Hercules™ ARM® Cortex®-R4 System Architecture

System Control Coprocessor (CP15)

System Control Coprocessor (CP15)

The purpose of the system control coprocessor, CP15, is to control and provide *status information* for the functions implemented in the processor. The main functions of the system control coprocessor are:

- Overall system control and configuration
- Cache configuration and management
- Memory Protection Unit (MPU) configuration and management
- System performance monitoring.

Note: Only some of the functions can be accessed in User Mode

System Control Coprocessor (CP15) functions

Table 4-1 System control coprocessor register functions

Function	Register/operation	Reference to description
System control and configuration	Control	*c1, System Control Register* on page 4-44
	Auxiliary control	*Auxiliary Control Registers* on page 4-47
	Coprocessor Access Control	*c1, Coprocessor Access Register* on page 4-55
	Main ID[a]	*c0, Main ID Register* on page 4-17
	Product Feature IDs	*The Processor Feature Registers* on page 4-22 *c0, Debug Feature Register 0* on page 4-24 *c0, Auxiliary Feature Register 0* on page 4-25 *Memory Model Feature Registers* on page 4-26 *Instruction Set Attributes Registers* on page 4-32
	Multiprocessor ID	*c0, Multiprocessor ID Register* on page 4-21
	Slave Port Control	*c11, Slave Port Control Register* on page 4-75
	Context ID	*c13, Context ID Register* on page 4-76
	FCSE PID	*c13, FCSE PID Register* on page 4-76
Software compatibility	Thread And Process ID	*c13, Thread and Process ID Registers* on page 4-77
MPU control and configuration	Data Fault Status	*c5, Data Fault Status Register* on page 4-57
	Auxiliary Fault Status	*c5, Auxiliary Fault Status Registers* on page 4-59
	Instruction Fault Status	*c5, Instruction Fault Status Register* on page 4-58
	Instruction Fault Address	*c6, Instruction Fault Address Register* on page 4-61
	Data Fault Address	*c6, Data Fault Address Register* on page 4-61

