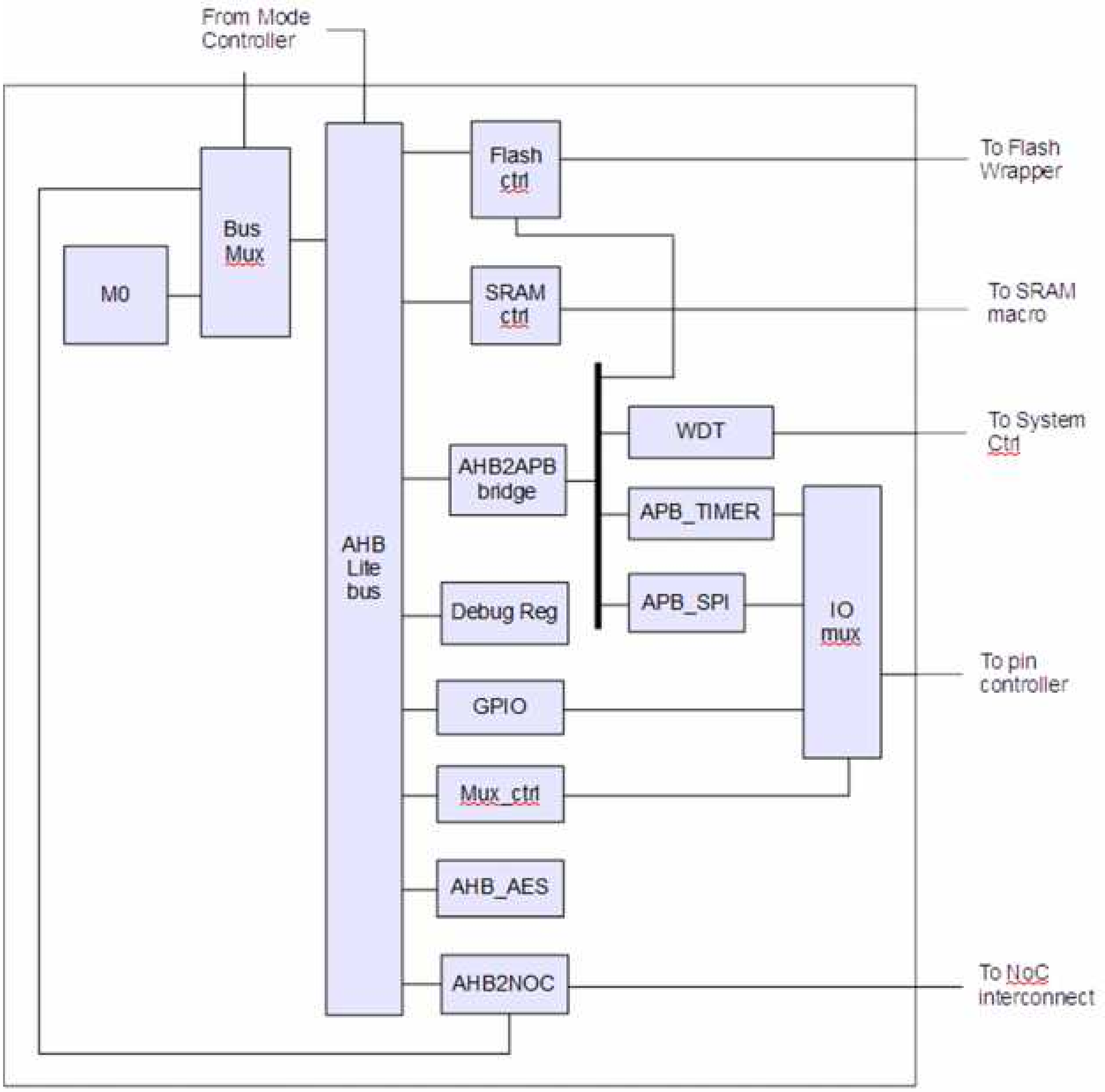
2.4.2 RUMPS Digital



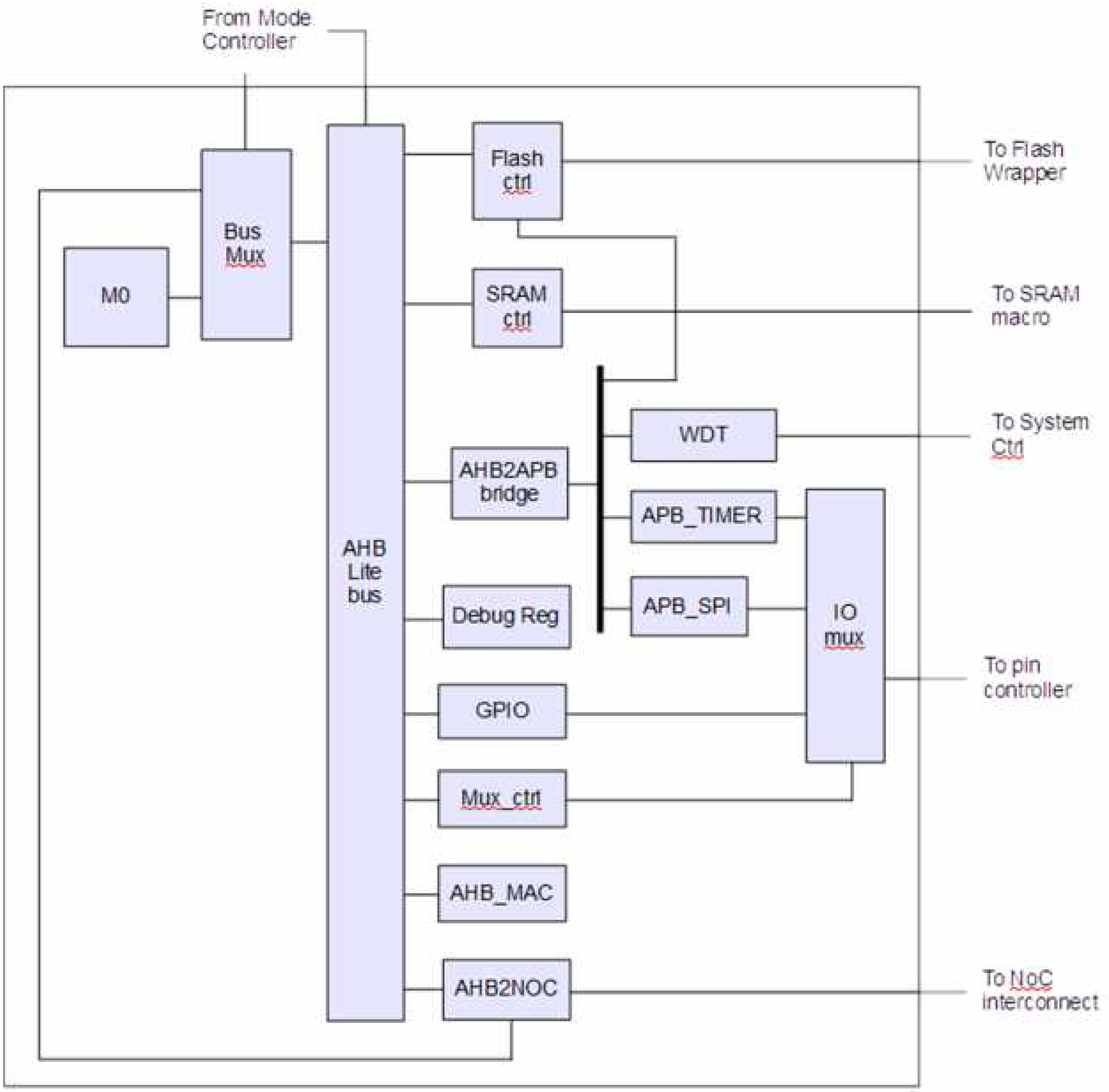
2.4.3 IO-control Core



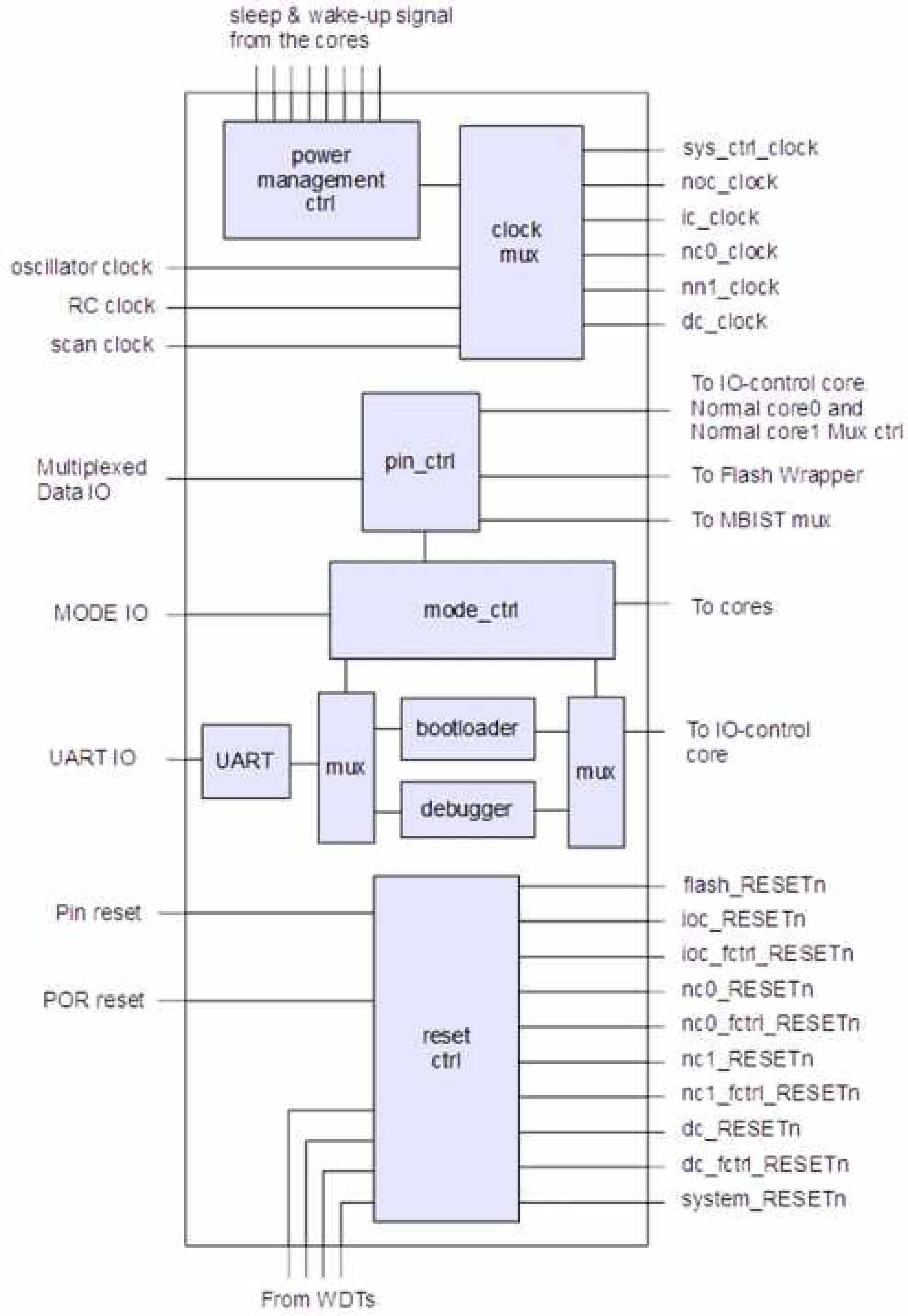
2.4.4 Normal Core



2.4.5 DSP Core

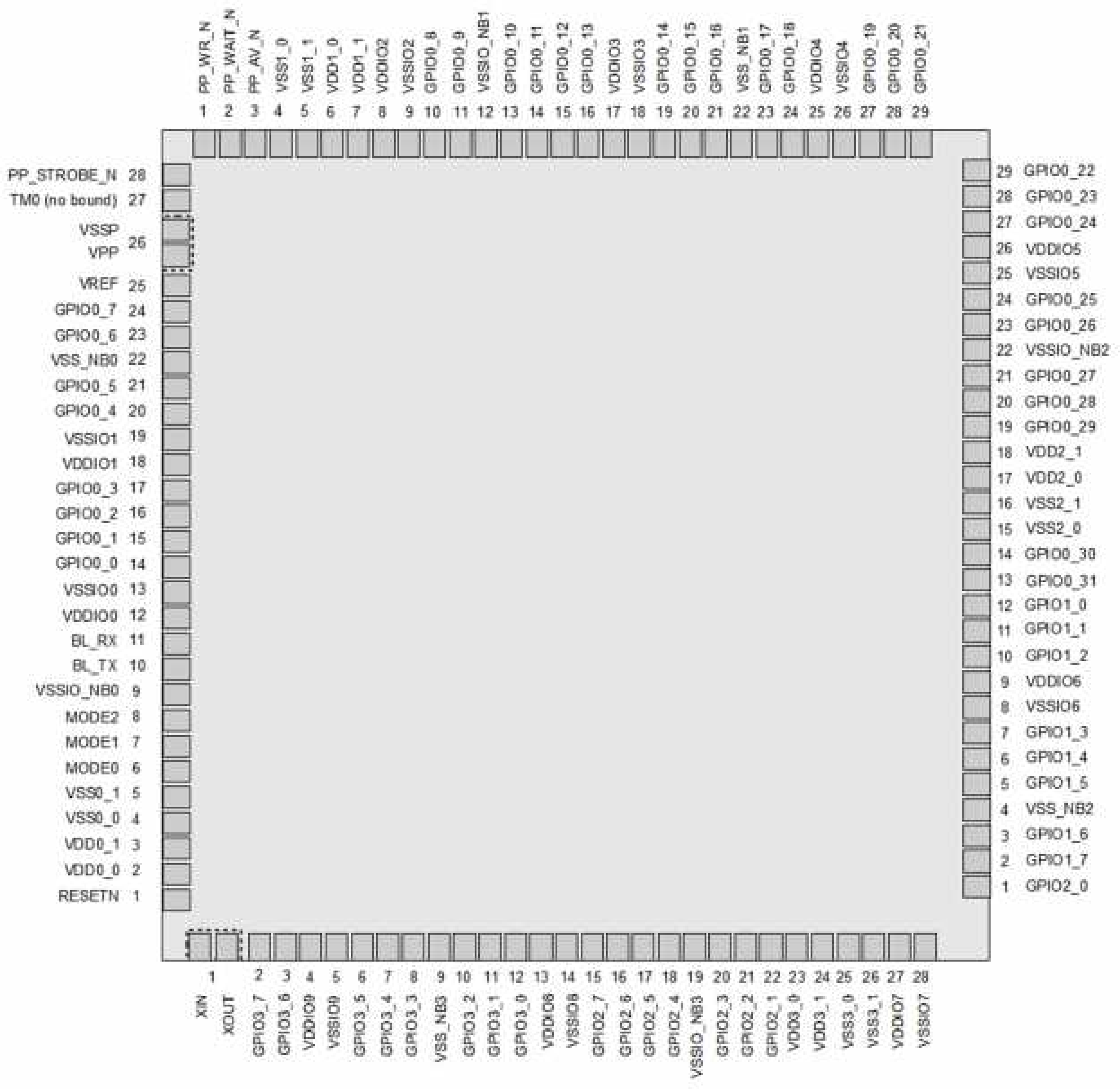

From Mode Controller
M0
Bus Mux
AHB Lite bus
Flash ctrl
SRAM ctrl
AHB2APB bridge
WDT
APB_TIMER
APB_SPI
IO mux
Debug Reg
GPIO
Mux_ctrl
AHB_MAC
AHB2NOC
To Flash Wrapper
To SRAM macro
To System Ctrl
To pin controller
To NoC interconnect

2.4.6 System Controller



2.5 IO

2.5.1 IO Pads Layout Diagram



2.5.2 IO Pads Description

The PVDD and PVSS pads are selected to be used for the 1.8 V power-ground pairs (VDDn and VSSn), the PDVDD and PDVSS pads are for the 3.3 V power-ground pairs (VDDIOn and VSSIOn). The 3.3 V power-ground supplies are only connected to the IO power rings and the flash macros. The 1.8 V power-ground supplies are connected to the IO power rings and the core power rings. There are pairs of VDDn and VSSn pads at each side of the chip. One of the VDDn and VSSn is used to be connected to the core power ring, while the others are dedicated for the flash macro’s 1.8V supply.

There are four VSSIO_NBn (PDVSS) and four VSS_NBn (PVSS) pads that are bounded to the paddle. In addition, the VSSIO6 is also bounded to the paddle. The paddle is bounded to two VSSIOn package pins (refer to section 2.4.4). The

Custom design pads are used for the VREF, VPP, VSSP and TM0. The pads are analog pads with digital power rings. The pads are created to avoid the usage of break cells hence the chip can implement full IO power rings. TM0 pad is not bounded out.

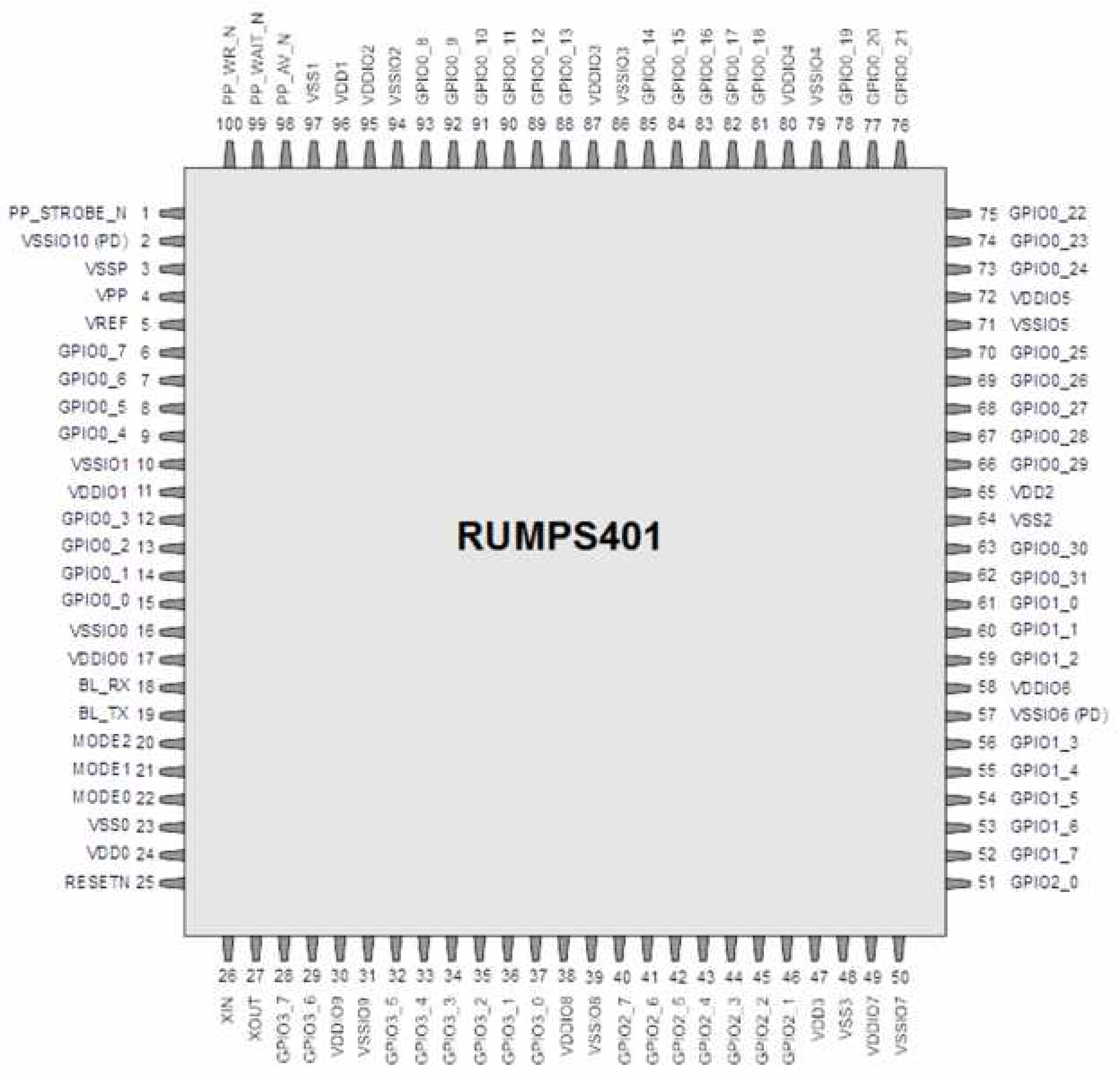
no	side	IO name	IO buffer name	description
1	W	RESETN	PIC	Reset pad
2	W	VDD0_0	PVDD	1.8 V supply
3	W	VDD0_1	PVDD	1.8 V supply
4	W	VSS0_0	PVSS	GND for 1.8 V supply
5	W	VSS0_1	PVSS	GND for 1.8 V supply
6	W	MODE0	PICD	Mode input pad
7	W	MODE1	PICD	Mode input pad
8	W	MODE2	PICD	Mode input pad
9	W	VSSIO_NB0	PDVSS	GND for 3.3 V supply (bound to paddle)
10	W	BL_TX	POC2A	bootloader UART TX pad
11	W	BL_RX	PICU	bootloader UART RX pad
12	W	VDDIO0	PDVDD	3.3 V supply
13	W	VSSIO0	PDVSS	GND for 3.3 V supply
14	W	GPIO0_0	PBCCTU2A	IO-Control Core GPIO
15	W	GPIO0_1	PBCCTU2A	IO-Control Core GPIO
16	W	GPIO0_2	PBCCTU2A	IO-Control Core GPIO
17	W	GPIO0_3	PBCCTU2A	IO-Control Core GPIO
18	W	VDDIO1	PDVDD	3.3 V supply
19	W	VSSIO1	PDVSS	GND for 3.3 V supply
20	W	GPIO0_4	PBCCTU2A	IO-Control Core GPIO
21	W	GPIO0_5	PBCCTU2A	IO-Control Core GPIO
22	W	VSS_NB0	PVSS	GND for 1.8 V supply (bound to paddle)
23	W	GPIO0_6	PBCCTU2A	IO-Control Core GPIO
24	W	GPIO0_7	PBCCTU2A	IO-Control Core GPIO
25	W	VREF	PANALOG_DIG	Flash VREF (custom pad)
26	W	VPP	PANSST_VPP	Flash VPP (custom pad)
	W	VSSP	PANSST_VPP	VSS for the VPP (custom pad)
27	W	TM0	PANALOG_DIG	Flash TM0 (custom pad), not bounded
28	W	PP_STROBE_N	PICU	Parallel port strobe_n input pad

