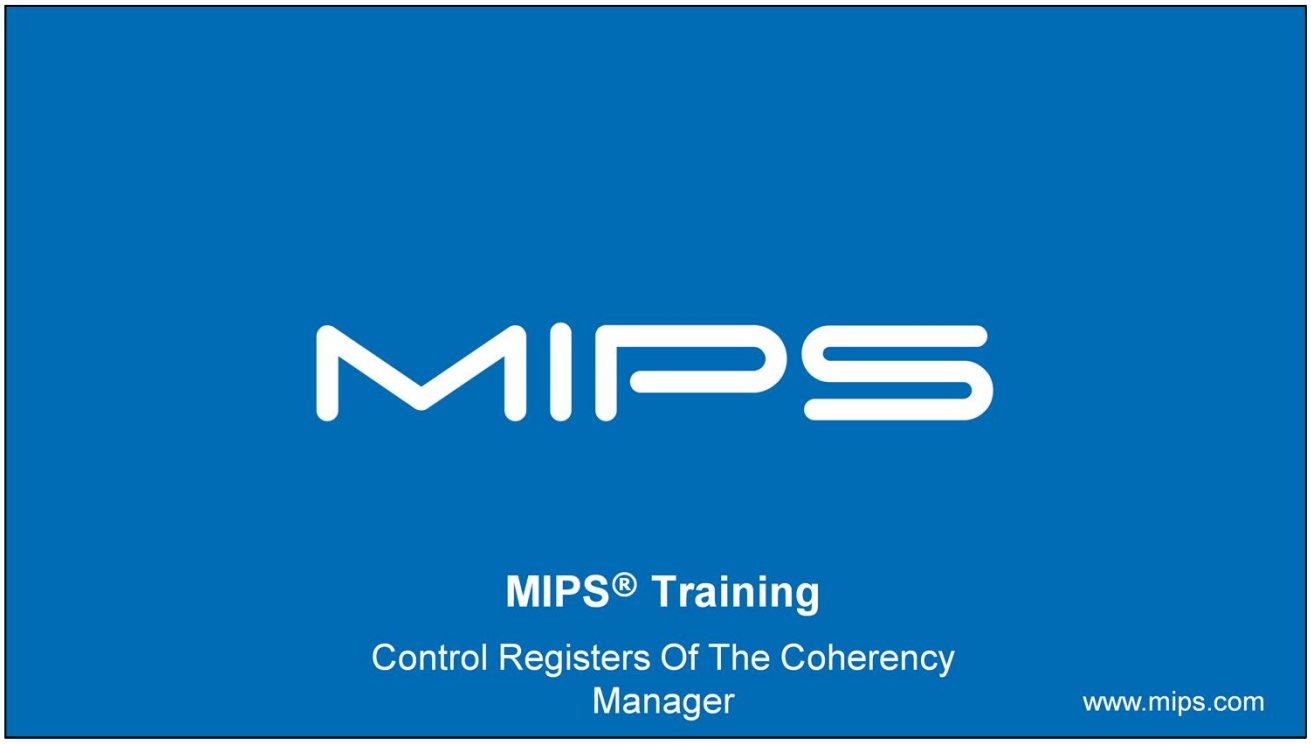

This section covers the control registers of the Coherency Manager

Global Configuration Registers

- **Global Configuration Registers (GCR)**
  - A set of memory-mapped controller registers that are used to configure and control various aspects of the coherence scheme and Coherence Manager.
    - Provides information on system configuration
    - Configure address map locations of the Global Interrupt Controller and Cluster power controller.
    - Configure the non coherent areas of memory to either real memory or I/O devices.
    - Control the coherency of the default shared memory region.
    - Controls the handling and reporting of coherency manager errors.
    - Controls other options of the Coherency manager

MIPS

The Global Configuration Registers referred to as GCR, are the system programmers interface to the Coherency manager. The location of these register in the memory map is determined at core build time.

+ The GCRs are used to set the addresses for the Global Interrupt Controller and Cluster power controller.

+ They can configure the non coherent areas of memory to either real memory or I/O devices.

+ Control the coherency of the default shared memory region.

+ Controls the handling and reporting of coherency manager errors.

+ and any other options of the Coherency manager

Global Configuration Registers

MIPS32

- **GCR - Memory Mapped Registers**
  - Control and Status registers
  - Total size of 32 Kbytes, divided into 8 KByte (0x2000) blocks

GCR Base Offset	Description
0x0000 - 0x1FFF	Global Control Block. Contains registers pertaining to the global system functionality.
0x2000 - 0x3FFF	Core-Local Control Block (aliased for each CPU core). Contains registers pertaining to the core issuing the request. Each CPU has its own copy of registers within this block.
0x4000 - 0x5FFF	Core-Other Control Block (aliased for each CPU core). This block of addresses gives each Core a window into another CPU's Core-Local Control Block. Before accessing this space, the Core-Other_Addressing Register in the Core Local Control Block must be set with the CORENum of the target Core.
0x6000 - 0x7FFF	Global Debug Block. Contains global registers useful in debugging the MPS.

MIPS

3

The Global Control Registers are memory mapped registers located in the system's address space. The GCR space contains control and status registers for the entire Coherent Processing System cluster and for the individual CPUs in the cluster.

