龙芯中科
LOONGSON TECHNOLOGY

LOONGSON

Loongson 3a2000/3b2000 processor user manual

Part i

Multicore processor architecture, register description and system software programming guide

V1.7

02, 2017

Loongson technology co. LTD

北京市海淀区温泉镇中关村环保科技示范园龙芯产业园2号楼 100095
Loongson Industrial Park, building 2，Zhongguancun environmental protection park
Haidian District，Beijing
www.loongson.cn

Copyright statement

disclaimer

This document only provides periodic information, and the content can be updated at any time according to the actual situation of the product without prior notice. The company shall not be liable for any direct or indirect loss caused by improper use of the documents.

Loongson technology co. LTD

Loongson Technology Corporation Limited
Address: Loongson Industrial Park, Building No. 2, longson Industrial Park, zhongguancun environmental science and technology demonstration Park, haidian district, Beijing
Zhongguancun Environmental Protection Park, Haidian District, Beijing
Tel: 010-62546668 Fax: 010-62600826

Reading guide

Loongson 3a2000/3b2000 processor user manual is divided into the first volume and the second volume.

The first part introduces the architecture and register description of the longson 3a2000/3b2000 multi-core processor, and gives a detailed description of the chip system architecture, the functions and configuration of the main modules, the register list and the bit domain.
Loongson 3a2000/3b2000 processor user manual, the second volume, introduces loongson in detail from the perspective of system software developers

3a2000/3b2000 USES GS464e high performance processor core.

Revision history

Document update record	The document name:	Loongson 3a2000/3b2000 processor user manual - the days
	The version number	V1.7
	The founders:	Chip r&d department
	Date of creation:	2017-02-14

Update history

The serial number	Updated date	The version number	Update the content
1	2015-03-31	V1.0	The initial release
2	2015-05-27	V1.1	Add a description of hardware and software changes
3	2015-07-17	V1.2	Supplementary register description
4	2016-02-23	V1.3	Fixed some register description errors, modified "software and hardware modification instructions"
5	2016-04-13	V1.4	Fix the HT receive window description, modify "software and hardware change description"
6	2016-06-21	V1.5	Fix CLKSEL[15] description
7	2016-10-10	V1.6	Added software configuration HT frequency description
8	2017-02-14	V1.7	Modify the "hardware and software change specification" and add the SPI address space specification
9			

Manual information feedback: service@loongson.cn

Problem feedback web site, http://bugs.loongnix.org/, also can be submitted to our chip problem in the process of product use, and obtain technical support.

目录

Figure orders to record

Table item record

1 An overview of the

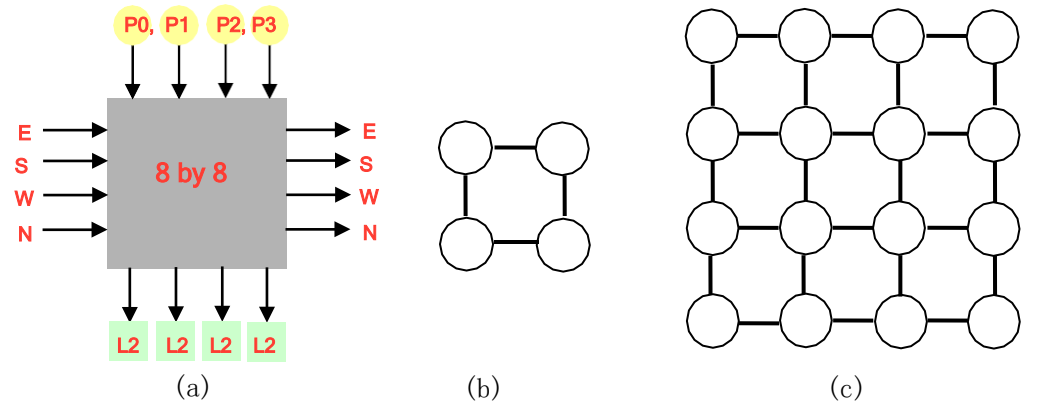
1.1 Introduction to loongson series processors

The loongson processor mainly includes three series. Loongson 1 processor and its IP series are mainly for embedded applications, loongson 2 superstandard processor and its IP series are mainly for desktop applications, and loongson 3 multi-core processor series are mainly for server and high-performance machine applications. According to the application needs, part of the loongson 2 can also be oriented to part of the high-end embedded should With, part of low - end loongson 3 can also face part of the desktop application. The three series will develop in parallel.

Based on the scalable multi-core interconnect architecture, the loong chip 3 multi-core series integrates multiple high-performance processor cores and a large number of level-2 caches on a single chip, and interconnects multiple chips through high-speed I/O interfaces to form a larger system.

The retractable interconnection structure adopted by loongson 3 is shown in figure 1-1 below.Loongson 3 and multi - chip system are adopted 2

D mesh interconnection structure, in which each node is composed of 8*8 cross switches, each cross switch connects four processor cores and four Shared Cache, and interconnects with other nodes in the four directions of east (E), south (N), west (W) and north (N). Therefore, a 2*2 mesh can connect 16 processor cores, and a 4*4 mesh can connect 64 processor cores.



Loongson no. 3 node and 2d interconnect structure, (a) node structure, (b) 2*2 mesh network connecting 16 processors, (c)

Figure 1-1 longson 3 system structure

The node structure of loongson 3 is shown in figure 1-2 below. Each node has two AXI cross-switches connected to the processor and Shared

Cache, memory controller, and IO controller. The first level is a AXI cross Switch (called the X1 Switch, or X1) that connects the processor to the Shared Cache. The second level cross Switch (called the X2 Switch) connects the Shared Cache to the memory controller.

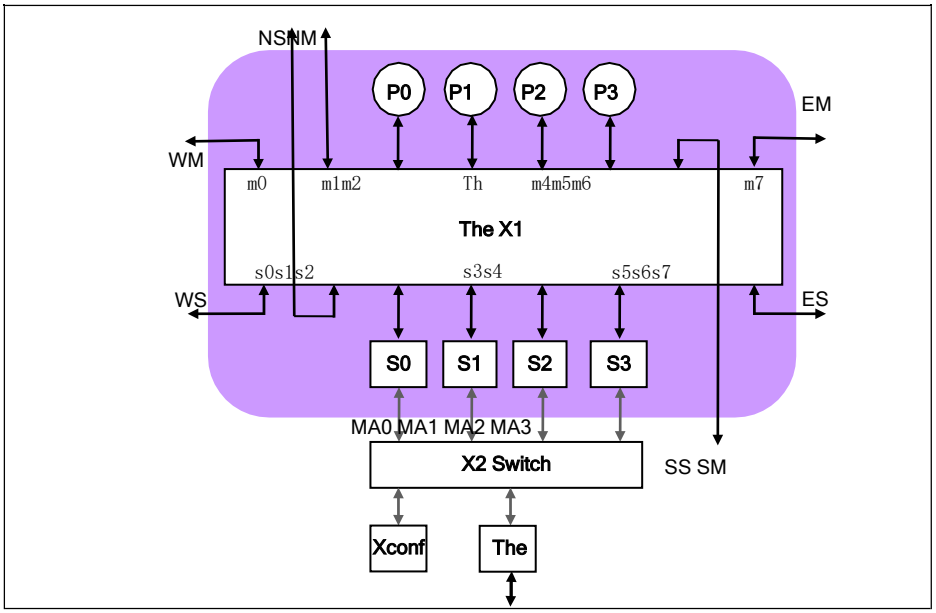


Figure 1-2 structure of longson node3

At most 8*8 X1 crossovers are connected to four GS464 processor cores via four Master ports at each node(P0, P1, P2, P3 in the figure), connect the four interleave Shared Cache blocks (S0, S1, S2, S3 in the figure) with four Slave ports, and connect other nodes or IO nodes (EM/ES, SM/SS, WM/WS, NM/NS in the figure) with four Master/Slave ports in the east, south, west, and north directions.

The X2 cross-switch connects four Shared caches through four Master ports, at least one Slave port to a memory controller, and at least one Slave port to a cross-switch configuration module (Xconf) to configure the local node's X1 and X2 address Windows. You can also connect more memory controllers and IO ports as needed.

1.2 introduction of loongson 3a2000/3b2000

Loongson 3a2000/3b2000 is an upgraded version of the loongson 3A1000 quad-core processor. The package pin is compatible with loongson 3A1000. The godson 3a2000/3b2000 is a processor configured as a single node with 4 cores. It is manufactured by the 40nm process and its main operating frequency is 800MHz-1GHz. The main technical features are as follows:

- 4 64-bit quad-emission superscalar GS464e high-performance processor cores are integrated on the chip.
- 4 MB split Shared three-level Cache(composed of 4 individual modules, each with a capacity of 1MB);

- Maintain Cache consistency for multi-core and I/O DMA access through directory protocols;
- Two 64-bit ddr2/3 controllers with ECC and 667MHz are integrated on the chip.
- 3B2000 chips are integrated with 2 16-bit 1.6ghz HyperTransport controllers (HT);
- HT1 in 3A2000 is a 16-bit 1.6ghz HT controller, HT0 is not available;
- Each 16-bit HT port is split into two 8-way HT ports for use.
- Integrated 32-bit 33MHz PCI;
- In - chip integration 1 LPC, 2 UART, 1 SPI, 16 GPIO

  interface. Compared with loongson 3A1000, the main

  improvements are as follows:
- Processor core microstructure upgrade;
- Memory controller structure, frequency upgrade;
- The structure and frequency of HT controller were upgraded comprehensively.
- Comprehensive upgrade of internal interconnection structure;
- Comprehensive upgrade of external extended interconnection structure;
- Support SPI startup function;
- Support full chip software frequency configuration;
- Full - chip performance optimization.

The overall architecture of loongson 3A2000 is based on two-level interconnection, as shown in figure 1-3 below.

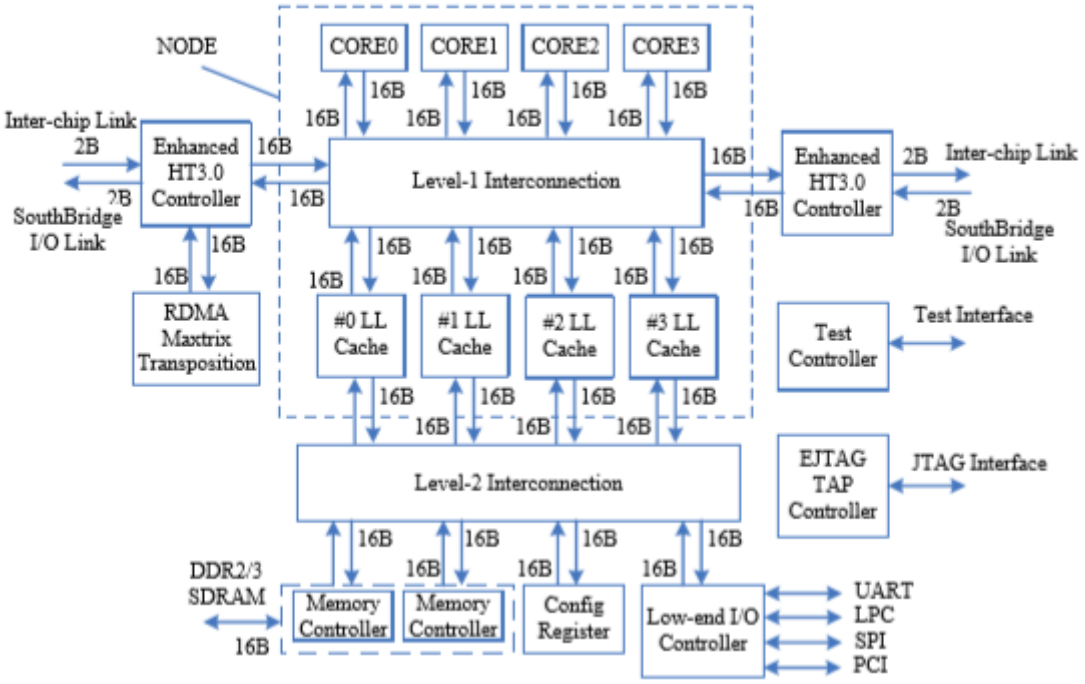


Figure 1-3 structure of loongson 3A2000 chip

Level 1 interconnect USES 6x6 cross-switches to connect four GS464e cores (as Master devices), four Shared Cache modules (as Slave devices), and two IO ports (one Master and one Slave for each port). Each IO port connected by the level 1 interconnect switch is connected to a 16-bit HT controller, and each 16-bit HT port can also be used as two 8-bit HT ports. The HT controller is connected to a level 1 interconnect switch by a DMA controller, which is responsible for the DMA control of IO and the maintenance of inter-chip consistency. The loongson 3's DMA controller can also be configured to prefetch and matrix transpose or shift.

The second level interconnection USES a 5x4 cross-switch to connect four Shared Cache modules (as the main device) and two ddr2/3

Memory controller, low speed and high speed I/O (including PCI, LPC, SPI, etc.) and the configuration register module inside the chip. The above two level interconnect switches all adopt the data channel separated by read and write. The data channel width is 128bit and works at the same place

The same frequency is used to provide high-speed on-chip data transmission.

The difference between 3B2000 and 3A2000 processors is that 3B2000 supports HT0 as the consistent interconnect interface. Based on the loongson 3 extensible interconnect architecture, 4 pieces of four-core loongson 3B2000 can be connected through HT port to form a NUMA structure with 16 cores on 4 chips. The 3A2000 processor only supports the IO use of the HT1 controller.

The following will not make a distinction between 3B2000 and 3A2000, referred to as longson 3A2000.

1.3 loongson 3A2000 commercial and industrial grade chip description

Loongson 3A2000 chip has two kinds of industrial grade and commercial grade. Its main features are as follows:

configuration	Commercial grade	industrial-grade
Working temperature	0 ℃ ~ 70 ℃	- 40 ℃ ~ 85 ℃
Whether screening	-	Square root
Quality consistency test	-	Square root
Quality consistency test standard	-	GB 4937-1995

As with most semiconductor devices, the failure rate of the loongson 3A chip conforms to the bathtub curve model. In order to ensure a more long-term, stable and reliable operation, and to adapt to the more stringent environmental temperature requirements, loongson 3A industrial-grade chip was selected for reliability, so as to eliminate the early failure of the chip. This reliability screening is 100% trial, through which the chips are screened for industrial-grade applications.

The operating voltage of 3A2000 commercial - and industrial-grade chips is slightly different. The working voltage of the industrial-grade chip is required to be 1.15v, and the jitter of the power supply is less than 50mV. Commercial - grade chips require working voltage of 1.25V, power jitter less than 50mV.

The main contents of longson 3A screening test are as follows:

2 System configuration and control

Screening program	Methods and conditions (summary)	requirements
1, visual inspection	Clear identification, no contamination, no oxidation of welding ball, chip intact	100%
2. Stable baking	125 ℃ for 24 h	100%
3. The temperature changes rapidly	10 cycles at maximum and minimum storage temperatures	100%
4. Serial number		100%
5, intermediate (before the old) electrical test	The normal temperature	100%
6, aging	TC = 85 ℃, 160 h	100%
7. Intermediate (after aging) electrical test	The normal temperature	100%
8. Percentage of defective products allowed (PDA) To calculate	PDA≤5%, normal temperature, when 5% < PDA≤10%, can be heavy New submissions, but only once	All batch
9. Terminal electrical test	Three temperature, record all test data	100%
10. External visual inspection	Clear identification, no contamination, no oxidation of welding ball, chip intact	100%

2.1 Chip working mode

According to the structure of the composing system, loongson 3A2000 mainly includes three working modes:

- Single chip mode. The system consists of only one loongson 3A2000, which is a symmetric multiprocessor system (SMP).
- Multi-chip interconnection model. The system contains 2 or 4 pieces of loongson 3A2000, which are interlinked through the HT port of loongson 3A2000. It is a non-uniform memory multi-processor system (cc-numa).
- Large-scale interconnection model. A large scale non-uniform memory access multiprocessor system (cc-numa) is constructed by means of a special extension bridge.

2.2 Control pin instructions

The main control pins include DO_TEST, ICCC_EN, NODE_ID[1:0], CLKSEL[15:0], and PCI_CONFIG.

Table 2-1 control pin description

<table>
<tr><th>signal</th><th>Pull up and down</th><th>role</th></tr>
<tr><td>DO_TEST</td><td>On the pull</td><td>1 'b1 represents the functional mode
1 'b0 represents the test mode</td></tr>
<tr><td>ICCC_EN</td><td>The drop-down</td><td>1 'b1 represents the consistent interconnection mode of multiple chips
1 'b0 stands for single chip mode</td></tr>
<tr><td>NODE_ID [1:0]</td><td></td><td>Represents the processor number in the multichip consistent interconnection mode</td></tr>
<tr><td>CLKSEL [15:0]</td><td></td><td>HT clock control
<table>
<tr><th>signal</th><th>role</th></tr>
<tr><td>CLKSEL [15]</td><td>1 'b1 means that HT PLL frequency is only set by hardware
1 'b0 means HT PLL frequency can be set by software</td></tr>
<tr><td>CLKSEL [14]</td><td>1 'b1 represents HT PLL with normal clock input
1 'b0 means HT PLL USES differential clock input</td></tr>
</table></td></tr>
</table>

CLKSEL [12]	2 'b00 represents the PHY clock of 1.6GHZ 2 'b01 represents the PHY clock of 3.2GHZ 2 'b10 represents the PHY clock of 1.2GHz 2 'b11 represents the PHY clock of 2.4GHz
CLKSEL [10]	2 'b00 means that the HT controller clock is the PHY clock 8 frequency division 2 'b01 means that the HT controller clock is the PHY clock 4 frequency division 2 'b10 means that the HT controller clock is the PHY clock 2 frequency division 2 'b11 means that HT controller clock is SYSCLOCK

Note: when CLKSEL[13:10] == 4 'b1111, the HT controller clock is bypass mode, and the external input 100MHz reference clock is used directly

MEM clock control

The signal function

CLKSEL [will]	5 'b11111 means MEM clock directly adopts memclk 5 'b01111 means the MEM clock is set by software. See the setting method Section 2.6 shows In other cases, the MEM clock is Memclk * (clksel [and] + 30)/(clksel [9] + 3) Note: Memclk *(clksel[8:5]+30) must be 1.2ghz ~ 3.2ghz memclk as the input reference clock, must be 20~40MHz

CORE clock control

signal	role

		CLKSEL [Wednesday]	5 'b11111 means that the CORE clock directly adopts sysclk 5 'b011xx means CORE clock is set by software. See the setting method Section 2.6 description. 5 'b01111 is in normal working mode, otherwise it is in debugging mode 5 'b0110x means that the FIFO depth is set to 2 5 'b011x0 represents the DCDL control mode. In other cases, the CORE clock is sysclk*(clksel[3:0]+30)/(clksel[4]+1). Note: Sysclk *(clksel[3:0]+30) must be 1.2ghz ~ 3.2ghz Sysclk is the input reference clock and must be 20~40MHz
PCI_CONFIG [away]		IO configuration control The 7HT bus is forced to start in 1.0 mode 6:4 needs to be set to 000 3PCI main device mode	
		2 needs to be set to 0 1 use external PCI mediation 0 use SPI to start the function	

2.3 The Cache consistency

The loongson 3A2000 is maintained by the hardware for Cache consistency between the processor and the I/O accessed through the HT port, but the hardware does not maintain the Cache consistency of the I/O devices accessed through PCI into the system. At the time of driver development, the software is required to maintain Cache consistency for DMA (Direct Memory Access) transfers to devices accessed through PCI.

2.4 Physical address space distribution at the system node level

The system physical address distribution of the loongson 3 series processor adopts globally accessible hierarchical addressing design to ensure

System development extension compatibility. The overall system physical address width is 48 bits. According to the height of the address, the whole address is empty

It is evenly distributed among 16 nodes, that is, each node is allocated with a 44-bit address space.

Logodson 3A2000 processor can directly use 4 chips to build cc-numa system. The processor number of each chip is determined by the pin NODEID. The address space distribution of each chip is as follows:

Table 2-2 node-level system global address distribution

Chip node number (NODEID)	Address [47:44]	The starting address	End address
0	0	0 x0000_0000_0000	0 x0fff_ffff_ffff
1	1	0 x1000_0000_0000	0 x1fff_ffff_ffff
2	2	0 x2000_0000_0000	0 x2fff_ffff_ffff
3	3	0 x3000_0000_0000	0 x3fff_ffff_ffff

The godson 3A2000 adopts a single-node 4-core configuration, so the corresponding addresses of the DDR memory controller, HT bus and PCI bus integrated by the godson 3A2000 chip are all contained in the 44-bit position from 0x0 (inclusive) to 0x1000_0000_0000 (not inclusive) inside each node, and the 44-bit address space is further uniformly distributed to up to 8 devices connected within the node. The lower 43-bit address is owned by four Shared Cache modules, and the higher 43-bit address is further determined by the [43:42] bit of the address Distributed to devices connected to four directional ports. Depending on the chip and system configuration, if there is no slave device connected to a port, the corresponding address space is reserved address space and access is not allowed.

The address space of the internal first-level cross-switch of loongson 3A2000 chip is as follows:

Table 2-3 address distribution within nodes

equipment	Address [43:41]	The initial address within the node	End of node address
The Shared Cache	0,1,2,3	0 x000_0000_0000	0 x7ff_ffff_ffff
HT0 controller	6	0 xc00_0000_0000	0 xdff_ffff_ffff
HT1 controller	7	0 xe00_0000_0000	0 xfff_ffff_ffff

Unlike directional port mapping, loongson 3A2000 can determine the cross-addressing method of Shared Cache based on the actual access behavior of the application. The four Shared Cache modules in the node correspond to a total of 43 bit address Spaces, and the address space corresponding to each module is determined according to a certain two selection bits of the address bit, which can be dynamically configured and modified by the software. A configuration register named SCID_SEL is set in the system to determine the address selection bit, as shown in the following table. By default, it is distributed in a [7:6] status hash, where the address [7:6] determines the corresponding Shared Cache number. The register address is 0x3FF00400.

Table 2-4 address distribution within nodes

SCID_SEL	Address bit selection	SCID_SEL	Address bit selection
4 'h0	7:6	4 'h8	"
4 'h1	9:8	4 'h9	Thus for
4 'h2	"	4 'ha	But after
4 'h3	She answered	4 'hb	then
4 'h4	The lowest	4 'hc	charm
4 'h5	"	4 'hd	33:32
4 'h6	7	4 'he	"
4 'h7	mark	4 'hf	meanwhile

2.5 Address routing distribution and configuration

The routing of loongson 3A2000 is mainly realized through the two-stage crossover switch of the system. A level 1 cross switch can configure the routing of requests received by each Master port. Each Master port has 8 address Windows, which can complete the target routing of 8 address Windows. Each address window consists of three 64-bit registers, BASE, MASK and MMAP. BASE is aligned with K

bytes. The MASK used a format similar to the network MASK with a high position of 1.The lower three bits of MMAP represent the number of the corresponding target Slave port, MMAP[4] means the allowed reference, MMAP[5] means the allowed block read, MMAP[6] means the allowed interleaved access enable to Scache, and MMAP[7] means the window enable.

Table 2-5 MMAP field corresponding to this spatial access property

[7]	[6]	[5]	[4]
The window can make	Allows interlaced access to SCACHE, valid with Slave zero, and routes requests to hit window addresses as configured in the previous section SCID_SEL	Allow the block read	Allowed to take to

Window hit formula :(IN_ADDR & MASK) == BASE

Because the loongson 3 adopts fixed route by default, the configuration window is closed when it is powered on and started, so the system software is required to enable it to be configured when it is in use.

The address window conversion register is shown in the following table.

Table 2-6 level 1 cross switch address window register table

address	register	address	register
0 x3ff0_2000	CORE0_WIN0_BASE	0 x3ff0_2100	CORE1_WIN0_BASE
0 x3ff0_2008	CORE0_WIN1_BASE	0 x3ff0_2108	CORE1_WIN1_BASE
0 x3ff0_2010	CORE0_WIN2_BASE	0 x3ff0_2110	CORE1_WIN2_BASE
0 x3ff0_2018	CORE0_WIN3_BASE	0 x3ff0_2118	CORE1_WIN3_BASE
0 x3ff0_2020	CORE0_WIN4_BASE	0 x3ff0_2120	CORE1_WIN4_BASE
0 x3ff0_2028	CORE0_WIN5_BASE	0 x3ff0_2128	CORE1_WIN5_BASE
0 x3ff0_2030	CORE0_WIN6_BASE	0 x3ff0_2130	CORE1_WIN6_BASE
0 x3ff0_2038	CORE0_WIN7_BASE	0 x3ff0_2138	CORE1_WIN7_BASE
0 x3ff0_2040	CORE0_WIN0_MASK	0 x3ff0_2140	CORE1_WIN0_MASK
0 x3ff0_2048	CORE0_WIN1_MASK	0 x3ff0_2148	CORE1_WIN1_MASK
0 x3ff0_2050	CORE0_WIN2_MASK	0 x3ff0_2150	CORE1_WIN2_MASK
0 x3ff0_2058	CORE0_WIN3_MASK	0 x3ff0_2158	CORE1_WIN3_MASK
0 x3ff0_2060	CORE0_WIN4_MASK	0 x3ff0_2160	CORE1_WIN4_MASK
0 x3ff0_2068	CORE0_WIN5_MASK	0 x3ff0_2168	CORE1_WIN5_MASK
0	CORE0_WIN6_MASK	0	CORE1_WIN6_MASK

x3ff0_2070		x3ff0_2170	
0 x3ff0_2078	CORE0_WIN7_MASK	0 x3ff0_2178	CORE1_WIN7_MASK
0 x3ff0_2080	CORE0_WIN0_MMAP	0 x3ff0_2180	CORE1_WIN0_MMAP
0 x3ff0_2088	CORE0_WIN1_MMAP	0 x3ff0_2188	CORE1_WIN1_MMAP
0 x3ff0_2090	CORE0_WIN2_MMAP	0 x3ff0_2190	CORE1_WIN2_MMAP
0 x3ff0_2098	CORE0_WIN3_MMAP	0 x3ff0_2198	CORE1_WIN3_MMAP
0 x3ff0_20a0	CORE0_WIN4_MMAP	0 x3ff0_21a0	CORE1_WIN4_MMAP
0 x3ff0_20a8	CORE0_WIN5_MMAP	0 x3ff0_21a8	CORE1_WIN5_MMAP
0 x3ff0_20b0	CORE0_WIN6_MMAP	0 x3ff0_21b0	CORE1_WIN6_MMAP
0 x3ff0_20b8	CORE0_WIN7_MMAP	0 x3ff0_21b8	CORE1_WIN7_MMAP
0 x3ff0_2200	CORE2_WIN0_BASE	0 x3ff0_2300	CORE3_WIN0_BASE
0 x3ff0_2208	CORE2_WIN1_BASE	0 x3ff0_2308	CORE3_WIN1_BASE
0 x3ff0_2210	CORE2_WIN2_BASE	0 x3ff0_2310	CORE3_WIN2_BASE
0 x3ff0_2218	CORE2_WIN3_BASE	0 x3ff0_2318	CORE3_WIN3_BASE
0 x3ff0_2220	CORE2_WIN4_BASE	0 x3ff0_2320	CORE3_WIN4_BASE
0 x3ff0_2228	CORE2_WIN5_BASE	0 x3ff0_2328	CORE3_WIN5_BASE
0 x3ff0_2230	CORE2_WIN6_BASE	0 x3ff0_2330	CORE3_WIN6_BASE
0 x3ff0_2238	CORE2_WIN7_BASE	0 x3ff0_2338	CORE3_WIN7_BASE
0 x3ff0_2240	CORE2_WIN0_MASK	0 x3ff0_2340	CORE3_WIN0_MASK
0 x3ff0_2248	CORE2_WIN1_MASK	0 x3ff0_2348	CORE3_WIN1_MASK
0 x3ff0_2250	CORE2_WIN2_MASK	0 x3ff0_2350	CORE3_WIN2_MASK
0 x3ff0_2258	CORE2_WIN3_MASK	0 x3ff0_2358	CORE3_WIN3_MASK
0 x3ff0_2260	CORE2_WIN4_MASK	0 x3ff0_2360	CORE3_WIN4_MASK

0 x3ff0_2268	CORE2_WIN5_MASK	0 x3ff0_2368	CORE3_WIN5_MASK
0 x3ff0_2270	CORE2_WIN6_MASK	0 x3ff0_2370	CORE3_WIN6_MASK
0 x3ff0_2278	CORE2_WIN7_MASK	0 x3ff0_2378	CORE3_WIN7_MASK
0 x3ff0_2280	CORE2_WIN0_MMAP	0 x3ff0_2380	CORE3_WIN0_MMAP
0 x3ff0_2288	CORE2_WIN1_MMAP	0 x3ff0_2388	CORE3_WIN1_MMAP
0 x3ff0_2290	CORE2_WIN2_MMAP	0 x3ff0_2390	CORE3_WIN2_MMAP
0 x3ff0_2298	CORE2_WIN3_MMAP	0 x3ff0_2398	CORE3_WIN3_MMAP
0 x3ff0_22a0	CORE2_WIN4_MMAP	0 x3ff0_23a0	CORE3_WIN4_MMAP
0 x3ff0_22a8	CORE2_WIN5_MMAP	0 x3ff0_23a8	CORE3_WIN5_MMAP
0 x3ff0_22b0	CORE2_WIN6_MMAP	0 x3ff0_23b0	CORE3_WIN6_MMAP
0 x3ff0_22b8	CORE2_WIN7_MMAP	0 x3ff0_23b8	CORE3_WIN7_MMAP
0 x3ff0_2600	HT0_WIN0_BASE	0 x3ff0_2700	HT1_WIN0_BASE
0 x3ff0_2608	HT0_WIN1_BASE	0 x3ff0_2708	HT1_WIN1_BASE
0 x3ff0_2610	HT0_WIN2_BASE	0 x3ff0_2710	HT1_WIN2_BASE
0 x3ff0_2618	HT0_WIN3_BASE	0 x3ff0_2718	HT1_WIN3_BASE

0 x3ff0_2620	HT0_WIN4_BASE	0 x3ff0_2720	HT1_WIN4_BASE
0 x3ff0_2628	HT0_WIN5_BASE	0 x3ff0_2728	HT1_WIN5_BASE
0 x3ff0_2630	HT0_WIN6_BASE	0 x3ff0_2730	HT1_WIN6_BASE
0 x3ff0_2638	HT0_WIN7_BASE	0 x3ff0_2738	HT1_WIN7_BASE
0 x3ff0_2640	HT0_WIN0_MASK	0 x3ff0_2740	HT1_WIN0_MASK
0 x3ff0_2648	HT0_WIN1_MASK	0 x3ff0_2748	HT1_WIN1_MASK
0 x3ff0_2650	HT0_WIN2_MASK	0 x3ff0_2750	HT1_WIN2_MASK
0 x3ff0_2658	HT0_WIN3_MASK	0 x3ff0_2758	HT1_WIN3_MASK
0 x3ff0_2660	HT0_WIN4_MASK	0 x3ff0_2760	HT1_WIN4_MASK
0 x3ff0_2668	HT0_WIN5_MASK	0 x3ff0_2768	HT1_WIN5_MASK
0 x3ff0_2670	HT0_WIN6_MASK	0 x3ff0_2770	HT1_WIN6_MASK
0 x3ff0_2678	HT0_WIN7_MASK	0 x3ff0_2778	HT1_WIN7_MASK
0 x3ff0_2680	HT0_WIN0_MMAP	0 x3ff0_2780	HT1_WIN0_MMAP
0 x3ff0_2688	HT0_WIN1_MMAP	0 x3ff0_2788	HT1_WIN1_MMAP
0 x3ff0_2690	HT0_WIN2_MMAP	0 x3ff0_2790	HT1_WIN2_MMAP
0 x3ff0_2698	HT0_WIN3_MMAP	0 x3ff0_2798	HT1_WIN3_MMAP
0 x3ff0_26a0	HT0_WIN4_MMAP	0 x3ff0_27a0	HT1_WIN4_MMAP
0 x3ff0_26a8	HT0_WIN5_MMAP	0 x3ff0_27a8	HT1_WIN5_MMAP
0 x3ff0_26b0	HT0_WIN6_MMAP	0 x3ff0_27b0	HT1_WIN6_MMAP
0 x3ff0_26b8	HT0_WIN7_MMAP	0 x3ff0_27b8	HT1_WIN7_MMAP

There are three ip-related address Spaces in the secondary XBAR of loongson 3, namely, the configuration register address space, the DDR2 address space, and the PCI address space. The address window is set for routing and address translation between the CPU and pci-dma IP with the main device function. Both the CPU and pci-dma have eight address Windows that enable the selection of the target address space and the conversion from the source address space to the target address space.

Each address window is composed of three 64-bit registers, BASE, MASK and MMAP. BASE is aligned with K bytes, and MASK USES a format similar to the network MASK with the high digit of 1. MMAP contains the transformed address, routing and enabling alleles, as shown in the following table:

48] [63:	[47:10]	[17]	[3-0]
Alternating selective bit	Converted address	The window can make	From the device number

Among them, the corresponding device from the device number is shown in the following table:

Table 2-7 at level 2 XBAR, the corresponding relationship between the device number and the module

From the device number	The default value
0	No. 0 ddr2/3 controller
1	No. 1 ddr2/3 controller
2	Low speed I/O (PCI, LPC, etc.)
3	Configuration register

The meaning of the window enable bit is shown in the following table:

Table 2-8 MMAP field corresponding to this spatial access property

[7]	[6]	[5]	[4]
The window can make	Allows interleaved access to the DDR, which is valid when the device number is 0, and routes requests to the address of the hit window according to the configuration of "interleaved selection bits". Requires a staggered enable bit More than 10	Allow the block read	Allowed to take to

It is important to note that the window configuration of level 1 XBAR cannot translate the address of the Cache consistent request, otherwise the address at SCache will be different from that at the processor level Cache, resulting in the maintenance error of Cache consistency.

Window hit formula :(IN_ADDR & MASK) == BASE

New address conversion formula :OUT_ADDR = (IN_ADDR & ~MASK) | {MMAP[63:10],10 'h0}

Address window conversion register is shown in the following table:

Table 2-9 the register table is converted from the level 2 XBAR address window

address	register	describe	The default value
3 ff0 0000	CPU_WIN0_BASE	Base address of CPU window 0	0 x0
3 ff0 0008	CPU_WIN1_BASE	Base address of CPU window 1	0 x1000_0000

3 ff0 0010	CPU_WIN2_BASE	Base address of CPU window 2	0 x0
3 ff0 0018	CPU_WIN3_BASE	Base address of CPU window 3	0 x0
3 ff0 0020	CPU_WIN4_BASE	Base address of CPU window 4	0 x0
3 ff0 0028	CPU_WIN5_BASE	Base address of CPU window 5	0 x0
3 ff0 0030	CPU_WIN6_BASE	Base address of CPU window 6	0 x0
3 ff0 0038	CPU_WIN7_BASE	The base address of CPU window 7	0 x0
3 ff0 0040	CPU_WIN0_MASK	The mask for CPU window 0	0 xffff_ffff_f000_0000
3 ff0 0048	CPU_WIN1_MASK	Mask for CPU window 1	0 xffff_ffff_f000_0000
3 ff0 0050	CPU_WIN2_MASK	CPU window 2 mask	0 x0
3 ff0 0058	CPU_WIN3_MASK	Mask for CPU window 3	0 x0
3 ff0 0060	CPU_WIN4_MASK	Mask for CPU window 4	0 x0
3 ff0 0068	CPU_WIN5_MASK	Mask for CPU window 5	0 x0
3 ff0 0070	CPU_WIN6_MASK	Mask for CPU window 6	0 x0
3 ff0 0078	CPU_WIN7_MASK	CPU window 7 mask	0 x0
3 ff0 0080	CPU_WIN0_MMAP	New base address for CPU window 0	0 xf0
3 ff0 0088	CPU_WIN1_MMAP	New base address for CPU window 1	0 x1000_00f2
3 ff0 0090	CPU_WIN2_MMAP	New base address for CPU window 2	0
3 ff0 0098	CPU_WIN3_MMAP	New base address for CPU window 3	0
3 ff0 00 a nought	CPU_WIN4_MMAP	New base address for CPU window 4	0 x0
3 ff0 00 a8	CPU_WIN5_MMAP	New base address for CPU window 5	0 x0
3 ff0 00 b0	CPU_WIN6_MMAP	New base address for CPU window 6	0
3 ff0 00 b8	CPU_WIN7_MMAP	New base address for CPU window 7	0
3 ff0 0100	PCI_WIN0_BASE	Base address for PCI window 0	0 x8000_0000
3 ff0 0108	PCI_WIN1_BASE	Base address for PCI window 1	0 x0

3 ff0 0110	PCI_WIN2_BASE	PCI window 2 base address	0 x0
3 ff0 0118	PCI_WIN3_BASE	Base address for PCI window 3	0 x0
3 ff0 0120	PCI_WIN4_BASE	Base address for PCI window 4	0 x0
3 ff0 0128	PCI_WIN5_BASE	Base address for PCI window 5	0 x0
3 ff0 0130	PCI_WIN6_BASE	Base address for PCI window 6	0 x0
3 ff0 0138	PCI_WIN7_BASE	Base address for PCI window 7	0 x0
3 ff0 0140	PCI_WIN0_MASK	Mask for PCI window 0	0 xffff_ffff_8000_0000
3 ff0 0148	PCI_WIN1_MASK	Mask for PCI window 1	0 x0
3 ff0 0150	PCI_WIN2_MASK	Mask for PCI window 2	0 x0
3 ff0 0158	PCI_WIN3_MASK	Mask for PCI window 3	0 x0
3 ff0 0160	PCI_WIN4_MASK	Mask for PCI window 4	0 x0
3 ff0 0168	PCI_WIN5_MASK	Mask for PCI window 5	0 x0
3 ff0 0170	PCI_WIN6_MASK	Mask for PCI window 6	0 x0
3 ff0 0178	PCI_WIN7_MASK	Mask for PCI window 7	0 x0
3 ff0 0180	PCI_WIN0_MMAP	New base address for PCI window 0	0 xf0
3 ff0 0188	PCI_WIN1_MMAP	New base address for PCI window 1	0 x0
3 ff0 0190	PCI_WIN2_MMAP	New base address for PCI window 2	0
3 ff0 0198	PCI_WIN3_MMAP	New base address for PCI window 3	0
3 ff0 a0 01	PCI_WIN4_MMAP	New base address for PCI window 4	0 x0
3 ff0 01 a8	PCI_WIN5_MMAP	New base address for PCI window 5	0 x0
3 ff0 b0 01	PCI_WIN6_MMAP	New base address for PCI window 6	0
3 ff0 b8 01	PCI_WIN7_MMAP	New base address for PCI window 7	0

According to the default register configuration, CPU 0x00000000-0x0fffffff address range after chip startup (256M) is mapped to the address range of 0x00000000-0x0fffffff of DDR2, 0x10000000-0x1fffffff of CPU (256M) is mapped to 0x10000000-0x1fffffff of PCI, and 0x80000000 of PCIDMA

The address interval of -0x8fffffff (256M) maps to the address interval of 0x00000000-0x0fffffff of DDR2. The software can realize the new address space routing and transformation by modifying the corresponding configuration registers.