1	N	PP_WR_N	PICU	Parallel port wr_n input pad
2	N	PP_WAIT_N	POC2A	Parellel port wait_n output pad
3	N	PP_AV_N	POC2A	Parellel port av_n output pad
4	N	VSS1_0	PVSS	GND for 1.8 V supply
5	N	VSS1_1	PVSS	GND for 1.8 V supply
6	N	VDD1_0	PVDD	1.8 V supply
7	N	VDD1_1	PVDD	1.8 V supply
8	N	VDDIO2	PDVDD	3.3 V supply
9	N	VSSIO2	PDVSS	GND for 3.3 V supply
10	N	GPIO0_8	PBCCTU2A	IO-Control Core GPIO
11	N	GPIO0_9	PBCCTU2A	IO-Control Core GPIO
12	N	VSSIO_NB1	PDVSS	GND for 3.3 V supply (bound to paddle)
13	N	GPIO0_10	PBCCTU2A	IO-Control Core GPIO
14	N	GPIO0_11	PBCCTU2A	IO-Control Core GPIO
15	N	GPIO0_12	PBCCTU2A	IO-Control Core GPIO
16	N	GPIO0_13	PBCCTU2A	IO-Control Core GPIO
17	N	VDDIO3	PDVDD	3.3 V supply
18	N	VSSIO3	PDVSS	GND for 3.3 V supply
19	N	GPIO0_14	PBCCTU2A	IO-Control Core GPIO
20	N	GPIO0_15	PBCCTU2A	IO-Control Core GPIO
21	N	GPIO0_16	PBCCTU2A	IO-Control Core GPIO
22	N	VSS_NB1	PVSS	GND for 1.8 V supply (bound to paddle)
23	N	GPIO0_17	PBCCTU2A	IO-Control Core GPIO
24	N	GPIO0_18	PBCCTU2A	IO-Control Core GPIO
25	N	VDDIO4	PDVDD	3.3 V supply
26	N	VSSIO4	PDVSS	GND for 3.3 V supply
27	N	GPIO0_19	PBCCTU2A	IO-Control Core GPIO
28	N	GPIO0_20	PBCCTU2A	IO-Control Core GPIO
29	N	GPIO0_21	PBCCTU2A	IO-Control Core GPIO
29	E	GPIO0_22	PBCCTU2A	IO-Control Core GPIO
28	E	GPIO0_23	PBCCTU2A	IO-Control Core GPIO
27	E	GPIO0_24	PBCCTU2A	IO-Control Core GPIO
26	E	VDDIO5	PDVDD	3.3 V supply
25	E	VSSIO5	PDVSS	GND for 3.3 V supply
24	E	GPIO0_25	PBCCTU2A	IO-Control Core GPIO
23	E	GPIO0_26	PBCCTU2A	IO-Control Core GPIO
22	E	VSSIO_NB2	PDVSS	GND for 3.3 V supply (bound to paddle)
21	E	GPIO0_27	PBCCTU2A	IO-Control Core GPIO
20	E	GPIO0_28	PBCCTU2A	IO-Control Core GPIO
19	E	GPIO0_29	PBCCTU2A	IO-Control Core GPIO
18	E	VDD2_1	PVDD	1.8 V supply

17	E	VDD2_0	PVDD	1.8 V supply
16	E	VSS2_1	PVSS	GND for 1.8 V supply
15	E	VSS2_0	PVSS	GND for 1.8 V supply
14	E	GPIO0_30	PBCCTU2A	IO-Control Core GPIO
13	E	GPIO0_31	PBCCTU2A	IO-Control Core GPIO
12	E	GPIO1_0	PBCCTU2A	Normal Core0 GPIO
11	E	GPIO1_1	PBCCTU2A	Normal Core0 GPIO
10	E	GPIO1_2	PBCCTU2A	Normal Core0 GPIO
9	E	VDDIO6	PDVDD	3.3 V supply
8	E	VSSIO6	PDVSS	GND for 3.3 V supply (bound to paddle)
7	E	GPIO1_3	PBCCTU2A	Normal Core0 GPIO
6	E	GPIO1_4	PBCCTU2A	Normal Core0 GPIO
5	E	GPIO1_5	PBCCTU2A	Normal Core0 GPIO
4	E	VSS_NB2	PVSS	GND for 1.8 V supply (bound to paddle)
3	E	GPIO1_6	PBCCTU2A	Normal Core0 GPIO
2	E	GPIO1_7	PBCCTU2A	Normal Core0 GPIO
1	E	GPIO2_0	PBCCTU2A	Normal Core1 GPIO
28	S	VSSIO7	PDVSS	GND for 3.3 V supply
27	S	VDDIO7	PDVDD	3.3 V supply
26	S	VSS3_1	PVSS	GND for 1.8 V supply
25	S	VSS3_0	PVSS	GND for 1.8 V supply
24	S	VDD3_1	PVDD	1.8 V supply
23	S	VDD3_0	PVDD	1.8 V supply
22	S	GPIO2_1	PBCCTU2A	Normal Core1 GPIO
21	S	GPIO2_2	PBCCTU2A	Normal Core1 GPIO
20	S	GPIO2_3	PBCCTU2A	Normal Core1 GPIO
19	S	VSSIO_NB3	PDVSS	GND for 3.3 V supply (bound to paddle)
18	S	GPIO2_4	PBCCTU2A	Normal Core1 GPIO
17	S	GPIO2_5	PBCCTU2A	Normal Core1 GPIO
16	S	GPIO2_6	PBCCTU2A	Normal Core1 GPIO
15	S	GPIO2_7	PBCCTU2A	Normal Core1 GPIO
14	S	VSSIO8	PDVSS	GND for 3.3 V supply
13	S	VDDIO8	PDVDD	3.3 V supply
12	S	GPIO3_0	PBCCTU2A	DSP Core GPIO
11	S	GPIO3_1	PBCCTU2A	DSP Core GPIO
10	S	GPIO3_2	PBCCTU2A	DSP Core GPIO
9	S	VSS_NB3	PVSS	GND for 1.8 V supply (bound to paddle)
8	S	GPIO3_3	PBCCTU2A	DSP Core GPIO
7	S	GPIO3_4	PBCCTU2A	DSP Core GPIO
6	S	GPIO3_5	PBCCTU2A	DSP Core GPIO
5	S	VSSIO9	PDVSS	GND for 3.3 V supply

4	S	VDDIO9	PDVDD	3.3 V supply
3	S	GPIO3_6	PBCCTU2A	DSP Core GPIO
2	S	GPIO3_7	PBCCTU2A	DSP Core GPIO
1	S	XOUT	POSC4	Crystal in
	S	XIN	POSC4	Crystal out

2.5.3 IO Pins Package Diagram



2.5.4 IO Pins Description

There are in total 100 pins. Most of the pins are bounded to the corresponding IO pads. The VSSIO6 and VSSIO10 pins are bounded to the paddle to provide ground connection for the pads that are bounded to the paddle (refer to section 2.3.2).

pin no	name	type	side	description	alternate	alt type	alternate description
1	PP_STROBE_N	DI	W	Parallel port strobe_n input pin			
2	VSS10 (PD)	G	W	GND to be bounded to the paddle			
3	VSSP	A	W	Flash VPP			
4	VPP	A	W	VSS for the VPP			
5	VREF	A	W	Flash VREF			
6	GPIO0_7	DIO	W	IO-Control Core GPIO	PP_DATA7	DIO	Parallel port data line
7	GPIO0_6	DIO	W	IO-Control Core GPIO	PP_DATA6	DIO	Parallel port data line
8	GPIO0_5	DIO	W	IO-Control Core GPIO	PP_DATA5	DIO	Parallel port data line
9	GPIO0_4	DIO	W	IO-Control Core GPIO	PP_DATA4	DIO	Parallel port data line
10	VSSIO1	G	W	GND for 3.3 V supply			
11	VDDIO1	P	W	3.3 V supply			
12	GPIO0_3	DIO	W	IO-Control Core GPIO	PP_DATA3	DIO	Parallel port data line
13	GPIO0_2	DIO	W	IO-Control Core GPIO	PP_DATA2	DIO	Parallel port data line
14	GPIO0_1	DIO	W	IO-Control Core GPIO	PP_DATA1	DIO	Parallel port data line
15	GPIO0_0	DIO	W	IO-Control Core GPIO	PP_DATA0	DIO	Parallel port data line
16	VSSIO0	G	W	GND for 3.3 V supply			
17	VDDIO0	P	W	3.3 V supply			
18	BL_RX	DI	W	bootloader UART RX pin			
19	BL_TX	DO	W	bootloader UART TX pin			
20	MODE2	DI	W	Mode input pin			
21	MODE1	DI	W	Mode input pin			
22	MODE0	DI	W	Mode input pin			
23	VSS0	G	W	GND for 1.8 V supply			
24	VDD0	P	W	1.8 V supply			
25	RESETN	R	W	Reset pin			
26	XIN	C	S	Crystal in			
27	XOUT	C	S	Crystal out			
28	GPIO3_7	DIO	S	DSP Core GPIO	MOSI3	DO	DSP core SPI MOSI
29	GPIO3_6	DIO	S	DSP Core GPIO	SCLK3	DO	DSP core SPI clock
30	VDDIO9	P	S	3.3 V supply			
31	VSSIO9	G	S	GND for 3.3 V supply			
32	GPIO3_5	DIO	S	DSP Core GPIO	SS3	DO	DSP core SPI slave select
33	GPIO3_4	DIO	S	DSP Core GPIO	MISO3	DI	DSP core SPI MISO
34	GPIO3_3	DIO	S	DSP Core GPIO	TM3_COM1	DO	DSP core Timer Compare
35	GPIO3_2	DIO	S	DSP Core GPIO	TM3_COM0	DO	DSP core Timer Compare
36	GPIO3_1	DIO	S	DSP Core GPIO	TM3_CAP1	DI	DSP core Timer Capture
37	GPIO3_0	DIO	S	DSP Core GPIO	TM3_CAP0	DI	DSP core Timer Capture
38	VDDIO8	P	S	3.3 V supply			

39	VSSIO8	G	S	GND for 3.3 V supply			
40	GPIO2_7	DIO	S	Normal Core1 GPIO	MOSI2	DO	normal core 1 SPI MOSI
41	GPIO2_6	DIO	S	Normal Core1 GPIO	SCLK2	DO	normal core 1 SPI clock
42	GPIO2_5	DIO	S	Normal Core1 GPIO	SS2	DO	normal core 1 SPI slave select
43	GPIO2_4	DIO	S	Normal Core1 GPIO	MISO2	DI	normal core 1 SPI MISO
44	GPIO2_3	DIO	S	Normal Core1 GPIO	TM2_COM1	DO	normal core 1 Timer Compare
45	GPIO2_2	DIO	S	Normal Core1 GPIO	TM2_COM0	DO	normal core 1 Timer Compare
46	GPIO2_1	DIO	S	Normal Core1 GPIO	TM2_CAP1	DI	normal core 1 Timer Capture
47	VDD3	P	S	1.8 V supply			
48	VSS3	G	S	GND for 1.8 V supply			
49	VDDIO7	P	S	3.3 V supply			
50	VSSIO7	G	S	GND for 3.3 V supply			
51	GPIO2_0	DIO	E	Normal Core1 GPIO	TM2_CAP0	DI	normal core 1 Timer Capture
52	GPIO1_7	DIO	E	Normal Core0 GPIO	MOSI1	DO	normal core 0 SPI MOSI
53	GPIO1_6	DIO	E	Normal Core0 GPIO	SCLK1	DO	normal core 0 SPI clock
54	GPIO1_5	DIO	E	Normal Core0 GPIO	SS1	DO	normal core 0 SPI slave select
55	GPIO1_4	DIO	E	Normal Core0 GPIO	MISO1	DI	normal core 0 SPI MISO
56	GPIO1_3	DIO	E	Normal Core0 GPIO	TM1_COM1	DO	normal core 0 Timer Compare
57	VSSIO6	G	E	GND to be bounded to the paddle			
58	VDDIO6	P	E	3.3 V supply			
59	GPIO1_2	DIO	E	Normal Core0 GPIO	TM1_COM0	DO	normal core 0 Timer Compare
60	GPIO1_1	DIO	E	Normal Core0 GPIO	TM1_CAP1	DI	normal core 0 Timer Capture
61	GPIO1_0	DIO	E	Normal Core0 GPIO	TM1_CAP0	DI	normal core 0 Timer Capture
62	GPIO0_31	DIO	E	IO-Control Core GPIO	PP_DATA31/ MOSI0	DIO/ DO	Parallel port data line/ IO-Control core SPI MOSI
63	GPIO0_30	DIO	E	IO-Control Core GPIO	PP_DATA30/ SCLK0	DIO/ DO	Parallel port data line/ IO-Control core SPI clock
64	VSS2	G	E	GND for 1.8 V supply			
65	VDD2	P	E	1.8 V supply			
66	GPIO0_29	DIO	E	IO-Control Core GPIO	PP_DATA29/ SS0	DIO/ DO	Parallel port data line/ IO-Control core SPI slave select
67	GPIO0_28	DIO	E	IO-Control Core GPIO	PP_DATA28/ MISO0	DIO/ DI	Parallel port data line/ IO-Control core SPI MISO
68	GPIO0_27	DIO	E	IO-Control Core GPIO	PP_DATA27/ TM0_COM1	DIO/ DO	Parallel port data line/ IO-Control core Timer Compare
69	GPIO0_26	DIO	E	IO-Control Core GPIO	PP_DATA26/ TM0_COM0	DIO/ DO	Parallel port data line/ IO-Control core Timer Compare
70	GPIO0_25	DIO	E	IO-Control Core GPIO	PP_DATA25/ TM0_CAP1	DIO/ DI	Parallel port data line/ IO-Control core Timer Capture
71	VSSIO5	G	E	GND for 3.3 V supply			
72	VDDIO5	P	E	3.3 V supply			
73	GPIO0_24	DIO	E	IO-Control Core GPIO	PP_DATA24/ TM0_CAP0	DIO/ DI	Parallel port data line/ IO-Control core Timer Capture
74	GPIO0_23	DIO	E	IO-Control Core GPIO	PP_DATA23	DIO	Parallel port data line
75	GPIO0_22	DIO	E	IO-Control Core GPIO	PP_DATA22	DIO	Parallel port data line
76	GPIO0_21	DIO	N	IO-Control Core GPIO	PP_DATA21	DIO	Parallel port data line

77	GPIO0_20	DIO	N	IO-Control Core GPIO	PP_DATA20	DIO	Parallel port data line
78	GPIO0_19	DIO	N	IO-Control Core GPIO	PP_DATA19	DIO	Parallel port data line
79	VSSIO4	G	N	GND for 3.3 V supply			
80	VDDIO4	P	N	3.3 V supply			
81	GPIO0_18	DIO	N	IO-Control Core GPIO	PP_DATA18	DIO	Parallel port data line
82	GPIO0_17	DIO	N	IO-Control Core GPIO	PP_DATA17	DIO	Parallel port data line
83	GPIO0_16	DIO	N	IO-Control Core GPIO	PP_DATA16	DIO	Parallel port data line
84	GPIO0_15	DIO	N	IO-Control Core GPIO	PP_DATA15	DIO	Parallel port data line
85	GPIO0_14	DIO	N	IO-Control Core GPIO	PP_DATA14	DIO	Parallel port data line
86	VSSIO3	G	N	GND for 3.3 V supply			
87	VDDIO3	P	N	3.3 V supply			
88	GPIO0_13	DIO	N	IO-Control Core GPIO	PP_DATA13	DIO	Parallel port data line
89	GPIO0_12	DIO	N	IO-Control Core GPIO	PP_DATA12	DIO	Parallel port data line
90	GPIO0_11	DIO	N	IO-Control Core GPIO	PP_DATA11	DIO	Parallel port data line
91	GPIO0_10	DIO	N	IO-Control Core GPIO	PP_DATA10	DIO	Parallel port data line
92	GPIO0_9	DIO	N	IO-Control Core GPIO	PP_DATA9	DIO	Parallel port data line
93	GPIO0_8	DIO	N	IO-Control Core GPIO	PP_DATA8	DIO	Parallel port data line
94	VSSIO2	G	N	GND for 3.3 V supply			
95	VDDIO2	P	N	3.3 V supply			
96	VDD1	P	N	1.8 V supply			
97	VSS1	G	N	GND for 1.8 V supply			
98	PP_AV_N	DO	N	Parellel port av_n output pin			
99	PP_WAIT_N	DO	N	Parellel port wait_n output pin			
100	PP_WR_N	DI	N	Parallel port wr_n input pin			

2.5.5 Test Pins

pin no	name	test	test type	test description
1	GPIO0_0	FW_Ceb	DI	Flash Wrapper direct Ceb
2	GPIO0_1	FW_Web	DI	Flash Wrapper direct Web
3	GPIO0_2	FW_Oeb	DI	Flash Wrapper direct Oeb
4	GPIO0_3	FW_PROG2	DI	Flash Wrapper direct PROG2
5	GPIO0_4	FW_TMEN	DI	Flash Wrapper direct TMEN
6	GPIO0_5	FW_IP_MODE	DI	Flash Wrapper IP_MODE select
7	GPIO0_6	FW_Gbus0	DIO	Flash Wrapper Register data IO
8	GPIO0_7	FW_Gbus1	DIO	Flash Wrapper Register data IO
9	GPIO0_10	MBIST_test_mode	DI	MBIST test enable
10	GPIO0_11	MBIST_start	DI	MBIST test start
11	GPIO0_12	MBIST_done	DO	MBIST test done output
12	GPIO0_13	MBIST_fail0	DO	MBIST test fail0 output
13	GPIO0_14	MBIST_fail1	DO	MBIST test fail1 output
14	GPIO0_15	MBIST_fail2	DO	MBIST test fail2 output

15	GPIO0_16	MBIST_fail3	DO	MBIST test fail3 output
16	GPIO0_22	FW_Reg_sel0	DI	Flash Wrapper Register select
17	GPIO0_23	FW_Reg_sel1	DI	Flash Wrapper Register select
18	GPIO0_24	FW_Reg_sel2	DI	Flash Wrapper Register select
19	GPIO0_25	FW_tm_latch	DI	Flash Wrapper Register latch
20	GPIO0_26	FW_Gbus2	DIO	Flash Wrapper Register data IO
21	GPIO0_27	FW_Gbus3	DIO	Flash Wrapper Register data IO
22	GPIO0_28	FW_Gbus4	DIO	Flash Wrapper Register data IO
23	GPIO0_29	FW_Gbus5	DIO	Flash Wrapper Register data IO
24	GPIO0_30	FW_Gbus6	DIO	Flash Wrapper Register data IO
25	GPIO0_31	FW_Gbus7	DIO	Flash Wrapper Register data IO
26	GPIO1_0	ScanDataIn0	DI	Scan chain data input
27	GPIO1_1	ScanDataIn1	DI	Scan chain data input
28	GPIO1_2	ScanDataIn2	DI	Scan chain data input
29	GPIO1_3	ScanDataIn3	DI	Scan chain data input
30	GPIO1_4	ScanEnable	DI	Scan enable
31	GPIO1_5	ScanDataOut0	DO	Scan chain data output
32	GPIO1_6	ScanDataOut1	DO	Scan chain data output
33	GPIO1_7	ScanClock/Test clock	DI	Scan chain clock/ Flash and MBIST test clock
34	GPIO2_0	ScanDataOut2	DO	Scan chain data output
35	GPIO2_1	ScanDataOut3	DO	Scan chain data output
36	GPIO2_2	LSRCout	DO	LSRC signal in exposed mode
37	GPIO2_3	HXTALout	DO	HXTAL signal in exposed mode

2.5.6 IO Pins Bounding

2.6 Process Technology

The RUMPS401 design is targeted for Silterra CL180E+CL180LP mix process. The digital logic, SRAM and IO buffers are based on CL180LP technology, while the Flash and other analog devices are based on CL180E technology.

Points of note in design and verification process are:

1. Silterra does not have the CL180LP PDK. The CL180E PDK is used with CL180LP Spice simulation model and CL180LP library for low power circuitries (Std. Cells, SRAM and IO buffers).
    a. Customer is required to draw LP NMOS recognition layer (GDS # 201) and LP PMOS recognition layer (GDS # 202)
    b. LP NMOS reticle (53AA) and LP PMOS reticle (54AA) will be generated by Silterra for LP transistor implants (Vt and LDD)

2. Use the CL180E PDK and CL180G library for the rest of the “non-low power circuitries”, i.e SST flash and the analog devices.
3. DRC : use CL180E DRC deck since CL180LP design rules are same as CL180G, CL180E for 1.8V/3.3V devices.
4. LVS : use CL180E LVS

2.7 Tools Version

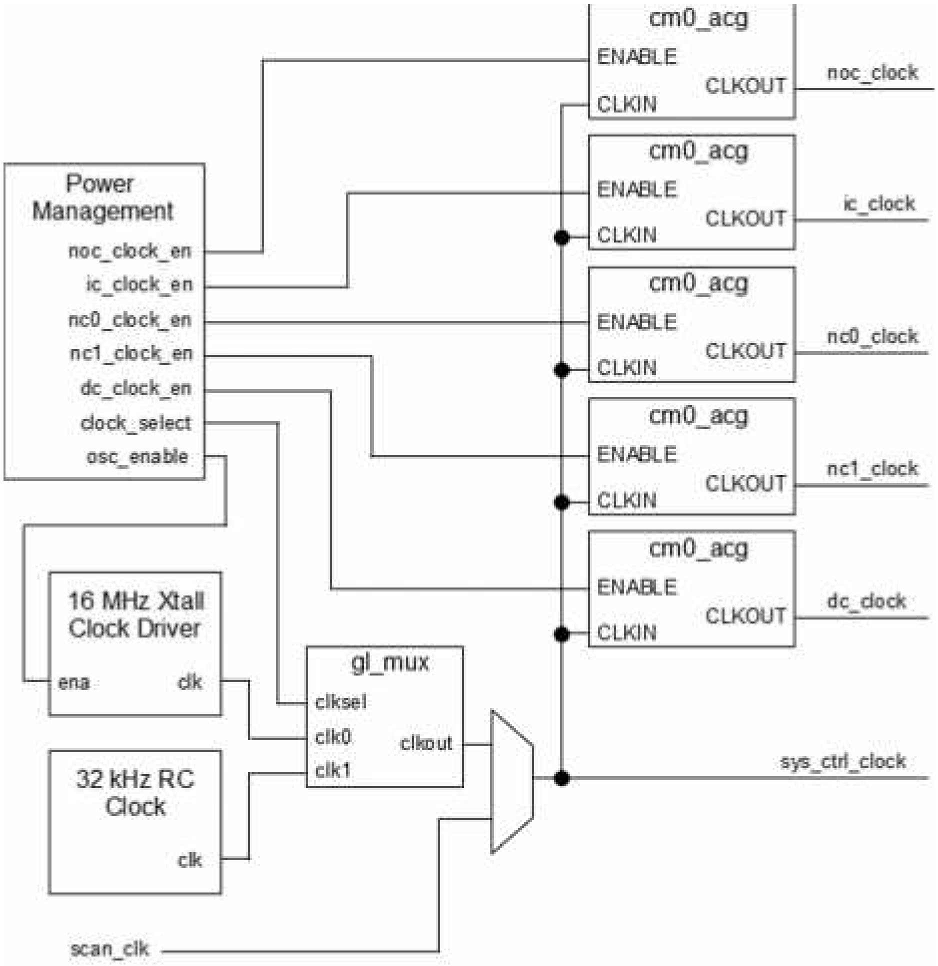
The versions of the tools that are used in the design process:

1. RTL design and simulation: Synopsys VCS G-2012.09
2. Synthesis: Synopsys DC 2013.03-SP5
3. Equivalence check: Synopsys Formality H-2013.03-SP2
4. Library Preparation: Milkyway Environment Tools H-2013.03-SP3
5. Physical design: Synopsys ICC 2013.03-SP3

3. System Control

3.1. Clock

The clock for the entire part of the chip is supplied by 16 MHz crystal clock. The crystal driver circuit is embedded in the IO buffer. In addition, 32 kHz RC clock is used for sleep mode (refer to section 3.5). In the scan test mode, the clock signal is supplied by the external scan clock source. In this mode, the clock gating circuit is bypassed to let the scan clock directly supply the clock pin of the flip-flops.