+ All registers in the Global Control Block are 32 bits wide and should only be accessed using 32-bit uncached loads and stores. Reads from unpopulated registers in the GCR address space return 0x0, and writes to those locations are silently dropped without generating any exceptions.

+ The GCR address space has a total size of 32 Kbytes, which is divided into 8 Kbyte blocks

+ The first block is a global control block. This is a system wide block that cover all the global aspects of the Coherent Processing System.

Some configuration values are on a per core basis so each core has a set of registers that pertain to these local configuration items. This set of registers is called the Core-Local control block. The address range for this block is the same for all cores relative to the local core performing the ld or store to a register in this block.

A core can access another core's local registers by using the Core-Other Control Block. That way one core can read the status or change the configuration of another core. The address range for this block is also the same for all cores relative to the local core performing the ld or store to a register in this block. The selection of which cores registers will be accessed in this block will be covered later.

The Global Debug Block is system wide block. It used for debugging purposes and Performance information. The Debug Block is covered in upcoming sections.

Location of the GCR address space

Register Fields		CMGCR Base Register (CP0 Register 15, Select 3)	Reset State
Name	Bits		
CMGCR_ BASE_ADDR	31:11	Bits 31:11 of the base physical address of the memory mapped Coherency Manager GCR registers. This register field reflects the value of the GCR_BASE field within the memory-mapped Coherency Manager GCR Base Register. The number of implemented physical address bits is implementation-specific. For the unimplemented address bits - writes are ignored, returns zero on read.	Preset

```
#define C0_CMGCR  $15,3
#define getcmgcrg() \
({ unsigned int __value; \
__asm__ __volatile ("mfc0 %0, $%1, 0" : "=d" (__value) : "i" (C0_CMGCR)); \
__value;})
unsigned int x;
x = getcmgcrg() ;
```

MIPS

4

The memory mapped address for the base of the Global Configuration Registers can be found by reading CP0 register 15 select 3.

+ Here are the macros that will allow you to access the register from a c program.

+ Here is an example of calling the macro from C

Global Configuration Register (CM 2.5)

Register Fields		Global Config Register (GCR_CONFIG Offset 0x0000)	Reset State
Name	Bits		
NUM_ADDR_ REGIONS	19:16	Total number of CM Address Regions. Note: only 0, 4,or 6 Address Regions are currently supported. Encoding: 0x0 — Meaning: 0 Address Regions Encoding: 0x1 — Meaning: 1 Address Regions	IP Configuration Value
NUMIOCU	11:8	Total number of IOCUs in the system. Note: 0 - 2 IOCU are currently supported. Encoding: 0x0 — Meaning: 0 IOCU Encoding: 0x1 — Meaning: 1 IOCU Encoding: 0x2 — Meaning: 2 IOCUs	IP Configuration Value
PCORES	7:0	Total number of Cores in the system not including the IOCUs. Encoding: 0x0 — Meaning: 1 core Encoding: 0x1 — Meaning: 2 cores	IP Configuration Value

MIPS 5

The first register in the Global block section is the global configuration register. This register is a read-only register the gives you information on the configuration of the Coherency Manager in your system.

NUM_ADDR_REGIONS is the number of Address Regions. This tells you how many regions you can configure to support different coherency policies. You would use regions, for example, to configure an area of memory to be used for memory mapped I/O or to change the coherency of the region between the l2 cache controller and real memory. I'll talk more about regions later in this section.

NUMIOCU Total number of I/O Control Units in the system. Our cores support either none or up to 2 at this time.

PCORES tells you the number of cores in the Coherent Processing System.

GCR Base Register Offset

Register Fields		GCR Base Register (GCR_BASE Offset 0x0008)	Reset State
Name	Bits		
GCR_BASE	31:15	This field sets the base address of the 32KB GCR block of the CPS. This register has a fixed value after reset if configured as Read-Only (an IP Configuration Option).	IP

The GCR Base register lets you configure the GCR base address and Coherency attributes under certain conditions. In most cases this value is read only value having been configured as a IP option when the core was built.

GCR Base Register

Register Fields		GCR Base Register (GCR_BASE Offset 0x0008)	Reset State
Name	Bits		
CCA_DEFAULT_ OVERRIDE_VALUE	7-5	Used with CCA_DEFAULT_OVERRIDE_ENABLE to force the Cache Coherency Attribute (CCA) value for transactions on the L2/Memory OCP. See CCA_DEFAULT_OVERRIDE_ENABLE field.	0
CCA_DEFAULT_ OVERRIDE_ENABLE	4	If CCA_DEFAULT_OVERRIDE_ENABLE is set to 1 and CM_DEFAULT_TARGET is set to Memory, then transactions with addresses that do not map to any region will have a CCA value set to CCA_DEFAULT_ OVERRIDE_VALUE when driven to L2/Memory	0

Encoding	Name	Description
0x0	WT	Write Through
0x2	UC	Uncached
0x3	WB	Write Back non-coherent
0x4	CWBE	Mapped to WB
0x5	CWB	Mapped to WB
0x7	UCA	Uncached Accelerated

MIPS

7

Default Address Region. For the default Address Region there is one configuration register that determines if the Cache coherency attribute default override is enable, the CCA if the override is enabled and the memory port that is associated with the default region.

