RTL9310

CPU, Memory Control, and Peripherals

DATASHEET

(CONFIDENTIAL: Development Partners Only)

Rev. 1.0
20 Apr 2018

Realtek Semiconductor Corp.
No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300,
Taiwan Tel.: +886-3-578-0211 Fax: +886-3-577-6047

www.realtek.com

TRADEMARKS

Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

DISCLAIMER

Realtek provides this document "as is", without warranty of any kind, neither expressed nor implied, including, but not limited to, the particular purpose. Realtek may make improvements and/or changes in this document or in the product described in this document at any time. This document could include technical inaccuracies or typographical errors.

USING THIS DOCUMENT

This document is intended for use by the system engineer when integrating with Realtek switch products. Though every effort has been made to assure that this document is current and accurate, more information may have become available subsequent to the production of this guide. In that event, please contact your Realtek representative for additional information that may help in the development process.

Revision	Release Date	Summary
1.0	2018/04/20	Initial

Contents

1. SOC Block Diagram

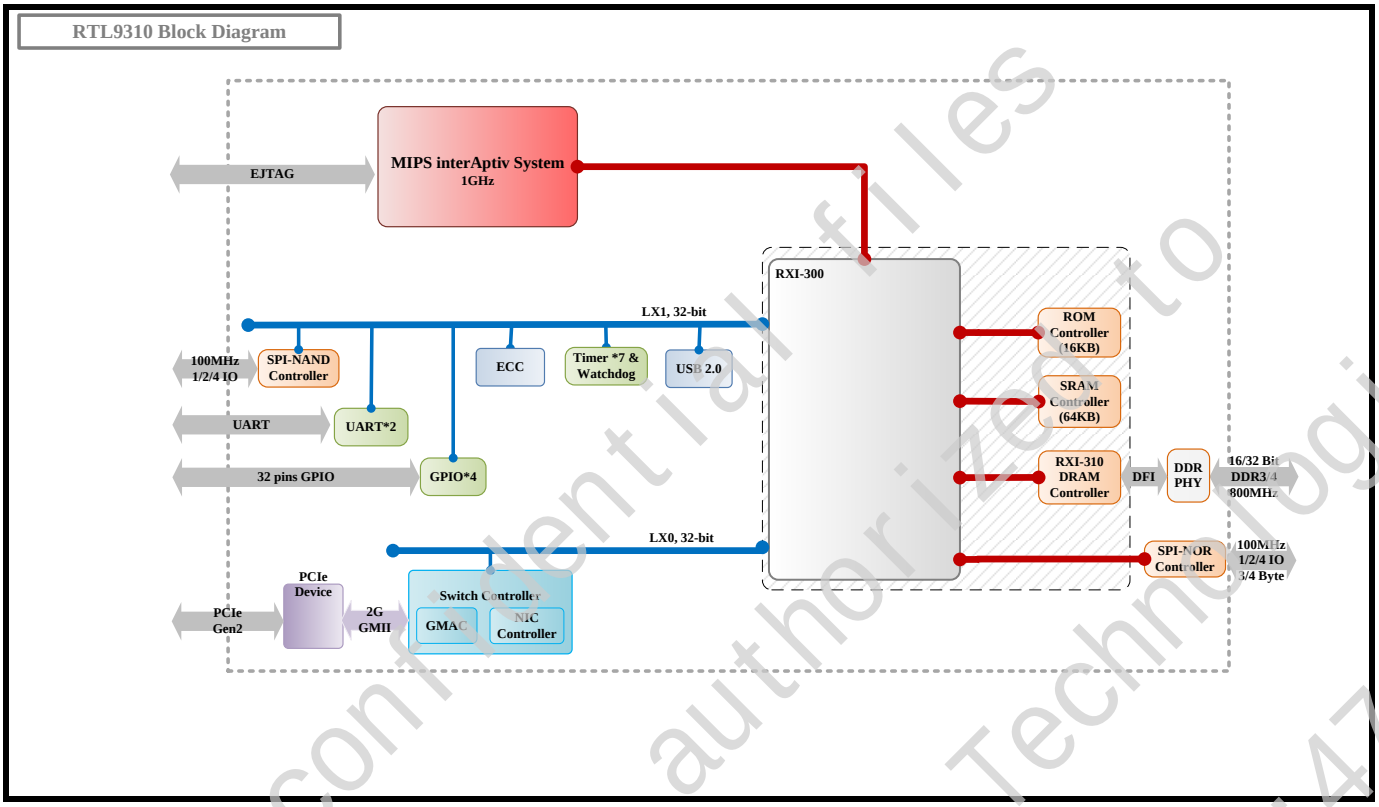


Figure. RTL9310 SOC Block Diagram

2. RTL9310 SoC General

2.1. *Feature Lists*

- **CPU**
  - MIPS InterAptiv, dual-core, each core has 2 VPEs
  - up to 1GHz
  - ICache: 32KB
  - DCache: 32KB
  - L2Cache: 256KB
- **SRAM**
  - On-Chip, 64KB
- **ROM**
  - On-Chip, 16KB
- **DRAM**
  - DDR3/DDR4, up to total 2GB address space
  - 16/32-bit, up to 800MHz
  - Two 16-bit chips supported, combined as 32-bit
- **SPI-NOR Flash**
  - Up to total 64MB address space
  - Up to 100MHz
  - 2 Chip Selects supported
- **SPI-NAND Flash**
  - Up to total 512MB address space
  - Up to 100MHz
  - 2 Chip Selects supported
- **USB 2.0**
- **Peripheral**
  - **UART:** 2 sets of UARTs supported
  - **Internal GPIO:** 4 sets, total 32 pins
  - **Timer/Counter**: 7 sets
  - **Watchdog**
  - **Delayed Interrupt**
- **Boot Type**
  - SPI NOR Flash Boot
  - ROM Boot for SPI-NAND

2.2. Hardware Pin Definition

Strapped-pin Description

Name	Pin Counts	Description
DDR_TYPE	2	2'b00: DDR3 2'b01: DDR4
DDR_DATA_BUS_WIDTH	1	1'b0: DQ16 1'b1: DQ32
BOOT_SEL	1	1'b0: SPI-NOR 1'b1: ROM boot + SPI-NAND
RST_CMD_DIS	1	Reference SFCR bit 8

3. SOC PLL Requirement

3.1. *PLL Output*

- **CPU**: OCP_PLL = 500 ~ (1000+10%) MHz
  - Divisor: 1, 2, 4
  - OCP_CLK = 125~1000 MHz

- **DRAM**: DDR_PLL = 250~800MHz
  - Using DFI architecture, MAC:PHY frequency ratio = 1:2
  - DDR_CLK = 100 (fixed) or 250~800MHz (to Device)

- **Lexra Bus/ SPI NAND Flash/USB 2.0**: up to 200MHz
  - Lexra Bus CLK = up to 200MHz
  - SPI_NAND Divisor: 2, 4, 6, …. 16 (IP Internal)
  - SPI NAND CLK = up to 100MHz
  - USB2.0 CLK = up to 200MHz

- **SPI NOR Flash**: up to 200MHz
  - SPI_NOR Divisor: 2, 4, 6, …. 16 (IP Internal)
  - SPI_NOR_CLK = up to 100MHz

4. CPU Specification

4.1. Features List

- MIPS InterAptiv CPU Core
  - 9 stage-pipeline
  - MIPS32 Release 2 instruction set
  - Additional MIPS16e instruction set support
  - 2 Virtual Processing Elements (VPEs) support
  - Vectored interrupts, Non-maskable interrupts (NMI) supports
- Cache Configuration:
  - I-Cache: 32KB, 4 way set associative, 32 byte line size
  - D-Cache: 32KB, 4 way set associative, 32 byte line size, write back policy
  - L2-Cache: 256K, 8 way set associative, 32 byte line size
- Memory Management Unit (MMU) Configuration
  - 4-5 entry MT-optimized ITLB
  - 8 entry DTLB
  - 16/32/64 dual-entry JTLB per VPE
- Misc.
  - Power-down mode (triggered by WAIT instruction)
  - EJTAG support
  - Internal BIST
  - Internal real-time timer interrupts (Count/Compare registers)
  - CPU breakpoints

4.2. *Global Interrupt Controller (GIC)*

The Global Interrupt Controller (GIC) handles the distribution of interrupts between and among the VPEs in the cluster.

This block has the following features:

- Software interface through relocatable memory-mapped address range.
- Configurable number of system interrupts - from 8 to 256 in multiples of 8.
- Support for different interrupt types:
- Level-sensitive: active high or low.
- Edge-sensitive: positive, negative, or double-edge-sensitive.
- Ability to mask and control routing of interrupts to a particular CPU.
- Support for NMI routing.
- Standardized mechanism for sending inter-processor interrupts.

5. SPI-NOR Flash Controller Specification

5.1. *Features*

- Targeted frequency: up to 100 MHz
- Chip Select(s): CS0# and CS1#
- Max SPI-Flash Size: 64MB (64MB*1)
- Read Operation: Programmed I/O (PIO) and memory-mapped I/O (MMIO) interface
  - In MMIO mode, 8-bit command code is user-configurable for accommodate different commands of different vendors.
- Multiple I/O Support: serial I/O (SIO), dual I/O (DIO) and quad I/O mode (QIO)
- Address Mode Support: 3 or 4 bytes Address mode

5.1.1. Pin Mode and Definition of Serial, Dual, and Quad I/O

1. Serial I/O mode:
   - SI: flash chip input pin
   - SO: flash chip output pin
2. Dual I/O mode:
   - SIO0 (SI): flash chip bi-directional pin. (LSB)
   - SIO1 (SO): flash chip bi-directional pin. (MSB)
3. Quad I/O mode:
   - SIO0 (SI): flash chip bi-directional pin. (LSB)
   - SIO1 (SO): flash chip bi-directional pin.
   - SIO2 (WP#): flash chip bi-directional pin.
   - SIO3 (HOLD#): flash chip bi-directional pin. (MSB)

5.1.2. Programmed I/O Operation

PIO (Programmed I/O) operation is done by register access through software execution. Suggested basic software procedure is composed of following steps:

1. Ensure that SPI-NOR Flash controller is in ready condition by checking SFCSR.
2. Setting clock divisor properly, read/write ordering, and chip deselect time in SFCR1. (Usually required to be set only once)
3. Select CS# through setting SFCSR, and also properly set read/write data length and I/O width mode (serial, dual, or quad).
4. Write SFDR to specify SPI-NOR Flash command / address / data. Read SFDR to get SPI data.
5. De-select CS# through setting SFCSR.

5.1.3. Memory Mapped I/O Operation

MMIO (Memory Mapped I/O) operation is done by memory access through software execution. Flash controller would go through following phases/steps to complete a read transaction.

1. Command Phase: (8 bit)
   - Output command code, which is specified in SFCR2.
   - Serial, dual or quad I/O is selected in SFCR2
     - Serial I/O: 8 cycles
     - Dual I/O: 4 cycles
     - Quad I/O: 2 cycles
2. Address Phase: (24 bit)

- Output SPI-NOR Flash I/O address, which is translated from SPI-NOR Flash memory address.
- Serial, dual or quad I/O is selected in SFCR2
  - Serial I/O: 24 cycles
  - Dual I/O: 12 cycles
  - Quad I/O: 6 cycles

3. Dummy Phase: (N cycles)
   - Output some cycles (N) of dummy data, which is specified in SFCR2
   - Dummy data shall be 0. (Avoid being recognized to be meaning code)
4. Data Phase: (M bits)
   - Get input data which is driven by flash chip
   - The number of bits (M) is multiple of 8 bits
   - Serial, dual or quad I/O is selected in SFCR2
     - Serial I/O: M cycles
     - Dual I/O: M/2 cycles
     - Quad I/O: M/4 cycles

5.2. Signal – Hardware Pin Mapping

TABLE 5-1 Signal & Hardware Pin Name Mapping

SPI-NOR FLASH Control			
Signal Name	**Hardware Pin Name**	**Type**	**Description**
CS0#	SF_CS#[0]	O	SPI Serial Flash chip select 0.
CS1#	SF_CS#[1]	O	SPI Serial Flash chip select 1.
SIO0 (SI)	SF_SIO0 (SF_SDI)	I/O	In single IO mode: SPI serial flash serial data input pin. (It is **input** for SPI-NOR Flash chip; it is **output** for the flash controller.) In multi-IO mode: SPI serial flash **bidirectional** data pin.
SIO1 (SO)	SF_SIO1 (SF_SDO)	I/O	In single IO mode: SPI serial flash serial data output. (It is **output** for SPI-NOR Flash chip; it is **input** for the flash controller.) In multi-IO mode: SPI serial flash **bidirectional** data pin.
SIO2	SF_SIO2	I/O	In multi-IO mode: SPI serial flash bidirectional data pin.
SIO3	SF_SIO3	I/O	In multi-IO mode: SPI serial flash bidirectional data pin.
SCK	SF_SCK	O	SPI Serial Flash Serial clock output The SF_SDI will be drive on the falling edge, The SF_SDO will be latch on the rising edge.
RESET#	RESET#	I/O	Hardware Reset Pin Active Low

[Notes]

1. Pin sharing is not taken into consideration here.

5.3. *Register Set (Base Address: 0xB800_1200)*

Base Address: 0xB800_1200			
Offset	**Size (byte)**	**Name**	**Description**
00	4	SFCR	SPI-NOR Flash Configuration Register

Base Address: 0xB800_1200			
Offset	**Size (byte)**	**Name**	**Description**
04	4	SFCR2	SPI-NOR Flash Configuration Register 2
08	4	SFCSR	SPI-NOR Flash Control & Status Register
0C	4	SFDR	SPI-NOR Flash Data Register
10	4	SFDR2	SPI-NOR Flash Data Register 2
14	4	SFRDCR	SPI-NOR Flash RX Delay Control Register
18	4	SFRDR	SPI-NOR Flash RX Delay Register 0
1C	4	SFPPR	SPI Flash Persistent Property Register ()

5.3.1. SPI-NOR Flash Configuration Register (SFCR) (0xB800_1200)

This configuration register is used both for PIO and MMIO.

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:29	SPI_CLK_DIV	SPI operating clock rate selection. The value defines the divisor to generate SPI clock SPIFSCK = CLK_SPIF / SPI_CLK_DIV 000 : DIV = 2 001 : DIV = 4 010 : DIV = 6 011 : DIV = 8 100 : DIV = 10 101 : DIV = 12 110 : DIV = 14 111 : DIV = 16	R/W (pio) (mmio)	111B
28	RBO	Serial Flash Read Byte Ordering. 0: The byte order is from low to high 1: The byte order is from high to low	R/W (pio) (mmio)	1B
27	WBO	Serial Flash Write Byte Ordering. 0: The byte order is from low to high 1: The byte order is from high to low	R/W (pio) (mmio)	1B
26:22	SPI_TCS	SPI chip deselect time. Basic unit = 1 * DRAM clock cycle 00000 means 1 unit, 00001 means 2 units, etc.	R/W (pio) (mmio)	11111B
21	*must_be_zero*	*Previously, this bit is for SPI_DIV_DIS. Though it is not functional anymore, one should not use this bit for the sake for software compatibility.*	R	0
20:13	*must_be_zero*	*All bits must be 0*	R	0
12	OCP0_FRQ_SLOWER	Report whether the OCP0 clock <= CLK_SPIF. 0: OCP0 clock > CLK_SPIF 1: OCP0 clock <= CLK_SPIF Note: It is required to setup this strap pin statically for the running environment to keep the integrity of the memory operations. We also need to set proper value by software before the OCP0/SPIF frequencies changes on runtime.	R (System H/W Pin)	0
11:10	*must_be_zero*	*All bits must be 0*	R	0
9	HW_RESET_EN	Indicates whether triggering hardware reset signal or not before booting from SPI flash. 0: Don't trigger hardware reset signal. 1: Trigger hardware reset signal.	R	1B
8	RST_CMD_DIS	Indicates whether issuing 0x66, 0x99 SPI flash commands or not when before booting from SPI flash. 0: Issue 0x66, 0x99 SPI flash commands. 1: Issue no reset command.	R (HW Strapping pin)	1B

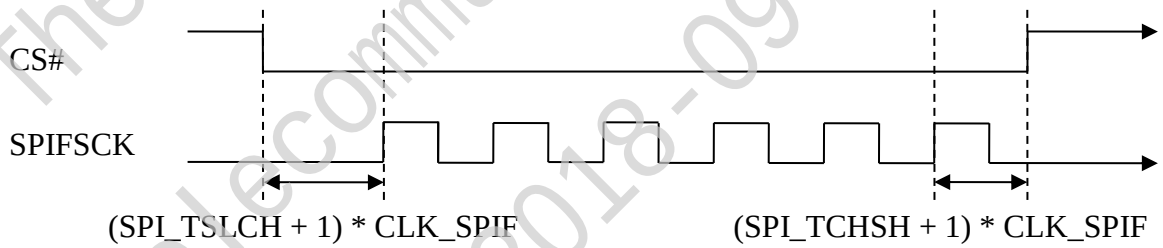
Bit	Name	Description	Mode	Default
7:4	SPI_TCHSH	SPI CS# active hold time (relative to SPIFSCK) Basic unit = 1 * CLK_SPIF 00000 means 1 unit, 00001 means 2 units, etc.	R/W (pio) (mmio)	0100B
3:0	SPI_TSLCH	SPI CS# active setup time. Basic unit = 1 * CLK_SPIF 00000 means 1 unit, 00001 means 2 units, etc.	R/W (pio) (mmio)	0100B

[Design Notes]

1. When booting up from SPI-NOR Flash, memory mapped I/O interface is through access to SPI Region 1 and SPI Region 2.
2. SFSIZE is only useful in memory mapped read mode.
   - CS0# is asserted when 0 * SFSIZE <= memory address < 1 * SFSIZE
   - CS1# is asserted when 1 * SFSIZE <= memory address < 2 * SFSIZE
   - CS2# is asserted when 2 * SFSIZE <= memory address < 3 * SFSIZE
   - CS3# is asserted when 3 * SFSIZE <= memory address < 4 * SFSIZE
   - Memory address exceeding 4 * SFSIZE will wrap back, and still map to CS0# and CS1#.
3. In current design, SPI_TCS is imposed when CS# goes high from low, no matter for SPI read or SPI write transactions.
4. Since ESD or Surge test may cause SoC power-on-reset resets but SPI-F does not (and vice versa), HW_RESET_EN and RST_CMD_EN must be handled by either strapping pin or bonding.

[Software Notes]

1. When RD_OPT is turned on in memory mapped read mode, CS is manipulated by SPI flash controller and may be left either in selected state or deselected state. In order to ensure correct program I/O operation at a later time, a CS toggle (deselected->selected->deselected) must be issued by software prior to any program I/O operation. This applies even when debugging with ICE.
2. A known FPGA issue is when debugging with ICE memory-mapped read operation may be unstable. This is due to the very long turn-around time of SPI flash memory-mapped read operation which may cause ICE timeout. A possible solution is to make higher the SPI flash clock frequency by setting SPI_CLK_DIV = 0.
3. There is a known bug that switching RBO between 0 and 1 causes SPI-F controller returns bad data.
4. Following figure shows how SPI_TCHSH and SPI_TSLCH affect CS# and SPIFSCK:



5.3.2. SPI-NOR Flash Configuration Register 2 (SFCR2) (0xB800_1204)

This configuration register is only used for memory mapped I/O.