In addition, when there is read access to illegal address caused by CPU guess execution, the 8 address Windows are not hit, and the configuration register module returns all 0 data to CPU to prevent CPU dead and so on.

Table 2-10 XBAR level 2 default address configuration

Base address	high	The owner of the
0 x0000_0000_0000_0000	0 x0000_0000_0fff_ffff	No. 0 DDR controller
0 x0000_0000_1000_0000	0 x0000_0000_1fff_ffff	Low speed I/O (PCI, etc.)

2.6 Chip configuration and sampling register

The chip configuration register (Chip_config) and the chip sampling register (Chip_sample) in the loong chip 3A2000 provide the mechanism for reading and writing the chip configuration.

Table 2-11 chip configuration register (physical address 0x1fe00180)

A domain	The field name	access	Reset value	describe
3-0	-	RW	4 'b7	reserve
4	MC0_disable_ddr2_confspace	RW	1 'b0	Whether to disable the MC0 DDR configuration space
5	-	RW	1 'b0	reserve
6	-	RW	1 'b0	reserve
7	MC0_ddr2_resetn	RW	1 'b1	MC0 software reset (low efficiency)
8	MC0_clken	RW	1 'b1	Whether to enable MC0
9	MC1_disable_ddr2_confspace	RW	1 'b0	Whether to disable the MC1 DDR configuration space
10	-	RW	1 'b0	reserve
11	-	RW	1 'b0	reserve
12	MC1_ddr2_resetn	RW	1 'b1	MC1 software reset (low efficiency)
13	MC1_clken	RW	1 'b1	Whether to enable MC1
26:24	HT0_freq_scale_ctrl	RW	3 'b111	HT controller 0 frequency division
27	HT0_clken	RW	1 'b1	Whether to enable HT0
30:28	HT1_freq_scale_ctrl	RW	3 'b111	HT controller 1 frequency division
31	HT1_clken	RW	1 'b1	Whether to enable HT1
42:40	Node0_freq_ CTRL	RW	3 'b111	Node 0 frequency division
43	-	RW	1 'b1	
46:44	Node1_freq_ CTRL	RW	3 'b111	Node 1 frequency division

47	-	RW	1 'b1	
63:56	Cpu_version	R	2 'h37	The CPU version
95:64				(empty)
127:96	Pad1v8_ctrl	RW	6 'h780	1 v8 control pad
other		R		reserve

Table 2-12 chip sampling register (physical address 0x1fe00190)

A domain	The field name	access	Reset value	describe
31:0	Compcode_core	R		
47:32	Sys_clkseli	R		On board frequency doubling setting
55:48	Bad_ip_core	R		Core7 - core0 is bad
57:56	Bad_ip_ddr	R		Whether 2 DDR controllers are bad
61:60	Bad_ip_ht	R		Is the 2 HT controllers bad
83:80	Compcode_ok	R		
88	Thsens0_overflow	R		Overflow of temperature sensor 0 (over 125℃)
89	Thsens1_overflow	R		Overflow of temperature sensor 1 (over 125℃)
103:96	Thsens0_out	R		Temperature sensor: 0 ℃ Node temperature = thens0_out-100 temperature range -40 ° -125 °
111:104	Thsens1_out	R		Temperature sensor 1 ℃ temperature Node temperature = thens1_out-100 temperature range -40 ° -125 °
other		R		reserve

The following sets of software frequency doubling Settings registers are used to set the working frequencies of each clock when CLKSEL is configured in software control mode (refer to the CLKSEL setting method in section 2.2). Among them, the MEM CLOCK configuration corresponds to the memory controller and the bus CLOCK frequency;The CORE CLOCK corresponds to the CLOCK frequency of processor CORE, on-chip network and high-speed Shared cache.HT CLOCK corresponds to HT controller CLOCK frequency.

Each clock configuration typically has two parameters, DIV_LOOPC and DIV_OUT. The final clock frequency is (reference clock * DIV_LOOPC)/DIV_OUT.

The configuration method of HT CLOCK is special, please refer to the specific configuration method in section 10.5.28.

In the software control mode, the default corresponding CLOCK frequency is the frequency of the external reference CLOCK (for CORE CLOCK, is the corresponding frequency of pin SYS_CLK; For MEM CLOCK, is the corresponding frequency of pin MEM_CLK), you need to set the software for the CLOCK during processor startup. The process of setting each clock should be as follows:

1) Other registers in the setting register except SEL_PLL_* and SOFT_SET_PLL, that is, these two registers are written as 0 in the setting process;
2) Set SOFT_SET_PLL to 1 with other register values unchanged;
3) The lock signal LOCKED_* in the wait register is 1;
4) Set SEL_PLL_* to 1, and the corresponding clock frequency will be switched to the frequency set by the software.

Table 2-13 chip node and processor core software frequency doubling setting register (physical address 0x1fe00180)

A domain	The field name	access	Reset value	describe
0	SEL_PLL_NODE	RW	0 x0	Bypass the entire Node clock PLL
1	-	RW	0 x0	-
2	SOFT_SET_PLL	RW	0 x0	Allows software to set up PLL
3	BYPASS_L1	RW	0 x0	Bypass L1 PLL
6:4	-	RW	0 x0	-
7	LOCKEN_L1	RW	0 x0	Allows you to lock L1 PLL
9:8	-	RW	0 x0	-
11:10	LOCKC_L1	RW	0 x0	Determines whether L1 PLL locks the phase used The accuracy of the
15:12	-	RW	0 x0	-
16	LOCKED_L1	R	0 x0	Whether L1 PLL is locked
18:17	-	R	0 x0	-

19	PD_L1	R/W	0 x0	Close the L1 PLL
31:20	-	RW	0 x1	-
38:32	L1_DIV_LOOPC	RW	0 x1	L1 PLL input parameter
41:39	-			-
47:42	L1_DIV_OUT	RW	0 x1	L1 PLL input parameter
other	-	RW		reserve

Note: PLL ouput = (clk_ref * div_loopc)/div_out.

The PLL VCO frequency (in brackets above) must be in the range of 1.2ghz to 3.2ghz. This requirement applies equally to MEM PLL and HT PLL.

Table 2-14 chip memory and HT clock software frequency doubling setting register (physical address 0x1fe001c0)

A domain	The field name	acces s	Reset value	desc ribe
0	SEL _MEM_PLL	RW	0 x0	MEM clock non-software bypass entire PLL
1	SOFT_SET_MEM_PLL	RW	0 x0	Allows software to set up MEM PLL
2	BYPASS_MEM_PLL	RW	0 x0	Bypass MEM_PLL
3	LOCKEN_MEM_PLL	RW	0 x0	Allows you to lock MEM_PLL
when	LOCKC_ MEM_PLL	RW	0 x0	Determines whether MEM PLL is locked to the accuracy of the phase used
6	LOCKED_MEM_PLL	R	0 x0	Whether MEM_PLL is locked or not
7	PD_MEM_PLL	RW	0 x0	Close the MEM PLL
Will you	-	RW	0 x1	-
brake	MEM_PLL_DIV_LOOPC	RW	0 x41	MEM PLL input parameters
A partner	MEM_PLL_DIV_OUT	RW	0 x0	MEM PLL input parameters
32	SEL_HT0_PLL	RW	0 x0	HT0 bypass PLL
33	SOFT_SET_HT0_PLL	RW	0 x0	Allows software to set HT0 PLL
34	BYPASS_HT0_PLL	RW	0 x0	Bypass HT0_PLL
35	LOCKEN_HT0_PLL	RW	0 x0	Allows locking HT0 PLL
meanw hile	LOCKC_HT0_PLL	RW	0 x0	Determine whether the HT0 PLL is locked Precision of phase
38	LOCKED_HT0_PLL	R	0 x0	Whether HT0_PLL is locked or not
45:40: 15	HT0_DIV_HTCORE	RW	0 x1	HT0 Core PLL input parameter
48	SEL_HT1_PLL	RW	0 x0	HT1 bypass PLL
49	SOFT_SET_HT1_PLL	RW	0 x0	Allows software to set HT1 PLL

50	BYPASS_HT1_PLL	RW	0 x0	Bypass HT1_PLL
51	LOCKEN_HT1_PLL	RW	0 x0	Allows locking of HT1 PLL
53:52	LOCKC_HT1_PLL	RW	0 x0	Determine whether the HT1 PLL is locked using the phase accuracy
54	LOCKED_HT1_PLL	R	0 x0	Whether HT1_PLL is locked or not
61:56	HT1_DIV_HTCORE	RW	0 x1	HT1 Core PLL input parameter
other		RW		reserve

Table 2-15 chip processor core software frequency division setting register (physical address 0x1fe001d0)

A domain	The field name	access	Reset value	describe
The 2-0	core0_freqctrl	RW	0 x7	Kernel 0 frequency division control value
3	core0_en	RW	0 x1	Nuclear 0 clock enabled
6:4	core1_freqctrl	RW	0 x7	Kernel 1 frequency division control value
7	core1_en	RW	0 x1	Nuclear 1 clock enabled
10:8	core2_freqctrl	RW	0 x7	Kernel 2 frequency division control value
11	core2_en	RW	0 x1	Nuclear 2 clock enabled
then	core3_freqctrl	RW	0 x7	Kernel 3 frequency division control value
15	core3_en	RW	0 x1	Nuclear 3 clock enabled
			Note:	The clock frequency value after software frequency division is equal to the original (frequency division control value +1) /8

GS464e processor core

The GS464e is a four-emission 64-bit high-performance processor core. The processor core can be used as a single core for high-end embedded applications and desktop applications, and can also be used as a basic processor core for in-chip multi-core systems for server and high-performance machine applications. Multiple GS464 cores in loongson 3A2000 form a multicore structure of a last-level Cache on a distributed Shared chip through a Shared Cache module and a AXI interconnection network. The main features of GS464 are as follows:

- MIPS64 compatible, support loongson extended instruction set;
- Four emission superscalar structure, two fixed points, two floating point, two memory access parts;

- Each floating point component supports full-flow 64-bit/double-32-bit floating point multiplication and addition;
- Memory access parts support 128 bit storage access, virtual address is 64 bit, physical address is 48 bit;
- Support register renaming, dynamic scheduling, transfer prediction and other out-of-order execution techniques;
- 64 items are fully linked, and 1024 items are connected to 8-channel group, a total of 1088 TLB items, 64 instructions TLB, variable page size;
- The size of the first-level instruction Cache and the data Cache is 64KB respectively, and the four-way groups are connected.
- Victim Cache is a private second-level Cache with a size of 256KB and a 16-way connection.
- Support the optimization technology of non-blocking visit and load-speculation;
- Support Cache consistency protocol, which can be used in chip multi-core processors;
- The instruction Cache implements parity check, and the data Cache implements ECC check.
- Standard EJTAG debugging standard is supported to facilitate debugging of software and hardware.
- Standard 128 - bit AXI interface.

The structure of GS464e is shown in the following figure. For more details on GS464e, please refer to the GS464e user manual as well

MIPS64 user's manual.

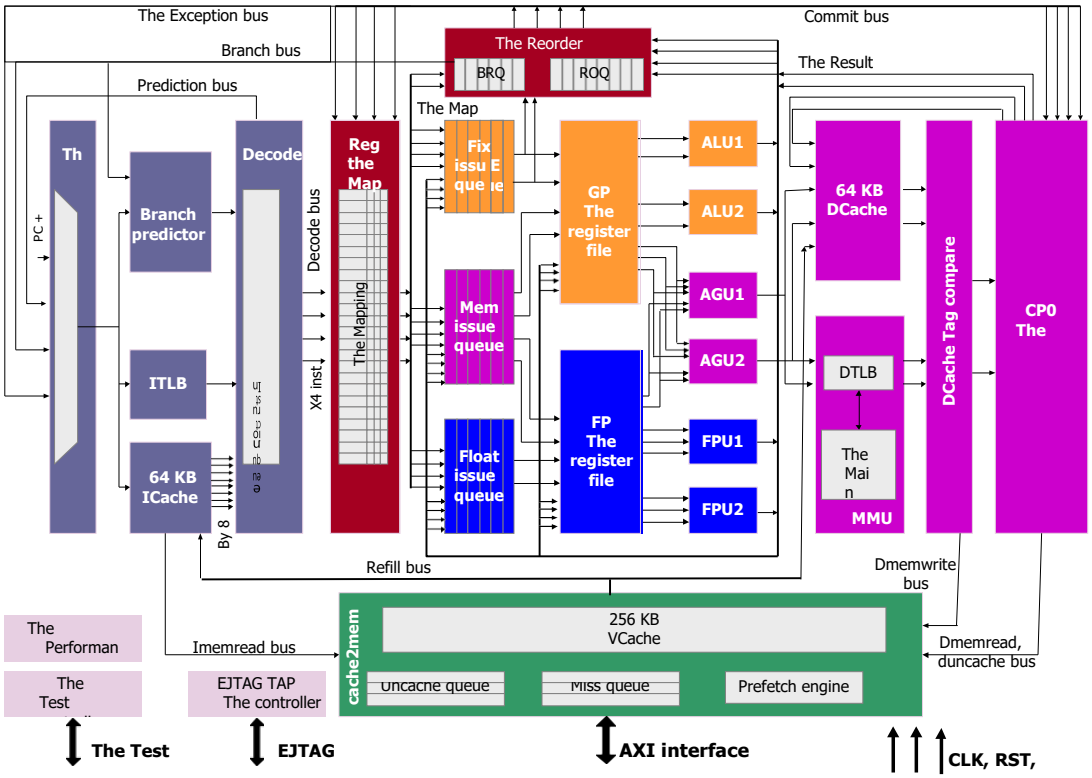


Figure 3-1 GS464e structure diagram

3 Shared Cache (SCache)

The SCache module is a three-level Cache Shared by all processor cores in the loongson 3A2000 processor.The main features of the SCache module include:

- Use 128 - bit AXI interface.
- 16 Cache items access the queue.
- Keyword first.
- The fastest time to receive a read failure request is 12 beats.
- Cache consistency protocol is supported through directories.
- It can be used for multi-core structure on chip, and also can be directly connected with IP of single processor.
- Adopt 16 - way group - linked structure.
- Support ECC validation.
- Support DMA consistent read-write and prefetch reads.
- Support 16 Shared Cache hashes.
- Support for sharing Cache by window lock.
- Ensure that the read data returns atomicity.

Shared Cache module includes Shared Cache management module scachemanage and Shared Cache access module scacheaccess. The Scachemanage module is responsible for the processor's access requests from the processor and DMA, while information such as tags, directories, and data that share the Cache is stored in the scacheaccess module. To reduce power consumption, tags, directories, and DATA in a Shared Cache can be accessed separately. The Shared Cache status bit and w bit are stored with tags, which are stored in TAG RAM, DIR RAM, and DATA RAM. Invalid requests to access the Shared Cache, read all the tags and directories, select the directory based on the TAG, and read the data based on the hit. Replace requests, refill requests, and write back requests operate only on tags, directories, and data along the way.

To improve the performance of certain computing tasks, a locking mechanism has been added to the Shared Cache. A Shared Cache block in a locked area is locked and will not be replaced from the Shared Cache (unless the 16-way Shared Cache is full of locked blocks). The four groups of lock window registers inside the Shared Cache module can be dynamically configured through the chip configuration register space, but one of the 16 Shared caches must not be locked. The size of each group of Windows can be adjusted according to the mask, but not more than 3/4 of the overall Shared Cache size. In addition, when the Shared Cache receives a DMA write request, if the region being written is hit in the Shared Cache and locked, then the DMA write will be written directly to the Shared Cache instead of to the Shared Cach Memory.

Table 4-1 Shared Cache lock window register configuration

The name of the	address	A domain	describe
Slock0_valid	0 x3ff00200	[63-63]	Zero lock window valid bit
Slock0_addr	0 x3ff00200	[47:0]	Lock window 0 lock address
Slock0_mask	0 x3ff00240	[47:0]	Lock window mask no. 0
Slock1_valid	0 x3ff00208	[63-63]	No. 1 lock window valid bit
Slock1_addr	0 x3ff00208	[47:0]	No. 1 lock window lock address
Slock1_mask	0 x3ff00248	[47:0]	No. 1 lock window mask
Slock2_valid	0 x3ff00210	[63-63]	No. 2 lock window valid bit
Slock2_addr	0 x3ff00210	[47:0]	No. 2 lock window lock address
Slock2_mask	0 x3ff00250	[47:0]	No. 2 lock window mask
Slock3_valid	0 x3ff00218	[63-63]	No. 3 lock window valid bit
Slock3_addr	0 x3ff00218	[47:0]	No. 3 lock window lock address
Slock3_mask	0 x3ff00258	[47:0]	No. 3 lock window mask

For example, when an address addr makes slock0_valid && ((addr & slock0_mask) == (slock0_addr & slock0_mask)) equal to 1, the address is locked by lock window 0.

4 Matrix processing accelerator

Logodson 3A2000 is built with two matrix processing accelerators independent of the processor core. Its basic function is to realize the function of transposing or moving the matrix stored in memory from the source matrix to the target matrix through the configuration of software. The two accelerators are respectively integrated in the two HyperTransport controllers of loongson 3A2000, and the read-write of SCache and memory can be realized through a cross switch.

Before due to transpose the same Cache line element order after the transposed matrix is distributed, in order to improve the efficiency of reading and writing, need read many rows of data, makes the data can be in after the transposed matrix unit to write to the Cache behavior, thus set up a size 32 in the module the buffer zone, realize transverse writing (matrix into the buffer from the source), longitudinal read (matrix) by the buffer is written to the target.

The working process of matrix processing is to read in the 32 rows of source matrix data, and then write the 32 rows of data to the target matrix, and so on, until the entire matrix is transposed or moved. The matrix processing accelerator can also prefetch the source matrix without writing the target matrix as required, so as to achieve the operation of prefetching the data to SCache.

The source matrix involved in transpose or shift may be a small matrix located in a large matrix, so its matrix address may not be completely contiguous, and the addresses between adjacent rows will be spaced, requiring more programming control interfaces to be implemented. The following tables 5-1

through 5-4 illustrate the programming interfaces involved in matrix processing.

Table 5-1 matrix processing programming interface description

addr ess	The name of the	attri bute	inst ruct ions
0 x3ff00600	src_start_addr	RW	The starting address of the source matrix
0 x3ff00608	dst_start_addr	RW	The starting address of the destination matrix
0 x3ff00610	The row	RW	The number of elements in a row in the source matrix
0 x3ff00618	Col.	RW	The number of elements in a column of the source matrix
0 x3ff00620	length	RW	The row span (bytes) of the large matrix in which the source matrix is located
0 x3ff00628	width	RW	The row span (bytes) of the large matrix in which the target matrix is located
0 x3ff00630	trans_ctrl	RW	Transpose the control register
0 x3ff00638	trans_status	RO	Transpose the status register

Table 5-2 matrix processing register address description

addr ess	The name of the
0 x3ff00600	Src_start_addr of the zero transpose module
0 x3ff00608	Dst_start_addr of the zero transpose module
0 x3ff00610	Row of the zero transpose module
0 x3ff00618	Col of the 0 transposed module
0 x3ff00620	Length of the zero transpose module
0 x3ff00628	Width of the zero transpose module
0 x3ff00630	Trans_ctrl for the zero transpose module
0 x3ff00638	Trans_status of the zero transpose module
0 x3ff00700	Src_start_addr of the # 1 transpose module
0 x3ff00708	Dst_start_addr of the # 1 transpose module
0 x3ff00710	Src_row_stride of no. 1 transpose module
0 x3ff00718	Src_last_row_addr of the # 1 transpose module
0 x3ff00720	Length of the number 1 transpose module
0 x3ff00728	Width of no. 1 transpose module

0 x3ff00730	Trans_ctrl for the # 1 transpose module
0 x3ff00738	Trans_status for the # 1 transpose module

Table 5-3 trans_ctrl register description

field	instructions
0	Can make a
1	Whether writing the target matrix is allowed. When the value is 0, the transpose process only prefetches the source matrix, but does not write the target matrix.
2	After the source matrix is read, whether it is valid or not is interrupted.
3	After the completion of writing the target matrix, whether it is valid or not,
7.. 4	Arcmd, read command internal control bit. When arcache is 4 'hf, it must be set to 4' hc. When arcache is another value, is meaningless.
11.. 8	Arcache, read command internal control bit. The cache path is used when 4 'hf, and the uncache path is used when 4' h0. other The value is meaningless.
15..12	Awcmd, write command internal control bit. When awcache is 4 'hf, it must be set to 4' hb. It is meaningless when awcache is other values.
19..16	Awcache, write command internal control bit. The cache path is used when 4 'hf, and the uncache path is used when 4' h0. other
	The value is meaningless.
21..20	The element size of the matrix, 00 for 1 byte, 01 for 2 bytes, 10 for 4 bytes, and 11 for 8 bytes
22	Trans_yes, 1 means transpose; Zero means no transpose

Table 5-4 trans_status register description

field	instructions
0	The source matrix is read
1	The target matrix is written

5 Interrupt and communication between processor cores

Loongson 3A2000 implements 8 intercore interrupt registers (IPI) for each processor core to support interrupt and communication between processor cores during multi-core BIOS startup and operating system runtime. The instructions and addresses are shown in tables 6-1 through 6-5.

Table 6-1 registers associated with interrupts between processor cores and

their functional descriptions

The name of the	Read and write access	describe
IPI_Status	R	The 32-bit status register, where any bit is set to 1 and the corresponding bit is enabled, is set to the processor core INT4 interrupts.
IPI_Enable	RW	The 32-bit enable register controls whether the corresponding interrupt bit is valid
IPI_Set	W.	32 position bit register, write 1 to the corresponding bit, the corresponding STATUS register bit is set 1
IPI_Clear	W.	32 bit clear register, write 1 to the corresponding bit, the corresponding STATUS register bit is clear 0
MailBox0	RW	Cache register, used to pass parameters at startup, 64 or 32 bits Uncache method for access.
MailBox01	RW	Cache register, used to pass parameters at startup, 64 or 32 bits Uncache method for access.
MailBox02	RW	Cache register, used to pass parameters at startup, 64 or 32 bits Uncache method for access.
MailBox03	RW	Cache register, used to pass parameters at startup, 64 or 32 bits Uncache method for access.

The registers related to intercore interrupts in loongson 3A2000

and their functions are described as follows:

The name of the	address	permissions	describe
Core0_IPI_Status	communication registers list	R	The IPI_Status register for the number 0 processor core
Core0_IPI_Enalbe	0 x3ff01004	RW	The IPI_Enalbe register of the number 0 processor core
Core0_IPI_Set	0 x3ff01008	W.	The IPI_Set register of the number 0 processor core
Core0 _IPI_Clear	0 x3ff0100c	W.	The IPI_Clear register for the number 0 processor core
Core0_MailBox0	0 x3ff01020	RW	Register IPI_MailBox0 of the number 0 processor

Table 6-3 intercore interrupt and communication registers list for processor core no. 1

The name of the	address	permissions	describe
Core1_IPI_Status	0 x3ff01100	R	The IPI_Status register for the number 1 processor core
Core1_IPI_Enalbe	0 x3ff01104	RW	The IPI_Enalbe register of the number 1 processor core
Core1_IPI_Set	0 x3ff01108	W.	The IPI_Set register of the number 1 processor core
Core1 _IPI_Clear	0 x3ff0110c	W.	The IPI_Clear register of the number 1 processor core
Core1_MailBox0	0 x3ff01120	R	Register IPI_MailBox0 for the number 1 processor core
Core1_ MailBox1	0 x3ff01128	RW	Register IPI_MailBox1 of the number 1 processor core
Core1_ MailBox2	0 x3ff01130	W.	The IPI_MailBox2 register of the number 1 processor core
Core1_ MailBox3	0 x3ff01138	W.	Register IPI_MailBox3 for the number 1 processor core

Table 6-4 intercore interrupt and communication registers list for no. 2 processor core

The name of the	address	permissions	describe
Core2_IPI_Status	0 x3ff01200	R	The IPI_Status register of the number 2 processor core
Core2_IPI_Enalbe	0 x3ff01204	RW	The IPI_Enalbe register of the number 2 processor core
Core2_IPI_Set	0 x3ff01208	W.	The IPI_Set register of the number 2 processor core
Core2 _IPI_Clear	0 x3ff0120c	W.	The IPI_Clear register of the number 2 processor core
Core2_MailBox0	0 x3ff01220	R	Register IPI_MailBox0 of the number 2 processor core
Core2_ MailBox1	0 x3ff01228	RW	Register IPI_MailBox1 in the number 2 processor core
Core2_ MailBox2	0 x3ff01230	W.	The IPI_MailBox2 register of the number 2 processor core
Core2_ MailBox3	0 x3ff01238	W.	The IPI_MailBox3 register of the number 2 processor core

Table 6-5 intercore interrupt and communication registers list for no. 3 processor core

The name of the	address	permissions	describe
Core3_IPI_Status	0 x3ff01300	R	The IPI_Status register of the number 3 processor core

Core3_IPI_Enalbe	0 x3ff01304	RW	The IPI_Enalbe register of the number 3 processor core
Core3_IPI_Set	0 x3ff01308	W.	The IPI_Set register of the number 3 processor core
Core3 _IPI_Clear	0 x3ff0130c	W.	The IPI_Clear register of the number 3 processor core
Core3_MailBox0	0 x3ff01320	R	Register IPI_MailBox0 for the number 3 processor core
Core3_ MailBox1	0 x3ff01328	RW	Register IPI_MailBox1 for the number 3 processor core
Core3_ MailBox2	0 x3ff01330	W.	The IPI_MailBox2 register of the number 3 processor core
Core3_ MailBox3	0 x3ff01338	W.	The IPI_MailBox3 register of the number 3 processor core

The above is a list of intercore interrupt related registers for a single-node multiprocessor system consisting of a single loongson 3A2000 chip. When a multi-node cc-numa system is constructed by multi-chip godson 3A2000 interconnection, each node in the chip corresponds to a system global node number, and the IPI register address of the processor core in the node is according to the above table and the base of its node

Address into a fixed offset relationship. For example, the IPI_Status address of node 0 processor core 0 is 0x3ff01000 and 1

The address of processor no. 0 of node no. 1 is 0x10003ff01000, and so on.

6 I/O interrupt

As shown in figure 7-1 below, any IO interrupt source can be configured to enable or disable, trigger mode, and target processor core interrupt pin to be outed.

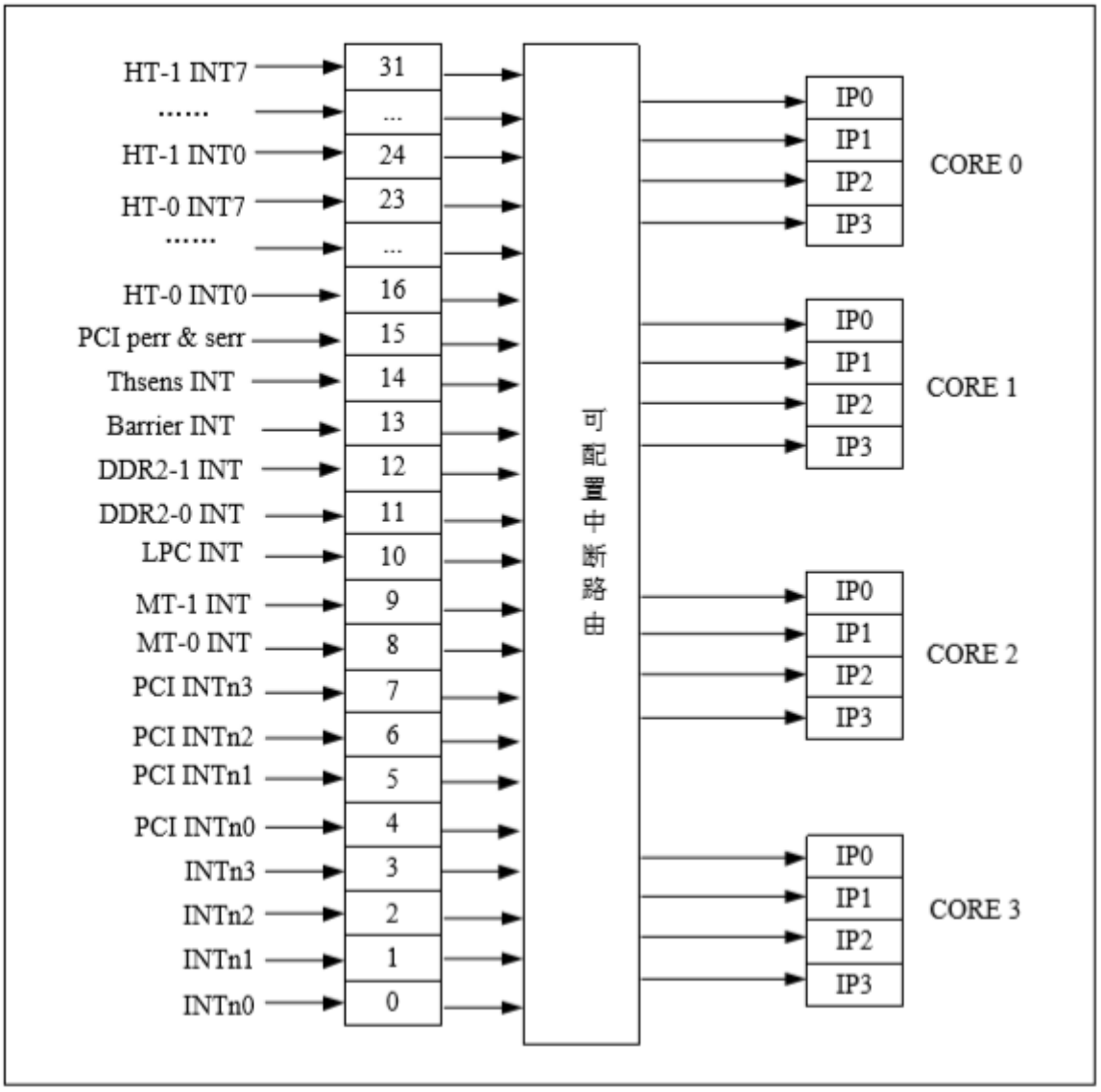


Figure 7-1 loongson 3A2000 processor interrupt routing diagram

The interruption-related configuration registers control the corresponding interrupts in the form of bits. See table 7-1 below for the connection and property configuration of interrupt control bits.The interrupt Enable configuration has three registers: Intenset, Intenclr, and Inten. The Intenset sets the interrupt enable, and the interrupt corresponding to the bit write 1 in the Intenset register is enabled. The Intenclr clears interrupts to enable, and the interrupt corresponding to the Intenclr register write 1 is cleared. The Inten register reads the current status of each interrupt enable.Interrupt signals in the form of pulses (such as PCI_SERR) are selected by the Intedge configuration register, with write 1 for pulse trigger and write 0 for level trigger. The interrupt handler can clear the pulse record by the corresponding bit of the Intenclr.

Table 7-1 interrupt control registers

A domain	Access properties/default values				
	Intedge	Inten	Intenset	Intenclr	The interrupt source
3:0	RW / 0	R / 0	W / 0	W / 0	Sys_int0-3
7:4	RO / 0	R / 0	RW / 0	RW / 0	PCI_INTn
8	RO / 0	R / 0	RW / 0	RW / 0	Matrix_int0
9	RO / 1	R / 0	RW / 0	RW / 0	Matrix_int1
10	RO / 1	R / 0	RW / 0	RW / 0	Lpc
12:11	RW / 0	reserve	reserve	reserve	Mc0-1
13	RW / 0	R / 0	RW / 0	RW / 0	The Barrier
14	RW / 0	R / 0	RW / 0	RW / 0	Thsens int
15	RW / 0	R / 0	RW / 0	RW / 0	Pci_perr
23:16	RW / 0	R / 0	RW / 0	RW / 0	HT0 int0-7
31:24	RW / 0	R / 0	RW / 0	RW / 0	HT1 int0-7

Table 7-2 IO control register addresses

The name of the	Address offset	describe
Intisr	0 x3ff01420	32-bit interrupt status register
Inten	0 x3ff01424	32-bit interrupt enabled status register
Intenset	0 x3ff01428	The 32-bit setting enables the register
Intenclr	0 x3ff0142c	32-bit clear enable register
Intedge	0 x3ff01438	32-bit trigger mode register
CORE0_INTISR	0 x3ff01440	Routing to the 32-bit interrupt state of CORE0
CORE1_INTISR	0 x3ff01448	Routing a 32-bit interrupt state to CORE1
CORE2_INTISR	0 x3ff01450	Routing a 32-bit interrupt state to CORE2
CORE3_INTISR	0 x3ff01458	Routing a 32-bit interrupt state to CORE3

With four processor cores integrated into the loson 3A2000, the 32-bit interrupt source can be configured to select the target processor core to interrupt. Further, the interrupt source can choose to route to any of the processor core interrupts INT0 through INT3, IP2 through IP5 corresponding to CP0_Status.Each of the 32 I/O interrupt sources corresponds to an 8-bit routing controller, whose format and address are shown in tables 7-3 and 7-4 below. The routing register USES a vector approach for

routing, such as 0x48 to indicate routing to INT2 of processor no. 3.