CCA_Override_Value and CCA_Override_enable are used in conjunction with CM_DEFAULT_TARGET memory region shown in the next slide. As you will understand when I explain CM regions, you can configure each with a starting address and size. The CM_DEFAULT_TARGET region covers any memory not already covered by a configurable CM region. If you don't have any configurable regions then this default region will cover all of memory. The CM_DEFAULT_TARGET can be set to 2 different types of addresses, addresses that go to memory and addresses that go to the I/O Coherency Unit for memory mapped I/O. Of course if you don't have any other regions in your system this default region should be set to Memory. In a typical system you would use the other regions to map to the IOCU type for memory mapped I/O and this default region to map to a memory type.

The CCA_DEFAULT_OVERRIDE_VALUE controls the cache coherency algorithm between the L2 cache and memory. So If this is set to Write through any writes to the L2 cache will also be written to memory. If set to uncached writes will not be stored in the L2 and go directly to memory. If set to write back all writes will be stored in the cache and only be written to memory if evicted or flushed, Last if set to uncached accelerated writes will not be stored in the L2 but will be gathered in a cache line size buffer before being written to memory.

The CCA_DEFAULT_OVERIDE_ENABLE bit controls the enabling of the default CCA_DEFAULT_OVERRIDE_VALUE if the bit is set the region will use the CCA_DEFAULT_OVERRIDE_VALUE. If it is zero the it will use the same CCA as the L1 cache.

GCR Base Register

Register Fields		GCR Base Register (GCR_BASE)	Reset State
Name	Bits		
CM_DEFAULT_ TARGET	1:0	Determines the target device for the default memory region.	Value of signal SI_CM_Default_Target [1:0]

Encoding	Meaning
0	Memory
2	1st IOCU
3	2nd IOCU

More on the GCR Base register:

The CM_DEFAULT_TAGET controls if the region is used for memory or memory mapped IO.

Global CSR Access Privilege Register

Register Fields		Global CSR Access Privilege Register (GCR_ACCESS Offset 0x0020)	Reset State
Name	Bits		
CM_ ACCESS_EN	7:0	Each bit in this field represents a coherent requester. If the bit is set, that requester is able to write to the GCR registers (this includes all registers within the Global, Core-Local, Core-Other, and Global Debug control blocks. The GIC is always writable by all requestors). If the bit is clear, any write request from that requestor to the GCR registers (Global, Core-Local, Core-Other, or Global Debug control blocks) will be dropped.	0xff

CPU 0 can control access to the GCR registers by other CPUs through the use of this field. Setting the corresponding bit to 1 enables writes to the GCR from that processor.

MIPS

9

On bring up your system CPU 0 can control access to the GCR registers by other CPUs through the use of this field. Each bit represents a processor or VPE in an MT system. Setting the corresponding bit to 1 enables writes to the GCR from that processor. Bits 0 – 3 correspond to Cores 0 – 1. Bits 4 and 5 correspond to IOCU 0 and 1.

GCR Revision Register

Register Fields		GCR Revision Register (GCR_REV Offset 0x0030)	Reset State
Name	Bits		
MAJOR_REV	15:8	This field reflects the major revision of the GCR block. A major revision might reflect the changes from one product generation to another.	Preset
MINOR_REV	7:0	This field reflects the minor revision of the GCR block. A minor revision might reflect the changes from one release to another.	Preset

The GCR Revision Register is useful when reporting issues to MIPS Technologies.

CM Error Detection

- **Two Types:**
  - Invalid requests to the GCR, GIC, or MMIO
  - Invalid intervention responses due to inconsistent L1 cache states
- **Reporting Error Registers**
  - Global CM Error Cause Register
  - Global CM Error Address Register
  - CM Error Multiple Register (Second Error)
- **Interrupt Response**
  - Global CM Error Mask Register

MIPS

11

The Coherency Manager can detect ,

+ An Invalid request to the GCR, The Global Interrupt Controller or Memory Management I/O controller. These invalid requests are usually caused by trying to access these controllers through something other than uncached accesses.

+ It can also detect an invalid intervention request due to inconsistent L1 cache states caused by improper coherence domain switch or inconsistent Cache Attributes between 2 CPUs for the same cache line.

+ The GIC has several registers

+++ that are use to communicate error conditions which I will cover in detail in upcoming slides.

+ All errors can be configured to cause an interrupt using the CM error mask register.

CM Error Detection

- **Types of errors**
  - 1 - GC_WR_ERR or 2 - GC_RD_ERR
    - GIC or GCR register accessed through cache address instead of an uncached address.
  - 3 - COH_WR_ERR or 4 - COH_RD_ERR
    - Coherent read or write error caused by GIC, GCR or MMIO areas accessed with a coherent address (address with a CCA of 4 or 5) instead of an uncached accessed.
  - 5 - MMIO_WR_ERR or 6 - MMIO_RD_ERR
    - Caused by an IOCU device trying to write to Memory Mapped I/O space.
  - 17 - INTVN_WR_ERR or 18 - INTVN_RD_ERR
    - Intervention read or write error caused by the Coherence Domain not properly entered or existed coherence domain or inconsistent CCA value for cache lines in a coherent domain.
  - 24 – 26 L2 Cache Errors

MIPS 12

There are 8 types of errors;

+ Types 1 and 2 are Global Control register write or read errors that are caused by trying to access the GCR or GIC register space in some way other than with an uncached address with a CCA of 2 causing a read of more than one word.

