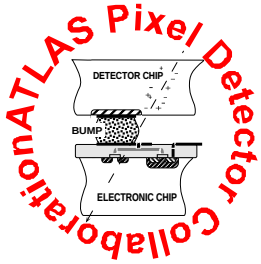

Atlas Pixel Demonstrator

Pixel Electronics Specifications

System Architecture

Issue:
Revision: 2.0
Reference: ATLAS ID-xx
Created: 27 December 1996
Last modified: 12 October 1999 15:25
Edited By: G. Darbo, K. Einsweiler, P. Fischer
ATLAS / Pixel Collaboration

Abstract

This document describes the system architecture for the ATLAS pixel detector module demonstrator. The module is build around two kind of electronic chips: the Front-end (FE) and the Module Controller Chip (MCC). We give here only the specifications which involve the interfacing of the two chips, while we relate to specific documents for the complete description of the two chips.

Table Of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose of the Document

The purpose of this document is to define the specifications for the ATLAS Pixel Detector Module which will be built for the TDR demonstrator. A simplified block diagram of the module with Data, Control and Timing interconnections is shown in Figure 1-1.

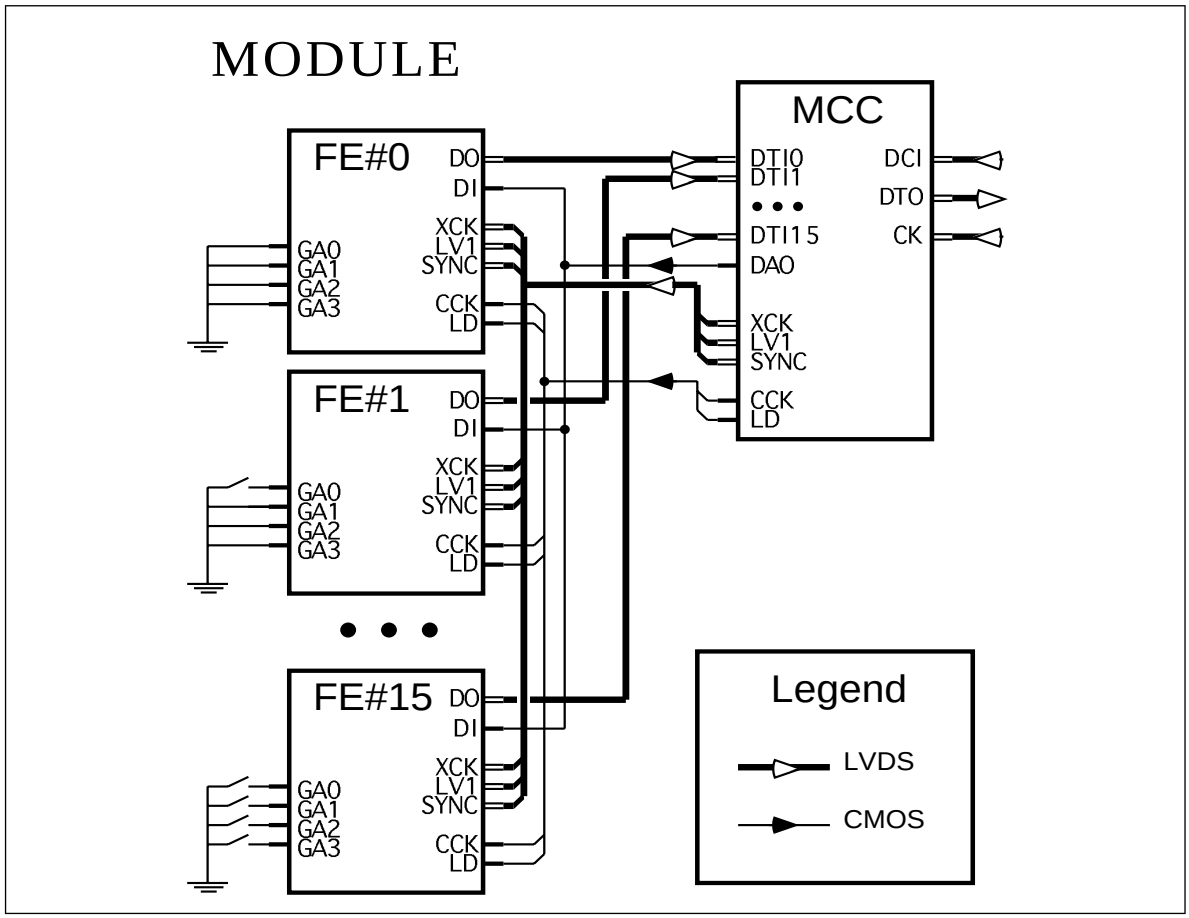

Figure 1-1 Block Diagram of Module Demonstrator

The module is built using two types of chips:

- ***Front-End Chip (FE):***

  which includes an array of 400 x 50 µm pixels arranged in 18 columns with 160 rows each, the End-of-Column (EoC) logic which does zero suppression and performs the L1 trigger coincidence, a serial transmitter and a minimum of control logic to initialize the mask and test flip-flops in the pixel cells. DAC's for analog parameter setting are in the chip as well.

- ***Module Controller Chip (MCC):***

  which performs the two main functions of event building and error handling, and of timing and control for the FE chips in the module. This chip can also work in *"transparent mode"* to allow direct access to the FE chips, to support more flexible operation during de-

bugging. Other diagnostic functionality is embedded in the chip both for on-chip and system tests.

1.2 Document Outline

Chapter 1 contains a summary of the document purpose, history and a list of references.

Chapter 2 explains the functionality of the two chips --front-end (FE) and module controller (MCC). This chapter is not meant to provide the reference specifications, which are to be found in related documents [1],[2].

Chapter 3 contains the pin list and power specifications for the FE and MCC chips. The I/O protocols and signal timing relationships also defined here.

1.3 Definitions

1.4 Document History

Draft 0.1: 31-January-1997

First Draft release based on the outcome from the Pixel Electronic Workshop of 17-18 Dec 97

Draft 0.2: 17-February-1997

Small corrections to previous draft.

Added the following sections:

Section 2.2.6, "Chip Configure: Frame Decoding"

Section 2.3.1, "Internal Registers"

Section 2.3.2, "Command Set"

Section 2.3.8, "FE Test Features: Transparent Mode"

Section 3.3.3, "Event Format at MCC Output"

Section 3.3.4, "Configuration Data Format at MCC Output"

Section 3.3.5, "Control/Data Format and Protocol at MCC Input"

Draft 1.0 2- March-1997

First Draft revision circulated to the Pixel Collaboration. Modifications to several paragraphs as a result of the discussions and decisions in the “Pixel Electronics Workshop” of 24-25 February 1997.

Added the following sections:

Section 3.4.1.1, "Signals in Data Taking Operation."

Section 3.4.1.2, "Signals in Initialization and Control Operation"

Section 3.4.2.1, "MCC to FE Timing: Data Taking Operation."

Section 3.4.2.2, "MCC to FE Timing: Control Signals"

Section 3.4.2.3, "MCC to LCC Timing: Control and Command-Data Signals"

Draft 1.1 6- June -1997

Second Draft revision circulated to the Pixel Collaboration. Modifications to several paragraphs as a result of the discussions and decisions in the “Pixel Electronics Workshop” of 23 May 1997.

Added the following sections:

Section 2.2.4, "Chip Configure: Registers"

Section 2.3.3, "System Initialization"

Section 2.3.6, "Trigger, Timing & Control"

Section 2.3.7, "Chip Test Features"

Draft 1.2 21-July -1997

Third Draft revision. Modifications to several paragraphs as a result of clarifications during the design of MCC and FE chips.

Added the following sections:

Section 2.4.1, "Signal Routing and Line Terminations."

Section 3.2, "Input/Output Pin Electrical Specifications"

Section 3.2.1, "CMOS Type Pins"

Section 3.2.2, "LVDS Type Pins"

Draft 1.3 24-July -1998

In Table 3-1 2 columns have beenremoved as now all the pin names are the same for FE -A/B chips. VCALa is now pin 23 (before it was pin 24) and-some pin names have been changed.

Added one column in Table 3-4 and inTable 3-4.

Removed DriveHiD and changed DriveHi (from 1.5 mA to 3.0 mA with a swing of 300 mV instead of 150 mV) in Section 3.2.2, "LVDS Type Pins"

Specified better how long LV1 pulses are generated setting LV1Bd<7:4> in Section 2.3.1, "Internal Registers".

Entered MCC pin names in Table 3-6.

Issue 2.0 Rev.0 Added definition for bit number 5 of MCC GCR in Table 2.3.1.
Updated Table 3-6 with MCC pad positions and pin numbers.
Corrected Figures 2-17 and Figures 3-3.

1.5 Related Documents

1. *Pixel Front-End Chip Specification,* K. Einsweiler and P. Fischer
2. *Module Controller Chip Specification,* G. Darbo and G. Meddeler

2 System Architecture Description

Only a simplified description of the FE chip and of the MCC is given in this document. For a detailed description see the documents [1][2] listed in Section 1.5.

2.1 System Architecture and Block Interconnects

A simplified block diagram of the module demonstrator is shown in Figure 1-1. Figure 2-1 shows a schematic from Cadence™ with the 16 FE chips, the MCC and their interconnections.

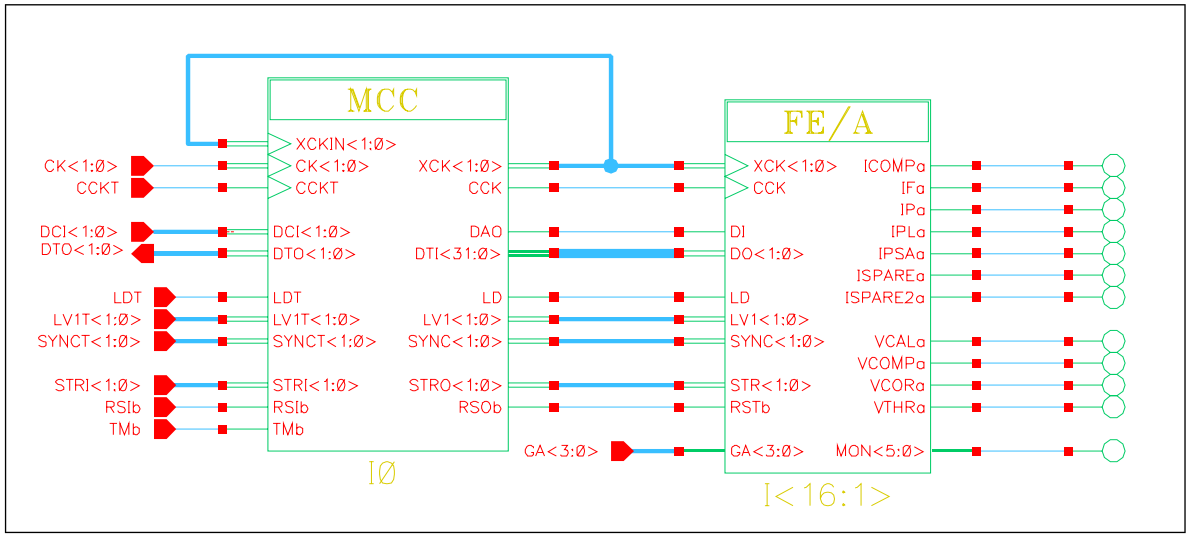

Figure 2-1 Schematic of the Pixel Module. The FE chip in the figure is the FE/A chip; FE/B chip has the same pinout being different only the test pins on the right end side of the chip symbol.

The interconnect topology between the MCC and the 16 FE chips in a module is a star topology which uses unidirectional serial links. There are two protocols defined for data transfer between the FE chips and the MCC:

- Event readout protocol.

  Data are generated by the FE's in response to MCC-*LV1* signals and transmitted serially from the FE-*DO* pin. Data at the output of the FE are in phase with the positive edge of the 40 MHz clock (MCC-*XCK*). The *MCC-XCK* is a fanout of the MCC-*CK* input clock which is delayed internally by the MCC to take into account different cable lengths between detector modules. This phase adjust is necessary because the MCC-*XCK* signal is used to correlate pixel hits from particles which are produced in time with a particular bunch crossing. The MCC for the Pixel TDR Demonstrator will not implement the logic to adjust the MCC-*XCK* / MCC-*CK* phase.

- Configuration data upload / download protocol.