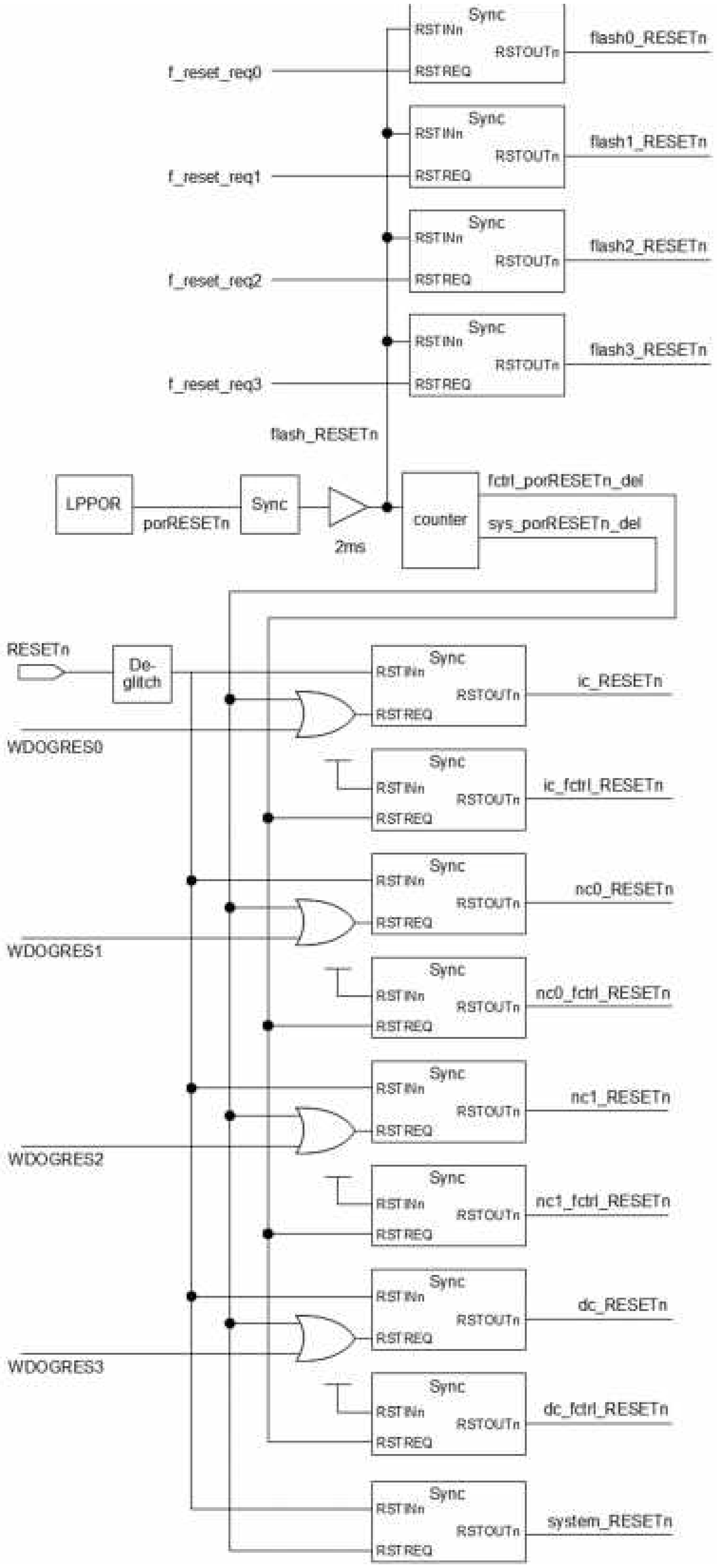


There are six generated clock signal (refer to section 3.5 for the details):

- sys_ctrl_clock : System ctrl always on clock. This clock is also used by the special GPIO driver to detect external wake-up signals.
- noc_clock : NoC + AHB2NOC bridge clock. This clock is only gated off when all the cores are in the sleep mode.
- ic_clock : IO-control core clock. Gated off when the IO-control core is in the sleep mode.
- nc0_clock : normal core 0 clock. Gated off when the normal core 0 is in the sleep mode.

- nc1_clock : normal core 0 clock. Gated off when the normal core 1 is in the sleep mode.
- dc_clock : DSP core clock. Gated off when the DSP core is in the sleep mode.

3.2. Reset


Sync
RSTINn
RSTOUTn
RSTREQ
flash0_RESETn
f_reset_req0
Sync
RSTINn
RSTOUTn
RSTREQ
flash1_RESETn
f_reset_req1
Sync
RSTINn
RSTOUTn
RSTREQ
flash2_RESETn
f_reset_req2
Sync
RSTINn
RSTOUTn
RSTREQ
flash3_RESETn
f_reset_req3
flash_RESETn
LPPOR
porRESETn
Sync
2ms
counter
fctrl_porRESETn_del
sys_porRESETn_del
RESETn
De-glitch
Sync
RSTINn
RSTOUTn
RSTREQ
ic_RESETn
WDOGRES0
Sync
RSTINn
RSTOUTn
RSTREQ
ic_fctrl_RESETn
Sync
RSTINn
RSTOUTn
RSTREQ
nc0_RESETn
WDOGRES1
Sync
RSTINn
RSTOUTn
RSTREQ
nc0_fctrl_RESETn
Sync
RSTINn
RSTOUTn
RSTREQ
nc1_RESETn
WDOGRES2
Sync
RSTINn
RSTOUTn
RSTREQ
nc1_fctrl_RESETn
Sync
RSTINn
RSTOUTn
RSTREQ
dc_RESETn
WDOGRES3
Sync
RSTINn
RSTOUTn
RSTREQ
dc_fctrl_RESETn
Sync
RSTINn
RSTOUTn
RSTREQ
system_RESETn

There are three reset sources: power-on-reset (POR), RESETN pin and watchdog timer reset request. The POR is generated every time the chip is powered on. All components in the chip are affected by the POR. The POR reset is generated by the internal POR circuit. 2ms delay is inserted to delay the de-assertion of POR. This delayed reset signal is then used to reset the flash macro. Two additional delays are used to generate two reset signals, one to reset the flash controller start-up sequence sub-module (fctrl_porRESETn_del) and the other one to reset the rest of the circuits (sys_porRESETn_del). The fctrl_porRESETn_del reset is released 500 ns after the reset to the flash macro is released. The delay is required as the timing specification of the flash IP. After comes out of reset, the flash controllers read and rewrite the configuration registers of the flash IP. The sys_porRESETn_del is released after the flash controllers finish their tasks.

The RESETN pin is a normal reset pin. There is a de-glitch circuit to remove glitch from the pin. Reset signal from RESETN pin resets all the circuit except the flash macro and flash controller start-up sequence sub-module.

There are four watchdog timer reset request lines. Each request comes from the each processor core. A watchdog timer reset request is asserted when the corresponding core failed to update the watchdog timer hence the timer is overrun. Reset signal is generated to reset only the individual core where the request came from.

To enable this reset scheme, there are 10 reset signals generated by the reset controller:

- flash_RESETn: reset the flash macro
- sc_fctrl_RESETn: reset the flash controller start-up sequence sub-module of the IO-control core
- sc_RESETn: reset the rest of the components in the IO-control core
- nc0_fctrl_RESETn: reset the flash controller start-up sequence sub-module of the 1st normal core
- nc0_RESETn: reset the rest of the components in the 1st normal core
- nc1_fctrl_RESETn: reset the flash controller of start-up sequence sub-module the 2nd normal core
- nc1_RESETn: reset the rest of the components in the 2nd normial core
- dc_fctrl_RESETn: reset the flash controller start-up sequence sub-module of the DSP core
- dc_RESETn: reset the rest of the components in the DSP core
- system_RESETn: reset the registers outside the cores (system controller and NoC)

3.3. Mode

There are three mode pins that are provided to set the chip into several modes.

mode 2	mode 1	mode 0	description
0	0	0	Normal mode 0 (4 kB)
0	0	1	Normal mode 1 (8 kB)
0	1	0	Software Bootloading mode
0	1	1	Hardware Bootloading mode
1	0	0	Scan Chain test mode
1	0	1	MBIST test mode
1	1	0	Flash test mode
1	1	1	Exposed mode

The modes are:

- Normal mode 0: normal chip operating mode with the IO-control core program code address shifted to 4kB (refer to section 4.1).
- Normal mode 1: normal chip operating mode with the IO-control core program code address shifted to 8kB (refer to section 4.1).
- Software bootloading mode: chip in software bootloading mode, IO-control core acts as a bootloader (refer to section 8.2).
- Hardware bootloading mode: chip in hardware bootloading mode, the hardware bootloader is active and ready to store program codes to the flash memory of the IO-control core (refer to section 8.1).
- Scan chain test mode: chip in the scan chain test mode.
- MBIST test mode: chip in the MBIST test mode, the MBIST circuits are active and ready to perform test to the SRAM modules. Refer to section 2.5.5 for the list of the pins used.
- Flash test mode: chip in the flash test mode, the flash wrapper is active and ready to perform test to the Flash modules. Refer to section 2.5.5 for the list of the pins used.
- Exposed mode: chip in the exposed mode, the 32kHz and 16 MHz clock signals are exposed out to the pins. Refer to section 2.5.5 for the list of the pins used.

3.4. Pin Muxing

Each of the cores has a set of IO pins. The IO-control core has 32 IO pins while each of the normal cores has eight pins. The available IO pins are shared by several peripheral in the cores. There is a local mux in every core that control the multiplexing of the peripheral IOs (refer to section 4.9 and 5.8). In addition, some of the IOs of the IO-control core are multiplexed with the test circuits IOs. The modes are MBIST test mode and Flash test mode. The pin controller circuit in the system controller obtains the modes from the modes controller and switches the pins accordingly. The configuration of the test pins is tabulated in section 2.4.

3.5. Power Management Control

Power Management control module is designed to reduce the power consumption of the chip when the processor core(s) go into sleep mode. The cores go to sleep mode when it execute WFI instruction. The approaches are:

- Gate off the clock signal to the core(s) that is in the sleep mode.
- Power down the flash macro of the core(s) that is in the sleep mode.
- Use the lower speed 32 kHz RC clock source.

The sleep and wake-up procedures for each of the cores are:

- Core active => sleep signal => gate off clock => isolate flash output => power down flash => Core sleep
- Core sleep => wake up signal => power up flash => turn off the isolation of flash output => gate on clock => Core active

There are three wake-up sources that can be used to activate the core(s) that is in the sleep mode:

- GPIO wake-up signal. The GPIO module of each the cores have a special driver that is not gated off during the sleep mode. This special GPIO driver can detect external wake-up signal from the GPIO pin to activate the core.
- AHB2NOC bridge receive interrupt. Other cores can activate the sleeping core by sending a flit to it. The receive interrupt signal is used as the wake-up source. With this mechanism, the AHB2NOC brides of the cores are not gated off except in the case when all the cores are sleeping.
- Debug event. When one of the cores starts a debug event, all sleeping cores are activated. Once all the cores are ready, only the debug register values are pulled by the debugger circuit.

With the sleep and wake-up schemes as described, there are in total six generated clocks:

- sys_ctrl_clock : System ctrl always on clock. This clock is also used by the special GPIO driver to detect external wake-up signals.
- noc_clock : NoC + AHB2NOC bridge clock. This clock is only gated off when all the cores are in the sleep mode.
- ic_clock : IO-control core clock. Gated off when the IO-control core is in the sleep mode.
- nc0_clock : normal core 0 clock. Gated off when the normal core 0 is in the sleep mode.
- nc1_clock : normal core 0 clock. Gated off when the normal core 1 is in the sleep mode.
- dc_clock : DSP core clock. Gated off when the DSP core is in the sleep mode.

The 16 MHz oscillator clock source is selected in normal operating mode. The 32 kHz RC clock source is only selected when all the cores are in the sleep mode.

4. IO-control Core

The IO-control core is the main processor core in the system. It contains more resources as compared to the other cores. It also has different bootloading mechanism to store the program codes and has two logical program code partitions in it flash memory. One of the partition is used to store the application code, while the other one is for the software bootloader code. The IO-control core can be set to execute this software bootloader code and act as software bootloader to load its own application code or to load the normal cores application code.

4.1. Memory Mapping

The IO-control core has nine slave modules attached to the AHB-lite bus. The address of the slaves are:

Slave num	Slave		start address	end address
1	Flash	Software Bootloading Mode	0x0000_0000	0x0000_83FF
		Normal Mode 0	0x0000_0000	0x0000_73FF
		Normal Mode 1	0x0000_0000	0x0000_63FF
2	SRAM		0x2000_0000	0x2000_1FFF
3	APB		0x4000_0000	0x4000_3FFF
4	GPIO		0x5000_0000	0x5000_FFFF
5	Mux controller		0x5001_0000	0x5001_FFFF
6	Debug		0x5002_0000	0x5002_FFFF
7	Parallel port		0x5003_0000	0x5003_FFFF
8	NoC bridge		0x5004_0000	0x5004_FFFF

In addition some peripherals are attached to the APB bus. The address mapping of the APB peripherals are:

Slave num	Slave	start address	end address
1	APB Flash	0x4000_0000	0x4000_0FFF
2	APB Timer	0x4000_1000	0x4000_1FFF
3	APB SPI	0x4000_2000	0x4000_2FFF
4	APB WDT	0x4000_3000	0x4000_3FFF

The address range of the flash memory of the IO-control core depends on the current mode of the chip. The AHB-lite bus of the IO-control core has the capability to remap the AHB address (HADDR) of the transfer to the flash memory according to the mode. In the Software Bootloading mode, the AHB-lite bus does not implement the address remap, hence the M0 processor is able to access the whole flash memory cells. In the normal mode 0 and 1, the AHB-lite bus performs 4 kB and 8 kB address remap respectively. For an example, when the M0 processor tries to access address 0x200 in normal mode 0, the AHB-lite bus remap the HADDR to 0x1200. This mechanism is used to reserve the software bootloader that is stored in the flash memory start from address 0. The application program can be stored at the flash memory start from address 0x1000 (4kB) or 0x2000 (8kB). The chip can be either set to run in normal mode 0 or 1 depending on the start address where the application program is stored. Refer to section 8 for more detail of the bootloading mechanism.

4.2. Interrupt Vector Table

The highest priority interrupt (NMI) of the IO-control core is used for debug event (refer to section 7). 16 interrupt pins are used as peripheral interrupts. The interrupt vector table are:

no	interrupt	description
0	flash	flash APB interface interrupt
1	parallel port tx	parallel port transmit interrupt
2	parallel port rx	parallel port receive interrupt
3	noc tx	NoC transmit interrupt
4	noc rx	NoC receive interrupt
5	noc bd	NoC bootloader-debugger interrupt
6	timer	timer interrupt
7	spi	SPI interrupt
8	gpio0	GPIO interrupt 0
9	gpio1	GPIO interrupt 1
10	gpio2	GPIO interrupt 2
11	gpio3	GPIO interrupt 3
12	gpio4	GPIO interrupt 4
13	gpio5	GPIO interrupt 5
14	wdt	Watch Dog Timer interrupt
15		

4.3. Flash Controller

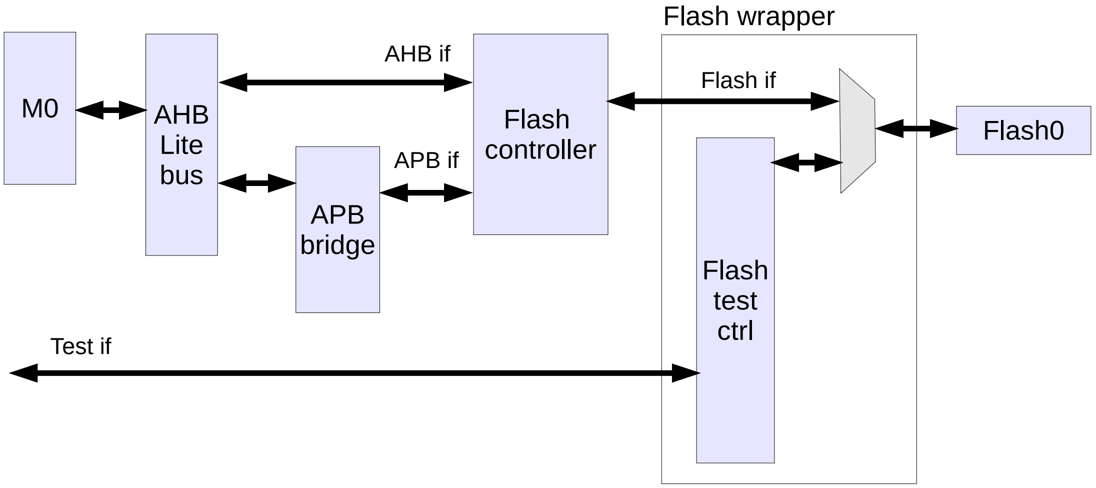
32 Kbytes SuperFlash from SST is used as the non-volatile memory of the IO-control core. The flash controller module interfaces the M0 processor and the flash IP. This module provides two access interface, AHB and APB. The fast AHB interface is provided for read access, while the APB interface provides the registers for program, erase and other operations.

4.3.1 Flash Controller Features

- 32 Kbytes main memory array, organized into 512 bytes sectors. In total there is 64 sectors.
- Two 512 bytes NVR sectors. The first NVR sector is user access-able, while the second one is only for manufacturer configuration storage.
- Support byte, half word and word read transfer.

- Support chip erase
- Support sector erase
- Support byte programming
- Support erase-retry and no-retry erase mechanism
- Support erase and program protection to main array
- Support read, erase and program protection to NVR sector
- Interrupt generation to indicate program/erase is done
- Support positive and negative timing offset

4.3.3 Flash Controller Connection Diagram

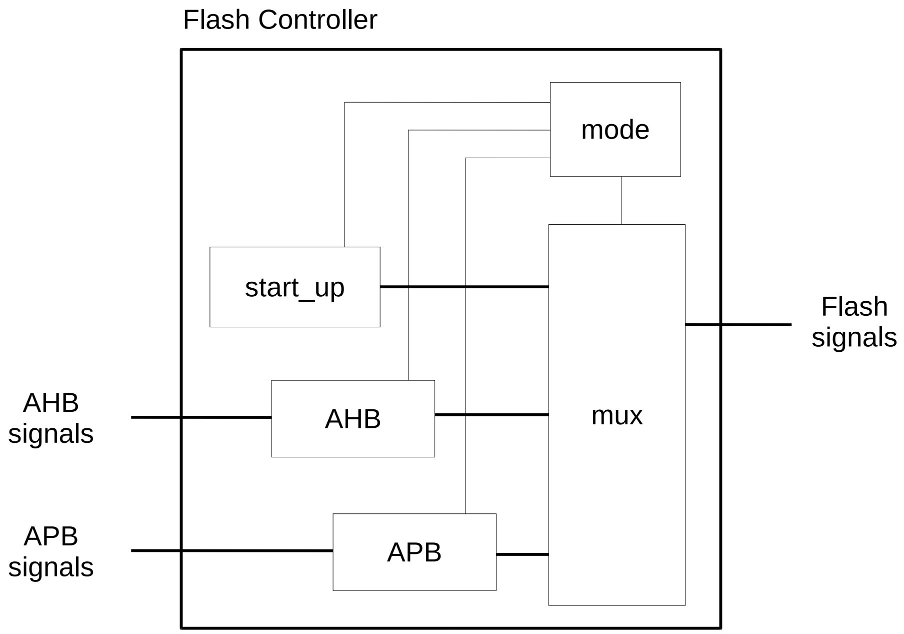


4.3.4 Flash Controller IO Pins

Signal	IO	Source	Dest’ n	Function
AMBA APB				
PRESETn	DI	APB Bridge	Controller	AMBA APB active low reset
PCLK	DI	APB Bridge	Controller	AMBA APB clock. The rising edge of pclk times all transfers on APB
PSEL	DI	APB Bridge	Controller	AMBA APB slave select
PENABLE	DI	APB Bridge	Controller	AMBA APB enable strobe
PWRITE	DI	APB Bridge	Controller	AMBA APBA Write/Read strobe
PADDR[5:0]	DI	APB Bridge	Controller	AMBA APB address bus
PWDATA[31:0]	DI	APB Bridge	Controller	AMBA APB write data
PRDATA[31:0]	DO	Controller	APB Bridge	AMBA APB read data
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Controller	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Controller	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Controller	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Controller	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Controller	Indicates the transfer type of the current transfer

HWDATA[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Controller	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Controller	AHB ready input
HRDATA[31:0]	DO	Controller	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Controller	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Controller	AHB Lite Bus	Indicates the transfer response
System Control				
Flash				
A[14:0]	DO	Controller	Flash	Address bus
DIN[7:0]	DO	Controller	Flash	Data in
NVR	DO	Controller	Flash	NVR select. NVR=1 to sellect the NVR sectors, NVR=0 to select main array.
CEb	DO	Controller	Flash	Chip enable (active low)
WEb	DO	Controller	Flash	Write enable (active low)
PROG	DO	Controller	Flash	To enable program operation.
PROG2	DO	Controller	Flash	To enable YMUX during program.
ERASE	DO	Controller	Flash	To indicate erase operation.
CHIP	DO	Controller	Flash	to indicate chip selection.
OEb	DO	Controller	Flash	Output enable (active low)
CONFEN	DO	Controller	Flash	To write to configuration register.
VREAD	DO	Controller	Flash	VREAD=1 to enable verify read for erase cell
DOUT[7:0]	DI	Flash	Controller	Data out
Interrupt				
FLASH_INT	DO	Controller	M0	Program/Erase interrupt

4.3.5 Flash Controller Block Diagram



4.3.6 Flash Controller Memory Map

Usage	Sector	Address	Size (bytes)	Access allowed
Code	Sector 0	0x0000_0000 - 0x0000_01FF	512	Read
	Sector 1	0x0000_0200 - 0x0000_03FF	512	Read
	…	…	…	Read
	Sector 64	0x0000_7E00 - 0x0000_7FFF	512	Read
Registers	NVR0	0x0000_8000 - 0x0000_81FF	512	Read
Foundry	NVR1	0x0000_8200 - 0x0000_83FF	512	Read

4.3.7 Flash Controller Timing

delay type	required duration	number of clock cycle	added 10% margin / mid	selected count
web_setup	1000 ns	16	17.6	18
prog_setup	2500 - 3150 ns	40 - 50.4	45.2	46
prog	20000 - 25000 ns	320 - 400	360	360
recovery	5000 ns	80	88	88
latency	100 ns	1.6	1.76	2
erase_web_setup	1000 ns	16	17.6	18
erase	10 - 12.5 ms	160,000 - 200,000	180,000	180,000
erase_recovery	50000 ns	800	880	880
chip_erase_web_setup	1000 ns	16	17.6	18
chip_erase	40 - 50 ms	640,000 - 800,000	720,000	720,000

chip_erase_recovery	50000 ns	800	880	880
por_flash_reset	500 ns	8	8.8	9
por_system_reset	1500 ns (8*5 cycle)	40	44	44

4.3.8 Flash Controller Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	W	FLASH_KEY	0x00000000	Key to enable erase/program
0x04	R/W	FLASH_NVRP	0x00000000	NVR sector protection
0x08	R/W	FLASH_ERASCTR	0x00000000	Sector index for erase
0x0C	R/W	FLASH_ERA	0x00000000	Command to start erase
0x10	R/W	FLASH_PROGADDR	0x00000000	Address for program (includes main sectors and NVR)
0x14	R/W	FLASH_PROGDATA	0x00000000	Data for program (includes main sectors and NVR)
0x18	R/W	FLASH_PROG	0x00000000	Command to start program
0x1C	R/W	FLASH_IE	0x00000000	Interrupt enable
0x20	R	FLASH_IF	0x00000000	Interrupt flag
0x24	R/W	FLASH_TIME	0x00000000	Flash timing offset register

Register Description

FLASH_KEY

address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:0	W	KEY	0	Erase/Program Key 0xC6A5: Erase/Program is allowed Others: Erase/Program is not allowed

FLASH_NVRP

address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31:0	W	NVRP	0	NVR Protection 0xA55A: Read access to NVR is enabled 0x5AA5: Read/Erase/Program for NVR is enabled Others: Access to NVR is disabled

FLASH_ERASCTR

address offset: 0x08

Bits	R/W	Field Name	Reset value	Description
31:9	R	RESERVED	0	unused

8:0	R/W	SCTRERAI	0	Sector index Select the index of the main sector to be erased.