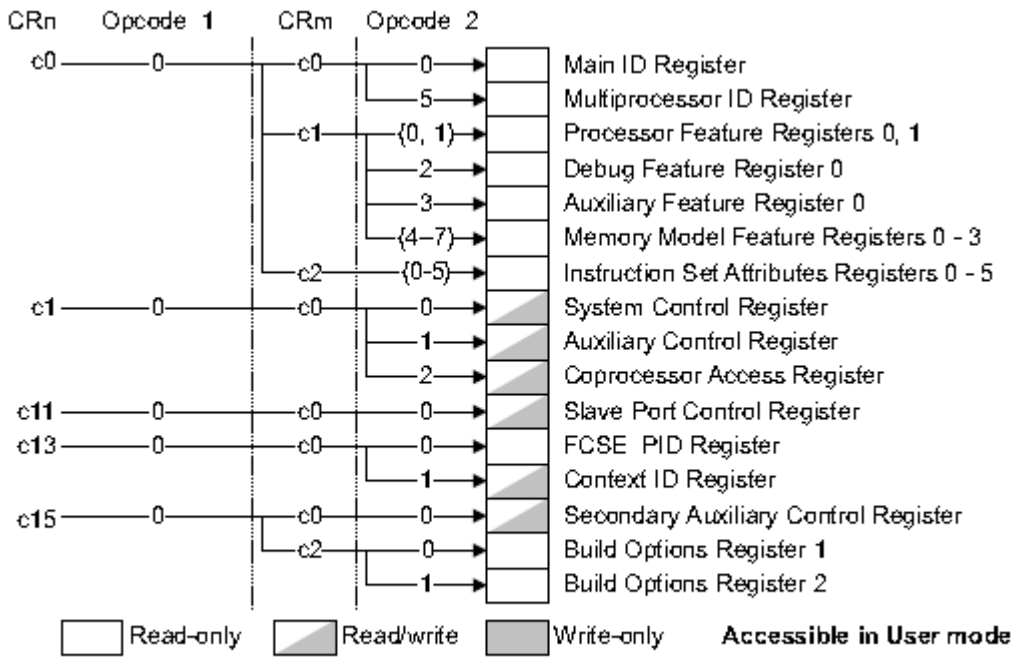


Figure 4-1 System control and configuration registers

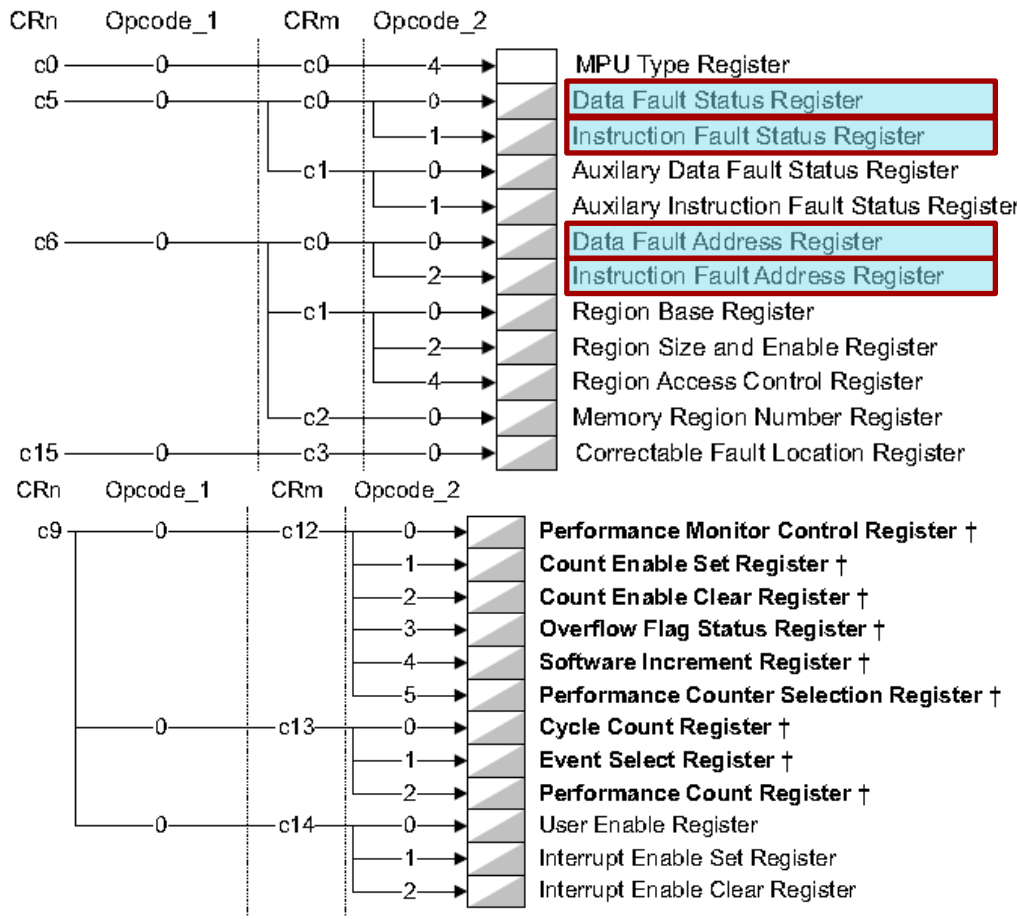
<MRC|MCR>{<cond>} <cp_num>,<opc_1>,Rd,CRn,CRm,<opc_2>

System Control Coprocessor (CP15)

	Data Fault Address	*c6, Data Fault Address Register* on page 4-61
	MPU Type	*c0, MPU Type Register* on page 4-21
	Region Base Address	*c6, MPU Region Base Address Registers* on page 4-62
	Region Size and Enable	*c6, MPU Region Size and Enable Registers* on page 4-63
	Region Access Control	*c6, MPU Region Access Control Registers* on page 4-65
	Memory Region Number	*c6, MPU Memory Region Number Register* on page 4-67

Table 4-1 System control coprocessor register functions (continued)