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	SFCMD	SPI-NOR Flash 8-bit command code of a read transaction. Ex. “Read Data” is 0x03. “Fast Read” is 0x0B.	R/W (mmio)	0x03H

Bit	Name	Description	Mode	Default
23:21	SFSIZE	SPI-NOR Flash size. **3-Byte Mode**: 000: 128Kbyte 001: 256Kbyte 010: 512Kbyte 011: 1Mbyte 100: 2Mbyte 101: 4Mbyte 110: 8Mbyte 111: 16Mbyte **4-Byte Mode**: 000: 512Kbyte 001: 1Mbyte 010: 2Mbyte 011: 4Mbyte 100: 8Mbyte 101: 16Mbyte 110: 32Mbyte 111: 64Mbyte	R/W (mmio)	111B
20	RD_OPT	SPI-NOR Flash sequential access optimization. 0: No optimization 1: Optimization for sequential access	R/W (mmio)	0B
19:18	CMD_IO	SPI-NOR Flash I/O mode selection for the command phase of a read transaction 00: Serial I/O (8 cycles) 01: Dual I/O (4 cycles) 10: Quad I/O (2 cycles) 11: reserved	R/W (mmio)	00B
17:16	ADDR_IO	SPI-NOR Flash I/O mode selection for the address phase of a read transaction 00: Serial I/O (24 cycles) 01: Dual I/O (12 cycles) 10: Quad I/O (6 cycles) 11: reserved	R/W (mmio)	00B
15:13	DUMMY_CYCLES	SPI-NOR Flash inserted dummy cycles for the dummy cycle phase of a read transaction 000: 0 cycle 001: 2 cycles 010: 4 cycles 011: 6 cycles 100: 8 cycles 101: 10 cycles 110: 12 cycles 111: 14 cycles	R/W (mmio)	000B
12:11	DATA_IO	SPI-NOR Flash I/O mode selection for the data phase of a read transaction (assume 8*N cycles) 00: Serial I/O (8*N cycles) 01: Dual I/O (4*N cycles) 10: Quad I/O (2*N cycles) 11: reserved	R/W (mmio)	00B
10	HOLD_TILL_SFDR2	If this bit is '1', it indicates the write operation to this register (SFCR2) will not take effect immediately but will be delayed until another write operation to SFDR2.	R/W (mmio)	0B
9	ADDR_Mode	0: 3-byte address mode 1: 4-byte address mode	R/W (mmio)	0B
8:0	*must_be_zero*	*All bits must be 0*	R	0

[Design Notes]

1. When RD_OPT is '0', memory controller will deselect CS when a memory mapped read operation completes. When RD_OPT is '1', SPI-NOR Flash's sequential read access will be utilized. That is, memory controller will not actively deselect CS when a memory mapped read operation completes. Therefore if the next memory mapped read operation is consecutive, sequential read access can be applied without issuing instruction code and addresses. And if the next memory mapped read is not consecutive, memory controller will deselect CS and issue a new read transaction.

[Software Notes]

1. It is suggested that software shall ensure CS# is deselected for power saving.
2. When using 4-Byte mode SPI-NOR Flash, the configuration of SFSIZE for MMIO is different from 3-byte mode.

5.3.3. SPI-NOR Flash Control & Status Register (SFCSR) (0xB800_1208)

This configuration register is used both for PIO (programmed I/O) and MMIO (memory mapped I/O).

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	SPI_CSB0	SPI-NOR Flash Chip select 0 (CS0#) 0 = active 1 = not active	R/W (PIO)	1B
30	SPI_CSB1	SPI-NOR Flash Chip select 1 (CS1#) 0 = active 1 = not active	R/W (PIO)	1B
29:28	LEN	SPI read/write data length (unit = byte) 00: 1 byte 01: 2 byte 10: 3 byte 11: 4 byte	R/W (PIO) (MMIO)	11B
27	SPI_RDY	SPI-NOR Flash operation Busy indication flag 0: Busy (operation in progress) 1: Ready (idle or SPI access command is ready)	R (PIO)	1B
26:25	IO_WIDTH	SPI-NOR Flash I/O mode selection of a transaction. 00: Serial I/O 01: Dual I/O 10: Quad I/O 11: reserved	R/W (MMIO)	00B
24	CHIP_SEL	Chip selection 0: CS0# 1: CS1#	R/W (MMIO)	0B
23:16	CMD_BYTE	SPI-NOR Flash 8-bit command code of a transaction. (This field is only used in MMIO mode, when a) Ex. "Read Data" is 0x03. "Read ID" is 0x9F.	R/W (MMIO)	0B
15	*must_be_zero*	*Previously, this bit is for SPI_CSB2. Though it is not functional anymore, one should not use this bit for the sake for software compatibility.*	*R*	*0B*
14	*must_be_zero*	*Previously, this bit is for SPI_CSB3. Though it is not functional anymore, one should not use this bit for the sake for software compatibility.*	*R*	*0B*
13:12	*must_be_zero*	*All bits must be 0.*	R	0
11	SPI_CSB0_STATUS	CS0# status on SPI_CLK domain. 0 = active 1 = not active	R	1
10	SPI_CSB1_STATUS	CS1# status on SPI_CLK domain. 0 = active 1 = not active	R	1

Bit	Name	Description	Mode	Default
9:5	*must_be_zero*	*All bits must be 0.*	R	0
4	SPI_IDLE	0: Not idle 1: Controller is in idle state	R	0
3:1	*must_be_zero*	*All bits must be 0.*	R	0
0	MIO_PIO_ERR	Indicate MMIO and PIO access conflict error. 0: No MMIO and PIO access conflict error. 1: There is a MMIO and PIO access conflict error.	RW1C	0

[Design Notes]

1. Software would write 1/0 to SPI_CSB0 or SPI_CSB1 to select/deselect the target SPI-NOR Flash. Design shall guarantee all write requests will take effect without loss.
2. During a write transaction, memory controller shall make SPI_RDY indicate Busy.
3. Each time CSB0 or CSB1 change state from 0 to 1 (rising edge), memory controller shall make SPI_RDY indicate Busy before SPI_TCS is satisfied.

[Software Notes]

1. Software must ensure SPI_RDY indicate Ready before selecting SPI-NOR Flash and before doing read/write transactions.
2. When software executes in SPI-NOR Flash only mode by utilizing memory mapped I/O interface, no programmed I/O is allowed in order to prevent unexpected results.

5.3.4. SPI-NOR Flash Data Register (SFDR) (0xB800_120C)

This configuration register is only used for PIO (programmed I/O).

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	Data3	Read/write data byte 3	R/W	0B
23:16	Data2	Read/write data byte 2	R/W	0B
15:8	Data1	Read/write data byte 1	R/W	0B
7:0	Data0	Read/write data byte 0	R/W	0B

[Design Notes]

1. If the RBO (Read Byte Ordering) or the WBO (Write Byte Ordering) fields of SFCR (SPI-NOR Flash Configuration Register) are from high to low, memory controller shall do read/write byte access in the order {Data3, Data2, Data1, Data0}. If low to high, the byte access order is {Data0, Data1, Data2, Data3}.

5.3.5. SPI-NOR Flash Data Register 2 (SFDR2) (0xB800_1210)

This configuration register is only used for memory mapped I/O.

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	Data3	Read/write data byte 3	R/W	0B
23:16	Data2	Read/write data byte 2	R/W	0B
15:8	Data1	Read/write data byte 1	R/W	0B
7:0	Data0	Read/write data byte 0	R/W	0B

[Design Notes]

1. This register – SFDR2 and newly-added fields of SFCSR, come to exist mainly for issuing simple SPI transaction while running in MMIO mode since existing SFCSR and SFDR do not serve the purpose. For safety reason, the default timing setting of SPI-NOR Flash controller is relatively slow, so software would set SFCR1 and SFCR2 for high speed read and multi-IO read. In order to achieve this goal, software firstly need to identify flash chips (via RDID command) and secondly might issue necessary commands to enable some features.
2. For example:
   - SST's SST26VF016 needs an EQIO (Enable Quad I/O) command before quad I/O feature is enabled. - Winbond's W25Q80 needs a HPM (High Performance Mode) command before any dual or quad I/O instruction can run at high frequency.
3. If the RBO (Read Byte Ordering) or the WBO (Write Byte Ordering) fields of SFCR (SPI-NOR Flash Configuration Register) are from high to low, memory controller shall do read/write byte access in the order {Data3, Data2, Data1, Data0}. If low to high, the byte access order is {Data0, Data1, Data2, Data3}.

[Software Notes]

1. For most SPI-NOR Flash that supports quad IO, there is a non-volatile QE (Quad Enable) bit in the status register, to indicate whether the shared pins serve as WP# and HOLD#, or as SIO2 and SIO3. It is not feasible to set QE = 1 in flash-only mode by facilitating SFDR2 since the complete procedure takes multiple SPI read/write transaction. It shall be emphasized that the SFDR2 can only be used for simple transaction that takes only one read/write operation.
2. To read ID of 1st flash chip, software shall
   - Set SFCSR
     - CMD_BYTE = 0x9F (RDID)
     - CHIP_SEL = 0B
     - IO_WIDTH = 00B
     - LEN = 4B
   - Read out ID information from SFDR2's BYTE1, BYTE2, BYTE3
3. To issue a complete HPM for Winbond's W25Q80 of the 1st flash chip, software shall
   - Set SFCSR
     - CMD_BYTE = 0xA3 (HPM)
     - CHIP_SEL = 0B
     - IO_WIDTH = 00B
     - LEN = 4B
   - Write 0 to SFDR2's BYTE1, BYTE2, BYTE3, for 3 dummy bytes
4. To enable quad I/O mode for SST's SST26VF016 of the 1st flash chip, software shall
   - Set SFCSR
     - CMD_BYTE = 0x38 (EQIO)
     - CHIP_SEL = 0B
     - IO_WIDTH = 00B
     - LEN = 1B
   - **Set SFCR2's CMD_IO, ADDR_IO and DATA_IO to quad I/O mode, and HOLD_TILL_SFDR2 to 1.**
   - Write to SFDR2 (Written value is don't care since LEN=1B). This will cause memory controller to issue EQIO command and also memory controller itself will enter the corresponding quad I/O mode.

5.3.6. SPI-NOR Flash RX Delay Register 0 (SFRDR) (0xB800_1218)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0B
28:24	IO3_Delay	Selection of 32 taps delay line for SPI-NOR Flash RX IO3 delay. 00000: 1st tap 00001: 2nd tap …. 11111:32nd tap	R/W	0B
23:21	*must_be_zero*	*All bits must be 0*	R	0B
20:16	IO2_Delay	Selection of 32 taps delay line for SPI-NOR Flash RX IO2 delay. 00000: 1st tap 00001: 2nd tap …. 11111:32nd tap	R/W	0B
15:13	*must_be_zero*	*All bits must be 0*	R	0B
12:8	IO1_Delay	Selection of 32 taps delay line for SPI-NOR Flash RX IO1 delay. 00000: 1st tap 00001: 2nd tap …. 11111:32nd tap	R/W	0B
7:5	*must_be_zero*	*All bits must be 0*	R	0B
4:0	IO0_Delay	Selection of 32 taps delay line for SPI-NOR Flash RX IO0 delay. 00000: 1st tap 00001: 2nd tap …. 11111:32nd tap	R/W	0B

[Design Notes]

- In order to prevent SPI-NOR Flash I/O timing issue from PAR with loosely constraints, SFRDR is added to adjust IO0~IO3 RX path timing.

5.3.7. SPI Flash Persistent Property Register (SFPPR) (0xB800_121C)

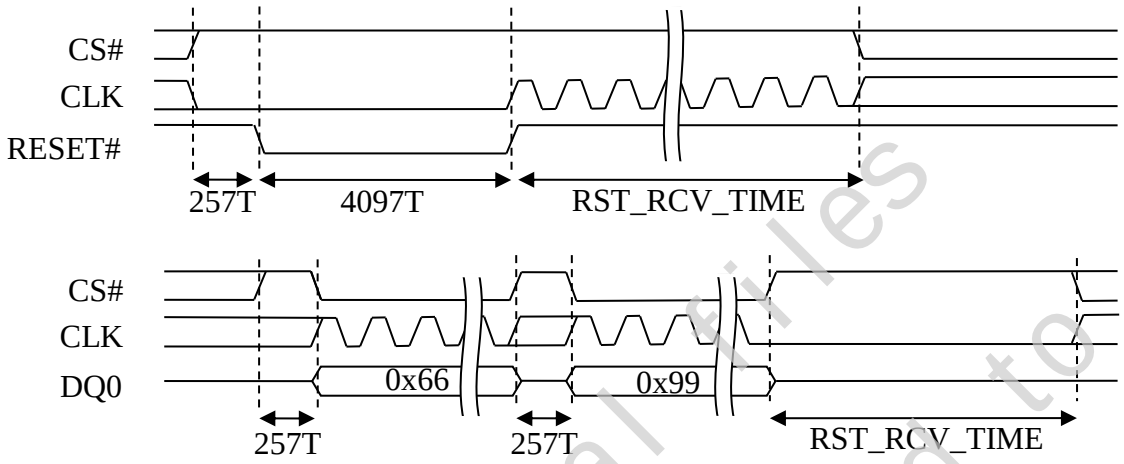
(This register only resets with power cycle)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:4	*must_be_zero*	*All bits must be 0.*	R	0
3:0	RST_RCV_TIME	After both HW and SW reset, SPI-F controller waits reset recovery time before further operations. 0x0: 2^{16} x 5ns 0x1: 2^{17} x 5ns 0x2: 2^{18} x 5ns … 0xF: 2^{31} x 5ns	R/W	0xB

[Design Notes]

1. The following chart illustrates HW and SW reset behaviors:



[Software Notes]

1. Following table highlights some RST_RCV_TIME and the matching recovery time in milliseconds (or seconds).

	RST_RCV_TIME								
	0x0	**0x2**	**0x4**	**0x6**	**0x8**	**0xA**	**0xC**	**0xE**	**0xF**
200MHz	0.33	1.31	5.25	20.97	83.89	335.55	1.34(s)	5.37(s)	10.74(s)

6. UART Specification

6.1. *Features*

Two 16C550 compatible UARTs are provided, and each contains a 16-byte First-In-First-Out (FIFO) buffer. In addition, auto flow control is provided, in which auto-CTS mode (CTS controls transmission) and auto-RTS mode (Receiver FIFO contents and threshold control RTS) are both supported. The baud rate can be up to 1Mbps and a programmable baud rate generator allows division of any input reference clock by 1 to (2^16-1) and generates an internal 16x clock. And a fully programmable serial interface is provided, it can be configured to support 6, 7, or 8 bit characters, even, odd, no parity generation and detection, and 1 or 2 stop bit generation. Fully prioritized interrupt control and loopback functionality for diagnostic capability are also provided.

6.1.1. Interface Pins

The UART interface pins are shown in the following table.
(Note: 2 UART interfaces support)

TABLE 6-1 UART Control Interface Pins

Signal Name	Type	Function
TXD#	O	Transmit Data.
RXD#	I	Receive Data.
RTS#	O	Request To Send.
CTS#	I	Clear To Send.

6.1.2. Baud Rate

Divisor Latch = $(System\ Clock)/(16 \times BaudRate) - 1$

System Clock = Lexra Bus Clock

6.1.3. Time-out Interrupt

A time-out interrupt occurs when the following conditions exists in at least one byte in the active descriptor buffer or receiver FIFO.
The most recent serial character was received more than four character times ago.

6.1.4. Auto-flow Control

Auto-RTS — When the receiver FIFO reaches the trigger level, the RTS# is de-asserted. RTS# is automatically re-asserted once the FIFO is emptied. If RTS# is de-asserted, the UART must receive the incoming data until the FIFO is full.

Auto-CTS — The transmitter checks CTS# before sending the next data byte. When CTS# is active, the transmitter sends the next data byte, otherwise it stops the transmission.

6.1.5. Loopback Diagnostic

When the LOOP bit is set, the following occurs.
TXD# still transmits the normal signal output.
RXD# is disconnected.

CTS#, DSR#, DCD#, and RI# are disconnected.
The output of the transmitter shift register is looped back into the input of the receiver shift register.
The MCR's DTR, RTS, OUT1, and OUT2 bits are internally connected to CTS#, DSR#, DCD#, and RI# respectively.
The DTR#, RTS#, OUT1#, and OUT2# pins are forced high.

6.1.6. Interrupt Priority

Interrupt Identification Register				Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Method
Bit3	Bit2	Bit1	Bit0				
0	0	0	1	None	None	None	None
0	1	1	0	1	Receiver line status	Overrun, parity, framing errors or break	Read LSR
0	1	0	0	2	Received data available	DR bit is set	Read RBR
1	1	0	0	2	Character time-out indication	No characters have been removed from or input to FIFO during the last character times and at 1 character in it.	Read RBR
0	0	1	0	3	Transmitter holding register empty	THRE bit set	Reading IIR or write THR
0	0	0	0	4	Modem status	CTS#, DSR#, RI#, DCD#	Reading MSR

6.1.7. Loopback Diagnostic

A break characters is an all-zero character whose length is start bit (1) + word length (7 or 8) + parity bit (0 or 1) + stop bits (1 or 2). The break event occurs after the first break character is received.

6.2. *Register Set (Base Address: 0xB800_2000)*

Base Address: 0xB800_2000				
Offset	Size (byte)	Name	Description	Access
000	1	UART0_RBR	Receiver buffer register (DLAB=0).	R
000	1	UART0_THR	Transmitter holding register (DLAB=0).	W
000	1	UART0_DLL	Divisor latch LSB (DLAB=1).	R/W
004	1	UART0_IER	Interrupt enable register (DLAB=0).	R/W
004	1	UART0_DLM	Divisor latch MSB (DLAB=1).	R/W
008	1	UART0_IIR	Interrupt identification register.	R
008	1	UART0_FCR	FIFO control register.	W
00c	1	UART0_LCR	Line control register.	R/W
010	1	UART0_MCR	Modem control register.	R/W
014	1	UART0_LSR	Line status register.	R/W
018	1	UART0_MSR	Modem status register.	R/W
01c	1	UART0_SCR	Scratch register.	R/W
100	1	UART1_RBR	Receiver buffer register (DLAB=0).	R
100	1	UART1_THR	Transmitter holding register (DLAB=0).	W
100	1	UART1_DLL	Divisor latch LSB (DLAB=1).	R/W
104	1	UART1_IER	Interrupt enable register (DLAB=0).	R/W
104	1	UART1_DLM	Divisor latch MSB (DLAB=1).	R/W
108	1	UART1_IIR	Interrupt identification register.	R
108	1	UART1_FCR	FIFO control register.	W

Base Address: 0xB800_2000				
Offset	**Size (byte)**	**Name**	**Description**	**Access**
10c	1	UART1_LCR	Line control register.	R/W
110	1	UART1_MCR	Modem control register.	R/W
114	1	UART1_LSR	Line status register.	R/W
118	1	UART1_MSR	Modem status register.	R/W
11c	1	UART1_SCR	Scratch register.	R/W

6.2.1. UART Receiver Buffer Register (DLAB=0) (UART0_RBR, UART1_RBR) (0xB800_2000, 0xB800_2100)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	RBR[7:0]	Receiver buffer data.	R	00H
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.2. UART Transmitter Holding Register (DLAB=0) (UART0_THR~UART1_THR) (0xB800_2000, 0xB800_2100)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	THR[7:0]	Transmitter holding data.	W	00H
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.3. UART Divisor Latch LSB (DLAB=1) (UART0_DLL, UART1_DLL) (0xB800_2000, 0xB800_2100)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	DLL[7:0]	Divisor latch LSB.	W/R	00H
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.4. UART Divisor Latch MSB (DLAB=1) (UART0_DLM, UART1_DLM) (0xB800_2004, 0xB800_2104)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	DLM[7:0]	Divisor latch MSB.	W/R	00H
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.5. UART Interrupt Enable Register (UART0_IER, UART1_IER) (0xB800_2004, 0xB800_2104)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	*must_be_zero*	*All bits must be 0.*	R	0
29	ELP	Low power mode enable.	R/W	0B
28	ESLP	Sleep mode enable.	R/W	0B
27	EDSSI	Enable modem status register interrupt.	R/W	0B

26	ELSI	Enable receiver line status interrupt.	R/W	0B
25	ETBEI	Enable transmitter holding register empty interrupt.	R/W	0B
24	ERBI	Enable received data available interrupt.	R/W	0B
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.6. UART Interrupt Identification Register (UART0_IIR, UART1_IIR) (0xB800_2008, 0xB800_2108)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	FIFO16[1:0]	00=No FIFO 11=16-byte FIFO	R	11B
29:28	*must_be_zero*	*All bits must be 0.*	R	0
27:25	IID[2:0]	Interrupt ID. IID[1:0] indicates the interrupt priority.	R	000B
24	IPND	Interrupt pending. 0=Interrupt pending	R	1B
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.7. UART Interrupt Identification Register (UART0_FCR, UART1_FCR) (0xB800_2008, 0xB800_2108)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	RTRG[1:0]	Receiver Trigger level. Trigger level: 16-byte 00=01, 01=04, 10=08, 11=14	W	11B
29:27	*must_be_zero*	*All bits must be 0.*	R	0
26	TFRST	Transmitter FIFO Reset. Writes 1 to clear the transmitter FIFO.	W	0B
25	RFRST	Receiver FIFO Reset. Writes 1 to clear the receiver FIFO.	W	0B
24	EFIFO	Enable FIFO. When this bit is set, enables the transmitter and receiver FIFOs. Changing this bit clears the FIFOs.	W	0B
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.8. UART Line Control Register (UART0_LCR, UART1_LCR) (0xB800_200C, 0xB800_210C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	DLAB	Divisor Latch Access Bit.	R/W	0B
30	BRK	Break control. Set this bit force TXD to the spacing (low) state (break). Clear this bit to disable break condition.	R/W	0B
29:28	EPS[1:0]	Even Parity Select. 00=odd parity 01=even parity 10=mark parity 11=space parity	R/W	00B
27	PEN	Parity Enable.	R/W	0B