  To upload data to the FE's, the MCC provides data (*MCC-DAO*) together with a clock to latch them (*MCM-CCK*) on the positive edge. The *MCC-LD* signal is used to distinguish, in the bit stream from the *MCC-DAO* pin, the address plus control words from the subsequent data words. The *CCK* is generated by the MCC only when serial data present at the *DAO* output must be latched in the FE. This clock has a lower frequency (5 MHz) than the

XCK to reduce the timing requirements on the internal registers which can be loaded or read back using this protocol.

FE chips are selected for parameter read / write by serial geographical addressing. Before sending configuration data to a specific FE chip, an address / control bit stream in the downloaded data packet selects both the FE chip and the secondary address or command to execute. Since some of the internal registers in the FE chip are organized as long chains of shift registers their readout will be destructive. Furthermore, no partial data loading is possible within each memory structure.

The protocol which the MCC uses to communicate to the outside world uses two data lines: the *DTO* for the data output and the *DCI* for both data and command input. The two lines transmit or receive information using the *XCK* timing signal as a reference clock.

2.2 Front-End (FE) Chip.

For the TDR demonstrator program, two different versions of the FE chip are being designed. One, referred to in the following text as *FE-A*, will be implemented in the double metal 0.8µ AMS BiCMOS process, and the second, referred to as *FE-B*, will be implemented in the triple metal 0.8µ HP CMOS process. These two chips have been defined to be “functionally pin compatible”, by which we mean that they can be read out by a common MCC chip and a common data acquisition system. Internally, there will be several significant differences, which are summarized briefly here, and described in more detail in the Reference Documents. The information provided should describe the aspects of the FE chips which are visible to an external user, and which must be understood to correctly operate the chips.

2.2.1 Pixel Cell Architecture

Both FE chips are designed to process the negative polarity input signals expected from n^+ implants on silicon sensors (they collect electrons not holes). The *FE-A* chip uses a charge-sensitive preamplifier as a single high-gain stage to drive an AC-coupled discriminator fabricated using bipolar input transistors to minimize DC offsets which cause threshold dispersion. The *FE-B* chip uses a similar preamplifier configuration to drive a dual-threshold discriminator, designed to minimize the effects of capacitive cross-coupling and out-of-time hits. Both preamplifiers provide a quasi-linear TOT (time-over-threshold) output to give a modest resolution (4-5 bits) charge measurement with a baseline return time of roughly 500 nsec. The TOT behavior saturates to prevent very large pulses from generating excessive deadtime.

The digital back-end of the pixels is adapted to the particular read-out architecture in each chip. In the case of *FE-A*, the readout occurs using a 40 MHz dynamic shift register to transfer the address of the hit pixel to the End-of-Column buffering. This shift register acts as a clocked delay line, and provides the timing information required to make a trigger coincidence after the L1 latency. The *FE-B* architecture brings a 7-bit Gray-coded timestamp directly to each pixel to associate a hit with a unique beam crossing. This data is then transferred down to the End-of-Column buffering along with the hit pixel address. The timestamp is used to make a coincidence with the corresponding L1 trigger. Both architectures transfer the data from a hit pixel as quickly as possible to minimize inefficiencies (the pixels do not support multiple hits). Large End-of-Column buffers then provide the storage of the hits during the L1 latency.

In order to minimize the space taken by this complex digital circuitry, columns are arranged in pairs, which are “mirrored” about the vertical axis in order to place the digital back-ends of two

adjacent columns together. The readout circuitry is then shared between a pair of columns, and feeds into a common End-of-Column buffer pool.

2.2.2 Event read out: End-of-Column Logic

The End-of-Column logic receives input from a pair of pixel columns as described above. Storage for 20 hit pixels is provided. This storage must keep track of the pixel hits until a L1 trigger coincidence is performed, and for those hits associated with a triggered beam crossing, it keeps track of them until they are transferred off the FE chip into the MCC. These basic operations, namely hit storage, L1 trigger coincidence, and event readout, must be simultaneously performed by the End-of-Column logic in order to prevent generating significant inefficiencies. Each FE chip uses a 4-bit Trigger Number to uniquely label events which are awaiting readout into the MCC. The MCC then will suppress additional L1 triggers to all of the FE chips on a given module if their transmission to the FE chips would cause more than the maximum of 15 events to be simultaneously stored in any one FE chip.

Given the “as-quickly-as-possible” approach to getting data out of hit pixels and into the End-of-Column buffers, the data itself becomes somewhat scrambled, and will no longer arrive organized sequentially into events. Hence the buffer will not behave as a FIFO, and data from a given event will not be stored contiguously. It is therefore necessary for the buffer pool to be scanned for the first free location prior to writing new data. It is also necessary for the buffer pool to be scanned to find all of the hits corresponding to a given event in order to provide the MCC with data ordered by event.

2.2.3 Chip Configure: Geographical Address

Each FE chip has a unique hard-coded `geographical address` which is set by binary programming of the four *GA* pins done by connecting either to ground (logical ‘0’) or to VDD (logical ‘1’). Each of the 16 FE chips in a module must have a different geographical address.

Information is sent from the MCC to all the FE chips simultaneously using the data input *FE-DI* and the clock signal *FE-CCK*. Address information must precede each data stream. Each FE chip looks at the address it has received and compares it with its own `geographical address` to decide whether the chip is selected. The encoding scheme defined for FE addressing is as follows:

Encoded address Use 5 bits. Four bits (A_0, ..., A_3) from 0 to F (hex) select one out of 16 FE chips, while the fifth bit (B) is used for broadcasting the same command to all chips on the module. Only one FE can be selected at a given time, or all are selected together in the case of a broadcast.

2.2.4 Chip Configure: Registers

Several internal control registers have been implemented in the FE chips, and their present formats are defined in this section. There are two basic types of registers. Those bits which are in the pixel are implemented in the simplest possible fashion as a single long register which has destructive readout. Those bits in the other control registers are implemented using a serial input register and a separate parallel D-FF latch. It is assumed that each such control register contains its own serial input register and parallel-load latch, to allow convenient input and output. In this case, they can be separately written and readback in a non-destructive way, and their latched version does not take on transient values when the serial input register is modified. Fi-

nally, there are certain functions which are controlled by *Enable* commands below, where the bits from the command register directly control the corresponding logic.

Pixel Shift Register Long shift register which contains 1 bit per pixel (*Select*) in the shift register, and allows access to a second bit in a latch (*Mask*). This register is implemented as a single long register for the entire array. The *Select* bit is used to specify which pixels participate in many of the pixel-level operations defined below. The *Mask* bit sits between the pixel front-end and the digital readout, and is capable of blocking access to the readout for any set of pixels.

DAC Control Register Long serial register for loading the DACs used to control the adjustable currents for the front-ends (DAC Serial Input Register), and the corresponding parallel-load DAC Registers. There are 8 bits reserved per DAC, and a maximum of 8 DACs defined. All DAC Serial Input Registers are connected in series to avoid sub-address decoding. The contents of the DAC Serial Input Register are then strobed into the individual DAC Registers to avoid producing transient DAC values as the serial input register is updated. The order of the DACs is defined so that the first bit on *FE-DI* will become the MSB of the last DAC after the correct number of *FE-CCK* are applied. This ordering allows chips which implement fewer DACs to simply let the extra bits fall off the end of the DAC Control Register.

Global Control Register Single register for controlling various global parameters, implemented as a Global Serial Input Register and an associated Global Register, just as for the DAC control registers. This register presently includes the L1 latency (either the preset value for the count-down scalers used in making the L1 trigger coincidence in *FE-A* or a preset for the timestamp generator to define the L1 latency in *FE-B*). It also includes two bits which are used to control the serial output multiplexing (whether the *FE-DO* output is connected to the output of the event data serializer, or the output of one of the configuration registers described here). The most significant 8 bits are reserved for the L1 Latency value, and the least significant 2 bits control the output multiplexing. Bit 0 enables the multiplexing of the event data off-chip, bit 1 enables the multiplexing of the internal registers off-chip, and any other values for this field are presently un-defined.

Diagnostic Registers These are registers which would be placed in critical data paths, such as the entrance and exit from the End-of-Column buffers, and strobed using the *STR* signal. They allow internal latching of values of critical bus states, allowing subsequent readout for diagnostic studies. They are not yet well-defined for either FE chip.

All registers follow the convention that the first bit in the serial stream enters as the LSB of the first register in the chain, and finally becomes the MSB of the last register (see Figure 2-5 for reference). The ordering of the *Select* bits in the Pixel Shift Register is shown schematically in Figure 2-2 including the reference numbering of the individual pixels in the 2D array.

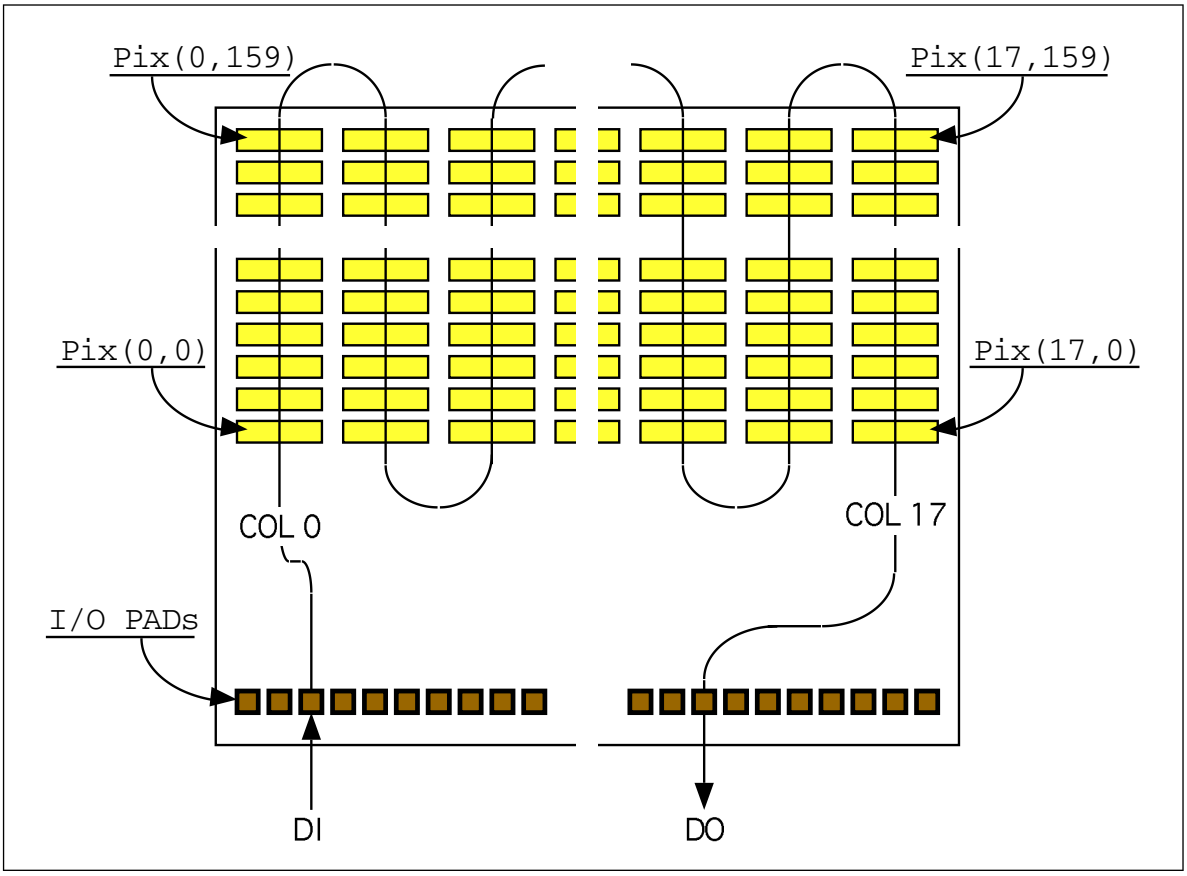

Figure 2-2 The ordering of the Select bits in the Pixel Shift Register. The reader is assumed to be looking down on the electronics die from the electronics side.

2.2.5 Chip Configure: Commands

Each time a FE chip is addressed using the serial input, a `command` is sent together with the `address`. The `command` has fixed length and immediately precedes the `address`. In this way, it is possible to increase the length of the `command` string at a later date, and the extra bits will be clocked off the end of the decoding register in earlier generations of chips, allowing backwards compatibility.

The commands can immediately perform an action (so that they do not need `data` accompanying them, such as a SoftReset) or they may determine the routing and meaning of the incoming `data`. In particular, the serial data stream can be directed to any of the various shift registers defined in the previous section. Since the number of commands that we have is not exceedingly high, we have chosen to reserve one bit per command, in order to have maximum flexibility in combining command operations. If later on, we need more command bits, this could still be done with a ‘secondary command register’ or by increasing the length of the `command` word.