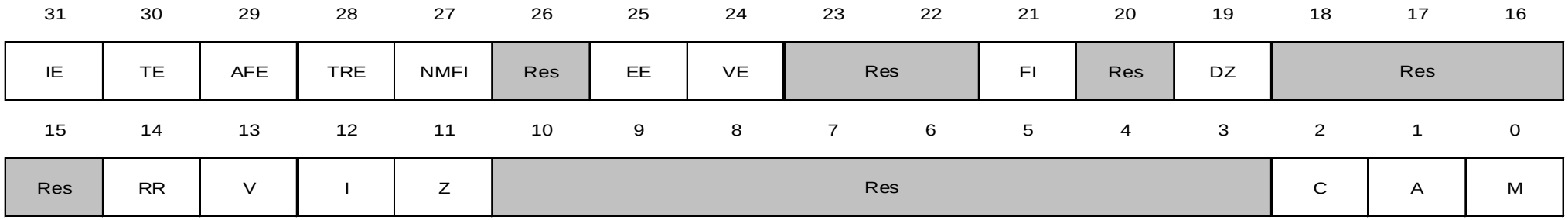
Function	Register/operation	Reference to description
Cache control and configuration	Cache Type	*c0, Cache Type Register* on page 4-18
	Current Cache Size Identification	*c0, Current Cache Size Identification Register* on page 4-40
	Current Cache Level	*c0, Current Cache Level ID Register* on page 4-42
	Cache Size Selection	*c0, Cache Size Selection Register* on page 4-43
	c7, Cache Operations	*Cache operations* on page 4-68
	c15, Invalidate all data cache	
TCM control and configuration	TCM Status	*c0, TCM Type Register* on page 4-19
	Region	*c9, BTCM Region Register* on page 4-72 *c9, TCM Selection Register* on page 4-75
System performance monitoring	Performance monitoring	Chapter 6 *Events and Performance Monitor*
Validation	System validation	*Validation Registers* on page 4-78



<MRC|MCR>{<cond>} <cp_num>,<opc_1>,Rd,CRn,CRm,<opc_2>

System Control Coprocessor (CP15)

System Control Register: This register provides control and configuration information



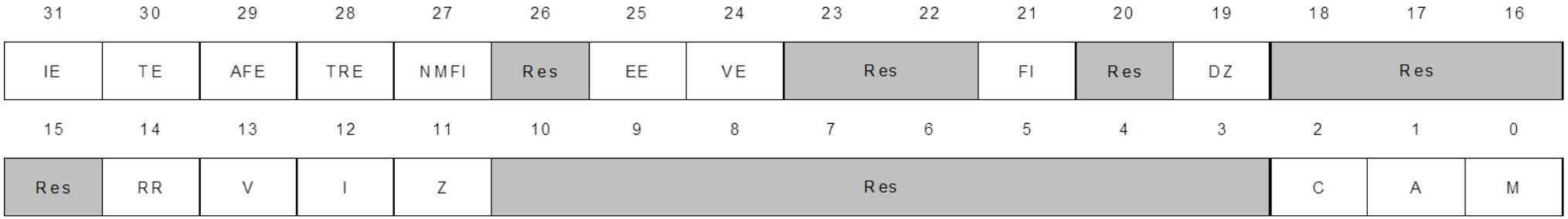
Read: MRC P15, 0, <Rd>, C1, C0, 0

Write: MCR P15, 0, <Rd>, C1, C0, 0

IE:	1	**Endianess** (TMS570 => Big Endian)
TE:	x	Thumb exception enable
AFE:	0	Access Flag enable (always 0)
TRE:	0	TEX Remap enable (always 0)
NMFI:	x	**Non-maskable fast interrupt enable**
EE:	x	Determines how the E bit in CPSR is set on an exception
VE:	x	**Configure vectored interrupt**

System Control Coprocessor (CP15)

System Control Register (continued)



FI:	1	**Fast Interrupt Enable**
DZ:	x	Divide by zero
RR:	x	Controls of the replacement strategy for instruction/data caches
V:	x	**Determines the location of the exception vectors (TMS570 => 0x0)**
I:	x	Enables L1 instruction cache
Z:	1	Branch Prediction; Controlled by Aux Control register
C:	x	Enabled L1 data cache
A:	x	Enables strict alignment checking for data accesses
M:	x	**Enables MPU**