Bit	Name	Description	Mode	Default
26	STB	Number of Stop Bits. 0=1 bit 1=2 bits	R/W	0B
25	WLS_1	Tied to 1.	RO	1B
24	WLS_0	Word Length Select. Combining with WLS_1: 10=7 bits 11=8 bits	R/W	1B
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.9. UART Modem Control Register (UART0_MCR, UART1_MCR) (0xB800_2010, 0xB800_2110)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	*must_be_zero*	*All bits must be 0.*	R	0
30	LXCLK_SEL	Uart/timer clock select 0 = peripheral clk 1 = fixed 200MHz clk	W	0B
29	AFE	Auto Flow control Enable	R/W	0B
28	LOOP	Loopback	R/W	0B
27	OUT2	Out 2	R/W	0B
26	OUT1	Out 1	R/W	0B
25	RTS	Request To Send. 0=Set RTS# high 1=Set RTS# low	R/W	0B
24	DTR	Data Terminal Ready. 0=Set DTR# high 1=Set DTR# low	R/W	0B
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.10. UART Line Status Register (UART0_LSR, UART1_LSR) (0xB800_2014, 0xB800_2114)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	RFE	Receiver FIFO Error. Either a parity, framing, or break error in the FIFO.	R	0B
30	TEMT	Transmitter Empty. Character mode: Both THR and TSR are empty. FIFO mode: Both transmitter FIFO and TSR are empty	R	1B
29	THRE	Transmitter Holding Register Empty. Character mode: THR is empty FIFO mode: Transmitter FIFO is empty	R	1B
28	BI	Break Interrupt indicator.	R	0B
26	PE	Parity Error.	R	0B
25	OE	Overrun Error. An overrun occurs when the receiver FIFO is full and the next character is completely received in the receiver shift register. An OE is indicated. The character in the shift register will be overwritten.	R	0B
24	DR	Data Ready. Character mode: Data ready in RBR FIFO mode: Receiver FIFO is not empty.	RC	0B
23:0	*must_be_zero*	*All bits must be 0.*	R	0

6.2.11. UART Modem Status Register (UART0_MSR, UART1_MSR) (0xB800_2018, 0xB800_2118)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	DCD	Data Carrier Detect. 0=DCD# detected high 1=DCD# detected low In loopback mode, returns bit 2 of MCR. In normal mode, returns 1.	R	0B
30	RI	Ring Indicator. 0=RI# detected high 1=RI# detected low In loopback mode, returns bit 3 of MCR. In normal mode, returns 0.	R	0B
29	DSR	Data Set Ready. 0=DSR # detected high 1=DSR# detected low In loopback mode, returns bit 0 of MCR In normal mode, returns 1.	R	0B
28	CTS	Clear To Send. 0=CTS# detected high 1=CTS# detected low	R	0B
27	DDCD	Delta data carrier detect. DCD# signal transmits. Returns 0.	R	0B
26	TERI	Trailing edge ring indicator. RI# signal changes from low to high. Returns 0.	R	0B
25	DDSR	Delta data set ready. DSR# signal transmits. Returns 0.	R	0B
24	DCTS	Delta clear to send. CTS# signal transmits	R	0B
23:0	*must_be_zero*	*All bits must be 0.*	R	0B

7. Timer

7.1. *Features*

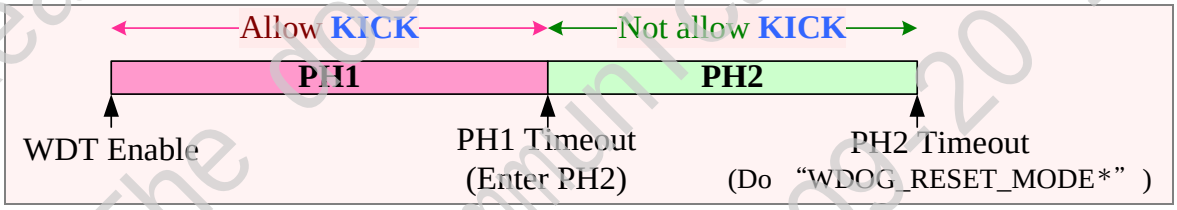
The chip provides 5 sets of hardware timers and one watchdog timer. Each timer can be configured as timer mode or counter mode. Counter mode means the timer only times-out once. The initial time-out values are configured via TCxDATA register. The current count values are shown in TCxCNT register. TCxCTL defines the base clock for counting, which is based on a multiple of the system clock. TCIR controls the interrupt resulting from timer time-out. The Watchdog timer is controlled by WDTCNR.

7.1.1. Introduction

- The new WDT architecture is doing two-phase timeout mechanism.
  - Phase one (PH1) timeout
  - Phase two (PH2) timeout

- In PH1:
  - If the system is alive, then it can "**KICK**" WDOG to reset.
  - If WDT is timeout in PH1, then it will enter PH2.
    - If "WDT_PH1TO_IE" is enabled and "WDT_PH1TO_RS" is set to "NMI" mode, then PH1_IP will trigger S/W reset and will not enter PH2.

- In PH2:
  - This is the reserved period for storing error information and current status.
  - "**KICK**" is not allowed in PH2
  - H/W or S/W reset is guaranteed to be triggered after PH2 WDOG timeout.
    - Depends on the setting of WDT_RESET_Mode*.
  - For dual processor, this H/W WDT is only applied to the master CPU use.



7.1.2. WDT Counter Introduction

Register Name	Description
WDT_CLK_CNTR [27:0]	➢ A 28-bit auto up-counter per cycle with reset. ➢ Clock domain: LX bus clock. ➢ When it is overflow (which means that it counts to the threshold "**WDT_CLK_SC***"), it will trigger WDT_TO_CNTR to add "1". ➢ Used both in PH1 and PH2.
WDT_TO_CNTR [4:0]	➢ A 5-bit up-counter. ➢ The overflow threshold is set as the register "**PH1_TO*** / **PH2_TO***". ➢ Reset: ■ (1) Only in PH1 is clearable.

	■ (2) By WDT_KICK* ➢ Used both in PH1 and PH2.

[Design Notes]

1. *: LX Bus accessible register

7.1.3. WDOG Initialization Steps

1. System initialization
2. Disable WDT_E* (For safety, doing this step to clear WDOG counter and remained status.)
3. Set WDOG reset mode and timeout threshold:
    1. WDT_RESET_Mode*
    2. WDT_CLK_SC*
    3. PH1_TO* and PH2_TO*
4. Set interrupt routing: WDT_PH1TO_RS*
5. Enable WDOG interrupt:
    - ➢ Clear PH1_IP*
    - ➢ Clear PH2_IP*
    - ➢ Enable WDT_PH1TO_IE* (Optional)
6. Enable WDOG Timer
    - ➢ Enable WDT_E*

7.2. *Register Set (Base Address: 0xB800_3200)*

Base Address: 0xB800_3200			
Offset	**Size (byte)**	**Name**	**Description**
0x00	4	TC0DATA	Timer/Counter 0 Data Register. It specifies the time-out duration.
0x04	4	TC0CNT	Timer/Counter 0 Counter Register.
0x08	4	TC0CTL	Timer/Counter 0 Control Register.
0x0C	4	TC0INT	Timer/Counter 0 Interrupt Register
0x10	4	TC1DATA	Timer/Counter 1 Data Register. It specifies the time-out duration.
0x14	4	TC1CNT	Timer/Counter 1 Counter Register.
0x18	4	TC1CTL	Timer/Counter 1 Control Register.
0x1C	4	TC1INT	Timer/Counter 1 Interrupt Register
0x20	4	TC2DATA	Timer/Counter 2 Data Register. It specifies the time-out duration.
0x24	4	TC2CNT	Timer/Counter 2 Counter Register.
0x28	4	TC2CTL	Timer/Counter 2 Control Register.
0x2C	4	TC2INT	Timer/Counter 2 Interrupt Register
0x30	4	TC3DATA	Timer/Counter 3 Data Register. It specifies the time-out duration.
0x34	4	TC3CNT	Timer/Counter 3 Counter Register.
0x38	4	TC3CTL	Timer/Counter 3 Control Register.
0x3C	4	TC3INT	Timer/Counter 3 Interrupt Register
0x40	4	TC4DATA	Timer/Counter 4 Data register. It specifies the time-out duration.
0x44	4	TC4CNT	Timer/Counter 4 Counter Register.
0x48	4	TC4CTL	Timer/Counter 4 Control Register.
0x4C	4	TC4INT	Timer/Counter 4 Interrupt Register
0x50	4	TC5DATA	Timer/Counter 5 Data register. It specifies the time-out duration.
0x54	4	TC5CNT	Timer/Counter 5 Counter Register.

Base Address: 0xB800_3200			
Offset	**Size (byte)**	**Name**	**Description**
0x58	4	TC5CTL	Timer/Counter 5 Control Register.
0x5C	4	TC5INT	Timer/Counter 5 Interrupt Register
0x60	4	WDTCNTRR	Watchdog Timer Counter Register.
0x64	4	WDTINTRR	Watchdog Timer Interrupt Register.
0x68	4	WDTCTRLR	Watchdog Timer Control Register.
0x70	4	TC6DATA	Timer/Counter 6 Data register. It specifies the time-out duration.
0x74	4	TC6CNT	Timer/Counter 6 Counter Register.
0x78	4	TC6CTL	Timer/Counter 6 Control Register.
0x7C	4	TC6INT	Timer/Counter 6 Interrupt Register
0x80 0xFC	-	-	Reserved for Timer/Counter 7~15

7.2.1. Timer/Counter 0 Data Register (TC0DATA) (0xB800_3200)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC0Data[27:0]	The timer or counter initial value. 0 and 1 not allowed	R/W	0

7.2.2. Timer/Counter 0 Counter Register (TC0CNT) (0xB800_3204)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC0Value[27:0]	The timer or counter value	R	0

7.2.3. Timer/Counter 0 Control Register (TC0CTL) (0xB800_3208)

(This register does not provide byte access)

Bit	Name	Description	R/W	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0
28	TC0En	Timer/Counter 0 enable	R/W	0
27:25	*must_be_zero*	*All bits must be 0.*	R	0
24	TC0Mode	Timer/Counter 0 mode 0=counter mode 1=timer mode	R/W	0
23:16	*must_be_zero*	*All bits must be 0.*	R	0
15:0	TC0DivFactor[15:0]	The divide factor of clock source. Assume DivFactor = N, then Base clock = System_clock /N. Both values 0x0000 and 0x0001 disable the clock. The TC0DivFactor[15:0] field defines the clock base for Timer/Counter 0 counting which is based on the input clock source (Normally Lexra bus clock) which is reflect on the clock control of the whole system.	R/W	0x0000

7.2.4. Timer/Counter 0 Interrupt Register (TC0INT) (0xB800_320C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:21	*must_be_zero*	*All bits must be 0.*	R	0
20	TC0IE	Timer/Counter 0 interrupt enable.	R/W	0
19:17	*must_be_zero*	*All bits must be 0.*	R	0
16	TC0IP	Timer/Counter 0 interrupt pending. Write '1' to clear the interrupt.	R/WC	0
15:0	*must_be_zero*	*All bits must be 0.*	R	0

7.2.5. Timer/Counter 1 Data Register (TC1DATA) (0xB800_3210)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC1Data[27:0]	The timer or counter initial value. 0 and 1 not allowed	R/W	0

7.2.6. Timer/Counter 1 Counter Register (TC1CNT) (0xB800_3214)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC1Value[27:0]	The timer or counter value	R	0

7.2.7. Timer/Counter 1 Control Register (TC1CTL) (0xB800_3218)

(This register does not provide byte access)

Bit	Name	Description	R/W	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0
28	TC1En	Timer/Counter 1 enable	R/W	0
27:25	*must_be_zero*	*All bits must be 0.*	R	0
24	TC1Mode	Timer/Counter 1 mode 0=counter mode 1=timer mode	R/W	0
23:16	*must_be_zero*	*All bits must be 0.*	R	0
15:0	TC1DivFactor[15:0]	The divide factor of clock source. Assume DivFactor = N, then Base clock = System_clock /N. Both values 0x0000 and 0x0001 disable the clock. The TC1DivFactor[15:0] field defines the clock base for Timer/Counter 1 counting which is based on the input clock source (Normally Lexra bus clock) which is reflect on the clock control of the whole system.	R/W	0x0000

7.2.8. Timer/Counter 1 Interrupt Register (TC1INT) (0xB800_321C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:21	*must_be_zero*	*All bits must be 0.*	R	0

20	TC1IE	Timer/Counter 1 interrupt enable.	R/W	0
19:17	*must_be_zero*	*All bits must be 0.*	R	0
16	TC1IP	Timer/Counter 1 interrupt pending. Write '1' to clear the interrupt.	R/WC	0
15:0	*must_be_zero*	*All bits must be 0.*	R	0

7.2.9. Timer/Counter 2 Data Register (TC2DATA) (0xB800_3220)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC2Data[27:0]	The timer or counter initial value. 0 and 1 not allowed	R/W	0

7.2.10. Timer/Counter 2 Counter Register (TC2CNT) (0xB800_3224)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC2Value[27:0]	The timer or counter value	R	0

7.2.11. Timer/Counter 2 Control Register (TC2CTL) (0xB800_3228)

(This register does not provide byte access)

Bit	Name	Description	R/W	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0
28	TC2En	Timer/Counter 2 enable	R/W	0
27:25	*must_be_zero*	*All bits must be 0.*	R	0
24	TC2Mode	Timer/Counter 2 mode 0=counter mode 1=timer mode	R/W	0
23:16	*must_be_zero*	*All bits must be 0.*	R	0
15:0	TC2DivFactor[15:0]	The divide factor of clock source. Assume DivFactor = N, then Base clock = System_clock /N. Both values 0x0000 and 0x0001 disable the clock. The TC2DivFactor[15:0] field defines the clock base for Timer/Counter 2 counting which is based on the input clock source (Normally Lexra bus clock) which is reflect on the clock control of the whole system.	R/W	0x0000

7.2.12. Timer/Counter 2 Interrupt Register (TC2INT) (0xB800_322C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:21	*must_be_zero*	*All bits must be 0.*	R	0
20	TC2IE	Timer/Counter 2 interrupt enable.	R/W	0
19:17	*must_be_zero*	*All bits must be 0.*	R	0
16	TC2IP	Timer/Counter 2 interrupt pending. Write '1' to clear the interrupt.	R/WC	0
15:0	*must_be_zero*	*All bits must be 0.*	R	0

7.2.13. Timer/Counter 3 Data Register (TC3DATA) (0xB800_3230)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC3Data[27:0]	The timer or counter initial value. 0 and 1 not allowed	R/W	0

7.2.14. Timer/Counter 3 Counter Register (0xB800_3234)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC3Value[27:0]	The timer or counter value	R	0

7.2.15. Timer/Counter 3 Control Register (TC3CTL) (0xB800_3238)

(This register does not provide byte access)

Bit	Name	Description	R/W	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0
28	TC3En	Timer/Counter 3 enable	R/W	0
27:25	*must_be_zero*	*All bits must be 0.*	R	0
24	TC4Mode	Timer/Counter 3 mode 0=counter mode 1=timer mode	R/W	0
23:16	*must_be_zero*	*All bits must be 0.*	R	0
15:0	TC3DivFactor[15:0]	The divide factor of clock source. Assume DivFactor = N, then Base clock = System_clock /N. Both values 0x0000 and 0x0001 disable the clock. The TC3DivFactor[15:0] field defines the clock base for Timer/Counter 3 counting which is based on the input clock source (Normally Lexra bus clock) which is reflect on the clock control of the whole system.	R/W	0x0000

7.2.16. Timer/Counter 3 Interrupt Register (TC3INT) (0xB800_323C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:21	*must_be_zero*	*All bits must be 0.*	R	0
20	TC3IE	Timer/Counter 3 interrupt enable	R/W	0
19:17	*must_be_zero*	*All bits must be 0.*	R	0
16	TC3IP	Timer/Counter 3 interrupt pending. Write '1' to clear the interrupt.	R/WC	0
15:0	*must_be_zero*	*All bits must be 0.*	R	0

7.2.17. Timer/Counter 4 Data Register (TC4DATA) (0xB800_3240)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC4Data[27:0]	The timer or counter initial value. 0 and 1 not allowed	R/W	0

7.2.18. Timer/Counter 4 Counter Register (TC4CNT) (0xB800_3244)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC4Value[27:0]	The timer or counter value	R	0

7.2.19. Timer/Counter 4 Control Register (TC4CTL) (0xB800_3248)

(This register does not provide byte access)

Bit	Name	Description	R/W	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0
28	TC4En	Timer/Counter 4 enable	R/W	0
27:25	*must_be_zero*	*All bits must be 0.*	R	0
24	TC4Mode	Timer/Counter 4 mode 0=counter mode 1=timer mode	R/W	0
23:16	*must_be_zero*	*All bits must be 0.*	R	0
15:0	TC4DivFactor[15:0]	The divide factor of clock source. Assume DivFactor = N, then Base clock = System_clock /N. Both values 0x0000 and 0x0001 disable the clock. The TC4DivFactor[15:0] field defines the clock base for Timer/Counter 4 counting which is based on the input clock source (Normally Lexra bus clock) which is reflect on the clock control of the whole system.	R/W	0x0000

7.2.20. Timer/Counter 4 Interrupt Register (TC4INT) (0xB800_324C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:21	*must_be_zero*	*All bits must be 0.*	R	0
20	TC4IE	Timer/Counter 4 interrupt enable.	R/W	0
19:17	*must_be_zero*	*All bits must be 0.*	R	0
16	TC4IP	Timer/Counter 4 interrupt pending. Write '1' to clear the interrupt.	R/WC	0
15:0	*must_be_zero*	*All bits must be 0.*	R	0

7.2.21. Timer/Counter 5 Data Register (TC5DATA) (0xB800_3250)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC5Data[27:0]	The timer or counter initial value. 0 and 1 not allowed	R/W	0

7.2.22. Timer/Counter 5 Counter Register (TC5CNT) (0xB800_3254)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC5Value[27:0]	The timer or counter value	R	0

7.2.23. Timer/Counter 5 Control Register (TC5CTL) (0xB800_3258)

(This register does not provide byte access)

Bit	Name	Description	R/W	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0
28	TC5En	Timer/Counter 5 enable	R/W	0
27:25	*must_be_zero*	*All bits must be 0.*	R	0
24	TC5Mode	Timer/Counter 5 mode 0=counter mode 1=timer mode	R/W	0
23:16	*must_be_zero*	*All bits must be 0*	R	0
15:0	TC5DivFactor[15:0]	The divide factor of clock source. Assume DivFactor = N, then Base clock = System_clock /N. Both values 0x0000 and 0x0001 disable the clock. The TC5DivFactor[15:0] field defines the clock base for Timer/Counter 5 counting which is based on the input clock source (Normally Lexra bus clock) which is reflect on the clock control of the whole system.	R/W	0x0000

7.2.24. Timer/Counter 5 Interrupt Register (TC5INT) (0xB800_325C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:21	*must_be_zero*	*All bits must be 0.*	R	0
20	TC5IE	Timer/Counter 5 interrupt enable.	R/W	0
19:17	*must_be_zero*	*All bits must be 0.*	R	0
16	TC5IP	Timer/Counter 5 interrupt pending. Write '1' to clear the interrupt.	R/WC	0
15:0	*must_be_zero*	*All bits must be 0.*	R	0