FLASH_ERA

address offset: 0x0C

Bits	R/W	Field Name	Reset value	Description
31:9	R	RESERVED	0	unused
8	R/W	ERARETRY	0	When this bit is set to 1, the 'erase-verify-retry' mechanism is performed for the sector erase operation. Maximum of 4 erase attempts will be performed. When this bit is clear to 0, the sector erase will be performed with the maximum required erase time applied.
7:3	R	RESERVED	0	unused
2	R/W	NVRERA	0	Writing 1 will initiate erase operation to 1st NVR sector.
1	R/W	CERA	0	Writing 1 will initiate erase operation to all main sectors.
0	R/W	MERA	0	Writing 1 will initiate erase operation to selected main sector.

FLASH_PROGADDR

address offset: 0x10

Bits	R/W	Field Name	Reset value	Description
31:17	R	RESERVED	0	unused
16:0	R/W	PROGADDR	0	Define the address of the byte to be programed. When NVRPROG is set to 1, the PROGADDR[6:0] define the address in the 1st NVR. When MPROG is set to 1, PROGADDR define the address in the main flash sectors.

FLASH_PROGDATA

address offset: 0x14

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	R/W	PROGDATA	0	Data to be written to the flash.

FLASH_PROG

address offset: 0x18

Bits	R/W	Field Name	Reset value	Description
31:2	R	RESERVED	0	unused
1	R/W	NVRPROG	0	Writing 1 will initiate program operation to a byte in the 1st NVR sector. The address of the byte to be programmed in the 1st NVR sector is defined by PROGADDR[6:0]. PROGADDR[16:10] takes no effect.

0	R/W	MPROG	0	Writing 1 will initiate program operation to a byte in the main flash sectors. The address of the byte to be programmed in the 1st NVR sector is defined by PROGADDR

FLASH_IE

address offset: 0x1C

Bits	R/W	Field Name	Reset value	Description
31:1	R	RESERVED	0	unused
0	R/W	FIE	0	Set to 1 to enable flash interrupt. Interrupt is generated every time program/erase process is completed.

FLASH_IF

address offset: 0x20

Bits	R/W	Field Name	Reset value	Description
31:1	R	RESERVED	0	unused
0	R/W	FIF	0	This bit is set by hardware when program/erase is completed. It is cleared by a read to this register. An interrupt is generated if FIE is 1.

FLASH_TIME

address offset: 0x24

Bits	R/W	Field Name	Reset value	Description
31:3	R	RESERVED	0	unused
2	R/W	TIMEMODE	0	Time offset mode. When this flag is 0, the time offset applied increases the flash access time duration (positive offset), when this flag is 0, the time offset decreses the flash access time duration (negative offset).
1:0	R/W	TIMEOFFSET	0	Time offset: 'b00: no offset 'b01: 5% offset 'b10: 10% offset 'b11: 15% offset

4.4. SRAM Controller

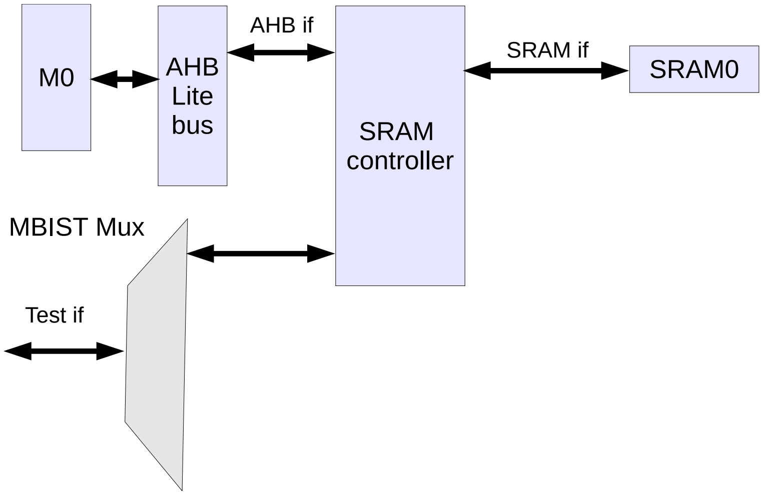
8 Kbytes SRAM is implemented as the data memory. The SRAM is generated from SRAM generator provided by ARM. SRAM controller provides interface for the M0 processor to access the SRAM module.

4.4.1 SRAM Controller Features

- Support byte, halfword and word read/write access
- Zero-turn-around for read-write operation

- Embedded Memory-Build-In-Self-Test (MBIST)

4.4.2 SRAM Controller Connection Diagram

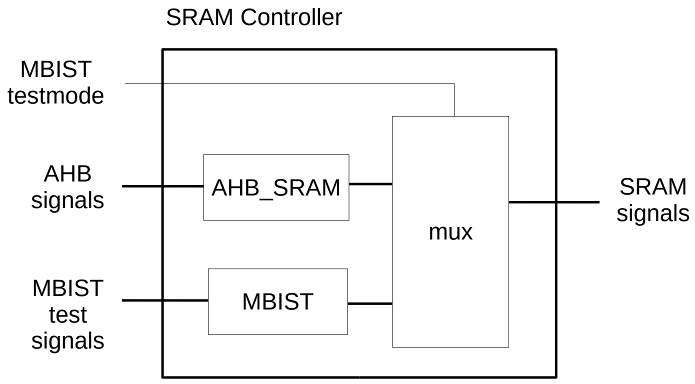


4.4.3 SRAM Controller IO Pins

Signal	IO	Source	Dest’n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Controller	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Controller	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Controller	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Controller	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Controller	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Controller	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Controller	AHB ready input
HRDATA[31:0]	DO	Controller	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Controller	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Controller	AHB Lite Bus	Indicates the transfer response

SRAM				
SADDR[12:0]	DO	Controller	SRAM	SRAM address bus
SWDATA[31:0]	DO	Controller	SRAM	SRAM Data in
SCE	DO	Controller	SRAM	SRAM Chip enable
SR_WB	DO	Controller	SRAM	SRAM read/write
SBEN[31:0]	DO	Controller	SRAM	SRAM bit select
SRDATA[31:0]	DI	SRAM	Controller	SRAM Data out
MBIST				
MBIST_clk	DI	Tester	Controller	Clock in MBIST test mode
MBIST_nreset	DI	Tester	Controller	Reset signal to reset MBIST circuit
MBIST_testmode	DI	Tester	Controller	Test mode pin to the MBIST
MBIST_start	DI	Tester	Controller	Start signal to start the test
MBIST_done	DO	Controller	Tester	Done signal that indicate the test is done
MBIST_fail	DO	Controller	Tester	Signal to indicate whether the test is fail or pass

4.4.4 SRAM Controller Block Diagram



4.5. GPIO

4.5.1 GPIO Features

- 32-bit I/O ports with pull up capability
- Level/edge-triggered interrupt generation
- Bit masking access to output register using address values.
- Atomic bitwise set, clear and toggle access to output register
- Thread safe separate set and clear addresses for control registers.

4.5.2 GPIO IO Pins

Signal	IO	Source	Dest’n	Function
AMBA AHB				

HRESETn	DI	AHB Lite Bus	Controller	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Controller	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Controller	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Controller	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Controller	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Controller	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Controller	AHB ready input
HRDATA[31:0]	DO	Controller	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Controller	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Controller	AHB Lite Bus	Indicates the transfer response
GPIO				
GPIOIN [31:0]	DI	Pin mux	Controller	GPIO data in
GPIOOEN [31:0]	DO	Controller	Pin mux	GPIO output enable
GPIOOUT [31:0]	DO	Controller	Pin mux	GPIO data out
INTERRUPT				
GPIOINT[5:0]	DO	Controller	M0	Transmit interrupt

4.5.3 Interrupt generation

The GPIO inputs [5:0] are able to trigger interrupt to the core. IE, IT, IP and IF are all 6-bit registers.

IE[n]	IT[n]	IP[n]	Interrupt feature
0	-	-	Disabled
1	0	0	Low-level
1	0	1	High-level
1	1	0	Falling-edge
1	1	1	Rising-edge

4.5.4 Masked Access

Masked access allows bit-wise access of 8-bit GPIO pins in a single transfer. The address of the access is used as the mask for the read or write operations.

There are 1024 (4x256) word addresses, corresponding to all possible bit mask pattern for 4x8 bits.

The mask is from HADDR[9:2]. The base address of the bit mask address is at 0x0400, 0x0800, 0x0C00 and 0x1000 for GPIO index [7:0], [15:8], [23:16] and [31:24] respectively.

Example 1:

Set bit 2 & 3, clear bit 5 & 7.

HADDR[9:2] = 0xAC (which is b10101100)

HADDR[12:10] = 0x1

HWDATA = 0x0C

Example 2:

Set bit 8 & 9, clear bit 10 & 11.

HADDR[9:2] = 0x0F (which is b00001111)

HADDR[12:10] = 0x2

HWDATA = 0x03

4.5.5 GPIO Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	R	DATAIN	0x00000000	Data input value
0x04	R/W	DATAOUT	0x00000000	Data output register
0x08	R/W	OEN_SET	0x00000000	Output enable control
0x0C	R/W	OEN_CLR	0x00000000	Output enable control
0x10	W1C	IF	0x00000000	Interrupt flag
0x14	W1S	IE_SET	0x00000000	Interrupt enable control
0x18	W1C	IE_CLR	0x00000000	Interrupt enable control
0x1C	W1S	IT_SET	0x00000000	Interrupt type control
0x20	W1C	IT_CLR	0x00000000	Interrupt type control
0x24	W1S	IP_SET	0x00000000	Interrupt polarity control
0x28	W1C	IP_CLR	0x00000000	Interrupt polarity control
0x2C	W	BSET	Not defined	Bit-wise set the DATAOUT

0x30	W	BCLR	Not defined	Bit-wise clear the DATAOUT
0x34	W	BTGL	Not defined	Bit-wise toggle the DATAOUT

Register Description

DATAIN

address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:0	R	-	0	Data input value Read – sample at pin

DATAOUT

address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31:0	R/W	-	0	Data output register Read – current value of data output register Write – to data output register

OEN_SET

address offset: 0x08

Bits	R/W	Field Name	Reset value	Description
31:0	R	-	0	Output enable control. Write 0 – no effect Write 1 – set the OEN bit Read 0 – indicate the IO direction as input Read 1 – indicate the IO direction as output

OEN_CLR

address offset: 0x0C

Bits	R/W	Field Name	Reset value	Description
31:0	R	-	0	Output enable control. Write 0 – no effect Write 1 – clear the OEN bit Read 0 – indicate the IO direction as input Read 1 – indicate the IO direction as output

IF

address offset: 0x10

Bits	R/W	Field Name	Reset value	Description
31:6	R	RESERVED	0	unused
5:0	R	-	0	Interrupt flag. Write 0 – no effect Write 1 – clear the interrupt flag Read 0 – no interrupt Read 1 – interrupt has triggered

IE_SET

address offset: 0x14

Bits	R/W	Field Name	Reset value	Description
31:6	R	RESERVED	0	unused
5:0	R/W	-	0	Interrupt enable control Write 0 – no effect Write 1 – set the IE bit Read 0 – interrupt is disabled Read 1 – interrupt is enabled

IE_CLR

address offset: 0x18

Bits	R/W	Field Name	Reset value	Description
31:6	R	RESERVED	0	unused
5:0	R/W	-	0	Interrupt enable control Write 0 – no effect Write 1 – clear the IE bit Read 0 – interrupt is disabled Read 1 – interrupt is enabled

IT_SET

address offset: 0x1C

Bits	R/W	Field Name	Reset value	Description
31:6	R	RESERVED	0	unused
5:0	R/W	-	0	Interrupt type control Write 0 – no effect Write 1 – set the IE bit Read 0 – level triggered Read 1 – edge triggered

IT_CLR

address offset: 0x20

Bits	R/W	Field Name	Reset value	Description
31:6	R	RESERVED	0	unused
5:0	R/W	-	0	Interrupt type control Write 0 – no effect Write 1 – clear the IE bit Read 0 – level triggered Read 1 – edge triggered

IP_SET

address offset: 0x24

Bits	R/W	Field Name	Reset value	Description
31:6	R	RESERVED	0	unused
5:0	R/W	-	0	Interrupt polarity control Write 0 – no effect Write 1 – set the IE bit Read 0 – low-level or falling-edge triggered Read 1 – high-level or rising-edge triggered

IP_CLR
address offset: 0x28

Bits	R/W	Field Name	Reset value	Description
31:6	R	RESERVED	0	unused
5:0	R/W	-	0	Interrupt polarity control Write 0 – no effect Write 1 – clear the IE bit Read 0 – low-level or falling-edge triggered Read 1 – high-level or rising-edge triggered

BSET
address offset: 0x2C

Bits	R/W	Field Name	Reset value	Description
31:0	W	-	-	Bit-wise set the DATAOUT Write 0 – no effect Write 1 – set the DATAOUT bit

BCLR
address offset: 0x30

Bits	R/W	Field Name	Reset value	Description
31:0	W	-	-	Bit-wise clear the DATAOUT Write 0 – no effect Write 1 – clear the DATAOUT bit

BTGL
address offset: 0x34

Bits	R/W	Field Name	Reset value	Description
31:0	W	-	-	Bit-wise toggle the DATAOUT Write 0 – no effect Write 1 – toggle the DATAOUT bit

4.6. Parallel Port Controller

Parallel port module provides a parallel data transfer protocol controller as the inter-chip data transfer media. In the parallel data transfer protocol that is implemented, the module acts as the slave module. The device at the other end should be programmed to act as the master. The parallel data bus can be set to 4, 8, 12, 16, 20, 24, 28 or 32 bits. The width of the data bus is controlled by the mux controller module (refer to section 4.9). Two interrupt pins, PP_TX_INT and PP_RX_INT are provided for the M0 processor. PP_TX_INT pin will go high if a data successfully transmitted out from the module. Meanwhile the PP_RX_INT pin can be configured to be high when there is a data in the receive buffer or when the buffer is full.

4.6.1 Parallel Port Features

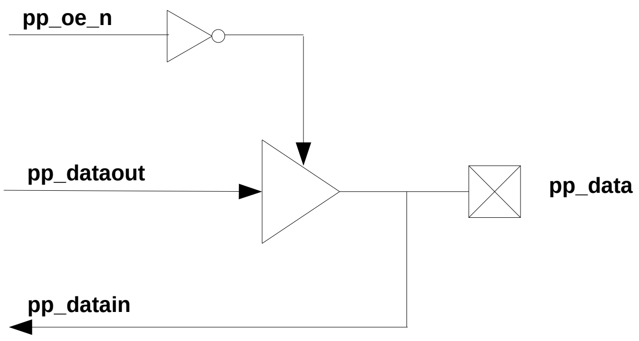
- Support 4, 8, 12, 16, 20, 24, 28 and 32 bits parallel data transfer
- Two interrupt signals provided for transmit and receive process
- Two receive interrupt generation scheme
- Two buffers with four depth and 32-bit data width

4.6.2 Parallel Port IO Pins

Signal	IO	Source	Dest’n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Controller	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Controller	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Controller	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Controller	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Controller	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Controller	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Controller	AHB ready input
HRDATA[31:0]	DO	Controller	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Controller	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Controller	AHB Lite Bus	Indicates the transfer response
PARALLEL PORT				
pp_strobe_n	DI	Other Chip	Controller	Parallel port strobe signal
pp_datain [31:0]	DI	Pin mux	Controller	Parallel port data in
pp_wr_n	DO	Controller	Other Chip	Parallel port write/read signal
pp_oe_n [31:0]	DO	Controller	Pin mux	Parallel port output enable
pp_dataout [31:0]	DO	Controller	Pin mux	Parallel port data out
pp_av_n	DO	Controller	Other Chip	Parallel port data available signal
pp_wait_n	DO	Controller	Other Chip	Parallel port wait signal
INTERRUPT				
PP_TX_INT	DO	Controller	M0	Transmit interrupt
PP_RX_INT	DO	Controller	M0	Receive interrupt

4.6.3 Parallel Port Data Pins

The data pins of the parallel port are bidirectional pins. There are 3 signals from the parallel port controller that related to the parallel port data pins, namely: pp_oe_n, pp_data_in and pp_data_out. The pp_data_in is used to retrieve the data meanwhile the pp_data_out is used to transmit the data. The pp_oe_n pin is used to control the output enable of the bidirectional buffers. The connection of these pins is:

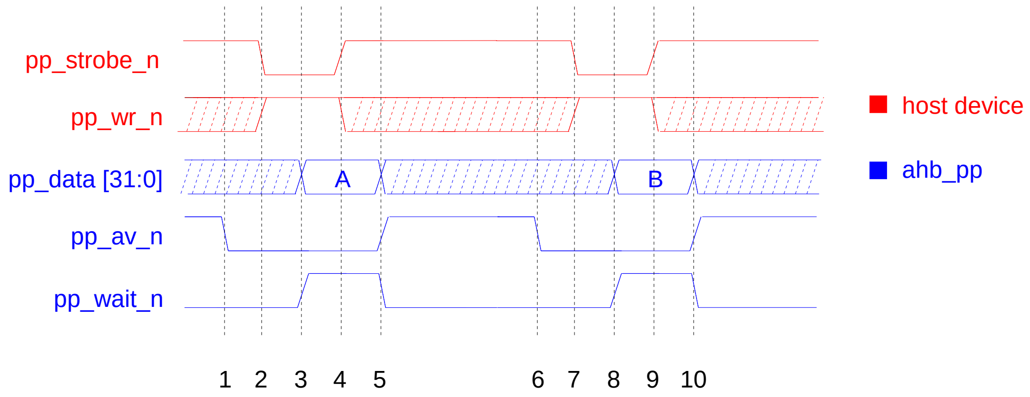


4.6.4 Parallel Port Signals

signal	driver	description
pp_strobe_n	host device	Pin to drive the signal to start read/write transfer. Host device should keep the pp_strobe_n pin high during idle period. When the host device want to start read/write transfer, pp_strobe_n pin should be driven low during the read/write. The pp_strobe_n should be pulled high after one read/write transfer is done.
pp_wr_n	host device	Pin to drive the signal to indicate the type of the transfer. Need to be driven low/high during the period when the pp_strobe_n is low. The state of pp_wr_n is ignored if the pp_strobe_n is high. If the pp_wr_n pin is high when the pp_strobe_n is low, the type of the transfer is read (host device transfer data to parallel port controller). Meanwhile if the pp_wr_n pin is low when the pp_strobe_n is low, the type of the transfer is read (parallel port controller transfer data to host device).
pp_data [31:0]	host device/ parallel port controller	The parallel port data. pp_data are bidirectional pins. High state represent logic high (1'b1) meanwhile Low state represent logic low (1'b0). The pin are driven by the host device during the write transfer, and by the ahp_pp during read transfer.
pp_av_n	Parallel port controller	Pin to indicate the availability of data in the transmit buffer of parallel port controller. When there is the data in the transmit buffer, the parallel port controller will drive this pin low. The host device is only allowed to do the read transfer when the pp_av_n pin is low.
pp_wait_n		Pin to inform the host device wheter the transfer is already processed or not. After the host device drive the pp_strobe_n pin low, it needs to wait the pp_wait_n pin to be high. If the parallel port controller module is not ready to process the data, the parallel port controller can use this signal to stall the host device until it ready. After the host device detect pp_wait_n pin high, the host device is allowed to terminate the transfer by pulling the pp_strobe_n high.

4.6.5 Parallel Port Protocol

read transfer:



1. Parallel port controller drive pp_av_n low to inform about the availability of data in its transmit buffer.

2. host device that detect the pp_av_n low, drive the pp_strobe_n low to start the read transfer, the pp_wr_n pin is driven high to indicate the transfer type is read.

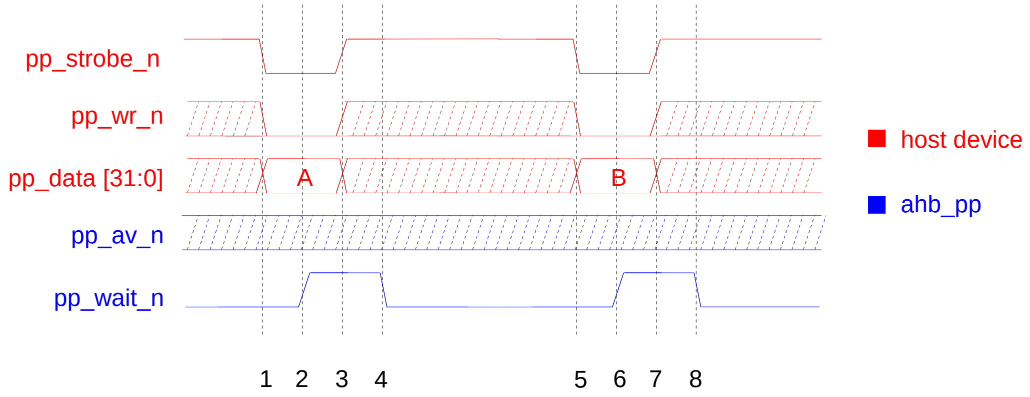
3. when the Parallel port controller is ready, it drive the pp_wait_n pin high, at the same time the valid data is available in the pp_data pins.

4. host device read the data and drive the pp_strobe_n pin high to end the transfer.

5. Parallel port controller drive the pp_wait_n low. if there is still data available in the buffer, the pp_av_n pin will be low, if no, the pp_av_n pin is pulled high.

6 - 10 another read transfer is performed.

write transfer:



1. host device drive the pp_strobe_n low to start the write transfer, the pp_wr_n pin is driven low to indicate the transfer type is write. The valid data to be written is also driven by the host device.

2. when the Parallel port controller is ready, it read the data and drive the pp_wait_n pin high.

3. host device drive the pp_strobe_n pin high to end the transfer.

4. Parallel port controller drive the pp_wait_n low.

5 - 8 another write transfer is performed.

4.6.6 Parallel Port Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	R/W	PP_BUFF	0x00000000	Parallel port data
0x04	R/W	PP_ST	0x00000000	Parallel port status

Register Description

PP_BUFF
address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:0	R/W		0	Parallel port data buffer. Write to this register will move the data to the tx buffer. Read to this register will discard the message from rx buffer (if one exist).

PP_ST
address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31:20	R	RESERVED	0	unused
19	R/W	PP_RX_INT_CLR	0	Write 1 to this bit to clear PP_RX_INT when the PP_RX_IM = 1, write 0 will take no effect.
18	R/W	PP_RX_IM_CLR	0	Write 1 to this bit to clear PP_RX_IM, write 0 will take no effect.
17	R/W	PP_RX_IE_CLR	0	Write 1 to this bit to disable the rx interrupt, write 0 will take no effect.
16	R/W	PP_TX_IE_CLR	0	Write 1 to this bit to disable the tx interrupt, write 0 will take no effect.
15:11	R	RESERVED	0	unused
10	R/W	PP_RX_IM_SET	0	Write 1 to this bit to set PP_RX_IM, write 0 will take no effect.
9	R/W	PP_RX_IE_SET	0	Write 1 to this bit to enable the rx interrupt, write 0 will take no effect.

