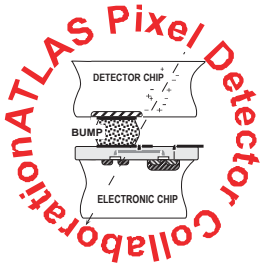

MCC-I2.1 Specifications

MCC Design Group

Register Bank

Revision:	1.0
Reference:	ATLAS ID-xx
Created:	08 December 2003
Last modified:	1 September 2004 10:57
Edited By:	R. Beccherle and G. Darbo
	ATLAS/Pixel Collaboration

Abstract

This document describes the Register Bank of the MCC-I2.1.

Table Of Contents

List of Figures

List of Tables

2 Register Bank

2.1 Register Set

The *MCC* architecture foresees up to 16 general-purpose 16-bit registers, out of which only 11 have been implemented.

The Register Bank is the *MCC* block that contains all the configurations and status register needed to correctly operate the system. Those registers are readable and writable by the *ROD* and some of them are modified by the *MCC* itself during its normal operation.

The LV1 register is not completely writable with a WrRegister command. Only LV1[11:8] can be written because LV1[7:0] is mapped to the Lv1Id counter inside the TTC.

As can be seen in Table 2-1 not all bits are always used and non existing bits are always read back as '0'. In order to clear a register content one can write all zeroes into it or to issue a GlobalResetMCC which brings all registers to their default value (see Table 2-4 for the default value of all registers).

Table 2-1 MCC internal registers.

Register	Address	Content	Description
CSR	0000	ssss,ssss,ccc-,-ccc	Control Status Register
LV1	0001	----,CCCC,LLLL,LLLL	Trigger Register
FEEN	0010	dddd,dddd,dddd,dddd	Front-End Enable
WFE	0011	dddd,dddd,dddd,dddd	Warning from FE
WMCC	0100	dddd,dddd,dddd,dddd	Warning from MCC Receiver
CNT	0101	dddd,dddd,dddd,dccc	Control & Data Counters, Delay Line settings
CAL	0110	----,-pRR,RRDD,DDDD	Calibration Register
PEF	0111	SSSS,LLLL,BBBB,BBBB	Pending Event FIFO
SBSR	1000	ssss,ssss,ssss,ssss	ScoreBoard Status Register
WBITD	1001	---w,wwww,wwww,wwww	Warning Bit disable
WRECD	1010	wwww,wwww,wwww,wwww	Wrning Receiver Disable

Note
B,c,d,D,l,n,p,R,s,S,w: used data bit
-: Non existing bit. It is always read back a '0'

Registers can be loaded by the *ROD* using a WrRegister command followed by a data stream as shown in Figure 2-1.

The incoming data stream is shifted into a receiving shift register (Shadow I/O Shift Register) and eventually copied to the destination register. The use of the Shadow I/O Shift Register, instead of directly shifting of the data into the destination register, avoids any temporary change of those bits that would not be modified by the read or write operation.

In case of a write operation the MCC nothing will be sent back to the *ROD*.

A read operation is started when the *ROD* sends the RdRegister command. When the *MCC* gets the read command, it first copies the addressed register contents to the Shadow I/O Shift Register and then it directs its content to the MCC-DTO output pins. Any output data stream is preceded by the Header ("11101") to wake up and synchronize the receivers in the *ROD* (See Figure 2-2).