7.2.25. Watch Dog Counter Register (WDTCNTRR) (0xB800_3260)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	WDT_KICK	Watch Dog "KICK" register. Write "1" to clear WDT_TO_CNTR. ***Note***: The counter is only clearable in PH1, so that this bit is actually effective only when "WDT_KICK_ALLOW=1".	WC	0
30:0	*must_be_zero*	*All bits must be 0.*	R	0

7.2.26. Watch Dog Interrupt Register (WDTINTRR) (0xB800_3264)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	PH1_IP	PH1 interrupt pending flag. [R] 0: Normal 1: Interrupt pending [WC] Write "1" to clear	R/WC	0
30	PH2_IP	PH1 interrupt pending flag. [R] 0: Normal 1: Interrupt pending [WC] Write "1" to clear	R/WC	0
29:0	*must_be_zero*	*All bits must be 0.*	R	0

7.2.27. Watch Dog Control Register (WDTCTRLR) (0xB800_3268)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	WDT_E	Watch dog timer enable 0: Disable WDT 1: Enable WDT and cause a system reset in PH2 when an overflow signal occurs.	R/W	0B
30:29	WDT_CLK_SC	WDT_CLK_CNTR overflow scale. These bits specify the overflow condition when WDT_CLK_CNTR counts to the value, which will trigger WDT_TO_CNTR to count one. $00_B = 2^{25}$ $01_B = 2^{26}$ $10_B = 2^{27}$ $11_B = 2^{28}$	R/W	00B
28:27	*must_be_zero*	*All bits must be 0.*	R	0
26:22	PH1_TO	PH1 timeout threshold: 00000_B = 1 … 11111_B = 32 ***Note***: This is the overflow threshold for WDT_TO_CNTR in PH1.	R/W	00000B
21:20	*must_be_zero*	*All bits must be 0.*	R	0
19:15	PH2_TO	PH2 timeout threshold: 00000_B = 1 … 11111_B = 32 ***Note***: This is the overflow threshold for WDT_TO_CNTR in PH2.	R/W	00000B
14:2	*must_be_zero*	*All bits must be 0.*	R	0
1:0	WDT_RESET_Mode	Select WDOG reset mode when it is in PH2 timeout. 00_B: H/W Full Chip Reset (Do full reset at PH2_IP rising edge); 01_B: H/W CPU (Only reset CPU) 10_B: S/W Reset * This mode should connect PH2_IP to NMI . 11_B: Reserved.	R/W	00B

7.2.28. Timer/Counter 6 Data Register (TC6DATA) (0xB800_3270)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC6Data[27:0]	The timer or counter initial value. 0 and 1 not allowed	R/W	0

7.2.29. Timer/Counter 6 Counter Register (TC6CNT) (0xB800_3274)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:28	*must_be_zero*	*All bits must be 0.*	R	0
27:0	TC6Value[27:0]	The timer or counter value	R	0

7.2.30. Timer/Counter 6 Control Register (TC6CTL) (0xB800_3278)

(This register does not provide byte access)

Bit	Name	Description	R/W	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0
28	TC6En	Timer/Counter 6 enable	R/W	0
27:25	*must_be_zero*	*All bits must be 0.*	R	0
24	TC6Mode	Timer/Counter 6 mode 0=counter mode 1=timer mode	R/W	0
23:16	*must_be_zero*	*All bits must be 0.*	R	0
15:0	TC6DivFactor[15:0]	The divide factor of clock source. Assume DivFactor = N, then Base clock = System_clock /N. Both values 0x0000 and 0x0001 disable the clock. The TC4DivFactor[15:0] field defines the clock base for Timer/Counter 6 counting which is based on the input clock source (Normally Lexra bus clock) which is reflect on the clock control of the whole system.	R/W	0x0000

7.2.31. Timer/Counter 6 Interrupt Register (TC6INT) (0xB800_327C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:21	*must_be_zero*	*All bits must be 0.*	R	0
20	TC6IE	Timer/Counter 6 interrupt enable.	R/W	0
19:17	*must_be_zero*	*All bits must be 0.*	R	0
16	TC6IP	Timer/Counter 6 interrupt pending. Write '1' to clear the interrupt.	R/WC	0
15:0	*must_be_zero*	*All bits must be 0.*	R	0

[Design Notes]

1. If WDT_RESET_Mode = 01_B or 10_B: WDOG IP will not be reset in these cases, so that the setting of this register will not be changed.

   If WDT_RESET_Mode = 00_B: When doing full chip reset, WDOG IP will also be reset.

2. WDT_CLK_CNTR is a 28-bit auto up-counter, which will counts based on the LX bus clock domain.
3. When WDT_CLK_CNTR counts to the overflow threshold (WDT_CLK_SC), it will trigger WDT_TO_CNTR to count one.
4. When WDT_TO_CNTR counts to the overflow threshold in PH1 (PH1_TO), WDT will enter PH2 and not allow "KICK" anymore.
5. When WDT_TO_CNTR counts to the overflow threshold in PH2 (PH2_TO), it will do "RESET" according to the settings in WDT_RESET_Mode*.
6. The results of different timeout threshold combinations:

LX Bus	WDT_CLK_SC	$Timeout_{min}$	$Timeout_{Max}$
100MHz	2^{28}	2.68s	85.75
	2^{27}	1.34s	42.88
	2^{26}	0.67s	21.44
200MHz	2^{28}	1.34s	42.88s
	2^{27}	0.67s	21.44
	2^{26}	0.34s	10.88
400MHz	2^{28}	0.67s	21.44
	2^{27}	0.32s	10.88
	2^{26}	0.16s	5.12

   - Because the "RESET" should be done after all the jobs on LX bus are finished, so that the total timeout period in PH2 will a little bit more than the value in the above table.
7. When CPU two-stage wakeup is enabled, the following bits will automatically set and clear by cpu_wakeup signal (high active)
   - TC6En: set by rising edge and clear by falling edge
   - TC6IP: clear by falling edge

[Software Notes]

1. This IP can also support only one phase WDOG timeout mechanism, when user enables "WDT_PH1TO_IE" and sets "WDT_PH1TO_RS" to NMI mode. Then if PH1 timeout happens, PH1_IP will trigger S/W and so that WDOG will not enter PH2 stage.
2. After reset, software can check WDT_RESET_Mode for knowing which reset was did last time.
   - WDT_RESET_Mode＝00_B: It was doing full chip reset.
   - WDT_RESET_Mode≠00_B: Referring to this register setting for the reset mode (H/W CPU or S/W reset).
3. TC6 and its delay interrupt is reserved for CPU wakeup, it will use in CPU status is move from sleep to wakeup and reduce CPU frequency to half of the original.

8. GPIO

The GPIO spec is expected to be inherited from RTL8389, with some changes. Since each pin might be a GPIO pin or a single-type peripheral pin, there is no need to incorporate registers to select between different peripheral types for a pin.

- Byte filed arrangement is big endian.
- In order to reduce IP modification overhead, now the register addresses follow the definition of RTL8652, and therefore intentionally leave several unused register addresses as reserved.

8.1. *Features*

General Purpose Input/Output (GPIO) pins could be dedicated pins or are shared with other peripherals. Each GPIO pin might either be bidirectional (input/output) or input only or output only; refer to the following table for details.

The GPIO Data registers may be used to control the signals of GPIO pins. The GPIO Control registers are used as selections between peripheral pins or GPIO pins. All GPIO sets can be used to generate interrupts and the interrupt mask and status register are provided.

8.1.1. Project Specific GPIO information

- Total sets: 4 sets, GPIO A, B, C and D
- Total pins: 32pins

8.2. *Register Set (Base Address: 0xB800_3300)*

Base Address: 0xB800_3300			
Offset	**Size (byte)**	**Name**	**Description**
0x00	4	*Reserved*	
0x04	4	*Reserved*	
0x08	4	PABCD_DIR	Port A,B,C,D direction register
0x0C	4	PABCD_DAT	Port A,B,C,D data register
0x10	4	PABCD_ISR	Port A,B,C,D interrupt status register
0x14	4	PAB_IMR	Port A,B interrupt mode register
0x18	4	PCD_IMR	Port C,D interrupt mode register
0x1C	4	*Reserved*	
0x20	4	*Reserved*	
0x24	4	*Reserved*	
0x28	4	*Reserved*	
0x2C	4	*Reserved*	
0x30	4	*Reserved*	
0x34	4	*Reserved*	
0x38	4		

8.2.1. GPIO Port A,B,C,D Direction Register (PABCD_DIR) (0xB800_3308)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	DRC_A[7:0]	Pin direction configuration of Port A. 0=Configured as input pin 1=Configured as output pin Note: PortA[7:0] as GPIO#7~GPIO#0	R/W	00H

23:16	DRC_B[7:0]	Pin direction configuration of Port B. 0=Configured as input pin 1=Configured as output pin Note: PortB[7:0] as GPIO#15~GPIO#8	R/W	00H
15:8	DRC_C[7:0]	Pin direction configuration of Port C. 0=Configured as input pin 1=Configured as output pin Note: PortC[7:0] as GPIO#23~GPIO#16	R/W	00H
7:0	DRC_D[7:0]	Pin direction configuration of Port D. 0=Configured as input pin 1=Configured as output pin Note: PortC[7:0] as GPIO#31~GPIO#24	R/W	00H

8.2.2. Port A,B,C,D Data Register (PABCD_DAT) (0xB800_330C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	PD_A[7:0]	Pin data of Port A. 0 : Data=0 1 : Data=1	Input pin: R Output pin: R/W	00H
23:16	PD_B[7:0]	Pin data of Port B. 0 : Data=0 1 : Data=1	Input pin: R Output pin: R/W	00H
15:8	PD_C[7:0]	Pin data of Port C. 0 : Data=0 1 : Data=1	Input pin: R Output pin: R/W	00H
7:0	PD_D[7:0]	Pin data of Port D. 0 : Data=0 1 : Data=1	Input pin: R Output pin: R/W	00H

8.2.3. Port A,B,C,D Interrupt Status Register (PABCD_ISR) (0xB800_3310)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	IPS_A[7:0]	Interrupt pending status of port A. Write '1' to clear the interrupt	R/WC	00H
23:16	IPS_B[7:0]	Interrupt pending status of port B. Write '1' to clear the interrupt	R/WC	00H
15:8	IPS_C[7:0]	Interrupt pending status of port C. Write '1' to clear the interrupt	R/WC	00H
7:0	IPS_D[7:0]	Interrupt pending status of port D. Write '1' to clear the interrupt	R/WC	00H

8.2.4. Port A,B Interrupt Mode Register (PAB_IMR) (0xB800_3314)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	PA7_IM[1:0]	PortA.7 interrupt mode. 00=Disable interrupt 01=Enable falling edge interrupt 10=Enable rising edge interrupt 11=Enable both falling or rising edge interrupt	R/W	00B
29:28	PA6_IM[1:0]	PortA.6 interrupt mode.	R/W	00B
27:26	PA5_IM[1:0]	PortA.5 interrupt mode.	R/W	00B
25:24	PA4_IM[1:0]	PortA.4 interrupt mode.	R/W	00B
23:22	PA3_IM[1:0]	PortA.3 interrupt mode.	R/W	00B
21:20	PA2_IM[1:0]	PortA.2 interrupt mode.	R/W	00B

Bit	Name	Description	Mode	Default
19:18	PA1_IM[1:0]	PortA.1 interrupt mode.	R/W	00B
17:16	PA0_IM[1:0]	PortA.0 interrupt mode.	R/W	00B
15:14	PB7_IM[1:0]	PortB.7 interrupt mode.	R/W	00B
13:12	PB6_IM[1:0]	PortB.6 interrupt mode.	R/W	00B
11:10	PB5_IM[1:0]	PortB.5 interrupt mode.	R/W	00B
9:8	PB4_IM[1:0]	PortB.4 interrupt mode.	R/W	00B
7:6	PB3_IM[1:0]	PortB.3 interrupt mode.	R/W	00B
5:4	PB2_IM[1:0]	PortB.2 interrupt mode.	R/W	00B
3:2	PB1_IM[1:0]	PortB.1 interrupt mode.	R/W	00B
1:0	PB0_IM[1:0]	PortB.0 interrupt mode.	R/W	00B

8.2.5. Port C,D Interrupt Mode Register (PCD_IMR) (0xB800_3318)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	PC7_IM[1:0]	PortC.7 interrupt mode. 00=Disable interrupt 01=Enable falling edge interrupt 10=Enable rising edge interrupt 11=Enable both falling or rising edge interrupt	R/W	00B
29:28	PC6_IM[1:0]	PortC.6 interrupt mode.	R/W	00B
27:26	PC5_IM[1:0]	PortC.5 interrupt mode.	R/W	00B
25:24	PC4_IM[1:0]	PortC.4 interrupt mode.	R/W	00B
23:22	PC3_IM[1:0]	PortC.3 interrupt mode.	R/W	00B
21:20	PC2_IM[1:0]	PortC.2 interrupt mode.	R/W	00B
19:18	PC1_IM[1:0]	PortC.1 interrupt mode.	R/W	00B
17:16	PC0_IM[1:0]	PortC.0 interrupt mode.	R/W	00B
15:14	PD7_IM[1:0]	PortD.7 interrupt mode.	R/W	00B
13:12	PD6_IM[1:0]	PortD.6 interrupt mode.	R/W	00B
11:10	PD5_IM[1:0]	PortD.5 interrupt mode.	R/W	00B
9:8	PD4_IM[1:0]	PortD.4 interrupt mode.	R/W	00B
7:6	PD3_IM[1:0]	PortD.3 interrupt mode.	R/W	00B
5:4	PD2_IM[1:0]	PortD.2 interrupt mode.	R/W	00B
3:2	PD1_IM[1:0]	PortD.1 interrupt mode.	R/W	00B
1:0	PD0_IM[1:0]	PortD.0 interrupt mode.	R/W	00B

9. SPI-NAND Flash Controller (NAFC)

9.1. *Feature lists*

- Support 4Gbits SPI-NAND
- Serial/Dual/Quad data width
- PIO and DMA data read/write operation
- Configurable flash access timing

note: We provide one DMA/PIO configurable function to move any length data bytes from spi-nand to target address(address mapping sram/dram) or from target destination to spi-nand

Device Set

Operation Mode	Chunk Sizes	Support
Micron-similar	2112 bytes	O

Note: Micron-similar type, like MT29F1G01AAADD / GD5F1GQ4UAYIG / W25N01GV

9.1.1. NAND Flash Booting Procedure

For SPI NAND flash controller, it will use ROM code to do the booting procedure.

9.1.2. Programmed I/O Operation

PIO (Programmed I/O) operation is done by register access through software execution.

- **PIO-Read**: The smallest PIO read data size is 4 bytes. The read can start from any page offset and stop at any page offset, and no ECC checking is performed. If ECC checking is enabled, the NAND flash read operation (controlled by NAND flash controller) must start at the first byte of the page to the end of page (including spare space). The hardware will run ECC correction and generate a signal of ECER.
- **PIO-Write**: The smallest PIO write data size is 4 bytes. The write can start from any page offset and stop at any page offset, and no ECC is appended to the spare space. If ECC generation is enabled, the write must start at the first byte of the page to the end of page (excluding spare space). The hardware will appends ECC at the spare space and generate a ECER signal.

[Suggested basic software PIO procedure (ex: CECS0, W/O ECC)]
Please refer to the document of "SPI Nand Flash & ECC Software Programming Guide_2013-10-08_v1.05.pdf".

9.1.3. DMA Operation

DMA (Direct Memory Access) operation is supported.

- **DMA-Read**: Every DMA read data size is a page (2080 bytes)(520*4). The destination address for DAM-Read is 4-byte-alignement. If ECC checking is enabled, the hardware will run ECC correction and generate a ECERand transfer 4*520B,{512+6(tag)+2(dummy)} to the destination.
- **DMA-Write**: Every DMA write data size is a page (520B),{512+6(tag)+2(dummy)}. The destination should be 4-byte alignment. If ECC generation is enabled, the hardware will appends ECC at the input data and generate a ECER.

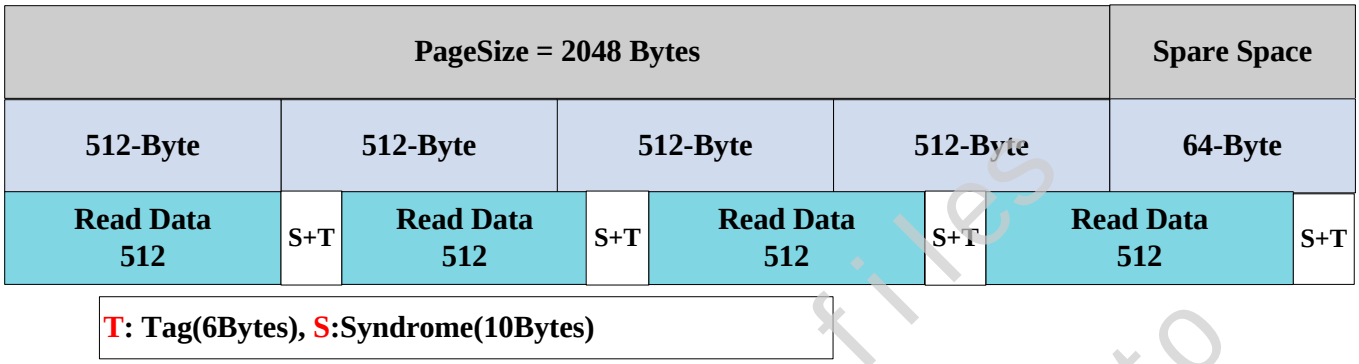
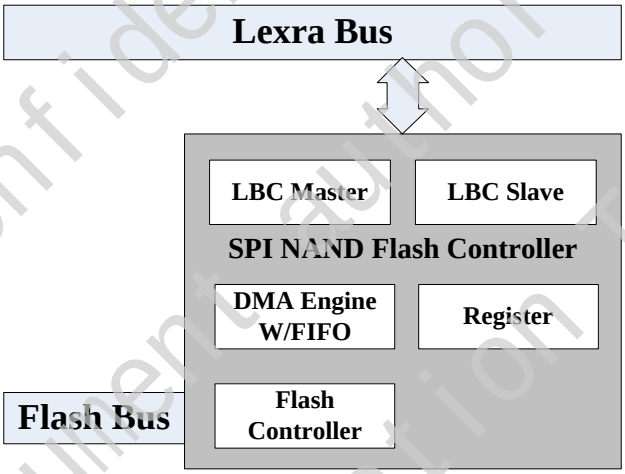


Figure 1 2048B Page Size with 6T ECC

9.1.4. NAND Flash Controller Block Diagram



9.1.5. NAND Flash Controller Command Sets and Support

Command Set

Function	1st Cycle	2nd ~ Cycle	NFC Support
Write Enable	06h	-	Yes
Write Disable	04h	-	Yes
Get Features	0Fh	Address	Yes
Set Feature	1Fh	Address	Yes
Page Read	13h	Address	Yes
Read from Cache	03h	Address	Yes
Read from Cache x2	3Bh	Address	Yes
Read from Cache x4	6Bh	Address	Yes
Read from Cache Dual IO	BBh	Address	Yes
Read from Cache Quad IO	Ebh	Address	Yes
Read ID	9Fh	Address	Yes
Program Load	02h	Address	Yes
Program Load x4	32h	Address	Yes
Program Execute	10h	Address	Yes
Program Load Random Data	84h	Address	No

Program Load Random Data x4	C4h	Address	No
Program Load Random Data Quad IO	72h	Address	No
Block Erase	D8h	Address	Yes
Reset	FFh	-	Yes

9.1.6. Pin Description of NAND Flash Interface

Ping Name	I/O	Pin Function
CS#	I	Chip Select input, active low
SO/SIO1	I/O	Serial Data Output / Serial Data Input Output 1
WP#/SIO2	I/O	Write Protect, active low / Serial Data Input Output 2
SI/SIO0	I/O	Serial Data Input / Serial Data Input Output 0
SCLK	I	Serial Clock input
HOLD#/SIO3	I/O	Hold input, active low / Serial Data Input Output3

9.2. *Register Set (Base Address: 0xB801_A400)*

Base Address: 0xB801_A400			
Offset	**Size (byte)**	**Name**	**Description**
00	4	SNFCFR	SPI NAND Flash Configuration Register
04	4	SNFCCR	SPI NAND Flash CS Control Register
08	4	SNFWCMR	SPI NAND Flash Write Command Register (PIO)
0C	4	SNFRCMR	SPI NAND Flash Read Command Register (PIO)
10	4	SNFRDR	SPI NAND Flash Read Data Register (PIO)
14	4	SNFWDR	SPI NAND Flash Write Data Register (PIO)
18	4	SNFDTR	SPI NAND Flash DMA Trigger Register
1C	4	SNFDRSAR	SPI NAND Flash DMA RAM Start Address Register
20	4	SNFDIR	SPI NAND Flash DMA Interrupt Register
24	4	SNFDLR	SPI NAND Flash DMA Length Register
28	4	SNFDCDSR	SPI NAND Flash DMA Current Data Status Register (ECDCDSR)
…	…	…	…
40	4	SNFSR	SPI NAND Flash Status Register

Note: These registers should be accessed in double word (4bytes).