8	R/W	PP_TX_IE_SET	0	Write 1 to this bit to enable the tx interrupt, write 0 will take no effect.
7	R	RESERVED	0	unused
6	R/W	PP_RX_IM	0	receive interrupt mode. 0 => interrupt triggered when there is data in the PP_RX_BUFF. A read to PP_BUFF will discard the data from the buffer. PP_RX_IE will remain high while the PP_RX flag is high. 1 => interrupt triggered when the PP_RX_BUFF is full. Write 1 to PP_RX_INT_CLR will pull down the interrupt.
5	R/W	PP_RX_IE	0	receive interrupt enable. 0 => interrupt disabled. 1 => interrupt enabled.
4	R/W	PP_TX_IE	0	transmit complete interrupt enable. 0 => interrupt disabled. 1 => interrupt enabled. PP_TX_INT line will be high when there is finished transfer. Write 1 to PP_TX will clear the interrupt.
3	R/W	PP_RXF	0	rx buffer full flag, 0 => tx buffer is not full; 1 => rx buffer full
2	R/W	PP_RX	0	receive flag, set high when there is data in the receive buffer. A read to PP_BUFF will discard the data from the buffer. If there is no more data in the buffer, PP will be cleared by hardware, if there is another message, the PP_RX will remain high.
1	R/W	PP_TXF	0	tx buffer full flag, 0 => tx buffer is not full; 1 => tx buffer full
0	R/W	PP_TX	0	transmit complete flag, set high when there were data written to PP_BUFF and the tx buffer is empty. Write 1 to this bit will clear the flag.

4.7. Timer

4.7.1 Timer Features

- 16-bit timer with 16-bit pre-scaler
- Two 16-bit input Capture with optional interrupt generation
- Three 16-bit compare channels that allow:
  - Continuous operation with optional interrupt
  - Stop timer on match with optional interrupt
  - Reset timer on match with optional interrupt
- Two external output corresponding to compare registers with the capability of:
  - Set low on match

- Set High on match
- Toggle on match
- PWM output

4.7.2 Timer IO Pins

Signal	IO	Source	Dest'n	Function
AMBA APB				
PRESETn	DI	APB Bridge	Timer	APB active low reset
PCLK	DI	APB Bridge	Timer	APB clock. The rising edge of pclk times all transfer on APB
PCLKG	DI	APB Bridge	Timer	APB gated clock
PSEL	DI	APB Bridge	Timer	APB slave select
PENABLE	DI	APB Bridge	Timer	APB enable strobe
PWRITE	DI	APB Bridge	Timer	APB Write/Read strobe
PADDR[6:2]	DI	APB Bridge	Timer	APB address bus
PWDATA[31:0]	DI	APB Bridge	Timer	APB write data
PRDATA[31:0]	DO	Timer	APB Bridge	APB read data
PREADY	DO	Timer	APB Bridge	APB ready
Interrupt				
TMIRQ	DO	Timer	M0	Timer interrupt
IO				
TM_CAP0	DI	Pad	Timer	Timer input capture 0
TM_CAP1	DI	Pad	Timer	Timer input capture 1
TM_COM0	DO	Timer	Pad	Timer output compare 0
TM_COM1	DO	Timer	Pad	Timer output compare 1
Debug				
DBG_HALT	DI	M0	WDT	M0 halted

4.7.3 Timer Registers

Register Overview

Address offset	R/W	Register Name	Reset value	Description
0x00	R/W	TM_CTRL	0x0000	Timer Control Register.
0x04	W0C	TM_IF	0x0000	Timer Interrupt Flag Register.
0x08	R/W	TM_CNT	0x0000	Timer Counter
0x0C	R/W	TM_PR	0x0000	Timer Prescale Register.
0x10	R/W	TM_PS	0x0000	Timer Prescale Counter.
0x14	R/W	TM_COMCR	0x0000	Timer Compare Control Register.
0x18	R/W	TM_COMR0	0x0000	Timer Compare Register 0.
0x1C	R/W	TM_COMR1	0x0000	Timer Compare Register 1.
0x20	R/W	TM_COMR2	0x0000	Timer Compare Register 2.
0x24	R/W	TM_CAPCR	0x0000	Timer Capture Control Register.
0x28	R/W	TM_CAPR0	0x0000	Timer Capture Register 0.
0x2C	R/W	TM_CAPR1	0x0000	Timer Capture Register 1.
0x30	R/W	TM_OCR	0x0000	Timer Output Compare Register.
0x34	R/W	TM_PWMC	0x0000	Timer PWM Control
0x38	R/W	TM_DBG	0x0000	Timer Debug Halt

Register Description

TM_CTRL
Address offset: 00h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:3	-	Reserved	-	Unused.
2	R/W	SRC	0	Timer clock source 0: PCLK 1: LXTAL (32,768Hz)
1	R/W	CRST	0	Counter reset. 0: Release the Timer Counter and Prescale Counter from reset. 1: The Timer Counter and Prescale Counter are synchronously reset on the next active clock edge. The counters remain reset until this bit is returned to 0.
0	0	CEN	0	Counter enable. 0: The counters are disabled. 1: The Timer Counter and Prescale Counter are enabled for counting.

TM_IF
Address offset: 04h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:5	-	Reserved	-	Unused.
4	W0C	CAP1_IF	0	Status flag for capture channel 1. It is set by hardware when the capture channel 1 event triggers; Software write 0 to clear it. When this flag is set with TM_CAPCR.CAP1_IE =1, tmirq is asserted to trigger an interrupt to the CPU.
3	W0C	CAP0_IF	0	Status flag for capture channel 0. It is set by hardware when the capture channel 0 event triggers; software write 0 to clear it. When this flag is set with TM_CAPCR.CAP0_IE =1, tmirq is asserted to trigger an interrupt to the CPU.
2	W0C	COMR2_IF	0	Status flag for compare channel 2. It is set by hardware when the COMR2 matches the TC; software writes 0 to clear it. When this flag is set with TM_COMCR.CR2_IE=1, tmirq is asserted to trigger and interrupt to the CPU.
1	W0C	COMR1_IF	0	Status flag for compare channel 1. It is set by hardware when the COMR1 matches the TC; software writes 0 to clear it. When this flag is set with TM_COMCR.CR1_IE=1, tmirq is asserted to trigger and interrupt to the CPU.
0	W0C	COMR0_IF	0	Status flag for compare channel 0. It is set by hardware when the COMR0 matches the TC; software writes 0 to clear it. When this flag is set with TM_COMCR.CR0_IE=1, tmirq is asserted to trigger and interrupt to the CPU.

TM_CNT
Address offset: 08h Access: Read-only

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
15:0	R	TC	0x0000	Timer Counter value. TC is incremented when the Prescale Counter reaches its terminal count. Unless it is reset before reaching its upper limit, the TC will count up to the value 65535 and then wrap back to 0.

TM_PR
Address offset: 0Ch Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
15:0	R/W	PCVAL	0x0000	Prescale terminal count value.

TM_PS
Address offset: 10h Access: Read-only

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
15:0	R	PC	0x0000	Prescale Counter value. PC is incremented on each active PCLK or LXTAL edge. It is reset to 0 when it matches the terminal count value stored in TM_PR.

TM_COMCR
Address offset: 14h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:9	-	Reserved	-	Unused.
8	R/W	CR2_STP	0	Stop on COMR2; the TC and PC will be stopped and TM_CTRL.CEN will be set to 0 if COMR1 matches the TC. 0: disabled. 1: enabled.
7	R/W	CR2_RST	0	Reset on COMR2; the TC will reset if COMR2 matches it. 0: disabled. 1: enabled.
6	R/W	CR2_IE	0	Interrupt on COMR2; an interrupt is generated when COMR2 matches the value in the TC. 0: disabled. 1: enabled.
5	R/W	CR1_STP	0	Stop on COMR1; the TC and PC will be stopped and TM_CTRL.CEN will be set to 0 if COMR1 matches the TC. 0: disabled. 1: enabled.
4	R/W	CR1_RST	0	Reset on COMR1; the TC will reset if COMR1 matches it. 0: disabled. 1: enabled.
3	R/W	CR1_IE	0	Interrupt on COMR1; an interrupt is generated when COMR1 matches the value in the TC. 0: disabled.

				1: enabled.
2	R/W	CR0_STP	0	Stop on CCOMR0; the TC and PC will be stopped and TM_CTRL.CEN will be set to 0 if COMR0 matches the TC. 0: disabled. 1: enabled.
1	R/W	CR0_RST	0	Reset on COMR0; the TC will reset if COMR0 matches it. 0: disabled. 1: enabled.
0	R/W	CR0_IE	0	Interrupt on COMR0; an interrupt is generated when COMCR0 matches the value in the TC. 0: disabled. 1: enabled.

TM_COMR0
Address offset: 18h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
15:0	R/W	COMPVAL0	0xFFFF	Timer Counter compare value for channel 0

TM_COMR1
Address offset: 1Ch Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
15:0	R/W	COMPVAL1	0xFFFF	Timer Counter compare value for channel 1

TM_COMR2
Address offset: 20h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
15:0	R/W	COMPVAL2	0xFFFF	Timer Counter compare value for channel 2

TM_CAPCR
Address offset: 24h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
5	R/W	CAP1_IE	0	Interrupt on capture events. A CAPR1 load due to CAP1_RE or CAP1_FE events will generate an interrupt. 0: disabled. 1: enabled.
4	R/W	CAP1_FE	0	Capture on TM_CAP1 falling edge. CAPR1 will be loaded with the contents of TC. 0: disabled. 1: enabled.
3	R/W	CAP1_RE	0	Capture on TM_CAP1 rising edge. CAPR1 will be loaded with the contents of TC. 0: disabled. 1: enabled.
2	R/W	CAP0_IE	0	Interrupt on capture events. A CAPR0 load due to

				CAP0_RE or CAP0_FE events will generate an interrupt. 0: disabled. 1: enabled.
1	R/W	CAP0_FE	0	Capture on TM_CPA0 falling edge. CAPR0 will be loaded with the contents of TC. 0: disabled. 1: enabled.
0	R/W	CAP0_RE	0	Capture on TM_CAP0 rising edge. CAPR0 will be loaded with the contents of TC. 0: disabled. 1: enabled.

Note: The edges are sampled by the system clock. Therefore, the adjacent edges must be more than one system clock period.

TM_CAPR0
Address offset: 28h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
15:0	RO	CAPVAL0	0x0000	Timer Counter capture value for channel 0

TM_CAPR1
Address offset: 2Ch Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:16	-	Reserved	-	Unused.
15:0	RO	CAPVAL1	0x0000	Timer Counter capture value channel 1

TM_OCR
Address offset: 30h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:9	-	Reserved	-	Unused.
8:7	R/W	OCC2	00	Output Compare Control 1. 00: OC2 does not change according to compare channel 2 01: Clear the OC2 bit when compare channel 2 matches. 10: Set the OC2 bit when compare channel 2 matches 11: Toggle the OC2 bit when compare channel 2 matches
6	R/W	OC2	0	This bit can be set or cleared by software and hardware. Software writes 0 to clear and writes 1 to set. Hardware set or clear access to this bit is defined by OCC2. Software takes precedence over the hardware when both of them write to this bit at the same time.
5:4	R/W	OCC1	00	Output Compare Control 1. 00: OC1 does not change according to compare channel 1 01: Clear the OC1 bit when compare channel 1 matches. 10: Set the OC1 bit when compare channel 1 matches 11: Toggle the OC1 bit when compare channel 1

				matches
3	R/W	OC1	0	This bit can be set or cleared by software and hardware. Software writes 0 to clear and writes 1 to set. Hardware set or clear access to this bit is defined by OCC1. Software takes precedence over the hardware when both of them write to this bit at the same time. The level of TM_COM1 output is the same as this bit.
2:1	R/W	OCC0	00	Output Compare Control 0 00: OC0 does not change according to compare channel 0 01: Clear the OC0 bit when compare channel 0 matches. 10: Set the OC0 bit when compare channel 0 matches 11: Toggle the OC0 bit when compare channel 0 matches
0	R/W	OC0	0	This bit can be set or cleared by software and hardware. Software writes 0 to clear and writes 1 to set. Hardware set or clear access to this bit is defined by OCC0. Software takes precedence over the hardware when both of them write to this bit at the same time. The level of TM_COM0 output is the same as this bit.

TM_PWMC
Address offset: 34h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:3	-	Reserved	-	Unused.
1	R/W	PWMEN1	0	0: TM_COM1 is controlled by OC1 1: PWM mode enable for TM_COM1. The LOW duration is given by COMR1. The period is given by COMR2.
0	R/W	PWMEN0	0	0: TM_COM0 is controlled by OC0 1: PWM mode enable for TM_COM0. The LOW duration is given by COMR0. The period is given by COMR2.

In PWM mode, the PWM period is given by COMR2. The CR2_RST bit is set to 1 to reset the TC when it matches the COMR2.

TM_DBG
Address offset: 34h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:1	-	Reserved	-	Unused.
0	R/W	DBG_HALT	0	0: Timer counting continues when M0 is halted. 1: Timer counting is halted when M0 is halted.

4.8. SPI

4.8.1 SPI Features

- Master mode only

- Data can be sent MSB or LSB first
- Support 1, 2, 3, and 4 bytes transfer counts
- 8 programmable clock rates

4.8.2 SPI IO Pins

Signal	IO	Source	Dest'n	Function
AMBA APB				
preset_n	DI	APB Bridge	SPI	APB active low reset
pclk	DI	APB Bridge	SPI	APB clock. The rising edge of pclk times all transfer on APB
pclkg	DI	APB Bridge	SPI	APB gated clock
psel	DI	APB Bridge	SPI	APB slave select
penable	DI	APB Bridge	SPI	APB enable strobe
pwrite	DI	APB Bridge	SPI	APB Write/Read strobe
paddr[3:2]	DI	APB Bridge	SPI	APB address bus
pwdata[31:0]	DI	APB Bridge	SPI	APB write data
prdata[31:0]	DO	SPI	APB Bridge	APB read data
pready	DO	SPI	APB Bridge	APB ready
Interrupt				
spiirq	DO	SPI	M0	SPI warning interrupt
IO				
MOSI	DO	SPI	Pad	Master-Out-Slave-In
MISO	DI	Pad	SPI	Master-In-Slave-Out
SCK	DO	SPI	Pad	SPI clock to slave
SS	DO	SPI	Pad	Slave select

4.8.3 SPI Registers

Register Overview

Address offset	R/W	Register Name	Reset value	Description
0x00	R/W	SPI_DATA	0x--	Data register
0x04	R/W	SPI_STATE	0x0	Status register
0x08	R/W	SPI_CTRL	0x00	Control register
0x0C	R/W	SPI_CRSEL	0x0	Clock rate select register

Register Description

SPI_DATA
Address offset: 00h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:0	R/W	DATA	0	Data value. Read – Received data Write – Transmit data

SPI_STATE
Address offset: 04h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:1	-	Reserved	-	Unused.

1	RO	SP_TIP	0	Transmission in progress flag. When data is written to the SPI_DATA with the SPI state machine in idle state, the serial transmission starts and this bit is set. It is cleared when the SPI state machine returns to idle state after the data is transmitted completely.
0	W1C	SP_IF	0	Serial peripheral interrupt flag. Set after completion of a transmission. If SPI_EN is set, an interrupt is generated. Write 1 to clear.

SPI_CTRL
Address offset: 08h Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:8	-	Reserved	-	Unused.
7	R/W	SS	0	There is 1 SS line that controls SS If bit n is cleared, SPI does not control the SS line. If bit n is set, SS is driven by hardware to the appropriate level during a transfer. SS is asserted LOW for one SCK period before the first active edge. It is de-asserted one SCK period after SP_IF is asserted.
6	R/W	FSB	0	First significant bit. MSB or LSB first transfer 0: Data is sent MSB first 1: Data is sent LSB first
5:4	R/W	BC	00	Byte count for one transfer. 00: one byte 01: two bytes 10: three bytes 11: four bytes
3	R/W	SPI_IE	0	Serial peripheral interrupt enable. 0: disabled. 1: enabled.
2	R/W	SPI_EN	0	Serial peripheral enable. 0: disable. 1: enable.
1	R/W	CPOL	0	Clock polarity.
0	R/W	CPHA	0	Clock phase.

SPI_CRSEL
Address offset: 0Ch Access: Read/Write

Bits	R/W	Field Name	Reset value	Description
31:3	-	Reserved	-	Unused.
[2:0]	R/W	CR	000	SCK rate = PCLK / $2^{(CR+1)}$. 000: 2 001: 4 010: 8 011: 16 100: 32 101: 64 110: 128 111: 256

4.9. Watch Dog Timer (WDT)

4.9.1 WDT Features

- APB interface
- 32-bit down-counter running on PCLK.
- Can be halted in debug mode
- Can trigger interrupt or reset upon timeout

4.9.2 WDT IO Pins

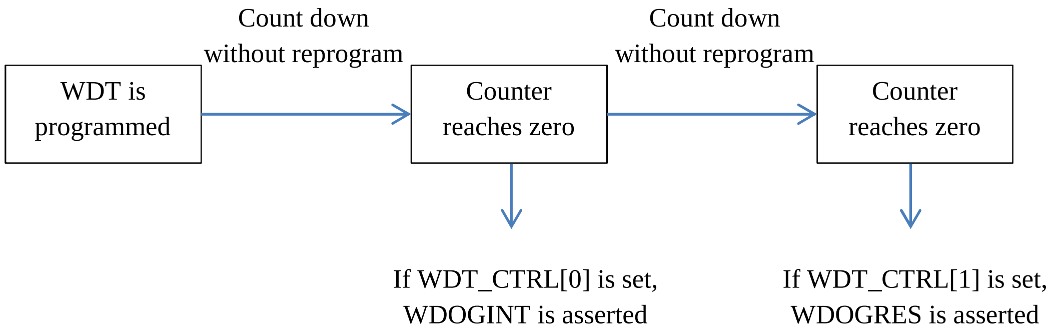
Signal	IO	Source	Dest'n	Function
AMBA APB				
PRESETn	DI	APB Bridge	WDT	APB active low reset
PCLK	DI	APB Bridge	WDT	APB clock. The rising edge of pclk times all transfer on APB
PCLKG	DI	APB Bridge	WDT	APB gated clock
PSEL	DI	APB Bridge	WDT	APB slave select
PENABLE	DI	APB Bridge	WDT	APB enable strobe
PWRITE	DI	APB Bridge	WDT	APB Write/Read strobe
PADDR[5:2]	DI	APB Bridge	WDT	APB address bus
PWDATA[31:0]	DI	APB Bridge	WDT	APB write data
PRDATA[31:0]	DO	WDT	APB Bridge	APB read data
PREADY	DO	WDT	APB Bridge	APB ready
Interrupt				
WDOGINT	DO	WDT	M0	WDT warning interrupt
System Control				
WDOGRES	DO	WDT	SYS_CTRL	WDT active high timeout reset to system
Debug				
DBG_HALT	DI	M0	WDT	M0 halted

4.9.3 WDT Operations

The WDT is based on a 32-bit down-counter. The counter is initialized from the WDT_LOAD register. The WDT generates a regular interrupt, WDOGINT, depending on the programmed value. The counter decrements by one on each positive clock edge of PCLK.

After the WDT count-down is enabled, it shall assert an interrupt WDOGINT signal when the counter reaches 0. The counter is then reloaded from the WDT_LOAD register and the count-down sequence continues. If the interrupt is not cleared by the time the counter reaches the next 0, the WDT shall assert the WDOGRES signal to request a system reset.

The next figure illustrates the flow diagram for the WDT operations.


WDT is programmed
Count down without reprogram
Counter reaches zero
Count down without reprogram
Counter reaches zero
If WDT_CTRL[0] is set, WDOGINT is asserted
If WDT_CTRL[1] is set, WDOGRES is asserted

4.9.4 WDT Registers

Register Overview

Address offset	R/W	Register Name	Reset value	Description
0x00	R/W	WDT_LOAD	0xFFFFFFFF	Load value register
0x04	RO	WDT_VAL	0xFFFFFFFF	Current value register
0x08	R/W	WDT_CTRL	0x0	Control register
0x0C	WO	WDT_INTCLR	-	Interrupt clear register
0x10	RO	WDT_RIS	0x0	Raw interrupt register
0x14	R/W	WDT_LOCK	0x0	Lock register
0x18	R/W	WDT_DBG	0x0	Debug halt register

Register Description

WDT_LOAD
Address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
[31:0]	R/W	LOAD	0xFFFFFFFF	Contains the 32-bit value from which the counter is to decrement. When this register is written to, the count is immediate restarted from the new value. The minimum valid value is 1.

WDT_VAL
Address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
[31:0]	R/W	VAL	0xFFFFFFFF	Contains the 32-bit value of the decrementing counter.

WDT_CTRL
Address offset: 0x08

Bits	R/W	Field Name	Reset value	Description
[31:2]	-	Reserved	-	Unused.
1	R/W	RESEN	0x0	Enable the WDOGRES. Set HIGH to enable reset, and set LOW to disable reset.
0	R/W	INTEN	0x0	Enable the interrupt WDOGINT. Set HIGH to enable the interrupt and the counter, and set low to

				disable the interrupt and the counter. Reload the counter from the value in WDT_LOAD when it is changed from LOW to HIGH.

WDT_INTCLR
Address offset: 0x0C

Bits	R/W	Field Name	Reset value	Description
[31:1]	-	Reserved	-	Unused.
0	WO	INTCLR	-	A write of any value to this register clears the interrupt and reloads the counter from the value in WDT_LOAD.

WDT_RIS
Address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
[31:1]	-	Reserved	-	Unused
0	RO	RIS	0x0	Raw interrupt status. It is set HIGH when the counter reaches 0 and INTEN is HIGH. The WDOGINT outputs the value of this bit.

WDT_LOCK
Address offset: 0x14

Bits	R/W	Field Name	Reset value	Description
[31:0]	R/W	LOCK	0x0	Write 0x1ACCE551 to unlock write access to all other registers. Write any other value to lock the write access. A read from this register always returns the bit 0. 0: write access is not locked. 1: write access is locked.

WDT_DBG
Address offset: 0x18

Bits	R/W	Field Name	Reset value	Description
[31:1]	-	Reserved	-	Unused.
0	R/W	DBG_HALT	0x0	0: counter is not stopped when M0 is halted. 1: counter is stopped when M0 is halted.