System Control Coprocessor (CP15)

Auxiliary Control Register: This register provides control for branch prediction, performance features and error parity logic

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DICDI	DIB2DI	DIB1DI	DIADI	D1TCMPCEN	D0TCMPCEN	ITCMPCEN	AXISCEN	AXISCUEN	DILSM	DEOLP	DBHE	FRCDIS	Res	RSDIS	BP
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BP	DBWR	DLFO	ERPEG	DNCH	FOR A	FWT	FDSnS	sMOV	DILSM	DA	HER	Res	D1TCMECEN	D0TCMECEN	ITCMECEN

Read: MRC P15, 0, <Rd>, C1, C0, 1
Write: MCR P15, 0, <Rd>, C1, C0, 1

DICDI:	x	Disable Case C dual issue control
DIB2DI:	x	Disable Case B2 dual issue control
DIB1DI:	x	Disable Case B1 dual issue control
DIADI:	x	Disable Case A dual issue control
D1TCMP:	**x**	**Data 1 TCM parity check enable**
D0TCMP:	**x**	**Data 0 TCM parity check enable**
ITCMPCEN:	x	ITCM parity check enable

System Control Coprocessor (CP15)

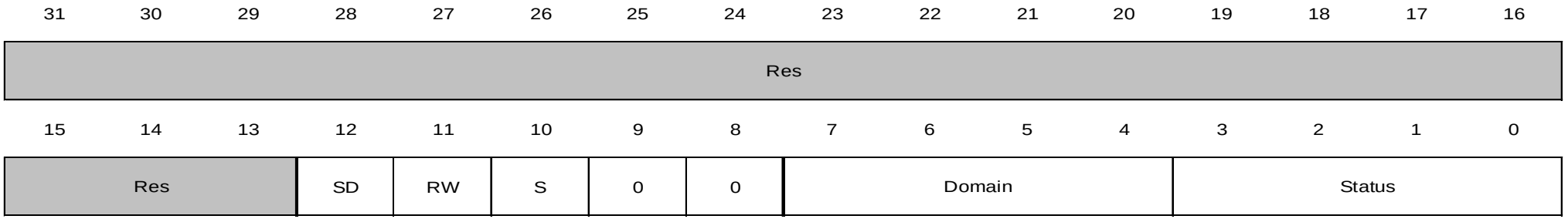
Auxiliary Control Register (continued)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DICDI	DIB2DI	DIB1DI	DIADI	D1TCMPCEN	D0TCMPCEN	ITCMPCEN	AXISCEN	AXISCUEN	DILSM	DEOLP	DBHE	FRCDIS	Res	RSDIS	BP
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BP	DBWR	DLFO	ERPEG	DNCH	FOR A	FWT	FDSnS	sMOV	DILSM	DA	HER	Res	D1TCMECEN	D0TCMECEN	ITCMECEN

DILS:	x	Disable low interrupt latency on all load/store instructions
DA:	x	Disable abort on cache parity error
HER:	x	Enable hardware recovery from cache parity error
D1TCMECEN:	**x**	**Enable D1TCM error check**
D0TCMECEN:	**x**	**Enable D0TCM error check**
ITCMECEN:	**x**	**Enable ITCM error check**
ITCMPCEN:	**x**	**Enable Flash error check (ECC)**

System Control Coprocessor (CP15)

Data Fault Status Register: This register provides information about the last data fault