9.2.1. SPI NAND Flash Configuration Register (SNFCFR) (0xB801_A400)

This configuration register is used both for PIO (programmed I/O) and MMIO (memory mapped I/O).

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:29	*must_be_zero*	*All bits must be 0.*	R	0
28	NAFC_NF	0: Not boot from nand flash 1: Boot from nand flash	R	(Strap)
27	*must_be_zero*	*All bits must be 0.*	R	0
26:24	DEBUG_SELECT	Select debug port message	R/W	000B
23	*must_be_zero*	*All bits must be 0.*	R	0
22	RBO	NAND Flash Read Byte Ordering. 0: The byte order is from low to high {0, 1, 2, 3} 1: The byte order is from high to low {3, 2, 1, 0}	R/W	0B
21	WBO	NAND Flash Write Byte Ordering. 0: The byte order is from low to high {0, 1, 2, 3} 1: The byte order is from high to low {3, 2, 1, 0}	R/W	0B

Bit	Name	Description	Mode	Default
20	DMA_IE	NAND flash controller DMA interrupt enable 0: disable interrupt 1: enable interrupt	R/W	0B
19:15	*must_be_zero*	*All bits must be 0.*	R	0
14	SLV_Endian	0: Slave data is the same order with LX slave bus 1: Slave data is re-ordered to LX slave bus	R/W	0B
13	DMA_Endian	0: DMA data is the same order with LX master bus 1: DMA data is re-ordered to LX master bus	R/W	0B
12	Precise	1: LX bus DMA precise 0: LX bus DMA imprecise	R/W	1B
11:10	*must_be_zero*	*All bits must be 0.*	R	0
9:8	pipe_lat	If chip feedback latency is too large, sw can increase pipe_lat to cover it. 00: data latch pipe latency=0 01: data latch pipe latency=1 10: data latch pipe latency=2 11: data latch pipe_ latency=3	R/W	00B
7	*must_be_zero*	*All bits must be 0.*	R	0
6:4	SPI_CLK_DIV	SPI operating clock rate selection. The value defines the divisor to generate SPI clock SPI Clock = (BUS Clock) / (SPI_CLK_DIV) 000 : DIV = 2 001 : DIV = 4 010 : DIV = 6 011 : DIV = 8 100 : DIV = 10 101 : DIV = 12 110 : DIV = 14 111 : DIV = 16	R/W	111B
3:2	*must_be_zero*	*All bits must be 0.*	R	0
1:0	LBC_BSZ	LBC burst size 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: 128 bytes	R/W	11B

9.2.2. SPI NAND Flash CS Control Register (SNFCCR) (0xB801_A404)

This configuration register is used both for PIO (programmed I/O) and MMIO (memory mapped I/O).

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:5	*must_be_zero*	*All bits must be 0.*	R	0
4	CECS1	Command enable to SPI NAND flash Chip CS#1 0: Command Enable 1: Command Disable	R/W	1B
3:0	*must_be_zero*	*All bits must be 0.*	R	0
0	CECS0	Command enable to SPI NAND flash Chip CS#0 0: Command Enable 1: Command Disable	R/W	1B

9.2.3. SPI NAND Flash Write Command Register (SNFWCMR) (0xB801_A408)

This configuration register is used both for PIO (programmed I/O)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	*must_be_zero*	*All bits must be 0.*	R	0

Bit	Name	Description	Mode	Default
29:28	W_IO_WIDTH	SPI Flash I/O mode selection of a transaction. 00: Serial I/O 01: Dual I/O 10: Quad I/O 11: reserved	R/W	00B
3:2	*must_be_zero*	*All bits must be 0.*	R	0
1:0	W_LEN	SPI write data length (unit = byte) 00: 1 byte 01: 2 byte 10: 3 byte 11: 4 byte	R/W	00B

9.2.4. SPI NAND Flash Read Command Register (SNFRCMR) (0xB801_A40C)

This configuration register is used both for PIO (programmed I/O)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	*must_be_zero*	*All bits must be 0.*	R	0
29:28	R_IO_WIDTH	SPI Flash I/O mode selection of a transaction. 00: Serial I/O 01: Dual I/O 10: Quad I/O 11: reserved	R/W	00B
3:2	*must_be_zero*	*All bits must be 0.*	R	0
1:0	R_LEN	SPI read data length (unit = byte) 00: 1 byte 01: 2 byte 10: 3 byte 11: 4 byte	R/W	00B

[Design Notes]

1. After sending command through this register (SNFRCMR), it will trigger the "PIO read" command.

9.2.5. SPI NAND Flash Read Data Register (SNFRDR) (0xB801_A410)

This configuration register is used for PIO (programmed I/O)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	RDATA3	NAND flash DATA3 port.	RO	00H
23:16	RDATA2	NAND flash DATA2 port.	RO	00H
15:8	RDATA1	NAND flash DATA1 port.	RO	00H
7:0	RDATA0	NAND flash DATA0 port.	RO	00H

9.2.6. SPI NAND Flash Write Data Register (SNFWDR) (0xB801_A414)

This configuration register is used for PIO (programmed I/O)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	WDATA3	NAND flash DATA3 port. Note: WDATA0~3 will always be 0 while read this register	R/W	00H
23:16	WDATA2	NAND flash DATA2 port.	R/W	00H
15:8	WDATA1	NAND flash DATA1 port.	R/W	00H
7:0	WDATA0	NAND flash DATA0 port.	R/W	00H

[Design Notes]

1. After sending command through this register (SNFWDR), it will trigger the "PIO write" command.
2. When RBS or WBS is 1, the DATA3 is the highest address of the register word, and the DATA0 is the lowest address of the register word.
3. When RBS or WBS is 0, the DATA3 is the lowest address of the register word, and the DATA0 is the highest address of the register word.

9.2.7. SPI NAND Flash DMA Trigger Register (SNFDTR) (0xB801_A418)

This configuration register is used both for MMIO (memory mapped I/O) and PIO (programmed I/O).

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:1	*must_be_zero*	*All bits must be 0.*	R	0
0	DMARWE	DMA Read or Write Direction to NAND flash memory 1: Write 0: Read	R/W	0B

[Design Notes]

1. After sending command through this register (SNFDTR), it will trigger the "DMA write/read" command.

9.2.8. SPI NAND Flash DMA RAM Start Address Register (SNFDRSAR) (0xB801_A41C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	ADDR	NAND Flash DMA Start Address in RAM	R/W	0B

9.2.9. SPI NAND Flash DMA Interrupt Register (SNFDIR) (0xB801_A420)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:1	*must_be_zero*	*All bits must be 0.*	R	0
0	DMA_IP	SPI NAND Flash DMA interrupt pending flag. Write '1' to clear the interrupt.	R/WC	0B

[Design Notes]

1. This pending flag will only be useable when enabling "DMA_IE" in the register "SNFCR".

9.2.10. SPI NAND Flash DMA Length Register (SNFDLR) (0xB801_A424)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	*must_be_zero*	*All bits must be 0.*	R	0
29:28	DMA_IO_WIDTH	SPI Flash DMA I/O mode selection of a transaction. 00: Serial I/O 01: Dual I/O 10: Quad I/O 11: reserved	R/W	00B

Bit	Name	Description	Mode	Default
27:20	*must_be_zero*	*All bits must be 0.*	R	0
19:0	LEN	SPI NAND Flash DMA Length	R/W	0B

9.2.11. SPI NAND Flash DMA Current Data Status Register (SNFDCDSR) (0xB801_A428)

(This register does not provide byte access)

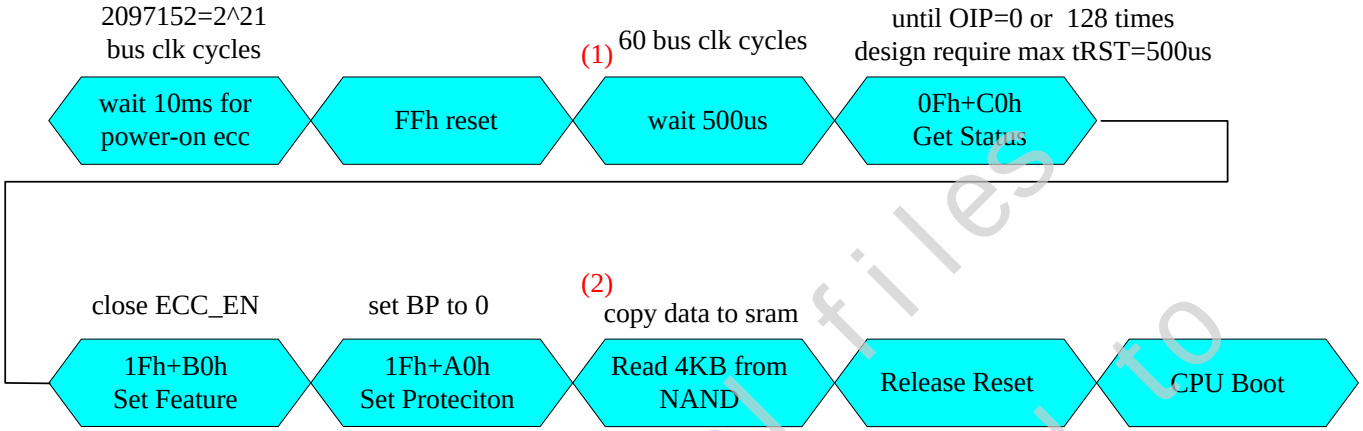
Bit	Name	Description	Mode	Default
31:20	*must_be_zero*	*All bits must be 0.*	R	0
19:0	CLENC	DMA current data move length counter. This is a read only register for debugging, which can view the current finished DMA data length. Note: When S/W read the data length, it will not be the actually length that DMA had moved at that time, just use it to check if DMA is on-going or not.	RO	0B

9.2.12. SPI NAND Flash Status Register (SNFSR) (0xB801_A440)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:5	*must_be_zero*	*All bits must be 0.*	R	0
4	CS#1	SPI NAND Flash CS#1 Status: 0: CS#1 low 1: CS#1 high	RO	1B
3	NFCOS	NAND flash Controller operation status indication bit 0: Ready 1: Busy	RO	0B
2	NFDRS	NAND flash DMA read status 0: Ready 1: Busy (DMA is on-going)	RO	0B
1	NFDWS	NAND flash DMA write status 0: Ready 1: Busy (DMA is on-going)	RO	0B
0	CS#1	SPI NAND Flash CS#0 Status: 0: CS#0 low 1: CS#0 high	RO	1B

[Software Notes]



(1). tRST is a control register which can be read or written. Only SW writes it or power-on can change its value.

(2). Power-on strap pin can choose 4K Bytes or 8K Bytes data copy from nand flash to sram.

10. ECC Controller

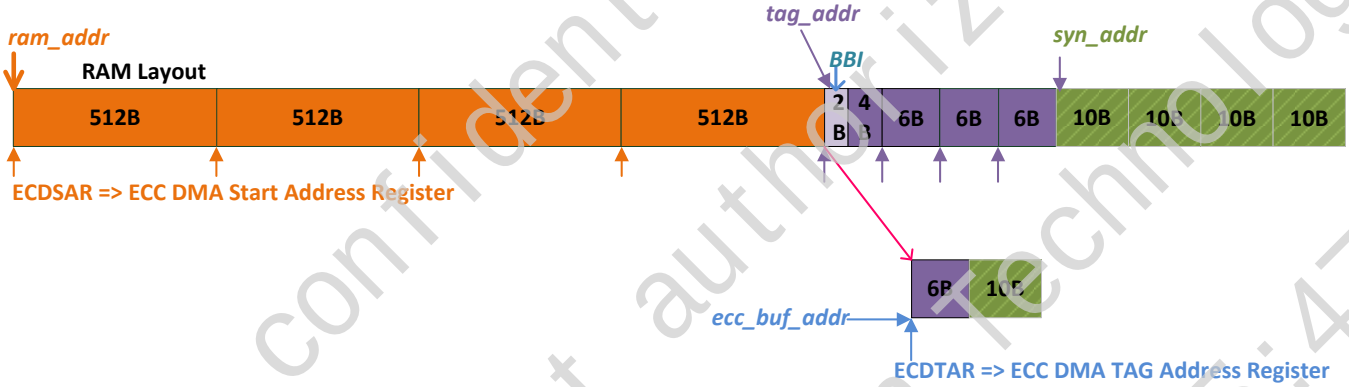
10.1. *Feature lists*

- Support encoding and decoding by ECC Correction Code (BCH)
- Built-in intelligent hardware ECC circuits. Do ECC on the fly based on 518 Bytes per block. (6/12 bit error correction by BCH codes, 6T/12T)
  - 6T: M=13, T=6, N=8191, K=8113 (512B + 6B + 78bits(10B))
  - 12T: M=13, T=12, N=8191, K=8035 (512B + 6B + 156bits(20B))

note: When choosing the ECC algorithm(6T/12T), it must considers on how much the spare space is and how ECC ability it needs.

ECC Encode and Decode Procedure

Example: 6T ECC



- **ECC Encode Steps:**

1. S/W moves the ECC (tag_addr) to the temp buffer for storing ECC data and syndrome (ecc_buf_addr).
2. Sending the ECC encode address (ram_addr) and the temp buffer address (ecc_buf_addr) to ECC-Controller.
3. Kick off ECC-Controller to do ECC Encode
4. Write the generated ECC data & syndrome back to the original RAM array (tag_addr & syn_addr).
5. Repeat Step.1~4 until the whole data been encoded.

- **ECC Decode Steps:**

1. S/W moves the ECC data (tag_addr) to the temp buffer (ecc_buf_addr).
2. S/W moves the ECC syndrome (syn_addr) to the temp buffer (ecc_buf_addr+6B).
3. Setting the ECC decode address (ram_addr) and the temp buffer address (ecc_buf_addr) to ECC Controller.
4. Kick off ECC Controller to do ECC Decode
5. Write the corrected ECC data back to the original array (tag_addr).
6. Repeat Step.1~4 until the whole data been decoded.

10.2. *Register Set (Base Address: 0xB801_A600)*

Base Address: 0xB801_A600			
Offset	**Size (byte)**	**Name**	**Description**

00	4	ECCFR	ECC Controller Configuration Register
04	4	*Reserved*	
08	4	ECDTR	ECC Controller DMA Trigger Register
0C	4	ECDSAR	ECC Controller DMA Start Address Register
10	4	ECDTAR	ECC Controller DMA TAG Address Register
14	4	ECSR	ECC Controller Status Register
18	4	ECIR	ECC Controller Interrupt Register
1C	4	ECCDCDSR	ECC Controller DMA Current Data Status Register
…	…	…	…
40	4	ECNAPR1_1	ECC Controller NAND Flash Parity Register Part1_1
44	4	ECNAPR1_2	ECC Controller NAND Flash Parity Register Part1_2
48	4	ECNAPR1_3	ECC Controller NAND Flash Parity Register Part1_3
4C	4	ECNAPR2_1	ECC Controller NAND Flash Parity Register Part2_1
50	4	ECNAPR2_2	ECC Controller NAND Flash Parity Register Part2_2

10.2.1. ECC Controller Configuration Register (ECCFR) (0xB801_A600)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	*must_be_zero*	*All bits must be 0.*	R	0
29:28	ECC_CFG	BCH algorithm configuration 6T/12T 00: 6bits error correction per 518Bytes 01: 12bits error correction per 518Bytes 1x: reserved	R/W	**00B**
27	*must_be_zero*	*All bits must be 0.*	R	0
26:24	DEBUG_SELECT	Debug port selection	R/W	0000B
23	*must_be_zero*	*All bits must be 0.*	R	0
22	RBO	ECC Controller Read Byte Ordering. 0: The byte order is from low to high {0, 1, 2, 3} 1: The byte order is from high to low {3, 2, 1, 0}	R/W	0B
21	WBO	ECC Controller Write Byte Ordering. 0: The byte order is from low to high {0, 1, 2, 3} 1: The byte order is from high to low {3, 2, 1, 0}	R/W	0B
20	IE	ECC Controller interrupt enable 0: disable interrupt 1: enable interrupt	R/W	0B
19:15	*must_be_zero*	*All bits must be 0.*	R	0
14	SLV_Endian	0: Slave data is the same order with LX slave bus 1: Slave data is re-ordered to LX slave bus	R/W	0B
13	DMA_Endian	0: DMA data is the same order with LX master bus 1: DMA data is re-ordered to LX master bus	R/W	0B
12	Precise	1: LX bus DMA precise 0: LX bus DMA imprecise	R/W	1B
11:2	*must_be_zero*	*All bits must be 0.*	R	0
1:0	LBC_BSZ	LBC burst size 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: 128 bytes	R/W	11B

10.2.2. ECC Controller DMA Trigger Register (ECDTR) (0xB801_A608)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:1	*must_be_zero*	*All bits must be 0.*	R	0
0	DMARE	1: Trigger ECC DMA Write Encoder 0: Trigger ECC DMA Read Decoder	R/W	0B

[Software Notes]

1. In ECC DMA mode, the read/write data unit size is 518Bytes (Data + TAG)
2. Software need to allocate different buffer size (10Bytes/20Bytes) when applying different ECC algorithm (ex: 6T/12T).

10.2.3. ECC Controller DMA Start Address Register (ECDSAR) (0xB801_A60C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	ADDR	ECC DMA Start Address	R/W	0B

[Design Notes]

1. This address is used to indicate source data (512Bytes) for encoding and decoding.

[Software Notes]

1. Software need to meet 4Bytes alignment when allocating ECDSAR.

10.2.4. ECC Controller DMA TAG Address Register (ECDTAR) (0xB801_A610)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	ADDR	ECC DMA TAG Address	R/W	0B

[Design Notes]

1. This address is used to indicate TAG data (tag+parity) for encoding to store it.
2. This address is also used to indicate TAG data (tag+parity) for decoding to load it.