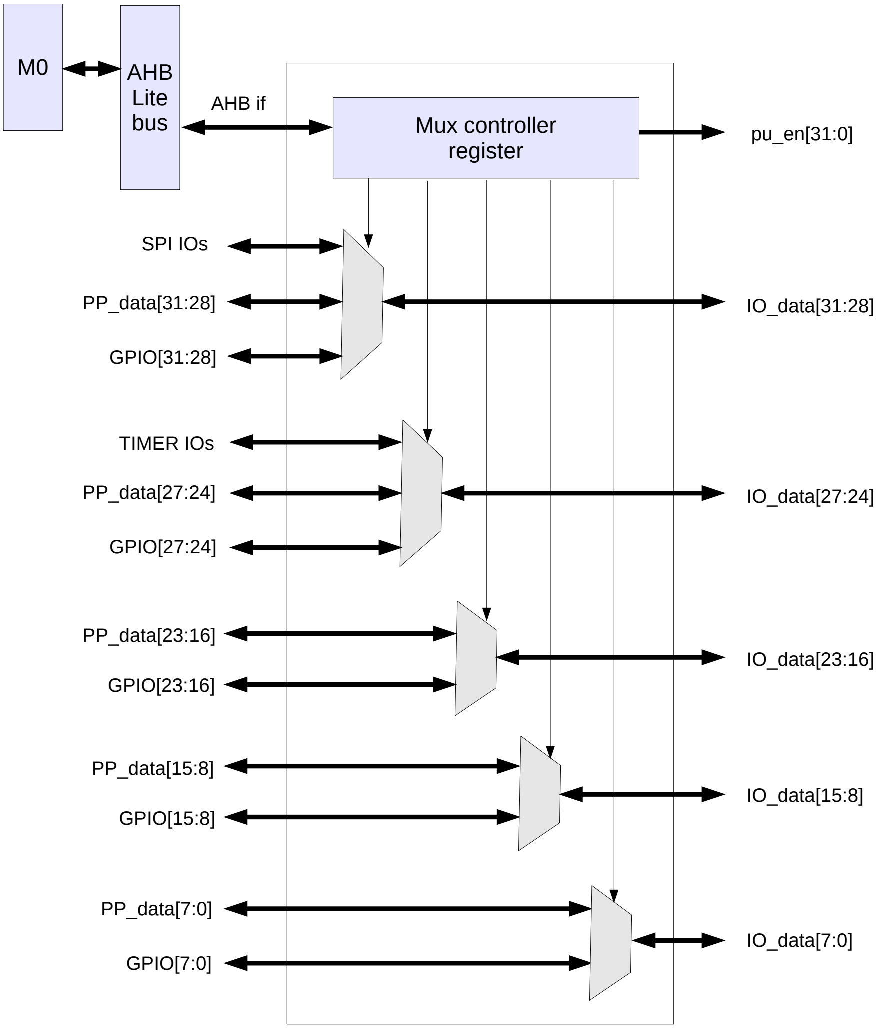
4.10. Mux Controller

Mux controller module controls the selection of the usage of the IO pins. IO-control core has 32 IO pins. The pins are shared for parallel port, GPIO, SPI and timer. The M0 processor can set the IO configuration by writing to the mux controller register. The mux controller also controls the pull-ups on the IO pins.

4.10.1 Mux Controller Features

- Provide control to 32 IO pins to enable IOs for parallel port, GPIO, SPI and timer
- Provide control to the pull up on the IO pins

4.10.2 Mux Controller Connection Diagram



4.10.3 Mux Controller IO Pins

Signal	IO	Source	Dest’ n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Controller	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Controller	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Controller	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Controller	Indicates the size of transfer, that is typically byte, halfword or word

HTRANS[1:0]	DI	AHB Lite Bus	Controller	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Controller	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Controller	AHB ready input
HRDATA[31:0]	DO	Controller	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Controller	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Controller	AHB Lite Bus	Indicates the transfer response
INTERNAL IOs				
pp_dataout[31:0]	DI	Parallel port	Controller	Parallel port data out
pp_datain [31:0]	DI	Parallel port	Controller	Parallel port output enable
pp_oe_n [31:0]	DO	Controller	Parallel port	Parallel port data in
GPIOOUT[31:0]	DI	GPIO	Pin mux	GPIO out
GPIOOEN[31:0]	DI	GPIO	Pin mux	GPIO output enable
GPIOIN[31:0]	DO	Controller	GPIO	GPIO in
TM_CAP0	DO	Controller	Timer	Timer capture
TM_CAP1	DO	Controller	Timer	Timer capture
TM_COM0	DI	Timer	Controller	Timer compare
TM_COM1	DI	Timer	Controller	Timer compare
sck	DI	SPI	Controller	SPI clock
mosi	DI	SPI	Controller	SPI master out slave in
ss_n	DI	SPI	Controller	SPI slave select
miso	DO	Controller	SPI	SPI master in slave out
EXTERNAL IOs				
io_data_out[31:0]	DO	controller	IO pin	Multiplexed data out
io_data_out_en[31:0]	DO	controller	IO pin	Multiplexed output enable
io_data_in[31:0]	DI	IO pin	controller	Multiplexed data in

4.10.4 Mux Controller Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	R/W	MUXC_SELECT	0x00000000	mux select
0x04	R/W	MUXC_PU	0x11111111	pull up enable

Register Description

MUX_SELECT
address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:7	R	RESERVED	0	unused
6:5	R/W	MUXC_PP_SPI	0	SPI IOs select. 3 => reserved 2 => SPI IOs are selected 1 => pp_data [31:28] are selected 0 => GPIO [31:28] are selected
4:3	R/W	MUXC_PP_TIMER	0	TIMER IOs select. 3 => reserved 2 => TIMER IOs are selected 1 => pp_data [27:24] are selected 0 => GPIO [27:24] are selected
2	R/W	MUXC_PP2	0	pp_data[23:16] select. 1 => pp_data[23:16] are selected 0 => GPIO[23:16] are selected
1	R/W	MUXC_PP1	0	pp_data[15:8] select. 1 => pp_data[15:8] are selected 0 => GPIO[15:8] are selected
0	R/W	MUXC_PP0	0	PP_data[7:0] select. 1 => pp_data[7:0] are selected 0 => GPIO[7:0] are selected

MUXC_PU
address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31	R/W	MUXC_PU31	1	pull up for IO[31] enable
…	R/W	…	1	…
1	R/W	MUXC_PU1	1	pull up for IO[1] enable
0	R/W	MUXC_PU0	1	pull up for IO[0] enable, set to 1 to enable

4.11. Debug Register

Debug Register Module provide mechanism to perform software-controlled debug. This module implements four debug registers to store user defined data and has access to the M0 internal registers. M0 processor can start a debug event by writing to the control register of this debug register module. The debug register then generate NMI interrupt to put the core into ISR. At the same time, the debug registers in the normal cores also generate the NMI interrupt. Furthermore, the HREADY to the M0 processors are pulled low to completely halt the cores. The data from the debug registers of the normal cores are transferred to the AHB to NoC bridge of the IO-control core (refer to section 7). The dedicated hardware debugger circuit then take over the AHB-lite bus of the IO-control core and start to pull out the data from this debug register and the AHB to NoC bridge.

4.11.1 Debug Register Features

- Four 32-bit user defined debug registers
- Access to internal M0 registers
- Support practically unlimited software-controlled debug break point

4.11.2 Debug Register IO Pins

Signal	IO	Source	Dest’n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Debug reg	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Debug reg	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Debug reg	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Debug reg	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Debug reg	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Debug reg	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Debug reg	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Debug reg	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Debug reg	AHB ready input
HRDATA[31:0]	DO	Debug reg	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Debug reg	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Debug reg	AHB Lite Bus	Indicates the transfer response
M0 REGISTERS				
cm0_r00[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r01[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r02[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r03[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r04[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r05[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r06[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r07[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r08[31:0]	DI	M0	Debug reg	M0 internal register

cm0_r09[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r10[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r11[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r12[31:0]	DI	M0	Debug reg	M0 internal register
cm0_msp[31:0]	DI	M0	Debug reg	M0 internal register
cm0_psp[31:0]	DI	M0	Debug reg	M0 internal register
cm0_r14[31:0]	DI	M0	Debug reg	M0 internal register
cm0_pc[31:0]	DI	M0	Debug reg	M0 internal register
cm0_xpsr[31:0]	DI	M0	Debug reg	M0 internal register
cm0_control[31:0]	DI	M0	Debug reg	M0 internal register
cm0_primask[31:0]	DI	M0	Debug reg	M0 internal register
DEBUG CONTROLLER				
debug_request	DO	Debug reg	Debug ctrl	Debug event request to the debug ctrl
debug_granted	DI	Debug ctrl	Debug reg	Debug event request grant signal
debug_done	DI	Debug ctrl	Debug reg	Debug event done, all data latched out
INTERRUPT				
DEBUG_INT	DO	Debug reg	M0	NMI debug interrupt

4.11.3 Debug Register Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	W	DEBUG_START	0x00000000	debug event start register
0x04	R/W	DEBUG_REG0	0x00000000	debug register 0
0x08	R/W	DEBUG_REG1	0x00000000	debug register 1
0x0C	R/W	DEBUG_REG2	0x00000000	debug register 2
0x10	R/W	DEBUG_REG3	0x00000000	debug register 3
0x14	R	DEBUG_CM0_R00	0x00000000	M0 internal register
0x18	R	DEBUG_CM0_R01	0x00000000	M0 internal register
0x1C	R	DEBUG_CM0_R02	0x00000000	M0 internal register
0x20	R	DEBUG_CM0_R03	0x00000000	M0 internal register
0x24	R	DEBUG_CM0_R04	0x00000000	M0 internal register
0x28	R	DEBUG_CM0_R05	0x00000000	M0 internal register
0x2C	R	DEBUG_CM0_R06	0x00000000	M0 internal register
0x30	R	DEBUG_CM0_R07	0x00000000	M0 internal register
0x34	R	DEBUG_CM0_R08	0x00000000	M0 internal register
0x38	R	DEBUG_CM0_R09	0x00000000	M0 internal register
0x3C	R	DEBUG_CM0_R10	0x00000000	M0 internal register
0x40	R	DEBUG_CM0_R11	0x00000000	M0 internal register
0x44	R	DEBUG_CM0_R12	0x00000000	M0 internal register

0x48	R	DEBUG_CM0_MSP	0x00000000	M0 internal register
0x4C	R	DEBUG_CM0_PSP	0x00000000	M0 internal register
0x50	R	DEBUG_CM0_R14	0x00000000	M0 internal register
0x54	R	DEBUG_CM0_PC	0x00000000	M0 internal register
0x58	R	DEBUG_CM0_XPSR	0x00000000	M0 internal register
0x5C	R	DEBUG_CM0_CONTROL	0x00000000	M0 internal register
0x60	R	DEBUG_CM0_PRIMASK	0x00000000	M0 internal register

Register Description

DEBUG_START
address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:1	R	RESERVED	0	unused
0	W	DEBUG_START_F	0	Debug event start flag. Write 1 to this bit will trigger the debug event.

DEBUG_REGn
address offset: 0x04 - 0x10

Bits	R/W	Field Name	Reset value	Description
31	R/W	-	0	User defined debug register

DEBUG_CM0_xxxx
address offset: 0x14 - 0x60

Bits	R/W	Field Name	Reset value	Description
31	R	-	0	M0 internal register

4.12. AHB to NoC Bridge

AHB2NOC (AHB to NoC) is an AHB slave module that act as a bridge that connect AHB master device with the NoC interconnect. The module is able to transmit and receive flit (NoC packet) through the network. One transmit buffer and multiple receive buffers are implemented in the module to contain the flit that is going to be transmitted and the received flit. Although there are multiple transmit buffers that are exist in the register list of the AHB2NOC register table (refer to section 4.11.3), there are only one physical transmit buffer. The depth of the transmit buffer is four. This multiple virtual transmit buffers are implemented as the mechanism to select the destination of the flit to be sent. Write to a specific virtual transmit buffer will trigger the module to perform NoC transfer to correspondent router. The number of the receive buffers are the same as the number of routers in the network. Each receive buffer is used to save the flits that are transmitted from a specific router to this AHB2NOC module. Each receive buffer has eight buffer depth, except the one that is used for the loopback only has two depth. The receive buffer that handle the loopback transfer is the buffer which has the index that is the same as the network address of the core (refer to section 7). The IO-control core's receive buffer index 0 is it loopback buffer, while the Normal core 0 loopback buffer is the receive buffer index 1, and so on. With this mechanism, the AHB master device can recognize the source of the flit that is received by the AHB2NOC. Two interrupt line, NOC_TX_INT and NOC_RX_INT are provided. The NOC_TX_INT line will be pulled high if there are finished transmit done by the module. Meanwhile the NOC_RX_INT signal can be generated when there is a flit received or after a number of flits received in a specific buffer. The number of flit to be received can be configured. One additional interrupt, NOC_BD_INT is used for bootloading process.

4.12.1 AHB to NoC Bridge Features

- Transmit and Receive 32-bit data in the form of flit through NoC
- Address decoding mechanism to set the destination for the flit
- Transmit buffer with the depth of four
- Multiple receive buffers to contains flits from different sources, each buffer has eight buffer depth except for the loopback buffer only has two depth
- Three interrupt signals generation for transmit, receive and bootloading event
- Two interrupt generation scheme
- Provide data and status registers for bootloading-debugging mechanism

4.12.2 AHB to NoC Bridge IO Pins

Signal	IO	Source	Dest'n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Bridge	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Bridge	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Bridge	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Bridge	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Bridge	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Bridge	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Bridge	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Bridge	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Bridge	AHB ready input

HRDATA[31:0]	DO	Bridge	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Bridge	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Bridge	AHB Lite Bus	Indicates the transfer response
NOC				
EN_putFlit	DO	Bridge	NoC	Enable the transmit of flit
putFlit[37:0]	DO	Bridge	NoC	The flit to be transmitted
EN_getCredits	DO	Bridge	NoC	Enable the receive of credit
getCredits[1:0]	DI	NoC	Bridge	The credit to be received
EN_getFlit	DO	Bridge	NoC	Enable the receive of flit
getFlit[37:0]	DI	NoC	Bridge	The flit to be received
EN_putCredits	DO	Bridge	NoC	Enable the transmit of credit
putCredits[1:0]	DO	Bridge	NoC	The credit to be transmitted
Router_id[1:0]	DI	NoC	Bridge	Attached router index
SYSTEM CONTROL				
mode	DI	Sys ctrl	Bridge	Bridge operating mode
DEBUGGER				
debug_data_ready	DO	Bridge	Debugger	Debug data ready in the buffers
INTERRUPT				
NOC_TX_INT	DO	Bridge	M0	Transmit interrupt
NOC_RX_INT	DO	Bridge	M0	Receive interrupt
NOC_BD_INT	DO	Bridge	M0	Bootloader interrupt

4.12.3 AHB to NoC Bridge Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	R	NOC_CSR0	0x00000000	control and status register 0
0x04	R/W	NOC_CSR1	0x00000000	control and status register 1
0x08	R/W	NOC_CSR2	0x0000ffff	control and status register 2
0x0C	R	NOC_RX_BUFF0	0x00000000	receive buffer 0
0x10	R	NOC_RX_BUFF1	0x00000000	receive buffer 1
0x14	R	NOC_RX_BUFF2	0x00000000	receive buffer 2
0x18	R	NOC_RX_BUFF3	0x00000000	receive buffer 3
0x1C	R/W	NOC_TX_BUFF0	0x00000000	transmit buffer 0
0x20	R/W	NOC_TX_BUFF1	0x00000000	transmit buffer 1
0x24	R/W	NOC_TX_BUFF2	0x00000000	transmit buffer 2
0x28	R/W	NOC_TX_BUFF3	0x00000000	transmit buffer 3
0x2C	R	NOC_RX_BD_BUFF0	0x00000000	bootloader/debugger receive buffer 0

0x30	R	NOC_RX_BD_BUFF1	0x00000000	bootloader/debugger receive buffer 1
0x34	R	NOC_RX_BD_BUFF2	0x00000000	bootloader/debugger receive buffer 2
0x38	R	NOC_RX_BD_BUFF3	0x00000000	bootloader/debugger receive buffer 3
0x3C	R/W	NOC_TX_BD_BUFF0	0x00000000	bootloader/debugger transmit buffer 0
0x40	R/W	NOC_TX_BD_BUFF1	0x00000000	bootloader/debugger transmit buffer 1
0x44	R/W	NOC_TX_BD_BUFF2	0x00000000	bootloader/debugger transmit buffer 2
0x48	R/W	NOC_TX_BD_BUFF3	0x00000000	bootloader/debugger transmit buffer 3
0x4C	R/W	NOC_BD_CSR	0x00000000	bootloader/debugger control and status register

Register Description

NOC_CSR0

address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7	R	NOC_RXF3	0	see NOC_RXF0
6	R	NOC_RX3	0	see NOC_RX0
5	R	NOC_RXF2	0	see NOC_RXF0
4	R	NOC_RX2	0	see NOC_RX0
3	R	NOC_RXF1	0	see NOC_RXF0
2	R	NOC_RX1	0	see NOC_RX0
1	R	NOC_RXF0	0	Receive buffer full flag; 0 => receive buffer 0 still have empty space; 1 => receive buffer full
0	R	NOC_RX0	0	Receive flit flag; 0 => no received flit, 1 => there is a received flit in NOC_RX_BUFF0. A read to NOC_RX_BUFF will discard the data from the buffer. If there is no more data in the buffer, NOC_RX will be cleared by hardware, if there is another data, the NOC_RX will remain high.

NOC_CSR1

address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31	W	NOC_BDIE_CLR	0	Write 1 to this bit to clr NOC_BDIE, write 0 will take no effect.
30	W	NOC_BDIE_SET	0	Write 1 to this bit to set NOC_BDIE, write 0 will take no effect.
29	W	NOC_RIM_CLR	0	Write 1 to this bit to clr NOC_RIM, write 0 will take no effect.
28	W	NOC_RIM_SET	0	Write 1 to this bit to set NOC_RIM, write 0 will take no effect.

27	W	NOC_RIE_CLR	0	Write 1 to this bit to disable the rx interrupt, write 0 will take no effect.
26	W	NOC_RIE_SET	0	Write 1 to this bit to enable the rx interrupt, write 0 will take no effect.
25	W	NOC_TIE_CLR	0	Write 1 to this bit to disable the tx interrupt, write 0 will take no effect.
24	W	NOC_TIE_SET	0	Write 1 to this bit to enable the tx interrupt, write 0 will take no effect.
23:6	R	RESERVED	0	unused
5	R	NOC_BDIE	0	BD interrupt enable. 0 => interrupt disabled. 1 => interrupt enabled. NOC_BDIE_INT line will be high when there is ACK received.
4	R	NOC_RIM	0	receive interrupt mode. 0 => receive interrupt mode 0, interrupt is generated every time a flit received. 1 => receive interrupt mode 1, interrupt is generated when N number of flit arrived
3	R	NOC_RIE	0	receive interrupt enable. 0 => interrupt disabled. 1 => interrupt enabled. NOC_RX_INT line will be high when there is data in the RX_BUFFER
2	R	NOC_TIE	0	transmit complete interrupt enable. 0 => interrupt disabled. 1 => interrupt enabled. NOC_TX_INT line will be high when there is finished transfer. Write 1 to PP_TX will clear the interrupt.
1	R	NOC_TXF	0	Transmit buffer full flag; 0 => transmit buffer still have empty space; 1 => transmit buffer full, AHB master are not allowed to write into NOC_TX_BUFFn
0	RW	NOC_TX	0	transmit complete flag, set high when there were data written to TX_BUFF and the tx buffer is empty. Write 1 to this bit will clear the flag.

NOC_CSR2
Address offset: 0x08

Bits	R/W	Field Name	Reset value	Description
31:28	RW	NOC_RX3_COUNT	0	see NOC_RX0_COUNT
27:24	RW	NOC_RX2_COUNT	0	see NOC_RX0_COUNT
23:20	RW	NOC_RX1_COUNT	0	see NOC_RX0_COUNT
19:16	RW	NOC_RX0_COUNT	0	Receive flit counter, count the number of flits that are received and stored in the receive buffer 0. If the counter value is equal or more than the value of NOC_RX0_TRIG, each read to the receive buffer 0 reduces the counter value by 1. User can also over-write the value of this counter.
15:12	RW	NOC_RX3_TRIG	f	see NOC_RX0_TRIG
11:8	RW	NOC_RX2_TRIG	f	see NOC_RX0_TRIG

7:4	RW	NOC_RX1_TRIG	f	see NOC_RX0_TRIG
3:0	RW	NOC_RX0_TRIG	f	The number of flits to be received and stored in the receive buffer 0 before the interrupt is triggered. This value is only valid if the NOC_RIM is high.

NOC_RX_BUFFn

address offset: 0x0C - 0x18

Bits	R/W	Field Name	Reset value	Description
31:8	R	-	0	Receive buffer. Read from this buffer will discard one flit from buffer if the flit is available.

NOC_TX_BUFFn

address offset: 0x1C - 0x28

Bits	R/W	Field Name	Reset value	Description
31:8	W	-	0	Transmit buffer. write to this buffer will trigger transfer through NoC to the module that connected to router n.

NOC_RX_BD_BUFFn

address offset: 0x2C - 0x38

Bits	R/W	Field Name	Reset value	Description
31:8	R	-	0	Bootloader/Debugger (BD) Receive buffer. Read from this buffer will discard one flit from buffer if the BD flit is available.

NOC_TX_BD_BUFFn

address offset: 0x3C - 0x48

Bits	R/W	Field Name	Reset value	Description
31:8	W	-	0	Bootloader/Debbuger (BD) Transmit buffer. write to this buffer will trigger transfer through NoC to the module that connected to router n.

NOC_BD_CSR

address offset: 0x4C

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7	R	NOC_BD_ACK3	0	see NOC_BD_ACK0
6	R	NOC_BD_RX3	0	see NOC_BD_RX0
5	R	NOC_BD_ACK2	0	see NOC_BD_ACK0
4	R	NOC_BD_RX2	0	see NOC_BD_RX0
3	R	NOC_BD_ACK1	0	see NOC_BD_ACK0

2	R	NOC_BD_RX1	0	see NOC_BD_RX0
1	R	NOC_BD_ACK0	0	Receive ACK flag; 0 => no ACK; 1 => received ACK. Write 1 to clear this flag.
0	R	NOC_BD_RX0	0	Receive bd flit flag; 0 => no received bd flit, 1 => there is a received bd flit in NOC_RX_BD_BUFF.

5. Normal Core

There are two normal cores in the system. The normal cores has slightly less resources. However each of the normal cores is equipped by an AES accelerator module. The flash memory of the normal cores only support to be used as the storage of application codes. The two normal cores are identical. The only different that the cores are attached to different routers hence they are assigned with different network addresses.

5.1. Memory Mapping

The normal core has eight slave modules attached to the AHB-lite bus. The address of the slaves are:

Slave num	Slave	start address	end address
1	Flash	0x0000_0000	0x0000_83FF
2	SRAM	0x2000_0000	0x2000_1FFF
3	APB	0x4000_0000	0x4000_3FFF
4	GPIO	0x5000_0000	0x5000_FFFF
5	Mux controller	0x5001_0000	0x5001_FFFF
6	Debug	0x5002_0000	0x5002_FFFF
7	AES	0x5003_0000	0x5003_FFFF
8	NoC bridge	0x5004_0000	0x5004_FFFF

The address of the peripherals that are attached to APB bus are:

Slave num	Slave	start address	end address
1	APB Flash	0x4000_0000	0x4000_0FFF
2	APB Timer	0x4000_1000	0x4000_1FFF
3	APB SPI	0x4000_2000	0x4000_2FFF
3	APB WDT	0x4000_3000	0x4000_3FFF

5.2. Interrupt Vector Table

The highest priority interrupt (NMI) of the normal core is used for debug event (refer to section 7). Eight interrupt pins are used as peripheral interrupts. The interrupt vector table are:

no	interrupt	description
0	flash	flash APB interface interrupt
1	noc tx	NoC transmit interrupt
2	noc rx	NoC receive interrupt
3	aes	AES interrupt
4	timer	timer interrupt

5	spi	SPI interrupt
6	gpio0	GPIO interrupt 0
7	gpio1	GPIO interrupt 1
8	wdt	Watch Dog Timer interrupt

5.3. Flash Controller

Flash controller of the normal cores are identical to the flash controller of the IO-control core. Refer to section 4.3 for more details.

5.4. SRAM Controller

SRAM controller of the normal cores are identical to the SRAM controller of the IO-control core. Refer to section 4.4 for more details.

5.5. GPIO

GPIO module of the normal cores only control eight IO pins. The rest of the features are similar to the GPIO module of the IO-control core.

5.5.1 GPIO Features

- 8-bit I/O ports with pull up capability
- Level/edge-triggered interrupt generation
- Bit masking access to output register using address values.
- Atomic bitwise set, clear and toggle access to output register
- Thread safe separate set and clear addresses for control registers.