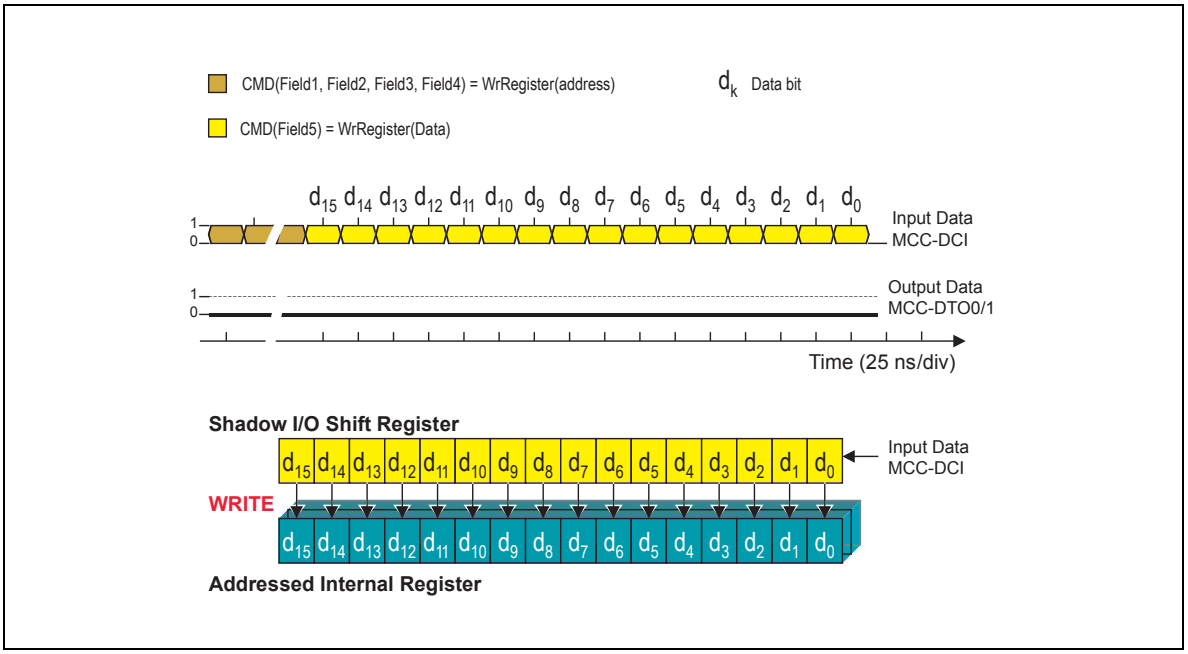

Figure 2-1 Internal Register: Write operation.

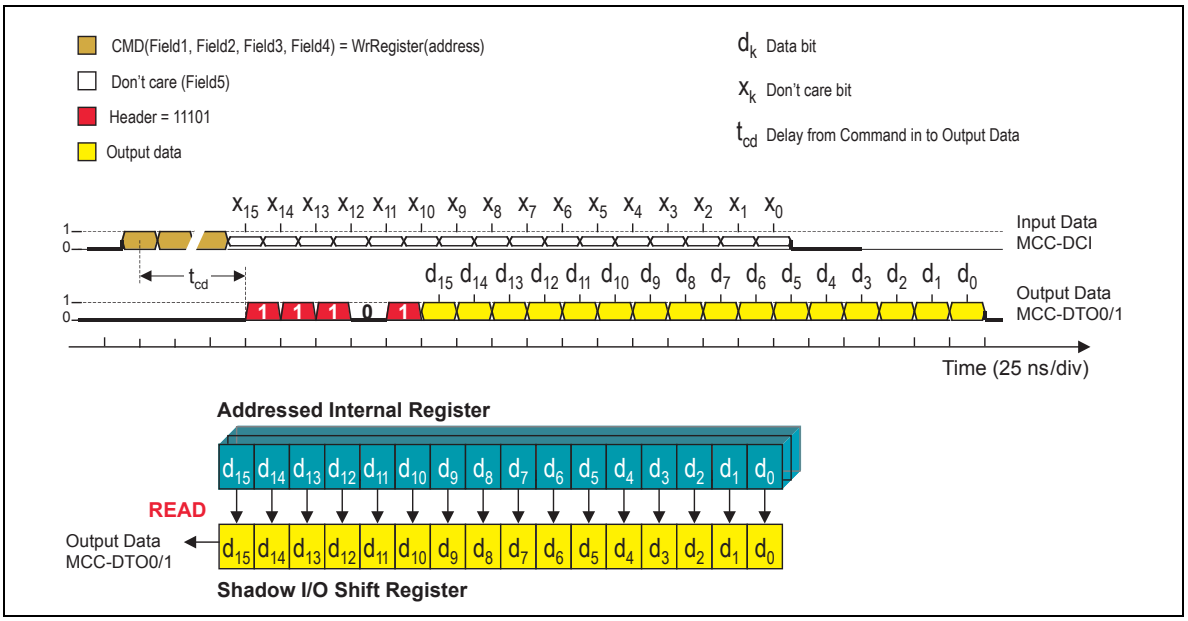

Figure 2-2 Internal Register: Read operation.

Almost all register contents can be overwritten by a WrRegister command and the new value read back directly by a RdRegister command with two exceptions: LV1 and PEF registers.
The bit field LV1 <7:0> cannot be overwritten by a write command, but can only be cleared by GlobalResetMCC or ECR commands.
The PEF register is mapped to the PendingLv1Fifo; each time a WrRegister command with PEF address is issued a new word is written to the *FIFO* and the write pointer is incremented. Each time a RdRegister command with PEF address is issued a word is extracted from the *FIFO* and the read pointer is incremented.

The CommandDecoder will ignore a WrRegister command to a non existing register address.

A RdRegister command addressing a non existing register will cuse the *MCC* to respond with the Header followed by "0000_0000_0000_0000".

After a GlobalResetMCC command both read and write pointers of the PEF are set to zero (first location of the PEF).

In case a GlobalResetMCC command is followed by a WrRegister(PEF), then the RdRegister(PEF) returns the last value written into the *FIFO* (first location of the PEF).

Here follows a list of the implemented *MCC* registers.

2.1.1 CSR: Control & Status Register

The Control and Status Register (CSR) contains global operation parameters (bits from 0 to 2 and from 5 to 7) and records status informations produced by *MCC* functional blocks (bits from 8 to 15).

The contents of this register can be read by a RdRegister and overwritten by a WrRegister command.

Bit definitions for this register are in Table 2-2 and are described in the following subsections.

Table 2-2 Control (0-2,4-7) & Status (8-15) Register bit definition.