[Software Notes]

1. Software needs to meet 4Bytes alignment when allocating ECDTAR.
2. TAG 16Bytes is need when using 6T BCH
3. TAG 28Bytes is need when using 12T BCH
   I. Useful data length is 26Bytes (TAG+PARITY) and 2Bytes is dummy for alignment

10.2.5. ECC Controller Status Register (ECSR) (0xB801_A614)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:20	*must_be_zero*	*All bits must be 0.*	R	0

Bit	Name	Description	Mode	Default
19:12	ECCN	ECC correction count 0000_0000: No error 0000_0001: 1 error 0000_0010: 2 error 0000_0011: 3 error 0000_0100: 4 error 0000_0101: 5 error 0000_0110: 6 error 0000_0111: 7 error 0000_1000: 8 error 0000_1001: 9 error 0000_1010: 10 error 0000_1011: 11 error 0000_1100: 12 error ……: reserved Note: After ECC engine being triggered @ECDTR, this status will be cleared by ECC controller.	RO	0000B
11:9	*must_be_zero*	*All bits must be 0.*	R	0
8	ECER	ECC result for DMA Decode Operation 1: Error 0: OK Note: After ECC engine being triggered @ECDTR, this status will be cleared by ECC controller.	RO	0B
7:5	*must_be_zero*	*All bits must be 0.*	R	0
4	ALL_ONE	Note the false error after block erasing (all FF)	RO	0B
3:1	*must_be_zero*	*All bits must be 0.*	R	0
0	EOS	ECC Operation status indication bit 0: Ready 1: Busy	RO	0B

10.2.6. ECC Controller Interrupt Register (ECIR) (0xB801_A618)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:1	*must_be_zero*	*All bits must be 0.*	R	0
0	DMA_IP	ECC controller DMA interrupt pending flag. Write '1' to clear the interrupt.	R/WC	0B

10.2.7. ECC Controller DMA Current Data Status Register (ECDCDSR) (0xB801_A61C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:20	*must_be_zero*	*All bits must be 0.*	R	0
19:0	CLENC	DMA current data move length counters This is a read only register for debugging, which can view the current finished DMA data length. Note: When S/W read the data length, it will not be the actually length that DMA had moved at that time, just use it to check if DMA is on-going or not.	RO	0B

10.2.8. ECC Controller NAND Flash Parity Register Part1_1 (ECNAPR1_1) (0xB801_A640)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	NAPR1_1	NAPR1_1: 4 bytes parity check part	RO	0000B

10.2.9. ECC Controller NAND Flash Parity Register Part1_2 (ECNAPR1_2) (0xB801_ A244)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	NAPR1_2	NAPR1_2: 4 bytes parity check part	RO	0000B

10.2.10. ECC Controller NAND Flash Parity Register Part1_3 (ECNAPR1_3) (0xB801_ A648)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	NAPR1_3	NAPR1_3: 4 bytes parity check part	RO	0000B

10.2.11. ECC Controller NAND Flash Parity Register Part2_1 (ECNAPR2_1) (0xB801_ A64C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:16	NAPR1_4	NAPR1_4: final 2 bytes parity check part	RO	0000B

10.2.12. ECC Controller NAND Flash Parity Register Part2_2 (ECNAPR2_2) (0xB801_ A650)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	NAPR2_1	NAPR2_1: 4 bytes parity check part	RO	0000B

[Design Notes]

1. 6T BCH algorithm use 0xA03C~0xA044 for TAG_SEL0, 0xA048~0xA050 for TAG_SEL1, 0xA054~0xA05C for TAG_SEL2, 0xA060~0xA068 for TAG_SEL3.
2. 12T BCH algorithm use 0xA03C~0xA050 for TAG_SEL0, 0xA054~0xA068 for TAG_SEL1. (Because 12T algorithm has 20Bytes parity check information)

11. DDR PHY IP

11.1. *Features*

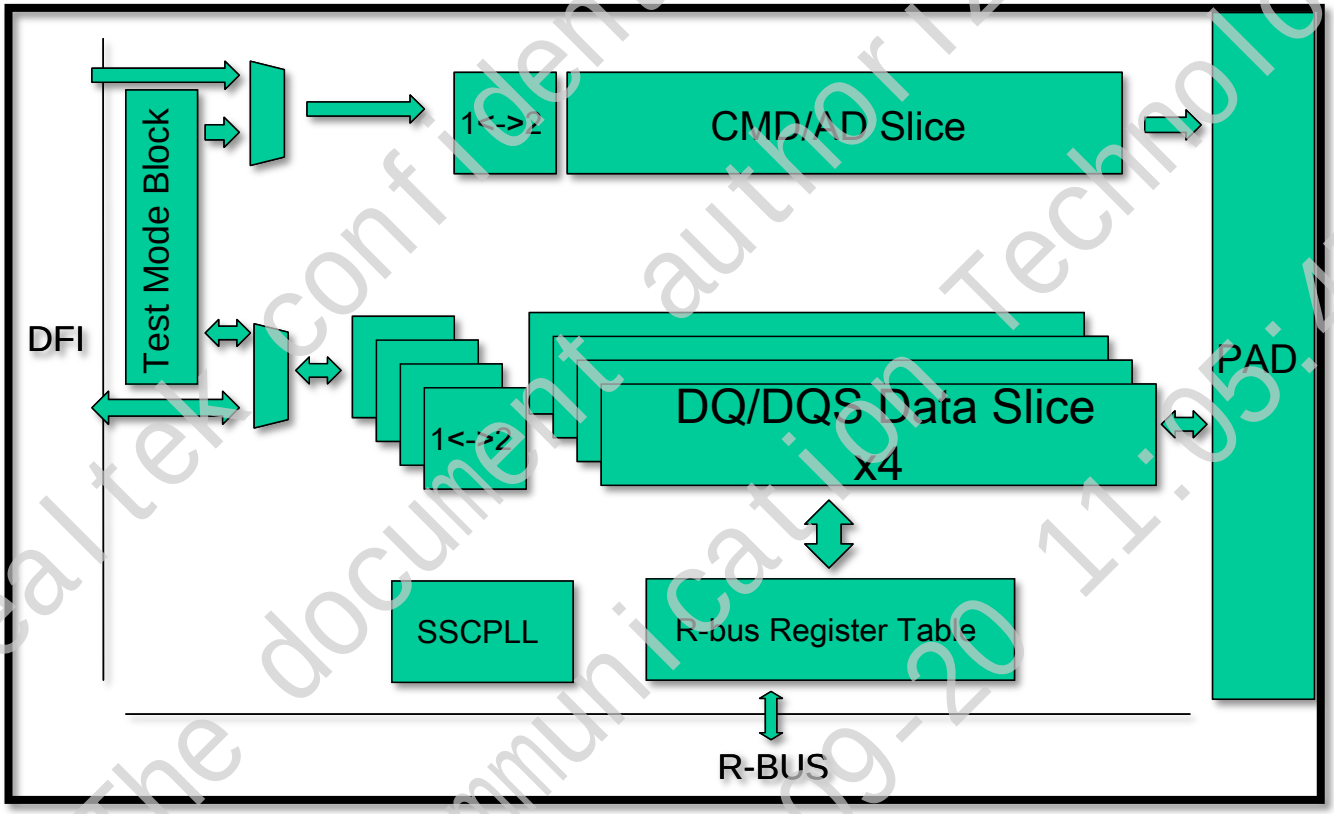
11.1.1. Overview

DPI (DDR PHY IP) supports x32 DDR SDRAM devices and provides DFI (DDR PHY Interface) protocol for the connectivity between a DDR memory controller (MC) and DPI. The MC and DPI can operate in either matched frequency or at a frequency ratio of 1:2.

Training operations write and read leveling, which allow for more accurate alignment of critical timing signals. The ZQ calibration is also provided by DPI for PAD calibration.

11.1.2. Architecture

The following is block diagram of DPI.



11.1.3. Clock

SSCPLL is used to generate 6 DPI clocks and each clock has 64 phases. The clock structure is shown in Figure2. The MCK_CMD is main clock domain in DPI. For write-leveling phase adjustment, the MCK_CMD domain is converted to MCK_DQS domain in data slice.

In order to improve clock duty to DDR, SSCPLL output will enter DCC and DCD and the one to DRAM CK are placed near to PAD. The clock tree in DPI is shown in Figure3.

When connect to x16 or x8 DDR device, the MCK_DQS, which are to no operating data slice, should be turned off for power saving.

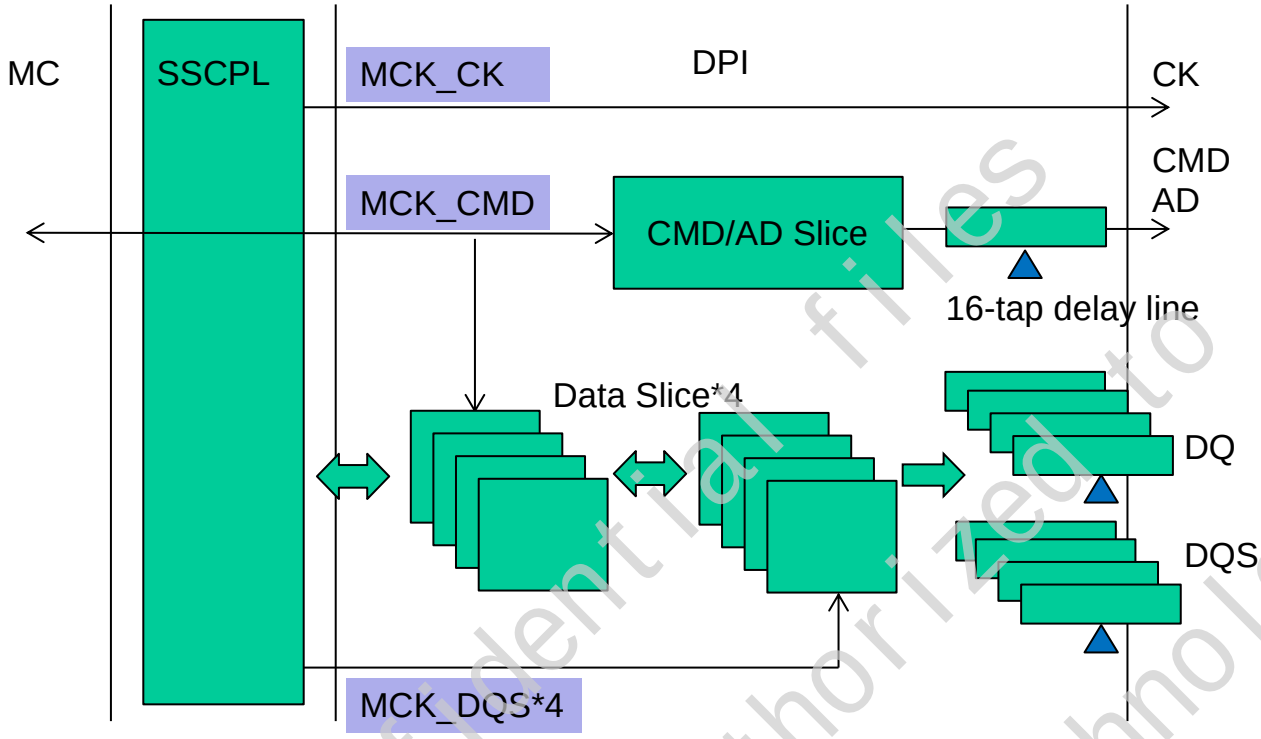


Figure: Clock Structure

11.2. *Registers Set (Base Address: 0xB814_2000)*

Reference RLE0709_dpi_crt_spec.doc and RLE0709_dpi_dll_spec.doc

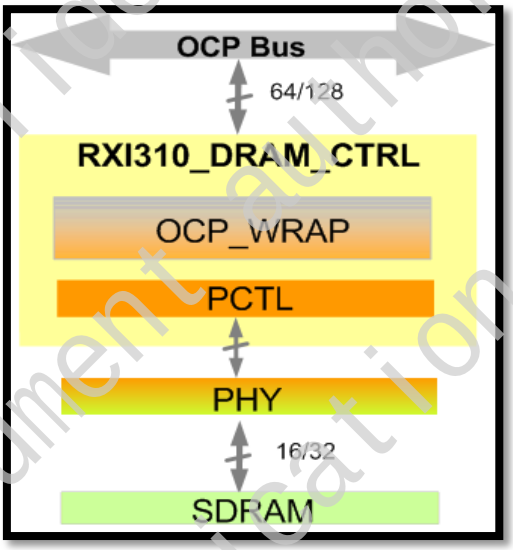
12. DRAM Controller RXI-310

12.1. *Memory Controller Features*

RXI-310 DRAM Controller (DRAM_CTRL) is memory controller used to access SDR, DDR1, DDR2, DDR3 and DDR4. This IP is asynchronous clock between OCP bus wrapper (OCP_WRAP) and DRAM protocol controller (PCTL). In OCP wrapper it is used to transfer OCP commands and data to PCTL for communicating with DRAM. In PCTL it controls DRAM access protocol to avoid DRAM command and data transition violates the DRAM timing constraints.

12.1.1. System Overview

DRAM_CTRL is a slave device on OCP Bus system. It supports high-performance interface for DRAM data and peripheral interface for setting internal control registers of OCP_WRAP and PCTL. In the system it also allows system designers to plan single or 2 chips (DQ32) of x16 DRAM architecture. If system is designed to DQ32, it would have double data bandwidth. And it can increase DRAM performance obviously, but the cost is higher than single x16 DRAM and it would have other some effect in data access.



DRAM Controller Architecture

12.1.2. DRAM Address Decode

In DRAM addressing it decodes data address to {row, bank, column} address formats. According to different DRAM size chips user should program the related information in PAGE_SIZE and BANK_SIZE fields of CR_MISC control register correctly. The page size depends on column address and DQ bits. The bank size is defined in the used DRAM specification. If setting a not corresponding DRAM SIZE, the device may cause the error address implementation to access DARM. In this situation it usually maybe multiple addresses mapping to the same location.
In this module the bank size supports to 4, 8, and 16. And the page size supports to 256B, 512B, 1K, 2K, 4K, 8K, 16K, 32K, and 64K.
The page size equation is from the following: Unit is Byte

Page size = $(2^{\text{column bits}} \times \text{DRAM DQ-bit}) / 8$

For example:
If DRAM column address is 10 bit and using 2 chip of 16-bit DQ width.
Then the page size should be setting to 4K ($(2^{10} \times 2 \times 16)/8$). Therefore the OCP_WRAP would be correct to access DRAM.

12.1.3. DRAM Init Flow

DRAM must be powered up and initialized in a predefined manner. But SDR, DDR2 and DDR3 initialization flow are different. For this reason RXI310_DRAM supports the related initial flow to DRAM according to the control registers setting. Programmer should set correct contents to the related registers before INIT field of CR_CCR is set. After setting to INIT field of CR_CCR register, RXI310_DRAM would start to issue DARM initialization function (INIT) automatically. During INIT implementation user needs to wait until INIT is done. Then user can switch the operation function in CR_CCR to access DRAM.

The following steps descript INIT for DDR3 power-up and initialization sequence:

1. Set CR_CCR control register (Address offset: 0x0), start to implement INIT.
2. PCTL ties RST_N to low and tie CKE to low, wait over 200 us.
3. PCTL ties RST_N to high and keep CKE to low, wait over 500 us.
4. PCTL ties CKE to high.
5. PCTL issues NOP command.
6. PCTL issues Mode register command: EMR2, according CR_MR2 register
7. PCTL issues Mode register command: EMR3, according CR_MR3 register
8. PCTL issues Mode register command: EMR1, according CR_MR1 register
9. PCTL issues Mode register command: MR, according CR_MR0 register
10. PCTL issues ZQCL command and wait over tZQinit timing.
11. PCTL would set INIT_DONE then all automatic initial function is end. User can check INIT bit field of CR_CCR by reading.

The following steps descript INIT for DDR4 power-up and initialization sequence:

1. Set CR_CCR control register (Address offset: 0x0), start to implement INIT function.
2. PCTL ties RESET_n signal to low, wait over 100 ns.
3. PCTL ties CKE signal to low and RESET_n signal to high, wait over 500 us.
4. PCTL issues Mode register command: MR3, according CR_MR3 register
5. PCTL issues Mode register command: MR6, according CR_MR6 register
6. PCTL issues Mode register command: MR5, according CR_MR5 register
7. PCTL issues Mode register command: MR4, according CR_MR4 register
8. PCTL issues Mode register command: MR3, according CR_MR3 register
9. PCTL issues Mode register command: MR2, according CR_MR2 register
10. PCTL issues Mode register command: MR1, to enable CR_DLL
11. PCTL issues Mode register command: MR0, according CR_MR0 and reset DLL(A8=0)
12. PCTL issues ZQCL command
13. PCTL would set INIT_DONE. All automatic initial function is end. User can check INIT bit field of CR_CCR by reading

The following steps descript INIT for PHY power-up and initialization sequence:

1. initial PWDPAD15N=0 (lock PAD)
2. setting ddr parameter
3. wait ddr phy pll stable
4. set PWDPAD15N=1 (unlock PAD) and wait at least 20 DCK clock before DRAM command start
5. do DRAM initialization

12.1.4. DPIT Function

For flexibility, RXI310_DRAM keeps a method to operate DRAM command directly. User can program CMD_DPIN and TIE_DPIN register to implement DPIT function. By the way user only can control DRAM command pins and user must issue this function carefully. After DPIN is enabled, RXI310_DRAM would be setting to the DRAM command pins. When DPIT is completed, DPIN_DONE would be set in CCR.
The following steps descript to enable DPIT:

1. Enter ALL_IDLE mode, set 3'h7 to CR_CSR[10:8].
2. User can program TIE_DPIN according the target DRAM specification. RXI310_DRAM would always set the related value to {odt, cke, rst} pins according to TIE_DPIN register.
3. User also can program CMD_DPIN to the related DRAM pins.
4. After to enable DPIN field of CR_CCR, RXI310_DRAM starts DPIN implementation. And it would operate the related DRAM command pins according to CMD_DPIN only at one DRAM clock cycle.
5. RXI310_DRAM would set DPIN_DONE field of CR_CCR. User should check this status to wait until this function is finishing. And user can program new operation to DRAM.

12.1.5. Software Initial Flow

First user must implement initial code setting RXI310_DRAM before to access DRAM. Otherwise it would make unexpected operation, even the controller may be in the deadlock state. Programmer must set the related control registers and the correct value according to DRAM clock frequency and the target DRAM spec. If the contents of these register are error, RXI310_DRAM would execute fail operation to DRAM. And DRAM would not work correctly. Therefore the software engineers must be careful to handle the setting.
The initial settings of RXI310_DRAM are as following:

1. Set BSTC, DT, and MEM fields of CR_CSR control register (Address offset: 0xC), check these bits until DRAM_CTRL is in idle state. Ensure DRAM_CTRL has been disabled all operations to DRAM when user implements initial settings. Otherwise DRAM_CTRL may cause unsafe and unexpected situation.
2. Set CR_MISC control register (Address offset: 0x240) according to the target DRAM information: page size, bank size, and dram burst length.
3. Set CR_DCR control register (Address offset: 0x4) according to the target DRAM information. And if the DRAM architecture DQ32, setting DQ32 field.
4. Other control registers (Address offset: 0x8, 0x10 ~ 0x20, 0x34~0x4c) are used to set the target DRAM operation constrains to avoid timing violation.
5. Set PCTL: CR_CCR control register (Address offset: 0x0), start to implement automatic DRAM initial function
6. Read PCTL: CR_CCR control register (Address offset: 0x0), check if DRAM initial function until

initial flow is complete.

7. Set MEM fields of CR_CSR control register (Address offset: 0xC), start to access DRAM.

12.2. *Register Set (Base Address: 0xB814_3000)*

Refer to RXI-310_DRAM_CTRL_Technical_Reference_Manual for more detailed information.