Table 7-3 description of interrupt routing registers

A domain	Said Ming
3-0	Routing the processor core vector number
The log	Routing the processor core interrupt pin vector number

Table 7-4 interrupt routing register addresses

The name of the	Address offset	describe	The name of the	Address offset	describe
Entry0	0 x3ff01400	Sys_int0	Entry16	0 x3ff01410	HT0 - int0
Entry1	0 x3ff01401	Sys_int1	Entry17	0 x3ff01411	HT0 - int1
Entry2	0 x3ff01402	Sys_int2	Entry18	0 x3ff01412	HT0 - int2
Entry3	0 x3ff01403	Sys_int3	Entry19	0 x3ff01413	HT0 - int3
Entry4	0 x3ff01404	Pci_int0	Entry20	0 x3ff01414	HT0 - int4
Entry5	0 x3ff01405	Pci_int1	Entry21	0 x3ff01415	HT0 - int5
Entry6	0 x3ff01406	Pci_int2	Entry22	0 x3ff01416	HT0 - int6
Entry7	0 x3ff01407	Pci_int3	Entry23	0 x3ff01417	HT0 - int7
Entry8	0 x3ff01408	Matrix int0	Entry24	0 x3ff01418	HT1 - int0
Entry9	0 x3ff01409	Matrix int1	Entry25	0 x3ff01419	HT1 - int1
Entry10	0 x3ff0140a	Lpc int	Entry26	0 x3ff0141a	HT1 - int2
Entry11	0 x3ff0140b	Mc0	Entry27	0 x3ff0141b	HT1 - int3
Entry12	0 x3ff0140c	Mc1	Entry28	0 x3ff0141c	HT1 - int4
Entry13	0 x3ff0140d	The Barrier	Entry29	0 x3ff0141d	HT1 - int5
Entry14	0 x3ff0140e	Thsens int	Entry30	0 x3ff0141e	HT1 - int6
Entry15	0 x3ff0140f	Pci_perr/serr	Entry31	0 x3ff0141f	HT1 - int7

7 Temperature sensor

7.1 Real-time temperature sampling

The godson 3A2000 is internally integrated with two temperature sensors, which can be observed through the sampling register starting from 0x1FE00198. At the same time, it can be controlled by flexible high-low temperature interrupt alarm or automatic FM function. The corresponding bit of the temperature sensor in the sampling register is as follows (base address 0x1FE00198) :

Table 8-1 temperature sampling register description

A domain	The field name	access	Reset value	describe
24	Thsens0_overflow	R		Overflow of temperature sensor 0 (over 125℃)
25	Thsens1_overflow	R		Overflow of temperature sensor 1 (over 125℃)
39:32	Thsens0_out	R		Temperature sensor: 0 ℃ Node temperature = thens0_out-100 temperature range -40 ° -125 °
47:40	Thsens1_out	R		Temperature sensor 1 ℃ temperature Node temperature = thens1_out-100 temperature range -40 ° -125 °
other		R		reserve

By setting the control register, it can realize the functions of interrupting above the preset temperature, interrupting below the preset temperature and automatically lowering the frequency at high temperature.

7.2 High and low temperature interrupt triggered

For the high and low temperature interrupt alarm function, there are four groups of control registers to set the threshold value. Each set of registers contains the following three control bits:

GATE: sets the threshold of high or low temperature. When the input temperature is higher than the high temperature threshold or lower than the low temperature threshold, an interrupt will occur.

EN: interrupt enable control. The set of registers is only valid after setting 1;

SEL: input temperature selection. The register is used to configure which sensor's temperature is selected as the input. You can use either a 0 or a 1.

The high temperature interrupt control register contains four sets of Settings for controlling the high temperature interrupt trigger. Low temperature interrupt control register

It contains four sets of Settings to control the cold interrupt triggering. There is also a set of registers to display the interrupt status, corresponding to high temperature interrupts and low temperature interrupts, and any write to this register will clear the interrupt status.

The specific description of these registers is as follows:

Table 8-2 high and low temperature interrupt registers

register	address	control	instructions
High temperature interrupt control register Thsens_int_ctrl_Hi	0 x3ff01460	RW	[7:0] : Hi_gate0: high temperature threshold 0 above which an interrupt will occur [8:8] : Hi_en0: high temperature interrupt enable 0 [11:10] : Hi_Sel0: select high temperature interrupt 0 as the temperature sensor input source [23:16] : Hi_gate1: high temperature threshold 1 above which interruption will occur [24:24] : Hi_en1: high temperature interrupt enable 1 [27:26] : Hi_Sel1: select high temperature interrupt 1 for the temperature sensor input source [39:32] : Hi_gate2: high temperature threshold 2, beyond which there will be an interrupt [40:40] : Hi_en2: high temperature interrupt enable 2 [43:42] : Hi_Sel2: select high temperature interrupt 2 as the temperature sensor input source [55:48] : Hi_gate3: high temperature threshold 3, beyond which there will be an interrupt [56:56] : Hi_en3: high temperature interrupt enable 3 [59:58] : Hi_Sel3: select the temperature sensor input source of high temperature interrupt 3
Low temperature interrupt control register Thsens_int_ctrl_Lo	0 x3ff01468	RW	[7:0] : Lo_gate0: low temperature threshold 0 below which there will be an interrupt [8:8] : Lo_en0: low temperature interrupt enable 0 [11:10] : Lo_Sel0: select the temperature sensor input source of cryogenic interrupt 0 [23:16] : Lo_gate1: cryogenic threshold 1 below which there will be an interrupt [24:24] : Lo_en1: cryogenic interrupt enable 1 [27:26] : Lo_Sel1: select the temperature sensor input source of cryogenic interrupt 1 [39:32] : Lo_gate2: cryogenic threshold 2, below which there will be an interrupt [40:40] : Lo_en2: cryogenic interrupt enable 2 [43:42] : Lo_Sel2: select the temperature sensor input source for cold interrupt 2 [55:48] : Lo_gate3: cold threshold 3, below which an interrupt [56:56] will occur: Lo_en3: cold interrupt

			enable 3 [59:58] : Lo_Sel3: select the temperature sensor input source of cryogenic interrupt 3
Interrupt status register Thsens_int_status/CLR	0 x3ff01470	RW	Interrupt status register, write any value to clear the interrupt [0] : high temperature interrupt trigger [1] : low temperature interrupt trigger

7.3 High temperature automatic frequency down setting

In order to ensure the operation of the chip in high temperature environment, the chip can be set to automatically lower the frequency at high temperature, so that the chip can actively carry out clock frequency division when it exceeds the preset range, so as to reduce the chip turnover rate.

For the high temperature frequency down function, there are four sets of control registers to set its behavior. Each set of registers contains the following four control bits:

GATE: sets the threshold of high or low temperature. It is triggered when the input temperature is above or below the high temperature threshold

Frequency division operation;

EN: interrupt enable control. The set of registers is only valid after setting 1;

SEL: input temperature selection. The register is used to configure which sensor's temperature is selected as the input. You can use either a 0 or a 1.

FREQ: frequency division. When triggering a frequency division operation, adjust the frequency to FREQ/8 times the current clock

register	address	control	instructions
High temperature frequency down control register Thsens_freq_scale	0 x3ff01480	RW	Four groups set priority from high to low [7:0] : Scale_gate0: high temperature threshold 0, above which the frequency will be reduced [8:8] : Scale_en0: high temperature frequency reduction enables 0 Scale_Sel0: select the temperature sensor input source of high temperature frequency reduction 0 [14:12] : Scale_freq0: frequency separation value when frequency reduction [23:16] : Scale_gate1: high temperature threshold 1, beyond which the frequency will be reduced [24:24] : Scale_en1: high temperature frequency reduction enable 1 Scale_Sel1: select the temperature sensor input source of high temperature frequency reduction 1 [30:28] : Scale_freq1: frequency separation value when frequency reduction [39:32] : Scale_gate2: high temperature threshold 2, above which the frequency will be reduced [40:40] : Scale_en2: high temperature frequency reduction enable 2 Scale_Sel2: select the temperature sensor input source of high temperature frequency reduction 2 [46:44] : Scale_freq2: frequency separation value when frequency reduction [55:48] :

8 Ddr2/3 SDRAM controller configuration

The integrated memory controller inside the loongson 3 processor is designed to comply with the ddr2/3 SDRAM industry standard (jesd79-2 and jesd79-3). In the loongson 3 processor, all memory read/write operations are implemented in accordance with jesd79-2b and jesd79-3.

8.1 Ddr2/3 SDRAM controller features overview

The loongson 3 processor supports up to 4 CS (implemented by 4 DDR2 SDRAM chip selectors, namely two double-sided memory strips) and contains a total of 19 bit address bus (i.e., 16 bit line and column address bus and 3 bit logical Bank bus).

Loongson 3 processor can adjust the parameter setting of ddr2/3 controller to support the use of different memory chip types. Among them, the maximum number of supported block selections (CS_n) is 4, the number of row addresses (RAS_n) is 16, and the number of column addresses

The number (CAS_n) is 15, and the logical body choice (BANK_n) is 3.

The physical address of the memory request sent by the CPU can be mapped to a variety of different addresses according to different configurations within the controller.

The memory control circuit integrated by the loongson 3 processor only accepts memory read/write requests from the processor or external devices. In all memory read/write operations, the memory control circuit is in Slave State.

The memory controller in loongson 3 processor has the following characteristics:

- Interface command, read and write data full flow operation
- Memory command merge, sort to improve the overall bandwidth
- Configure the register read-write port to modify the basic parameters of the memory device
- Built - in dynamic delay compensation circuit (DCC) for reliable data sending and receiving
- The ECC function can detect 1 - and 2-bit errors in the data path and automatically correct 1 - bit errors
- Support 133-667mhz operating frequency

8.2 Ddr2/3 SDRAM read operation protocol

The protocol for ddr2/3 SDRAM read operations is shown in figure 11-2. In the figure, the Command (CMD) consists of RAS_n, CAS_n and WE_n. For read operations, RAS_n=1, CAS_n = 0,

and WE_n =1.

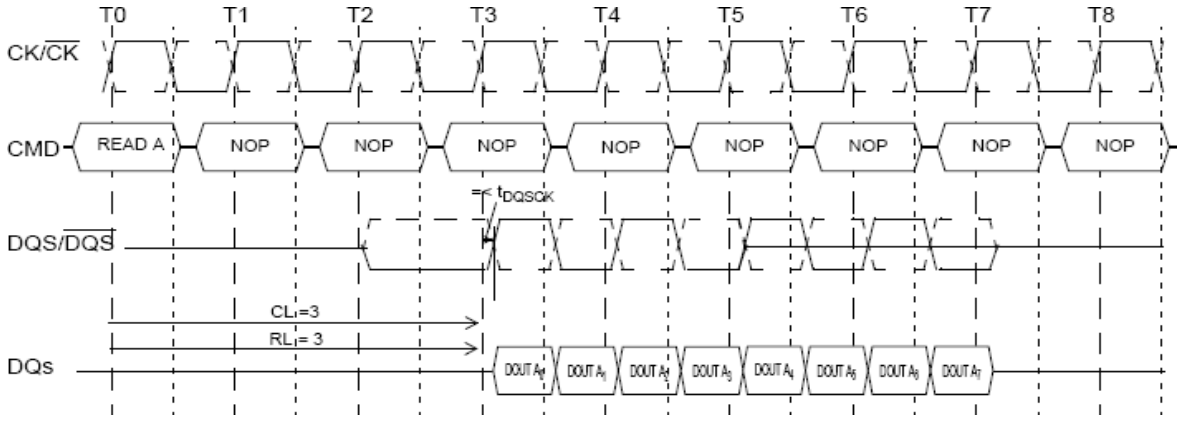


Figure 9-1 DDR2 SDRAM read operation protocol

In the figure above, Cas Latency (CL) = 3, Read Latency (RL) = 3, and Burst Length = 8.

8.3 Ddr2/3 SDRAM write operation protocol

The protocol for ddr2/3 SDRAM write operations is shown in figure 11-3. In the figure, the command CMD is composed of RAS_n, CAS_n, and WE_n. For write operations, RAS_n=1, CAS_n = 0, and WE_n = 0. Also, unlike read operations, write operations require DQM to identify the mask of the write operation, that is, the number of bytes to write.DQM synchronizes with DQs signal in the figure.

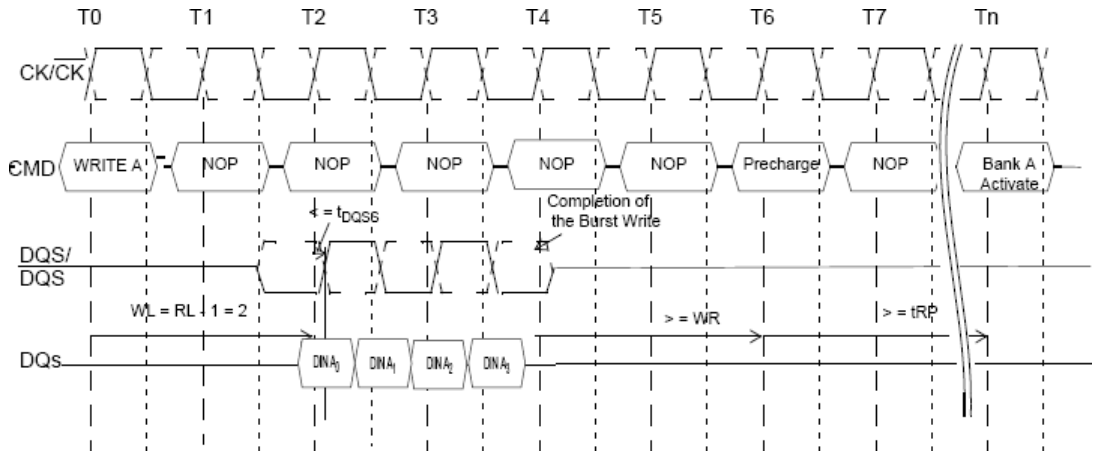


Figure 9-2 DDR2 SDRAM write operation protocol

In the figure above, Cas Latency (CL) = 3, Write Latency (WL) = Read Latency (RL) -- 1 = 2, and Burst Length = 4.

8.4 Ddr2/3 SDRAM parameter configuration format

The visible parameter list and description of the memory controller software are shown in the following table:

	63:56	55:48	47:40	39:32	came	Ephron;	"	away
x000	Dll_value_0 / Dll_adj_cnt (RD)		Dll_value_ck (RD)		Dll_init_done (RD)		Version (RD)	
x008	Dll_value_4 (RD)		Dll_value_3 (RD)		Dll_value_2 (RD)		Dll_value_1 (RD)	
x010	Dll_value_8 (RD)		Dll_value_7 (RD)		Dll_value_6 (RD)		Dll_value_5 (RD)	
x018	Dll_ck_3	Dll_ck_2	Dll_ck_1	Dll_ck_0	Dll_increment	Dll_start_point	Dll_bypass	Init_start
x020	Dq_oe_end_0	Dq_oe_begin_0	Dq_stop_edge_0	Dq_start_edge_0	Rddata_delay_0	Rddqs_lt_half_0	Wrdqs_lt_half_0	Wrdq_lt_half_0
x028	Rd_oe_end_0	Rd_oe_begin_0	Rd_stop_edge_0	Rd_start_edge_0	Dqs_oe_end_0	Dqs_oe_begin_0	Dqs_stop_edge_0	Dqs_start_edge_0
x030	Enzi_end_0	Enzi_begin_0	Wrclk_sel_0	Wrdq_clkdelay_0	Odt_oe_end_0	Odt_oe_begin_0	Odt_stop_edge_0	Odt_start_edge_0
x038	Enzi_stop_0	Enzi_start_0		Dll_rddqs_n_0	Dll_rddqs_p_0	Dll_wrdqs_0	Dll_wrdata_0	Dll_gate_0
x040	Dq_oe_end_1	Dq_oe_begin_1	Dq_stop_edge_1	Dq_start_edge_1	Rddata_delay_1	Rddqs_lt_half_1	Wrdqs_lt_half_1	Wrdq_lt_half_1
x048	Rd_oe_end_1	Rd_oe_begin_1	Rd_stop_edge_1	Rd_start_edge_1	Dqs_oe_end_1	Dqs_oe_begin_1	Dqs_stop_edge_1	Dqs_start_edge_1
x050	Enzi_end_1	Enzi_begin_1	Wrclk_sel_1	Wrdq_clkdelay_1	Odt_oe_end_1	Odt_oe_begin_1	Odt_stop_edge_1	Odt_start_edge_1
x058	Enzi_stop_1	Enzi_start_1		Dll_rddqs_n_1	Dll_rddqs_p_1	Dll_wrdqs_1	Dll_wrdata_1	Dll_gate_1
x060	Dq_oe_end_2	Dq_oe_begin_2	Dq_stop_edge_2	Dq_start_edge_2	Rddata_delay_2	Rddqs_lt_half_2	Wrdqs_lt_half_2	Wrdq_lt_half_2
x068	Rd_oe_end_2	Rd_oe_begin_2	Rd_stop_edge_2	Rd_start_edge_2	Dqs_oe_end_2	Dqs_oe_begin_2	Dqs_stop_edge_2	Dqs_start_edge_2
x070	Enzi_end_2	Enzi_begin_2	Wrclk_sel_2	Wrdq_clkdelay_2	Odt_oe_end_2	Odt_oe_begin_2	Odt_stop_edge_2	Odt_start_edge_2
x078	Enzi_stop_2	Enzi_start_2		Dll_rddqs_n_2	Dll_rddqs_p_2	Dll_wrdqs_2	Dll_wrdata_2	Dll_gate_2
x080	Dq_oe_end_3	Dq_oe_begin_3	Dq_stop_edge_3	Dq_start_edge_3	Rddata_delay_3	Rddqs_lt_half_3	Wrdqs_lt_half_3	Wrdq_lt_half_3
x088	Rd_oe_end_3	Rd_oe_begin_3	Rd_stop_edge_3	Rd_start_edge_3	Dqs_oe_end_3	Dqs_oe_begin_3	Dqs_stop_edge_3	Dqs_start_edge_3
x090	Enzi_end_3	Enzi_begin_3	Wrclk_sel_3	Wrdq_clkdelay_3	Odt_oe_end_3	Odt_oe_begin_3	Odt_stop_edge_3	Odt_start_edge_3
x098	Enzi_stop_3	Enzi_start_3		Dll_rddqs_n_3	Dll_rddqs_p_3	Dll_wrdqs_3	Dll_wrdata_3	Dll_gate_3
x0A0	Dq_oe_end_4	Dq_oe_begin_4	Dq_stop_edge_4	Dq_start_edge_4	Rddata_delay_4	Rddqs_lt_half_4	Wrdqs_lt_half_4	Wrdq_lt_half_4
X0A8	Rd_oe_end_4	Rd_oe_begin_4	Rd_stop_edge_4	Rd_start_edge_4	Dqs_oe_end_4	Dqs_oe_begin_4	Dqs_stop_edge_4	Dqs_start_edge_4
x0B0	Enzi_end_4	Enzi_begin_4	Wrclk_sel_4	Wrdq_clkdelay_4	Odt_oe_end_4	Odt_oe_begin_4	Odt_stop_edge_4	Odt_start_edge_4
x0B8	Enzi_stop_4	Enzi_start_4		Dll_rddqs_n_4	Dll_rddqs_p_4	Dll_wrdqs_4	Dll_wrdata_4	Dll_gate_4
x0C0	Dq_oe_end_5	Dq_oe_begin_5	Dq_stop_edge_5	Dq_start_edge_5	Rddata_delay_5	Rddqs_lt_half_5	Wrdqs_lt_half_5	Wrdq_lt_half_5
x0C8	Rd_oe_end_5	Rd_oe_begin_5	Rd_stop_edge_5	Rd_start_edge_5	Dqs_oe_end_5	Dqs_oe_begin_5	Dqs_stop_edge_5	Dqs_start_edge_5
x0D0	Enzi_end_5	Enzi_begin_5	Wrclk_sel_5	Wrdq_clkdelay_5	Odt_oe_end_5	Odt_oe_begin_5	Odt_stop_edge_5	Odt_start_edge_5
x0D8	Enzi_stop_5	Enzi_start_5		Dll_rddqs_n_5	Dll_rddqs_p_5	Dll_wrdqs_5	Dll_wrdata_5	Dll_gate_5
x0E0	Dq_oe_end_6	Dq_oe_begin_6	Dq_stop_edge_6	Dq_start_edge_6	Rddata_delay_6	Rddqs_lt_half_6	Wrdqs_lt_half_6	Wrdq_lt_half_6
x0E8	Rd_oe_end_6	Rd_oe_begin_6	Rd_stop_edge_6	Rd_start_edge_6	Dqs_oe_end_6	Dqs_oe_begin_6	Dqs_stop_edge_6	Dqs_start_edge_6
x0F0	Enzi_end_6	Enzi_begin_6	Wrclk_sel_6	Wrdq_clkdelay_6	Odt_oe_end_6	Odt_oe_begin_6	Odt_stop_edge_6	Odt_start_edge_6
x0F8	Enzi_stop_6	Enzi_start_6		Dll_rddqs_n_6	Dll_rddqs_p_6	Dll_wrdqs_6	Dll_wrdata_6	Dll_gate_6
x100	Dq_oe_end_7	Dq_oe_begin_7	Dq_stop_edge_7	Dq_start_edge_7	Rddata_delay_7	Rddqs_lt_half_7	Wrdqs_lt_half_7	Wrdq_lt_half_7
x108	Rd_oe_end_7	Rd_oe_begin_7	Rd_stop_edge_7	Rd_start_edge_7	Dqs_oe_end_7	Dqs_oe_begin_7	Dqs_stop_edge_7	Dqs_start_edge_7
x110	Enzi_end_7	Enzi_begin_7	Wrclk_sel_7	Wrdq_clkdelay_7	Odt_oe_end_7	Odt_oe_begin_7	Odt_stop_edge_7	Odt_start_edge_7

x118	Enzi_stop_7	Enzi_start_7		Dll_rddqs_n_7	Dll_rddqs_p_7	Dll_wrdqs_7	Dll_wrdata_7	Dll_gate_7
	63:56	55:48	47:40	39:32	came	Ephron;	"	away
X120 measures how	Dq_oe_end_8	Dq_oe_begin_8	Dq_stop_edge_8	Dq_start_edge_8	Rddata_delay_8	Rddqs_lt_half_8	Wrdqs_lt_half_8	Wrdq_lt_half_8
x128	Rd_oe_end_8	Rd_oe_begin_8	Rd_stop_edge_8	Rd_start_edge_8	Dqs_oe_end_8	Dqs_oe_begin_8	Dqs_stop_edge_8	Dqs_start_edge_8
x130	Enzi_end_8	Enzi_begin_8	Wrclk_sel_8	Wrdq_clkdelay_8	Odt_oe_end_8	Odt_oe_begin_8	Odt_stop_edge_8	Odt_start_edge_8
x138	Enzi_stop_8	Enzi_start_8		Dll_rddqs_n_8	Dll_rddqs_p_8	Dll_wrdqs_8	Dll_wrdata_8	Dll_gate_8
x140	Pad_ocd_clk	Pad_ocd_ctl	Pad_ocd_dqs	Pad_ocd_dq	Pad_enzi		Pad_en_ctl	Pad_en_clk
x148	Pad_adj_code_dqs	Pad_code_dqs	Pad_adj_code_dq	Pad_code_dq		Pad_vref_internal	Pad_odt_se	Pad_modezi1v8
x150		Pad_reset_po	Pad_adj_code_clk	Pad_code_lk	Pad_adj_code_cmd	Pad_code_cmd	Pad_adj_code_addr	Pad_code_addr
x158		Pad_comp_code_o	Pad_comp_okn	Pad_comp_code_i		Pad_comp_mode	Pad_comp_tm	Pad_comp_pd
x160	Rdfifo_empty (RD)		Overflow (RD)		Dram_init (RD)	Rdfifo_valid	Cmd_timming	Ddr3_mode
x168	Ba_xor_row_offset	Addr_mirror	Cmd_delay	Burst_length	Bank	Cs_zq	Cs_mrs	Cs_enable
x170	Odt_wr_cs_map		Odt_wr_length	Odt_wr_delay	Odt_rd_cs_map		Odt_rd_length	Odt_rd_delay
x178								
x180	Lvl_resp_0 (RD)	Lvl_done (RD)	Lvl_ready (RD)		Lvl_cs	tLVL_DELAY	Lvl_req (WR)	Lvl_mode
x188	Lvl_resp_8 (RD)	Lvl_resp_7 (RD)	Lvl_resp_6 (RD)	Lvl_resp_5 (RD)	Lvl_resp_4 (RD)	Lvl_resp_3 (RD)	Lvl_resp_2 (RD)	Lvl_resp_1 (RD)
x190	Cmd_a		Cmd_ba	Cmd_cmd	Cmd_cs	Status_cmd (RD)	Cmd_req (WR)	Command_mode
x198			Status_sref (RD)	Srefresh_req	Pre_all_done (RD)	Pre_all_req (RD)	Mrs_done (RD)	Mrs_req (WR)
x1A0	Mr_3_cs_0		Mr_2_cs_0		Mr_1_cs_0		Mr_0_cs_0	
x1A8	Mr_3_cs_1		Mr_2_cs_1		Mr_1_cs_1		Mr_0_cs_1	
x1B0	Mr_3_cs_2		Mr_2_cs_2		Mr_1_cs_2		Mr_0_cs_2	
x1B8	Mr_3_cs_3		Mr_2_cs_3		Mr_1_cs_3		Mr_0_cs_3	
x1C0	tRESET	tCKE	tXPR	tMOD	tZQCL	tZQ_CMD	tWLDQSEN	tRDDATA
x1C8	tFAW	tRRD	tRCD	tRP	tREF	tRFC	tZQCS	tZQperiod
x1D0	tODTL	tXSRD	tPHY_RDLAT	tPHY_WRLAT	tRAS_max			tRAS_min
x1D8	tXPDLL	tXP	tWR	tRTP	tRL	these	tCCD	tWTR
x1E0	tW2R_diffCS	tW2W_diffCS	tR2P_sameBA	tW2P_sameBA	tR2R_sameBA	tR2W_sameBA	tW2R_sameBA	tW2W_sameBA
x1E8	tR2R_diffCS	tR2W_diffCS	tR2P_sameCS	tW2P_sameCS	tR2R_sameCS	tR2W_sameCS	tW2R_sameCS	tW2W_sameCS
x1F0	Power_up	Age_step	tCPDED	Cs_map	Bs_config	Nc	Pr_r2w	Placement_en
x1F8	Hw_pd_3	Hw_pd_2	Hw_pd_1	Hw_pd_0	Credit_16	Credit_32	Credit_64	Selection_en
x200	Cmdq_age_16		Cmdq_age_32		Cmdq_age_64		tCKESR	tRDPDEN
x208	Wfifo_age		Rfifo_age		Power_stat3	Power_stat2	Power_stat1	Power_stat0
x210	Active_age		Cs_place_0	Addr_win_0	Cs_diff_0	Row_diff_0	Ba_diff_0	Col_diff_0
x218	Fastpd_age		Cs_place_1	Addr_win_1	Cs_diff_1	Row_diff_1	Ba_diff_1	Col_diff_1
x220	Slowpd_age		Cs_place_2	Addr_win_2	Cs_diff_2	Row_diff_2	Ba_diff_2	Col_diff_2
x228	Selfref_age		Cs_place_3	Addr_win_3	Cs_diff_3	Row_diff_3	Ba_diff_3	Col_diff_3
x230	Win_mask_0				Win_base_0			
x238	Win_mask_1				Win_base_1			
x240	Win_mask_2				Win_base_2			
x248	Win_mask_3				Win_base_3			

x250		Cmd_monitor	Axi_monitor		Ecc_code (RD)	Ecc_enable	Int_vector	Int_enable
	63:56	55:48	47:40	39:32	came	Ephron;	"	away
x258								
x260	Ecc_addr (RD)							
x268	Ecc_data (RD)							
x270	Lpbk_ecc_mask (RD)	Prbs_init			Lpbk_error (RD)	Prbs_23	Lpbk_start	Lpbk_en
x278	Lpbk_ecc (RD)		Lpbk_data_mask (RD)		Lpbk_correct (RD)		Lpbk_counter (RD)	
x280	Lpbk_data_r (RD)							
x288	Lpbk_data_f (RD)							
x290	Axi0_bandwidth_w				Axi0_bandwidth_r			
x298	Axi0_latency_w				Axi0_latency_r			
x2A0	Axi1_bandwidth_w				Axi1_bandwidth_r			
x2A8	Axi1_latency_w				Axi1_latency_r			
x2B0	Axi2_bandwidth_w				Axi2_bandwidth_r			
x2B8	Axi2_latency_w				Axi2_latency_r			
x2C0	Axi3_bandwidth_w				Axi3_bandwidth_r			
x2C8	Axi3_latency_w				Axi3_latency_r			
x2D0	Axi4_bandwidth_w				Axi4_bandwidth_r			
x2D8	Axi4_latency_w				Axi4_latency_r			
x2E0	Cmdq0_bandwidth_w				Cmdq0_bandwidth_r			
x2E8	Cmdq0_latency_w				Cmdq0_latency_r			
x2F0	Cmdq1_bandwidth_w				Cmdq1_bandwidth_r			
x2F8	Cmdq1_latency_w				Cmdq1_latency_r			
Adaptive:	Cmdq2_bandwidth_w				Cmdq2_bandwidth_r			
x308	Cmdq2_latency_w				Cmdq2_latency_r			
x310	Cmdq3_bandwidth_w				Cmdq3_bandwidth_r			
x318	Cmdq3_latency_w				Cmdq3_latency_r			
x320								tREF_low
x328								
x330	Stat_en	Rdbuffer_max	Retry	Wr_pkg_num	Rwq_rb	Stb_en	Addr_new	tRDQidle
x338				Rd_fifo_depth	Retry_cnt			
x340	tREFretention					Ref_num	tREF_IDLE	Ref_sch_en
x348								
x350	Lpbk_data_en							
x358						Lpbk_ecc_mask_en	Lpbk_ecc_en	Lpbk_data_mask_en
x360			Int_ecc_cnt_fatal	Int_ecc_cnt_err The or	Ecc_cnt_cs_3	Ecc_cnt_cs_2	Ecc_cnt_cs_1	Ecc_cnt_cs_0
x368								

8.5 Software programming guide

9.5.1 Initialization operation

Initialization begins when the software writes 1 to register Init_start (0x018), setting Init_start

All other registers must be set to the correct value before signaling.The DRAM initialization process of software/hardware collaboration is as follows:

(1) The software writes the correct configuration values to all registers, but Init_start (0x018) must remain at 0 during this process;

(2) The software sets Init_start (0x018) to 1, which results in the start of hardware initialization.

(3) The initialization begins internally at PHY, and the DLL will attempt to lock.If the lock is successful, the corresponding state can be read from Dll_init_done (0x000) and the number of current lock delay lines can be read and written from Dll_value_ck (0x000).If the lock is not successful, the initialization will not continue (this can be done by setting Dll_bypass (0x018));

(4) After DLL lock (or bypass setting), the controller will be directed to the DRAM according to the initialization requirements of the corresponding DRAM

Issue the corresponding initialization sequence, such as the corresponding MRS command, ZQCL command, etc.

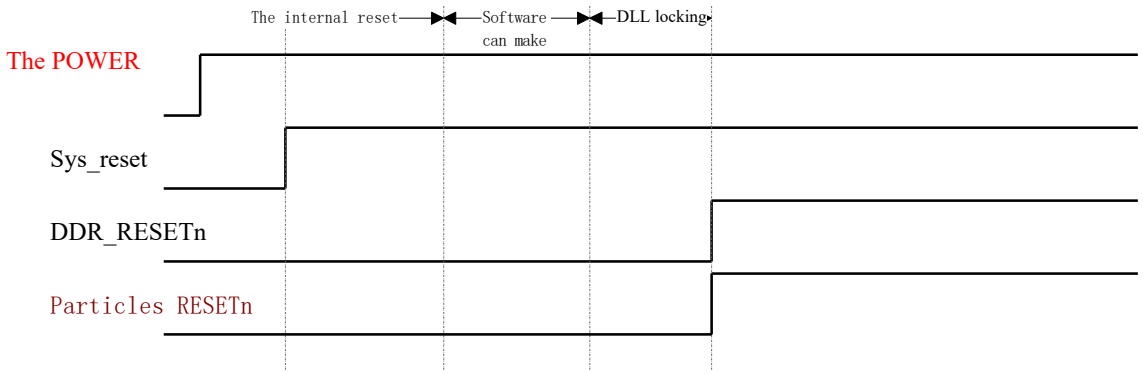
(5) The software can determine if the memory initialization operation is complete by sampling the Dram_init (0x160) register.

9.5.2 Reset pin control

In order to control the reset pin more easily in the state of STR and so on, special reset pin (DDR_RESETn) control can be carried out through the reset_ctrl (0x150) register. There are two main control modes:

In general mode, reset_ctrl[1:0] == 2 'b00.In this mode, the reset signal pin behavior is compatible with the general control mode. DDR_RESETn is directly connected to the corresponding pin on the memory slot on the motherboard. The behavior of the pin is:

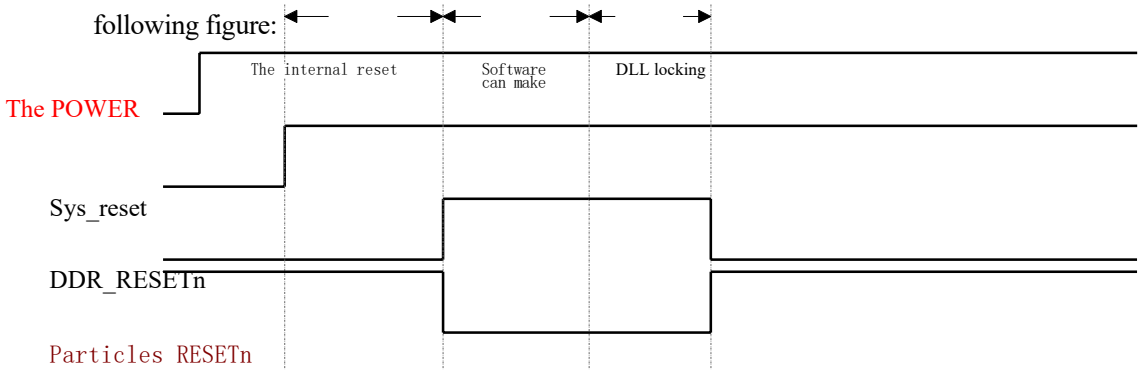
- When the power is not on: the pin state is low;
- When power on: the pin state is low;
- When the controller starts to initialize, the pin state is high;
- When working normally, the pin state is high.The sequence is shown in the following figure:



Reverse mode, reset_ctrl[1:0] == 2 'b10.In this mode, the active level of reset pin is contrary to the normal control mode.So DDR_RESETn needs to be connected to the corresponding pin on the memory slot on the motherboard through the inverter.The behavior of the pin is:

- When the power is not on: the pin state is low;
- When power on: the pin state is low;
- When the controller is configured: the pin state is high;
- When the controller starts to initialize: the pin state is low;
- Normal operation: the pin state

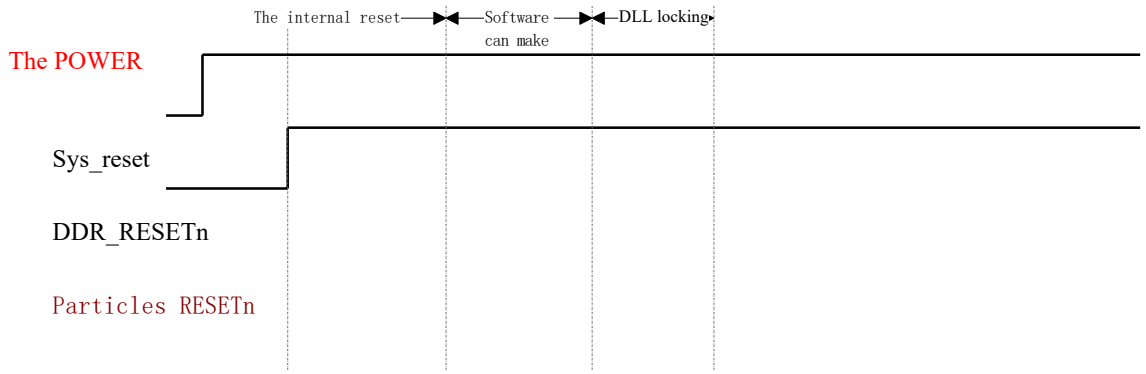
is low.The sequence is shown in the following figure:



Reset disable mode, pm_reset_ctrl[1:0] == 2 'b01.In this mode, the reset signal pin remains low throughout the memory operation. So DDR_RESETn needs to be connected to the corresponding pin on the memory slot on the motherboard through the inverter. The behavior of the pin is:

- Always low;

The sequence is shown in the following figure:



With the combination of the latter two reset modes, STR control can be implemented directly using the reset signal of the memory controller. When the whole system starts from the closed state, use the method in (2) to reset the memory bar normally and start working. When the system recovers from STR, the method in (3) is used to reconfigure the memory bar so that it can resume normal operation without destroying the original state of the memory bar.

9.5.3 Leveling

It was used in DDR3 to intelligently configure the phase relationship between the various signals in the read and write operations of the memory controller. It's often the case that it was written like that, like Read like that, like Gate like that. In this controller, there was as much Write as Gate and as little Read as possible. The software needed to be able to Read as well as Read. In addition to the DQS phase and GATE phase operated in the process, the configuration method of writing DQ phase and reading DQ phase can also be calculated based on these final confirmed phases.

9.5.3.1 The Write Leveling

(1) Write ever was used to configure the phase relationship between Write DQS and the clock. Software programming required the following steps.

(2) Complete the initialization of the controller, as described in the previous section.