The present implementation provides a total of 24 command bits (C_0, ..., C_{23}), of which the first 16 (C_0, ..., C_{15}) are reserved for “common” commands which have the same meaning for both the *FE-A* and *FE-B* chips, and the final 8 (C_{16}, ..., C_{23}) have specific meanings for each of the two front-end chips. The currently defined command set includes the following common commands (first 16 bits of the `command` input):

Null Command corresponding to no bits set in the command string. This would be the state of the command register after a FE chip reset *FE-RST*.

SoftReset Triggers the internal “soft reset” sequence, which will empty data buffers and reset counters.

ClockSelect Connects the *FE-DI* input to the Pixel Shift Register, and connects the output from the Pixel Shift Register to the *FE-DO* output. It will also enable the *FE-CCK* clock to enter the Pixel Shift Register clock generator, and hence to shift data into the Pixel Shift Register to set the *Select* bits.

ClockDAC Enable *FE-CCK* clock to shift data into the DAC Serial Input Register to load DAC’s. Also connects the *FE-DI* input to the input of the DAC Serial Input Register, and the output of the DAC Serial Input Register to the *FE-DO* output.

ClockGlobal Enable *FE-CCK* clock to shift data into the Global Serial Input Register to load the Global Control Register, including the programmable latency register and the output multiplexor control bits. Also connects the *FE-DI* input to the input of the Global Serial Input Register, and the output of the Global Serial Input Register to the *FE-DO* output.

WriteMask Strobe which latches the present *Select* bits into the *Mask* registers in each pixel.

WriteDAC Strobe which transfers data from the DAC Serial Input Register loaded previously to the individual DAC Registers which determine the DAC values.

WriteGlobal Strobe which transfers data from the Global Serial Input Register loaded previously to the Global Register.

ReadDAC Transfer individual DAC Register data values to the corresponding locations in the DAC Serial Input Register for read back.

ReadGlobal Transfer Global Register data values to the corresponding locations in the Global Serial Input Register for read back.

EnableCalibration Enable analog charge injection using *VCAL* and *STR* input pins, and based on *Select* bit of Pixel Shift Register

EnableDigitalHit Enable digital injection using *STR* input pin, and based on *Select* bit of Pixel Shift Register

EnableHitBus Enable transmission of *HitBus* signals from array (also requires *Select* signal to be present in given pixel)

Front-end chip specific commands (last 8 bits of the `command` input):

EnableAnalog Enable access to vernier capacitor via *VCOR* pin for threshold adjustments on *FE-A* chip.

EnableRef Enable control of reference input to the differential amplifier in each pixel. Reference is either derived from an individual tuning capacitor in each pixel, or from a replica preamplifier in the column bias circuitry, whose baseline is in turn controlled by a global reference current. Used only for *FE-B* chip.

EnableTuning Enable operation of automatic threshold tuning circuitry upon receipt of a *STR* input. The *STR* input first injects charge into the Selected pixels, and then if EnableTuning is set, after an adjustable delay, it checks to see whether the pixel high threshold discriminator fired or not. This in turn controls the injection of a correction current into the individual tuning capacitor for that pixel, and thereby tunes the threshold. Used only on *FE-B* chip.

These commands represent a first attempt to define the necessary internal registers and their access. Their implementation is summarized in Table 2-1.

Table 2-1 Summary of Serial Commands for FE-A and FE-B chips. The value corresponds to the value for the 16-bit “common” command field or the 8-bit “private” command field.

Command	Value	Chip	Description
Null	0		Do Nothing
SoftReset	1		Soft reset of chip
ClockSelect	2		Enable CCK to Pixel Shift Register
ClockDAC	4		Enable CCK to DAC Serial Register
ClockGlobal	8		Enable CCK to Global Serial Register
WriteMask	16		Strobe Select bits into Mask Register
WriteDAC	32		Strobe DAC Serial Register into DAC
WriteGlobal	64		Strobe Global Serial Register into Global
ReadDAC	128		Strobe DAC into DAC Serial Register
ReadGlobal	256		Strobe Global into Global Serial Register
EnableCalibration	512		Enable charge injection to Selected pixels
EnableDigitalHit	1024		Enable digital hits into Selected pixels
EnableHitBus	2048		Enable HitBus from Selected pixels
EnableAnalog	1	FE-A	Enable VCor to tuning capacitor
EnableRef	1	FE-B	Enable global reference input
EnableTuning	2	FE-B	Enable threshold tuning circuitry

The behavior of the serial decoding logic when the *RST* global reset is applied should include the following features. First, the command register will be reset too *Null*. Second, the values of various internal registers should be set to their default power-up settings.

2.2.6 Chip Configure: Frame Decoding

As long as *FE-LD* is low, the chips expect `address` and `command` data on the *FE-DI* input. The input data is latched on the rising edge of *FE-CCK*. In principle, more than 5+16+8 bits could be sent, but only the last 29 bits would be used, the others just drop off the shift register. Together with the `address`, a `command` is always sent. The `command` tells the chip what to do with the incoming `data` or which operations to execute. The `address` plus `command` stream has fixed length. As long as *FE-LD* is low, the FE-*DI* stream is recognized as `address` plus `command`.

`Data` are sent after `address` and `command`, and the `command` specifies what to do with them. Some commands need `data`, others do not need `data`. The size of the data block can therefore vary between 0 and e.g. 160x18 (the number of pixels). Data can be loaded directly into internal FE structures, like the pixel *Select* bits or, as in the case of the DAC’s, into “shadow registers”,

which need a second command to be copied into the actual DAC registers. This is to avoid transient values of analog parameters while loading operations are being performed.

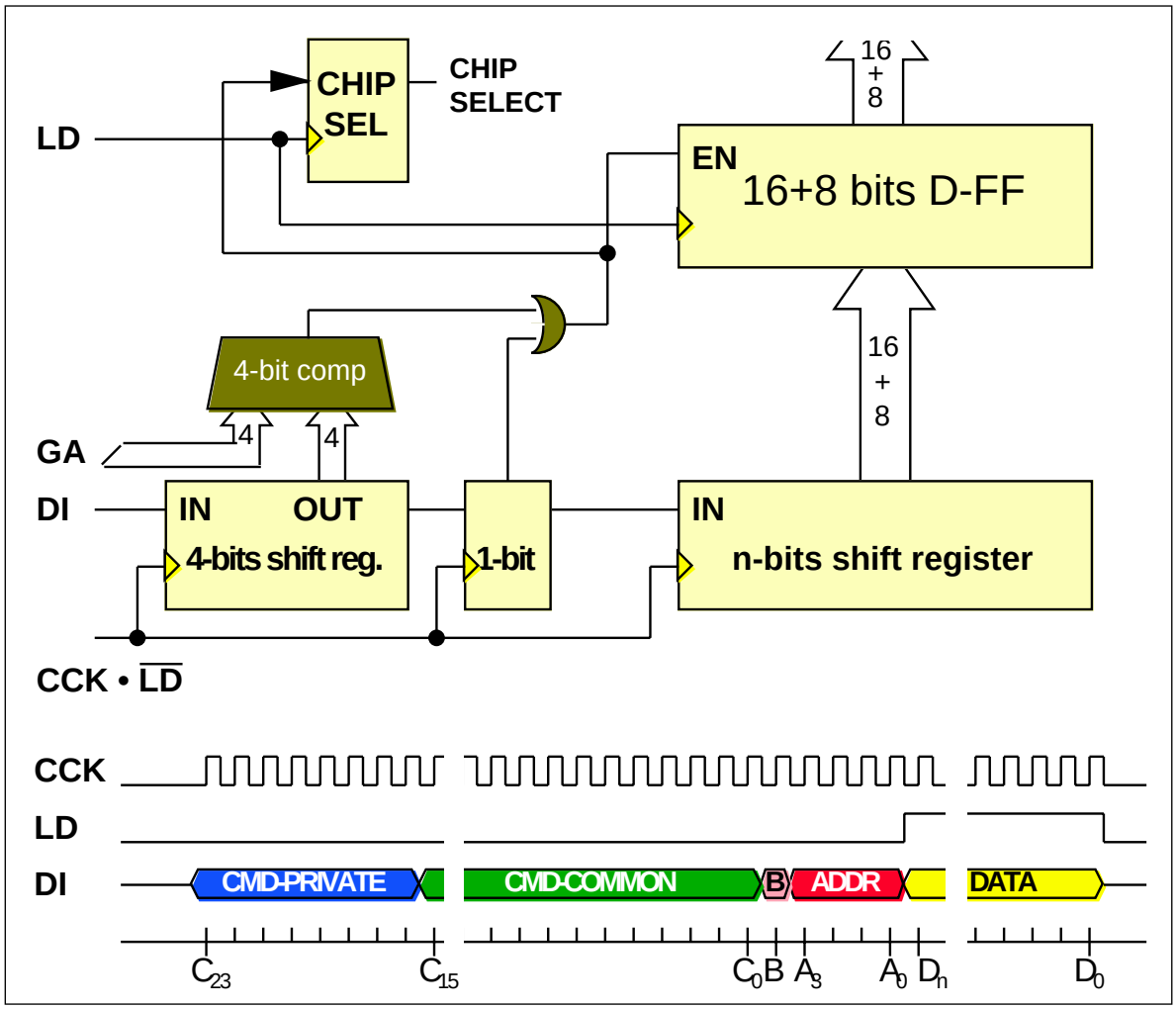

Figure 2-3 Simplified scheme for decoding commands and addresses in a FE chip.

With the rising edge of *FE-LD*, all chips compare the `address` pattern with their *GA* and the result is latched into a D-FF. The `command` data is latched into a 24-bit D-FF with the rising edge of *FE-LD*, only if the `address` bit for this chip is set. The block diagram which implements the `address` and `command` decoding is sketched in Figure 2-3.

The `address` is supplied after the `command`. The advantage is that if in later versions more `command` bits are needed, they can be added and the protocol is unchanged (the extra bits are just ignored by 'old' chips).

The *FE-LD* signal is used to distinguish two information blocks: `address+control` when *FE-LD* is low and `data` when it is high. When *FE-LD* is high, only the selected chips expect `data`. If the command does not use `data`, *FE-LD* can be pulled back low after one *CCK* cycle. The bits in the `command` determine what the chips do with the data. Since we mainly want to load shift registers, which do not have separate enable pins, we have to switch on and off the shift register clocks. If we do it like this, the clock of these registers is simply *CCK• ClockDAC*, as an example for the DAC shift register.

The output *FE-DO* of the chip will come from a MUX which can switch several signal sources to *FE-DO*: the event data stream or the output of the selected shift registers. The selection of the signal source can be modified by writing to the Global Control Register.

The data stream `command` / `address` (*FE-LD* rises) `data` (*FE-LD* falls) is supplied by the MCC. In a 'transparent' mode, the three signals needed can simply be passed through the MCC. In normal mode, the MCC generates the correct timing itself. The input data to the MCC via the serial link would then be something like `length command address data`. The `length` information is needed, as the MCC does not know how much data will be supplied. There is no need for length information for `command` plus `address` since this length is fixed.

The FE's can send back the information previously loaded in the shift registers through *FE-DO*. The MCC contains a register to select which of the *MCC-DTI* to connect to the *MCC-DTO* to transmit upstream FE configuration data.

2.2.7 Chip Configure: Example

To illustrate the full protocol for communication with the FE chips through the MCC chip, we provide a list of the exact commands that must be executed to load the *Mask* bits into FE#2 on a module, and check the operation by reading back the input stream. The format of the commands at the MCC requires the user to specify *field3* (MCC command), *field4* (first data value), and *field5* (second data value), as detailed in Table 3-13.

1. WrRegister FEEN 2 : select which FE chip will have its *FE-DO* connected to *MCC-DTO*.
2. WrRegister CCNT 29 : write command / address length to MCC register.
3. WrRegister DCNT 10 : write data transfer length for write to FE.
4. WrFrontEnd ClockGlobal 2 : write bit to FE Global Configuration Reg to multiplex parameter data from *FE-DO* instead of event data.
5. WrRegister DCNT 0 : set data transfer length for write to FE.
6. WrFrontEnd WriteGlobal 0 : strobe data from previous write into FE Global Configuration Reg.
7. WrRegister DCNT 2880 : set data transfer length for write to FE.
8. WrFrontEnd ClockSelect MaskData : transfer *Mask* data into Pixel Shift Register, with pattern of bits ordered as shown in Figure 2-2.
9. WrRegister DCNT 0 : set data transfer length for write to FE.
10. WrFrontEnd WriteMask 0 : strobe data from Pixel Shift Register into individual *Mask* registers in pixels.
11. WrRegister DCNT 2880: set data transfer length for write to FE.
12. RdFrontEnd ClockSelect MaskData : re-write original values into Pixel Shift Register while simultaneously clocking data out from *FE-DO*, through MCC to *MCC-DTO* for check of downloading.