5.5.2 GPIO IO Pins

Signal	IO	Source	Dest’n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Controller	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Controller	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Controller	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Controller	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Controller	Indicates the transfer type of the current transfer

HWDATA[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Controller	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Controller	AHB ready input
HRDATA[31:0]	DO	Controller	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Controller	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Controller	AHB Lite Bus	Indicates the transfer response
GPIO				
GPIOIN [7:0]	DI	Pin mux	Controller	GPIO data in
GPIOOEN [7:0]	DO	Controller	Pin mux	GPIO output enable
GPIOOUT [7:0]	DO	Controller	Pin mux	GPIO data out
INTERRUPT				
GPIOINT[1:0]	DO	Controller	M0	Transmit interrupt

5.5.3 Interrupt generation

The GPIO inputs [1:0] are able to trigger interrupt to the core. IE, IT, IP and IF are all 2-bit registers.

IE[n]	IT[n]	IP[n]	Interrupt feature
0	-	-	Disabled
1	0	0	Low-level
1	0	1	High-level
1	1	0	Falling-edge
1	1	1	Rising-edge

5.5.4 Masked Access

Masked access allows bit-wise access in a single transfer. The address of the access is used as the mask for the read or write operations.

There are 256 word addresses, corresponding to all possible bit mask pattern for 8 bits.

The mask is from HADDR[9:2]. The base address of the bit mask address is at 0x0400.

Example 1:

Set bit 2 & 3, clear bit 5 & 7.

HADDR[9:2] = 0xAC (which is b10101100)

HADDR[12:10] = 0x1

HWDATA = 0x0C

5.5.5 GPIO Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	R	DATAIN	0x00000000	Data input value
0x04	R/W	DATAOUT	0x00000000	Data output register
0x08	R/W	OEN_SET	0x00000000	Output enable control
0x0C	R/W	OEN_CLR	0x00000000	Output enable control
0x10	W1C	IF	0x00000000	Interrupt flag
0x14	W1S	IE_SET	0x00000000	Interrupt enable control
0x18	W1C	IE_CLR	0x00000000	Interrupt enable control
0x1C	W1S	IT_SET	0x00000000	Interrupt type control
0x20	W1C	IT_CLR	0x00000000	Interrupt type control
0x24	W1S	IP_SET	0x00000000	Interrupt polarity control
0x28	W1C	IP_CLR	0x00000000	Interrupt polarity control
0x2C	W	BSET	Not defined	Bit-wise set the DATAOUT
0x30	W	BCLR	Not defined	Bit-wise clear the DATAOUT
0x34	W	BTGL	Not defined	Bit-wise toggle the DATAOUT

Register Description

DATAIN

address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	R	-	0	Data input value Read – sample at pin

DATAOUT

address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	R/W	-	0	Data output register Read – current value of data output register Write – to data output register

OEN_SET

address offset: 0x08

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	R	-	0	Output enable control. Write 0 – no effect

				Write 1 – set the OEN bit Read 0 – indicate the IO direction as input Read 1 – indicate the IO direction as output

OEN_CLR

address offset: 0x0C

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	R	-	0	Output enable control. Write 0 – no effect Write 1 – clear the OEN bit Read 0 – indicate the IO direction as input Read 1 – indicate the IO direction as output

IF

address offset: 0x10

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
1:0	R	-	0	Interrupt flag. Write 0 – no effect Write 1 – clear the interrupt flag Read 0 – no interrupt Read 1 – interrupt has triggered

IE_SET

address offset: 0x14

Bits	R/W	Field Name	Reset value	Description
31:2	R	RESERVED	0	unused
1:0	R/W	-	0	Interrupt enable control Write 0 – no effect Write 1 – set the IE bit Read 0 – interrupt is disabled Read 1 – interrupt is enabled

IE_CLR

address offset: 0x18

Bits	R/W	Field Name	Reset value	Description
31:2	R	RESERVED	0	unused
1:0	R/W	-	0	Interrupt enable control Write 0 – no effect Write 1 – clear the IE bit Read 0 – interrupt is disabled Read 1 – interrupt is enabled

IT_SET

address offset: 0x1C

Bits	R/W	Field Name	Reset value	Description
31:2	R	RESERVED	0	unused

1:0	R/W	-	0	Interrupt type control Write 0 – no effect Write 1 – set the IE bit Read 0 – level triggered Read 1 – edge triggered

IT_CLR

address offset: 0x20

Bits	R/W	Field Name	Reset value	Description
31:2	R	RESERVED	0	unused
1:0	R/W	-	0	Interrupt type control Write 0 – no effect Write 1 – clear the IE bit Read 0 – level triggered Read 1 – edge triggered

IP_SET

address offset: 0x24

Bits	R/W	Field Name	Reset value	Description
31:2	R	RESERVED	0	unused
1:0	R/W	-	0	Interrupt polarity control Write 0 – no effect Write 1 – set the IE bit Read 0 – low-level or falling-edge triggered Read 1 – high-level or rising-edge triggered

IP_CLR

address offset: 0x28

Bits	R/W	Field Name	Reset value	Description
31:2	R	RESERVED	0	unused
1:0	R/W	-	0	Interrupt polarity control Write 0 – no effect Write 1 – clear the IE bit Read 0 – low-level or falling-edge triggered Read 1 – high-level or rising-edge triggered

BSET

address offset: 0x2C

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	W	-	-	Bit-wise set the DATAOUT Write 0 – no effect Write 1 – set the DATAOUT bit

BCLR

address offset: 0x30

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	W	-	-	Bit-wise clear the DATAOUT Write 0 – no effect

				Write 1 – clear the DATAOUT bit

BTGL
address offset: 0x34

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	W	-	-	Bit-wise toggle the DATAOUT Write 0 – no effect Write 1 – toggle the DATAOUT bit

5.6. Timer

Timer module of the normal cores are identical to the time module of the IO-control core. Refer to section 4.7 for more details.

5.7. SPI

SPI module of the normal cores are identical to the SPI module of the IO-control core. Refer to section 4.8 for more details.

5.8. Watch Dog Timer (WDT)

WDT module of the normal cores are identical to the WDT module of the IO-control core. Refer to section 4.9 for more details.

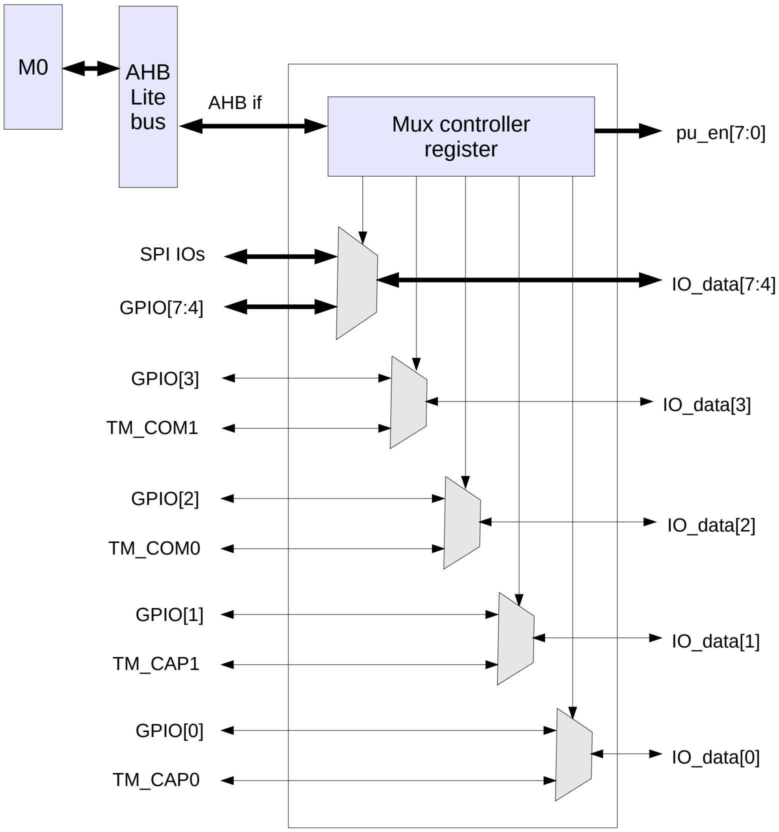
5.9. Mux Controller

Normal cores have only 8 IO pins. The pins are shared for GPIO, SPI and timer. The mux controller in the normal cores also controls the pull-ups on the IO pins.

5.9.1 Mux Controller Features

- Provide control to 8 IO pins to enable IOs for GPIO, SPI and timer
- Provide control to the pull up on the IO pins

5.9.2 Mux Controller Connection Diagram



5.9.3 Mux Controller IO Pins

Signal	IO	Source	Dest’ n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Controller	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Controller	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Controller	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Controller	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Controller	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Controller	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Controller	Indicates the transfer direction

HREADY	DI	AHB Lite Bus	Controller	AHB ready input
HRDATA[31:0]	DO	Controller	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Controller	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Controller	AHB Lite Bus	Indicates the transfer response
INTERNAL IOs				
GPIOOUT[7:0]	DI	GPIO	Pin mux	GPIO out
GPIOOEN[7:0]	DI	GPIO	Pin mux	GPIO output enable
GPIOIN[7:0]	DO	Controller	GPIO	GPIO in
TM_CAP0	DO	Controller	Timer	Timer capture
TM_CAP1	DO	Controller	Timer	Timer capture
TM_COM0	DI	Timer	Controller	Timer compare
TM_COM1	DI	Timer	Controller	Timer compare
sck	DI	SPI	Controller	SPI clock
mosi	DI	SPI	Controller	SPI master out slave in
ss_n	DI	SPI	Controller	SPI slave select
miso	DO	Controller	SPI	SPI master in slave out
EXTERNAL IOs				
io_data_out[7:0]	DO	controller	IO pin	Multiplexed data out
io_data_out_en[7:0]	DO	controller	IO pin	Multiplexed output enable
io_data_in[7:0]	DI	IO pin	controller	Multiplexed data in

5.9.4 Mux Controller Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	R/W	MUXC_SELECT	0x00000000	mux select
0x04	R/W	MUXC_PU	0x000000ff	pull up enable

Register Description

MUX_SELECT
address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:5	R	RESERVED	0	unused
4	R/W	MUXC_SPI	0	SPI IOs select. 1 => SPI IOs are selected 0 => GPIO [7:4] are selected

3	R/W	MUXC_COM1	0	TIMER IO, TM_COM1 select. 1 => TM_COM1 is selected 0 => GPIO [3] is selected
2	R/W	MUXC_COM0	0	TIMER IO, TM_COM0 select. 1 => TM_COM0 is selected 0 => GPIO [2] is selected
1	R/W	MUXC_CAP1	0	TIMER IO, TM_CAP1 select. 1 => TM_CAP0 is selected 0 => GPIO [1] is selected
0	R/W	MUXC_CAP0	0	TIMER IO, TM_CAP0 select. 1 => TM_CAP0 is selected 0 => GPIO [0] is selected

MUXC_PU
address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7	R/W	MUXC_PU31	1	pull up for IO[31] enable
…	R/W	…	1	…
1	R/W	MUXC_PU1	1	pull up for IO[1] enable
0	R/W	MUXC_PU0	1	pull up for IO[0] enable, set to 1 to enable

5.10. Debug Register

Debug register of the normal cores are identical to the debug register of the IO-control core. Refer to section 4.11 for more details.

5.11. AHB to NoC Bridge

Most of the features of the AHB to NoC bridge of the normal cores are similar to the bridge of the IO-control core. The different is related to the bootloading-debugging mechanism that requires the use of the NoC interconnect to share data (refer to section 6 and 7). The AHB to NoC bridge of the normal cores has only two interrupts. The module also doesn’t have the bootloading-debugging related registers.

5.11.1 AHB to NoC Bridge Features

- Transmit and Receive 32-bit data in the form of flit through NoC
- Address decoding mechanism to set the destination for the flit
- Transmit buffer with the depth of four
- Multiple receive buffers to contains flits from different sources, each buffer has eight buffer depth except for the loopback buffer only has two depth

- Two interrupt signals generation for transmit, receive and bootloading event

5.11.2 AHB to NoC Bridge IO Pins

Signal	IO	Source	Dest’ n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Bridge	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Bridge	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Bridge	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Bridge	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Bridge	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Bridge	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Bridge	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Bridge	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Bridge	AHB ready input
HRDATA[31:0]	DO	Bridge	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Bridge	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Bridge	AHB Lite Bus	Indicates the transfer response
NOC				
EN_putFlit	DO	Bridge	NoC	Enable the transmit of flit
putFlit[37:0]	DO	Bridge	NoC	The flit to be transmitted
EN_getCredits	DO	Bridge	NoC	Enable the receive of credit
getCredits[1:0]	DI	NoC	Bridge	The credit to be received
EN_getFlit	DO	Bridge	NoC	Enable the receive of flit
getFlit[37:0]	DI	NoC	Bridge	The flit to be received
EN_putCredits	DO	Bridge	NoC	Enable the transmit of credit
putCredits[1:0]	DO	Bridge	NoC	The credit to be transmitted
Router_id[1:0]	DI	NoC	Bridge	Attached router index
SYSTEM CONTROL				
mode	DI	Sys ctrl	Bridge	Bridge operating mode
MASTER MUX				

DB_HADDR[31:0]	DO	Bridge	Mux	Debug access address bus
DB_HSIZE[2:0]	DO	Bridge	Mux	Debug access transfer size
DB_HTRANS[1:0]	DO	Mux	Bridge	Debug access transfer type
DB_HWDATA[31:0]	DO	Mux	Bridge	Debug access write data
DB_HWRITE	DO	Mux	Bridge	Debug access transfer direction
DB_HRDATA[31:0]	DI	Bridge	Mux	Debug access read data
DB_HREADY	DI	Bridge	Mux	Debug access slave ready signal
DB_HRESP	DI	Bridge	Mux	Debug access slave response signal
BL_HADDR[31:0]	DO	Bridge	Mux	Bootloalding access address bus
BL_HSIZE[2:0]	DO	Bridge	Mux	Bootloalding access transfer size
BL_HTRANS[1:0]	DO	Bridge	Mux	Bootloalding access transfer type
BL_HWDATA[31:0]	DO	Bridge	Mux	Bootloalding access write data
BL_HWRITE	DO	Bridge	Mux	Bootloalding access transfer direction
debug_ready	DI	Mux	Bridge	Signal that indicate the mux is ready to serve debug access
FLASH CONTROLLER				
flash_int	DI	Flash ctrl	Bridge	Flash controller interrupt, used in bootloading process
INTERRUPT				
NOC_TX_INT	DO	Bridge	M0	Transmit interrupt
NOC_RX_INT	DO	Bridge	M0	Receive interrupt

5.11.3 AHB to NoC Bridge Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	R	NOC_CSR0	0x00000000	control and status register 0
0x04	R/W	NOC_CSR1	0x00000000	control and status register 1
0x08	R/W	NOC_CSR2	0x0000ffff	control and status register 2
0x0C	R	NOC_RX_BUFF0	0x00000000	receive buffer 0
0x10	R	NOC_RX_BUFF1	0x00000000	receive buffer 1
0x14	R	NOC_RX_BUFF2	0x00000000	receive buffer 2
0x18	R	NOC_RX_BUFF3	0x00000000	receive buffer 3
0x1C	R/W	NOC_TX_BUFF0	0x00000000	transmit buffer 0
0x20	R/W	NOC_TX_BUFF1	0x00000000	transmit buffer 1
0x24	R/W	NOC_TX_BUFF2	0x00000000	transmit buffer 2
0x28	R/W	NOC_TX_BUFF3	0x00000000	transmit buffer 3

Register Description

NOC_CSR0
address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7	R	NOC_RXF3	0	see NOC_RXF0
6	R	NOC_RX3	0	see NOC_RX0
5	R	NOC_RXF2	0	see NOC_RXF0
4	R	NOC_RX2	0	see NOC_RX0
3	R	NOC_RXF1	0	see NOC_RXF0
2	R	NOC_RX1	0	see NOC_RX0
1	R	NOC_RXF0	0	Receive buffer full flag; 0 => receive buffer 0 still have empty space; 1 => receive buffer full
0	R	NOC_RX0	0	Receive flit flag; 0 => no received flit, 1 => there is a received flit in NOC_RX_BUFF0. A read to NOC_RX_BUFF will discard the data from the buffer. If there is no more data in the buffer, NOC_RX will be cleared by hardware, if there is another data, the NOC_RX will remain high.

NOC_CSR1

address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31	W	NOC_BDIE_CLR	0	Write 1 to this bit to clr NOC_BDIE, write 0 will take no effect.
30	W	NOC_BDIE_SET	0	Write 1 to this bit to set NOC_BDIE, write 0 will take no effect.
29	W	NOC_RIM_CLR	0	Write 1 to this bit to clr NOC_RIM, write 0 will take no effect.
28	W	NOC_RIM_SET	0	Write 1 to this bit to set NOC_RIM, write 0 will take no effect.
27	W	NOC_RIE_CLR	0	Write 1 to this bit to disable the rx interrupt, write 0 will take no effect.
26	W	NOC_RIE_SET	0	Write 1 to this bit to enable the rx interrupt, write 0 will take no effect.
25	W	NOC_TIE_CLR	0	Write 1 to this bit to disable the tx interrupt, write 0 will take no effect.
24	W	NOC_TIE_SET	0	Write 1 to this bit to enable the tx interrupt, write 0 will take no effect.
23:6	R	RESERVED	0	unused
5	R	NOC_BDIE	0	BD interrupt enable. 0 => interrupt disabled. 1 => interrupt enabled. NOC_BDIE_INT line will be high when there is ACK received.
4	R	NOC_RIM	0	receive interrupt mode. 0 => receive interrupt

				mode 0, interrupt is generated every time a flit received. 1 => receive interrupt mode 1, interrupt is generated when N number of flit arrived
3	R	NOC_RIE	0	receive interrupt enable. 0 => interrupt disabled. 1 => interrupt enabled. NOC_RX_INT line will be high when there is data in the RX_BUFFER
2	R	NOC_TIE	0	transmit complete interrupt enable. 0 => interrupt disabled. 1 => interrupt enabled. NOC_TX_INT line will be high when there is finished transfer. Write 1 to PP_TX will clear the interrupt.
1	R	NOC_TXF	0	Transmit buffer full flag; 0 => transmit buffer still have empty space; 1 => transmit buffer full, AHB master are not allowed to write into NOC_TX_BUFFn
0	RW	NOC_TX	0	transmit complete flag, set high when there were data written to TX_BUFF and the tx buffer is empty. Write 1 to this bit will clear the flag.

NOC_CSR2

Address offset: 0x08

Bits	R/W	Field Name	Reset value	Description
31:28	RW	NOC_RX3_COUNT	0	see NOC_RX0_COUNT
27:24	RW	NOC_RX2_COUNT	0	see NOC_RX0_COUNT
23:20	RW	NOC_RX1_COUNT	0	see NOC_RX0_COUNT
19:16	RW	NOC_RX0_COUNT	0	Receive flit counter, count the number of flits that are received and stored in the receive buffer 0. If the counter value is equal or more than the value of NOC_RX0_TRIG, interrupt is triggered (NOC_RIM need to be high).Each read to the receive buffer 0 reduces the counter value by 1. User can also over-write the value of this counter.
15:12	RW	NOC_RX3_TRIG	f	see NOC_RX0_TRIG
11:8	RW	NOC_RX2_TRIG	f	see NOC_RX0_TRIG
7:4	RW	NOC_RX1_TRIG	f	see NOC_RX0_TRIG
3:0	RW	NOC_RX0_TRIG	f	The number of flits to be received and stored in the receive buffer 0 before the interrupt is triggered. This value is only valid if the NOC_RIM is high.

NOC_RX_BUFFn
address offset: 0x08 - 0x14

Bits	R/W	Field Name	Reset value	Description

31:8	R	-	0	Receive buffer. Read from this buffer will discard one flit from buffer if the flit is available.

NOC_TX_BUFFn
address offset: 0x18 - 0x24

Bits	R/W	Field Name	Reset value	Description
31:8	W	-	0	Transmit buffer. write to this buffer will trigger transfer through NoC to the module that connected to router n.

5.12. AES Accelerator

AES Accelerator (AHB_AES) is an AHB slave module that can perform 128-bit AES encryption. This module takes 128-bit of AES key that is stored in the AES_KEYn registers and perform AES encryption to 128-bit plain text from the AES_PLAINn registers. The encryption is started every time AES_PLAIN3 register is written. The module support polling and interrupt method to determine the availability of the cipher text (after an encryption process is finished). A plain text buffer is implemented inside the module that enable the master to latch a new plain text to the AHB_AES module while it still performing the encryption. The buffered plain text will be processed after the master module read AES_CHIPER3.

5.12.1 AES Accelerator Features

- 128-bit AES encryption calculation in ten cycles
- Interrupt signal generation to indicate the availability of cipher text
- Plain text buffer implementation for pipeline encryption calculation

5.12.2 AES Accelerator IO Pins

Signal	IO	Source	Dest’ n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Accelerator	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Accelerator	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Accelerator	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Accelerator	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Accelerator	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Accelerator	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Accelerator	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Accelerator	Indicates the transfer direction

HREADY	DI	AHB Lite Bus	Accelerator	AHB ready input
HRDATA[31:0]	DO	Accelerator	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Accelerator	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Accelerator	AHB Lite Bus	Indicates the transfer response
INTERRUPT				
AES_INT	DO	Accelerator	M0	Cipher text available interrupt

5.12.3 AES Accelerator Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	W	AES_CSR	0x00000000	AES control and status register
0x04	R/W	AES_KEY0	0x00000000	AES key [31:0]
0x08	R/W	AES_KEY1	0x00000000	AES key [63:32]
0x0C	R/W	AES_KEY2	0x00000000	AES key [95:64]
0x10	R/W	AES_KEY3	0x00000000	AES key [127:96]
0x14	R/W	AES_PLAIN0	0x00000000	AES plaintext [31:0]
0x18	R/W	AES_PLAIN1	0x00000000	AES plaintext [63:32]
0x1C	R/W	AES_PLAIN2	0x00000000	AES plaintext [95:64]
0x20	R/W	AES_PLAIN3	0x00000000	AES plaintext [127:96]
0x24	R	AES_CIPHER0	0x00000000	AES ciphertext [31:0]
0x28	R	AES_CIPHER1	0x00000000	AES ciphertext [63:32]
0x2C	R	AES_CIPHER2	0x00000000	AES ciphertext [95:64]
0x30	R	AES_CIPHER3	0x00000000	AES ciphertext [127:96]

Register Description

AES_CSR
address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:17	R	RESERVED	0	unused
16	R	AES_FF		full flag; 0 => AHB_AES1 module still can receive another plain text ; 1 => there is a plain text processed and a plain text queued in the AHB_AES module, can't affort to receive another one, this flag will be cleared when a read done to AES_CIPHER3
15:9	R	RESERVED	0	unused

8	R	AES_CF		ciphertext flag; 0 => no new chypertext, 1 => there is new chypertext in AES_CHIPER0-3, this flag will be cleared when a read done to AES_CIPHER3.
7:1	R	RESERVED	0	unused
0	R/W	AES_IE	0	chipertext interupt enable

AES_KEYn
address offset: 0x04 - 0x10

Bits	R/W	Field Name	Reset value	Description
31	R/W	-	0	AES key for AES encryption calculation, stored in little-endian manner

AES_PLAINn
address offset: 0x14 - 0x20

Bits	R/W	Field Name	Reset value	Description
31	R	-	0	AES plaintext to be encrypted, stored in little-endian manner

AES_CIPHERn
address offset: 0x24 - 0x30

Bits	R/W	Field Name	Reset value	Description
31	R	-	0	AES ciphertext as the result of encryption process, stored in little-endian manner

6. DSP Core

The DSP core is similar to the normal cores. The only different that the DSP core is equipped with MAC instead of AES accelerator. The MAC helps the DSP core to perform DSP tasks.