Bit	Name	To / From Blocks	Description
0-1	OM	EventBuilder	Output Mode select: 40/80 Mb/s, single/dual.
2	RECD	Receiver	Determines the meaning of WRECD register.
3-4	-	-	Not implemented.
5	OPAT	EventBuilder	Generate a 1010... pattern at the MCC-DTO 0/1 outputs.
6	PLBK	ModulePort, FePort	Event Playback
7	EVBDIS	EventBuilder	Disables the EventBuilder.
8	WLV1	CommandDecoder	Warning of 1-bit flip in LV1 command.
9	WFST	CommandDecoder	Warning bad Fast command.
10	WSLW	CommandDecoder	Warning bad Slow command.
11	BCIdChkFail	EventBuilder	Error on BCId check amongst FE's.
12	LV1ChkFail	EventBuilder	Error on LV1 check amongst FE's.
13	RegSEU	RegisterBank	A SEU has been detected in the RegisterBank block.
14	CmdSEU	CommandDecoder	A SEU has been detected in the CommandDecoder block.
15	EvbSEU	EventBuilder	A SEU has been detected in the EventBuilder block.

2.1.1.1 OM<1:0> : Output Mode select

These two bits (CSR <1:0>) select amongst 4 different output modes for the operations of the two MCC-DTO output links. (see Section 5-8, "Data transmission on the two MCC outputs to the ROD for different link operation modes.")

The four possible operation modes are only for the event read out and do not affect the configuration data transmission from the *MCC* to the *ROD*. Configuration data are always transmitted from the *MCC* at 40 Mb/s on both output pins.

The setting of the OM bits affects both the event data transmitted to the *ROD* and data transmitted while *MCC* is in OutputPattern mode (OPAT = 1).

- OM = 00: Same data on both links at 40 Mb/s with 40 Mb/s bandwidth;
- OM = 01: Data on both links at 40 Mb/s with 80 Mb/s bandwidth;
- OM = 10: Same data on both links at 80 Mb/s with 80 Mb/s bandwidth;
- OM = 11: Data on both links at 80 Mb/s with 160 Mb/s bandwidth;

At chip startup, or afeter a GlobalResetMCC command, OM is set to “00”, data on both links at 40 Mb/s.

2.1.1.2 RECD: Receiver disable

This bit (CSR <2>) determnies the meaning of the bits in the WRECD register and affects the FE-Flag word generation of the *MCC*.

- RECD = 0: All the flags disabled by WBITD in the corresponding Receiver (WRECD register) are disabled, i.e. do not trigger a FE-Flag generation in the EventBuilder state machine. This means that for all the Receiver’s where WRECD = 0 any flag detected will cause the *MCC* to produce a FE-Flag word;
- RECD = 1: The Flag generation is disabled for all bits in the Receivers for which WRECD is set, while for the remaining Receivers (WRECD = 0) only the bits which are “0” in WBITD contribute to the generation of the FE-Flag;

At chip startup, or afeter a GlobalResetMCC command, this bit is set to “0”.

Here is an example on how this bit affetcts FE-Flag generation.
Let us suppose WBITD = 0001_1111_0000_0000 and WRECD = 0001_0001_0001_0001 and FEEN = 1111_1111_1111_1111.

See Section 2.1.12and Section 2.1.13 for the meaning of single bits inside WBITD and WRECD.

If RECD = 0 in Receiver 0, 4, 8 and 12 (setting of WRECD) all MCC-Flags are disabled (setting of WBITD), while in the remaining Receiver blocks any type of FE-Flag word can be produced.

If RECD = 1 in Receiver 0, 4, 8 and 12 (setting of WRECD) all FE-Flags are disabled, while in the remaining Receiver blocks only flags coming from the *FE* chips (i.e. flagged in FE-EoE words) contribute to the FE-Flag word generation (setting of WBITD).

At chip startup, or afeter a GlobalResetMCC command, RECD is set to “0”.

2.1.1.3 OPAT: Set Output Pattern generation

When this bit (CSR <5>) is set the *MCC* generates a predefined pattern of 0’s and 1’s and it sends them to the *ROD* by the MCC-DTO 0, MCC-DTO 1 output pins.

The generated pattern depends on the setting of OM <1:0>. Table 2-3 shows the selected output pattern.

Table 2-3 Output pattern selection.

OM	Description
00	Output is a 40 Mbit/s “101010...” alternating pattern on both DTO pins.
01	Output is always “1” on both DTO pins.
10	Output is a 80 Mbit/s “101010...” alternating pattern on both DTO pins.
11	Output is always “1” on both DTO pins.

OM = 01 and OM = 11 are intended for a faster annealing of the Pin Diode.

During Output Pattern generation one can, in principle, still execute RdRegister and WrRegister commands. After a RdRegister command the *MCC* interrupts the selected Output Pattern, sends out the Header (11101) followed by the value stored in the selected register and then resumes the selected Output Pattern mode.


To turn off the output pattern generator, and resume normal execution mode, this bit has to be cleared.

At chip startup, or afeter a GlobalResetMCC command, OPAT is set to “0”.