2.3 Module Controller Chip (MCC)

The MCC (see block diagram in Figure 2-4) has 3 basic functions

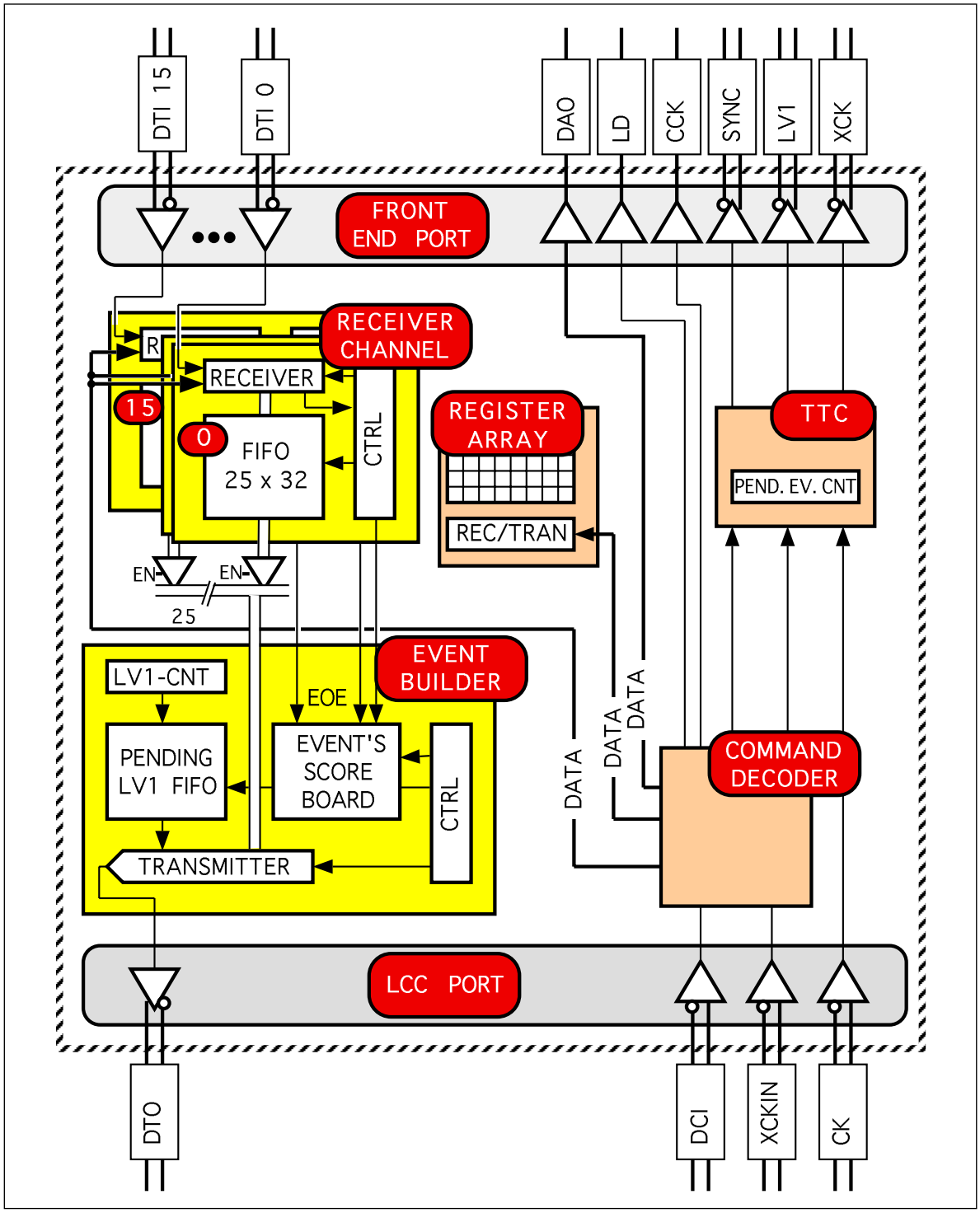

Figure 2-4 MCC block diagram.

- FE chip read out and event building
- Loading of parameter and configuration data into FE chips and into the MCC itself
- Distribution of timing signals like bunch crossing clock and LV1 trigger

The MCC has built in additional testing features to help the debugging of the system.

2.3.1 Internal Registers

The MCC architecture foresee up to 16 general purpose registers of 16 bits. In the MCC for the “Pixel TDR Demonstrator ”only 11 registers have been implemented. Not all the bits are always used and some registers are automatically reset after being read out. The list of registers is in Table 2-2 .

Table 2-2 MCC internal registers

Register	Auto Reset	Address	Content	Description
GCR	No	0000	----,----,---d,dddd	Global Control Register
GSR	Yes	0001	----,----,---d,dddd	Global Status Register
LV1A	No	0010	----,----,dddd,dddd	Level 1 Trigger: LV1A
LV1B	No	0011	----,eeee,dddd,cccc	Level 1 Trigger: LV1B(e,d,c)
FEEN	No	0100	dddd,dddd,dddd,dddd	Front-End Enable
ERR0	Yes	0101	dddd,dddd,dddd,dddd	Error from FE FIFO
WNG0	Yes	0110	dddd,dddd,dddd,dddd	Warning #0 from FE input
WNG1	Yes	0111	dddd,dddd,dddd,dddd	Warning #1 from FE FIFO
ILLC	Yes	1000	-aaa,bbbb,cccc,dddd	Illegal command
Not used		1001		
Not used		1010		
Not used		1011		
Not used		1100		
Not used		1101		
CCNT	No	1110	----,----,-nnn,nnnn	Counter of control bits in FE protocol
DCNT	No	1111	--nn,nnnn,nnnn,nnnn	Counter of data bits in FE protocol

Note
a,b,c,d,e,n,: Used data bit
-: Not existing bit. It is always read back a ‘0’

Registers can be loaded by the LCC using a `command` followed by a `data` stream as shown in Figure 2-5. The incoming data stream is shifted into a receiving shift register (`command / data register`) and eventually copied to the destination register. Destination register bits are only changed if they are different from their previous value. For the case of a write operation, nothing will be sent back to the LCC.

A read operation is started when the LCC sends the appropriate command. The MCC, when it sees the read command, first copies the addressed register contents to the data / command register and then it directs its content to the *DTO* output pin. Any data stream is preceded by a `marker bit` (‘1’) to wake up and synchronize the receivers in the LCC or in the ROD (Read Out Driver).

The implemented MCC registers are:

GCR Global Control Register

Bit definitions for this register are in Table 2-3

GSR Global Status Register

Bit definitions for this register are in Table 2-4

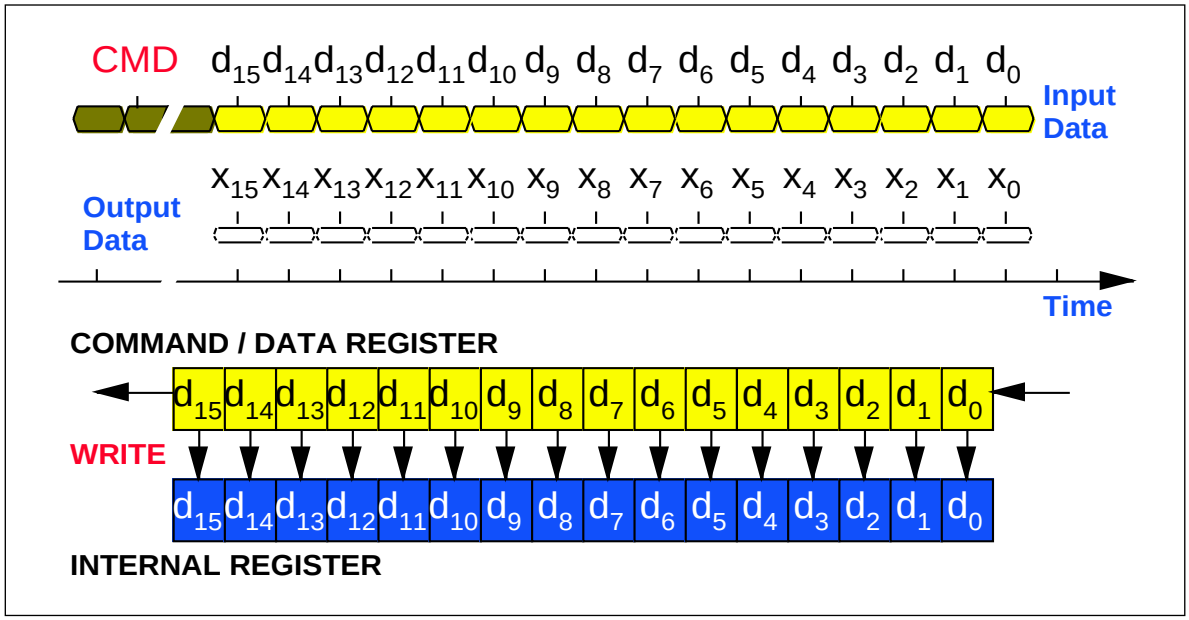

Figure 2-5 Internal Register: Write operation

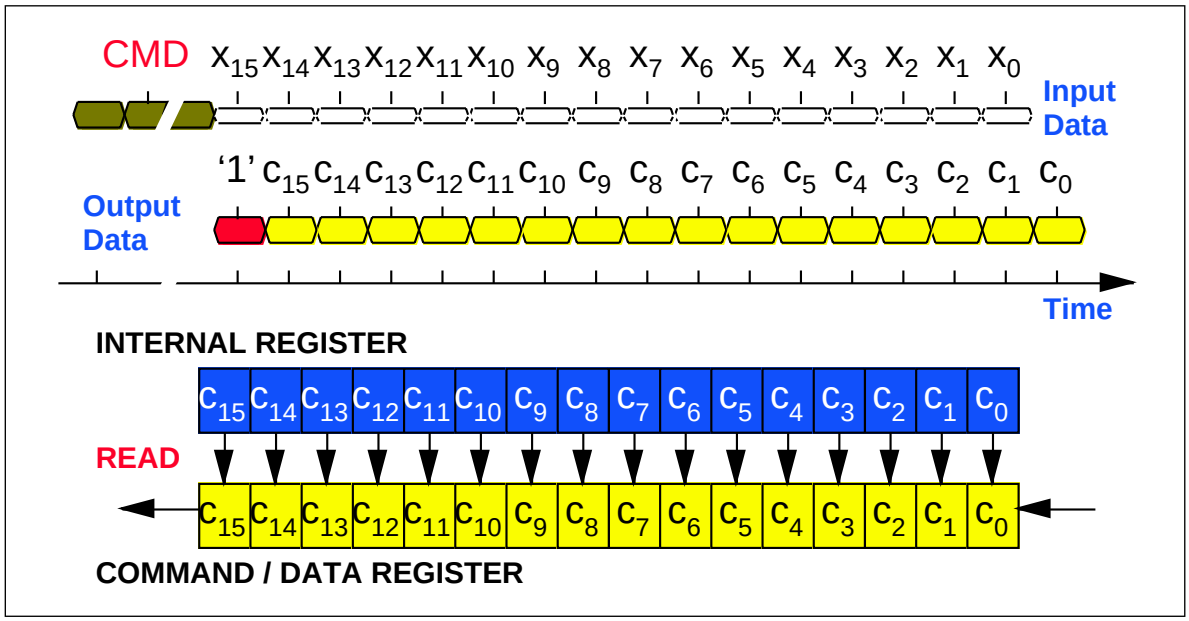

Figure 2-6 Internal Register: Read operation

CCNT Control Counter

Counter to program the number of bits in the `address / command` sent to the FE before the `data` stream. A *CCK* pulse is issued the number of times specified in CCNT, while the *LD* signal stays low. The number in CCNT is in units of *CCK* pulses, which is 1/8 of the master *XCK* clock frequency.

DCNT Data Counter

Counter to program the `data` stream length which follows the `address` / `command` sent to the FE. *CCK* is issued the number of times specified by the content of DCNT. The *LD* signal is raised before the first data bit and lowered at the end of data stream. The number in DCNT is in units of *CCK* pulses, which is 1/8 of the master *XCK* clock frequency.

ERR0 Error #0 Register

FIFO errors (see Section 2.3.5) are recorded in this register.