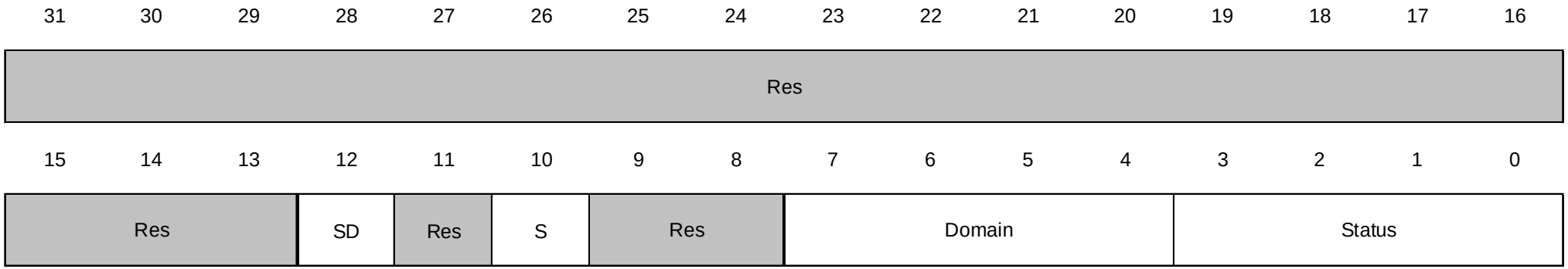


Read: MRC P15, 0, <Rd>, C5, C0, 0
Write: MCR P15, 0, <Rd>, C5, C0, 0

SD:	x	Indicates if an AXI decode or slave error caused an imprecise/precise abort. Only valid for external (to the processor) aborts.
RW:	x	Indicates whether a read or write access caused an abort
S:	**x**	**Part of status field**
Domain:	0000b	Domains are not implemented on this processor
Status:	**x**	**Indicates the type of exception generated. Use in conjunction with SD and S**

System Control Coprocessor (CP15)

Instruction Fault Status Register: This register provides information about the last instruction fault



Read: MRC P15, 0, <Rd>, C5, C0, 1
Write: MCR P15, 0, <Rd>, C5, C0, 1

SD:	x	Indicates if an AXI decode or slave error caused an imprecise/precise abort. Only valid for external (to the processor) aborts.
S:	**x**	**Part of status field**
Domain:	0000b	Domains are not implemented on this processor
Status:	**x**	**Indicates the type of exception generated. Use in conjunction with SD and S**

System Control Coprocessor (CP15)

Auxiliary Fault Status Registers: These registers provide parity information about the last data/instruction fault

Read: MRC P15, 0, <Rd>, C5, C1, 0	Auxiliary Data Fault Status Register
Write: MCR P15, 0, <Rd>, C5, C1, 0	Auxiliary Data Fault Status Register
Read: MRC P15, 0, <Rd>, C5, C1, 1	Auxiliary Instruction Fault Status Register
Write: MCR P15, 0, <Rd>, C5, C1, 1	Auxiliary Instruction Fault Status Register

CacheWay:	x	Indicates the cache way(s) in which the parity error occurred (only valid for data cache store parity errors)
Side:	x	Indicates the source of the parity error (Cache, ITCM, DTCM)
Index:	x	Indicates the index value of the access giving the parity error Returns 0x0 on instruction fault status register and TCM access

System Control Coprocessor (CP15)

Data/Instruction Fault Address Registers

The *Data Fault Address Register (**DFAR**)*

- Provides the address of the fault when a precise abort occurred.

The *Instruction Fault Address Register (**IFAR**):*

- Provides the address of the instruction which caused a prefetch abort

Read: MRC P15, 0, <Rd>, C6, C0, 0	Data Fault Address Register
Write: MCR P15, 0, <Rd>, C6, C0, 0	Data Fault Address Register
Read: MRC P15, 0, <Rd>, C5, C1, 1	Instruction Fault Address Register
Write: MCR P15, 0, <Rd>, C5, C1, 1	Instruction Fault Address Register

For more information

- **SafeTI Web Page: www.ti.com/safeti**
- **Hercules Web Page: www.ti.com/hercules**
  - Data sheets
  - Technical Reference Manual
  - Application notes
  - Software & tools downloads and updates
  - Order evaluation and development kits
- **Hercules Safety Microcontrollers Training Series training.ti.com/hercules**
  - Cortex-R Processor Architecture
  - Peripherals
  - Software
  - Functional Safety
- **For questions about this training, refer to the Engineer-2-Engineer Support Forum www.ti.com/hercules-support**
  - News and announcements
  - Ask technical questions
  - Search for technical content