2.1.1.4 PLBK: Event Playback


When this bit (CSR <6>) is set the *MCC* is in EventPlayback mode. This mode can be used in conjunction with the WrReceiver command (see Section 1.2.3.7, "WrReceiver: Write data into a Receiver") in order to exercise the *MCC* Receiver and EventBuilder blocks with simulated events. The correct sequence of operations to use this test mode is described in full detail in the Receiver and EventBuilder chapters.(XXX add references!!!)


During EventPlayback mode all LV1 commands decoded by the *MCC* act on the *MCC* as a real Trigger command with the only exception that this information is not transmitted to the *FE* chips.


During EventPlayback mode all data coming from enabled *FE* chips is cleared and therefore the only way to write data to the ReceiverFifo’s is to use the WrReceiver command.

The only way to exit from EventPlayback mode (beside a GlobalResetMCC command) is to clear the contents of the PLBK bit of the CSR register. This means that even after issuing a Slow command (which brings the *MCC* out of Run mode), and re enabling the run mode the *MCC* is still in EventPlayback mode.

At chip startup, or afeter a GlobalResetMCC command, PLBK is set to “0”.

2.1.1.5 EVBDIS: Event Builder disable


This bit (CSR <7>) allows to disable the EventBuilder state machine. This allows a user to stop the EventBuilder, read out the contents of all structures that contribute to event building, and eventually resume event building.


If one reads out the contents of any FIFO one has to ensure that all FIFO pointers are left in the same position as before stopping the EventBuilder in order to resume correctly the event building process. This can be done for example reading 128 times a Receiver FIFO, being the pointers cyclic.

Event building is resumed by writing a zero to CSR <7> and enabling RunMode again.

At chip startup, or afeter a GlobalResetMCC command, EVBDIS is set to “0”.

2.1.1.6 WLV1: Warning on LV1


This bit (CSR <8>) is set each time that a Trigger command with a single bit-flip has been detected by the CommandDecoder (sse Section 1.5.1, "Trigger: LV1"). This allows a quick check in order to see if there were bit flips in the run.

This bit is cleared by writing a zero to CSR <8>.

At chip startup, or afeter a GlobalResetMCC command, WLV1 is set to “0”.

2.1.1.7 WFST: Warning on Fast command


This bit (CSR <9>) is set each time that a bad Fast command has been decoded (see Section 1.5, "Command Robustness in Case of Transmission Errors"). This can only happen if there has been a single bit-flip in the Fast command body. This allows a quick check in order to see if there were bit flips in the run.

This bit is cleared by writing a zero to CSR <9>.

At chip startup, or afeter a GlobalResetMCC command, WFST is set to “0”.

2.1.1.8 WSLW: Warning on Slow command


This bit (CSR <10>) is set each time that a the Field 3 of a Slow command does not match to any known Slow command (see Section 1.5.6, "Slow: Any Command"). This can only happen if there has been a single bit-flip in the 4 most significant bits of the Slow command body. This allows a quick check in order to see if there were bit flips in the run.

This bit is cleared by writing a zero to CSR <10>.

At chip startup, or afeter a GlobalResetMCC command, WSLW is set to “0”.

2.1.1.9 BCIdChkFail: Error in the BCId check inside the Evb

This bit (CSR <11>) is set each time EventBuilder detects an error in the check of the BCId field, present in every EoE word, amongst different *FE* chips.

One can disable this check setting WBITD<11> to “1”, regardless of the setting of the WRECD register;

This bit is the logical OR of the errors for all the enabled *FE* chips and therefore it does not give information on which event caused the error but might be used to check if there has been errors of this type since last GlobalResetMCC command.

This bit is cleared by writing a zero to CSR <11> and is “0” at chip startup or after a GlobalResetMCC command.

2.1.1.10 LV1ChkFail: Error in the LV1 check inside the Evb

This bit (CSR <12>) is set each time EventBuilder detects an error in the check of the LV1 field, present in every EoE word, amongst different *FE* chips.

One can disable this check setting WBITD<12> to “1”, regardless of the setting of the WRECD register;

This bit is the logical OR of the errors for all the enabled *FE* chips and therefore it does not give information on which event caused the error but might be used to check if there has been errors of this type since last GlobalResetMCC command.

This bit is cleared by writing a zero to CSR <12> and is “0” at chip startup or after a GlobalResetMCC command.

2.1.1.11 RegSEU: SEU inside the RegisterBank

This bit (CSR <13>) is set each time inside the RegisterBank a SEU is detected.

This bit is the logical OR of all detected SEU’s and therefore it might be used to check if there has been a SEU in the RegisterBank since last GlobalResetMCC command.

See Section 2.3, "SEU Protection" for the definition of which registers have been protected against SEU.

This bit is cleared by writing a zero to CSR <13> and is “0” at chip startup or after a GlobalResetMCC command..

2.1.1.12 SEU inside the CommandDecoder

This bit (CSR <14>) is set each time inside the CommandDecoder a SEU is detected.