FEEN Front-End Enable Register

The 16-bits in this registers enable (FEEN<i>='1') or disable (FEEN<i>='0') the corresponding Front-End. In RUN mode only enabled Front-Ends contribute to the final event, while in HALT mode it allows dedicated operations to be performed on enabled FIFO, RECEIVER, etc. An example is when issuing slow commands to the FE chips, this register will control which *FE-DO* is multiplexed out to the *MCC-DTO* (if more than one is selected, the output is the OR of the selected inputs).

ILLC Illegal Command Register

When the *CommandDecoder* detects an illegal command it writes up to 15 bits into this register. As soon as the *CommandDecoder* finds a bit in the serial command stream which is illegal it writes all the received bits (illegal one included) into the ILLC register. If the *CommandDecoder* finds an illegal bit after n correctly interpreted bits it writes '0' from bit 15 to *n+1*, a '1' in position *n* (commnd header) and then follows all the other bits with the illegally interpreted bit in position *0*. Depending where the wrong bit is detected, up to 15 bits and four Fields of the illegal command are copied into ILLC register. The four command fields (see Section 3.3.5) are:

Field1: Trigger or Fast / Slow command

Field2: Type of fast command or switch to slow command

Field3: Type of slow command

Field4: Address (if existing) used by the slow command

LV1A Level 1 Trigger Register A

This 8-bits register is mapped to the LV1 counter. LV1 counter can be loaded with any value and its content can be read back trough this register. Each LV1 trigger increments this counter. Each event block, transmitted by the MCC, includes the content of this counter.

LV1B(e,d,c) Level 1 Trigger Register B

***c <3:0>*:** maximum number of triggered events inside the FE before the MCC blocks new incoming LV1 signals. If a value of 0 is selected no check is performed by the MCC and all incoming LV1 are transmitted to the FE chips.

d <7:4>: number of contiguous LV1 generated by the MCC for each received LV1 command. Contiguous LV1 are truncated if the number of LV1 to be transmitted by any FE chip would become larger than the number set in LV1B(-,-,c). A value of *d*=0 generates a LV1 pulse lasting a single clock unit (no contiguous LV1). The lenght of the LV1 pulse in XCK units is given by *d+1*.

e <11:8>: LV1 latency in clock units (40 MHz). A programmable delay from 0 to 15 clock units can be selected. The MCC-*SYNC* signal is only generated (if GCR<1> is set), when there are no LV1 pending in the FE chips or in the MCC FIFO's and no new LV1 has been decoded by the MCC command decoder. Increasing the value of this delay is a way to protect MCC and FE logics from collision between *SYNC* and *LV1* signals.

WNG0 Warning # 0 Register

Warnings generated by the FE chips are flagged in the data stream and recorded in this register

WNG1 Warning # 1 Register

Warnings generated by the FIFO's (see Section 2.3.5) are flagged in the data stream and recorded in this register.

Table 2-3 Global Control Register bit definition

Bit	Name	To Blocks	Description
0	DPUSH	EVB	Select DataPush / DataRequest mode for event readout.
1	SYNC	TTC	Automatic SYNC generation.
2	TOT	RCH / EVB	Time over Threshold option.
3	ERMSK1	EVB	Mask LV1 number check in FE blocklets
4	ERMSK2	EVB	Mask LV1 number check amongst FE blocklets.
5	DISFEIN		Disable Input from FE chips.

Table 2-4 Global Status Register bit definition

Bit	Name	From Block	Description
0	RUN	CMD	Set by a EnDataTake command, reset by SoftReset.
1	SYFE	TTC	Record of a SynFE or of an internal SYNC action.
2	SYMCC	TTC	Record of a SyncMCC action.
3	ERR1	EVB	Error in FE blocklet
4	ERR1	EVB	Error amongst FE blocklets

2.3.2 Command Set

Commands can be issued to MCC and through it to FE chips by the serial `command / data` line *DCI*. The additional PowerUpReset and pin reset *RSIb* are to force the system to a well-known initial state. Serial command formats are listed in Table 3-12 and Table 3-13 of Section 3.3.5.

There are 3 groups of serial control commands: Trigger, Fast and Slow:

Triggger:LV1 This is the most frequently issued packet and hence the smallest. If this command is received, the MCC transmits a pulse to all FE chips through its *LV1* output pin. LV1 is delayed by the number (d) of clock cycles and fraction of clock cycles (c) specified respectively in LV1(n,d,c) and DLY1(c,d) registers (see Section 2.3.1).

Fast: SyncFE This command generates a MCC-*SYNC* signal in the same way as it is automatically done by the MCC if the SYNC bit is selected in the GCR. If there are pending events in the FE or in the MCC this command is delayed and it is executed when the PendingEventCounter reaches zero. Its purpose is to clear all the data in the chip, while leaving its configuration unaffected. This type of reset could (will) be issued to the chip periodically during data taking to eliminate (prevent) synchronization errors.

Fast: SyncMCC This command does the same as the SyncFE command but it zeroes the LV1 counter in addition. This command is made to resynchronize the MCC in a ladder in a similar way as the SyncFE resynchronizes the FE chips in a module.

Fast: PushEvent This command is used when the MCC is in DataRequest mode (instead of DataPush mode). Its action is to trigger event building and to extract one event (if any) from the MCC FIFO's. In DataPush mode, it has no effect.

Slow: Slow commands have at least four bits present in Field 3. Up to 16 slow commands are supported by the MCC architecture. In the MCC demonstrator several slow commands are implemented (see Table 3-13). They are for reading and writing into the 16 MCC internal registers, into the 16 MCC FIFO's, from and to the FE chips and to the RECEIVERs. For those commands, Field 4 specifies the register address, while Field 5 provides the data to be loaded or 'dummy data' for the case of a read operation.

Slow: WrRegister This command writes into the addressed MCC internal register.

Slow: RdRegister This command reads the content of the addressed MCC internal register.

Slow: WrFifo This command writes a word into the enabled MCC FIFO's. FIFO's are enabled by setting to '1' the corresponding bit in the FEEN register.

Slow: RdFifo This command reads a word from the addressed MCC FIFO.

Slow: WrFrontEnd This command is for writing configuration data into FE chips. The bit stream in Field 5 is sent to the MCC-*DAO* pin. Input data routed to *DAO* have a total length of CCNT + DCNT bits (in *CCK* units). The FE interprets the first CCNT bits as `address / command`, while *LD* is low, and the following DCNT bits as `data`, while *LD* is high. The whole information stream going to the FE must be sent to the MCC with every bit repeated 8 times in order to account for the lower clock frequency used to configure the FE chip.

Slow: RdFrontEnd This command is for reading configuration data from the FE chips. The content of Field 5 is transmitted to the FE chips (usually only `address + command` without any `data`). The FE which recognizes the address responds with the requested data. The MCC adds the header bit to the received data and retransmits them through the MCC-*DTO* pin. The data received from the FE chip by MCC-*DTI* pins are validated by the rising edge of the MCC-*CCK* clock (running at 5 MHz) and successively retransmitted through the MCC-*DTO* pin using the MCC-*XCK* clock (40 MHz). RdFrontEnd can write new data into the addressed FE chip while is reading present data: this is useful when the read operation would destroy the data (like for the *Pixel Shift Register* in Section 2.2.4, "Chip Configure: Registers") and would require a subsequent reload of them. In this last case, the “don’t care” field of the instruction (see Field5 of Table 3-13) carries the new data to be reloaded. The MCC does not affect the input data stream: since the data input clock (MCC-*CCK*) is eight times slower than the output one (MCC-*CK*), each bit from MCC-*DTO*, excluding the header bit, transporting FE configuration data lasts for 8 *CK* clock cycles.

Slow: WrReceiver Input data in Field 5 of this command, feeds into the RECEIVER inputs of the enabled FE receivers. RECEIVER(s) are enabled by the corresponding bit set in the FEEN register. More than one FE can be enabled at one time, meaning that several FE will receive the same data stream. The length of the data stream which is sent to the RECEIVER(s) is the content of the DCNT counter times 8 (*XCK* is used to feed data to the RECEIVER(s)). WrReceiver is foreseen for debugging purposes; this is a way of simulating both the FIFO filling and the event building processes without real data coming from the FE’s.

Slow: EnDataTake This command enable the MCC to data taking. The MCC exits the data taking mode when any slow command is issued.

Slow: SoftReset This command resets all the registers (data and configuration) in the MCC.

2.3.3 System Initialization

It is possible, during system initialization, to read and write into all the MCC internal registers and FIFO’s and to write into the RECEIVERs (see Section 2.3.5, "Event Building"). It is also possible to read and write into the FE chips the configuration data (see Section 2.2.4, "Chip Configure: Registers").

MCC signals that are active during system initialization are graphically represented in Figure 2-7:

CCK: Control Clock to FE.

The 5 MHz *CCK* clock is generated by dividing the X*CKIN* clock by 8 inside the MCC. In This low frequency clock is used in the protocol to load (or read back) the configuration data in the FE chips.

CK: Input Clock from the Ladder Interface

CK is a 40 Mhz clock, meant to be synchronous with the LHC bunch crossing. It is transmitted using the LVDS electrical standard. Both signals from *DTO* and to *DCI* use this reference clock.

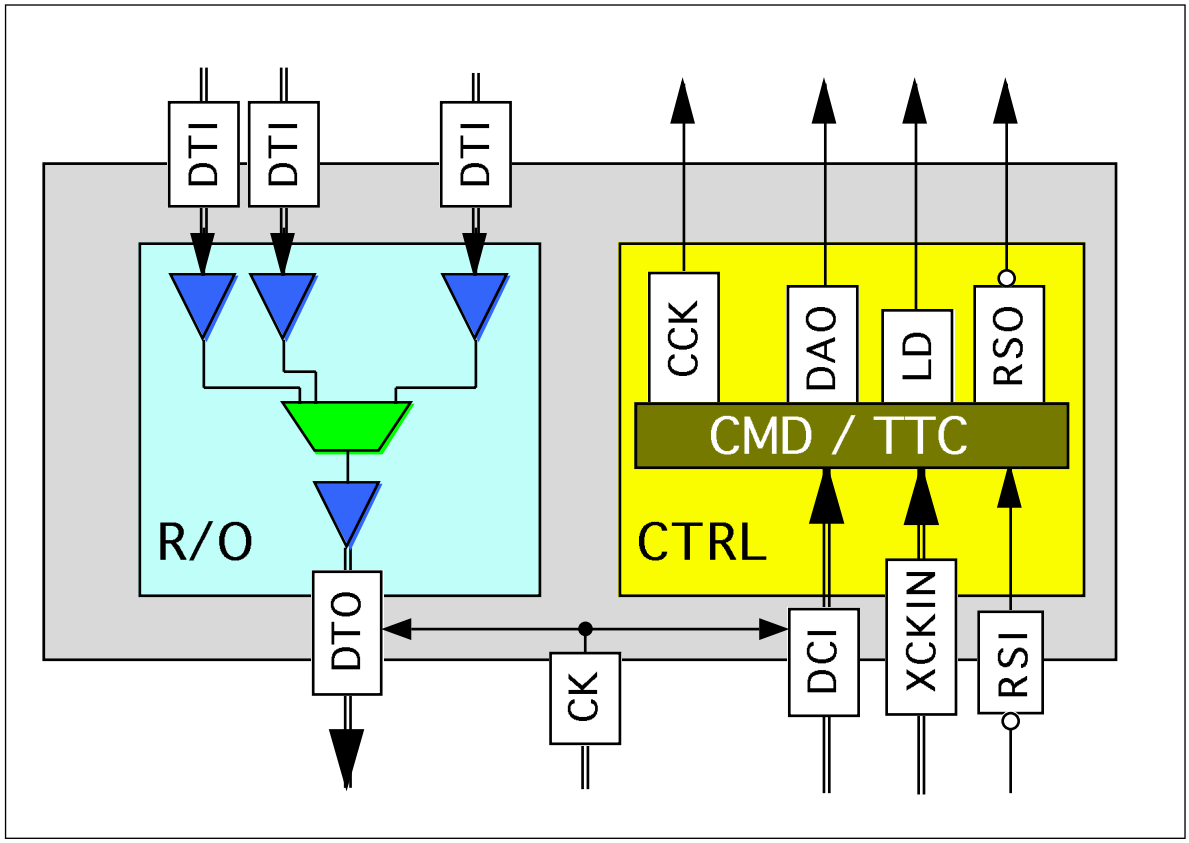

Figure 2-7 MCC signals used to initialize MC & FE chips.

DCI Data and Command Input.