6.1. Memory Mapping

The DSP core has eight slave modules attached to the AHB-lite bus. The address of the slaves are:

Slave num	Slave	start address	end address
1	Flash	0x0000_0000	0x0000_83FF
2	SRAM	0x2000_0000	0x2000_1FFF

3	APB	0x4000_0000	0x4000_3FFF
4	GPIO	0x5000_0000	0x5000_FFFF
5	Mux controller	0x5001_0000	0x5001_FFFF
6	Debug	0x5002_0000	0x5002_FFFF
7	MAC	0x5003_0000	0x5003_FFFF
8	NoC bridge	0x5004_0000	0x5004_FFFF

The address of the peripherals that are attached to APB bus are:

Slave num	Slave	start address	end address
1	APB Flash	0x4000_0000	0x4000_0FFF
2	APB Timer	0x4000_1000	0x4000_1FFF
3	APB SPI	0x4000_2000	0x4000_2FFF
4	APB WDT	0x4000_3000	0x4000_3FFF

6.2. Interrupt Vector Table

The highest priority interrupt (NMI) of the normal core is used for debug event (refer to section 7). Eight interrupt pins are used as peripheral interrupts. The interrupt vector table are:

no	interrupt	description
0	flash	flash APB interface interrupt
1	noc tx	NoC transmit interrupt
2	noc rx	NoC receive interrupt
3	timer	timer interrupt
4	spi	SPI interrupt
5	gpio0	GPIO interrupt 0
6	gpio1	GPIO interrupt 1
7	wdt	Watch Dog Timer interrupt
8		

6.3. Flash Controller

Flash controller of the DSP core is identical to the flash controller of the IO-control core and normal cores. Refer to section 4.3 for more details.

6.4. SRAM Controller

SRAM controller of the DSP core is identical to the SRAM controller of the IO-control core and normal cores. Refer to section 4.4 for more details.

6.5. GPIO

GPIO module of the DSP core is identical to the GPIO module of the normal cores. Refer to section 5.5 for more details.

6.6. Timer

Timer module of the DSP core is identical to the Timer module of the IO-control core and normal cores. Refer to section 4.7 for more details.

6.7. SPI

SPI module of the DSP core is identical to the SPI module of the IO-control core and normal cores. Refer to section 4.8 for more details.

6.8. Watch Dog Timer (WDT)

WDT module of the DSP core is identical to the WDT module of the IO-control core and normal cores. Refer to section 4.9 for more details.

6.9. Mux Controller

Mux controller of the DSP core is identical to the mux controller of the normal cores. Refer to section 5.9 for more details.

6.10. Debug Register

Debug register of the DSP core is identical to the debug register of the IO-control core and normal cores. Refer to section 4.11 for more details.

6.11. AHB to NoC Bridge

AHB to NoC Bridge of the DSP core is identical to the AHB to NoC Bridge of the normal cores. Refer to section 5.11 for more details.

6.12. AHB_MAC

AHB_MAC is an AHB-lite slave that can be used to perform multiplication and multiply-accumulate operations in one clock cycle. The module supports 16-bit and 32-bit multiplication and 32-bit multiply-accumulate operations. The module also support unsigned and two’s compliment multiplication.

6.12.1 AHB_MAC Features

- Single cycle 16-bit and 32-bit multiplication operations
- Single cycle 32-bit multiply-accumulate operation
- 72-bit result register to store the result of continuous multiply-accumulate operations

- Unsigned and two’s compliment multiplication mode

6.12.2 AHB_MAC IO Pins

Signal	IO	Source	Dest’ n	Function
AMBA AHB				
HRESETn	DI	AHB Lite Bus	Accelerator	AMBA APB active low reset
HCLK	DI	AHB Lite Bus	Accelerator	AMBA AHB clock. All signal timings are related to the rising edge of this clock
HSEL	DI	AHB Lite Bus	Accelerator	Indicates that the current transfer is intended for the selected slave
HADDR[31:0]	DI	AHB Lite Bus	Accelerator	AMBA AHB 32-bit system address bus
HSIZE[2:0]	DI	AHB Lite Bus	Accelerator	Indicates the size of transfer, that is typically byte, halfword or word
HTRANS[1:0]	DI	AHB Lite Bus	Accelerator	Indicates the transfer type of the current transfer
HWDATA[31:0]	DI	AHB Lite Bus	Accelerator	AMBA AHB write data bus
HWRITE	DI	AHB Lite Bus	Accelerator	Indicates the transfer direction
HREADY	DI	AHB Lite Bus	Accelerator	AHB ready input
HRDATA[31:0]	DO	Accelerator	AHB Lite Bus	AMBA AHB read data bus
HREADYOUT	DO	Accelerator	AHB Lite Bus	Indicates that a transfer has finished on the bus
HRESP	DO	Accelerator	AHB Lite Bus	Indicates the transfer response

6.12.3 AHB_MAC Registers

Register Overview

address offset	R/W	Register Name	Reset value	Description
0x00	R/W	MAC_CSR	0x00000000	AHB_MAC control and status register
0x04	W	MAC_A	0x00000000	A input
0x08	R/W	MAC_B	0x00000000	B input
0x0C	R/W	MAC_MUL0	0x00000000	Result register
0x10	R/W	MAC_MUL1	0x00000000	Result register
0x14	R/W	MAC_MUL2	0x00000000	Result register

Register Description

MAC_CSR

address offset: 0x00

Bits	R/W	Field Name	Reset value	Description
31:4	R	RESERVED	0	unused
3	W	MAC_CLR		Clear bit, write 1 to clear the MUL registers. Write 0 take no effect.
2	R/W	MAC_TC	0	Two’s compliment control bit. The unsigned multiplication operation is performed as the default.

				Write this bit to 1 to set the MAC to operate in two's compliment mode.
1:0	R/W	MAC_MODE	0	Mode bits: 00: 32-bit multiplication mode. In this mode, a write to MAC_A starts 32-bit multiplication (MAC_A*MAC_B). The result is stored into MAC_MUL0 and MAC_MUL1. 01: 16-bit multiplication mode. In this mode, a write to MAC_A starts 16-bit multiplication by only using MAC_A content (MAC_A[31:16]*MAC_A[15:0]). The result is stored into MAC_MUL0. 10: 32-bit multiply-accumulate mode. In this mode, a write to MAC_A starts 32-bit multiply-accumulate (MAC_A*MAC_B+MUL). The result is stored into MAC_MUL0, MAC_MUL1 and MAC_MUL2. 11: unused.

MAC_A

address offset: 0x04

Bits	R/W	Field Name	Reset value	Description
31:0	R/W	-	0	Input A. A write to this register starts the multiplication/multiply-accumulate operation.

MAC_B

address offset: 0x08

Bits	R/W	Field Name	Reset value	Description
31:0	R	-	0	Input B.

MAC_MUL0

address offset: 0x0C

Bits	R/W	Field Name	Reset value	Description
31:0	R/W	-	0	MUL[31:0]

MAC_MUL1

address offset: 0x10

Bits	R/W	Field Name	Reset value	Description
31:0	R/W	-	0	MUL[63:32]

MAC_MUL2

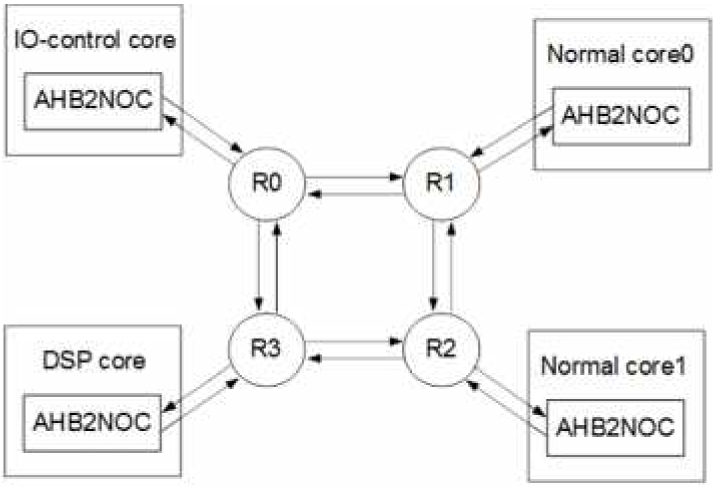
address offset: 0x14

Bits	R/W	Field Name	Reset value	Description
31:8	R	RESERVED	0	unused
7:0	R/W	-	0	MUL[71:64]

7. NoC Interconnect

NoC interconnect is implemented as the inter-core communication media. NoC concept adopts terrestrial network communication protocol and applied it into on-chip interconnect network. An NoC consists of several routers that are interconnected in a specific network topology, such as ring, mesh or tree. Each router is considered as one communication node. Multiple processing elements are attached to these routers. NoC implements packet switching method to pass data from one communication node to the others. The NoC packets that are called flits, travel multiple hops through the NoC interconnect. The routers applied a routing algorithm to route these flits to the destination.

The NoC that is used in the UV1 chip is generated from CONNECT generator. CONNECT is a web-based tool from Carnegie Mellon University project. The tool accepts several parameters to configure the NoC to be generated, such as network topology, number of nodes and flit size.



The NoC interconnect in UV1 chip implements double ring topology. There are four routers in the network. Each single core system are attached to each of the router. The AHB to NoC bridges interface the single core systems that use AHB protocol and the routers in NoC. Each core is assigned with a fix network address.

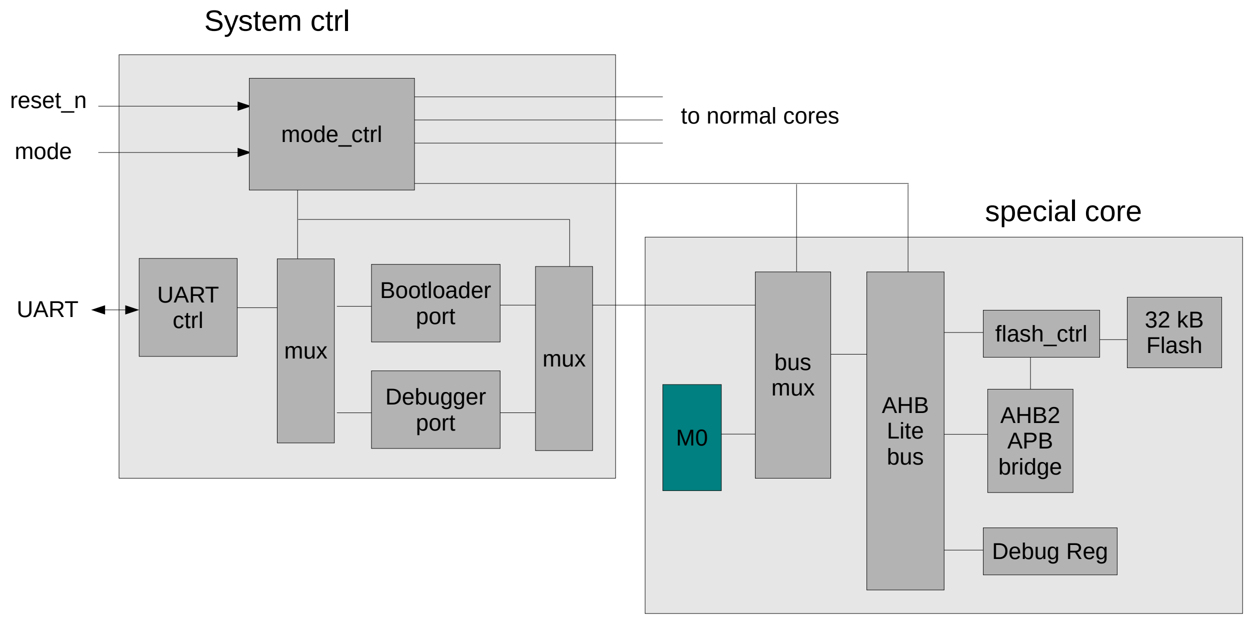
core	network address
IO-control core	0
normal core 0	1
normal core 1	2
DSP core	3

8. Bootloader

The hardware bootloader module and the other modules that support the bootloading mechanism are used to write program codes to the flash memory of the cores. There are two bootloading modes, hardware bootloading mode and software bootloading mode. The hardware bootloading mode is used to store the program codes of the IO-control core. The flash memory of the IO-control core can be used to store two type of program codes, software bootloader code and application code. The codes are stored in the different memory partition. Both can be loaded through the hardware bootloader. The software bootloader code is executed when the chip is in the software bootloading mode. The IO-control core can be programmed to store the application codes of the normal cores or to store its own application code.

8.1. Hardware bootloading Mode

The hardware bootloader is specifically used to store the program codes of the IO-control core. The module use UART interface as the media for command and data transfer from the host programmer (PC program or microcontroller). To enter hardware bootloading mode, during reset, the mode0 and mode2 pin should be pulled low and the mode1 pin should be pulled high. After comes out of reset, the bootloader module is ready to receive the command from the host programmer. The sequence of commands to define the program transfer from the host programmer to the bootloader is defined in the protocol sub-chapter. In the hardware bootloading mode, the bootloader module take over the AHB lite bus of the IO-control core and use AHB signals to write the program data to the flash. The flash memory of the IO-control core is divided into two partition. The first partition, which starts from address 0x00000000 is used to store software bootloader code. The second partition is used to store the application program. The application program partition has two starting address, 0x00001000 and 0x00002000. Two normal chip operating mode (refer to section 3.3) can be selected to set the M0 processor of the IO-control core to execute the application program. Each normal mode set the processor to access one of the possible starting address of the application program.



8.1.1 Protocol

Physical Layer

The UART used to transmit information has the following configuration:

- Character: 8-bit data
- Parity: none
- Stop: 2 bits
- Flow control: none
- Baud rate: 115200

Transfer Format

Program transfer is started by the host programmer. The command and data to be transferred are in the form of ASCII characters. A pair of ASCII characters is used to represent one byte hexadecimal value, except for the mark field. The mark field is used by the host programmer to start the transfer. It represented by one ASCII character ':'.

	mark ':'	length	type	data	checksum
number of character sent	1 byte	2 bytes	2 bytes	2n bytes	2 bytes
number of byte represented	1 byte	1 byte	1 byte	n bytes	1 byte

- Transfer mark:
  Character to be sent by the host programmer to start of the transfer

- Transfer length:
  Specify the number of data bytes to be transferred
- Transfer type:
  Specify the type of the transfer
- Transfer data:
  Variable length field that consist of the data/info to be transferred
- Transfer checksum:
  Two's complement of the 8-bit Bytes that result from converting each pair of ASCII hexadecimal digits to one Byte of binary (including the length and type fields). Therefore, the sum of all the ASCII pairs in the transfer after converting to binary, from length field and including the checksum field, is zero.

Type fields

type	description
0x00	load program to bootload program partition of the flash memory of the IO-control core
0x01	load program to application program partition of the IO-control core start from address 0x00001000 (4kB)
0x02	load program to application program partition of the IO-control core start from address 0x00002000 (8kB)

Length fields

The length fields define the number of data bytes to be sent. The size one block of flash memory is 512 kB. The size of the transfer of program data is the multiple of 512 Bytes. The size follow the equation:

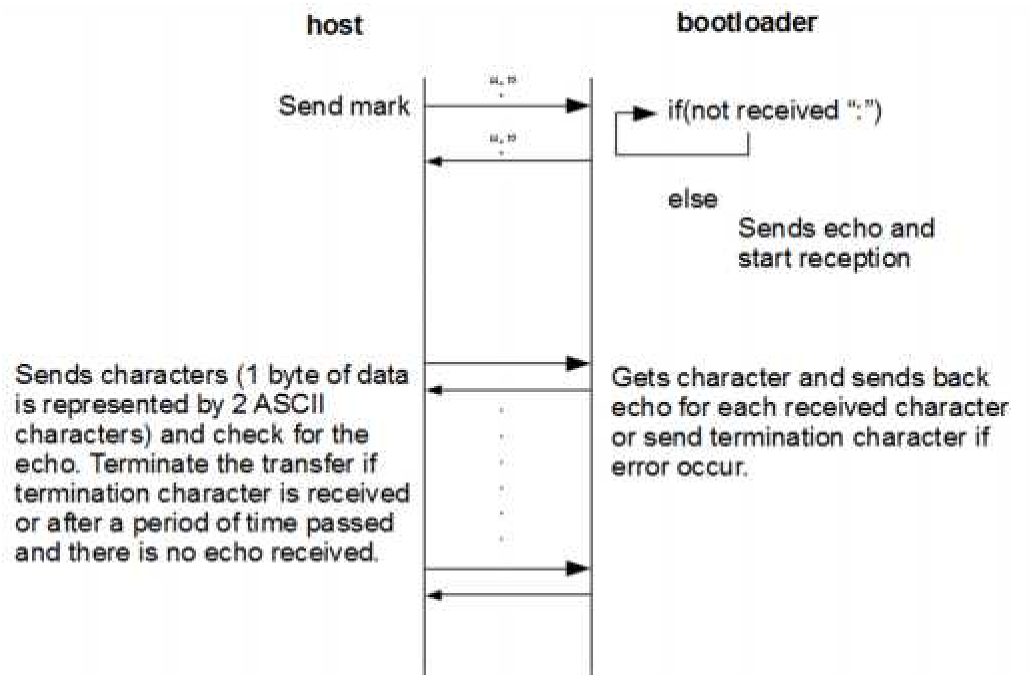
N = (L+1) * 512 Bytes

where:

N : number of data bytes to be sent

L : length field content

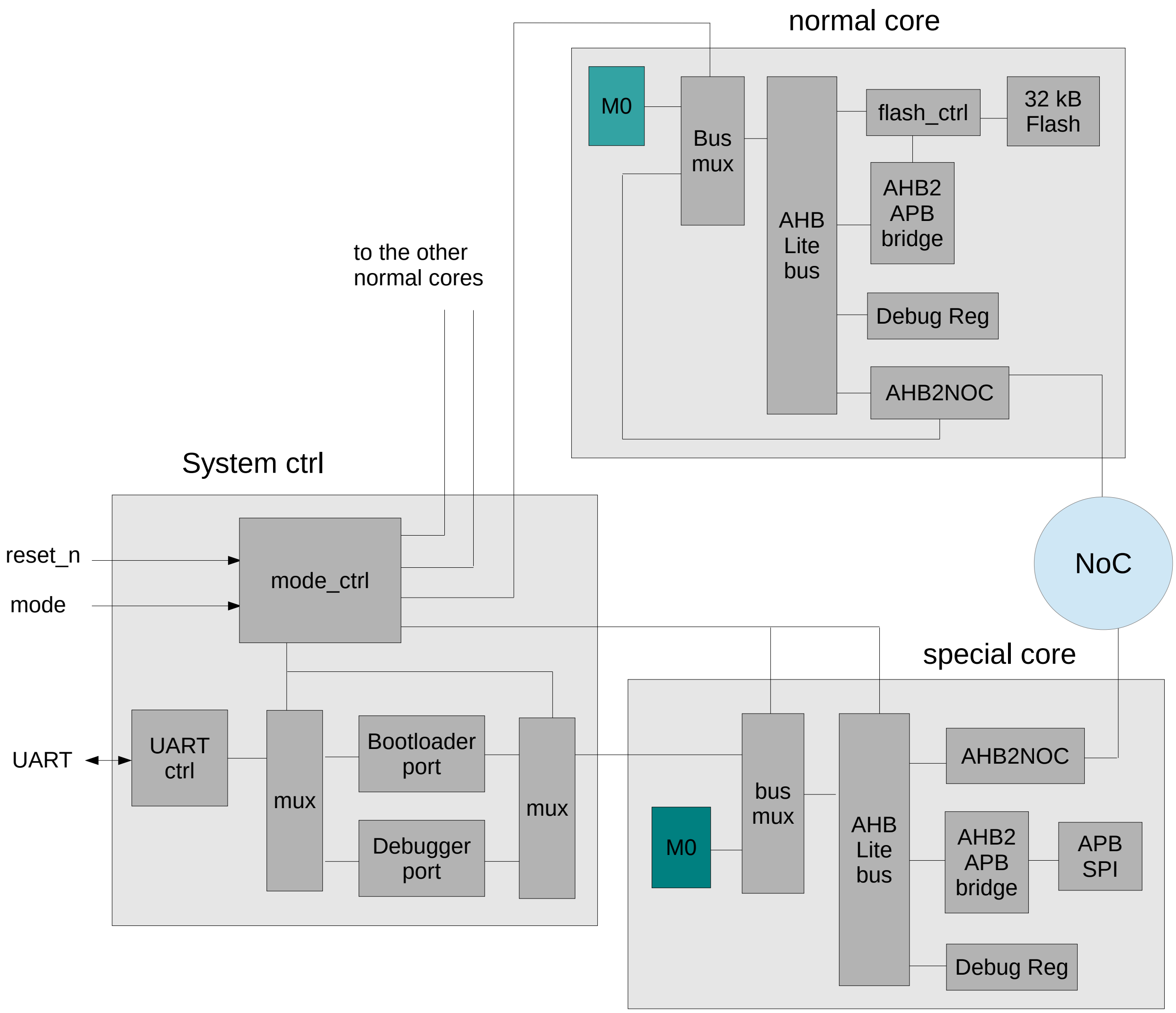
Flow Diagram


host
bootloader
":"
Send mark
if(not received ":")
":"
else
Sends echo and
start reception
Sends characters (1 byte of data
is represented by 2 ASCII
characters) and check for the
echo. Terminate the transfer if
termination character is received
or after a period of time passed
and there is no echo received.
Gets character and sends back
echo for each received character
or send termination character if
error occur.

As defined in the last sub chapter, the transfer is started by the programmer. The host programmer send ':' mark character to bootloader. Bootloader that receive the mark character will send it back as the acknowledgement to the host programmer. If the host programmer receive the echo from the bootloader, it can start to transfer the pair of ASCII characters to represent each byte of command/data. Bootloader module will send back the echo every time it receive one ASCII character. Bootloader module can also sent '.' as termination character to replace the echo. This termination character is sent if the host programmer had sent a unexpected value. The host programmer should monitor the echo character that is sent by the bootloader and terminate the transfer if a wrong echo value is received or the bootloader send the termination character. The host programmer also needs to implement a time out counter to anticipate that the bootloader never be able to send the echo or the connection between the two of them are interrupted.

8.2. Software bootloading Mode

During the software bootloading mode, the AHB2NOC module of the normal cores get the control of the AHB lite bus inside the normal cores. The IO-control core can be programmed to act as the secondary bootloader for the normal cores. The program data for the normal cores can be obtained from the I/O ports and distributed through the NoC. The AHB2NOC modules will handle the write to the flash memory of the normal cores.


normal core
M0
Bus mux
AHB Lite bus
flash_ctrl
32 kB Flash
AHB2 APB bridge
Debug Reg
AHB2NOC
to the other normal cores
System ctrl
reset_n
mode
mode_ctrl
UART
UART ctrl
mux
Bootloader port
Debugger port
mux
NoC
special core
bus mux
M0
AHB Lite bus
AHB2NOC
AHB2 APB bridge
APB SPI
Debug Reg

8. Debugger

The debug process utilizes the same hardware components that are used for bootloading process.

Debugger procedure:

- External debug controller connected
- One of the core write to the debug register to start debug event
- If granted, debug register in all of the cores generate interrupt (NMI)
- Master mux wait until the M0 processor try to access the NMI vector in flash and the pull HREADY low to halt the core
- AHB2NOC of the normal cores get the access to AHB-lite bus, read debug register, send the data to IO-control core
- The debugger port get the access to the AHB-lite bus of the IO-control core, read IO-control core's debug register and AHB2NOC register, drives the data out through UART
- The process repeated until all of the data are latched out
- The Master mux release the HREADY