Base Address: 0xB814_3000			
Offset	**Size (byte)**	**Name**	**Description**
DRAM Protocol Control Registers			
0x00	4	CCR	Configuration Control Register
0x04	4	DCR	DRAM Configuration Control Register
0x08	4	IOCR	I/0 Configuration Control Register
0x0C	4	CSR	Controller Status Register
0x10	4	DRR	DRAM Refresh Control Register
0x14	4	TPR0	DRAM Timing Parameter Register 0
0x18	4	TPR1	DRAM Timing Parameter Register 1
0x1C	4	TPR2	DRAM Timing Parameter Register 2
0x20	4	TPR3	DRAM Timing Parameter Register 3
0x28	4	CMD_DPIN	Command Data Pin Register
0x2C	4	TIE_DPIN	Tied Data Pin Register
0x30	4	MR_INFO	Mode Latency Information Register
0x34	4	MR0	DRAM Mode Register 0
0x038	4	MR1	DRAM Mode Register 1
0x03C	4	MR2	DRAM Mode Register 2
0x040	4	MR3	DRAM Mode Register 3
0x044	4	MR4	DRAM Mode Register 4
0x048	4	MR5	DRAM Mode Register 5
0x04C	4	MR6	DRAM Mode Register 6
0x0D0	4	BCR	BSTC Control Register
0x0D4	4	BCT	BSTC compare loop counter register
0x0D8	4	BCM	BSTC compare bit mask Register
0x0DC	4	BST	BSTC Status Register
0x0E0	4	BSRAM0	BSTC SRAM Status Register 0
0x0E4	4	BSRAM1	BSTC SRAM Status Register 1
0x0E8	4	BER	BSTC Error Bit Register
…	…	…	…
0x0F4	4	PCTL_SVN_ID	PCTL Version ID Register
0x0F8	4	PCTL_IDR	PCTL Identification Register
…	…	…	…
SDR PHY Control Registers			
Offset	**Size (byte)**	**Name**	**Description**
0x100	4	DLY0	SDR PHY Delay Register 0
0x104	4	DLY1/DCM_RST	SDR PHY Delay Register 1 for ASIC/DCM Reset Register for FPGA
0x108	4	DLY_CLK_PHA	SDR PHY Delay Clock Phase Register
0x10C	4	DLY_ST	SDR PHY Delay Status Register
…	…	…	…
OCP_WRAP Control Registers			
Offset	**Size (byte)**	**Name**	**Description**
0x224	4	MISC	Memory Information Setting Control Register

Base Address: 0xB814_3000			
Offset	**Size (byte)**	**Name**	**Description**
DRAM Protocol Control Registers			
…	…	…	…
0x2A0	4	OCP_WRAP_IDR	OCP_WRAP ID number Register
0x2A4	4	OCP_WRAP_VERSION	OCP_WRAP version numbers

12.2.1. Configuration Control Register (CCR) (0xB814_3000)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	CR_UPDATE	Indicate to update the related control register to timing counter register: TPR0, TPR1, TPR2, TPR3 without any implementation (DPIT, BTT, INIT).	W	0x0
30:9	*must_be_zero*	*All bits must be 0.*	R	0x0
8	FLUSH_FIFO	Flush all FIFO in RXI310_DRAM_CTRL Write: 0 - Not to flush FIFO 1 – Start to Flush FIFO Read: 0 - Flushing FIFO operation is done 1 - It is still active to flush FIFO.	R/W	0x0
7:4	NO_USE	Internal Information	R	0x1
3	DPIT	Start to set DRAM pins function. According to CMD_DPIN register setting to the related DRAM command pins one DRAM clock cycle. Write: 0 - Not to set value to DRAM command pins. 1 – Set value to DRAM command pins one Cycles. Read: 1 – Set value to DRAM command pins is done (DPIT_DONE)	R/W	0x0
2	BTT	Start to run BIST function. Write: 0 – Disable BIST function 1 – Run BIST Function * Write mode is enable if BIST module is implemented Read: 0 –BIST function is still active 1 – BIST function is done (BIST_DONE).	R/W	0x1
1	*must_be_zero*	*All bits must be 0.*	R	0x0
0	INIT	Start to issue DRAM initialization sequence Write: 0 – Disable DRAM initialization function 1 – Enable DRAM initialization function Read: 0 –DRAM initialization function is still active 1 –DRAM initialization function is done (INIT_DONE).	R/W	0x0

12.2.2. DRAM Configuration Control Register (DCR) (0xB814_3004)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	*must_be_zero*	*All bits must be 0.*	R	0x0
30	ZQC	Issue DRAM ZQCS/ZQCL command after serial Auto-refresh command implementation. This function should set with CR_DRR:ZQCL_INV and CR_TPR2:ZQCS_INV	R/W	0x0
29:22	*must_be_zero*	*All bits must be 0.*	R	0x0
21	BG2	Support BankGroup2 function for DDR4	R/W	0x0
20	*must_be_zero*	*All bits must be 0.*	R	0x0

Bit	Name	Description	Mode	Default
19	GEAR_DOWN	Setting GEAR_DOWN function for DDR4 DRAM command and DQ pins would be operated valid at 2 cycle	R/W	0x0
18	PAR	Setting PAR function for DDR4 DRAM priority pin would execute with the related pins	R/W	0x0
17	DBI	Setting READ DBI function for DDR4 If setting DBI function, PCTL would inverted DRAM read data if DM_n/DBI_n is active low	R/W	0x0
16:13	*must_be_zero*	*All bits must be 0.*	R	0x0
12	RANK2	Set DRAM RANK2 architecture implement. In RANK2 architecture, it should have 2 CS pins and Command/DQ bus is shared to connect with 2 DRAM dies. *Only support RANK2 function if configuration: RANK_BITS > 2	R/W	0x0
11	*must_be_zero*	*All bits must be 0.*	R	0x0
10:8	DFI_RATE	Setting DFI ratio. 3'b001→1:1, 3'b010→2:1, others are reserved. *If using SDR, it doesn't support DFI.	R/W	0x0
7	*must_be_zero*	*All bits must be 0.*	R	0x0
6	HYBR_DQ	Indicate access DRAM DQ_32 hardware architecture in DQx16-bit or DQx32-bit according to CR_MISC:HYBR_SIZE . It needs hardware configuration set HYBR_DRAM. If setting HYBR_DQ, CR_DCR:{HALF_DQ, DQ32} should set 0.	R/W	0x0
5	HALF_DQ	Indicate only using 16-bit DRAM on the DQ_32 hardware architecture. If setting HALF_DQ, CR_DCR:{HYBR_DQ, DQ32} should set 0.	R/W	0x0
4	DQ_32	Setting the DDR architecture is 2 x 16-Bit DRAM. It is valid to set if the hardware architecture is valid configuration. If setting DQ_32, CR_DCR:{HYBR_DQ, HALF_DQ32} should set 0.	R/W	0x0
3:0	SDR_DDR	Indicate the target DRAM type 4'b1000 →SDR 4'b0001→DDR1,4'b0010→DDR2, 4'b0011→DDR3 4'b0100→DDR4 Others are reserved	R/W	0x0

12.2.3. I/O Configuration Control Register (IOCR) (0xB814_3008)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	FPGA_DFI	Enable to connect external PHY on FPGA	R/W	0x0
30	FPGA_LP	Enable to loopback mode form output DFI ports to input DFI ports on FPGA.	R/W	0x0
29:25	*must_be_zero*	*All bits must be 0.*	R	0x0
24:20	TPHY_RDATA_EN	Setting the delay latency from DFI read command to dfi_rddata_en signal. If it works on DDR, the smallest value is 1.	R/W	0x1
19:17	TPHY_WLAT	Setting the delay latency from DFI write command to dfi_wrdata_en signal. If it works on DDR, the smallest value is 0.	R/W	0x0
16:12	TPHY_WDATA	Setting the delay latency from dfi_wrdata_en signal to dfi_wddata. If it works on DDR, the smallest value is 1.	R/W	0x1
11:8	RD_PIPE	Setting the extra delay latency of CAS on SDR. If it works on SDR, the smallest value is 0.	R/W	0x0
7	*must_be_zero*	*All bits must be 0.*	R	0x0

Bit	Name	Description	Mode	Default
6	HALF_CS_N	Setting DRAM:CS	R/W	0x0
5	tRTW_2T_DIS	Setting DRAM: read -> write command interval time > 2t*	R/W	0x0
4	2N_PRE_EN	Setting DRAM: command pins = keeps 2t but DRAM:CS keeps only 1t*. If DFI ratio is1, it only can be allowed setting to 0. If DFI ratio is 2, it would suggest setting to 1. The timing margin is better at DRAM.	R/W	0x0
3	STC_ODT	Setting DRAM static ODT function. Else DRAM_CTRL supports dynamic ODT function.	R/W	0x0
2	STC_CKE	Setting DRAM static CKE function. Else DRAM_CTRL supports dynamic CKE function.	R/W	0x0
1	*must_be_zero*	*All bits must be 0.*	R	0x0
0	ODT_2PINS	Indicate to connect 2 ODT_PINS. If `ODT_BITS =1, this bit would be tied 0.	R/W	0x0

12.2.4. Controller Status Register (CSR) (0xB814_300C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:11	*must_be_zero*	*All bits must be 0.*	R	0x0
10	BSTC_IDLE	Disable BSTC function state Write: To set BSTC idle state 0– Enable BSTC function 1– Set BSTC idle state; disable BSTC function. Read: To check if BSTC state is in idle 0 – BSTC function implementation is allowed. 1 – BSTC state is idle and user can't implement BSTC function.	R/W	0x0
9	*reserved*	*reserved*	R	0x1
8	MEM_IDLE	Disable memory access state Write: To set memory access idle state 0 –Enable memory access state. 1 –Set memory access idle state; disable memory access. Read: To check if memory access state is in idle 0 –Access DRAM is allowed. 1 – Memory state is idle and user can't access DRAM through data bus.	R/W	0x0
7:0	*must_be_zero*	*All bits must be 0.*	R	0x0

12.2.5. DRAM Refresh Control Register (DRR) (0xB814_3010)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:29	ZQCL_INV	It is used for indicate ZQCL implementation interval time after ZQCS implementation times. For example: set ZQC_INV to 2 and ZQCL to1 -----||||| -------||||| +ZQCS -----|||||--------||||| +ZQCL---- Serial-REF Serial-REF+ZQCS Serial-REF Serial-REF+ZQCS	R/W	0x0
28	REF_DIS	Disable RXI310_DRAM issues Refresh command to DRAM.	R/W	0x0
27:24	REF_NUM	Setting the number of Refresh commands to issue in between two surrounding Refresh commands	R/W	0x8
23:8	tREF	Maximum average Refresh commands delay cycles. tREF[4:0] is also used to indicate the early cycles to inform RTK_PHY before ZQC command implementation.	R/W	0xF00
7:0	tRFC	Refresh to Active or Refresh commands delay cycles	R/W	0x72

12.2.6. DRAM Timing Parameter Register 0 (TPR0) (0xB814_3014)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	tZQCS	ZQCS command cycles	R/W	0x0
23:21	*must_be_zero*	*All bits must be 0.*	R	0x0
20:17	tCKE	max(tCKE, tXP) cycles	R/W	0x0
16:13	tRTP	Read to Precharge command delay cycles. The tRTP should be follow as Equation/ DFI_RATIO: DRAM Type / Equation SDR / BL/2 DDR1 / BL/2 DDR2 / AL + BL/2 - 2 + tRTP DDR3/4 / AL + tRTP * BL: DRAM burst length	R/W	0x3
12:9	tWR	Write Recovery delay cycles	R/W	0x4
8:4	tRAS	Active to Precharge command delay cycles	R/W	0x5
3:0	tRP	Precharge command cycles	R/W	0x6

12.2.7. DRAM Timing Parameter Register 1 (TPR1) (0xB814_3018)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:29	*must_be_zero*	All bits must be 0.	R	0x0
28:24	tFAW	Four active window cycles	R/W	0x0
23:20	tRTW	Read to Write command delay cycles. The equation of tRTW: CR_tRTW = (tRTW/ DFI_RATIO) -1 * minima value is 0.	R/W	0x0
19:17	tWTR	Delay cycles start of internal Write transaction to internal read command The equation of tWTR: CR_tWTR = (tWTR/ DFI_RATIO) -1 * minima value is 0.	R/W	0x3
16:14	tCCD	CAS# to CAS# command delay cycles	R/W	0x1
13:10	tRCD	Active to internal Read or Write delay cycles	R/W	0xF
9:4	tRC	Active to Active or Refresh command delay cycles(in the same BANK).	R/W	0x8
3:0	tRRD	Active to Active command delay cycles (in the different BANK)	R/W	0x2

12.2.8. DRAM Timing Parameter Register 2 (TPR2) (0xB814_301C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:22	RST_tUS	If non-dfined CR_INTI_RMBIT configuration. For DRAM initiation flow, RESET_n signal would be set high until $2^{(RST\_tUS<<10)}$ cycles *RST_tUS can't be 0.	R/W	0x80
21:12	tUS	If non-dfined CR_INTI_RMBIT configuration. For different DRAM initiation flow, it has to wait tUS ns. It can set this register. The wait time is $2^{(tUS<<10)}$ cycles. *tUS can't be 0.	R/W	0x80
11:8	tNS	If non-dfined CR_INTI_RMBIT configuration. For different DRAM initiation flow, it has to wait tNS ns. It can set this register. The wait time is $2^{(tUS<<7)}$ cycles. *tNS can't be 0.	R/W	0x2

Bit	Name	Description	Mode	Default
7:4	tMRD	Mode register operations delay cycles. Setting the maximum value between Mode register set command delay cycles and Mode Register set command update delay cycles.	R/W	0x2
3:0	INIT_REF_NUM/ ZQC_INV	Set the number of Refresh command to issue in DRAM(SDR/DDR1/DDR2) initialization flow. After DRAM initial flow, it is also used for indicate ZQCS implementation interval times after serial AUTO_Refresh implementation times. For example: set ZQC_INV to 2 and ZQCL to1 -----|||| -------|||| +ZQCS -------||||--------|||| +ZQCL---- Serial-REF Serial-REF+ZQCS Serial-REF Serial-REF+ZQCS	R/W	0x2

12.2.9. DRAM Timing Parameter Register 3 (TPR3) (0xB814_3020)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:29	tCCD_R	tCCD delay cycles for different rank DRAM die. It should be > CR_TPR1: tCCD.	R/W	0x0
28:20	*must_be_zero*	*All bits must be 0.*	R	0x0
19:17	tWTR_S	Delay cycles start of internal Write transaction to internal read command at different bank group in DDR4. *tWTR is used to the same bank group in DDR4	R/W	0x0
16:14	tCCD_S	CAS# to CAS# command delay cycles at different bank group in DDR4. *tCCD is used to the same bank group in DDR4	R/W	0x0
13:7	*must_be_zero*	*All bits must be 0.*	R	0x0
6:4	DPIN_CMD_LAT	Set CS_N pin delay cycles with others command pins. *It is usually CS_N to avoid setup/hold in setting DDR4 gear down function.	R/W	0x0
3:0	*must_be_zero*	*All bits must be 0.*	R	0x0

12.2.10. Command Data Pin Register (CMD_DPIN) (0xB814_3028)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	CS_N_1	Set the value of CS_1 pin if parameter: CS_BITS>1	R/W	0x1
30:27	*must_be_zero*	All bits must be 0.	R	0x0
26	PARITY	Set the value of PARITY pin in DDR4	R/W	0x0
25	ACT_N	Set the value of ACT_N pin in DDR4	R/W	0x1
24	CS_N	Set the value of CS_N pin	R/W	0x1
23	RAS_N	Set the value of RAS_N pin	R/W	0x1
22	CAS_N	Set the value of CAS_N pin	R/W	0x1
21	WE_N	Set the value of WE_N pin	R/W	0x1
20:17	BA	Set the value of BA pins	R/W	0x0
16:0	ADDR	Set the value of ADDR pins	R/W	0x0

12.2.11. Tied Data Pin Register (TIE_DPIN) (0xB814_302C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	ODT_1	Set the value of ODT_1 pin if parameter: ODT_BITS>1	R/W	0x0
30	CKE_1	Set the value of CKE_1 pin if parameter: CKE_BITS>1	R/W	0x1
29	RST_N_1	Set the value of RST_N_1 pin if parameter: RST_BITS>1	R/W	0x1
28:5	*must_be_zero*	All bits must be 0.	R	0x0
4	ALERT_N	Set the value of ODT pin	R/W	0x0

Bit	Name	Description	Mode	Default
3	TEN	Set the value of CKE pin	R/W	0x0
2	ODT	Set the value of ODT pin	R/W	0x0
1	CKE	Set the value of CKE pin	R/W	0x1
0	RST_N	Set the value of RST_N pin	R/W	0x1

12.2.12. Mode Latency Information Register (MR_INFO) (0xB814_3030)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:18	*must_be_zero*	All bits must be 0.	R	0x0
14:10	ADD_LAT	It is indicate DRAM additive latency.	R/W	0x0
9:5	RD_LAT	It is indicate DRAM read data latency. The equation is: (CAS latency + Additive latency +*Parity latency(DDR4)) /DFI_RATIO	R/W	0x0
4:0	WR_LAT	It is indicate DRAM write data latency. The equation is: (CAS write latency+ Additive latency + *Parity latency(DDR4)) /DFI_RATIO	R/W	0x0

12.2.13. DRAM Mode Register 0 (MR0) (0xB814_3034)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	MR0	Setting the value to Mode register MR0 in DRAM initialization flow.	R/W	0x0aa2

12.2.14. DRAM Mode Register 1 (MR1) (0xB814_3038)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	MR1	Setting the value to Mode register EMR1 in DRAM initialization flow.	R/W	0x0

12.2.15. DRAM Mode Register 2 (MR2) (0xB814_303C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	MR2	Setting the value to Mode register EMR2 in DRAM initialization flow.	R/W	0x0

12.2.16. DRAM Mode Register 3 (MR3) (0xB814_3040)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	MR3	Setting the value to Mode register EMR3 in DRAM initialization flow.	R/W	0x0

12.2.17. DRAM Mode Register 4 (MR4) (0xB814_3044)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	MR4	Setting the value to Mode register MR4 in DRAM initialization flow.	R/W	0x0

12.2.18. DRAM Mode Register 5 (MR5) (0xB814_3048)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	MR5	Setting the value to Mode register MR5 in DRAM initialization flow.	R/W	0x0

12.2.19. DRAM Mode Register 6 (MR6) (0xB814_304C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31	MR6	Setting the value to Mode register MR6 in DRAM initialization flow.	R/W	0x0

12.2.20. BSTC Control Register (BCR) (0xB814_30D0)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	*must_be_zero*	All bits must be 0.	R	0x0
29:16	RD_EX_CNT	Set the expected word number of BSTC_RD_SRAM. The equation is (total return bits) / 32. For example: 4 Read commands with DQ16, DRAM_BST8 RD_EX_CNT = (4x16x8)/32 = 16 4 Read commands with DQ32 & DRAM_BST8 RD_EX_CNT = (4x32x8)/32 =32	R/W	0x0
15	*must_be_zero*	All bits must be 0.	R	0x0
14:12	RELOAD_SRAM	RELOAD_SRAM[2]: Reload to keep the previous BSTC RG SRAM level RELOAD_SRAM[1]: Reload to keep the previous BSTC WD SRAM level RELOAD_SRAM[0]: Reload to keep the previous BSTC CMD SRAM level	R/W	0x0
11	FUS_RD	Flush BSTC_RD_SRAM data.	R/W	0x0
10	FUS_RG	Flush BSTC_RG_SRAM data	R/W	0x0
9	FUS_WD	Flush BSTC_WD_SRAM data	R/W	0x0
8	FUS_CMD	Flush BSTC_CMD_SRAM data	R/W	0x0
7:6	*must_be_zero*	All bits must be 0.	R	0x0
5	CRR	Set to clean CR_BER and CR_BST:{ERR_CNT, ERR_FST_TH} registers	W	0x0
4	AT_ERR_STOP	RXI310_DRAM would stop the BSTC function implementation if BSTC_RD and BSTC_RG compare error data automatically. To enable this function it also must enable the loop mode and compare mode (BCR[2:1]).	R/W	0x0
3	DIS_MASK	Disable using BSTC_MSK_SRAM. RXI310_DRAM would write full data to DRAM without masking function.	R/W	0x0
2	LOOP	Enable loop mode. RXI310_DRAM repeats to implement BSTC function for overnight testing	R/W	0x0
1	CMP	Enable compare mode. RXI310_DRAM would compare read data form DRAM with the expected data in BSTC_RG_SRAM.	R/W	0x0
0	STOP	Disable to run BSTC hardware function	R/W	0x0

12.2.21. BSTC compare loop counter register (BCT) (0xB814_30D4)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:16	*must_be_zero*	All bits must be 0.	R	0x0
15:0	LOOP_CNT	If set LOOP_CNT >0 and enable CR_BCR:LOOP, BSTC test patterns would be tested in LOOP_CNT times.	R/W	0x0

12.2.22. BSTC compare bit mask Register (BCM) (0xB814_30D8)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	CMP_MBIT	It is used to mask data bit to compare in auto compare function. *If DQ32 configuration is not set, it is only 16 bit valid to set.	R/W	0x0

12.2.23. BSTC Status Register (BST) (0xB814_30DC)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:16	ERR_CNT	Indicate the entry number of the error read data if compare mode is setting. It can clear by CR_BCR:CRR. For example: if all return data are error BUS_64, DRAM BST8, DRAM x16, 2x DRAM Read command ERR_CNT max value = 2x (8x16)/64 = 4 BUS_128, DRAM BST8, DRAM x16 ERR_CNT max value = 2x(8x16)/128 = 2 BUS_128, DRAM BST8, DRAM x32 ERR_CNT max value = 2x(8x32)/128 = 4	R	0x0
15	RD_IN_ST	Indicate RXI310_DRAM had received read data from DRAM. It can clear by BSTC_START function.	R	0x0
14	*must_be_zero*	*All bits must be 0.*	R	0x0
13:0	ERR_FST_TH	Indicate the address of the first error data in the BSTC_RD_SRAM. It can clear by CR_BCR:CRR.	R	0x0

12.2.24. BSTC SRAM Status Register 0 (BSRAM0) (0xB814_30E0)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	*must_be_zero*	All bits must be 0.	R	0x0
29:16	WD_CNT	Indicate the word counts of the data in the write data SRAM (WD_SRAM) For example: There is one valid entry at WD_SRAM. If DATA BUS Width is 64-bit, WD_CNT is 2 If DATA BUS Width is 128-bit, WD_CNT is 4	R	CONF
15:12	*must_be_zero*	All bits must be 0.	R	0x0
11:0	CMD_CNT	Indicate the word counts of the data in the command SRAM (CMD_SRAM) For example: There is one valid entry at CMD_SRAM. (CMD_SRAM bit width is 37) CMD_CNT = [CMD_SRAM bit width / 32]= 2	R	CONF