DCI is used to send either data or commands to the MCC. During InitializationMode *DCI* is used for commands or data. The serial protocol use the *CK* clock to validate the data coming to the *DCI* line.

DTI: Data Input from the FE.
Each FE chip uses a separate input line (*MCC-DTI*). The transmission from each FE is serial, it uses low voltage differential signaling (LVDS), and data are in phase with the clock *CCK*. The maximum throughput is 5 Mbit/sec using this data/clock protocol since the clock *CCK* is 5 MHz.

DTO: Data Output to the Ladder Interface.

The MCC retransmits to the ladder interface (LCC) from the *DTO* line. In initialization both internal MCC or FE configuration data can be transmitted from the DTO serial LVDS line. Output data are in phase with the *CK* clock. If data are coming from a FE chip the throughput is 5 Mbit/sec, while for MCC internal data it is 40 Mbit/sec.

LD: Control Clock to FE.

The LD signal is used in the MCC / FE protocol to separate `address + command` (LD low) from `data` (LD high) information.

RSI: Reset Input

Hardware master reset. It is fanned out to the *RSO* line.

RSO: Reset Output.

Reset pin to the FE chips. It is the fanout of the MCC-*RSI*.

XCKIN XSystem clock for the MCC.

The *XCK* output must be externally connected to the *XCKIN* input. The *XCKIN* clock is used as master clock for the MCC,while the *CK* clock is only used to synchronize the *DTO* and *DTI* signals.

2.3.4 Data Acquisition

The architecture that we have chosen for the read-out of the pixel detector is a 3 level hierarchical structure that starts from the FE chips, then goes to the MCC and finally to the Ladder Control Chip (LCC). The interconnections between the FE chips and the MCC are unidirectional links. This interconnection topology is repeated at the upper level from MCC to LCC (see Figure 2-8). The transmission protocol is “data push”: the FE chips send the hit coordinates as

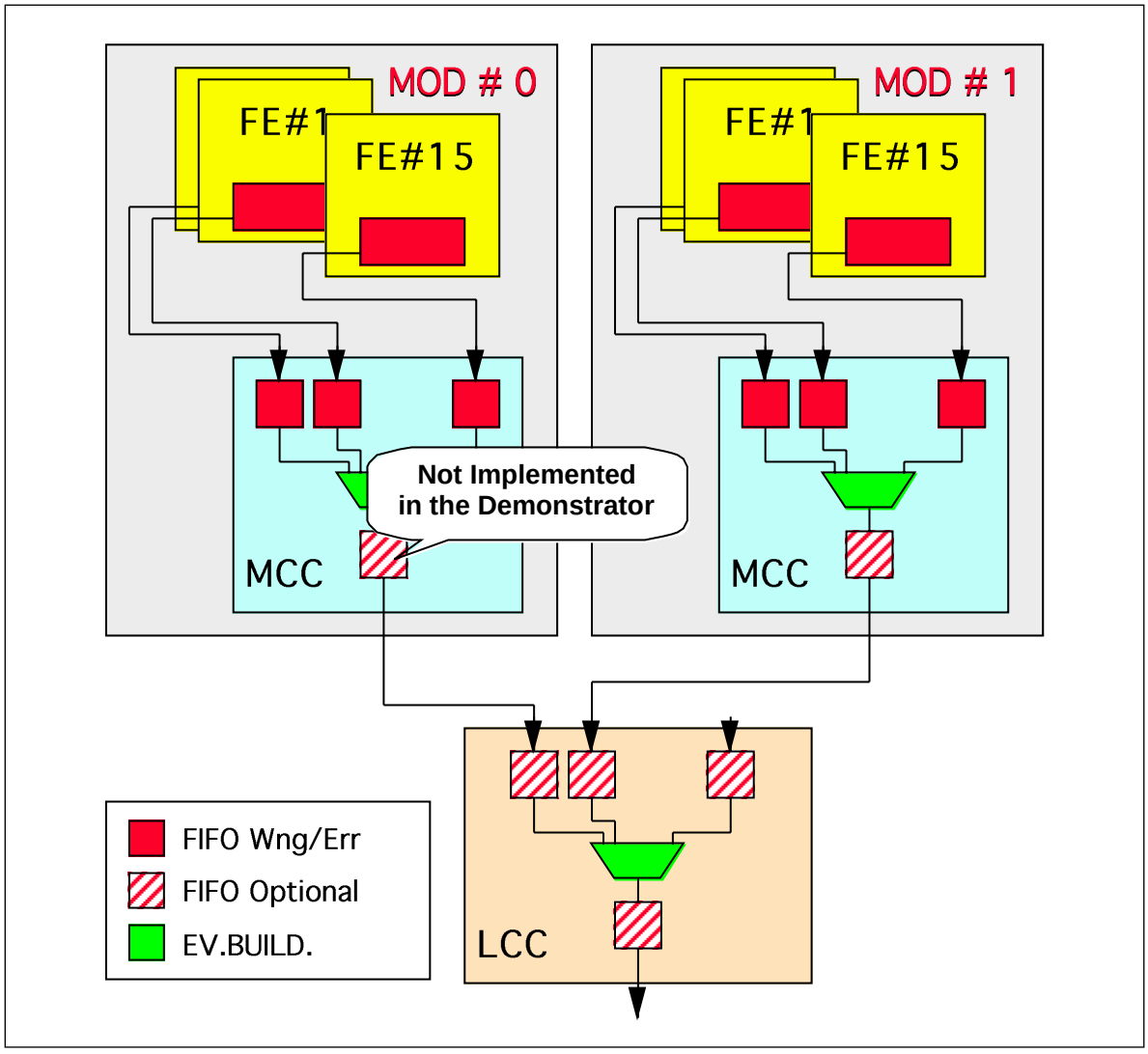

Figure 2-8 Event R/O data flow from FE chip to MCC

soon as they are ready in the end-of-column buffers. Since the chosen architecture is data push, the MCC cannot stop the FE chips from transmitting data.

In DataAcquisitionMode there are two ways for the MCC to interact with the data flow generated by the FE chips:

1. Buffer input data into FIFO's

   The FIFO is necessary to derandomize the hit data sent by the FE and to reduce the data lost due to random fluctuations of the data rate. Since it could always happen that such FIFO's overflow, a Warning / Error mechanism is introduced in the MCC event builder to flag truncated events or to re-synchronize the system when needed.

2. Block the LV1 trigger to the FE's.

   This happens when more than *n* events are still inside the module: either in the FE chips or in the MCC Event Builder (*n* is a parameter, *n=15* is the default). The FE chip transmits each hit together with the LV1 number (modulo 16) and the MCC knows which LV1 number was transmitted last. In case the difference between the two numbers is larger than 15, the MCC blocks the LV1 signal to all the FE's it controls. When the difference becomes lower than or equal to 15, LV1 issuing is resumed. This mechanism will cause some events to be truncated in some detector modules, but will avoid the risk of mixing hits belonging to different events due to lack of synchronization.

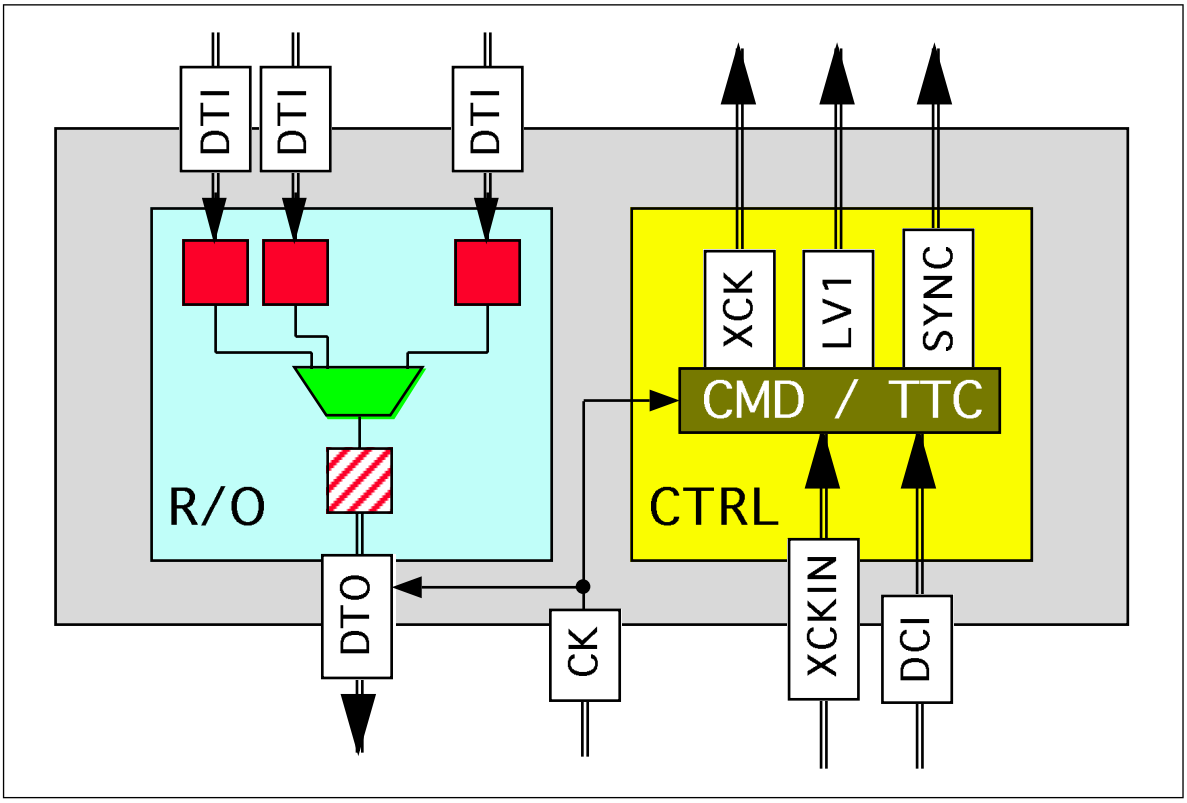

Figure 2-9 MCC signal used by the Event R/O protocol

MCC signals that are active during data acquisition are graphically represented in Figure 2-9:

CK: Input Clock from the Ladder Interface

CK is a 40 Mhz clock, meant to be synchronous with the LHC bunch crossing. It is transmitted using the LVDS electrical standard. Both signals from *DTO* and to *DCI* use this reference clock.

DCI Data and Command Input.

DCI is used to send either data or commands to the MCC. During DataAcquisitionMode *DCI* is used only for commands. The most frequent command received is *LV1* which is encoded using 3 bits, allowing the possibility of generating a trigger every third bunch crossing as required by ATLAS. Other commands are the reset or the resync of the MCC and FE chips.

DTI: Data Input from the FE.
Each FE chip uses a separate input line (*MCC-DTI*). The transmission from each FE is serial, it uses low voltage differential signaling (LVDS), and data are in phase with the clock *XCK*. The maximum throughput is 40 Mbit/sec using this data/clock protocol since the clock *XCK* is 40 MHz.

DTO: Data Output to the Ladder Interface.

The MCC puts together different slices of the same event received from the FE chips and retransmits with appropriate encoding into the readout chain using the *DTO* serial LVDS line. The maximum throughput is 40 Mbit/sec using this data/clock protocol since the clock *CK* is 40 MHz.

LV1: Level 1 trigger.

LV1 signal is generated in response to an 'encoded' trigger signal received by the MCC from the *DCI* input. *LV1* lasts one clock cycle and is synchronous with the *XCK* clock. LV1 can be suppressed in the case that one of the FE's has more than *n* events to send out (*n* is a value stored in the *c* field of the LV1B(e,d,c) register, the default is *n*=15).

SYNC: FE Synchronize.

MCC automatically generates, if the SYNC bit in the GCR is set, a *SYNC* signal every time it knows that there are no events in the FE chips and in the MCC FIFO's. This is done by comparing the number of LV1 issued with the number of events built and sent out by the MCC. When a *SYNC* signal occurs, the FE chips reset all their data registers and trigger counters. *SYNC* lasts one clock cycle and is synchronous with the *XCK* clock. A *SYNC* signal can be also generated by a SyncFE or a SyncMCC command.

XCK: Output clock to the FE chips and to MCC itself.

XCK is generated from the *CK* clock by the MCC. *XCK* has the same frequency as the *CK* clock. In the MCC that will be designed for the ATLAS experiment the *XCK* phase respect to the *CK* will be adjusted inside the MCC to take into account different cable delays from different modules. The delay of *XCK* relative to *CK*, up to a maximum of 25 ns, will be programmable in the MCC. This feature is not implemented in the demonstrator.

XCKIN: System clock for the MCC.