+ Types 3 and 4 are Coherent write or read errors that are caused by trying to access the GCR, GIC or MMIO with a coherent memory address (address with a CCA of 4 or 5).

+ Types 5 and 6 Memory mapped I/O read or write are caused by an I/O device trying to write to Memory Mapped I/O space. This could happen if an IO device is accessing itself or another IO device, by using the Coherency Manager.

+ types 17 and 18 are intervention read or write errors that can happen if the SW didn't follow the proper procedure for going into or out of the coherence domain. For example, if the Data Cache was not flushed when going from non-coherent to coherent, then it's possible for different cores to have incompatible states for the same line. Also, the cores could have different mappings for the same line, and this could lead to an error if one core mapped the line as coherent and the other mapped it as non-coherent.

+ types 24 – 26 are L2 cache errors. To accommodate L2 cache sizes greater than 1MB, when the index field is too small in the CP0 CacheErr register to hold all index tags, the information is captured in the CM2 Error GCRs. The previous CP0 CacheErr functionality is preserved for L2 cache sizes of 1MB and less.

Global CM Error Mask Register

Register Fields		Global CM Error Mask Register (GCR_ERROR_MASK Offset 0x0040)	Reset State
Name	Bits		
CM_ ERROR_MASK	31:0	Each bit in this field represents an Error Type. If the bit is set, an interrupt is generated if an error of that type is detected. If the bit is set, the transaction for Read-Type Errors completes with OK response to avoid double reporting of the error.	0x000A_002A This enables error numbers 2, 4, 5, 17, 19 (write errors cause interrupts; read errors provide error response)

MIPS

13

Each bit in the GCR Global CM Mask Register enables an interrupt for the corresponding interrupt type. The default interrupt mask is set to interrupt on write requests and intervention errors.

The interrupt asserts the SI_CM_Err signal. It is recommended that this signal be connected to SI_Cmint 0 of the GIC. The GIC section of this course will go into how the GIC is setup to do that.

If the bit is 0 the error will signal a Bus Error exception for a data access. The next slide tells you how to break it down to the exact eror.

BTW there is a typo in the default value. It sets bit 19 but there is no error type 19!

Global CM Error Cause Register

Register Fields		Global CM Error Cause Register (GCR_ERROR_CAUSE Offset 0x0048)	Reset State
Name	Bits		
CM_ ERROR_TYPE	31:27	Indicates type of error detected. When CM_ERROR_TYPE is zero, no errors have been detected. When CM_ERROR_TYPE is non-zero, another error will not be reloaded until a power-on reset or this field is written to 0.	0
CM_ERROR_ INFO	26:0	Information about the error. See next slide.	Undefined

MIPS

14

For either a Bus error exception or an interrupt the Global CM Error Cause Register will give you more information on the error. This register must be cleared after the exception or interrupt is handled.

One of the 8 error types given in a previous slide will be set in the cm error type field.

The cm error info gives you information on the error formatted based on the error type. The tables in the next slides will show you how to map the errors.

Global CM Error Cause Register Error Types 1 - 6

Format for types 1 – 6 (Memory Access Errors)

Bits	Description
17:15	CCA
14:12	Target Region (0: mem, 1:gcr, 2: gic, 3:mmio, 5:cpc)
11:7	OCP MCmd
6:3	Source TagID
2:0	Source port

MCmd Encoding	Description
0x01	Legacy Write
0x02	Legacy Read
0x08	Coherent Read Own
0x09	Coherent Read Share
0x0A	Coherent Read Discard
0x0B	Coherent Ready Share Always
0x0C	Coherent Upgrade
0x0D	Coherent Writeback
0x10	Coherent Copyback
0x11	Coherent Copyback Invalidate
0x12	Coherent Invalidate
0x13	Coherent Write Invalidate
0x14	Coherent Completion Sync

MIPS

15

There are 2 formats for the CM Error Info.

Here is the format for CM Error Types 1-6.

+Here is the meaning of the OCP MCmd:

Global CM Error Cause Register

Format for types 17 – 18 (Intervention Errors)

Bits	Description
20:19	Coherent state from core 3
18	Intervention SResp from core 3
17:16	Coherent state from core 2
15	Intervention SResp from core 2
14:13	Coherent state from core 1
12	Intervention SResp from core 1
11:10	Coherent state from core 0
9	Intervention SResp from core 0
8	Request was from a Store Conditional
7:3	OCP MCmd
2:0	Source port

Coherent state	
Encoding	Meaning
0	Invalid
1	Shared
2	Modified
3	Exclusive

Intervention SResp	
Encoding	Meaning
0	OK
1	Data (DVA)

MIPS

16

Here are the cm error info values for error types 16 through 31.

+Here is the encoding for the Coherent state errors

+ and here is the encoding for the intervention response errors

Global CM Error Cause Register

Format for types 24 – 26 (L2 Errors)

For CM Error types 24 – 26 (L2 Errors)	
Bits	**Description**
23	Multiple Uncorrectable
22:18	Instruction [4:0] associated with the error
17:16	Array type[1:0]: 00 = None 01 = Tag RAM single/double ECC error 10 = Data RAM single/double ECC error 11 = WS RAM uncorrectable dirty parity
15:12	DWord[3:0] with error, Array type=2 only
11:9	Way[2:0] associated with the error
8	Multi-way error for Tag or WS RAM
7:0	Syndrome associated with Tag or WS way, or Syndrome associated with Data Dword

Here are the encodings for the L2 error information.