12.2.25. BSTC SRAM Status Register 1 (BSRAM1) (0xB814_30E4)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:30	*must_be_zero*	All bits must be 0.	R	0x0
29:16	RD_CNT	Indicate the word counts of the data in the read data (RD_SRAM). For example: There is one valid entry at RD_SRAM. If DATA BUS Width is 64-bit, RD_CNT is 2 If DATA BUS Width is 128-bit, RD_CNT is 4	R	CONF
15	*must_be_zero*	All bits must be 0.	R	0x0
14:0	RG_CNT	Indicate the word counts of the data in the golden data SRAM (RG_SRAM) For example: There is one valid entry at RD_SRAM. If DATA BUS Width is 64-bit, RG_CNT is 2 If DATA BUS Width is 128-bit, RG_CNT is 4	R	CONF

12.2.26. BSTC Error Bit Register (BER) (0xB814_30E8)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	ERR_BIT	Indicate which bit ever had occurred error in auto compare function. It can clear by CR_BCR:CRR. *If DQ32 configuration is not set, it is only 16 bit valid to set.	R	0x0

12.2.27. PCTL Version ID Register (PCTL_SVN_ID) (0xB814_30F4)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:16	SVN_NUM	Indicate the SVN number of the released RTL code	R	0x8105
15:0	RELEASE_DATE	Indicate the IP release date for the particular project	R	0x0719

12.2.28. PCTL Identification Register (PCTL_IDR) (0xB814_30F8)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:16	CR_VER	The control register version number	R	0x2
15:0	CR_PCTL_DEF	If user ties this field at input port tie, it indicates user definition. Or it only shows one main support DRAM type.	R	0x3103

12.2.29. SDR PHY Delay Register 0 (DLY0) (0xB814_3100)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	DLY0	In ASIC this register would be connected to PHY interface. The bit width is 32-bit. In FPGA this register would be used to shift clock phase of clock cell as XILINX clock cell (DCM/MMCM). The bit width is according clock phase range of clock cell.	R/W	0x0

12.2.30. SDR PHY Delay Register 1 for ASIC/DCM Reset Register for FPGA (DLY1_CLK_RST) (0xB814_3104)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	DLY1_CLK_RST	In ASIC this register would be connected to PHY interface. The bit width is 32-bit. In FPGA this register would be active high to reset clock phase of clock cell as XILINX clock cell (DCM/MMCM). The bit width is 1.	R/W	0x0

12.2.31. SDR PHY Delay Clock Phase Register (DLY_CLK) (0xB814_3108)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	DLY_CLK	In ASIC this register would be connected to PHY interface. The bit width is 32-bit. In FPGA this register would be active high to select clock phase of clock cell as XILINX clock cell (DCM/MMCM). The bit width is 1.	R/W	0x0

12.2.32. SDR PHY Delay Status Register (DLY_ST) (0xB814_310C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	DLY_ST	In ASIC this register would be indicated to PHY status. The bit width is 32-bit. In FPGA this register would be used to check the status of the clock in SDR PHY. After the shift phase is stable, user can start to operate DRAM access. The bit width is 1.	R	0x0

12.2.33. Memory Information Setting Control Register (MISC) (0xB814_3224)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:11	*must_be_zero*	All bits must be 0.	R	0x0
11:10	HYBR_SIZE	Indicate the target DRAM page size (unit: Byte): 2'b000→256M, 2'b01→512M 2'b010→1G, 2'b11→2G According to CR_MISC:HYBR_SIZE to access DRAM in DQx16-bit or DQx32-bit . fOR *It needs hardware configuration set HYBR_DRAM For example: Setting HYBR_SIZE is 512MB, If bus_address >= 0x2000_0000, it only access DQx16-bit. If bus_address < 0x2000_0000, it would access DQx32-bit.	R/W	0x0
9:8	*must_be_zero*	All bits must be 0.	R	0x0
7:6	BST_SIZE	Indicate RXI310_DRAM to issue DRAM Burst size: 2'b00→4, 2'b01→8, and others are reserved. The valid setting should be according the specification hardware architecture in Table of The Architecture and DRAM Operation Support.	R/W	0x0
5:4	BANK_SIZE	Indicate the target DRAM bank size: 2'b00→2, 2'b01→4, 2'b10→8, 2'b11→16	R/W	0x0

Bit	Name	Description	Mode	Default
3:0	PAGE_SIZE	Indicate the target DRAM page size (unit: Byte): 4'b000→256B, 4'b001→512B 4'b010→1K, 4'b011→2k 4'b100→4K, 4'b101→8k 4'b110→16K, 4'b101→32k For example: If DRAM column address is 10-bit and DRAM bit number is x16, page size is equal to 2^{10}×16/8 => 2KB	R/W	0x2

12.2.34. OCP_WRAP ID number Register (OCP_WRAP_IDR) (0xB814_32A0)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:16	CR_WRAP_IDR	Indicate RXI310_OCP_WRAP ID number.	R	0x2
15:0	CR_WRAP_DEF	Indicate RXI310_DRAM number	R	0x310

12.2.35. OCP_WRAP version numbers (OCP_WRAP_VERSION) (0xB814_32A4)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:16	SVN_NUM	Indicate the SVN number of the release RTL code	R	0x8105
15:0	RELEASE_DATE	Indicate the IP release date for the particular project	R	0x0719

13. RXI-300

There are 1 register block(s) in the RXI-300. Developers can access the provided functions by programming the address-mapped registers in the corresponding register block(s).

13.1. *Features*

Key features of RXI-300 of RTL9310:

- Address hole detection
- SRAM remapping to DRAM
- Weighted round robin arbitration to DRAM

13.1.1. Interrupt Handler

The interrupt of the RXI-300 of RTL9310 platform is the union of the interrupt of RXI-300 itself .
One can handle the interrupt according to the following steps:
1. Read ERR_CODE of RXI-300. ERR_CODE[0] indicates whether address hole interrupt occurs.
2. If ERR_CODE[0] is 1'b1, then check information of the error transaction by reading ERR_ADR0, ERR_PLD0, ERR_PLD1, and ERR_TAG. After that, set INTR_CLR to 'b1 to clear the interrupt

13.2. *Register Set (Base Address: 0xB815_0000)*

13.2.1. RXI-300 Design Name Register (R300_NAME) (0xB815_0000)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	NAME	The design name of RXI-300.	R	0x0005_2300

13.2.2. RXI-300 Design Version Register (R300_VER) (0xB815_0004)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	VER	The design version of RXI-300.	R	0x0000_0300

13.2.3. RXI-300 Design Revision Register (R300_REV) (0xB815_0008)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	REV	The design reversion of RXI-300	R	0x0

13.2.4. RXI-300 instance number register (R300_INST) (0xB815_000C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	INST	The design instance number of RXI-300.	R	0x0

13.2.5. RXI-300 implementation year register (R300_IMPL_Y) (0xB815_0010)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	IMPL_Y	The implementatin year of RXI-300. Reset value depends on implementation date.	R	0x0000_2016

13.2.6. RXI-300 implementation date register (R300_IMPL_D) (0xB815_0014)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	IMPL_D	The implementatin date of RXI-300. Reset value depends on implementation date.	R	0x0614_1952

13.2.7. RXI-300 developer register (R300_DEV) (0xB815_0018)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	DEV	The developer of RXI-300.	R	0x0052_1439

13.2.8. Product number register (R300_PROD_NUM) (0xB815_001C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	PROD_NUM	Product number.	R	0x0000_9310

13.2.9. Error request payload register 0 (R300_ERR_PLD0) (0xB815_0200)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:24	ERR_BINDEX	Index in busrt length of the error request.	R	0x0
23:16	ERR_BLEN	BurstLength of the error request.	R	0x0
15:14	*must_be_zero*	All bits must be 0.	R	0x0
13:11	ERR_BTYPE	BurstSeq of the error request.	R	0x0
10:8	ERR_CMD	Cmd of the error request.	R	0x0
7:0	ERR_SRC	Error source number of the error request.	R	0x0

13.2.10. Error request payload register 1 (R300_ERR_PLD1) (0xB815_0204)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:23	ERR_REQINFO	RequestInfo of the error request.	R	0x0
22:19	*must_be_zero*	All bits must be 0.	R	0x0
18:16	ERR_SIZE	Transfer size of the error request. Available in AMBA protocol only	R	0x0
15:0	ERR_BYTEEN	ByteEn of the error request.	R	0x0

13.2.11. Error request tag register (R300_ERR_TAG) (0xB815_0208)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	ERR_TAG	TagID of the error request.	R	0x0

13.2.12. Error request address register 0 (R300_ERR_ADR0) (0xB815_020C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	ERR_ADDR0	The low [31:0] address of error request.	R	0x0

13.2.13. Error request address register 1 (R300_ERR_ADR1) (0xB815_0210)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	ERR_ADDR1	The high [63:32] address of error request.	R	0x0

13.2.14. Error code register (R300_ERR_CODE) (0xB815_0230)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:8	*must_be_zero*	All bits must be 0.	R	0x0
7:0	ERR_CODE	The error code register of rxi300. 8'h01: Address hole error.	R	0x0

13.2.15. RXI-300 interrupt clear register (R300_INTR_CLR) (0xB815_023C)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:1	*must_be_zero*	All bits must be 0.	R	0x0
0	INTR_CLR	RXI-300 interrupt clear. Writing 1'b1 to the register tr iggers a 1-cycle pulse interrupt clear to RXI-300. Note that this register is write-only, the return value of reading this register is meaningless	RW1C	0x0

13.2.16. Weighted RR control register (R300_WRR_CTRL) (0xB815_0400)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:16	*must_be_zero*	All bits must be 0.	R	0x3333
15:14	Slot 0 Owner	Specify the port ID to occurpy slot 0	R/W	0x0
13:12	Slot 1 Owner	Specify the port ID to occurpy slot 1	R/W	0x2
11:10	Slot 2 Owner	Specify the port ID to occurpy slot 2	R/W	0x1
9:8	Slot 3 Owner	Specify the port ID to occurpy slot 3	R/W	0x0
7:6	Slot 4 Owner	Specify the port ID to occurpy slot 4	R/W	0x0
5:4	Slot 5 Owner	Specify the port ID to occurpy slot 5	R/W	0x2
3:2	Slot 6 Owner	Specify the port ID to occurpy slot 6	R/W	0x1
1:0	Slot 7 Owner	Specify the port ID to occurpy slot 7	R/W	0x0

13.2.17. Base address of SRAM (R300_SRAM_REMAP) (0xB815_0404)

(This register does not provide byte access)

Bit	Name	Description	Mode	Default
31:0	Base_Addr	SRAM base address.	R/W	0x1F00_0000

14. Global Interrupt Controller of MIPS InterAptiv

14.1. Features

This chapter describes the Global Interrupt Controller (GIC) included in the MIPsIA Coherent Processing System. RTL9310 supports 48 interrupt sources which the last 8 sources are for inter processor interrupts (IPI).

External interrupt sources enter the GIC through the Shared Section. Below are related registers:

- **Global Interrupt Pending registers (GIC_SH_PENDi+31_i)**: This is the register that software will need to probe to discern the source of the interrupt.
- **Global Interrupt Mask registers (GIC_SH_MASKi+31_i)**: For software to temporarily disable any particular interrupt source.
- **Global Interrupt Set Mask Register (GIC_SH_SMASKi+31_i)**: Set individual bits within the mask registers to avoid any read-modify-write hazards within a MIPsIA system
- **Global Interrupt Reset Mask Register (GIC_SH_RMASKi+31_i)**: Clear individual bits within the mask registers to avoid any read-modify-write hazards within a MIPsIA system
- **VPE control registers (GIC_SH_MAPi_VPEj+31_j)**: Each of the external interrupt sources is then assigned to a particular VPE.
- **Pin control registers (GIC_SH_MAPi_PIN)**: The interrupt source is then mapped to a particular interrupt pin (Int[5:0] or NMI or YQ[15:0]) for that particular VPE.

The routing of interrupt as below:

Index *i*	Interrupt Source	Description
0	TC0	Timer/Counter #0 Interrupt
1	TC1	Timer/Counter #1 Interrupt
2	TC2	Timer/Counter #2 Interrupt
3	TC3	Timer/Counter #3 Interrupt
4	TC4	Timer/Counter #4 Interrupt
5	TC5	Timer/Counter #5 Interrupt
6	TC6	Timer/Counter #6 Interrupt
7	*Reserved for TC7*	*Reserved for Timer/Counter #7 Interrupt*
8	WDT_PH1TO	Watchdog Timer Phase 1 Timeout Interrupt
9	WDT_PH2TO	Watchdog Timer Phase 2 Timeout Interrupt
10	*Reserved*	*N/A*
11	RXI-300	RXI-300 ~~Sheipa (Bus Fabric)~~ Interrupt
12-14	*Reserved*	*N/A*
15	SWCORE	Switch Core interrupt
16	NIC	NIC Interrupt
17-19	*Reserved*	*N/A*
20	GPIO_ABCD	GPIO_ABCD Interrupt
21	*Reserved*	*N/A*
22	UART0	UART0 Interrupt
23	UART1	UART1 Interrupt
24-25	*Reserved*	*N/A*
26	*Reserved for TC2_DELAY_INT*	*Reserved for Delayed Interrupt with Timer/Counter 2*
27	*Reserved for TC3_DELAY_INT*	*Reserved for Delayed Interrupt with Timer/Counter 3*
28	TC4_DELAY_INT	Delayed Interrupt with Timer/Counter 4
29	TC5_DELAY_INT	Delayed Interrupt with Timer/Counter 5
30	TC6_DELAY_INT	Delayed Interrupt with Timer/Counter 6
31	*Reserved for TC7_DELAY_INT*	*Reserved for Delayed Interrupt with Timer/Counter 7*
32	LXSTO	LX Slave 0/1 Timeout Interrupt
33	LXMTO	LX Master 0/1Timeout Interrupt
34	BTG	LX0/1 BTG/GDMA Interrupt
35	*Reserved*	*N/A*

Index *i*	Interrupt Source	Description
36	USB_H2	USB 2.0 Host Interrupt
37	SPI_NAND	SPI-NAND DMA Interrupt
38	ECC	ECC Controller Interrupt
39	*Reserved*	*N/A*
40-47	*Reserved for IPI*	*For MIPS InterAptiv Inter Processor Interrupts*

14.2. *Register Set (Base Address: Configurable)*

Refer to MIPS32R InterAptiv Coherent Processing System User's Manual, for more detailed information.

Base Address: 0xBFBD_0000			
Offset	**Size (byte)**	**Name**	**Description**
0x300	4	Global Interrupt Reset Mask Register (GIC_SH_RMASK31_0)	Masks off (disables) that interrupt.
0x380	4	Global Interrupt Set Mask Register (GIC_SH_SMASK31_00)	Masks (enable) for that interrupt.
0x400	4	Global Interrupt Mask Register (GIC_SH_MASK31_00)	Shows the enabled global interrupts.
0x0480	4	Global Interrupt Pending Register (GIC_SH_PEND31_00)	Shows the pending global interrupts before masking.
0x0500 ~ 0x5C0	4 *48	Global Interrupt Map Src to Pin Registers (GIC_SH_MAP0_PIN ~ GIC_SH_MAP47_PIN)	Maps this interrupt source to a particular pin -within int[5:0] or NMI or YQ[31:0]
0x2000 ~ 0x2604	4 *2 *48	Global Interrupt Map Src to VPE Register (GIC_SH_MAP*0*_VPE31_*0* ~ GIC_SH_MAP47_VPE63_*32*)	Assigns this interrupt source to a particular VPE.

14.2.1. Global Interrupt Reset Mask Register (GIC_SH_RMASK31+i*32_i*32 Offset 0xBFBD_0300+0x4*i)

Register Fields		Description	Mode	Default
Name	**Bits**			
*GIC_SH_RMASK*i+31_i	31:0	Each bit in this register represents an interrupt source. Writing this register with a 0x1 in any bit position(s) will cause only the corresponding bit/interrupt(s) in the Global Interrupt Mask Register to be reset (value->0). This is used by software to temporarily disable interrupts.	W	Undefined

14.2.2. Global Interrupt Set Mask Register (GIC_SH_SMASK31+i*32_i*32 Offset 0xBFBD_0380+0x4*i)

Register Fields		Description	Mode	Default
Name	**Bits**			
*GIC_SH_SMASK*i+31_i	31:0	Each bit in this register represents an interrupt source. Writing this register with a 0x1 in any bit position(s) will cause only the corresponding bit/interrupt(s) in the Global Interrupt Mask Register to be set(value->0x1). This is used by software to enable interrupts.	W	Undefined

14.2.3. Global Interrupt Mask Register (GIC_SH_MASK31+i*32_i*32 Offset 0xBFBD_0400+0x4*i)

Register Fields	Description	Mode	Default

Name	Bits			
*GIC_SH_MASK*i+31_i	31:0	Each bit in this register represents an interrupt source. Reports which of the external interrupt sources are enabled. Used by software to determine which interrupt sources are currently enabled.	R	0x0

14.2.4. Global Interrupt Pending Register (GIC_SH_PEND31+i*32_i*32 Offset 0xBFBD_0480+0x4*i)

Register Fields		Description	Mode	Default
Name	Bits			
*GIC_SH_PEND*i+31_i	31:0	Each bit in this register represents an interrupt source. Reports which of the external interrupt sources are asserted/pending before masking. Used by software to find the external source that caused the CPU interrupt.	R	Undefined

14.2.5. Global Interrupt Map to PIN Register (GIC_SH_MAPi_PIN Offset 0xBFBD_0500+i*0x4)

Register Fields		Description	Mode	Default
Name	Bits			
MAP_TO_PIN	31	If this bit is set, this interrupt source is mapped to a VPE interrupt pin (specified by the *MAP* field below). Only one of the *MAP_TO_PIN*, *MAP_TO_NMI*, or *MAP_TO_YQ* bits can be set at any one time.	RW	0x1
MAP_TO_NMI	30	If this bit is set, this interrupt source is mapped to NMI. Only one of the *MAP_TO_PIN*, *MAP_TO_NMI*, or *MAP_TO_YQ* bits can be set at any one time.	RW	0x0
MAP_TO_YQ	29	If this bit is set, this interrupt source is mapped to an MT Yield Qualifier pin. Only one of the *MAP_TO_PIN*, *MAP_TO_NMI*, or *MAP_TO_YQ* bits can be set at any one time.	RW	0x0
RESERVED	28:6	Read as 0x0. Writes ignored. Must be written with a value of 0x0.	-	0
MAP	5:0	When the *MAP_TO_PIN* bit is set, this field contains the encoded value of the VPE interrupts signals *Int[63:0]*. The user should only use values of 0 to 5 (decimal). When *MAP_TO_YP* is set, this field contains the encoded signal selection of the Yield Qualifier. **Encoding** / **Meaning** 0x0 — *YR_ysi[0]* 0x1 — *YR_ysi[1]* … — … 0xF — *YR_ysi[15]* 0x10 — NULL … — … 0x3F — NULL Since YQ is per-CORE rather than per-VPE, software needs to apply proper protection across VPEs by using, for example, the cop0 *YQMask*.register	RW	0

14.2.6. Global Interrupt Map to VPE (GIC_SH_MAPi_VPE31+32*j-32*j Offset+0xBFBD_2000+i*0x20+j*0x4)

Register Fields		Description	Mode	Default
Name	Bits			

Register Fields		Description	Mode	Default
Name	Bits			
*GIC_SH_MAPi_VPE*j+31_j	31:0	If *GIC_SH_MAPi_PIN[MAP_TO_YQ]* is clear, each bit in this register represents a VPE. Writing a 0x1 to any bit in this register will cause the interrupt source to be routed to the corresponding VPE. If *GIC_SH_MAPi_PIN[MAP_TO_YQ]* is set, each bit in this register represents a CPU core. Writing a 0x1 to any bit in this register will cause the interrupt source to be routed to the corresponding CPU core. For all *GIC_SH_MAPi_VPE** registers, only one bit may be set at one time; that is, an interrupt source will be routed to one and only one VPE or CPU core.	W	0