The *XCK* output must be externally connected to the *XCKIN* input. The *XCKIN* clock is used as master clock for the MCC,while the *CK* clock is only used to synchronize the *DTO* and *DTI* signals.

2.3.5 Event Building

Event building is performed by two concurrent processes running in the MCC. The first (Receiver) deals with the filling of the 16 input FIFO's with data received from the corresponding FE chips, while the second one (EventBuilder) extracts data from the FIFO's and builds up the events. Each FE sends data as soon as they are available with two constraints: event hits must be

ordered by event number and for each event an `end-of-event` (*EoE*) word is always generated. *EoE* is also sent for the case of an empty event to keep event synchronization.

The event transmitted to the ladder interface (LCC) is organized by the EventBuilder process on an event-by-event basis, instead of a hit-by-hit basis as it is received from the Receiver process. The EventBuilder knows when to start the event building by checking a “score board” which keeps track of which event is completely stored in each of the 16 input FIFOs. When an event flagged with LV1 number *n* is ready in all the FIFO’s the EventBuilder process takes all the hits from FIFO#0, appends them to that of FIFO#1, and so on up to the last FIFO#15. While EventBuilder fetches the hits out of the FIFO’s, the MCC transmits them upstream to the *DTO* line. The event number is dropped from every hit and sent (only once) together with the FE addresses to the LCC (see Section 3.3.3, "Event Format at MCC Output"). The event building score board is explained in Figure 2-10.

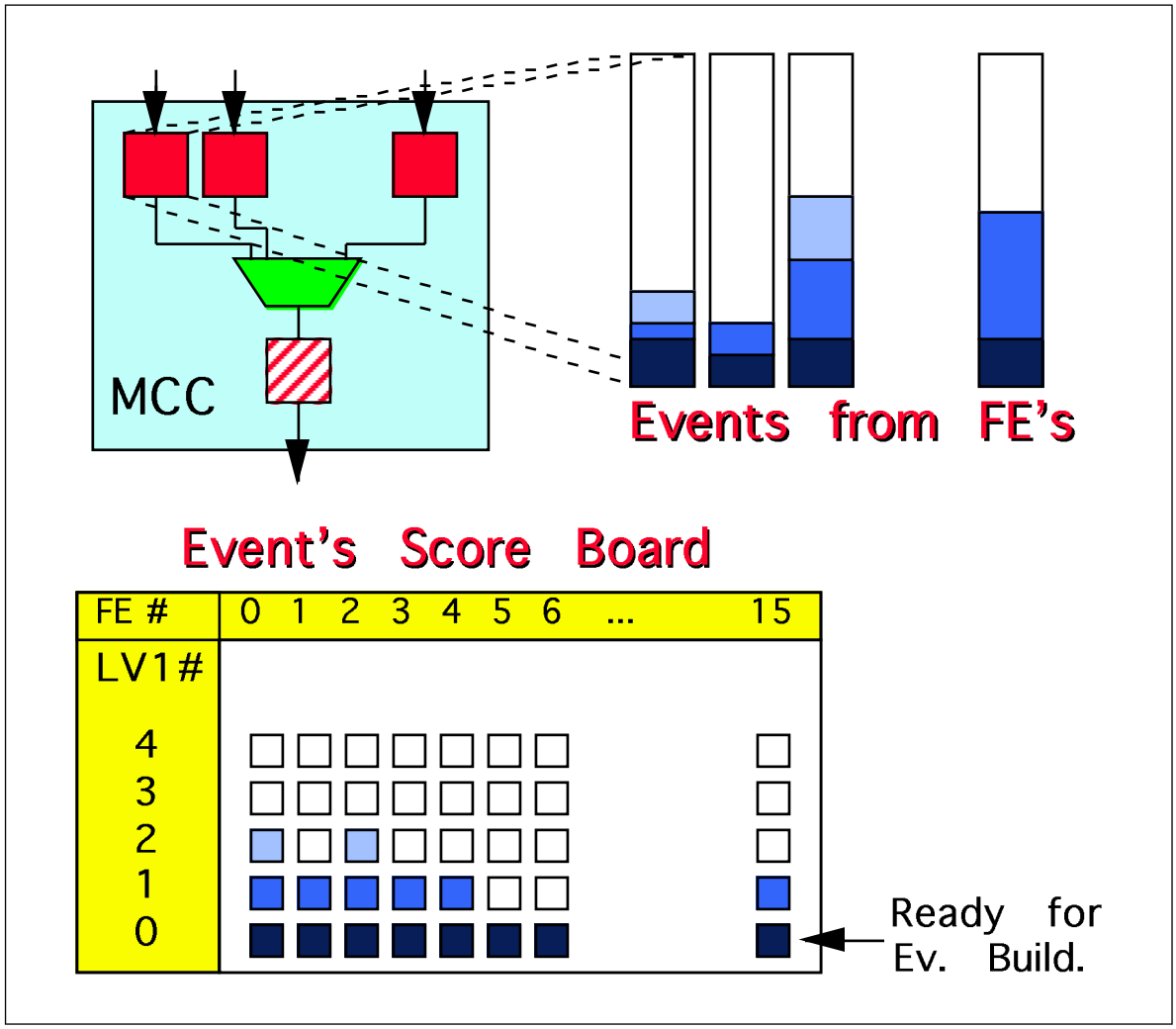

Figure 2-10 Event “score board” used in the MCC for event building

The block diagram of Figure 2-11 is an expanded view of one of the sixteen FIFO’s of Figure 2-8. Looking in more detail, we have the following blocks that together make the Receiver:

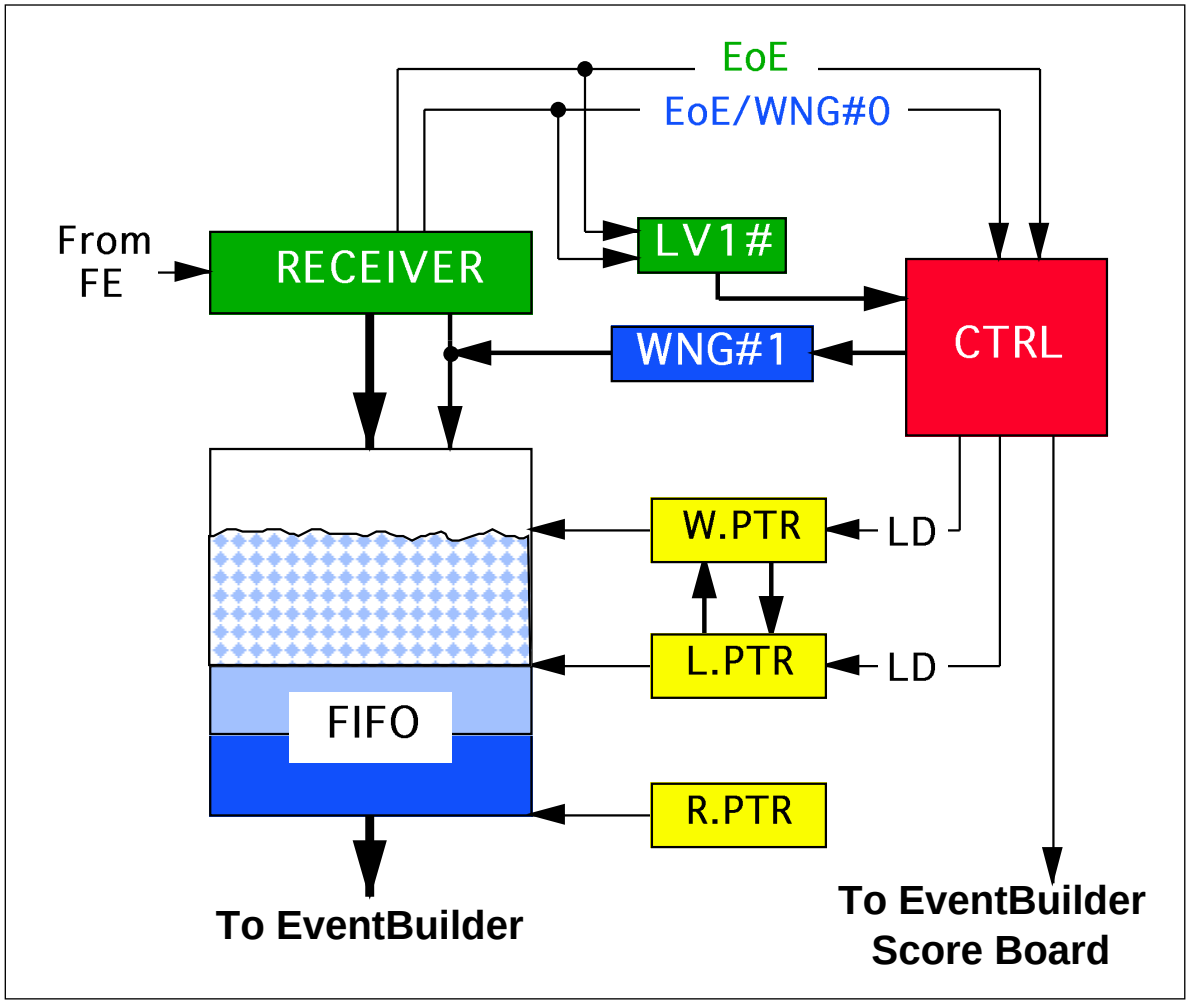

Figure 2-11 Event R/O by MCC: Front-End RECEIVER and input FIFO.

RECEIVER: FE's send data serially to the MCC. The RECEIVER has two functions: to transform serial messages into words to write into the FIFO and to recognize special messages like `end-of-event` (*EoE*) or `front-end-warning` (*WNG#0*) or `front-end-error` (*ERR#0*). *ERR#0* is defined but not implemented in the current demonstrator. The words at the output of the RECEIVER have fixed length (see Section 3.3.1, "FE to MCC: Event Data Format and Protocol"). The 1-bit header is dropped.

FIFO Each word from the RECEIVER is copied into the FIFO. The FIFO has 3 pointers that are incremented, decremented or loaded under the control of the CTRL block.

W.PTR This is the write pointer to the next free FIFO location. Each new word is written into the location pointed to by W.PTR. W.PTR is incremented after the write operation.

L.PTR This pointer always points to the beginning of the current input event. When an *EoE* is recognized the value of W.PTR is copied into L.PTR. L.PTR is used in case there is a FIFO overflow.

R.PTR The read pointer is used by the event builder to fetch data hits. R.PTR is incremented after each word is extracted from the FIFO.

CTRL The CTRL state machine increments, loads or resets W.PTR and L.PTR. It generates the write command to the FIFO. It adds a flag to the scoreboard when it sees an EoE. Finally CTRL treats the errors due to FIFO overflow.

During normal conditions, RECEIVER fills data into the FIFO. When an *EoE* word arrives CTRL