(3) Dll_wrdqs_x (x = 0... 8) set to 0;

(4) Set Lvl_mode (0x180) to 2 'b01;

(5) Sample the Lvl_ready (0x180) register. If it is 1, you can begin to Write a request.

(6) Set Lvl_req (0x180) to 1;

(7) Sample the Lvl_done (0x180) register. If it is 1, it shows that a Write request was done.

(8) Sample Lvl_resp_ (x 0x180, 0x188) registers, if 0, the corresponding Dll_wrdqs_x[6:0] Add 1 and repeat 5-7; If it was 1, it must have been successful.

(9) At this point, the value of Dll_wrdqs_x should be the correct setting value.

(10) Now it was written. If Lvl_resp_x is found to be 1 on the first sampling, this result is problematic and other registers should be checked for incorrect Settings, such as Wrdqs_lt_half, Dqs_start_edge, Dqs_stop_edge, Dqs_oe_begin, and Dqs_oe_end.

(11) Then set Wrdqs_lt_half_x based on whether the value of Dll_wrdqs_x is less than 0x40;

(12) Dll_wrdqs_x is set based on whether the value of Dll_wrdqs_x is less than 0x20. If Dll_wrdqs_x > 0x20, Dll_wrdata_x = Dll_wrdqs_x -- 0x20, otherwise Dll_wrdata_x = Dll_wrdqs_x
+ 0 x60;

(13) Set Wrdata_lt_half_x based on whether Dll_wrdata_x is less than 0x40;

(14) Determine if the following conditions exist: different Dll_wrdata_x values are near 0x40, and there are cases that cross the 0x40 boundary (meaning that some Dll_wrdata_x is slightly less than 0x40, and some Dll_wrdata_x is slightly greater than 0x40). If this happens, set Write_clk_delay_x to 1 for the Wrdata_lt_half_x == 0 data group. Then subtract 1 from the values of tPHY_WRDATA and tRDDATA.

(15) Set Lvl_mode (0x180) to 2 'b00 to exit from Write like this.

9.5.3.2 Gate Leveling

Gate was used to configure the timing of the DQS window in the controller. The following steps were used in software programming.

(1) Complete the initialization of the controller, as described in the previous section.

(2) See the last measure.

(3) Dll_gate_x (x = 0... 8) set to 0;

(4) Set Lvl_mode (0x180) to 2 'b10;

(5) Sample Lvl_ready (0x180) register. If it's 1, you can start Gate like this.

(6) Set Lvl_req (0x180) to 1;

(7) Sample the Lvl_done (0x180) register. If it is 1, it means a Gate like request was done.

(8) Sample the Lvl_resp_x[0] (0x180, 0x188) registers.If Lvl_resp_x[0] is found to be 1 on the first sampling, increment the corresponding Dll_gate_x[6:0] by 1 and repeat 6-8 until the sampling result is 0, otherwise proceed to the next step.

(9) If the sample result is 0, increment the corresponding Dll_gate_x[6:0] by 1 and repeat 6-9;If it is

(10)

(11) Gate was a success.

(12) So this was the end of Gate. Now the sum between Dll_gate_x[6:0] and Dll_wrdata_x[6:0] was in effect a phase relationship between DQS and the internal clock over the PHY. Now let's adjust the parameters accordingly.

(13) Dll_rddqs_lt_halt is set to 1 if the sum of Dll_gate_x[6:0] and Dll_wrdata_x[6:0] is less than or greater than 0x20 or 0x60. Because the phase relation of RDDQS is actually equal to a quarter of the input read DQS.

(14) At this point, if the value of Dll_gate_x is greater than 0x40, subtract 0x40 from the value of Dll_gate_x; Otherwise, set it to 0.

(15) Once the adjustment is completed, Lvl_req is performed for two more times respectively to observe the change in the values of Lvl_resp_x[7:5] and Lvl_resp_x[4:2]. If the values of Lvl_resp_x[7:5] and Lvl_resp_x[4:2] are increased to Burst_length/2, the 13th operation is continued. If it is not 4, you may need to add or subtract one from Rd_oe_begin_x, and if it is greater than Burst_length/2, you will most likely need to tweak the value of Dll_gate_x

(16) Set Lvl_mode (0x180) to 2 'b00 to exit Gate.

(17)

9.5.4 Issue the MRS command separately

The order of MRS commands sent to memory by the

memory controller is: MR2_CS0, MR2_CS1,

MR2_CS2, MR2_CS3, MR3_CS0, MR3_CS2,

MR1_CS3, MR1_CS3, MR1_CS0, MR1_CS1,

MR1_CS2, MR1_CS3, MR1_CS3, MR1_CS1,

MR1_CS2, MR1_CS3, MR1_CS3.

Among them, whether the MRS command of corresponding CS is valid or not is determined by Cs_mrs. Only if the bits of each slice selected on Cs_mrs are valid can this MRS command be issued to DRAM. The value of each MR is determined by the register MR *_cs*. These values are also used for the MRS command when initializing memory.

The specific operation is as follows:

(1) Set registers Cs_mrs (0x168), Mr*_cs* (0x190 -- 0x1B8) to the correct values;

(2) Set Command_mode (0x190) to 1 to enable the controller to enter command sending mode;

(3) Sample Status_cmd (0x190). If it is 1, it means that the controller has entered command sending mode and can proceed to the next operation. If it is 0, it needs to continue to wait.

(4) Write Mrs_req (0x198) to 1 and send MRS command to DRAM;

(5) Sample Mrs_done (0x198). If it is 1, it means that the MRS command has been sent and can exit. If it is 0, it needs to continue to wait.

(6) Set Command_mode (0x190) to 0 to enable the controller to exit command send mode.

9.5.5 Any operation controls the bus

The memory controller can issue any combination of commands to the DRAM in command send mode, which can be set by the software

Cmd_cs, Cmd_cmd, Cmd_ba, Cmd_a (0x168), issued to DRAM in command send mode.The specific operation is as follows:

(1) Set registers Cmd_cs, Cmd_cmd, Cmd_ba, Cmd_a (0x190) to the correct values;

(2) Set Command_mode (0x190) to 1 to enable the controller to enter command sending mode;

(3) Sample Status_cmd (0x190). If it is 1, it means that the controller has entered command sending mode and can proceed to the next operation. If it is 0, it needs to continue to wait.

(4) Write Cmd_req (0x190) to 1 and send the command to DRAM;

(5) Set Command_mode (0x190) to 0 to enable the controller to exit command send mode.

9.5.6 Self-looping test mode control

Since the cycle test pattern can be respectively in test mode or normal mode using the function, therefore, the memory controller, the two sets of independent control interface is implemented, a set of used in the test mode directly controlled by the test port, another set of used in normal function mode by the register allocation module configuration can make the test.

The multiplexing of these two interfaces is controlled by port test_phy. When test_phy is effective, the test_* port of the controller is used for control. At this time, the self-test is completely controlled by hardware. When test_phy is invalid, the parameters of pm_* programmed by the software are used for

control. The specific signal meaning for using the test port can be referred to the register parameter with the same name.

The two sets of interface from the control parameters are basically the same, only different access points, in this article introduces the control method of software programming.The specific operation is as follows:

(1) Set all parameters of the memory controller correctly;

(2) Set register Lpbk_en (0x270) to 1;

(3) Set register Init_start (0x018) to 1;

(4) The sample register Dll_init_done (0x000), if this value is 1, means that the DLL is locked, yes

Carry out the next operation; If this value is 0, you need to wait; (when using the test port for control, since the output of this register is not visible, it is not necessary to sample this register, but only need to wait here for a certain amount of time to ensure that the DLL lock is completed before proceeding to the next operation);

(5) Set register Lpbk_start (0x270) to 1; This is when the loop test begins.

Since the cycle test has started, the software needs to check whether there are any errors frequently. The specific operation is as follows:

(6) Sampling register Lpbk_error (0x270), if this value is 1, it means that an error has occurred. At this time, registers (0x270, 0x278, 0x280, 0x288) can be used to observe the error data and the correct data in the first error through Lpbk_* and other observations. If this value is 0, no data errors have occurred.

9.5.7 ECC functions use controls

ECC functionality is only available in 64-bit mode.

Ecc_enable includes the following four control bits:

Ecc_enable[0] controls whether the ECC function is enabled or not. ECC function will only be enabled if this significant bit is set.

Ecc_enable[1] controls whether or not an error is reported through the read response path inside the processor so that a read access with ECC two-digit errors can immediately result in an exception in the processor core.

Ecc_enable[2] controls whether an error is reported through the write response path within the processor so that a write access (read after write) with ECC two-digit errors can immediately cause an

exception to occur in the processor core.

Ecc_enable[3] controls when the error message is triggered in the register.These error messages will not be triggered continuously without software to process them, only the first error will be recorded. This information includes Ecc_code, Ecc_addr, Ecc_data. When Ecc_enable[3] is 0, as long as there is an ECC error (including 1 dislocation and 2 dislocation), the record will be triggered. When Ecc_enable[3] is 1, the record will be triggered only if there is an ECC two-digit error. This "first time" refers to the setting of the corresponding bit of the interrupt vector register. That is, the access that caused the interruption is recorded.

In addition, ECC errors can be notified to the processor core by means of an interrupt. This interrupt is controlled with Int_enable. The interrupt consists of two vectors, Int_vector[0], indicating ECC error (including 1 dislocation and 2 dislocation), and Int_vecotr[1], indicating ECC two-bit error.The clearing of Int_vector is achieved by writing 1 to the corresponding bit.

9 HyperTransport controller

In loongson 3A2000, the HyperTransport bus is used for external device connection and multi-chip interconnection. When used for connecting peripherals, but by the user program free to choose whether to support the IO Cache consistency (through the address window Uncache Settings, see section 10.5.13) : when configured to support the Cache consistency model, IO device internal DMA access for transparent Cache levels, namely by the hardware, automatically maintain consistency without software Cache instructions for maintenance by the program; When HyperTransport bus is used for multi-chip interconnection, HT0 controller (initial address 0x0C00_0000_0000)

0x0DFF_FFFF_FFFF) can be configured to support inter-chip Cache consistency transfer, while the HT1 controller (initial address 0x0E00_0000_0000 -- 0x0FFF_FFFF_FFFF) can be configured to support inter-chip Cache consistency maintenance, as shown in section 10.8.

The HyperTransport controller supports up to a two-way 16-bit width and a 2.0-ghz operating frequency. After the automatic initialization of the system to establish a connection, the user program can change the width and running frequency by modifying the corresponding configuration register in the protocol, and then re-initialize, as detailed in section 10.1.

The main features of loongson 3A2000 HyperTransport controller are as follows:

- Support for the HT1.0/HT3.0 protocol
- Support 200/400/800/1600/2000mhz operating frequency
- HT1.0 supports 8-bit widths
- HT3.0 supports 8/16 bit widths
- Each HT controller (HT0/HT1) can be configured as two 8-bit HT controllers

- The bus control signal (including PowerOK, Rstn, LDT_Stopn) direction is configurable
- Peripheral DMA space Cache/Uncache is configurable
- Cache consistency mode can be configured for multi-slice interconnection

9.1 HyperTransport hardware setup and initialization

The HyperTransport bus consists of a transmission signal bus and a control signal pin, as shown in the following table

HyperTransport bus-related pins and their functional descriptions.

Table 10-1 HyperTransport bus-related pin signals

pin	The name of the	describe
HT0_8x2	Bus width configuration	1: configure the 16-bit HyperTransport bus as two independent 8-bit buses, It is controlled by two independent controllers, and the address space is distinguished as HT0_Lo: address[40] = 0; HT0_Hi: address[40] = 1; 0: use the 16-bit HyperTransport bus as a 16-bit bus by HT0_Lo control, address space is the address of HT0_Lo, that is, address[40] = 0; All signals of HT0_Hi are invalid.
HT0_Lo_mode	Master device mode	1: set HT0_Lo as the main device mode. In this mode, bus control signals are driven by HT0_Lo, including HT0_Lo_Powerok, HT0_Lo_Rstn, HT0_Lo_Ldt_Stopn.In this mode, these control signals can also be bidirectional.At the same time, this pin determines the initial value of the register "Act as Slave". When this register is 0, the Bridge bit in the packet on the HyperTransport bus is 1, otherwise it is 0. In addition, when this register is 0, if the request address on the HyperTransport bus is not hit by the controller's receive window, it will be sent back to the bus as a P2P request, and if this register is 1, if it is not hit, it will be responded as an error request. 0: set HT0_Lo to slave device mode. In this mode, bus control signals are driven by other devices, including HT0_Lo_Powerok, HT0_Lo_Rstn, HT0_Lo_Ldt_Stopn.In this mode, these control

		signals are driven by each other's devices, or if they are not driven correctly, the HT bus Not working properly.
HT0_Lo_Powerok	Bus Powerok	When the HyperTransport master line Powerok number, HT0_Lo_Mode is 1, it is controlled by HT0_Lo; When HT0_Lo_Mode is 0, it is controlled by the other device.
HT0_Lo_Rstn	Bus Rstn	When the Rstn signal number of HyperTransport main line, HT0_Lo_Mode is 1, it is controlled by HT0_Lo; When HT0_Lo_Mode is 0, it is controlled by the other device.
HT0_Lo_Ldt_Stopn	Bus Ldt_Stopn	HyperTransport main line Ldt_Stopn, HT0_Lo_Mode = 1, is controlled by HT0_Lo; When HT0_Lo_Mode is 0, it is controlled by the other device.
HT0_Lo_Ldt_Reqn	Bus Ldt_Reqn	HyperTransport bus Ldt_Reqn signal,
HT0_Hi_mode	Master device mode	1: set HT0_Hi to the main device mode. In this mode, bus control signals are driven by HT0_Hi. These control signals include HT0_Hi_Powerok, HT0_Hi_Rstn, HT0_Hi_Ldt_Stopn.In this mode, these control signals can also be bidirectional.At the same time, this pin determines the initial value of the register "Act as Slave". When this register is 0, the Bridge bit in the packet on the HyperTransport bus is 1, otherwise it is 0. In addition, when this register is 0, if the request address on the HyperTransport bus is not hit by the controller's receive window, it will be sent back to the bus as a P2P request, and if this register is 1, if it is not hit, it will be responded as an error request.

		0: set HT0_Hi to slave device mode, in which bus control signals, etc Driven by the other device, these control signals include HT0_Hi_Powerok,

		HT0_Hi_Rstn HT0_Hi_Ldt_Stopn.In this mode, these control signals are driven by each other's devices, or if they are not driven correctly, the HT bus Not working properly.
HT0_Hi_Powerok	Bus Powerok	When the HyperTransport main line Powerok number, HT0_Lo_Mode is 1, it is controlled by HT0_Hi; When HT0_Lo_Mode is 0, it is controlled by the other device. When HT0_8x2 is 1, control the high 8-bit bus; Invalid when HT0_8x2 is 0.
HT0_Hi_Rstn	Bus Rstn	When Rstn number of HyperTransport main line, HT0_Lo_Mode is 1, it is controlled by HT0_Hi; When HT0_Lo_Mode is 0, it is controlled by the other device. When HT0_8x2 is 1, control the high 8-bit bus; Invalid when HT0_8x2 is 0.
HT0_Hi_Ldt_Stopn	Bus Ldt_Stopn	HyperTransport main line Ldt_Stopn, HT0_Lo_Mode = 1, is controlled by HT0_Hi; When HT0_Lo_Mode is 0, it is controlled by the other device. When HT0_8x2 is 1, control the high 8-bit bus; Invalid when HT0_8x2 is 0.
HT0_Hi_Ldt_Reqn	Bus Ldt_Reqn	HyperTransport bus Ldt_Reqn signal, HT0_8x2 is 1, control the high 8-bit bus; Invalid when HT0_8x2 is 0.

HT0_Rx_CLKp HT0_Rx_CLKn [1:0] [1:0] HT0_Tx_CLKp (1-0) HT0_Tx_CLKp [1:0]	CLK [1:0]	HyperTransport bus CLK signal When HT0_8x2 is 1, CLK[1] is controlled by HT0_Hi CLK[0] is controlled by HT0_Lo when HT0_8x2 is 0, CLK[1:0] is controlled by HT0_Lo
HT0_Rx_CTLp HT0_Rx_CTLn [1:0] [1:0] HT0_Tx_CTLp [1:0] HT0_Tx_CTLn (1-0)	CTL (1-0)	HyperTransport bus CTL signal When HT0_8x2 is 1, CTL[1] is controlled by HT0_Hi When CTL[0] is controlled by HT0_Lo and HT0_8x2 is 0, CTL[1] is invalid The CTL[0] is controlled by HT0_Lo
HT0_Rx_CADp HT0_Rx_CADn [15:0] [15:0] HT0_Tx_CADp [15:0] HT0_Tx_CADn [15:0]	CAD [15:0]	HyperTransport bus CAD signal When HT0_8x2 is 1, CAD[15:8] is controlled by HT0_Hi CAD[7:0] is controlled by HT0_Lo when HT0_8x2 is 0, CAD[15:0] is controlled by HT0_Lo

The initialization of HyperTransport starts automatically after each reset is completed. After cold startup, the HyperTransport bus will automatically work at the lowest frequency (200MHz) and minimum width (8bit), and attempt to perform the bus initialization handshake.Whether the initialization is Complete or not can be read by the register "Init Complete" (see section 10.5.2).After initialization, the Width of the bus can be read from the registers "Link Width Out" and "Link Width In" (see section 10.5.2).After the initialization is completed, the user can rewrite the registers "Link Width Out", "Link Width In" and "Link Freq". At the same time, the user also needs to configure the corresponding register of the other device. After the configuration is completed, the user needs to hot-reset the bus or pass through

The "HT_Ldt_Stopn" signal is reinitialized to give effect to the overwritten value of the register. The HyperTransport bus will work at the new frequency and width after the reinitialization is complete. It is important to note that the configuration of devices at both ends of HyperTransport needs to be one-to-one, otherwise the HyperTransport interface will not work properly.

9.2 HyperTransport protocol support

Loongson 3A2000's HyperTransport bus supports most of the commands in the 1.03/3.0 protocol, and includes some extended instructions in the extended conformance protocol that supports multi-chip interconnection. In both modes, the HyperTransport receiver can receive commands as shown in the following table. It is important to note that the atomic operation command of the HyperTransport bus is not supported.

Table 10-2 commands that the HyperTransport receiver can receive

coding	channe l	The command	The standard model	Extension (consistency)
000000	-	The NOP	Empty packet or flow control	
000001	NPC	FLUSH	No operation	
x01xxx	NPC The or PC	The Write	Bit 5:0 - Nonposted 1 - Posted bit 2:0 - Byte 1 -- Doubleword bit 1: Don't Care Bit 0: Don't Care	Bit 5: must be 1, POSTED Bit 2:0 -- Byte 1 -- Doubleword bit 1: Don't Care Bit 0: must be 1
01 XXXX	NPC	The Read	Bit 3: Don't Care bit 2:0 -- Byte 1 -- Doubleword bit 1: Don't Care Bit 0: Don't Care	Bit 3: Don't Care bit 2:0 -- Byte 1 -- Doubleword bit 1: Don't Care Bit 0: must be 1
110000	R	RdRespons e	Read operation return	
110011	R	TgtDone	Write operation return	
110100	The PC	WrCoherent	----	Write command extension
110101	The PC	WrAddr	----	Write address extension
111000	R	RespCohere nt	----	Read response extension
111001	NPC	RdCoherent	----	Read command extension
111010	The PC	Broadcast	No operation	
111011	NPC	RdAddr	----	Read address extension
111100	The PC	A FENCE	Guarantee order relation	
111111	-	The Sync/Error	The Sync/Error	

For the sending side, the commands sent out in both modes are shown in the following table.

Table 10-3 commands sent out in both modes

coding	channe l	The comm and	The standard model	Extension (consistency)
000000	-	The NOP	Empty packet or flow control	

x01x0x	NPC The or PC	The Write	Bit 5:0 - Nonposted 1 - Posted bit 2:0 - Byte 1 - Doubleword	Bit 5: must be 1, POSTED Bit 2:0 -- Byte 1 - Doubleword

			Bit 0: must be 0	Bit 0: must be 1
010 x0x	NPC	The Read	Bit 2:0 -- Byte 1 - Doubleword Bit 0: Don't Care	Bit 2:0 -- Byte 1 - Doubleword Bit 0: must be 1
110000	R	RdResponse	Read operation return	
110011	R	TgtDone	Write operation return	
110100	The PC	WrCoherent	----	Write command extension
110101	The PC	WrAddr	----	Write address extension
111000	R	RespCoherent	----	Read response extension
111001	NPC	RdCoherent	----	Read command extension
111011	NPC	RdAddr	----	Read address extension
111111	-	The Sync/Error	Will only forward	

9.3 HyperTransport interrupt support

The HyperTransport controller provides 256 interrupt vectors and supports Fix, Arbiter, and other interrupt types, but no hardware automatic EOI support. For the above two supported types of interrupts, the controller will automatically write to the interrupt register after receiving and notify the system interrupt controller according to the Settings of the interrupt mask register. For the specific interrupt control, see the interrupt control register group in section 10.5.8.

In addition, the controller has special support for PIC interrupts to speed up interrupt handling of this type.

A typical PIC interrupt is completed by the following steps: (1) the PIC controller sends a PIC interrupt request to the system; The system sends interrupt vector query to PIC controller; (3) PIC controller sends interrupt vector number to the system; The system clears the corresponding interrupt on the PIC controller. The PIC controller will not issue the next interrupt to the system until all 4 steps above have been completed. For loongson 3A2000 HyperTransport controller, the first 3 steps will be automatically processed and the PIC interrupt vector will be written to the corresponding position in the 256 interrupt vectors. After the interrupt is processed by the software system, the fourth step is required, which is to send a clear interrupt to the PIC controller. Then the processing of the next interrupt begins.

9.4 HyperTransport address window

10.4.1 HyperTransport space

In loongson 3A2000 processor, the address window distribution of the default 4

HyperTransport interfaces is as follows: table 10-4 the address window

Base address	End address	The size of the	define
0 x0c00_0000_0000	0 x0cff_ffff_ffff	One Tbytes	HT0_LO window
0 x0d00_0000_0000	0 x0dff_ffff_ffff	One Tbytes	HT0_HI window

distribution of the default 4 HyperTransport interfaces

0 x0e00_0000_0000	0 x0eff_ffff_ffff	One Tbytes	HT1_LO window
0 x0f00_0000_0000	0 x0fff_ffff_ffff	One Tbytes	HT1_HI window

By default (the system address window is not configured separately), the software accesses each HyperTransport interface according to the above address space. In addition, the software can configure the address window on the crossover switch to access it with other address Spaces (see section 2.5). The internal 40-bit address space of each HyperTransport interface has its address window distribution as shown in the following table.

Table 10-5 address window distribution inside the HyperTransport interface of loongson 3 processor

Base address	End address	The size of the	define
0 x00_0000_0000	0 xfc_ffff_ffff	1012 Gbytes	MEM space
0 xfd_0000_0000	0 xfd_f7ff_ffff	3968 Mbytes	reserve
0 xfd_f800_0000	0 xfd_f8ff_ffff	16 Mbytes	interrupt
0 xfd_f900_0000	0 xfd_f90f_ffff	1 Mbyte	PIC interrupt response
0 xfd_f910_0000	0 xfd_f91f_ffff	1 Mbyte	System information
0 xfd_f920_0000	0 xfd_faff_ffff	30 Mbytes	reserve
0 xfd_fb00_0000	0 xfd_fbff_ffff	16 Mbytes	HT controller configuration space
0 xfd_fc00_0000	0 xfd_fdff_ffff	32 Mbytes	I/O space
0 xfd_fe00_0000	0 xfd_ffff_ffff	32 Mbytes	HT bus configuration space
0 xfe_0000_0000	0 xff_ffff_ffff	8 Gbytes	reserve

10.4.2 HyperTransport controller internal window configuration

The HyperTransport interface of loongson 3A2000 processor provides a variety of rich address Windows for users to use. The functions and functions of these address Windows are described in the following table.

Table 10-6 address window provided in loongson 3A2000 processor HyperTransport interface

The address	Window number	Accept the bus	role	note

window				
Receiving window See window configuration 10.5.7)	3	HyperTransport	Judge whether to accept An access emitted on the HyperTransport bus.	In the main bridge mode (i.e., act_as_slave is 0 in the configuration register), only the access that falls into these address Windows will be responded to by the internal bus. Other access will be considered as P2P access and sent back to the HyperTransport bus.When in device mode (that is, act_as_slave is 1 in the configuration register), only the accesses that fall into these address Windows will be received and processed by the internal bus, and other accesses will be returned with protocol errors.
The Post window See window configuration 10.5.11)	2	Inside the bus	Determines whether Write access from the internal bus to the HyperTransport bus is treated as Post Write	Outgoing Write access that falls into these address Spaces will be treated as Post Write. In the Post Write: HyperTransport protocol, this Write access does not have to wait for the Write response, that is, after the controller issues the Write access to the bus, the Write access to the processor completes the response.
Prefetch window See window configuration 10.5.12)	2	Inside the bus	To determine whether to receive internal Cache access, fetch access.	When the processor core is out of order, some guess reads or fetches are given to the bus, which is wrong for some IO Spaces.By default, this access HT controller will return directly without access to the HyperTransport bus.These Windows enable such access to the HyperTransport bus.
Uncache window See window configuration 10.5.13)	2	HyperTransport	Determine if the access on the HyperTransport bus is to be accessed as Uncache on the inside	The IO DMA access inside the loongson 3A2000 processor will be judged as a hit by SCache in the case of Cache access, so as to maintain its IO consistency information. Through the configuration of these Windows, it is possible to make the access hit in these Windows access memory directly in the manner of Uncache, without maintaining its IO consistency information through hardware.

9.5 Configuration register

The configuration register module is mainly used to control the access requests to the configuration register from the AXI SLAVE or HT RECEIVER, perform external interrupt processing, and save a large number of configuration registers visible to the software to control the various working modes of the system.

First, the access and storage of the configuration registers used to control the HT controller's various behaviors are in this module, whose access offset address is 0xFD_FB00_0000 to 0xFD_FBFF_FFFF at the HT controller end. The visible registers of all software in HT controller are shown in the following table:

Table 10-7 software visible register list

offset	The name of the	describe
0 x30		
0 x34		
0 x38		
0 x3c	Bridge Control	Bus Reset Control
0 x40	Capability Registers	Command, Capabilities Pointer, Capability ID
0 x44		Link Config, Link Control
0 x48		Revision ID, Link Freq, Link Error, Link Freq Cap
0 x4c		Feature Capability
0 x50	Custom register	MISC
0 x54	Receive diagnostic register	Used to diagnose a sampled signal at the receiving end
0 x58	Interrupt routing mode selection register	Corresponds to three interrupt routing modes
0 x5c	Receive cache register	
0 x60	Receive address window configuratio n register	HT bus receive address window 0 enable (external access)
0 x64		HT bus receiving address window 0 base address (external access)
0 x68		HT bus receive address window 1 enable (external access)
0 x6c		HT bus receiving address window 1 base address (external access)
0 x70		HT bus receive address window 2 enable (external access)
0 x74		HT bus receiving address window 2 base address (external access)
0 x148		HT bus receive address window 3 enable (external access)
0 x14c		HT bus receive address window 3 base address (external access)
0 x150		HT bus receive address window 4 enable (external access)
0 x154		HT bus receive address window 4 base address (external access)
0 x80	Interrupt vector register	HT bus interrupt vector register [31:0]
0 x84		HT bus interrupt vector register [63:32]
0 x88		HT bus interrupt vector register [95:64]
0 x8c		HT bus interrupt vector register [127:96]
0 x90		HT bus interrupt vector register [159:128]

0 x94		HT bus interrupt vector register [191:160]
0 x98		HT bus interrupt vector register [223:192]
0 x9c		HT bus interrupt vector register [255:224]
0 xa0	Interrupt enable register	HT bus interrupts enable register [31:0]
0 xa4		HT bus interrupt enabled register [63:32]
0 xa8		HT bus interrupt enabled register [95:64]
0 xac		HT bus interrupt enabled register [127:96]
0 xb0		HT bus interrupt enabled register [159:128]
0 xb4		HT bus interrupt enabled register [191:160]
0 xb8		HT bus interrupt enabled register [223:192]
0 XBC		HT bus interrupt enabled register [255:224]
0 xc0	Interrupt Discovery & Configuration	Interrupt Capability
0 xc4		DataPort
0 xc8		IntrInfo [31:0]
0 XCC		IntrInfo [63:32]
0 xd0	Configure the register in the POST address window	HT bus POST address window 0 enable (internal access)
0 xd4		HT bus POST address window 0 base address (internal access)
0 xd8		HT bus POST address window 1 enable (internal access)
0 XDC		HT bus POST address window 1 base address (internal access)
0 xe0-0xfc	Pre-fetch address window configuration register	HT bus prefetching address window 0 enable (internal access)
0 xe4		HT bus can prefetch address window 0 base address (internal access)
0 xe8		HT bus prefetching address window 1 enable (internal access)
0 xec		Ht bus prefetch address window 1 base address (internal access)
0 xf0	Uncache address window configuration register	HT bus Uncache address window 0 enable (external access)
0 xf4		HT bus Uncache address window 0 base address (external access)
0 xf8		HT bus Uncache address window 1 enable (external access)
0 XFC		HT bus Uncache address window 1 base address (external access)
0 x168		HT bus Uncache address window 2 enable (external access)
0 x16c		HT bus Uncache address window 2 base address (external access)
0 x170		HT bus Uncache address window 3 enable (external access)
0 x174		HT bus Uncache address window 3 base address (external access)
0 x158	P2P address window configuration register	HT bus P2P address window 0 enable (external access)
0 x15c		HT bus P2P address window 0 base address (external access)
0 x160		HT bus P2P address window 1 enable (external access)
0 x164		HT bus P2P address window 1 base address (external access)
0 x100	The sending end cache size register	The sending command caches the size register
0 x104		The sending end data cache size register
0 x108	The sending end cache debug register	Used to manually set the size of the sender cache (for debugging)

0 x10c	PHY impedance matching configuration register	Used to configure the impedance matching configuration for the sending and receiving ends of the PHY
0 x110	Revision ID register	Used to configure the controller version
0 x118	Error Retry controls the register	Retry Count Rollover,Short Retry Attempts
0 x11c	The Retry Count register	Used for error retransmission counting in HyerTransport 3.0 mode
0 x130	Link Train register	HyperTransport 3.0 link initialization and link training control
0 x134	Training 0 short count sending over time store	Used for Training 0 short timeout threshold configuration
0 x138	Training 0 timeout long count register	Used for Training 0 long count timeout threshold configuration
0 x13c	Training 1 counting register	Used for Training 1 counting threshold configuration
0 x140	Training 2 counting register	Used for Training 2 counting threshold configuration
0 x144	Training 3 counting register	Used for Training 3 counting threshold configuration
0 x178	Software frequency configuration register	Realize the frequency switch of the controller in the working process
0 x17c	PHY configuration register	Used to configure phy-related physical parameters
0 x180	The link initializes the debug register	For ignoring the PHY CDR lock signal and customizing the wait time
0 x184	LDT debug register	Used to configure the time from the invalidation of the LDT signal to the start of the link initialization

The specific meaning of each register is shown in the following section:

10.5.1 Bridge Control

Offset: 0x3C

Reset value: 0x00000000

Name: Bus Reset Control

Table 10-8 Bus Reset Control register definitions

A domain	A domain name	A wide	Reset value	access	describe
For calamity	Reserved	4	0 x0		reserve
22	The Reset	12	0 x0	R/W	Bus reset control: 0 >1: HT_RSTn set 0, bus reset 1 >0: HT_RSTn set 1, bus unreset
21:0	Reserved	5	0 x0		reserve

10.5.2 Capability Registers

Offset: 0x40

Reset value: 0x20010008

Name: Command, Capabilities Pointer, Capability ID

Table 10-9 Command, Capabilities Pointer, Capability ID register definitions

A domain	A domain name	A wide	Reset value	access	describe
take	The HOST/Sec	3	0 x1	R	The Command format is HOST/Sec
Forepart thereof	Reserved	2	0 x0	R	reserve
26	Act as a Slave	1	0 x0 / 0 x1	R/W	The HOST/SLAVE mode The initial value is determined by the pin HOSTMODE HOSTMODE pull up: 0 HOSTMODE pull down: 1
25	Reserved	1	0 x0		reserve
24	The Host Hide	1	0 x0	R/W	Whether to disable register access from HT bus
23	Reserved	1	0 x0		reserve
"	The Unit ID	5	0 x0	R/W	HOST mode: can be used to record the number of use ids SLAVE mode: record self Unit ID
17	A Double Ended	1	0 x0	R	The dual HOST mode is not used
16	A Warm Reset	1	0 x1	R	The reset in Bridge Control adopts the thermal reset mode
"	"Capabilities Pointer	8	0 xa0	R	The next Cap register offset address
away	Capability ID	8	0 x08	R	HyperTransport capability ID

Offset: 0x44

Reset value: 0x00112000

Name: Link Config, Link Control

Table 10-10 Link Config, Link Control register definition

A domain	A domain name	A wide	Reset value	access	describe

31	ht_phase_select _disable	1	0 x0		Phase selection enable 0: enable phase selection function 1: disable the phase selection function
he	The Link Width Out	3	0 x0	R/W	Sending end width The value after cold reset is the maximum width of the current connection. The value written to this register will take effect after the next hot reset or HT Disconnect 000:8 bit mode 001:16 bit mode
27	Reserved	1	0 x0		reserve
they	The Link Width In	3	0 x0	R/W	Receiver width The value after cold reset is the maximum width of the current connection, write to this post The save value will take effect after the next hot reset or HT Disconnect
23	Dw Fc out	1	0 x0	R	Sending terminal does not support double word flow control
Lift up	Max Link Width out	3	0 x1	R	HT bus sending end maximum width: 16bits
19	Dw Fc In	1	0 x0	R	The receiver does not support double word flow control
thou	Max Link Width In	3	0 x1	R	HT bus receiver maximum width: 16bits
The lowest	Reserved	2	0 x0		reserve
13	LDTSTOP# Tristate Enable	1	0 x1	R/W	Whether to close HT PHY when HT bus enters HT Disconnect state 1: closed 0: not closed
12:10	Reserved	3	0 x0		reserve
9	CRC Error (hi)	1	0 x0	R/W	CRC errors occur in the high 8 bit
8	CRC Error (lo)	1	0 x0	R/W	CRC errors occur in the lower 8 bits
7	Trans off	1	0 x0	R/W	HT PHY close control In 16-bit bus working mode 1: close high/low 8- bit HT PHY 0: enable low 8 HT phys, The high 8-bit HT PHY is controlled by bit 0
6	The End of the Chain	0	0 x0	R	HT bus terminal
5	Init Complete	1	0 x0	R	Whether the HT bus initialization is completed

4	The Link Fail	1	0 x0	R	Indicating connection failure
3:2	Reserved	2	0 x0		reserve
1	CRC Flood Enable	1	0 x0	R/W	Whether flood HT bus occurs when CRC error occurs
0	Trans off (hi)	1	0 x0	R/W	High 8-bit PHY closes control when running 8-bit protocol using 16-bit HT bus 1: close high 8-bit HT PHY 0: enable high 8 bit HT PHY

Offset: 0x48

Reset value: 0x80250023

Name: Revision ID, Link Freq, Link Error, Link Freq Cap

Table 10-11 Revision ID, Link Freq, Link Error, Link Freq Cap register definition

A domain	A domain name	A wide	Reset value	access	describe
Caused the	The Link Freq Cap	16	0 x0025	R	The supported HT bus frequency is generated according to the external PLL Settings Different values
The lowest	Reserved	2	0 x0		reserve
13	Over Flow Error	1	0 x0	R	HT bus packet overflow
12	Protocol Error	1	0 x0	R/W	Protocol error, An unrecognized command received on the HT bus
and	The Link Freq	4	0 x0	R/W	HT bus operating frequency The value of this register will take effect after the next hot reset or HT Disconnect 0000-200 m 0010-400 m 0101-800 m
away	Revision ID	8	0 x23	R/W	Version number: 1.03

Offset: 0x4C

Reset value: 0x00000002

Name: Feature Capability

Table 10-12 Feature Capability register definitions

A domain	A domain name	A wide	Reset value	access	describe

Is wasted	Reserved	25	0 x0		reserve
8	Extended the Register	1	0 x0	R	There is no
The log	Reserved	3	0 x0		reserve
3	Extended CTL Time	1	0 x0	R	Don't need
2	CRC Test Mode	1	0 x0	R	Does not support
1	LDTSTOP#	1	0 x1	R	Support LDTSTOP#
0	Isochronous Mode	1	0 x0	R	Does not support

10.5.3 Custom register

Offset: 0x50

Reset value: 0x00904321

Name: MISC

Table 10-13 MISC register definitions

A domain	A domain name	A wide	Reset value	access	describe
31	Reserved	1	0 x0		reserve
30	Ldt Stop Gen	1	0 x0	R/W	Take the bus into LDT DISCONNECT mode The correct way is: 0 minus >, 1
29	Ldt the Req Gen	1	0 x0	R/W	Waken HT bus from LDT DISCONNECT, set LDT_REQ_n The right way to do it is to put a 0 and then a 1:0 minus >, 1 In addition, direct read and write requests to the bus can also automatically wake up the bus
through out	Interrupt the Index	5	0 x0	R/W	Redirect interrupts other than standard interrupts to which interrupt vector (including SMI, NMI, INIT, INTA, INTB, INTC, INTD) There are 256 interrupt vectors in total. This register represents the interrupt direction The high of the quantity is 5 bits, the internal interrupt vector is as follows: 000: SMI 001: NMI 010: INIT 011: Reservered 100: INTA

					101: intb.br deal 110: INTC 111: INTD

23	Dword Write	1	0 x1	R/W	For 32/64/128/256 bit write access, whether to use Dword Write command format 1: use Dword Write 0: use Byte Write (with MASK)
22	Coherent Mode	1	0 x0	R	Processor consistency mode Determined by pin ICCC_EN
21	Not Care Seqid	1	0 x0	R/W	Whether you don't care about HT
20	The Not Axi2Seqid	1	0 x1	R	Whether to convert the commands on the Axi bus to a different SeqID or not, if not, all read and write commands will take the Fixed ID number in the Fixed SeqID 1: no conversion 0: conversion
He hath	Fixed Seqid	4	0 x0	R/W	Configure the HT bus emitted when Not Axi2Seqid is valid Seqid
"	Priority Nop	4	0 x4	R/W	HT bus Nop flow control packet priority
and	Specify the NPC	4	0 x3	R/W	Non Post channel read and write priority
The log	Priority RC	4	0 x2	R/W	The Response channel reads and writes first
3-0	Priority PC	4	0 x1	R/W	Post channel read and write priority 0x0: highest priority 0xF: lowest priority The priority strategy of increasing with time is adopted for each channel, and this memory set is used to configure the initial priority of each channel

10.5.4 Receive diagnostic register

Offset: 0x54

Reset value: 0x00000000

Name: receive diagnostic register

Table 10-14 receive diagnostic registers

A domain	A domain name	A wide	Reset value	access	describe
0	Sample_en	1	0 x0	R/W	Enable sampling of input cad and CTL 0 x0: ban 0 x1: can make
"	rx_ctl_catch	24	0 x0	R/W	Save the input CTL from the sample (0, 2, 4, 6) corresponds to the four phases of CTL0 sampling (1, 3, 5, 7) corresponds to the four phases of CTL1 sampling
Caused the	rx_cad_phase_0	24	0 x0	R/W	Save the values of the input CAD[15:0] from the sample

10.5.5 Interrupt routing mode selection register

Offset: 0x58

Reset value: 0x00000000

Name: interrupt routing selection register

Table 10-15 interrupt routing selection registers

A domain	A domain name	A wide	Reset value	access	describe
o	ht_int_stripe	2	0 x0	R/W	Corresponding to three interrupt routing methods, see 0 interrupt vector register for details 0x0: ht_int_stripe_1 0x1: ht_int_stripe_2 0 x2: ht_int_stripe_4

10.5.6 Receive buffer initial register

Offset: 0x5c

Reset value: 0x07778888

Name: receive buffer initialization configuration register

Table 10-16 receive buffer initial registers

A domain	A domain name	A wide	Reset value	access	describe
he	rx_buffer_r_data	4	0 x0	R/W	Receive buffer's read data buffer initialization information
Behold,	rx_buffer_npc_data	4	0 x0	R/W	Receive the NPC data buffer initialization information for the buffer
He hath	rx_buffer_pc_data	4	0 x0	R/W	Receive the buffer's PC data buffer initialization information
"	rx_buffer_b_cmd	4	0 x0	R/W	Receive the bresponse command buffer initialization information for the buffer
and	rx_buffer_r_cmd	4	0 x0	R/W	Receive buffer's read command buffer initialization information
The log	rx_buffer_npc_cmd	4	0 x0	R/W	Receive the NPC command buffer initialization information for the buffer
3-0	rx_buffer_pc_cmd	4	0 x0	R/W	Receive the buffer's PC command buffer initialization information

10.5.7 Receive address window configuration register

The address window hit formula in HT controller is as follows:

Hit = (BASE & MASK) == (ADDR & MASK)

Addr_out = TRANS_EN? TRANS b0 ADDR & ~MASK: ADDR

It should be noted that when configuring the address window register, the MASK should be all 1 high and all 0 low. 0 in the MASK

Is the size of the address window.