Global CM Error Cause Register

Hex codes for action that caused the L2 cache error

Code	Description
00	L2_NOP
01	L2_ERR_CORR
02	L2_TAG_INV
03	L2_WS_CLEAN
04	L2_RD_MDYFY_WR
05	L2_WS_MRU
06	L2_EVICT_LN2
08	L2_EVICT
09	L2_REFL
0A	L2_RD
0B	L2_WR
0C	L2_EVICT_MRU

Bits	Description
0D	L2_SYNC
0E	L2_REFL_ERR
10	L2_INDX_WB_INV
11	L2_INDX_LD_TAG
12	L2_INDX_ST_TAG
13	L2_INDX_ST_DATA
14	L2_INDX_ST_ECC
18	L2_FTCH_AND_LCK
19	L2_HIT_INV
1A	L2_HIT_WB_INV
1B	L2_HIT_WB

MIPS

Here are the codes for the actions that caused the L2 error.

Global CM Error Address Registers

Register Fields		Global CM Error Address Register (GCR_ERROR_ADDR Offset 0x0050)	Reset State
Name	Bits		
CM_ERROR_ADDR	31:0	Request address which caused error. Loaded when the Global Error Cause Register is loaded. Bits 2:0 should always be 0.	Undefined

Register Fields		Global CM Error Multiple Register (GCR_ERROR_MULT Offset 0x0058)	Reset State
Name	Bits		
CM_ERROR_ 2ND	4:0	Type of second error. Loaded when the Global CM Error Cause Register has valid error information and a second error is detected.	0

MIPS

19

The Global CM error address register contains the address which caused the error. It should be cleared after the exception or interrupt is handled.

The Global CM Error Multiple Register gives the type of error if a second error is detected before the first one is handled. It must be cleared after the exception or interrupt has been handled.

Global CM Custom Status Register

Register Fields		Custom Status Register (GCR_CUSTOM_STATUS Offset 0x0068)	Reset State
Name	Bits		
GGU_EX	0	If this bit is set, the Custom GCRs connected to the CM2.	1

MIPS

20

The CM2 provides the capability of including custom (user-defined) GCRs. GGU_EX indicates if the customer GCR registers exist or not.

The GCR_CUSTOM_BASE register sets the physical address of the base address of a 64KByte slice of memory space, where requests are directed to the Custom GCRs.

GGU_EN enables the redirection of the memory request to the custom CGR memory registers instead of memory.

Global CM Custom Base Register

Register Fields		GCR Custom Base Register (GCR_CUSTOM_BASE Offset 0x0060)	Reset State
Name	Bits		
GCR_CUSTOM_BASE	31:16	This field sets the base address of the 64KB GCR custom user-defined block of the interAptiv MPS.	Undefined
GGU_EN	0	If this bit is set, the Address Region for the Custom GCR is enabled. This bit cannot be set to 1 if GGU_EX = 0, indicating that a custom GCR is not attached to the CM.	0

MIPS

21

The CM2 provides the capability of including custom (user-defined) GCRs. GGU_EX indicates if the customer GCR registers exist or not.

The GCR_CUSTOM_BASE register sets the physical address of the base address of a 64KByte slice of memory space, where requests are directed to the Custom GCRs.

GGU_EN enables the redirection of the memory request to the custom CGR memory registers instead of memory.

Global Interrupt Controller Status Register

Register Fields		Global Interrupt Controller Status Register (GCR_GIC_STATUS Offset 0x00D0)	Reset State
Name	Bits		
GIC_EX	0	If this bit is set, the GIC is connected to the CM.	1

MIPS

22

The GIC is configured into the system at IP build time. The chip designer can decide to include a GIC or not. GIC was configured into the system if the GIC_EX bit is set.

Global L2-Only Sync Base Register

Register Fields		GCR L2-Only Sync Base Register (GCR_L2_ONLY_SYNC_BASE Offset 0x0070)	Reset State
Name	Bits		
GCR_L2_ONLY_SYNC_BASE	31:12	This field sets the base address of the 4KB GCR L2 only Sync of the interAptiv MPS.	Undefined
CM_L2_ONLY_SYNC_EN	0	If this bit is set, the CM will treat an uncached write request as a L2 only Sync. If set to 0, the CM will treat the uncached write as a regular uncached request.	0

MIPS

23

The CM2 adds the ability to issue a barrier-sync to the L2 without executing a SYNC instruction, thus reducing the latency incurred for the sync. The L2-only sync provides a mechanism to guarantee that an uncached request does not pass previous cached requests in the L2 pipeline. For example, the L2-only SYNC can be used between a L2 HitWB cacheop and a subsequent uncached write, to ensure that the uncached write does not pass the writeback from the L2.

CM_L2_ONLY_SYNC_EN (bit 0) of the register must be set to a 1 for this feature to be enabled. The address match is performed on a 4KB boundary. An uncached write request address [31:12] that matches the address [31:12] in the GCR_L2_ONLY_SYNC_BASE will cause the CM2 to treat the uncached write request as an L2-only SYNC.