This bit is the logical OR of all detected SEU’s and therefore it might be used to check if there has been a SEU in the CommandDecoder since last GlobalResetMCC command.

See Section 1.6, "SEU protection" for the definition on how the CommandDecoder has been protected against the occurring of SEU events.

This bit is cleared by writing a zero to CSR <14> and is “0” at chip startup or after a GlobalResetMCC command..

2.1.1.13 SEU inside the EventBuilder

This bit (CSR <15>) is set each time inside the EventBuilder a SEU is detected.

This bit is the logical OR of all detected SEU’s and therefore it might be used to check if there has been a SEU in the EventBuilder since last GlobalResetMCC command.

See Section 4.4, "SEU protection" for the definition on how the EventBuilder has been protected against the occurring of SEU events.

This bit is cleared by writing a zero to CSR <15> and is “0” at chip startup or after a GlobalResetMCC command..

2.1.2 LV1: Level 1 Trigger

The LV1 register contains the parameters needed for the trigger operations in the *MCC*. This register uses 12 bits, which are logically subdivided into two fields.

The contents of this register can be read by a RdRegister command but a WrRegister command only affects the LV1 <11:8> field and leaves the LV1 <7:0> field as it was before.

2.1.2.1 LV1<7:0> : Level 1 ID Counter

These 8-bits are mapped from the Lv1Counter. The Lv1Counter keeps track of incoming triggers. Each LV1 trigger command stores the content of the Lv1Counter into the PendingLv1Fifo and then increments the counter. Each event, transmitted by the *MCC*, includes the content of this counter.

⚠ This field of the register is not affected by a WrRegister command. A RdRegister command returns the current value of the Lv1Counter. The ability to read the contents of the Lv1Counter are provided for debugging capability. The first RdRegister command, after reset, addressing the LV1 register, returns 0 in the LV1 <7:0> field.

2.1.2.2 LV1<11:8> Contiguous Level 1

This 4-bit field is the number of contiguous LV1 generated by the *MCC* for each received LV1 command. Contiguous LV1 are truncated if the number of LV1 transmitted to the *FE* chips becomes larger than 16 (size of the PendingLv1Fifo). A value of LV1 <11:8> = 0 generates a LV1 pulse lasting a single clock unit (no contiguous LV1). Allowed length of the LV1 pulse ranges from 1 to 16 (LV1 <11:8> + 1) CK units.

⚠ All events generated by contiguous LV1's, by a single LV1 command, have the same L1ID and consecutive BCID in the output data stream. If, while generating a contiguous LV1 signal, a second LV1 command is detected by the CommandDecoder the LV1ID is incremented.

LV1<11:8> is "0" at chip startup or after a GlobalResetMCC command.

2.1.3 FEEN: Front-End Enable

The 16-bits in this register enable (FEEN<i> = "1") or disable (FEEN<i> = "0") the corresponding Front-End. The i-th bit in the FEEN register, when set, has the following effects on the i-th Receiver:

- If the *MCC* is in RUN mode, the i-th Receiver is active and data from its ReceiverFifo are used to build the event;
- The WrFifo commands write into the i-th ReceiverFifo (RdFifo uses the address field in the command and does not look at the FEEN settings);
- The RdFrontEnd command routes the data coming from the i-th *FE* chip (trough the i-th MCC-DTI) to the MCC-DTO output.
  Only one *FE* should be enabled durig a RdFrontEnd command as otherwise one gets the logical OR of the signals coming from all enabled *FE* chips.;
- The WrReceiver command routes the bit stream in its data field to the i-th Receiver. All remaining Receivers are unaffected by this command. The addressed receiver sees the bit stream as simulated hits from *FE* chip and stores them in the ReceiverFifo. If an EoE word is detected by the Receiver the ScoreBoard is updated accordingly;
- When the *MCC* is in PlayBack mode (PLBK bit set in CSR register) the i-th Receiver takes part to the reconstruction of the event triggered by a LV1 command.

The contents of this register can be written by a WrRegister command and read back by a RdRegister command.

FEEN is "0" at chip startup or after a GlobalResetMCC command.

2.1.4 WFE: Warning from FE

When there is a warning recorded in the EoE word of the i-th *FE*, WFE <i> is set.

⚠ This happens only for those flag bits that are enabled and therefore depends on the settings of the RECD bit of the CSR register, the WBITD<7:0> and FEEN registers.

Bits in the WFE register can be cleared by writing zeroes. The bit pattern read out at the end of a data taking run gives an immediate information of the presence of warnings in any of the 16 *FE* chips for the whole run.

The contents of this register can be written by a WrRegister command and read back by a RdRegister command.

WFE is “0” at chip startup or after a GlobalResetMCC command.

2.1.5 WMCC: Warning from MCC Receivers

WMCC <i> is set each time an enabled Receiver sees an error in comparing the BCId field of a whole *FE* Event word (FE-Hit<24:21> and FE-EoE<24:21>) or if in the enabled ReceiverFifo there is a HitOverflow or an EoEOverflow.