The address of the receive address window is the address received on the HT bus. The HT address dropped in the P2P window will act as P2P

Commands are forwarded back to the HT bus. The HT addresses that fall in the normal receiving window and are not in the P2P window will be sent to the CPU, and commands from other addresses will be forwarded back to the HT bus as P2P commands.

Offset: 0x60

Reset value: 0x00000000

Name: HT bus receive address window 0 enable (external access)

Table 10-17 HT bus receive address window 0 enable (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
31	ht_rx_image0_en	1	0 x0	R/W	HT bus receives address window 0, enabling signal
30	ht_rx_image0_ trans_en	1	0 x0	R/W	HT bus receives address window 0, mapping enable signal
29:0	ht_rx_image0_ Trans [53:24]	30	0 x0	R/W	HT bus receiving address window 0, the mapped address [53:24]

Offset: 0x64

Reset value: 0x00000000

Name: HT bus receiving address window 0 base address (external access)

Table 10-18 HT bus receive address window 0 base address (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_rx_image0_ The base [they]	16	0 x0	R/W	HT bus receive address window 0, address base address [39:24]
15:0	ht_rx_image0_ Mask [they]	16	0 x0	R/W	HT bus receiving address window 0, address shielded [39:24]

Offset: 0x68

Reset value: 0x00000000

Name: HT bus receive address window 1 enable (external access)

Table 10-19 HT bus receive address window 1 enable (external access) register definitions

A domain	A domain name	A wide	Reset value	access	describe
31	ht_rx_image1_en	1	0 x0	R/W	HT bus receives address window 1, enabling signal

30	ht_rx_image1_ trans_en	1	0 x0	R/W	HT bus receives address window 1, mapping the enable signal
29:0	ht_rx_image1_ Trans [53:24]	30	0 x0	R/W	HT bus receives address window 1, the mapped address [53:24]

Offset: 0x6c

Reset value: 0x00000000

Name: HT bus receiving address window 1 base address (external access)

Table 10-20 HT bus receive address window 1 base address (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_rx_image1_ The base [they]	16	0 x0	R/W	HT bus receive address window 1, address base address [39:24]
15:0	ht_rx_image1_ Mask [they]	16	0 x0	R/W	HT bus receive address window 1, address mask [39:24]

Offset: 0x70

Reset value: 0x00000000

Name: HT bus receive address window 2 enable (external access)

Table 10-21 HT bus receive address window 2 enable (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
31	ht_rx_image2_en	1	0 x0	R/W	HT bus receives address window 2, enabling signal
30	ht_rx_image2_ trans_en	1	0 x0	R/W	The HT bus receives the address window 2, mapping the enable signal
29:0	ht_rx_image2_ Trans [53:24]	16	0 x0	R/W	HT bus receiving address window 2, translated address [53:24]

Offset: 0x74

Reset value: 0x00000000

Name: HT bus receiving address window 2 base address (external access)

Table 10-22 HT bus receive address window 2 base address (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_rx_image2_ The base [they]	16	0 x0	R/W	HT bus receive address window 2, address base address [39:24]
15:0	ht_rx_image2_ Mask [they]	16	0 x0	R/W	HT bus receive address window 2, address mask [39:24]

Offset: 0x148

Reset value: 0x00000000

Name: HT bus receive address window 3 enable (external access)

Table 10-23 HT bus receive address window 3 enable (external access) register definitions

A domain	A domain name	A wide	Reset value	access	describe
31	ht_rx_image3_en	1	0 x0	R/W	HT bus receives address window 3, enabling signal
30	ht_rx_image3_ trans_en	1	0 x0	R/W	HT bus receives address window 3, mapping the enable signal
A domain	A domain name	A wide	Reset value	access	describe
29:0	ht_rx_image3_ Trans [53:24]	16	0 x0	R/W	HT bus receiving address window 3, translated address [53:24]

Offset: 0x14C

Reset value: 0x00000000

Name: HT bus receiving address window 3 base address (external access)

Table 10-24 HT bus receive address window 3 base address (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_rx_image3_ The base [they]	16	0 x0	R/W	HT bus receive address window 3, address base address [39:24]
15:0	ht_rx_image3_ Mask [they]	16	0 x0	R/W	HT bus receive address window 3, address mask [39:24]

Offset: 0x150

Reset value: 0x00000000

Name: HT bus receive address window 4 enable (external access)

Table 10-25 HT bus receive address window 4 enable (external access) register definitions

A domain	A domain name	A wide	Reset value	access	describe
31	ht_rx_image4_en	1	0 x0	R/W	HT bus receives address window 4, enabling signal
30	ht_rx_image4_ trans_en	1	0 x0	R/W	HT bus receives address window 4, mapping enable signal
29:0	ht_rx_image4_ Trans [53:24]	16	0 x0	R/W	HT bus receiving address window 4, translated address [53:24]

Offset: 0x154

Reset value: 0x00000000

Name: HT bus receiving address window 4 base address (external access)

Table 10-26 HT bus receive address window 4 base address (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_rx_image4_ The base [they]	16	0 x0	R/W	HT bus receive address window 4, address base address [39:24]
15:0	ht_rx_image4_ Mask [they]	16	0 x0	R/W	HT bus receive address window 4, address mask [39:24]

10.5.8 Interrupt vector register

Interrupt vector registers 256 in all, except for Fix, Arbiter, and PIC interrupt direct mappings on HT bus

To these 256 interrupt vectors, other interrupts, such as SMI, NMI, INIT, INTA, INTB, INTC, INTD, can be mapped to any 8-bit interrupt vector by register 0x50 [28:24] in the order of {INTD, INTC, INTB, INTA, 1 'b0, INIT, NMI, SMI}. The corresponding value of Interrupt vector is {Interrupt Index, internal vector [2:0]}.

LS3A1000E and above versions, 256 interrupt vectors are mapped to different interrupt lines according to different interrupt routing mode selection register configuration. The specific mapping mode

is as follows:

Ht_int_stripe_1:

[0,1,2,3... 63] corresponds to the interrupted line 0 /HT HI corresponds to the interrupted line 4

[64,65,66,67... 127] corresponding to interrupted line 1 /HT HI corresponding to interrupted line 5

[128,129,130,131... 191] corresponding to discontinuous line 2 /HT HI corresponding to discontinuous line 6

[192,193,194,195... 255] corresponding to interrupted line 3 /HT HI

corresponding to interrupted line 7 ht_int_stripe_2:

[0,2,4,6... 126] corresponds to the interrupted line 0 /HT HI corresponds to the interrupted line 4

[1,3,5,7... 127] corresponds to interrupted line 1 /HT HI corresponds to interrupted line 5

[128,130,132,134... 254] corresponding to discontinuous line 2 /HT HI corresponding to discontinuous line 6

[129,131,133,135... 255] corresponding to discontinuous line 3 /HT

HI corresponding to discontinuous line 7 ht_int_stripe_4:

[0,4,8,12... 252] corresponds to the interrupted line 0 /HT HI corresponds to the interrupted line 4

[1,5,9,13... 253] corresponds to 1 /HT HI corresponds to 5

[2,6,10,14... 254] corresponding to discontinuous line 2 /HT HI corresponding to discontinuous line 6

[3,7,11,15... 255] corresponds to interrupted line 3 /HT HI corresponds to interrupted line 7

The following interrupt vector description corresponds to ht_int_stripe_1, and two other ways can be obtained as described above. For LS3A1000D and below, you can only use ht_int_stripe_1.

Offset: 0x80

Reset value: 0x00000000

Name: HT bus interrupt vector register [31:0]

Table 10-27 HT bus interrupt vector register definition (1)

A domain	A domain name	A wide	Reset value	access	describe

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_case [31:0]	32	0 x0	R/W	HT bus interrupt vector register [31:0], It's 0 over HT HI and it's 4

Offset: 0x84

Reset value: 0x00000000

Name: HT bus interrupt vector register [63:32]

Table 10-28 HT bus interrupt vector register definition (2)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_case [63:32]	32	0 x0	R/W	HT bus interrupt vector register [63:32], It's 0 over HT HI and it's 4

Offset: 0x88

Reset value: 0x00000000

Name: HT bus interrupt vector register [95:64]

Table 10-29 HT bus interrupt vector register definition (3)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_case [95-64]	32	0 x0	R/W	HT bus interrupt vector register [95:64], This corresponds to 1 /HT HI and this corresponds to 5

Offset: 0x8c

Reset value: 0x00000000

Name: HT bus interrupt vector register [127:96]

Table 10-30 HT bus interrupt vector register definition (4)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_case [127-96]	32	0 x0	R/W	HT bus interrupt vector register [127:96], This corresponds to 1 /HT HI and this corresponds to 5

Offset: 0x90

Reset value: 0x00000000

Name: HT bus interrupt vector register [159:128]

Table 10-31 HT bus interrupt vector register definition (5)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_case [159-128]	32	0 x0	R/W	HT bus interrupt vector register [159:128], So this corresponds to 2 /HT HI and this corresponds to 6

Offset: 0x94

Reset value: 0x00000000

Name: HT bus interrupt vector register [191:160]

Table 10-31 HT bus interrupt vector register definition (6)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_case [191-160]	32	0 x0	R/W	HT bus interrupt vector register [191:160], So this corresponds to 2 /HT HI and this corresponds to 6

Offset: 0x98

Reset value: 0x00000000

Name: HT bus interrupt vector register [223:192]

Table 10-32 HT bus interrupt vector register definition (7)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_case [223-192]	32	0 x0	R/W	HT bus interrupt vector register [223:192], So this corresponds to 3 /HT HI and this corresponds to 7

Offset: 0x9c

Reset value: 0x00000000

Name: HT bus interrupt vector register [255:224]

Table 10-33 HT bus interrupt vector register definition (8)

A domain	A domain name	A wide	Reset value	access	describe

31:0	Interrupt_case [255-224]	32	0 x0	R/W	HT bus interrupt vector register [255:224], So this corresponds to 3 /HT HI and this corresponds to 7

10.5.9 Interrupt enable register

There are 256 interrupt enabled registers, corresponding to the interrupt vector register. Set 1 to open the corresponding interrupt, set 0

Then it is interrupt shielding.

The 256 interrupt vectors are mapped to different interrupt lines according to the interrupt routing mode and register configuration. The specific mapping mode is as follows:

Ht_int_stripe_1:

[0,1,2,3... 63] corresponds to the interrupted line 0 /HT HI corresponds to the interrupted line 4

[64,65,66,67... 127] corresponding to interrupted line 1 /HT HI corresponding to interrupted line 5

[128,129,130,131... 191] corresponding to discontinuous line 2 /HT HI corresponding to discontinuous line 6

[192,193,194,195... 255] corresponds to interrupted line 3 /HT HI corresponds to interrupted line 7

Ht_int_stripe_2:

[0,2,4,6... 126] corresponds to the interrupted line 0 /HT HI corresponds to the interrupted line 4

[1,3,5,7... 127] corresponds to interrupted line 1 /HT HI corresponds to interrupted line 5

[128,130,132,134... 254] corresponding to discontinuous line 2 /HT HI corresponding to discontinuous line 6

[129,131,133,135... 255] corresponding to discontinuous line 3 /HT

HI corresponding to discontinuous line 7 ht_int_stripe_4:

[0,4,8,12... 252] corresponds to the interrupted line 0 /HT HI corresponds to the interrupted line 4

[1,5,9,13... 253] corresponds to 1 /HT HI corresponds to 5

[2,6,10,14... 254] corresponding to discontinuous line 2 /HT HI corresponding to discontinuous line 6

[3,7,11,15... 255] corresponds to interrupted line 3 /HT HI corresponds to interrupted line 7

The following interrupt vector description corresponds to ht_int_stripe_1, and two other ways can be obtained as described above.

Offset: 0xa0
Reset value: 0x00000000

Name: HT bus interrupt enabled register [31:0]

Table 10-34 HT bus interrupt enabled register definition (1)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_mask [31:0]	32	0 x0	R/W	HT bus interrupts enable register [31:0], It's 0 over HT HI and it's 4

Offset: 0xa4
Reset value: 0x00000000
Name: HT bus interrupt enabled register [63:32]

Table 10-35 HT bus interrupt enabled register definition (2)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_mask [63:32]	32	0 x0	R/W	HT bus interrupts enable register [63:32], It's 0 over HT HI and it's 4

Offset: 0xa8
Reset value: 0x00000000
Name: HT bus interrupt enabled register [95:64]

Table 10-36 HT bus interrupt enabled register definition (3)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_mask [95-64]	32	0 x0	R/W	HT bus interrupts enable register [95:64], This corresponds to 1 /HT HI and this corresponds to 5

Offset: 0xac

Reset value: 0x00000000

Name: HT bus interrupt enabled register [127:96]

Table 10-37 HT bus interrupt enabled register definition (4)

A domain	A domain name	A wide	Reset value	access	describe

31:0	Interrupt_mask [127-96]	32	0 x0	R/W	HT bus interrupts enable register [127:96], This corresponds to 1 /HT HI and this corresponds to 5

Offset: 0xb0

Reset value: 0x00000000

Name: HT bus interrupt enabled register [159:128]

Table 10-38 HT bus interrupt enabled register definition (5)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_mask [159-128]	32	0 x0	R/W	HT bus interrupts enable register [159:128], So this corresponds to 2 /HT HI and this corresponds to 6

Offset: 0xb4

Reset value: 0x00000000

Name: HT bus interrupt enabled register [191:160]

Table 10-39 HT bus interrupt enabled register definition (6)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_mask [191-160]	32	0 x0	R/W	HT bus interrupts enable register [191:160], So this corresponds to 2 /HT HI and this corresponds to 6

Offset: 0xb8

Reset value: 0x00000000

Name: HT bus interrupt enabled register [223:192]

Table 10-40 HT bus interrupt enabled register definition (7)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_mask [223-192]	32	0 x0	R/W	HT bus interrupts enable register [223:192], So this corresponds to 3 /HT HI and this corresponds to 7

Offset: 0xbc

Reset value: 0x00000000

Name: HT bus interrupt enabled register [255:224]

Table 10-41 HT bus interrupt enabled register definition (8)

A domain	A domain name	A wide	Reset value	access	describe
31:0	Interrupt_mask [255-224]	32	0 x0	R/W	HT bus interrupts enable register [255:224], So this corresponds to 3 /HT HI and this corresponds to 7

10.5.10 Interrupt Discovery & Configuration

Offset: 0xc0

Reset value: 0x80000008

Name: Interrupt Capability

Table 10-42 definitions of the Interrupt Capability register

A domain	A domain name	A wide	Reset value	access	describe
came	"Capabilities Pointer	8	0 x80	R	Interrupt discovery and configuration block
Ephron;	The Index	8	0 x0	R/W	Read register offset address
"	"Capabilities Pointer	8	0 x0	R	"Capabilities Pointer
away	Capability ID	8	0 x08	R	Hypertransport Capablity ID

Offset: 0xc4

Reset value: 0x00000000

Name: Dataport

Table 10-43 Dataport register definitions

A domain	A domain name	A wide	Reset value	access	describe
31:0	Dataport	32	0 x0	R/W	When the Index of the previous register is 0x10, the read-write result of this register is 0xa8 register, otherwise it is 0xac

Offset: 0xc8

Reset value: 0xF8000000

Name: IntrInfo [31:0]

Table 10-44 IntrInfo register definitions (1)

A domain	A domain name	A wide	Reset value	access	describe
came	IntrInfo [came]	32	0 xf8	R	reserve

A domain	A domain name	A wide	Reset value	access	describe
Isle;	IntrInfo [great]	22	0 x0	R/W	IntrInfo[23:2], the value of IntrInfo when sending PIC interrupt It's used to represent interrupt vectors
1-0	Reserved	2	0 x0	R	reserve

Offset: 0xcc

Reset value: 0x00000000

Name: IntrInfo [63:32]

Table 10-45 IntrInfo register definitions (2)

A domain	A domain name	A wide	Reset value	access	describe
31:0	IntrInfo [63:32]	32	0 x0	R	reserve

10.5.11 Configure the register in the POST address window

The address window hit formula is shown in section 10.5.7.

The address of this window is the one received on the AXI bus. All WRITE accesses that fall into this window will immediately be returned in the AXI B channel and sent to the HT bus in the command format of POST WRITE. WRITE requests that are not in this window are sent to the HT bus as NONPOST WRITE and wait for the HT bus to respond before returning to the AXI bus.

Offset: 0xd0

Reset value: 0x00000000

Name: HT bus POST address window 0 enable (internal access)

Table 10-46 HT bus POST address window 0 enable (internal access)

A domain	A domain name	A wide	Reset value	access	describe
31	ht_post0_en	1	0 x0	R/W	HT bus POST address window 0, enable signal
30	ht_depart0_en	1	0 x0	R/W	HT access unpack enable (corresponding to the external of the CPU core Uncache ACC operation window)
throne	Reserved	14	0 x0		reserve
15:0	ht_post0_trans [they]	16	0 x0	R/W	HT bus POST address window 0, translated address [39:24]

Offset: 0xd4

Reset value: 0x00000000

Name: HT bus POST address window 0 base address (internal access)

Table 10-47 HT bus POST address window 0 base address (internal access)

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_post0_base [they]	16	0 x0	R/W	HT bus POST address window 0, address base address [39:24]
15:0	ht_post0_mask [they]	16	0 x0	R/W	HT bus POST address window 0, address masked [39:24]

Offset: 0xd8

Reset value: 0x00000000

Name: HT bus POST address window 1 enable (internal access)

Table 10-48 HT bus POST address window 1 enable (internal access)

A domain	A domain name	A wide	Reset value	access	describe
31	ht_post1_en	1	0 x0	R/W	HT bus POST address window 1, enable signal
30	ht_depart1_en	1	0 x0	R/W	HT access unpack enable (corresponding to the external of the CPU core Uncache ACC operation window)
Was a	Reserved	14	0 x0		reserve
15:0	ht_post1_trans [they]	16	0 x0	R/W	HT bus POST address window 1, translated address [39:24]

Offset: 0xdc

Reset value: 0x00000000

Name: HT bus POST address window 1 base address (internal access)

Table 10-49 HT bus POST address window 1 base address (internal access)

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_post1_base [they]	16	0 x0	R/W	HT bus POST address window 1, address base address [39:24]
15:0	ht_post1_mask [they]	16	0 x0	R/W	HT bus POST address window 1, address masked [39:24]

10.5.12 Pre-fetch address window configuration register

The address window hit formula is shown in section 10.5.7.

The address of this window is the one received on the AXI bus. Other fetches or CACHE accesses will not be sent to the HT bus, but will be returned immediately. If it is a read command, invalid read data of the corresponding number will be returned.

Offset: 0xe0

Reset value: 0x00000000

Name: HT bus prefetching address window 0 enable (internal access)

Table 10-50 HT bus prefetching address window 0 enable (internal access)

A domain	A domain name	A wide	Reset value	access	describe
31	ht_prefetch0_en	1	0 x0	R/W	HT bus can prefetch address window 0, enable signal
then	Reserved	15	0 x0		reserve
15:0	ht_prefetch0_trans [they]	16	0 x0	R/W	HT bus prefetch address window 0, translated address [39:24]

Offset: 0xe4

Reset value: 0x00000000

Name: HT bus prefetch address window 0 base address (internal access)

Table 10-51 HT bus prefetch address window 0 base address (internal access)

A domain	A domain name	A wide	Reset value	access	describe
Caused the	Ht_prefetch0_ base [they]	16	0 x0	R/W	HT bus can prefetch address window 0, address base address [39:24] An address
15:0	ht_prefetch0_ Mask [they]	16	0 x0	R/W	HT bus can prefetch address window 0, address mask [39:24]

Offset: 0xe8

Reset value: 0x00000000

Name: HT bus prefetching address window 1 enable (internal access)

Table 10-52 HT bus prefetching address window 1 enable (internal access)

A domain	A domain name	A wide	Reset value	access	describe
31	ht_prefetch1_en	1	0 x0	R/W	HT bus can prefetch address window 1 to enable the signal
then	Reserved	15	0 x0		reserve
15:0	ht_prefetch1_ Trans. [they]	16	0 x0	R/W	HT bus prefetch address window 1, translated address [39:24]

Offset: 0xec

Reset value: 0x00000000

Name: HT bus prefetch address window 1 base address (internal access)

Table 10-53 HT bus prefetch address window 1 base address (internal access)

A domain	A domain name	A wide	Reset value	access	describe
A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_prefetch1_ The base [they]	16	0 x0	R/W	HT bus can prefetch address window 1, address base address [39:24]
15:0	ht_prefetch1_ Mask [they]	16	0 x0	R/W	HT bus can prefetch address window 1, address mask [39:24]

10.5.13 UNCACHE address window configuration register

The address window hit formula is shown in section 10.5.7.

The address of this window is the address received on the HT bus. The read/write command that falls at the address of this window will not be sent to SCACHE, nor will it invalidate the level 1 CACHE, but will be sent directly to memory or other address space, that is, the read/write command in this address window will not maintain the CACHE consistency of IO. This window focuses on operations that are not hit in the CACHE and therefore can improve memory efficiency, such as access to video memory.

Offset: 0xf0

Reset value: 0x00000000

Name: HT bus Uncache address window 0 enable (internal access)

Table 10-54 HT bus Uncache address window 0 enable (internal access)

A domain	A domain name	A wide	Reset value	access	describe
31	ht_uncache0_en	1	0 x0	R/W	HT bus uncache address window 0, enable signal
30	ht_uncache0_ trans_en	1	0 x0	R/W	HT bus uncache address window 1, mapping enable signal
29:0	Ht_uncache0_ trans [53:24]	16	0 x0	R/W	HT bus uncache address window 0, translated address [53:24]

Offset: 0xf4

Reset value: 0x00000000

Name: HT bus Uncache address window 0 base address (internal access)

Table 10-55 HT bus Uncache address window 0 base address (internal access)

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_uncache0_ The base [they]	16	0 x0	R/W	HT bus uncache address window 0, address base address [39:24]
15:0	ht_uncache0_ Mask [they]	16	0 x0	R/W	HT bus uncache address window 0, address mask [39:24]

Offset: 0xf8

Reset value: 0x00000000

Name: HT bus Uncache address window 1 enable (internal access)

Table 10-56 HT bus Uncache address window 1 enable (internal access)

A domain	A domain name	A wide	Reset value	access	describe
31	ht_uncache1_en	1	0 x0	R/W	HT bus uncache address window 1, enable the signal
30	ht_uncache1_ trans_en	1	0 x0	R/W	HT bus uncache address window 1, mapping enable signal
29:0	Ht_uncache1_ trans [53:24]	16	0 x0	R/W	HT bus uncache address window 1, translated address [53:24]

Offset: 0xfc

Reset value: 0x00000000

Name: HT bus Uncache address window 1 base address (internal access)

Table 10-57 HT bus Uncache address window 1 base address (internal access)

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_uncache1_ The base [they]	16	0 x0	R/W	HT bus uncache address window 1, address base address [39:24]
15:0	ht_uncache1_ Mask [they]	16	0 x0	R/W	HT bus uncache address window 1, address mask [39:24]

Offset: 0x168

Reset value: 0x00000000

Name: HT bus Uncache address window 2 enable (internal access)

Table 10-58 HT bus Uncache address window 2 enable (internal access)

A domain	A domain name	A wide	Reset value	access	describe
31	ht_uncache1_en	1	0 x0	R/W	HT bus uncache address window 2, enable the signal
30	Ht_uncache1_ trans_en	1	0 x0	R/W	HT bus uncache address window 2, mapping enable signal
29:0	ht_uncache1_ Trans [53:24]	16	0 x0	R/W	HT bus uncache address window 2, after translation of the address [53:24]

Offset: 0x16c

Reset value: 0x00000000

Name: HT bus Uncache address window 2 base address (internal access)

Table 10-59 HT bus Uncache address window 2 base address (internal access)

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_uncache1_ The base [they]	16	0 x0	R/W	HT bus uncache address window 2, address base address [39:24]
15:0	ht_uncache1_ Mask [they]	16	0 x0	R/W	HT bus uncache address window 2, address mask [39:24]

Offset: 0x170

Reset value: 0x00000000

Name: HT bus Uncache address window 3 enable (internal access)

Table 10-60 HT bus Uncache address window 3 enable (internal access)

A domain	A domain name	A wide	Reset value	access	describe
31	ht_uncache1_en	1	0 x0	R/W	HT bus uncache address window 3, enable the signal
30	ht_uncache1_ trans_en	1	0 x0	R/W	HT bus uncache address window 3, mapping enable signal

29:0	Ht_uncache1_ trans [53:24]	16	0 x0	R/W	HT bus uncache address window 3, after translation of the address [53:24]

Offset: 0x174

Reset value: 0x00000000

Name: HT bus Uncache address window 3 base address (internal access)

Table 10-61 HT bus Uncache address window 3 base addresses (internal access)

A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_uncache1_ The base [they]	16	0 x0	R/W	HT bus uncache address window 3, address base address [39:24]
15:0	ht_uncache1_ Mask [they]	16	0 x0	R/W	HT bus uncache address window 3, address mask [39:24]

10.5.14 P2P address window configuration register

The address window hit formula is shown in section 10.5.7.

The address of this window is the address received on the HT bus. Read and write commands that fall on the address of this window, directly as P2P

The command is forwarded back to the bus, which has the highest priority relative to the normal receive window and Uncache window.

Offset: 0x158

Reset value: 0x00000000

Name: HT bus P2P address window 0 enable (external access)

Table 10-62 HT bus P2P address window 0 enable (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
31	ht_p2p_image0_en	1	0 x0	R/W	HT bus P2P address window 0, enable signal

30	ht_p2p_image0_ trans_en	1	0 x0	R/W	HT bus P2P address window 0, mapping enable signal
29:0	ht_p2p_image0_ Trans [53:24]	16	0 x0	R/W	HT bus P2P address window 0, the translated address [53:24]

Offset: 0x15c

Reset value: 0x00000000

Name: HT bus P2P address window 0 base address (external access)

Table 10-63 HT bus P2P address window 0 base address (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
Caused the	Ht_p2p_image0_ base [they]	16	0 x0	R/W	HT bus P2P address window 0, address base address [39:24]
15:0	ht_p2p_image0_ Mask [they]	16	0 x0	R/W	HT bus P2P address window 0, address shielding [39:24]

Offset: 0x160

Reset value: 0x00000000

Name: HT bus P2P address window 1 enable (external access)

Table 10-64 HT bus P2P address window 1 enable (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
31	ht_p2p_image1_en	1	0 x0	R/W	HT bus P2P address window 1, enable signal
30	Ht_p2p_image1_ trans_en	1	0 x0	R/W	HT bus P2P address window 1, mapping enable signal
29:0	ht_p2p_image1_ Trans [53:24]	16	0 x0	R/W	HT bus P2P address window 1, the translated address [53:24]

Offset: 0x164

Reset value: 0x00000000

Name: HT bus P2P address window 1 base address (external access)

Table 10-65 HT bus P2P address window 1 base address (external access) register definition

A domain	A domain name	A wide	Reset value	access	describe
A domain	A domain name	A wide	Reset value	access	describe
Caused the	ht_p2p_image1_ The base [they]	16	0 x0	R/W	HT bus P2P address window 1, address base address [39:24]
15:0	Ht_p2p_image1_ mask [they]	16	0 x0	R/W	HT bus P2P address window 1, address shielded [39:24]

10.5.15 The command sends the cache size register

The command send cache size register is used to measure the number of caches available for each command channel at the sending end.

Offset: 0x100

Reset value: 0x00000000

Name: command sends cache size register

Table 10-66 sends the cache size register

A domain	A domain name	A wide	Reset value	access	describe
came	B_CMD_txbuffer	8	0 x0	R	Number of B channel command caches at sending end
2316	R_CMD_txbuffer	8	0 x0	R	Number of R channel command caches at sending end
"	NPC_CMD_txbuffer	8	0 x0	R	Number of sending end NPC channel command cache
away	PC_CMD_txbuffer	8	0 x0	R	Number of PC channel command caches at sending end

10.5.16 Data send cache size register

The data send cache size register is used to measure the number of caches available for each data channel at the sending end.

Offset: 0x104

Reset value: 0x00000000

Name: data send cache size register

Table 10-67 data sending cache size registers

A domain	A domain name	A wide	Reset value	access	describe
came	Reserved	8	0 x0	R	reserve
2316	R_DATA_txbuffer	8	0 x0	R	The number of R channel data caches at the sending end
"	NPC_DATA_txbuffer	8	0 x0	R	The number of data caches of sending end NPC channel
away	PC_DATA_txbuffer	8	0 x0	R	The number of data caches of sending PC channel

10.5.17 Send the cache debug register

The send cache debug register is used to manually set the number of buffers at the sending end of the HT controller by increasing or decreasing

Adjust the number of different send caches.

Offset: 0x108

Reset value: 0x00000000

Name: send cache debug register

Tables 10-68 send cache debug registers

A domain	A domain name	A wide	Reset value	access	describe
charm	Reserved	2	0 x0	R	reserve
29	Tx_neg	1	0 x0	R/W	Send side cache debug symbol 0: increase the number 1: reduce (the number of corresponding registers +1)
28	Tx_buff_adj_en	1	0 x0	R/W	The sending side cache debug enable register 0->1: causes the value of this register to increase or decrease
he	R_DATA_txadj	4	0 x0	R/W	The number of increase or decrease of R channel data cache at the sending end When tx_neg is 0, increase R_DATA_txadj; When tx_neg is 1, reduce R_DATA_txadj+1
Behold,	NPC_DATA_txadj	4	0 x0	R/W	Sending end NPC channel data cache increase or decrease When tx_neg is 0, increase

					NPC_DATA_txadj; When tx_neg is 1, reduce NPC_DATA_txadj+1
He hath	PC_DATA_txadj	4	0 x0	R/W	The number of data cache increase or decrease of PC channel at the sending end When tx_neg is 0, increase PC_DATA_txadj; When tx_neg is 1, reduce PC_DATA_txadj+1
"	B_CMD_txadj	4	0 x0	R/W	Send end B channel command cache increase or decrease When tx_neg is 0, add B_CMD_txadj; When tx_neg is 1, reduce B_CMD_txadj+1
and	R_CMD_txadj	4	0 x0	R/W	Send R channel command cache increase or decrease When tx_neg is 0, add R_CMD_txadj; When tx_neg is 1, reduce R_CMD_txadj+1
The log	NPC_CMD_txadj	4	0 x0	R/W	Sending end NPC channel command/data cache increase or decrease When tx_neg is 0, increase NPC_CMD_txadj; When tx_neg is 1, reduce NPC_CMD_txadj+1
3-0	PC_CMD_txadj	4	0 x0	R/W	Send PC channel command cache increase or decrease When tx_neg is 0, increase PC_CMD_txadj; When tx_neg is 1, reduce PC_CMD_txadj+1

10.5.18 PHY impedance matching control register

Used to control the impedance matching enabling of PHY, the sending and receiving impedance matching parameters are set to offset: 0x10C

Reset value: 0x00000000

Name: PHY impedance matching control register

Table 10-69 impedance matching control registers

A domain	A domain name	A wide	Reset value	access	describe
31	Tx_scanin_en	1	0 x0	R/W	TX impedance matching enabled
30	Rx_scanin_en	1	0 x0	R/W	RX impedance matching enabled
he	Tx_scanin_ncode	4	0 x0	R/W	TX impedance matching scan input ncode
Behold,	Tx_scanin_pcode	4	0 x0	R/W	TX impedance matching scan input pcode
then	Rx_scanin_code	8	0 x0	R/W	RX impedance matching scan input

10.5.19 Revision ID register

Used to configure the controller version, configured to a new version number, with a Warm Reset effect.