Global Interrupt Controller Base Address Register

Register Fields		Global Interrupt Controller Base Address Register (GCR_GIC_BASE Offset 0x0080)	Reset State
Name	Bits		
GIC_ BaseAddress	31:17	This field sets the base address of the 128KB Global Interrupt Controller.	0
GIC_EN	0	If this bit is set, the Address Region for the GIC is enabled. This bit can not be set to 1 if GIC_EX = 0, indicating that a GIC is not attached to the CM.	0

MIPS

24

The Global Interrupt Controller Base Address Register is dependent on the GIC_EX bit being set in the Global Interrupt Controller Status Register. Only if it is set is the Global Interrupt Controller Base Address Register valid. This register does 2 things, it is used to set the address of the GIC and to enable the GIC. The address is system dependent. You should consult your system specification or system designer to determine what address in your memory map the GIC is located at. That address must be on a 128k byte boundary. In other words the lower 17 bit of the address must be zeros. This register should be written at system initialization time with the value of that address.

In addition the GIC_EN bit controls the enabling of the GIC. Once the GIC has been programmed this bit needs to be set to begin receiving interrupts.

Cache Revision Register

- Is there an L2 Cache?

Register Fields		Cache Revision Register (GCR_CACHE_REV Offset 0x00E0)	Reset State
Name	Bits		
MAJOR_REV	15:8	This field reflects the major revision of the Cache block attached to the coherence Cluster. A major revision might reflect the changes from one product generation to another.	Preset
MINOR_REV	7:0	This field reflects the minor revision of the Cache block attached to the coherence Cluster. A minor revision might reflect the changes from one release to another.	Preset

MIPS

25

The cache revision register tells you the Major and Minor revision of the L2 cache.

Cluster Power Controller Status Register Register

Register Fields		Global Power Controller Status Register (GCR_GPC_STATUS Offset 0x00D0)	Reset State
Name	Bits		
CPC_EX	0	If this bit is set, the CPC is connected to the CM2.	1

MIPS

26

The Cluster power controller is configured into the system at IP build time. This register tells you if a CPC was configured into the system. NOTE: the CPC is always part of the core so this will be set to 1.

IOCU Revision Register

- Is there an IOCU?

Register Fields		IOCU Revision Register (GCR_IOCU1_REV Offset 0x0200)	Reset State
Name	Bits		
MAJOR_REV	15:8	This field reflects the major revision of the IOCU attached to the CM. A major revision might reflect the changes from one product generation to another. The value of 0x0 means that no IOCU is attached.	Preset
MINOR_REV	7:0	This field reflects the minor revision of the IOCU attached to the CM. A minor revision might reflect the changes from one release to another.	Preset

The I/O Coherency Unit is configured into the system at IP build time. This register tells you if a IOCU was configured into the system. If the Major Revision field is 0 then there is no IOCU.

Address Regions

- **Default Address Region**
- **0, 4, Additional Address Regions**
  - Maps address to specific controllers
    - Memory – CM controlled region
      - CCA Override for L2 to OCP Coherency control
    - IOCU – IOCU controlled region
      - non-coherent and uncached transactions
- **Region defined by 2 GCR registers**
  - Base Address
  - Mask to determine region size

MIPS

28

There are up to 5 programmable Address Regions that can be programmed to support memory mapped I/O or memory storage regions. The default region that covers any main memory not covered by another region.

+ A region that is mapped to memory storage can be configured to have a different CCA for the L2 to OCP interface than the L1 to L2 interface.

+ Reads and write requests to a region that has been mapped to the IOCU go directly to the IOCU as non-coherent and uncached transactions.

+ Each region is defined by a pair of GCR registers a Base Address Register and an Address Mask Register that determines its size.

The next slides will cover the configuration of these regions using these registers.

CM Region Registers

- Addresses of Register Pair Offsets into the GCR address map

Region	Base Address Register Offset	Address Mask Register
0	0x0090	0x0098
1	0x00A0	0x00A8
2	0x00B0	0x00B8
3	0x00C0	0x00C8

This table shows you the offsets into the GCR where the “region registers” are.

CM Region [0 - 3] Base Address Register

- **Base address aligned on 64K boundary**

Register Fields		Base Address Register (GCR_REGn_BASE Offsets 0x0090, 0x00A0, 0x00B0, 0x00C0)	Reset State
Name	Bits		
CM_Region_ BaseAddress	31:16	This field sets the base address of the Coherence Memory Region.	0

MIPS

30

You program the regions physical address by writing to the regions CM Base Address Register. Since CM Address Regions must start on 64K byte boundaries only addresses with the lower 16 bits set to 0 are valid. The lower 16 bits will always read back as zeros.

NOTE: CM2 has only 4 regions so address registers for the other regions have been dropped.

CM Region[0 - 3] Address Mask Register

Register Fields		Address Mask Register (GCR_REGn_MASK Offsets 0x0098, 0x00A8, 0x00B8, 0x00C8)	Reset State
Name	Bits		
CM_Region_ Address_Mask	31:16	This field is used to set the size of the CM Region. The only allowed values in this register are contiguous sets of leading 0x1's. An 0x1 preceded by a 0x0 is not allowed (e.g., the value of 0xfff0 is allowed, but the value 0xffef is not allowed).	UD

MIPS

31

CM_Region_Address_Mask determines the size of the region. This field is used along with its equivalent CM Region Base Address Register.