⚠ This happens only for those flag bits that are enabled and therefore depends on the settings of the RECD bit of the CSR register, the WBITD<10> bit and the FEEN register.

Bits in the WMCC register can be cleared by writing zeroes. The bit pattern read out at the end of a data taking run gives an immediate information of the presence of errors in the whole run.

WMCC is “0” at chip startup or after a GlobalResetMCC command.

2.1.6 CNT: FE Configuration Counters

This register is mapped on two counters. These counters are used to program the number of bits in the address / command and in the data fields sent to the *FE* chips.

The number of CCK pulses generated during the transmission to the *FE* is CNT<2:0> x 8 + CNT<15:3>. The LD signal, needed to distinguish address / command and data fields, stays low for the first CNT<2:0> x 8 CCK clock pulses and is risen in correspondence of the CCK trailing edge staying high for the remaining CNT<15:3> CCK pulses (see Chapter 5, "Output Data Format"). The CCK (Configuration Clock) runs 8 times slower than the system clock (CK).

The contents of this register can be written by a WrRegister command and read back by a RdRegister command.

CNT is “0” at chip startup or after a GlobalResetMCC command.

2.1.6.1 CNT<2:0> :

CNT <2:0> is related to the number of control bits that will be transmitted to the *FE* chips. The actual number of control bits transmitted is CNT <2:0> x 8.

In case the number of control bits is not a multiple of 8, zeroes must be added in the MSB part of the control bits to create a n-bit word, where n is multiple of 8, and n/8 must be loaded in the CNT<2:0> to set up the transmission to the *FE* chips.

2.1.6.2 CNT<15:3> :

CNT <15:3> is the number of data bits that will be transmitted to the *FE* chip.

2.1.7 CNT: WrReceiver settings

CNT <12:0> is also used to configure the length of the data bits of the WrReceiver command. The actual number of data bits transmitted is CNT <12:0> x 8.

In case the number of data bits is not a multiple of 8, zeroes must be added in the MSB part of the data bits to create a n-bit word, where n is multiple of 8, and n/8 must be loaded in the CNT<12:0> to set up the correct lenght of data bits sent to all FEEN enabled Receiver's.

If one needs to send more than 65535 bits (2^{13} x 8 – 1) to the enabled Receiver blocks more consecutive WrReceiver commands need to be used.

The contents of this register can be written by a WrRegister command and read back by a RdRegister command.

CNT is "0" at chip startup or after a GlobalResetMCC command.

2.1.8 CNT: Delay line settings

The same register is also used to configure the width of the calibration pulse generated by the Delay Line of the *MCC*.

The contents of this register can be written by a WrRegister command and read back by a RdRegister command.

CNT is "0" at chip startup or after a GlobalResetMCC command.

2.1.8.1 CNT<15:0> :

These 16-bits select the width of the strobe pulse in CK units generated in response to a CAL command. Allowed pulse widths range from 1 to (2^{16} – 1) CK units.

The value of this register is used for calibration pourposes only during a CAL command (see Chapter 1.2.2.3, "CAL: Calibration").

2.1.9 CAL: Calibration

The CAL register contains the parameters (enabling and delay settings) needed to generate the calibration pulse for the FE chips, generated issuing a CAL command. The width of the calibration pulse dis set via the CNT register.

The generated calibration pulse is sent to the *FE* chips via the MCC-STRO output pin.

The contents of this register can be written by a WrRegister command and read back by a RdRegister command.

CAL is “0” at chip startup or after a GlobalResetMCC command.

This register uses 11 bits, which are logically subdivided into three distinct fields.

2.1.9.1 CAL<5:0> : Calibration delay

These 6-bits select the delay of the strobe pulse, in delay units, generated in response to a CAL command. The width of a delay unit depends on the settings of the calibration range.

2.1.9.2 Cal<9:6> : Calibration range

These 4-bits allow to select the coarse tuning of the minimum delay step of the strobe pulse, generated in response to a CAL command. This allows to safely fulfill the requirement of the delay pulse being wider than two CK cycles.

A typical value of the delay unit is 0.5 ns, allowing the total selectable delay to range between 0 and 63.5 ns.

2.1.9.3 Cal<10> : Calibration enabling

This bit allows to enable the *Delay Line* inside the *MCC*.

By default at startup of the *MCC* the *Delay Line* is disaled in order to reduce power consumption.

2.1.10 PEF: Pending Event FIFO

This register is mapped to the PendingLv1Fifo. Each write operation writes into the *FIFO* input, while read operation gets data from the *FIFO* output. The main idea of implementing this register was to be able to read the status of the PendingLv1Fifo that is used by the EventBuilder to store BCID, L1ID and skipped events (SKEV) information for event building (Chapter 4.1.4, "PendingLV1FIFO").

At chip startup, after a GlobalResetMCC or an ECR command, both pointers of the PendingLv1Fifo are set to “0”.

Each time a RdRegister or a WrRegister command is issued by the ROD the read and write pointer of the PendingLv1Fifo are increment by one.