Offset: 0x110

Reset value: 0x00200000

Name: RevisionID register

Table 10-70 Revision ID register

A domain	A domain name	A wide	Reset value	access	describe
came	Reserved	8	0 x0	R	reserve
Ephron;	Revision ID	8	0 x20	R/W	Revision ID control register 0x20: HyperTransport 1.00 0 x60: HyperTransport 3.00
15:0	Reserved	16	0 x0	R	reserve

10.5.20 Error Retry controls the register

For error retransmission enablers in HyerTransport 3.0 mode, configure the maximum number of Short Retry, display

Whether the Retry counter is flipped.Offset: 0x118

Reset value: 0x00000000

Name: Error Retry control register

Table 10-71 Error Retry control register

A domain	A domain name	A wide	Reset value	access	describe
for	Reserved	22	0 x0	R	reserve
9	Retry Count Rollover	1	0 x0	R	Retry counter count flip
8	Reserved	1	0 x0	R	reserve
but	Short Retry Attempts	2	0 x0	R/W	The maximum number of Short Retry allowed

10.5.21 The Retry Count register

Used for error retransmission counting in HyerTransport 3.0 mode.Offset: 0x11C

Reset value: 0x00000000

Name: Retry Count register

Table 10-72 Retry Count register

A domain	A domain name	A wide	Reset value	access	describe
conspiracies	Reserved	12	0 x0	R	reserve
He hath	Rrequest delay	4	0 x0	R/W	Used to control the random delay range of Rrequest transmissions in consistent mode 000:0 delay 001: random delay 0-8 010: random delay of 8-15 011: random delay 16-31 100: random delay 32-63 101: random delay 64-127 110: random delay 128-255 111:0 delay
15:0	Retry Count	16	0 x0	R	Retry count

10.5.22 Link Train register

HyperTransport 3.0 link initialization and link training control registers.Offset: 0x130

Reset value: 0x00000070

Name: Link Train register

Table 10-73 Link Train register

A domain	A domain name	A wide	Reset value	access	describe
For calamity	Reserved	9	0 x0	R	reserve
"You	Transmitter LS the select	2	0 x0	R/W	Link state for the sender in Disconnected or Inactive state: 2 'b00 LS1 2 'b01 LS0 2 'b10 LS2 2 'bl1 you
14	DsiableCmd Throttling	1	0 x0	R/W	In HyperTransport 3.0 mode, only one non-info CMD can appear in any 4 consecutive DWS by default. 1 'b0 enables Cmd Throttling

					1 'b1 Cmd Throttling is prohibited
"	Reserved	4	0 x0	R	reserve
"	Receiver LS the select	2	0 x0	R/W	Link state for the receiver in Disconnected or Inactive state: 2 'b00 LS1
					2 'b01 LS0 2 'b10 LS2 2 'bl1 you
6:4	Long Retry Count	3	0 x7	R/W	The maximum number of Long Retry
3	Scrambling the Enable	1	0 x0	R/W	Do you make it possible to Scramble 1: can Scramble
2	8 b10b Enable	1	0 x0	R/W	Enable or disable 8B10B 0: disable 8B10B 1: can make 8 b10b
1	AC	1	0 x0	R	Is AC mode detected 0: no AC mode detected 1: AC mode has been detected
0	Reserved	1	0 x0	R	reserve

10.5.23 Training 0 timeout short timing register

Used to configure Training 0 short timeouts timeout threshold in HyerTransport 3.0 mode, and the counter clock frequency is

HyperTransport3.0 link bus clock frequency 1/4.Offset: 0x134

Reset value: 0x00000080

Name: Training 0 timeout short count register

Table 10-74 Training 0 timeout short timing register

A domain	A domain name	A wide	Reset value	access	describe
31:0	T0 time	32	By 8 0	R/W	Training 0 timeout short timing register

10.5.24 Training 0 timeout long timing register

Used for Training 0 long count timeout threshold in HyerTransport 3.0 mode, the counter clock frequency is

HyperTransport3.0 link bus clock frequency 1/4.Offset: 0x138

Reset value: 0x000fffff

Name: Training 0 timeout long count register

Table 10-75 Training 0 timeout long count register

A domain	A domain name	A wide	Reset value	access	describe
31:0	T0 time	32	0 XFFFFF	R/W	Training 0 timeout long count register

10.5.25 Training 1 counting register

For Training 1 counting threshold in HyerTransport 3.0 mode, the counter clock frequency is

HyperTransport3.0 link bus clock frequency 1/4.

Offset: 0x13C

Reset value: 0x0004fffff

Name: Training 1 counting register

Table 10-76 Training 1 counting register

A domain	A domain name	A wide	Reset value	access	describe
31:0	T1 time	32	0 x4fffff	R/W	Training 1 counting register

10.5.26 Training 2 counting register

For Training 2 counting threshold in HyerTransport 3.0 mode, the counter clock frequency is

HyperTransport3.0 link bus clock frequency 1/4.Offset: 0x144

Reset value: 0x0007fffff

Name: Training 2 counting register

Table 10-77 Training 2 counting registers

A domain	A domain name	A wide	Reset value	access	describe
31:0	T2 time	32	0 x7fffff	R/W	Training 2 counting register

10.5.27 Training 3 counting register

For Training 3 counting threshold in HyerTransport 3.0 mode, the counter clock frequency is

HyperTransport3.0 link bus clock frequency 1/4.Offset: 0x13C

Name: Training 3 counting register

Table 10-78 Training 3 counting registers

A domain	A domain name	A wide	Reset value	access	describe
31:0	T3 time	32	0 x7fffff	R/W	Training 3 counting register

10.5.28 Software frequency configuration register

In the case of CLKSEL[15] pulling down, it is used to realize the controller switching to any protocol and PLL supported link frequency and controller frequency during the working process. In the case of CLKSEL[15] pulling high, it has no effect.

The specific switching method is: on the premise of enabling software configuration mode, set the software frequency configuration register bit 1,

And write the new clock-related parameters, including div_refc and div_loop that determine the PLL output frequency, the frequency division coefficients of phy_hi_div and phy_lo_div on the link, and the frequency division coefficient of the controller, core_div. After entering warm reset or LDT disconnect, the controller will automatically reset PLL and configure the new clock parameters.

The calculation formula of clock frequency is:

HyperTransport 1.0:

PHY_LINK_CLK = 50MHz×div_loop /div_refc /phy_div

HT_CORE_CLK=100MHz×div_loop /div_refc /core_div

HyperTransport 3.0:

PHY_LINK_CLK =100MHz×div_loop /div_refc ht_clk

=100MHz×div_loop /div_refc /core_div

The time to wait for PLL relock is about 30us when system CLK is 33M by default. You can also write custom wait count caps in registers.

Offset: 0x178

Reset value: 0x00000000

Name: software frequency configuration register

Table 10-79 software frequency configuration registers

A domain	A domain name	A wide	Reset value	access	describe
behold	PLL relock counter	5	0 x0	R/W	Counter upper limit configuration register When set to counter select, the upper limit of the counter count is {PLL_relock_counter, 5 'h1f} Otherwise the count is capped at 10 '3ff
26	Counter the select	1	0 x0	R/W	Lock timer custom enable: 1 'b0 USES the default count upper limit; 1 'b1 is calculated from PLL_relock_counter
Struggled together	Soft_phy_lo_div	4	0 x0	R/W	High PHY frequency division coefficient
Lift up	Soft_phy_hi_div	4	0 x0	R/W	Low frequency division coefficient
"	Soft_div_refc	2	0 x0	R/W	PLL internal frequency division coefficient
Put no	Soft_div_loop	7	0 x0	R/W	PLL internal frequency multiplication coefficient
then	Soft_core_div	4	0 x0	R/W	Controller clock frequency division coefficient
4-2	Reserved	3	0 x0	R	reserve
1	Soft cofig enable	1	0 x0	R/W	Software configuration enable bit 1 'b0 disables software frequency configuration 1 'b1 enables software frequency configuration
0	Reserved	1	0 x0	R	reserve

10.5.29 PHY configuration register

For the configuration of phy-related physical parameters, when the controller ACTS as two independent 8bit controllers, the higher

The PHY and the low-lying PHY are controlled by two controllers independently. When the controller ACTS as a 16bit controller, the configuration parameters of high-bit and low-bit PHY are uniformly controlled by the low-bit controller.

Offset: 0x17C

Reset value: 0x83308000

Name: PHY configuration register

Table 10-80 PHY configuration registers

A domain	A domain name	A wide	Reset value	access	describe

31	Rx_ckpll_term	1	0 x1	R/W	The terminal impedance of the transmission line from PLL to RX
30	Tx_ckpll_term	1	0 x0	R/W	The terminal impedance of the transmission line from PLL to TX
29	Rx_clk_in_sel_	1	0 x0	R/W	The clock PAD supplies the clock selection of the data PAD, which is automatically selected as CLKPAD in HT1 mode: 1 'b0 external clock source 1 'b1 PLL clock
28	Rx_ckdll_sell	1	0 x0	R/W	Clock selection for locking DLL: 1 'b0 PLL clock 1 'b1 external clock source
But after	Rx_ctle_bitc	2	0 x0	R/W	PAD EQD high frequency gain
Thus for	Rx_ctle_bitr	2	0 x3	R/W	PAD EQD low frequency gain
"	Rx_ctle_bitlim	2	0 x0	R/W	PAD EQD compensation limitation
21	Rx_en_ldo	1	0 x1	R/W	They control 1 'b0 "disabled 1 'b1 can they make
20	Rx_en_by	1	0 x1	R/W	BandGap control 1 'b0 BandGap disabled 1' b1 BandGap enabled
michal	Reserved	3	0 x0	R	reserve
then	Tx_preenmp	5	0 x08	R/W	PAD preload control signal
11:0	Reserved	12	0 x0	R	reserve

10.5.30 The link initializes the debug register

It is used to configure whether to use the CDR lock signal provided by PHY as the symbol to complete the link CDR during the link initialization in HyperTransport 3.0 mode. If the lock signal is ignored, the controller counts and waits for a certain amount of time before the default CDR completes.

Offset: 0x180

Reset value: 0x00000000

Name: link initialization debug register

Table 10-81 link initializes debug registers

A domain	A domain name	A wide	Reset value	access	describe
15	Cdr_ignore_enable	1	0 x0	R/W	Whether CRC lock is ignored when the link is initialized, wait for completion through counter counting: 1 'b0 waits for CDR lock

					1 'b1 ignores the CDR lock signal and accumulates and waits through the counter
14:0	Cdr_wait_counter	15	0 x0	R/W	Wait for the counter to count the upper limit, based on the controller clock to complete the technique

10.5.31 LDT debug register

When the software changes the frequency of the controller, the timing of the LDT reconnect phase will not be accurate, so the count needs to be configured

The time between the LDT signal being invalid as a software configuration frequency and the controller starting the link initialization is based on the controller clock.

Offset: 0x184

Reset value: 0x00000000

Name: LDT debug register

Table 10-82 LDT debug registers

A domain	A domain name	A wide	Reset value	access	describe
Caused the	Rx_wait_time	16	0 x0	R/W	The RX terminal waits for the initial value of the counter
15:0	Tx_wait_time	16	0 x0	R/W	The TX end waits for the initial value of the counter

9.6 The HyperTransport bus conpoints the access methods to the space

The protocol of the HyperTransport interface software layer is basically the same as that of PCI. Since the access of the configuration space is directly related to the underlying protocol, the specific access details are slightly different.As listed in table 10-5, the address range of the HT bus configuration space is 0xFD_FE00_0000 ~ 0xFD_FFFF_FFFF. For configuration access in HT protocol, the following format is adopted in loongson 3A2000:

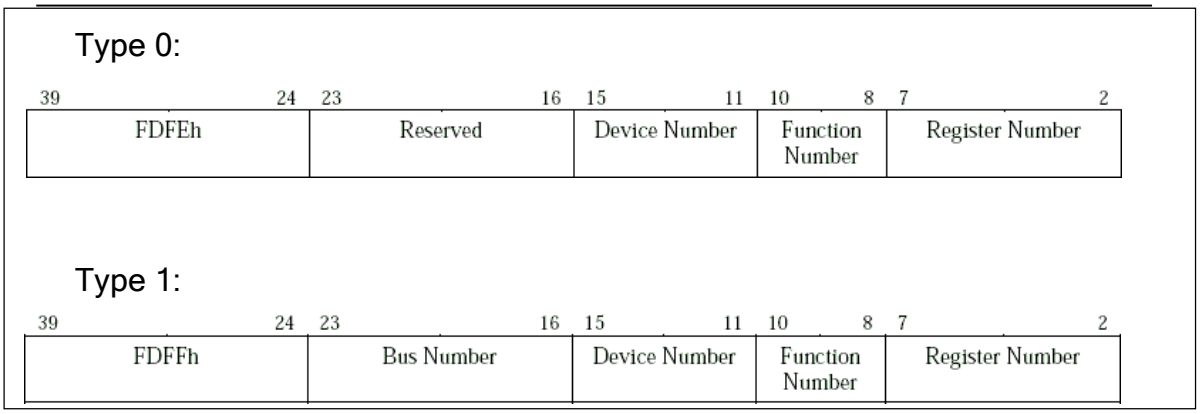


Figure 10-1 configuration access of HT protocol in loongson 3A2000

9.7 HyperTransport bus frequency software configuration method

The frequency of the HyperTransport interface bus can be controlled by two methods. The first is to configure the PLL frequency with the CLKSEL pin, and the second is to set the actual bus frequency with the configuration register Link Freq (offset 0x48, see 10.5.2).The other is to use the software frequency configuration register inside the controller (see 10.5.28) to set the PLL and the actual bus frequency. Compared with the pin setting method, a more abundant frequency combination can be obtained.Both methods eventually require a software reset via a one-time bus or LDT RECONNECT to take effect.

The method of using CLKSEL pin configuration is simple and will not be covered here. The method of setting using software registers is described in section 10.5.28, and some special notes are given here.

When a 16-bit HT is split into two 8-bit HTS, only the software frequency configuration register of HT LO can control PLL and bus frequency division, including the bus frequency division value of HT HI. That is to say, if left untreated, when HT LO resets the frequency, the HT HI frequency will also change. At this point, if HT HI is in a normal functional state, the bus may become unstable.

To avoid this, there are two ways to do it.

The first is to connect the reset signals of all HT together, so that after the software frequency configuration register of all HT controllers is configured, the reset signals of HT will be uniformly pulled down, and then pulled up to shake hands again. This enables HT LO and HT HI to switch clocks at the same time, ensuring the normal operation of the system. This method is suitable for HT0 connections in four-way interconnected systems.

The other is when the HT reset signal cannot be connected together. This needs to be prevented by means of software control

HT HI was affected when HT LO switched PLL frequency. The most straightforward approach is to set HT HI to the reset state,

Until HT LO completes the switch of PLL frequency, then HT HI's bus decompression is reset. This method is suitable for HT1 LO

The condition of the cross interconnect of the bridge piece, HT1 HI.

9.8 HyperTransport multi-processor support

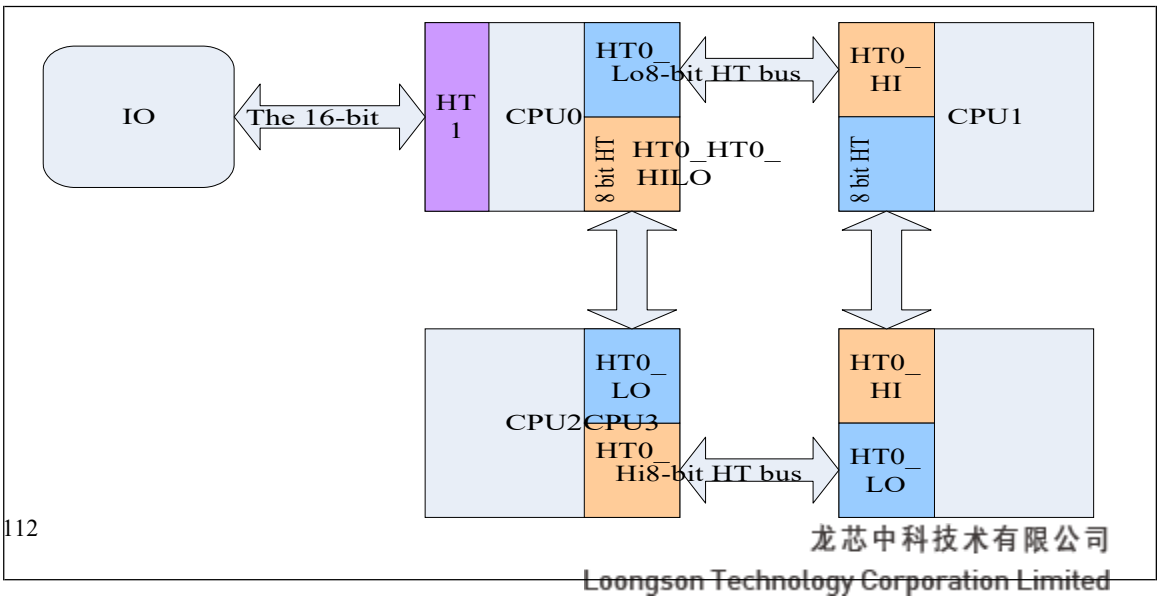
The loongson 3 processor USES the HyperTransport interface to interconnect multiple processors and is capable of hardware automatic maintenance

Consistency requests between 4 chips. Here are two ways to interconnect multiple processors:

Four pieces of loong core no. 3 interconnection structure

Four pieces of CPU are connected to form a ring structure. Each CPU USES two 8-bit controllers of HT0 to connect with two adjacent ones, among which HTx_LO is the main device and HTx_HI is the slave device. Thus, the following interconnection structure can be obtained:

FIG. 10-2 interconnection structure of four-piece longson no. 3



Loongson 3 interconnection routing

Loongson 3 interconnection routing adopts simple x-y routing method. That is, when routing, first X, then Y, take four chips as an example, ID

The Numbers are 0, 01, 10, 11. If a request is made from 11 to 00, it is routed from 11 to 00, first in the X direction,

Go from 11 to 10, then go in the Y direction, go from 10 to 00. When the request response returns from 00 to 11, the route goes first in the X direction, from 00 to 01, and then in the Y direction, from 01 to 11. As you can see, these are two different routing lines. Because of the characteristics of this algorithm, we will take a different approach when building the interconnection between two chips.

Two - piece loong core 3 interconnection structure

Because of the nature of the fixed routing algorithm, there are two different approaches to building a two-chip interconnection. The first is the use of 8-bit HT bus interconnection. In this mode of interconnection, only 8-bit HT interconnection can be used between two processors. The Numbers of the two chips are 00 and 01 respectively. According to the routing algorithm, we can know that when the two chips visit each other, they all pass through the 8-bit HT bus consistent with the four-chip interconnection. As follows:

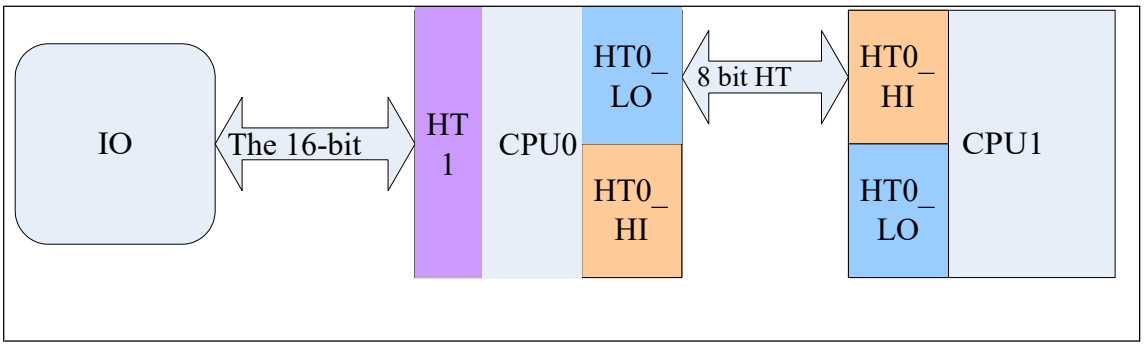


Figure 10-3 8-bit interconnection structure of two pieces of loong core no. 3

However, our HT bus can adopt 16-bit mode at its widest, so the connection mode to maximize bandwidth should be 16-bit interconnection structure. In loongson iii, as long as the HT0 controller is set

to 16-bit mode, all commands sent to the HT0 controller will be sent to HT0_LO instead of to HT0_HI or HT0_LO according to the routing table as before, so that we can use the 16-bit bus when interconnecting. Therefore, we only need to properly configure the 16-bit mode of CPU0 and CPU1 and properly connect the high-low bit bus to interconnect using the 16-bit HT bus. This interconnection structure can also use the 8-bit HT bus protocol to access each other. The resulting interconnection structure is as follows:

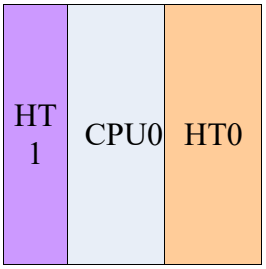


Figure 10-4 16-bit interconnection structure of two pieces of loong core 3

11 Low speed IO controller configuration

Loongson 3 I/O controller includes PCI controller, LPC controller, UART controller, SPI controller, GPIO and configuration register. These I/O controllers share a AXI port, and CPU requests are decoded to the corresponding device.

11.1 The PCI controller

Loongson 3's PCI controller can control the entire system either as a main bridge or as a normal PC device

On the PCI bus. Its implementation conforms to the PCI 2.3 specification. The loongson 3's PCI controller also has a PCI arbitrator built in.

The configuration header for the PCI controller is located at 256 bytes starting at 0x1FE00000, as shown in table 11-1.

Table 11-1 PCI controller configuration header

The byte 3	2 bytes	1 byte	Byte 0	addr ess

The Device ID		Vendor ID		00
The Status		The Command		04
The Class Code			Revision ID	08
BIST	The Header Type	Latency Timer	CacheLine Size	0 c
The Base Address Register 0				10
The Base Address Register 1				14
The Base Address Register 2				18
The Base Address Register 3				1 c
The Base Address Register 4				20
The Base Address Register 5				24
				28
Subsystem ID		Subsystem Vendor ID		2 c
				30
			"Capabilities Pointer	34
				38
Maximum Latency	A Minimum Grant	Interrupt Pin	Interrupt Line	3 c
Implementation Specific Register (ISR40)				40
Implementation Specific Register (ISR44)				44
Implementation Specific Register (ISR48)				48
Implementation Specific Register (ISR4C)				4 c
Implementation Specific Register (ISR50)				50
Implementation Specific Register (ISR54)				54
Implementation Specific Register (ISR58)				58
				.
PCIX Command Register				E0
PCIX Status Register				E4

Loongson 3A2000's PCIX controller supports three 64-bit Windows, including {BAR1, BAR0}, {BAR3, BAR2},

{BAR5, BAR4} three pairs of register configuration Windows 0, 1, 2 base addresses. The size, enabling, and other details of the window are controlled by the other three corresponding registers, PCI_Hit0_Sel, PCI_Hit1_Sel, and PCI_Hit2_Sel. See table 2 for the specific bit fields.

Table 11-2 PCI control registers

A doma in	The field name	access	Reset value	inst ruct ions
REG_40				

31	tar_read_io	Read and write (write 1 clear)	0	The target side receives access to IO or the unprefetched region
30	tar_read_discard	Read and write (write 1 clear)	0	The delay request on the target side is discarded
29	tar_resp_delay	Read and write	0	Target accesses when to give delay/split 0: after timeout 1: immediately
28	tar_delay_retry	Read and write	0	Target accesses the retry policy 0: according to internal logic (see 29 bits) 1: try again immediately
27	tar_read_abort_en	Read and write	0	If target times out for an internal read request, do you want to respond with target-abort
"	Reserved	-	0	
24	tar_write_abort_en	Read and write	0	If target times out for an internal write request, do you want to respond with target-abort
23	tar_master_abort	Read and write	0	Whether master-abort is allowed
Lift up	tar_subseq_timeout	Read and write	000	Target subsequent delay timeout 000:8 cycles Others: not supported
He hath	tar_init_timeout	Read and write	0000	Target initial delay timeout PCI mode 0:16 cycles 1-7: disable the counter 8-15:8 to 15 cycles The timeout count is fixed at 8 cycles in PCIX mode, where the configuration has the greatest impact Delay access number 0: 8 delay access 8: 1 delay access 9: 2 delay access 10: 3 delay access 11: 4 delay access
				12: 5 delay access 13: 6 delay access 14: 7 delay access 15: 8 delay access

Indeed,	tar_pref_boundary	Read and write	000 h.	Prefetching boundary configuration (in 16 bytes) FFF: 64KB to 16byte FFE: 64KB to 32byte FF8:64 KB to 128 byte
3	tar_pref_bound_en	Read and write	0	Configuration using tar_pref_boundary 0: prefetch to device boundary 1: using tar_pref_boundary
2	Reserved	-	0	
1	tar_splitw_ctrl	Read and write	0	Target split write control 0: blocks access other than Posted Memory Write 1: block all access until split is completed
0	mas_lat_timeout	Read and write	0	Disable mater access timeout 0: allows master access timeout 1: not allowed
REG_44				
31:0	Reserved	-	-	
REG_48				
31:0	tar_pending_seq	Read and write	0	The request number vector that target has not processed can be marked with 1
REG_4C				
charm	Reserved	-	-	
29	mas_write_defer	Read and write	0	Allow subsequent reads to override previous incomplete writes (for PCI only)
28	mas_read_defer	Read and write	0	Allow subsequent reads and writes to override previous incomplete reads (for PCI only)
27	mas_io_defer_cnt	Read and write	0	The maximum number of external IO requests Zero: by the control 1:1.
they	mas_read_defer_cnt	Read and write	010	Master supports maximum number of external reads (for PCI only) Zero: 8 1-7:1-7 Note: a dual address cycle access accounts for two items
Ephron;	err_seq_id	read-only	00 h	The target/master error number
15	err_type	read-only	0	Target /master error command type Zero:
14	err_module	read-only	0	Wrong module

				0: target 1: master
13	system_error	Read and write	0	Target /master system error (write 1 clear)
12	data_parity_error	Read and write	0	Target /master data parity error (write 1 clear)
11	ctrl_parity_error	Read and write	0	Target /master address odd and even wrong (write 1 clear)
10:0	Reserved	-	-	
REG_50				
31:0	mas_pending_seq	Read and write	0	The request number vector that the master hasn't finished processing is going to be 1
REG_54				
31:0	mas_split_err	Read and write	0	Split returns the error request bit vector
REG_58				
char m	Reserved	-	-	
then	tar_split_priority	Read and write	0	Target split returns the priority Zero is the highest, three is the lowest
But after	mas_req_priority	Read and write	0	The external priority of master Zero is the highest, three is the lowest
25	Priority_en	Read and write	0	Arbitrate algorithm (arbitrate between master's access and target's split return) 0: fixed priority 1: rotary
Wher eupon certai n	reserve	-	-	
17	mas_retry_aborted	Read and write	0	Master retry cancel (write 1 clear)
16	mas_trdy_timeout	Read and write	0	Master TRDY timeout count
"	mas_retry_value	Read and write	00 h	Number of master retries 0: infinite retry 1-255:1-255
away	mas_trdy_count	Read and write	00 h	Master TRDY timeout counter 0: disable 1-255:1-255

Before initiating configuration space reads and writes, the application should configure the PCIMap_Cfg register to tell the controller the type of configuration operation to initiate and the value on

the high 16-bit address line. The configuration header for the corresponding device can then be accessed by reading and writing to the 2K space starting at 0x1fe80000. The device number is obtained by encoding from low to high priority according to PCIMap_Cfg[15:0].

The configuration action address generation is shown in figure 11-1.

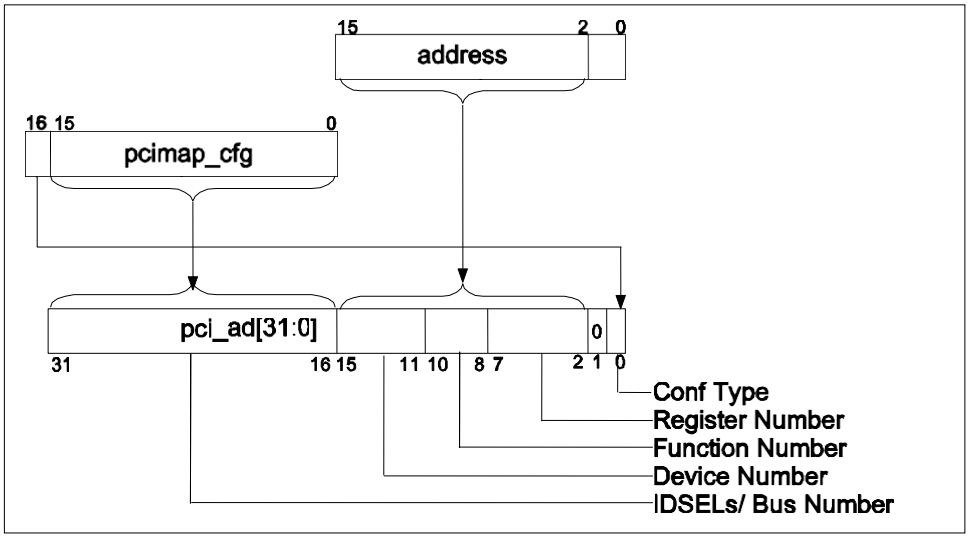


Figure 11-1 configure read and write bus address generation

PCI arbitrator realizes two-stage rotation arbitration, bus docking and isolation of damaged master devices.See its configuration and status

The PXArb_Config and PXArb_Status registers. PCI bus request and reply line assignment is shown in table 11-3.

Table 11-3 PCI/PCIX bus request and reply line allocation

Request and reply lines	describe
0	Internal integrated PCI/PCIX controller
7:1	External requests 6~0

The route-based arbitration algorithm provides two levels, the second level as a whole as a member of the first level scheduling together. When multiple devices apply for the bus at the same time, the first level device is transferred once per cycle, and the highest priority device in the second level can get the bus.

The mediators are designed to be switched whenever conditions permit, which can make them abnormal for some PCI devices that do not conform to the protocol. Using a forced priority allows these devices to occupy the bus through continuous requests.

Bus docking refers to whether one of the devices is selected to give a permit signal when no device requests to use the bus. Directly initiating bus operations can improve efficiency for devices that are already permitted. Internal PCI mediators provide two docking modes: the last master and the default master. If you can't dock on special occasions, you can set the mediator to dock to the default master device 0 (internal controller) with a dependency delay of 0.

11.2 LPC controller

LPC controller has the following characteristics:

- Comply with LPC1.1 specification
- Support for LPC access timeout counters
- Memory Read, Memory write access types are supported
- Firmware Memory Read, Firmware Memory Write access type (single byte)
- I/O read and I/O write access types are supported
- Support for Memory access type address translation
- Support for the Serizlized IRQ specification, providing 17 interrupt sources

The address space distribution of LPC controller is shown in table 11-4:

Table 11-4 address space distribution of LPC controller

Address name	Address range	The size of the
LPC Boot	0 x1fc0_0000 x1fd0_0000 0	1 mbyte
LPC Memory	0 x1c00_0000 x1d00_0000 0	16 mbyte
LPC I/O	0 x1ff0_0000 x1ff1_0000 0	64 kbyte
LPC Register	0 x1fe0_0200 x1fe0_0300 0	256 byte

The LPC Boot address space is the address space first accessed by the processor when the system is started. When PCI_CONFIG[0] pin is pulled down, the address of 0xBFC00000 is automatically routed to the LPC. This address space supports LPC Memory or Firmware Memory access types. The type of access emitted at system startup is controlled by LPC_ROM_INTEL pins. LPC_ROM_INTEL pins send out LPC Firmware Memory access when pulled up, and LPC_ROM_INTEL pins send out LPC Memory access type when pulled down.

The LPC Memory address space is the address space that the system accesses with Memory/Firmware Memory.The type of Memory access issued by an LPC controller is determined by the LPC controller's configuration register, LPC_MEM_IS_FWH.The address sent by the processor to this address space can be translated. The converted address is set by the LPC controller's configuration register LPC_MEM_TRANS.

The access that the processor sends to the LPC I/O address space is sent to the LPC bus according to

the LPC I/O access type. The address is 16 bits below the address space.

The LPC controller configuration registers have a total of 3 32-bit registers. The meaning of the configuration register is shown in table 11-5:

Table 11-5 LPC configuration register meanings

A domain	The field name	access	Reset value	instructions
REG0				
REG0 [hands]	SIRQ_EN	Read and write	0	SIRQ enables control
REG0 [take]	LPC_MEM_TRANS	Read and write	0	LPC Memory space address translation control
REG0 [15:0]	LPC_SYNC_TIMEOUT	Read and write	0	LPC access timeout counter
REG1				
REG1 [hands]	LPC_MEM_IS_FWH	Read and write	0	LPC Memory space Firmware Memory access type Settings
REG1 [17:0]	LPC_INT_EN	Read and write	0	LPC SIRQ interrupt enablement
REG2				
REG2 [17:0]	LPC_INT_SRC	Read and write	0	LPC SIRQ interrupts source indication
REG3				
REG3 [17:0]	LPC_INT_CLEAR	write	0	LPC SIRQ interrupt clear

11.3 UART controller

UART controllers have the following characteristics

- Full duplex asynchronous data receive/send
- Programmable data format
- 16 bit programmable clock counter
- Receive timeout detection is supported

- A multi-interrupt system with arbitration
- Work in FIFO mode only
- Register and function compatible NS16550A

Chip internal integration of two UART interfaces, function registers are exactly the same, but access to different base address.

The physical base address of the UART0 register is 0x1FE001E0. The physical base address of UART1 register is 0x1FE001E8.

11.3.1 Data register (DAT)

Chinese name: data transfer register register bit width: [7:0]

Offset: 0x00

Reset value: 0x00

A domain	A domain name	A wide	access	describe
away	Tx FIFO	8	W.	Data transfer register

11.3.2 Interrupt enabled register (IER)

Chinese name: interrupt enabled register register bit width: [7:0]

Offset: 0x01

Reset value: 0x00

A domain	A domain name	A wide	access	describe
The log	Reserved	4	RW	reserve
3	IME	1	RW	Modem status interrupted to enable '0' -- off '1' -- on
2	ILE	1	RW	Receiver line status interruption enables' 0 '-- off' 1 '-- on
1	ITxE	1	RW	Transfer save register for air break enable '0' -- close '1' -- open
0	IRxE	1	RW	Receive valid data interrupt enables' 0 '-- close' 1 '-- open

11.3.3 Interrupt identification register (IIR)

Chinese name: interrupt source register register bit width: [7:0]

Offset: 0x02

Reset value: 0xc1

A domain	A domain name	A wide	access	describe
The log	Reserved	4	R	reserve
3:1	II	3	R	Interrupt source representation bits, as shown in the table below
0	INTp	1	R	Interrupt representation bit

Interrupt control menu

Bit 3	2 -	Bit 1	priority	Interrupt type	The interrupt source	Interrupt reset control
0	1	1	1 st	Receiving line status	Odd or even, overflow or frame error, or hit Break the interrupt	Read the LSR
0	1	0	2 nd	A significant number is received According to the	The number of characters of FIFO is reached The level of the trigger	FIFO has a low number of characters In the trigger value
1	1	0	2 nd	Receive a timeout	There is at least one character in FIFO, But there are no operations, including read and write operations, in 4 character time	Read receive FIFO
0	0	1	3 rd	Transfer save hosting Device is empty	The transfer save register is empty	Write data to THR or More IIR
0	0	0	4 th	Modem state	CTS, DSR, RI or DCD.	Read MSR

11.3.4 FIFO control register (FCR)

Chinese name: FIFO control register register bit width: [7:0]

Offset: 0x02

Reset value: 0xc0

A domain	A domain name	A wide	access	describe
but	TL	2	W.	Receive the trigger value of FIFO's interrupt application '00' -- 1 byte '01' -- 4 bytes '10' -- 8 bytes' 11 '-- 14 bytes
o	Reserved	3	W.	reserve
2	Txset	1	W.	'1' clears the content of sending FIFO and reset its logic
1	Rxset	1	W.	'1' clears the contents of the received FIFO and reset its logic
0	Reserved	1	W.	reserve

11.3.5 Line control register (LCR)

Chinese name: circuit control register register bit width: [7:0]

Offset: 0x03

Reset value: 0x03

A domain	A domain name	A wide	access	describe
7	dlab	1	RW	Frequency divider latch access bit '1' - access operation frequency divider latch '0' - access operation normal register
6	BCB	1	RW	Interrupt control bit '1' - the output of the serial port is set to 0(interrupted state). '0' - normal operation
5	.spb	1	RW	Specifies parity bits '0' - do not specify parity bits '1' - if the LCR[4] bit is 1, the transmission and check parity bits are 0. If the LCR[4] bit is 0, the transmission and check parity bit is 1.