The request physical address is logically ANDed with the value of this register and ANDed with the value of the region's Base Address Register to see if the requested address is in range of this region. If it is the request it is routed to this CM region.

NOTE: CM2 has only 4 regions so mask registers for the other regions have been dropped.

CM Region[0 - 3] Address Mask Register

Register Fields		Address Mask Register (GCR_REGx_MASK Offsets 0x0098, 0x00A8, 0x00B8, 0x00C8)	Reset State
Name	Bits		
CCA_ OVERRIDE_VALUE	7-5	Used with CCA_OVERRIDE_ENABLE to force the Cache Coherence Attribute (CCA) value for transactions on the L2/Memory OCP.	0
CCA_ OVERRIDE_ENABLE	4	If set CCA_Override_Enable is set to 1 and CM_REGION_TARGET is set to Memory, then transactions with addresses that map to this region will have a CCA value set to CCA_Override_Value when driven to L2/Memory.	0

Encoding	Name	Description
0x0	WT	Write Through
0x2	UC	Uncached
0x3	WB	Write Back non-coherent
0x4	CWBE	Mapped to WB
0x5	CWB	Mapped to WB
0x7	UCA	Uncached Accelerated

MIPS

32

Continuing with the Address Mask Register:

CCA_Override_Value and CCA_Override_Enable are used in conjunction with CM_REGION_TARGET memory region. If the CCA_OVERRIDE_ENABLE is not set then the CCA for the L2 to memory is the same as the CCA form the L1 caches to the L2. If the CCA_OVERRIDE_EENABLE bit is set then the CCA value in the CCA_OVERRIDE_VALUE field will be used for the CCA from L2 to Memory.

CM Region[0 - 3] Address Mask Register

Register Fields		Address Mask Register (GCR_REGn_MASK Offsets 0x0098, 0x00A8, 0x00B8, 0x00C8)	Reset State
Name	Bits		
CM_Region_ TARGET	1:0	Maps this region to the specified device:. Encoding / Meaning: 0 – Disabled 1 – Memory 2 – 1st IOCU 3 – 2nd IOCU	0

More on the Address Mask Register:

The CM_REGION_TARGET directs the region to Memory or the I/O Coherency Unit.

NOTE: A second IOCU has been added with CM2.

Core Local +0x2000 and Core Other Block +0x4000

Register Offset	Name	Description
0x0008	Core Local Coherence Control Register	Controls which coherent Intervention transactions apply to the local core.
0x0010	Core Local Config Register	Indicates the number of VPEs in this core, etc.
0x0018	Core Other Addressing Register (only in Core-Local block)	Used to access the registers of another core.
0x0020	Core Local Reset Exception Base Register	Sets the Reset Exception Base for the local core.
0x0028	Core Local Identification Register	Indicates the interAptiv Number of the local core.

MIPS

34

The next 2 blocks in the Control Address space are the Core-Local and Core-Other Blocks. These blocks are local to a specific CPU.

The Core-Local block can be accessed at offset hex 2000 from the GCR Base Address. Using this address all CPUs will see their own register set due to the magic of hardware. The hardware knows what CPU the address is being requested from and it directs the request to the correct Core-local block.

The Core-Other block can be accessed at offset hex 4000 from the GCR Base Address This block gives any CPU a way of accessing another CPU's CM local registers. To do this the CPU writes to its own Core Other Address register. I'll cover this in a bit.

I will cover each register in-turn in slides that follow.

Core Local Coherence Control Register

- **Defining the Coherence Domain**
  - Allowing another Core to send interventions to this CPU
  - Setting Coherency Enable for the Local Core

Register Fields		Core Local Coherence Control Register (GCR_Cx_COHERENCE Offset 0x0008)	Reset State
Name	Bits		
COH_ DOMAIN_EN	7:0	Each bit in this field represents a coherent requester within the CPS. Setting a bit within this field will enable interventions to this Core from that requester. The requestor bit which represents the local core is used to enable or disable coherence mode in the local core. Changing the coherence mode for a local core from 0x1 to 0x0 can only be done after flushing and invalidating all the cache lines in the core; otherwise, the system behavior is UNDEFINED.	0x0

Bit 0	Core 0
Bit 1	Core 1
Bit 2	Core 2
Bit 3	Core 3
Bit 4	IOCU 0
Bit 5	IOCU 1

```
// enabling coherency between this core and other cores
li  a0, 0x0f
sw  a0, (CORE_LOCAL_CONTROL_BLOCK | GCR_CL_COHERENCE) (gcr_addr)
lw  a0, (CORE_LOCAL_CONTROL_BLOCK | GCR_CL_COHERENCE) (gcr_addr)
sync
```

MIPS

35

The Core Local Coherence Control Register sets up the Coherent domain for the CPU using the Coherency Domain Enable field. This field consists of 8 bits with each bit corresponding to a Core in the system. Setting a bit enables interventions from the corresponding Core. The bit that corresponds to the local CPU enables coherence mode for the local CPU.

This register is used by the local CPU to enter the Coherence Domain and to exit the Coherence Domain.