It is therefore very important, before entering RunMode to reset the pointers with a GlobalReset-MCC or an ECR command or to make sure one has issued an exact multiple of 16 RdRegister and WrRegister commands in order to ensure that the PendingLv1Fifo pointers are cleared.

The PendingLv1Fifo is 16 word deep and is subdivided into three logically distinct fields. Both RdRegister and WrRegister commands act on all three fields.

2.1.10.1 PEF<7:0> : BCID

These 8-bits are mapped to the PendingLv1Fifo <7:0>, that are interpreted by the EventBuilder as the BCID field. This information is written into the BCID field of the *MCC* output data stream.

2.1.10.2 PEF<11:8> : L1ID

These 4-bits are mapped to the PendingLv1Fifo <11:8>, that are interpreted by the EventBuilder as the L1ID field. Only the 4 LSB of the Lv1Counter are copied into the *FIFO*.

This information is written into the 4 LSB of the L1ID field of the *MCC* output data stream.

2.1.10.3 PEF<15:12> :Skipped events

These 4-bits are mapped to the PendingLv1Fifo <15:12>, that are interpreted by the EventBuilder as the skipped events (SKEV) field. This field is added to the 4 most significant bits of the L1ID output data stream and if different from 0 is used by the *ROD* to know how many events are missing in the *MCC* output data due to a PendingLv1Fifo overflow.

2.1.11 SBSR: ScoreBoard Status Register

This register is mapped to the ScoreBoard inside the EventBuilder. The main idea of implementing this register was to be able to read the status of the ScoreBoard, used by the EventBuilder to know when there is at least one complete event. This register should be read out only for debugging pourposes.
Each bit of the SBSR tells if the corresponding Receiver has seen at least one complete Event. Event Building inside the EventBuilder block starts as soon as all 16 bits of the SBSR are "1" and therefore all disabled FE's contribute with a "1" to this register.
As a consequence of this behaviour at chip startup, or after a GlobalResetMCC command (FEEN = 0) the contents of the SBSR is "1111_1111_1111_1111".

As soon as one reads the contents of this register the EventBuilder exits from RunMode and therefore, before resuming Event Building one needs to issue a RunMode command.


this register can be read back by a RdRegister command but cannot be written by a WrRegister command.

2.1.12 WBITD: Warning Bit Disable Register

This 13 bit register is used to prevent certain flags from contributing to the FE-Flag word generation in the *MCC* data stream. There are 8 *FE* and 5 *MCC* specific flags thet can be individually disbled.

The *FE* flag bits are mapped to WBITD<7:0> and for their meaning please refer to the *FE* documentation.

The *MCC* flag bits are mapped to WBITD<12:8> and here is their definition:

WBITD<8>: HitOverflow disable. When set an eventual overflow of the number of Hit's in the ReceiverFifo counter will not be flagged. Regardless of the setting of this bit no pointer overflow will accour as all pointers are protected against overflows (the counter is simply not incremented);

WBITD<9>: EoEOverflow disable. When set an eventual overflow of the number of EoE's in the ReceiverFifo counter will not be flagged. Regardless of the setting of this bit no overflow will accour as all pointers are protected against overflows (the counter is simply not incremented);

WBITD<10>: Receiver BCIdChk disable. When set the consistency check of BCId<3:0> between different FE-EoE words inside the Receiver is not performed;

WBITD<11>: EventBuilder BCIdChk disable. When set the consistency check of BCId<7:0> between different FE-EoE words inside the EventBuilder is not performed;

WBITD<12>: LV1Chk disable. When set the consistency check of LV1Id<3:0> between different FE-EoE words inside the EventBuilder is not performed;

As WBITD<10:8> acts on all Receiver blocks in order to know if a certain FE-Flag is suppressed or not for a specific Receiver one has to check the contents of FEEN, WREGD and the RECD bit of the CSR.

For an example of the FE-Flag generation inhibition see Section 2.1.1.2, "RECD: Receiver disable".

The contents of this register can be written by a WrRegister command and read back by a RdRegister command.

WBITD is “0” at chip startup or after a GlobalResetMCC command.

2.1.13 WRECD: Warning Receiver Disable Register

This register is used to prevent certain flags from contributing to the FE-Flag word generation in the *MCC* data stream on a per Receiver basis.

If WRECD<i> is “1” the i^{th} Receiver block will not generate one or more FE-Flag's. The correct action of this register depends on the settings of FEEN, WBITD and the RECD bit of the CSR register. For an example of the FE-Flag generation inhibition see Section 2.1.1.2, "RECD: Receiver disable".

The contents of this register can be written by a WrRegister command and read back by a RdRegister command.

WRECD is “0” at chip startup or after a GlobalResetMCC command.

2.2 Reset Actions

This section describes the action of all the reset commands on the single registers.

2.2.1 Response to a Pin Reset

The pin reset (MCC-RSIb) has the same functionality as the GlobalResetMCC command. It is a synchronous reset and therefore one has to ensure that the clock signal is applied in order to be able to reset the chip in this way.