4	eps	1	RW	Parity bit selection '0' - has an odd number of 1's in each character (including data and parity bits) '1' - there are an even number of 1's in each character
3	PE	1	RW	Parity bit enabled '0' - no parity bits '1' - the parity bit is generated on the output, and the parity bit is judged on the input
2	sb	1	RW	Defines the number of bits that generate the stop bit '0' - 1 stop bit '1' - 1.5 stop bits at 5 character length, others The length is 2 stop bits
1-0	bec	2	RW	Sets the number of bits per character '00' -- 5 '01' -- 6
				"10" - 7 posi tion "11" - position 8

11.3.6 MODEM control register (MCR)

Chinese name: Modem control
register register bit width: [7:0]
Offset: 0x04
Reset value: 0x00

A domain	A domain name	A wide	access	describe
7:5	Reserved	3	W.	reserve

4	Loop	1	W.	Loop mode control bit '0' - normal operation '1' - loop mode.In loopback mode, TXD output is always 1, and the output shift register is connected directly to the input shift register.The other links are as follows. DTR □ DSR RTS □ CTS Out1 □ RI Out2 □ DCD
3	OUT2	1	W.	Connect to DCD input in loopback mode
2	The OUT1	1	W.	Connect to RI input in loop mode
1	RTSC	1	W.	RTS signal control bit
0	DTRC	1	W.	DTR signal control bit

11.3.7 Line status register (LSR)

Chinese name: line status register
register bit width: [7:0]

Offset: 0x05

Reset value: 0x00

A domain	A domain name	A wide	access	describe
7	The ERROR	1	R	Error representation bit '1' - at least one with a parity bit error, frame error, or interrupt. '0' - no errors

6	TE	1	R	The null transport represents the bit '1' - transfer FIFO and transfer shift registers are empty. Zero when writing data to the transmitted FIFO '0' -- has data
5	TFE	1	R	Transmission of FIFO bit empty represents bit '1' -- the current transmission FIFO is empty, and when data is written to the transmission FIFO, it will be zero '0' -- has data
4	BI	1	R	Interrupts interrupt bits '1' - received start bit + data + odd and even bit + stop bit are all 0, that is, there is interrupt interrupt '0' -- no interruptions
3	FE	1	R	Frame errors represent bits '1' - received data without stop bits '0' - no errors
2	PE	1	R	A parity bit error represents a bit '1' - there is a parity error in the received data '0' - no parity errors
1	OE	1	R	Data overflow represents bits '1' -- data overflow '0' - no overflow
0	Dr.	1	R	Receiving data effectively represents bits '0' - no data in a FIFO
				'1' - data in FIFO

When this register is read, LSR[4:1] and LSR[7] are cleared, LSR[6:5] is cleared when writing data to transmit FIFO, and LSR[0] judges the receiving FIFO.

11.3.8 MODEM status register (MSR)

Chinese name: Modem status register
register bit width: [7:0]

Offset: 0x06

Reset value: 0x00

A domain	A domain name	A wide	access	describe
7	CDCD	1	R	DCD input the inverse of the value, or in loop mode to Out2
6	CRI	1	R	The inverse of the RI input value, or connected to OUT1 in loopback mode
5	CDSR	1	R	The inverse of the DSR input value, or in loopback mode connected to DTR
4	CCTS	1	R	Invert the input value of the CTS, or connect to the RTS in loopback mode
3	DDCD	1	R	DDCD indicating a
2	TERI	1	R	RI edge detection. RI state changes from low to high
1	DDSR	1	R	DDSR indicating a
0	DCTS	1	R	DCTS indicating a

11.3.9 Frequency divider latch

Chinese name: frequency division
latch 1

Register bit width: [7:0]

Offset: 0x00

Reset value: 0x00

A domain	A domain name	A wide	access	describe
away	LSB	8	RW	Store the lower 8 bits of the frequency divider latch

Chinese name: frequency division
latch 2

Register bit width: [7:0]

Offset: 0x01

Reset value: 0x00

A domain	A domain name	A wide	access	describe
away	The MSB	8	RW	Store the high 8 bits of the frequency divider latch

11.4 SPI controller

SPI controller has the following characteristics:

- Full duplex synchronous serial port data transmission
- Support for variable-length byte transfers up to 4
- Master mode support
- A mode failure generates an error flag and issues an interrupt request
- Double buffer receiver
- Polarity and phase programmable serial clock
- SPI can be controlled in wait mode
- Support for starting from SPI

The SPI controller register physical address is 0x1FE00220.

Table 11-6 SPI controller address space distribution

Address name	Address range	The size of the
SPI Boot	0 x1fc0_0000 x1fd0_0000 0	1 mbyte
SPI Memory	0 x1d00_0000 x1e00_0000 0	16 mbyte
SPI Register	0 x1fe0_0220 x1fe0_0230 0	16 byte

SPI Boot address space is the address space first accessed by the processor when the system is started. When PCI_CONFIG[0] pin is pulled up, the address of 0xBFC00000 is automatically routed to SPI.

The SPI Memory space can also be accessed directly through CPU read requests, with a minimum of 1M bytes overlapping the SPI BOOT space.

11.4.1 Control register (SPCR)

Chinese name: control
register register bit width:
[7:0]
Offset: 0x00
Reset value: 0x10

A domain	A domain name	A wide	access	describe
7	Spie	1	RW	Interrupt output to make the signal highly efficient
6	The spe	1	RW	The system works to make the signal highly efficient
5	Reserved	1	RW	reserve
4	MSTR	1	RW	Select bit in master mode, this bit is always 1
3	cpol	1	RW	Clock polarity
2	cpha	1	RW	Clock phase 1 is in opposite phase and 0 is the same
1-0	SPR	2	RW	Sclk_o frequency divider setting, to be used with sper's spre

11.4.2 Status register (SPSR)

Chinese name: status
register register bit width:
[7:0]

Offset: 0x01

Reset value: 0x05

A domain	A domain name	A wide	access	describe
7	spif	1	RW	The interrupt sign bit 1 indicates that there is an interrupt request, and if you write 1, it will clear zero
6	wcol	1	RW	A write register overflow bit of 1 indicates overflow, and a write of 1 clears
when	Reserved	2	RW	reserve
3	wffull	1	RW	Write register full flag 1 to indicate full
2	wfempty	1	RW	Write register empty flag 1 to indicate empty
1	rffull	1	RW	Read register full flag 1 to indicate full
0	rfempty	1	RW	Read register empty flag 1 to indicate empty

11.4.3 Data register (TxFIFO)

Chinese name: data transfer
register register bit width: [7:0]
Offset: 0x02
Reset value: 0x00

A domain	A domain name	A wide	access	describe
away	Tx FIFO	8	W.	Data transfer register

11.4.4 External register (SPER)

Chinese name: external register
register bit width: [7:0]

Offset: 0x03

Reset value: 0x00

A domain	A domain name	A wide	access	describe
but	icnt	2	RW	How many bytes have been transmitted before the interrupt request signal is sent 00 -- 1 byte 01 -- 2 bytes 10-3 bytes 11-3 bytes
5-2	Reserved	4	RW	reserve
1-0	spre	2	RW	Set the frequency division ratio with the Spr

Frequency division coefficient:

spre SPR	00 00	00 01	00 10	00 11	01 00	01 01	01 10	01 11	10 00	10 01	10 10	10 11
Frequency division coefficient	2	4	16	32	8	64	128	256	512	1024	2048	4096

11.4.5 Parameter control register (SFC_PARAM)

SPI Flash parameter control
register register bit width: [7:0]

Offset: 0x04

Reset value: 0x21

A domain	A domain name	A wide	access	desc ribe
The log	clk_div	4	RW	Clock frequency selection (frequency division coefficient is the same as {spre, SPR} combination)
3	dual_io	1	RW	With dual I/O mode, priority is given to fast read mode
2	fast_read	1	RW	Use quick read mode
1	burst_en	1	RW	Spi flash supports sequential address reading mode
0	memory_en	1	RW	Spi flash read enable, invalid CSN [0] can be controlled by the software.

11.4.6 Chip selection control register (SFC_SOFTCS)

SPI Flash chip selection control
register register bit width: [7:0]

Offset: 0x05

Reset value: 0x00

A domain	A domain name	A wide	access	desc ribe
The log	CSN	4	RW	CSN pin output value
3-0	csen	4	RW	When is 1, the corresponding bit cs line is controlled by 7:4 bit

address	regist er	instructions
00	PonCfg	The electric configuration
04	GenCfg	General configuration
08	reserve	
0 c	reserve	
10	PCIMap	PCI mapping
14	PCIX_Bridge_Cfg	PCI/X bridge related configuration
18	PCIMap_Cfg	PCI concatenates the read and write device address
1 c	GPIO_Data	GPIO data
20	GPIO_EN	GPIO direction
24	reserve	
28	reserve	

2 c	reserve	
30	reserve	
34	reserve	
38	reserve	
3 c	reserve	
40	Mem_Win_Base_L	Prefetch window base address low 32 bits
44	Mem_Win_Base_H	Prefetch window base address height 32 bits
48	Mem_Win_Mask_L	Prefetch window mask lower 32 bits
4 c	Mem_Win_Mask_H	Prefetch window mask height 32 bits
50	PCI_Hit0_Sel_L	PCI window 0 controls low 32 bits

11.4.7 Timing control register (SFC_TIMING)

SPI Flash timing control register
register bit width: [7:0]

Offset: 0x06

Reset value: 0x03

A domain	A domain name	A wide	access	describe
Booths,	Reserved	6	RW	reserve
1-0	it	2	RW	The minimum invalid time of SPI Flash is calculated by clock period T after frequency division 00:1 t 01:2 t 10:4 t 11:8 t

11.5 IO controller configuration

The configuration register is used to configure the address window of the PCI controller, the arbitrator, and the GPIO controller.Table 11-6

These registers are listed and detailed descriptions of them are given in tables 11-7. This part of the register is base address 0x1FE00100. Table 11-7 IO

control registers

54	PCI_Hit0_Sel_H	PCI window 0 controls the height of 32 bits
58	PCI_Hit1_Sel_L	PCI window 1 controls the low 32-bit
5 c	PCI_Hit1_Sel_H	PCI window 1 controls the high 32-bit
60	PCI_Hit2_Sel_L	PCI window 2 controls low 32 bits
64	PCI_Hit2_Sel_H	PCI window 2 controls the high 32-bit
68	PXArb_Config	PCIX arbitrator configuration
6 c	PXArb_Status	PCIX arbitrator state
70		
74		
78		
7 c		
80	Chip Config	Chip configuration register
84		
88		
8 c		
90	Chip Sample	Chip sampling register

Table 11-8 registers are described in detail

A doma in	The field name	acce ss	Reset value	inst ruct ions
CR00: PonCfg				
15:0	pcix_bus_dev	read-only	Lio_ad [away]	The total amount used by the CPU in the PCIX Agent mode Line and equipment number
"	reserve	read-only	Lio_ad [or]	
Ephro n;	pon_pci_configi	read-only	pci_configi	PCI_Configi pin value
came	reserve	read-only		
CR04: reserve				
31:0	reserve	read-only	0	
CR08: reserve				
31:0	reserve	read-only	0	

CR10: PCIMap				
5-0	trans_lo0	Read and write	0	The PCI_Mem_Lo0 window maps the address six bits higher
but	trans_lo1	Read and write	0	The PCI_Mem_Lo1 window maps the address six bits higher
"	trans_lo2	Read and write	0	The PCI_Mem_Lo2 window maps the address to a height of 6 bits
all	reserve	read-only	0	
CR14: PCIX_Bridge_Cfg				
5-0	pcix_rgate	Read and write	6 'h18	PCIX mode to DDR2 read number threshold
6	pcix_ro_en	Read and write	0	The PCIX bridge allows writing over reading
all	reserve	read-only	0	
CR18: PCIMap_Cfg				
15:0	dev_addr	Read and write	0	PCI configuration reads and writes when the AD line is 16 bits high
16	conf_type	Read and write	0	Configure the types of reads and writes
31:17	reserve	read-only	0	
CR1C: GPIO_Data				
15:0	gpio_out	Read and write	0	GPIO outputs data
Caused the	gpio_in	Read and write	0	GPIO enters data
CR20: GPIO_EN				
15:0	gpio_en	Read and write	FFFF	High is the input and low is the output
Caused the	reserve	read-only	0	
CR3C: reserve				
31:0	reserve	read-only	0	reserve
CR24, 2 c, 30,34,38: reservations				
As shown in table 11				
CR50, 60 (54/58, 5 C / : _Sel_ PCI_Hit * *				
0	reserve	read-only	0	
2:1	pci_img_size	Read and	2 'b11	00:32; 10:64; Others: invalid

		write		
3	pref_en	Read and write	0	Prefetching can make
4	reserve	read-only	0	
62:12	bar_mask	Read and write	0	Window size mask (high 1, low 0)
63	burst_cap	Read and write	1	Whether to allow burst transmission
CR68: PXArb_Config				
0	device_en	Read and write	1	External device allowed
1	disable_broken	Read and write	0	Disable damaged master devices
2	default_mas_en	Read and write	1	The bus is docked to the default main device 0: dock to the last main device 1: dock to the default main device
o	default_master	Read and write	0	The bus is docked with the default main device number
but	park_delay	Read and write	2 'bl1	The delay from the beginning of no device request bus to the triggering of docked default device behavior 00:0 cycle
				01:8 cycles 10:32 cycle 11:128 cycles
"	level	Read and write	8 'h01-2	A device in the first stage
Ephron;	rude_dev	Read and write	0	Force priority devices The PCI device corresponding to the bit of 1 can occupy the bus by continuous request after obtaining the bus
away	reserve	read-only	0	
CR6C: PXArb_Status				
away	broken_master	read-only	0	Damaged master device (reset when changing disabled policy)
10:8	Last_master	read-only	0	Finally, the main device of the bus is used
lustful	reserve	read-only	0	

CR80: Chip config (see section 2.6)
CR90: Chip Sample (see section 2.6)
CRA0: Chip Sample (see section 2.6)
CRB0: PLL config (see section 2.6)
CRC0: PLL config (see section 2.6)
CRD0: Core config (see section 2.6)

12 List of chip configuration registers

The Name	ADDR	R/W	Description(NULL means no effect)	The default value
CPU_WIN0_BASE	0 x3ff00000	RW	Base address of CPU window 0	0 x0
CPU_WIN1_BASE	0 x3ff00008	RW	Base address of CPU window 1	0 x1000_0000
CPU_WIN2_BASE	0 x3ff00010	RW	Base address of CPU window 2	0 x1000_8000_0000
CPU_WIN3_BASE	0 x3ff00018	RW	Base address of CPU window 3	0 x0
CPU_WIN4_BASE	0 x3ff00020	RW	Base address of CPU window 4	0 x0
CPU_WIN5_BASE	0 x3ff00028	RW	Base address of CPU window 5	0 x0
CPU_WIN6_BASE	0 x3ff00030	RW	Base address of CPU window 6	0 x0
CPU_WIN7_BASE	0 x3ff00038	RW	The base address of CPU window 7	0 x0
CPU_WIN0_MASK	0 x3ff00040	RW	The mask for CPU window 0	0 xffff_ffff_f000_0000
CPU_WIN1_MASK	0 x3ff00048	RW	Mask for CPU window 1	0 xffff_ffff_f000_0000
CPU_WIN2_MASK	0 x3ff00050	RW	CPU window 2 mask	0 xffff_ffff_f000_0000
CPU_WIN3_MASK	0 x3ff00058	RW	Mask for CPU window 3	0 x0
CPU_WIN4_MASK	0 x3ff00060	RW	Mask for CPU window 4	0 x0
CPU_WIN5_MASK	0 x3ff00068	RW	Mask for CPU window 5	0 x0
CPU_WIN6_MASK	0 x3ff00070	RW	Mask for CPU window 6	0 x0
CPU_WIN7_MASK	0 x3ff00078	RW	CPU window 7 mask	0 x0

CPU_WIN0_MMAP	0 x3ff00080	RW	New base address for CPU window 0	0 xf0
CPU_WIN1_MMAP	0 x3ff00088	RW	New base address for CPU window 1	0 x1000_00f2
CPU_WIN2_MMAP	0 x3ff00090	RW	New base address for CPU window 2	0 xf0
CPU_WIN3_MMAP	0 x3ff00098	RW	New base address for CPU window 3	0 x0
CPU_WIN4_MMAP	0 x3ff000a0	RW	New base address for CPU window 4	0 x0
CPU_WIN5_MMAP	0 x3ff000a8	RW	New base address for CPU window 5	0 x0
CPU_WIN6_MMAP	0 x3ff000b0	RW	New base address for CPU window 6	0 x0
CPU_WIN7_MMAP	0 x3ff000b8	RW	New base address for CPU window 7	0 x0
PCI_WIN0_BASE	0 x3ff00100	RW	Base address for PCI window 0	0 x8000_0000
PCI_WIN1_BASE	0 x3ff00108	RW	Base address for PCI window 1	0 x0
PCI_WIN2_BASE	0 x3ff00110	RW	PCI window 2 base address	0 x0
PCI_WIN3_BASE	0 x3ff00118	RW	Base address for PCI window 3	0 x0
PCI_WIN4_BASE	0 x3ff00120	RW	Base address for PCI window 4	0 x0
PCI_WIN5_BASE	0 x3ff00128	RW	Base address for PCI window 5	0 x0
PCI_WIN6_BASE	0 x3ff00130	RW	Base address for PCI window 6	0 x0
PCI_WIN7_BASE	0 x3ff00138	RW	Base address for PCI window 7	0 x0
PCI_WIN0_MASK	0 x3ff00140	RW	Mask for PCI window 0	0 xffff_ffff_8000_0000
PCI_WIN1_MASK	0 x3ff00148	RW	Mask for PCI window 1	0 x0
PCI_WIN2_MASK	0 x3ff00150	RW	Mask for PCI window 2	0 x0

PCI_WIN3_MASK	0 x3ff00158	RW	Mask for PCI window 3	0 x0
PCI_WIN4_MASK	0 x3ff00160	RW	Mask for PCI window 4	0 x0
PCI_WIN5_MASK	0 x3ff00168	RW	Mask for PCI window 5	0 x0
PCI_WIN6_MASK	0 x3ff00170	RW	Mask for PCI window 6	0 x0
PCI_WIN7_MASK	0 x3ff00178	RW	Mask for PCI window 7	0 x0
PCI_WIN0_MMAP	0 x3ff00180	RW	New base address for PCI window 0	0 xf0
PCI_WIN1_MMAP	0 x3ff00188	RW	New base address for PCI window 1	0 x0
PCI_WIN2_MMAP	0 x3ff00190	RW	New base address for PCI window 2	0 x0
PCI_WIN3_MMAP	0 x3ff00198	RW	New base address for PCI window 3	0 x0
PCI_WIN4_MMAP	0 x3ff001a0	RW	New base address for PCI window 4	0 x0
PCI_WIN5_MMAP	0 x3ff001a8	RW	New base address for PCI window 5	0 x0
PCI_WIN6_MMAP	0 x3ff001b0	RW	New base address for PCI window 6	0 x0
PCI_WIN7_MMAP	0 x3ff001b8	RW	New base address for PCI window 7	0 x0
Slock0_addr	0 x3ff00200	RW	Lock address of lock window no. 0 ([63]: valid, [47:0]: addr)	0 x0
Slock1_addr	0 x3ff00208	RW	Lock address of no. 1 lock window ([63]: valid, [47:0]: addr)	0 x0
Slock2_addr	0 x3ff00210	RW	Lock address of no. 2 lock window ([63]: valid, [47:0]: addr)	0 x0
Slock3_addr	0 x3ff00218	RW	Lock address of no. 3 lock window ([63]: valid, [47:0]: addr)	0 x0
Slock0_mask	0 x3ff00240	RW	No. 0 lock window mask ([47:0]: mask)	0 x0
Slock1_mask	0 x3ff00248	RW	No. 1 lock window mask ([47:0]: mask)	0 x0

Slock2_mask	0 x3ff00250	RW	No. 2 lock window mask ([47:0]: mask)	0 x0
Slock3_mask	0 x3ff00258	RW	No. 3 lock window mask ([47:0]: mask)	0 x0
BARRIER_SET	0 x3ff00300	send	The barrier value plus one	
BARRIER_CLR	0 x3ff00308	send	The barrier value minus 1	
BARRIER_REF	0 x3ff00310	RW	The barrier threshold	0 x0
BARRIER_CTRL	0 x3ff00318	RW	Bit [0]: add or subtract enablement /barrier interrupt enablement	0 x0
BARRIER_VEC	0 x3ff00320	RO	The current values of the barrier	
CONFSIGNAL_CR	0 x3ff00400	RW	24: ccsd_en 19:16: ccsd_id 8: xrouter_en 5: x2_pci_rdinterleave 4: x2_cpu_rdinterleave 3-0: scid_sel	0 xffff_0000
gs3_HPT	0 x3ff00408	RO	A counter that adds 1 to each clock cycle	
MTXO_SRC_START_ADDR	0 x3ff00600	RW		0 x0
MTXO_DST_START_ADDR	0 x3ff00608	RW		0 x0
MTXO_ORI_LENTH	0 x3ff00610	RW		0 x0
MTXO_ORI_WIDTH	0 x3ff00618	RW		0 x0
MTXO_SRC_ROW_STRIDE	0 x3ff00620	RW		0 x0

MTX0_DST_ROW_STRIDE	0 x3ff00628	RW		0 x0
MTX0_TRANS_CTRL	0 x3ff00630	RW		0 x0
MTX1_SRC_START_ADDR	0 x3ff00700	RW		0 x0
MTX1_DST_START_ADDR	0 x3ff00708	RW		0 x0
MTX1_ORI_LENTH	0 x3ff00710	RW		0 x0
MTX1_ORI_WIDTH	0 x3ff00718	RW		0 x0
MTX1_SRC_ROW_STRIDE	0 x3ff00720	RW		0 x0
MTX1_DST_ROW_STRIDE	0 x3ff00728	RW		0 x0
MTX1_TRANS_CTRL	0 x3ff00730	RW		0 x0
SCache0_perfctrl0	0 x3ff00800	RW		
SCache0_perfcnt0	0 x3ff00808	RO		
SCache0_perfctrl1	0 x3ff00810	RW		
SCache0_perfcnt1	0 x3ff00818	RO		
SCache0_perfctrl2	0 x3ff00820	RW		
SCache0_perfcnt2	0 x3ff00828	RO		
SCache0_perfctrl3	0 x3ff00830	RW		
SCache0_perfcnt3	0 x3ff00838	RO		
SCache1_perfctrl0	0 x3ff00900	RW		
SCache1_perfcnt0	0 x3ff00908	RO		

SCache1_perfctrl1	0 x3ff00910	RW		
SCache1_perfcnt1	0 x3ff00918	RO		
SCache1_perfctrl2	0 x3ff00920	RW		
SCache1_perfcnt2	0 x3ff00928	RO		
SCache1_perfctrl3	0 x3ff00930	RW		
SCache1_perfcnt3	0 x3ff00938	RO		
SCache2_perfctrl0	0 x3ff00a00	RW		
SCache2_perfcnt0	0 x3ff00a08	RO		
SCache2_perfctrl1	0 x3ff00a10	RW		
SCache2_perfcnt1	0 x3ff00a18	RO		
SCache2_perfctrl2	0 x3ff00a20	RW		
SCache2_perfcnt2	0 x3ff00a28	RO		
SCache2_perfctrl3	0 x3ff00a30	RW		
SCache2_perfcnt3	0 x3ff00a38	RO		
SCache3_perfctrl0	0 x3ff00b00	RW		
SCache3_perfcnt0	0 x3ff00b08	RO		
SCache3_perfctrl1	0 x3ff00b10	RW		
SCache3_perfcnt1	0 x3ff00b18	RO		
SCache3_perfctrl2	0 x3ff00b20	RW		

SCache3_perfcnt2	0 x3ff00b28	RO		
SCache3_perfctrl3	0 x3ff00b30	RW		
SCache3_perfcnt3	0 x3ff00b38	RO		
Core0_IPI_Status	0 x3ff01000	RO	The IPI_Status register for the number 0 processor core	
Core0_IPI_Enalbe	0 x3ff01004	RW	The IPI_Enalbe register of the number 0 processor core	0 x0
Core0_IPI_Set	0 x3ff01008	send	The IPI_Set register of the number 0 processor core	
Core0_IPI_Clear	0 x3ff0100c	send	The IPI_Clear register for the number 0 processor core	
Core0_MailBox0	0 x3ff01020	RW	Register IPI_MailBox0 of the number 0 processor core	0 x0
Core0_MailBox1	0 x3ff01028	RW	Register IPI_MailBox1 of processor core 0	0 x0
Core0_MailBox2	0 x3ff01030	RW	Register IPI_MailBox2 for the number 0 processor core	0 x0
Core0_MailBox3	0 x3ff01038	RW	Register IPI_MailBox3 of processor core 0	0 x0
Core0_int_interval	0 x3ff01060	RW		
Core0_int_compare	0 x3ff01068	RW		
Core1_IPI_Status	0 x3ff01100	RO	The IPI_Status register for the number 1 processor core	
Core1_IPI_Enalbe	0 x3ff01104	RW	The IPI_Enalbe register of the number 1 processor core	0 x0
Core1_IPI_Set	0 x3ff01108	send	The IPI_Set register of the number 1 processor core	
Core1_IPI_Clear	0 x3ff0110c	send	The IPI_Clear register of the number 1 processor core	
Core1_MailBox0	0 x3ff01120	RW	Register IPI_MailBox0 for the number 1 processor core	0 x0
Core1_MailBox1	0 x3ff01128	RW	Register IPI_MailBox1 of the number 1 processor core	0 x0

Core1_MailBox2	0 x3ff01130	RW	The IPI_MailBox2 register of the number 1 processor core	0 x0
Core1_MailBox3	0 x3ff01138	RW	Register IPI_MailBox3 for the number 1 processor core	0 x0
Core1_int_interval	0 x3ff01160	RW		
Core1_int_compare	0 x3ff01168	RW		
Core2_IPI_Status	0 x3ff01200	RO	The IPI_Status register of the number 2 processor core	
Core2_IPI_Enalbe	0 x3ff01204	RW	The IPI_Enalbe register of the number 2 processor core	0 x0
Core2_IPI_Set	0 x3ff01208	send	The IPI_Set register of the number 2 processor core	
Core2_IPI_Clear	0 x3ff0120c	send	The IPI_Clear register of the number 2 processor core	
Core2_MailBox0	0 x3ff01220	RW	Register IPI_MailBox0 of the number 2 processor core	0 x0
Core2_MailBox1	0 x3ff01228	RW	Register IPI_MailBox1 in the number 2 processor core	0 x0
Core2_MailBox2	0 x3ff01230	RW	The IPI_MailBox2 register of the number 2 processor core	0 x0
Core2_MailBox3	0 x3ff01238	RW	The IPI_MailBox3 register of the number 2 processor core	0 x0
Core2_int_interval	0 x3ff01260	RW		
Core2_int_compare	0 x3ff01268	RW		
Core3_IPI_Status	0 x3ff01300	RO	The IPI_Status register of the number 3 processor core	
Core3_IPI_Enalbe	0 x3ff01304	RW	The IPI_Enalbe register of the number 3 processor core	0 x0
Core3_IPI_Set	0 x3ff01308	send	The IPI_Set register of the number 3 processor core	
Core3_IPI_Clear	0 x3ff0130c	send	The IPI_Clear register of the number 3 processor core	
Core3_MailBox0	0 x3ff01320	RW	Register IPI_MailBox0 for the number 3 processor core	0 x0

Core3_MailBox1	0 x3ff01328	RW	Register IPI_MailBox1 for the number 3 processor core	0 x0
Core3_MailBox2	0 x3ff01330	RW	The IPI_MailBox2 register of the number 3 processor core	0 x0
Core3_MailBox3	0 x3ff01338	RW	The IPI_MailBox3 register of the number 3 processor core	0 x0
Core3_int_interval	0 x3ff01360	RW		
Core3_int_compare	0 x3ff01368	RW		
Int Entry [0 -- 31]	0 x3ff01400	RW	32 8-bit interrupt routing registers	0 x0
Intisr	0 x3ff01420	RO	32-bit interrupt status register	
Inten	0 x3ff01424	RO	32-bit interrupt enabled status register	
Intenset	0 x3ff01428	send	The 32-bit setting enables the register	
Intenclr	0 x3ff0142c	send	32-bit clear enable registers and pulse-triggered interrupts	
Intpol	0 x3ff01430	send	useless	0 x0
Intedge	0 x3ff01434	send	32-bit trigger mode register (1: pulse trigger; 0: level triggered)	0 x0
CORE0_INTISR	0 x3ff01440	RO	Routing to the 32-bit interrupt state of CORE0	
CORE1_INTISR	0 x3ff01448	RO	Routing a 32-bit interrupt state to CORE1	
CORE2_INTISR	0 x3ff01450	RO	Routing a 32-bit interrupt state to CORE2	
CORE3_INTISR	0 x3ff01458	RO	Routing a 32-bit interrupt state to CORE3	

Thsens_int_ctrl_Hi	0 x3ff01460	RW	Hi_gate0: high temperature threshold 0, above which interruption will occur [8:8] : Hi_en0: high temperature interrupt enable 0 [11:10] : Hi_Sel0: select high temperature interrupt 0 as the temperature sensor input source [23:16] : Hi_gate1: high temperature threshold 1 above which interruption will occur [24:24] : Hi_en1: high temperature interrupt enable 1 [27:26] : Hi_Sel1: select high temperature interrupt 1 for the temperature sensor input source [39:32] : Hi_gate2: high temperature threshold 2, beyond which there will be an interrupt [40:40] : Hi_en2: high temperature interrupt enable 2 [43:42] : Hi_Sel2: select high temperature interrupt 2 as the temperature sensor input source [55:48] : Hi_gate3: high temperature threshold 3, beyond which there will be an interrupt [56:56] : Hi_en3: high temperature interrupt enable 3 [59:58] : Hi_Sel3: select the temperature sensor input source of high temperature interrupt 3	

Thsens_int_ctrl_Lo	0 x3ff01468	RW	Lo_gate0: low temperature threshold 0, below which an interrupt will occur [8:8] : Lo_en0: low temperature interrupt enable 0 [11:10] : Lo_Sel0: select the temperature sensor input source of cryogenic interrupt 0 [23:16] : Lo_gate1: cryogenic threshold 1 below which there will be an interrupt [24:24] : Lo_en1: cryogenic interrupt enable 1 [27:26] : Lo_Sel1: select the temperature sensor input source of cryogenic interrupt 1 [39:32] : Lo_gate2: cryogenic threshold 2, below which there will be an interrupt [40:40] : Lo_en2: cryogenic interrupt enable 2 [43:42] : Lo_Sel2: select the temperature sensor input source for cold interrupt 2 [55:48] : Lo_gate3: cold threshold 3, below which an interrupt [56:56] will occur: Lo_en3: cold interrupt enable 3 [59:58] : Lo_Sel3: select the temperature sensor input source of cryogenic interrupt 3	

Thsens_int_status/CLR	0 x3ff01470	RW	Interrupt status register, write any value to clear the interrupt [0] : high temperature interrupt trigger [1] : low temperature interrupt trigger	

Thsens_freq_scale	0 x3ff01480	RW	Temperature sensor high temperature frequency down control register, four sets the priority from high to low [7:0] : Scale_gate0: high temperature threshold 0, beyond which the frequency will be down [8:8] : Scale_en0: high temperature frequency down enable 0 Scale_Sel0: select the temperature sensor input source of high temperature frequency reduction 0 [14:12] : Scale_freq0: frequency separation value when frequency reduction [23:16] : Scale_gate1: high temperature threshold 1, beyond which the frequency will be reduced [24:24] : Scale_en1: high temperature frequency reduction enable 1 Scale_Sel1: select the temperature sensor input source of high temperature frequency reduction 1 [30:28] : Scale_freq1: frequency separation value when frequency reduction [39:32] : Scale_gate2: high temperature threshold 2, above which the frequency will be reduced [40:40] : Scale_en2: high temperature frequency reduction enable 2	

			Scale_Sel2: select the temperature sensor input source of high temperature frequency reduction 2 [46:44] : Scale_freq2: frequency separation value when frequency reduction [55:48] : Scale_gate3: high temperature threshold 3, above which frequency reduction [56:56] : Scale_en3: high temperature frequency reduction enable 3 [59:58] : Scale_Sel3: select the temperature sensor input source with high temperature down frequency 3 [62:60] : Scale_freq3: frequency division value when frequency is down	

DFD_PARAM	0 x3ff01500	RW	Debug trigger condition enabled [7:0] : timer, trigger delay, set to 1 to indicate immediate trigger when the condition is met, set to 0 to indicate no trigger, and set to other values to indicate the number of beats delayed triggered after the condition is met +1 [15:8] : trigger_en, trigger condition enabled, corresponding to the enable control of the eight external trigger events	
DFD_TRIGGER	0 x3ff01508	send	A software trigger, which sends a write to this address, will cause a software trigger condition that causes a timer-1 Trigger after beat	

CORE0_AWCOND0	0 x3ff01800	RW	CORE0's AXI interface AW triggers the condition 0 setting [15:0] : awid [19:16] : awlen [22:20] : awburst [26:25] : awlock [30:27] : awcache [33:31] : awprot [37:34] : awcmd [41:38] : awdirqid [43:42] : awstate [47:44] : swscseti [48] : awvalid [49] : awready	

CORE0_AWMASK0	0 x3ff01808	RW	CORE0's AXI interface AW trigger enable 0 is set, with the highest bit being AW channel trigger enable [49:0] : awmask [62] : awdata_en: trigger is allowed only when the wdata trigger condition of wid is met at the same time [63] : awchannel_en: trigger condition enables the trigger condition to be (AW_IN & AWMASK) == (AWCOND & AWMASK)	
CORE0_AWCOND1	0 x3ff01810	RW	The trigger condition for AW must be COND0 and COND1 at the same time [47:0] : awaddr	
CORE0_AWMASK1	0 x3ff01818	RW		

CORE0_ARCOND0	0 x3ff01820	RW	CORE0 has a AXI interface with AR trigger conditions similar to AW [15:0] : arid [19:16] : Arlen [22:20] : arsize [24:23] : arburst [26:25] : arlock [30:27] : arcache [33:31] : arprot [37:34] : arcpuno [48] : arvalid [49] : arready	
CORE0_ARMASK0	0 x3ff01828	RW	CORE0's AXI interface AR trigger enable 0 is set, with the highest bit being the AR channel trigger enable [49:0] : armask [62] : ardata_en: only when the rdata triggering condition of rid is met [63] : archannel_en: enable the trigger condition	
CORE0_ARCOND1	0 x3ff01830	RW	[47:0] : araddr	

COREO_ARMASK1	0 x3ff01838	RW		

CORE0_WCOND0	0 x3ff01840	RW	CORE0 has a AXI interface W trigger condition similar to AW [15:0] : wid [31:16] : WSTRB [32] : wlast [33] : wvalid [34] : wready	
CORE0_WMASK0	0 x3ff01848	RW	CORE0's AXI interface W trigger enable 0 is set, with the highest bit being the W channel trigger enable [49:0] : wmask [63] : wchannel_en: trigger condition enabled, not set when awdata_en is valid	
CORE0_WCOND1	0 x3ff01850	RW		
CORE0_WMASK1	0 x3ff01858	RW		
CORE0_WCOND2	0 x3ff01860	RW		
CORE0_WMASK2	0 x3ff01868	RW		
CORE0_BCOND0	0 x3ff01870	RW	CORE0 has a AXI interface B trigger condition similar to AW [15:0] : bid [17:16] : bresp [18] : bvalid	

			[19]：bready	

CORE0_BMASK0	0 x3ff01878	RW	CORE0's AXI interface B trigger enable 0 is set, with the highest bit being the B channel trigger enable [19:0] : bmask [63] : bchannel_en	
CORE0_RCOND0	0 x3ff01880	RW	CORE0 has a AXI interface R trigger condition similar to AW [15:0] : rid [17:16] : rresp [18] : rlast [19] : rrequest [21:20] : rstate [25:22] : rscseti [26] : rvalid [27] : rready	
CORE0_RMASK0	0 x3ff01888	RW	CORE0's AXI interface R trigger enable 0 is set, with the highest bit being the R channel trigger enable [27:0] : rmask [63] : rchannel_en	
CORE0_RCOND1	0 x3ff01890	RW		

CORE0_RMASK1	0 x3ff01898	RW		
CORE0_RCOND2	0 x3ff018a0	RW		
CORE0_RMASK2	0 x3ff018a8	RW		

TUD0_CONF0	0 x3ff018e0	RW	TUD0 configuration register 0 [47:0] : count_target [55:48] : monitor_enable	
TUD0_CONF1	0 x3ff018e8	RW	TUD0 with register 1 [2:0] : DCDL_sel_signal [5:3] : DCDL_sel_clock [9:6] : signal_sel [13:10] : clok_sel [20:14] : reading_sel [21] : counter_clock_sel [22] : sticky [23] : reset_g [24] : stop [25] : start [26] : cg_en	
TUD0_RESULT	0 x3ff018f0	R	TUD0 result register	
CORE1_AWCOND0	0 x3ff01900	RW	CORE1's AXI interface AW triggers the condition 0 setting	