The CPU needs to follow the entering the domain sequence of initializing the Caches, Enabling the interventions from other cores and enabling coherence mode for itself at reset and power up of the CPU.

Before a power down of the CPU, the CPU needs to follow the exiting the domain sequence of disabling interventions , flushing its data cache, invalidating the its caches and disabling coherence mode.

CM Core Config Register

- Defines type of processor
- Defines number of Virtual Processing Elements

Register Fields		Core Local Config Register (GCR_Cx_CONFIG Offset 0x0010)	Reset State
Name	Bits		
IOCU_TYPE	11:10	Encoding: 0x0 — This is a interAptiv core and not an IOCU 0x1 — This is a non-caching IOCU (no intervention port) 0x2 — This is a caching IOCU (not implemented by MIPS)	IP Config Value
PVPE	9:0	Encoding: 0x0 — 1 VPE 0x1 — 2VPSs	IP Config Value

MIPS

36

The CM Core Config register is a read-only register that gives you information about the CPU.

The first field IOCU_TYPE will be set to 0 because a CPU is not an IOCU.

The second field PVPE will give the number of Virtual Processing elements for an MT system. If this is a single Processor system this field will be set to 0. If this is a Multi-Threaded system the number of VPES in the Core will the number in this field plus 1.

Core-Other Addressing Register

- **CPU number of the other CPU referred to by the core-other register block**

Register Fields		Core-Other Addressing Register (GCR_Cx_OTHER Offset 0x0018)	Reset State
Name	Bits		
CoreNum	31:16	CoreNum of the register set to be accessed in the Core-Other address space.	0

The Core Number field of the “Core other addressing register” needs to be written with the number of the other core who’s local registers you wish to access. Once this is written you can access that CPU’s CM Local registers through the Core-Other register block.

Core Local Reset Exception Base Register

- Setting Boot Exception vector

Register Fields		Core-Local Reset Exception Base Register (GCR_Cx_RESET_BASE Offset 0x0020)	Reset State
Name	Bits		
BEVEXCBase	31:12	Bits [31:12] of the virtual address that the local core will use as the exception base in the boot environment (C0P0 StatusBEV=1).	0

MIPS

38

The BEVEXCBase field in the Core Local Reset Exception Base Register controls where the CPU will fetch the first instruction when it is powered up and reset. If you wanted to have CPUs that used different boot code you could access this register through the core-other group from another CPU and set the boot address for this CPU. There will be more on this register the EVA section of this course.

Core Local Identification Register

- Tells the CPU its number

Register Fields		Core-Local Identification Register (GCR_Cx_ID Offset 0x0028)	Reset State
Name	Bits		
CoreNum	31:0	This number is used as an index to the registers within the GCR when accessing the Core-local control block for this core.	

The Core Local Identification Register is a read only register that tells the CPU which CPU it is. The code will use this number to setup things like enabling coherency in the Core Local Coherence Control Register.

Core Local Reset Exception Extended Base Register

Register Fields		Core-Local Reset Exception Extended Base Register (GCR_Cx_RESET_EXT_BASE Offset 0x0030)	Reset State
Name	Bits		
EVAReset	31	Indication to the local core to not use the legacy reset (RW)	IPCV
LegacyUseExceptionBase	30	Indication to the core to not use the ExceptionBase (RW)	IPCV
BEVExceptionBaseMask	27:20	Bits [27:20] of the virtual address that the local core will use as the exception base in the boot environment (C0P0 StatusBEV = 1). (RW)	IPCV
BEVExceptionBasePA	7:1	Bits [35:29] of the physical address that the local core will use as the exception base in the boot environment (C0P0 StatusBEV = 1). (RW)	IPCV
PRESENT	0	Reads as 0x1. Writes ignored. (R)	1

This register is an extension to the Core-Local Reset Exception Base Register. The value is used for placing the exception vectors within the virtual address map during core boot-up time. This will be covered in more detail in the segmentation and EVA section.

Core Local TCID Priority Register

- Setting for the TC Priority

Register Fields		Core Local TCID Priority Register (GCR_Cx_TCID_n_PRIORITY Offset 0x0040, 0x0048, 0x0050, 0x0058, 0x0060, 0x0068, 0x0070, 0x0078, 0x0080 for TCID 0 to 8)	Reset State
Name	Bits		
TCID_ PRIORITY	1:0	TCID priority value of 2 bits. Priority value of 0 has the lowest priority. Priority value of 3 has the highest priority. (RW)	0

There are nine thread context ID registers one for each possible thread. the 2 bits of Priority value is passed through with memory transactions to the OCP bus. This enables the system to identify the TC that is associated with the bus transaction.

Additional NOTE:

A new set of GCR registers,
GCR_CL_TCID_n_PRIORITY/GCR_CO_TCID_n_PRIORITY have been added to

the Core Local/Core Other Block Register Map at offset 0x0040 to 0x0080. The IOCU_TYPE (bit 11:10) in the

GCR_CL_CONFIG/GCR_CO_CONFIG has to be set value of 0,to enable this feature.

The *GCR_CL_TCID_n_PRIORITY/GCR_CO_TCID_n_PRIORITY register is accessed based on the TCID n,*

and the 2 bits of Priority value is passed through the MConnID.