One single clock cycle should be enough to fully reset the *MCC*, but as there has been no particular timing constraint on this signal during logic synthesis it is preferred to use at least a 2 clock wide signal to ensure that the *MCC* is properly initialized.

The pin reset signal is present only for testing pouposes and will not be used during the experiment. The *MCC* has been designed in such a way that the CommandDecoder will alway start-up in a well known state and therefore it is always possible to correctly initialise the whole chip using a GlobalResetMCC command.

2.2.2 Response to a GlobalResetMCC command

GlobalResetMCC acts on all the *MCC* internal registers.

The contents of the PendingEventFifo are cleared by a GlobalResetMCC command.

Table 2-4 describes the action of the GlobalResetMCC command on all the *MCC* registers.

Table 2-4 Action of the GlobalResetMCC command on the registers.

Register	Before reset	After reset	Note
CSR	ssss,ssss,ccc-,-ccc	---0,0000,000-,-000	Configuration mode, 40 Mb/s link.
LV1	----,CCCC,LLLL,LLLL	----,0000,0000,0000	No contiguous LV1, L1ID cleared.
FEEN	dddd,dddd,dddd,dddd	0000,0000,0000,0000	All FE are disabled.
WFE	dddd,dddd,dddd,dddd	0000,0000,0000,0000	Cleared.
WMCC	dddd,dddd,dddd,dddd	0000,0000,0000,0000	Cleared.
CNT	dddd,dddd,dddd,dccc	0000,0000,0000,0000	Cleared.
CAL	----,-pRR,RRDD,DDDD	----,-000,0000,0000	No delay, 1 CK pulse width.
PEF	SSSS,LLLL,BBBB,BBBB	xxxx,xxxx,xxxx,xxxx	First PendingLv1Fifo location.
SBSR	ssss,ssss,ssss,ssss	0000,0000,0000,0000	Cleared.
WBITD	---w,wwww,wwww,wwww	---0,0000,0000,0000	Cleared.
WRECD	wwww,wwww,wwww,wwww	0000,0000,0000,0000	Cleared.

Note
B,c,d,D,l,n,p,R,s,S,w: used data bit
x: unknown bit
-: not existing bit. It is always read back a ‘0’

2.2.3 Response to a BCR command

The BCR command has no effect on any register.

2.2.4 Response to an ECR command

The ECR command has no direct effect on the *MCC* registers but has an indirect effect on part of the LV1 register: as LV1 <7:0> is mapped to the Lv1Counter which is reset, it returns 0 if read out immediately after an ECR command.

ECR also resets the pointers of ReceiverFifo's and PendingLv1Fifo. Since this last one is read out through the PEF register, the word read out after ECR is the content of the first *FIFO* location.

The action of the ECR command on all *MCC* registers is shown in Table 2-5.

Table 2-5 Action of the ECR command on registers.

Register	Before reset	After reset	Note
CSR	ssss,ssss,-ccc,cccc	---s,ssss,ccc-,-ccc	Unchanged.
LV1	----,CCCC,LLLL,LLLL	----,CCCC,0000,0000	Contig. LV1 unchanged, L1ID cleared.
FEEN	dddd,dddd,dddd,dddd	dddd,dddd,dddd,dddd	Unchanged.
WFE	dddd,dddd,dddd,dddd	dddd,dddd,dddd,dddd	Unchanged.
WMCC	dddd,dddd,dddd,dddd	dddd,dddd,dddd,dddd	Unchanged.
CNT	dddd,dddd,dddd,dccc	cccd,dddd,dddd,dddd	Unchanged.
CAL	----,-pRR,RRDD,DDDD	----,-pRR,RRDD,DDDD	Unchanged.
PEF	SSSS,LLLL,BBBB,BBBB	xxxx,xxxx,xxxx,xxxx	First PendingLv1Fifo location.
SBSR	ssss,ssss,ssss,ssss	ssss,ssss,ssss,ssss	Unchanged.
WBITD	---w,wwww,wwww,wwww	---w,wwww,wwww,wwww	Unchanged.
WRECD	wwww,wwww,wwww,wwww	wwww,wwww,wwww,wwww	Unchanged.

Note
B,c,d,D,l,n,p,R,s,S,w: used data bit
x: unknown bit
-: not existing bit. It is always read back a '0'

2.3 SEU Protection

In order to protect critical configuration information stored inside the *MCC* a certain number of registers have been protected against SEU. Not all registers were protected in order to save die area and to minimize power consumption.

As a protection against SEU all configuration registers have been triplicated and a majority voting circuit at their output selects the correct value to be used inside the MCC.
In addition to the majority voting circuit there is an XOR tree of all triplicated registers to calculate, once every clock cycle, if there has been a detection of a SEU event inside the RegisterBank.
If an error is found the information is stored in CSR<13>.

A majority voting logic is a good method for protecting the registers as at each clock cycle the contents of the registers (if not changed via a WrRegister command) is updated with the correct value. Therefore the probability of having a wrong answer from one of this protected registers is completely negligible.

Triplicated registers are: CSR<7:5>, CSR<2:0>, FEEN, WRECD and WBITD.