CORE1_AWMASK0	0 x3ff01908	RW	CORE1's AXI interface AW trigger enable 0 is set, with the highest bit being AW channel trigger enable (AW_IN & AWMASK) == (AWCOND & AWMASK)	
CORE1_AWCOND1	0 x3ff01910	RW	The trigger condition for AW must be COND0 and COND1 at the same time	
CORE1_AWMASK1	0 x3ff01918	RW		
CORE1_ARCOND0	0 x3ff01920	RW	CORE1 has a AXI interface with AR trigger conditions similar to AW	
CORE1_ARMASK0	0 x3ff01928	RW		
CORE1_ARCOND1	0 x3ff01930	RW		
CORE1_ARMASK1	0 x3ff01938	RW		
CORE1_WCOND0	0 x3ff01940	RW	CORE1 has a AXI interface W trigger condition similar to AW	
CORE1_WMASK0	0 x3ff01948	RW		
CORE1_WCOND1	0 x3ff01950	RW		
CORE1_WMASK1	0 x3ff01958	RW		
CORE1_WCOND2	0 x3ff01960	RW		
CORE1_WMASK2	0 x3ff01968	RW		
CORE1_BCOND0	0 x3ff01970	RW	CORE1 has a AXI interface B trigger condition similar to AW	
CORE1_BMASK0	0 x3ff01978	RW		
CORE1_RCOND0	0 x3ff01980	RW	CORE1 has a AXI interface R trigger condition similar to AW	
CORE1_RMASK0	0 x3ff01988	RW		

CORE1_RCOND1	0 x3ff01990	RW		
CORE1_RMASK1	0 x3ff01998	RW		
CORE1_RCOND2	0 x3ff019a0	RW		
CORE1_RMASK2	0 x3ff019a8	RW		
TUD1_CONF0	0 x3ff019e0	RW	TUD1 configuration register 0 [47:0] : count_target [55:48] : monitor_enable	
TUD1_CONF1	0 x3ff019e8	RW	TUD0 with register 1 [2:0] : DCDL_sel_signal [5:3] : DCDL_sel_clock [9:6] : signal_sel [13:10] : clok_sel [20:14] : reading_sel [21] : counter_clock_sel [22] : sticky [23] : reset_g [24] : stop [25] : start [26] : cg_en	

TUD1_RESULT	0 x3ff019f0	R	TUD1 result register	
CORE2_AWCOND0	0 x3ff01a00	RW	CORE2's AXI interface AW triggers the condition 0 setting	
CORE2_AWMASK0	0 x3ff01a08	RW	CORE2's AXI interface AW trigger enable 0 is set, with the highest bit being AW channel trigger enable (AW_IN & AWMASK) == (AWCOND & AWMASK)	
CORE2_AWCOND1	0 x3ff01a10	RW	The trigger condition for AW must be COND0 and COND1 at the same time	
CORE2_AWMASK1	0 x3ff01a18	RW		
CORE2_ARCOND0	0 x3ff01a20	RW	CORE2 has a AXI interface with AR trigger conditions similar to AW	
CORE2_ARMASK0	0 x3ff01a28	RW		
CORE2_ARCOND1	0 x3ff01a30	RW		
CORE2_ARMASK1	0 x3ff01a38	RW		
CORE2_WCOND0	0 x3ff01a40	RW	CORE2 has a AXI interface W trigger condition similar to AW	
CORE2_WMASK0	0 x3ff01a48	RW		
CORE2_WCOND1	0 x3ff01a50	RW		
CORE2_WMASK1	0 x3ff01a58	RW		
CORE2_WCOND2	0 x3ff01a60	RW		
CORE2_WMASK2	0 x3ff01a68	RW		
CORE2_BCOND0	0 x3ff01a70	RW	CORE2 has a AXI interface B trigger condition similar to AW	

CORE2_BMASK0	0 x3ff01a78	RW		
CORE2_RCOND0	0 x3ff01a80	RW	CORE2 has a AXI interface R trigger condition similar to AW	
CORE2_RMASK0	0 x3ff01a88	RW		
CORE2_RCOND1	0 x3ff01a90	RW		
CORE2_RMASK1	0 x3ff01a98	RW		
CORE2_RCOND2	0 x3ff01aa0	RW		
CORE2_RMASK2	0 x3ff01aa8	RW		
TUD2_CONF0	0 x3ff01ae0	RW	TUD2 configuration register 0 [47:0] : count_target [55:48] : monitor_enable	

TUD2_CONF1	0 x3ff01ae8	RW	TUD0 with register 1 [2:0] : DCDL_sel_signal [5:3] : DCDL_sel_clock [9:6] : signal_sel [13:10] : clok_sel [20:14] : reading_sel [21] : counter_clock_sel [22] : sticky [23] : reset_g [24] : stop [25] : start [26] : cg_en	
TUD2_RESULT	0 x3ff01af0	R	TUD2 result register	
CORE3_AWCOND0	0 x3ff01b00	RW	CORE3's AXI interface AW triggers the condition 0 setting	
CORE3_AWMASK0	0 x3ff01b08	RW	CORE3's AXI interface AW trigger enable 0 is set, with the highest bit being AW channel trigger enable (AW_IN & AWMASK) == (AWCOND & AWMASK)	
CORE3_AWCOND1	0 x3ff01b10	RW	The trigger condition for AW must be COND0 and COND1 at the same time	
CORE3_AWMASK1	0 x3ff01b18	RW		

CORE3_ARCOND0	0 x3ff01b20	RW	CORE3 has a AXI interface with AR trigger conditions similar to AW	
CORE3_ARMASK0	0 x3ff01b28	RW		
CORE3_ARCOND1	0 x3ff01b30	RW		
CORE3_ARMASK1	0 x3ff01b38	RW		
CORE3_WCOND0	0 x3ff01b40	RW	CORE3 has a AXI interface W trigger condition similar to AW	
CORE3_WMASK0	0 x3ff01b48	RW		
CORE3_WCOND1	0 x3ff01b50	RW		
CORE3_WMASK1	0 x3ff01b58	RW		
CORE3_WCOND2	0 x3ff01b60	RW		
CORE3_WMASK2	0 x3ff01b68	RW		
CORE3_BCOND0	0 x3ff01b70	RW	CORE3 has a AXI interface B trigger condition similar to AW	
CORE3_BMASK0	0 x3ff01b78	RW		
CORE3_RCOND0	0 x3ff01b80	RW	CORE3 has a AXI interface R trigger condition similar to AW	
CORE3_RMASK0	0 x3ff01b88	RW		
CORE3_RCOND1	0 x3ff01b90	RW		
CORE3_RMASK1	0 x3ff01b98	RW		
CORE3_RCOND2	0 x3ff01ba0	RW		
CORE3_RMASK2	0 x3ff01ba8	RW		

TUD3_CONF0	0 x3ff01be0	RW	TUD3 configuration register 0 [47:0] : count_target [55:48] : monitor_enable	
TUD3_CONF1	0 x3ff01be8	RW	TUD0 with register 1 [2:0] : DCDL_sel_signal [5:3] : DCDL_sel_clock [9:6] : signal_sel [13:10] : clok_sel [20:14] : reading_sel [21] : counter_clock_sel [22] : sticky [23] : reset_g [24] : stop [25] : start [26] : cg_en	
TUD3_RESULT	0 x3ff01bf0	R	TUD3 result register	
TUD4_CONF0	0 x3ff01ce0	RW	TUD4 configuration register 0 [47:0] :	

			count_target [55:48] : monitor_enable	

TUD4_CONF1	0 x3ff01ce8	RW	TUD4 with a register 1 [2:0] : DCDL_sel_signal [5:3] : DCDL_sel_clock [8:6] : signal_sel [11:12] : clock_sel [18:12] : reading_sel [19] : counter_clock_sel [20] : sticky [21] : reset_g [22] : stop [23] : start [24] : cg_en	
TUD4_RESULT	0 x3ff01cf0	R	TUD4 result register	
TUD5_CONF0	0 x3ff01de0	RW	TUD5 configuration register 0 [47:0] : count_target [55:48] : monitor_enable	

TUD5_CONF1	0 x3ff01de8	RW	TUD5 with depositor 1 [2:0] : DCDL_sel_signal [5:3] : DCDL_sel_clock [8:6] : signal_sel [11:12] : clock_sel [18:12] : reading_sel [19] : counter_clock_sel [20] : sticky [21] : reset_g [22] : stop [23] : start [24] : cg_en	
TUD5_RESULT	0 x3ff01df0	R	TUD5 result register	
HT0_AWCOND0	0 x3ff01e00	RW	The AXI interface AW triggers the condition 0 setting for HT0	
HT0_AWMASK0	0 x3ff01e08	RW	HT0's AXI interface AW trigger enable 0 is set to the highest bit of AW channel trigger enable condition (AW_IN & AWMASK) == (AWCOND & AWMASK)	
HT0_AWCOND1	0 x3ff01e10	RW	The trigger condition for AW must be COND0 and COND1 at the same time	
HT0_AWMASK1	0 x3ff01e18	RW		

HT0_ARCOND0	0 x3ff01e20	RW	HT0 has a AXI interface with AR trigger conditions similar to AW	
HT0_ARMASK0	0 x3ff01e28	RW		
HT0_ARCOND1	0 x3ff01e30	RW		
HT0_ARMASK1	0 x3ff01e38	RW		
HT0_WCOND0	0 x3ff01e40	RW	The AXI interface of HT0 has AW trigger condition similar to AW	
HT0_WMASK0	0 x3ff01e48	RW		
HT0_WCOND1	0 x3ff01e50	RW		
HT0_WMASK1	0 x3ff01e58	RW		
HT0_WCOND2	0 x3ff01e60	RW		
HT0_WMASK2	0 x3ff01e68	RW		
HT0_BCOND0	0 x3ff01e70	RW	HT0 has a AXI interface B trigger condition similar to AW	
HT0_BMASK0	0 x3ff01e78	RW		
HT0_RCOND0	0 x3ff01e80	RW	HT0 has a AXI interface R trigger condition similar to AW	
HT0_RMASK0	0 x3ff01e88	RW		
HT0_RCOND1	0 x3ff01e90	RW		
HT0_RMASK1	0 x3ff01e98	RW		
HT0_RCOND2	0 x3ff01ea0	RW		
HT0_RMASK2	0 x3ff01ea8	RW		
HT1_AWCOND0	0 x3ff01f00	RW	HT1's AXI interface AW trigger condition is set to 0	

HT1_AWMASK0	0 x3ff01f08	RW	HT1's AXI interface AW trigger enable is set to 0, with the highest bit being AW channel trigger enable (AW_IN & AWMASK) == (AWCOND & AWMASK)	
HT1_AWCOND1	0 x3ff01f10	RW	The trigger condition for AW must be COND0 and COND1 at the same time	
HT1_AWMASK1	0 x3ff01f18	RW		
HT1_ARCOND0	0 x3ff01f20	RW	HT1 has a AXI interface with AR trigger conditions similar to AW	
HT1_ARMASK0	0 x3ff01f28	RW		
HT1_ARCOND1	0 x3ff01f30	RW		
HT1_ARMASK1	0 x3ff01f38	RW		
HT1_WCOND0	0 x3ff01f40	RW	The AXI interface of HT1 has AW trigger condition similar to AW	
HT1_WMASK0	0 x3ff01f48	RW		
HT1_WCOND1	0 x3ff01f50	RW		
HT1_WMASK1	0 x3ff01f58	RW		
HT1_WCOND2	0 x3ff01f60	RW		
HT1_WMASK2	0 x3ff01f68	RW		
HT1_BCOND0	0 x3ff01f70	RW	HT1 has a AXI interface B trigger condition similar to AW	
HT1_BMASK0	0 x3ff01f78	RW		
HT1_RCOND0	0 x3ff01f80	RW	HT1 has a AXI interface R trigger condition similar to AW	
HT1_RMASK0	0 x3ff01f88	RW		

HT1_RCOND1	0 x3ff01f90	RW		
HT1_RMASK1	0 x3ff01f98	RW		
HT1_RCOND2	0 x3ff01fa0	RW		
HT1_RMASK2	0 x3ff01fa8	RW		
CORE0_WIN0_BASE	0 x3ff02000	RW	Level 1 cross switch address window	0 x0
CORE0_WIN1_BASE	0 x3ff02008	RW	Level 1 cross switch address window	0 x0
CORE0_WIN2_BASE	0 x3ff02010	RW	Level 1 cross switch address window	0 x0
CORE0_WIN3_BASE	0 x3ff02018	RW	Level 1 cross switch address window	0 x0
CORE0_WIN4_BASE	0 x3ff02020	RW	Level 1 cross switch address window	0 x0
CORE0_WIN5_BASE	0 x3ff02028	RW	Level 1 cross switch address window	0 x0
CORE0_WIN6_BASE	0 x3ff02030	RW	Level 1 cross switch address window	0 x0
CORE0_WIN7_BASE	0 x3ff02038	RW	Level 1 cross switch address window	0 x0
CORE0_WIN0_MASK	0 x3ff02040	RW	Level 1 cross switch address window	0 x0
CORE0_WIN1_MASK	0 x3ff02048	RW	Level 1 cross switch address window	0 x0
CORE0_WIN2_MASK	0 x3ff02050	RW	Level 1 cross switch address window	0 x0
CORE0_WIN3_MASK	0 x3ff02058	RW	Level 1 cross switch address window	0 x0
CORE0_WIN4_MASK	0 x3ff02060	RW	Level 1 cross switch address window	0 x0
CORE0_WIN5_MASK	0 x3ff02068	RW	Level 1 cross switch address window	0 x0
CORE0_WIN6_MASK	0 x3ff02070	RW	Level 1 cross switch address window	0 x0

CORE0_WIN7_MASK	0 x3ff02078	RW	Level 1 cross switch address window	0 x0
CORE0_WIN0_MMAP	0 x3ff02080	RW	Level 1 cross switch address window	0 x0
CORE0_WIN1_MMAP	0 x3ff02088	RW	Level 1 cross switch address window	0 x0
CORE0_WIN2_MMAP	0 x3ff02090	RW	Level 1 cross switch address window	0 x0
CORE0_WIN3_MMAP	0 x3ff02098	RW	Level 1 cross switch address window	0 x0
CORE0_WIN4_MMAP	0 x3ff020a0	RW	Level 1 cross switch address window	0 x0
CORE0_WIN5_MMAP	0 x3ff020a8	RW	Level 1 cross switch address window	0 x0
CORE0_WIN6_MMAP	0 x3ff020b0	RW	Level 1 cross switch address window	0 x0
CORE0_WIN7_MMAP	0 x3ff020b8	RW	Level 1 cross switch address window	0 x0
CORE1_WIN0_BASE	0 x3ff02100	RW	Level 1 cross switch address window	0 x0
CORE1_WIN1_BASE	0 x3ff02108	RW	Level 1 cross switch address window	0 x0
CORE1_WIN2_BASE	0 x3ff02110	RW	Level 1 cross switch address window	0 x0
CORE1_WIN3_BASE	0 x3ff02118	RW	Level 1 cross switch address window	0 x0
CORE1_WIN4_BASE	0 x3ff02120	RW	Level 1 cross switch address window	0 x0
CORE1_WIN5_BASE	0 x3ff02128	RW	Level 1 cross switch address window	0 x0
CORE1_WIN6_BASE	0 x3ff02130	RW	Level 1 cross switch address window	0 x0
CORE1_WIN7_BASE	0 x3ff02138	RW	Level 1 cross switch address window	0 x0
CORE1_WIN0_MASK	0 x3ff02140	RW	Level 1 cross switch address window	0 x0
CORE1_WIN1_MASK	0 x3ff02148	RW	Level 1 cross switch address window	0 x0

CORE1_WIN2_MASK	0 x3ff02150	RW	Level 1 cross switch address window	0 x0
CORE1_WIN3_MASK	0 x3ff02158	RW	Level 1 cross switch address window	0 x0
CORE1_WIN4_MASK	0 x3ff02160	RW	Level 1 cross switch address window	0 x0
CORE1_WIN5_MASK	0 x3ff02168	RW	Level 1 cross switch address window	0 x0
CORE1_WIN6_MASK	0 x3ff02170	RW	Level 1 cross switch address window	0 x0
CORE1_WIN7_MASK	0 x3ff02178	RW	Level 1 cross switch address window	0 x0
CORE1_WIN0_MMAP	0 x3ff02180	RW	Level 1 cross switch address window	0 x0
CORE1_WIN1_MMAP	0 x3ff02188	RW	Level 1 cross switch address window	0 x0
CORE1_WIN2_MMAP	0 x3ff02190	RW	Level 1 cross switch address window	0 x0
CORE1_WIN3_MMAP	0 x3ff02198	RW	Level 1 cross switch address window	0 x0
CORE1_WIN4_MMAP	0 x3ff021a0	RW	Level 1 cross switch address window	0 x0
CORE1_WIN5_MMAP	0 x3ff021a8	RW	Level 1 cross switch address window	0 x0
CORE1_WIN6_MMAP	0 x3ff021b0	RW	Level 1 cross switch address window	0 x0
CORE1_WIN7_MMAP	0 x3ff021b8	RW	Level 1 cross switch address window	0 x0
CORE2_WIN0_BASE	0 x3ff02200	RW	Level 1 cross switch address window	0 x0
CORE2_WIN1_BASE	0 x3ff02208	RW	Level 1 cross switch address window	0 x0
CORE2_WIN2_BASE	0 x3ff02210	RW	Level 1 cross switch address window	0 x0
CORE2_WIN3_BASE	0 x3ff02218	RW	Level 1 cross switch address window	0 x0
CORE2_WIN4_BASE	0 x3ff02220	RW	Level 1 cross switch address window	0 x0

CORE2_WIN5_BASE	0 x3ff02228	RW	Level 1 cross switch address window	0 x0
CORE2_WIN6_BASE	0 x3ff02230	RW	Level 1 cross switch address window	0 x0
CORE2_WIN7_BASE	0 x3ff02238	RW	Level 1 cross switch address window	0 x0
CORE2_WIN0_MASK	0 x3ff02240	RW	Level 1 cross switch address window	0 x0
CORE2_WIN1_MASK	0 x3ff02248	RW	Level 1 cross switch address window	0 x0
CORE2_WIN2_MASK	0 x3ff02250	RW	Level 1 cross switch address window	0 x0
CORE2_WIN3_MASK	0 x3ff02258	RW	Level 1 cross switch address window	0 x0
CORE2_WIN4_MASK	0 x3ff02260	RW	Level 1 cross switch address window	0 x0
CORE2_WIN5_MASK	0 x3ff02268	RW	Level 1 cross switch address window	0 x0
CORE2_WIN6_MASK	0 x3ff02270	RW	Level 1 cross switch address window	0 x0
CORE2_WIN7_MASK	0 x3ff02278	RW	Level 1 cross switch address window	0 x0
CORE2_WIN0_MMAP	0 x3ff02280	RW	Level 1 cross switch address window	0 x0
CORE2_WIN1_MMAP	0 x3ff02288	RW	Level 1 cross switch address window	0 x0
CORE2_WIN2_MMAP	0 x3ff02290	RW	Level 1 cross switch address window	0 x0
CORE2_WIN3_MMAP	0 x3ff02298	RW	Level 1 cross switch address window	0 x0
CORE2_WIN4_MMAP	0 x3ff022a0	RW	Level 1 cross switch address window	0 x0
CORE2_WIN5_MMAP	0 x3ff022a8	RW	Level 1 cross switch address window	0 x0
CORE2_WIN6_MMAP	0 x3ff022b0	RW	Level 1 cross switch address window	0 x0
CORE2_WIN7_MMAP	0 x3ff022b8	RW	Level 1 cross switch address window	0 x0

CORE3_WIN0_BASE	0 x3ff02300	RW	Level 1 cross switch address window	0 x0
CORE3_WIN1_BASE	0 x3ff02308	RW	Level 1 cross switch address window	0 x0
CORE3_WIN2_BASE	0 x3ff02310	RW	Level 1 cross switch address window	0 x0
CORE3_WIN3_BASE	0 x3ff02318	RW	Level 1 cross switch address window	0 x0
CORE3_WIN4_BASE	0 x3ff02320	RW	Level 1 cross switch address window	0 x0
CORE3_WIN5_BASE	0 x3ff02328	RW	Level 1 cross switch address window	0 x0
CORE3_WIN6_BASE	0 x3ff02330	RW	Level 1 cross switch address window	0 x0
CORE3_WIN7_BASE	0 x3ff02338	RW	Level 1 cross switch address window	0 x0
CORE3_WIN0_MASK	0 x3ff02340	RW	Level 1 cross switch address window	0 x0
CORE3_WIN1_MASK	0 x3ff02348	RW	Level 1 cross switch address window	0 x0
CORE3_WIN2_MASK	0 x3ff02350	RW	Level 1 cross switch address window	0 x0
CORE3_WIN3_MASK	0 x3ff02358	RW	Level 1 cross switch address window	0 x0
CORE3_WIN4_MASK	0 x3ff02360	RW	Level 1 cross switch address window	0 x0
CORE3_WIN5_MASK	0 x3ff02368	RW	Level 1 cross switch address window	0 x0
CORE3_WIN6_MASK	0 x3ff02370	RW	Level 1 cross switch address window	0 x0
CORE3_WIN7_MASK	0 x3ff02378	RW	Level 1 cross switch address window	0 x0
CORE3_WIN0_MMAP	0 x3ff02380	RW	Level 1 cross switch address window	0 x0
CORE3_WIN1_MMAP	0 x3ff02388	RW	Level 1 cross switch address window	0 x0
CORE3_WIN2_MMAP	0 x3ff02390	RW	Level 1 cross switch address window	0 x0

CORE3_WIN3_MMAP	0 x3ff02398	RW	Level 1 cross switch address window	0 x0
CORE3_WIN4_MMAP	0 x3ff023a0	RW	Level 1 cross switch address window	0 x0
CORE3_WIN5_MMAP	0 x3ff023a8	RW	Level 1 cross switch address window	0 x0
CORE3_WIN6_MMAP	0 x3ff023b0	RW	Level 1 cross switch address window	0 x0
CORE3_WIN7_MMAP	0 x3ff023b8	RW	Level 1 cross switch address window	0 x0
EAST_WIN0_BASE	0 x3ff02400	RW	Level 1 cross switch address window	0 x0
EAST_WIN1_BASE	0 x3ff02408	RW	Level 1 cross switch address window	0 x0
EAST_WIN2_BASE	0 x3ff02410	RW	Level 1 cross switch address window	0 x0
EAST_WIN3_BASE	0 x3ff02418	RW	Level 1 cross switch address window	0 x0
EAST_WIN4_BASE	0 x3ff02420	RW	Level 1 cross switch address window	0 x0
EAST_WIN5_BASE	0 x3ff02428	RW	Level 1 cross switch address window	0 x0
EAST_WIN6_BASE	0 x3ff02430	RW	Level 1 cross switch address window	0 x0
EAST_WIN7_BASE	0 x3ff02438	RW	Level 1 cross switch address window	0 x0
EAST_WIN0_MASK	0 x3ff02440	RW	Level 1 cross switch address window	0 x0
EAST_WIN1_MASK	0 x3ff02448	RW	Level 1 cross switch address window	0 x0
EAST_WIN2_MASK	0 x3ff02450	RW	Level 1 cross switch address window	0 x0
EAST_WIN3_MASK	0 x3ff02458	RW	Level 1 cross switch address window	0 x0
EAST_WIN4_MASK	0 x3ff02460	RW	Level 1 cross switch address window	0 x0
EAST_WIN5_MASK	0 x3ff02468	RW	Level 1 cross switch address window	0 x0

EAST_WIN6_MASK	0 x3ff02470	RW	Level 1 cross switch address window	0 x0
EAST_WIN7_MASK	0 x3ff02478	RW	Level 1 cross switch address window	0 x0
EAST_WIN0_MMAP	0 x3ff02480	RW	Level 1 cross switch address window	0 x0
EAST_WIN1_MMAP	0 x3ff02488	RW	Level 1 cross switch address window	0 x0
EAST_WIN2_MMAP	0 x3ff02490	RW	Level 1 cross switch address window	0 x0
EAST_WIN3_MMAP	0 x3ff02498	RW	Level 1 cross switch address window	0 x0
EAST_WIN4_MMAP	0 x3ff024a0	RW	Level 1 cross switch address window	0 x0
EAST_WIN5_MMAP	0 x3ff024a8	RW	Level 1 cross switch address window	0 x0
EAST_WIN6_MMAP	0 x3ff024b0	RW	Level 1 cross switch address window	0 x0
EAST_WIN7_MMAP	0 x3ff024b8	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN0_BASE	0 x3ff02500	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN1_BASE	0 x3ff02508	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN2_BASE	0 x3ff02510	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN3_BASE	0 x3ff02518	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN4_BASE	0 x3ff02520	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN5_BASE	0 x3ff02528	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN6_BASE	0 x3ff02530	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN7_BASE	0 x3ff02538	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN0_MASK	0 x3ff02540	RW	Level 1 cross switch address window	0 x0

SOUTH_WIN1_MASK	0 x3ff02548	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN2_MASK	0 x3ff02550	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN3_MASK	0 x3ff02558	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN4_MASK	0 x3ff02560	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN5_MASK	0 x3ff02568	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN6_MASK	0 x3ff02570	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN7_MASK	0 x3ff02578	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN0_MMAP	0 x3ff02580	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN1_MMAP	0 x3ff02588	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN2_MMAP	0 x3ff02590	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN3_MMAP	0 x3ff02598	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN4_MMAP	0 x3ff025a0	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN5_MMAP	0 x3ff025a8	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN6_MMAP	0 x3ff025b0	RW	Level 1 cross switch address window	0 x0
SOUTH_WIN7_MMAP	0 x3ff025b8	RW	Level 1 cross switch address window	0 x0
WEST_WIN0_BASE	0 x3ff02600	RW	Level 1 cross switch address window	0 x0
WEST_WIN1_BASE	0 x3ff02608	RW	Level 1 cross switch address window	0 x0
WEST_WIN2_BASE	0 x3ff02610	RW	Level 1 cross switch address window	0 x0
WEST_WIN3_BASE	0 x3ff02618	RW	Level 1 cross switch address window	0 x0

WEST_WIN4_BASE	0 x3ff02620	RW	Level 1 cross switch address window	0 x0
WEST_WIN5_BASE	0 x3ff02628	RW	Level 1 cross switch address window	0 x0
WEST_WIN6_BASE	0 x3ff02630	RW	Level 1 cross switch address window	0 x0
WEST_WIN7_BASE	0 x3ff02638	RW	Level 1 cross switch address window	0 x0
WEST_WIN0_MASK	0 x3ff02640	RW	Level 1 cross switch address window	0 x0
WEST_WIN1_MASK	0 x3ff02648	RW	Level 1 cross switch address window	0 x0
WEST_WIN2_MASK	0 x3ff02650	RW	Level 1 cross switch address window	0 x0
WEST_WIN3_MASK	0 x3ff02658	RW	Level 1 cross switch address window	0 x0
WEST_WIN4_MASK	0 x3ff02660	RW	Level 1 cross switch address window	0 x0
WEST_WIN5_MASK	0 x3ff02668	RW	Level 1 cross switch address window	0 x0
WEST_WIN6_MASK	0 x3ff02670	RW	Level 1 cross switch address window	0 x0
WEST_WIN7_MASK	0 x3ff02678	RW	Level 1 cross switch address window	0 x0
WEST_WIN0_MMAP	0 x3ff02680	RW	Level 1 cross switch address window	0 x0
WEST_WIN1_MMAP	0 x3ff02688	RW	Level 1 cross switch address window	0 x0
WEST_WIN2_MMAP	0 x3ff02690	RW	Level 1 cross switch address window	0 x0
WEST_WIN3_MMAP	0 x3ff02698	RW	Level 1 cross switch address window	0 x0
WEST_WIN4_MMAP	0 x3ff026a0	RW	Level 1 cross switch address window	0 x0
WEST_WIN5_MMAP	0 x3ff026a8	RW	Level 1 cross switch address window	0 x0
WEST_WIN6_MMAP	0 x3ff026b0	RW	Level 1 cross switch address window	0 x0

WEST_WIN7_MMAP	0 x3ff026b8	RW	Level 1 cross switch address window	0 x0
NORTH_WIN0_BASE	0 x3ff02700	RW	Level 1 cross switch address window	0 x0
NORTH_WIN1_BASE	0 x3ff02708	RW	Level 1 cross switch address window	0 x0
NORTH_WIN2_BASE	0 x3ff02710	RW	Level 1 cross switch address window	0 x0
NORTH_WIN3_BASE	0 x3ff02718	RW	Level 1 cross switch address window	0 x0
NORTH_WIN4_BASE	0 x3ff02720	RW	Level 1 cross switch address window	0 x0
NORTH_WIN5_BASE	0 x3ff02728	RW	Level 1 cross switch address window	0 x0
NORTH_WIN6_BASE	0 x3ff02730	RW	Level 1 cross switch address window	0 x0
NORTH_WIN7_BASE	0 x3ff02738	RW	Level 1 cross switch address window	0 x0
NORTH_WIN0_MASK	0 x3ff02740	RW	Level 1 cross switch address window	0 x0
NORTH_WIN1_MASK	0 x3ff02748	RW	Level 1 cross switch address window	0 x0
NORTH_WIN2_MASK	0 x3ff02750	RW	Level 1 cross switch address window	0 x0
NORTH_WIN3_MASK	0 x3ff02758	RW	Level 1 cross switch address window	0 x0
NORTH_WIN4_MASK	0 x3ff02760	RW	Level 1 cross switch address window	0 x0
NORTH_WIN5_MASK	0 x3ff02768	RW	Level 1 cross switch address window	0 x0
NORTH_WIN6_MASK	0 x3ff02770	RW	Level 1 cross switch address window	0 x0
NORTH_WIN7_MASK	0 x3ff02778	RW	Level 1 cross switch address window	0 x0
NORTH_WIN0_MMAP	0 x3ff02780	RW	Level 1 cross switch address window	0 x0
NORTH_WIN1_MMAP	0 x3ff02788	RW	Level 1 cross switch address window	0 x0

NORTH_WIN2_MMAP	0 x3ff02790	RW	Level 1 cross switch address window	0 x0
NORTH_WIN3_MMAP	0 x3ff02798	RW	Level 1 cross switch address window	0 x0
NORTH_WIN4_MMAP	0 x3ff027a0	RW	Level 1 cross switch address window	0 x0
NORTH_WIN5_MMAP	0 x3ff027a8	RW	Level 1 cross switch address window	0 x0
NORTH_WIN6_MMAP	0 x3ff027b0	RW	Level 1 cross switch address window	0 x0
NORTH_WIN7_MMAP	0 x3ff027b8	RW	Level 1 cross switch address window	0 x0

13 Hardware and software design guide

Loongson 3A2000 processor pin downward compatible with loongson 3A1000 processor, but the corresponding software and hardware need to make some configuration changes to enable the original compatibility mode, or to open some new features of loongson 3A2000. This chapter focuses on the difference between the software and hardware Settings of loongson 3A2000 processor and loongson 3A1000 processor.

13.1 Hardware change guide

1. The original CORE_PLL_AVDD, DDR_PLL_AVDD, ht0/1_pll_avdd are now NC pin. If the original 3A motherboard is used, no modification can be made. However, if compatibility with future 3A3000 is considered, these supply voltages can be modified to 1.8v, or a 1.8v/2.5v configurable design can be adopted.
2. The original Mc0/1_comp_ref_res was changed to NC pin. If the original 3A motherboard is used, no modification can be made;
3. The original ht0/1_pll_ref was changed to NC pin. If the original 3A motherboard is used, no modification can be made;
4. The original Mc0/1_comp_ref_gnd was changed to Mc0/1_a15. If the original 3A motherboard is used, no modification can be made; But if connected to a memory stick, you can support a larger amount of memory;
5. The function of PCI_CONFIG[0] control is changed to SPI startup enable, which can be started from SPI FLASH after setting to 1.If you use the original 3A motherboard, you need to set it to 0 and start from LPC FLASH. If the mainboard already has SPI FLASH, you can connect GPIO[0] as SPI_CS, and set PCI_CONFIG[0] to 1 to start from SPI FLASH.
6. The control function of PCI_CONFIG[7] was changed to force HT1.0 mode, and HT was directly started in 1.0 mode after it was set to 1. If you use the 3A780E motherboard, you need to set it to 1 at present; If using 3A2H motherboard, no special Settings are required;
7. CLKSEL[15:10] needs to be set to 6 'b100001; If you need to use HT3.0 mode, you need to use CLKSEL[15:10]
8. Set it to 6 'b100101;
9. CLKSEL[9:5] needs to be set to 5 'b01111; Use PMON to set the memory frequency;
10. CLKSEL[4:0] needs to be set to 5 'b01111; PMON is used to set the processor core frequency.
11. For the 3A2H motherboard, the pull up resistance on ht0/1_powerok and ht0/1_resetn should be removed.(the original pull-up
12. The resistance of 300 ohms is not suitable for 3A, it can also be removed.)

13.2 Frequency setting instructions

13. In order to be basically compatible with the frequency configuration of loongson 3A1000, the hardware frequency configuration range of loongson 3A2000 is relatively narrow. In order to obtain a wider frequency range and better clock quality, the software configuration method in PMON is mainly used in loongson 3A2000. The configuration method is the same as that of loongson 3B1500. Refer to the PMON source code for the specific configuration method.
14. The frequency setting is completely set by the software, and the CLKSEL need not be modified when the frequency is changed.
15. 1.15v stable operating frequency of core voltage: the processor core frequency is set to 800MHz, the memory frequency is set to 500MHz, the HT controller is set to 400MHz, and the HT bus is 800MHz/1600MHz.

13.3 PMON change guide

16. Compared with the loong chip 3A1000, it has been upgraded to different degrees from processor core, memory controller, HT controller to cross switches at all levels, so PMON needs to make some changes, mainly including the following parts:
17. Initialization of L1 Dcache, L1 Icache, Vcache and L2 Cache after power is removed (hardware completed);
18. After the CPU is energized, close all the core Store Fill Buffer;
19. After the CPU is powered on, turn off all the core write merge functions;
20. If you need to maintain compatibility with 3A5, set the PRID hidden bit in CP0 Diag register for all cores.
21. Change all statements in assembly code that are jr rx and rx is not register 31 to jr $31;
22. Use the code to configure the processor core, memory and node PLL similar to 3B1500.
23. Use the memory controller configuration and parameter training code similar to 3B1500;
24. If HT works in mode 1.0, HT can only work in 8-bit mode.
25. If the SPI controller is used, the base address is changed from 0xBFE001F0 to 0xBFE00220;
26. In addition to these required changes, you can make the following changes to enhance PMON functionality:
27. Modify the delay of the buzzer to ensure that the user can hear the buzzer;
28. Added support for turning off defective nuclear clocks;

13.4 Kernel change guide

29. Changes that need to be made in the kernel include:
30. Modify the Cache description structure in the kernel, VCache and SCache are connected by 16-way groups;
31. Modify the calculation method of the temperature sensor to read -100 as the same as 3B1500. At present, the sample has not been tested and calibrated, and there may be a large deviation between the readout value of some chips and the actual temperature. Therefore, it is recommended not to use the temperature indicator of the temperature sensor inside theprocessor in the current kernel.
32. Modify the configuration register address when closing the core;
33. Change the operation of brush ICache/DCache to brush ICache/DCache/VCache;

34. If the SPI controller is used, the base address is changed from 0xBFE001F0 to 0xBFE00220;
35. Uncache DMA must be used to maintain the consistency of Cache data.
36. Add store fill buffer support: first, SYNC needs to be added before all Uncache requests to ensure
37. When the Uncache request occurs, all contents in the store fill buffer are written back to the Cache. Second, it is necessary to use LL/SC instruction to unlock all synchronous operations Shared among different cores.
38. Do not use the device's MSI functionality. When the MSI function must be used, the number of data receiving buffers of the HT controller's POST channel should be set to 1, and the HT bus should be reconnected.
39. Lock Cache operations cannot be used for DMA regions where hardware automatically maintains consistency.
40. Other modifications that can be used to improve performance are:
41. Increased support for FTLB;
42. Added support for TLB rapid refill;
43. Add wait command support;
44. Added prefetch instruction support;
45. Interrupt return is implemented using DI/EI. However, it should be noted that the [31:4] returned by EI instruction is a random value, which is different from the MIPS regulation.

13.5 Other change description

1. Performance counter overflow interrupts cannot achieve precise interrupts, resulting in limited use of current perf tools. If necessary, frequent mfc0 perfcnt instructions (user-mode available) must be added. For example, if the instruction is inserted into the processing function of high-frequency clock interrupt, it will still cause the interrupt to fail to be generated in time, resulting in a large error in event statistics.