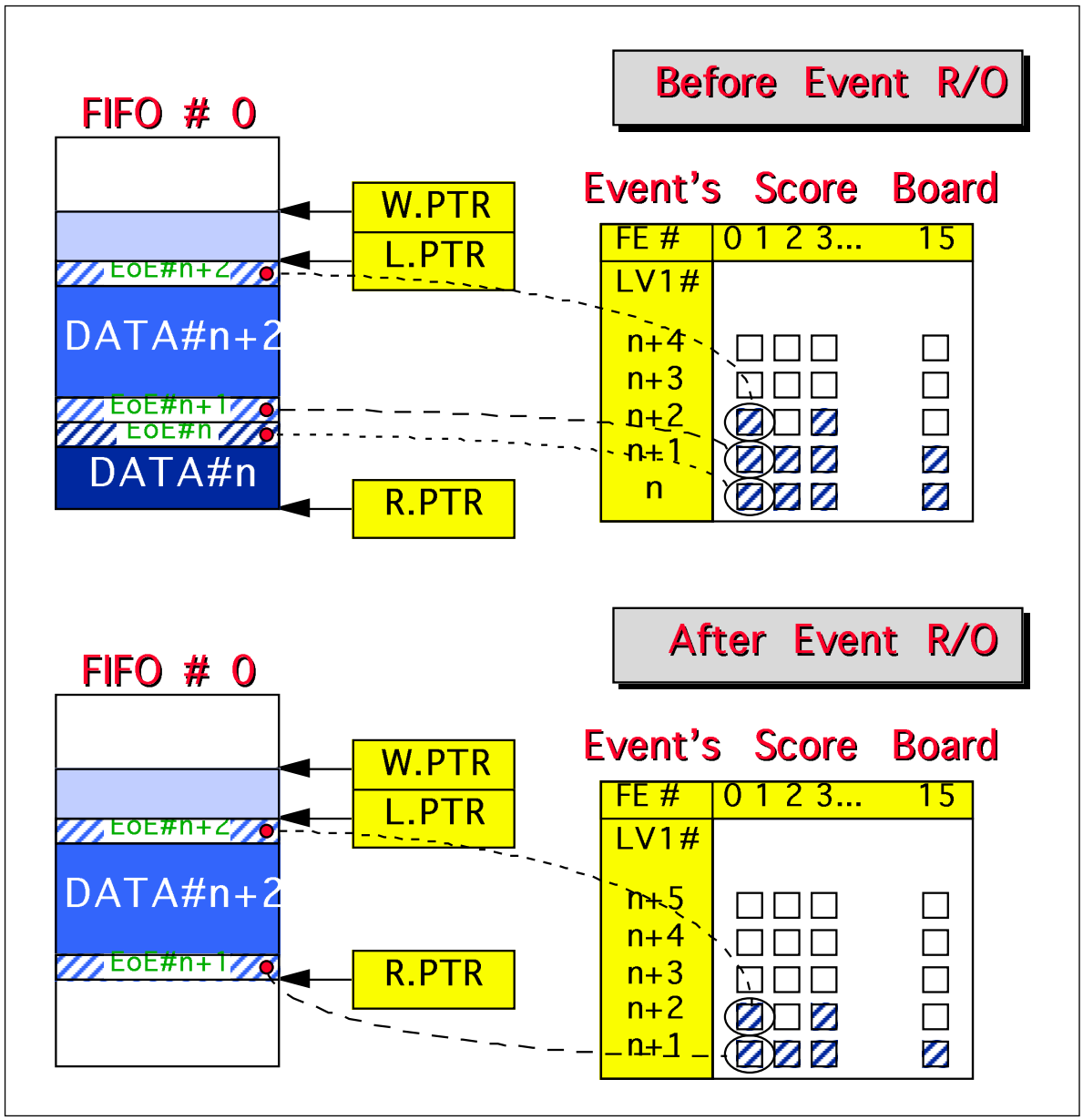

Figure 2-12 Event R/O: the normal case.

copies the contents of W.PTR into L.PTR. When the EventBuilder finds that an event is completely received from all of the 16 FE's (EventBuilder knows from the scoreboard about the existence of complete events) it starts fetching data. After every FIFO read operation its R.PTR is incremented. R.PTR will never overtake L.PTR and EventBuilder shall step to the next FIFO when it finds an *EoE* in the data. The EventBuilder erases for each *EoE* the corresponding flag from the scoreboard. The algorithm just explained is represented in Figure 2-12.

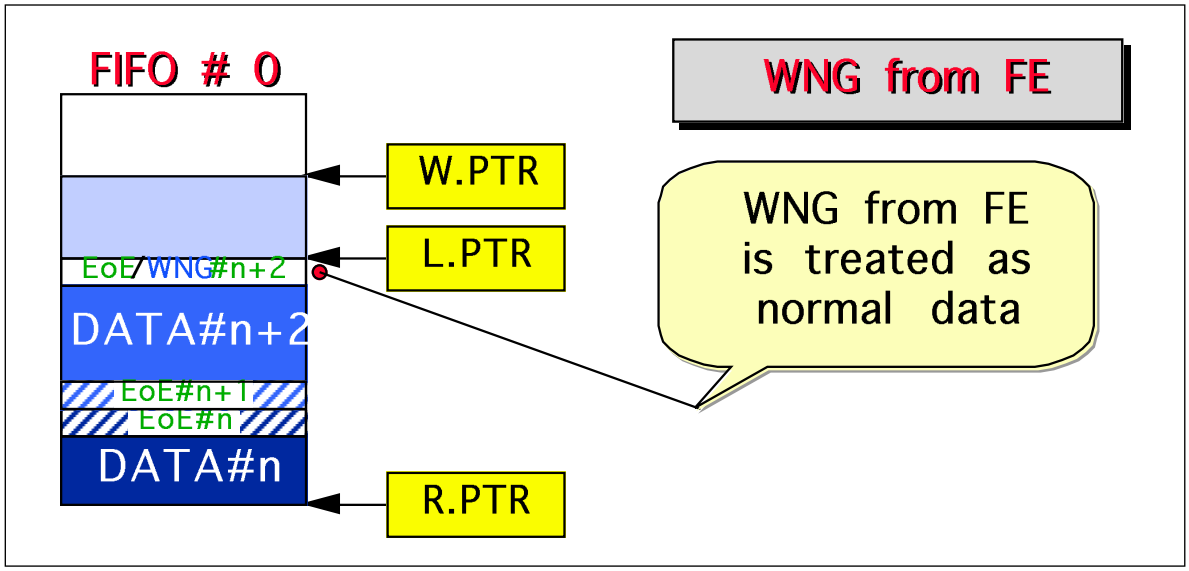

Figure 2-13 Event R/O: WNG generated by the FE chip.

When the FE recognizes that an event has been truncated because it is too large, it will communicate this status with a special *EoE / WNG#0* message. The MCC will treat this data as a normal *EoE* and the event builder will keep this information and it will transmit it upstream. Figure 2-13 explains the algorithm. No *ERR#0* generated by the FE is recognized by the present architecture.

In addition, the MCC can produce `warning` or `error` conditions. A `warning` is a condition which causes a partial event loss and a flagging of the condition in the output data, while an `error` is a destructive condition that cause the loss of one or more events until the MCC recovers. Due to the data push architecture, the overflow of an input FIFO could always happen. The MCC distinguishes between two different cases (shown in Figure 2-14 and Figure 2-15):

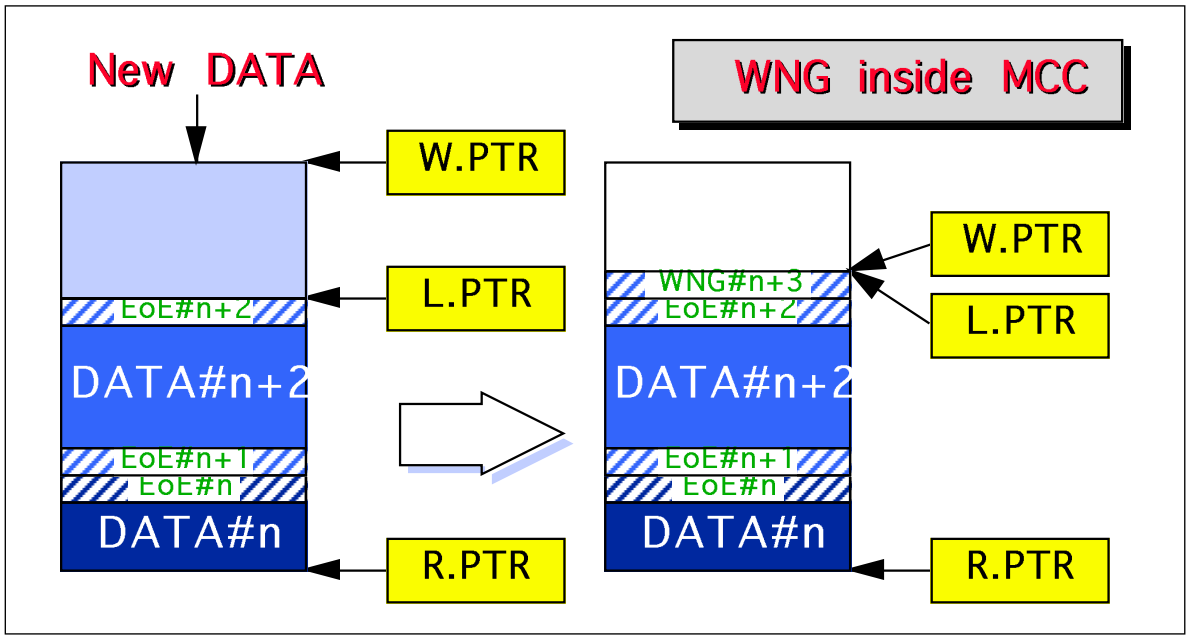

Figure 2-14 Event R/O: WNG generated by the MCC.

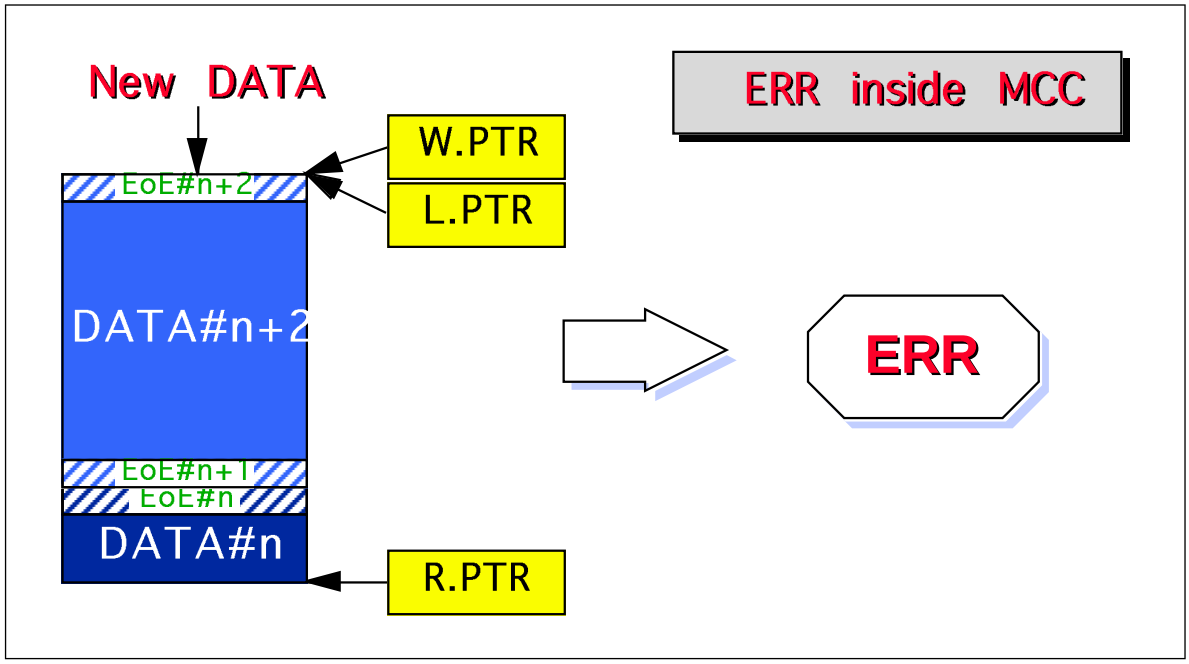

Figure 2-15 Event R/O: ERR generated by the MCC.

MCC WNG: If there is no space to write into the FIFO, but W.PTR is different from L.PTR, we have an internal MCC WNG (*EoE/WNG#1*). In this case, the last partially written event is erased by setting the W.PTR contents back to the L.PTR value. An *EoE/WNG#1* word is written and both L.PTR and W.PTR are incremented. In this way, there is some hope that the next event will be treated correctly either because it will be smaller or because the Event-Builder frees some space in the FIFO .

MCC ERR: If we are in the same situation as before, but L.PTR coincides with W.PTR, CTRL cannot write an *EoE/WNG#1* because it is not possible to obtain some FIFO empty space. In this case, it is not possible to associate unambiguously hits with events, and corruption of data for all of the following events would happen. To recover from this situation, the MCC blocks the writing of data from this FE, blocks *LV1* transmission and after completion of current events sends a *SYNC* signals to all FE's and resets all input FIFOs.

The LV1 counter in each of the FE's is reset each time a *SYNC* is generated. This would cause different pixel detector modules to have different event numbers if a look-up table in each MCC would not correlate a global event number with a local one. Finally, the *SYNC* signal will be issued each time the MCC recognizes that all enabled FE's are empty. This last option in the system is for avoiding or reducing the risk of acquiring many 'out-of-synchronization' events.

2.3.6 Trigger, Timing & Control

This module handles the serial encoded LV1 and generates the timing signals for the whole detector module: MCC-*LV1* and MCC-*SYNC*. In the MCC for the Pixel TDR Demonstrator there is no action of the Trigger Timing and Control (TTC) module on the MCC-*STRO* and on the phase adjust of the MCC-*XCK* needed for different cable delay compensation.

Each time a LV1 command is received by the MCC the TTC compares the content of the PEND.EV.CNT (PendingEventCounter) with the value stored in the *c* field of the LV1B(e,d,c) register: if the counter value is lower than the register content a LV1 is issued to the FE chips and the counter is incremented. If contiguous LV1's are selected in the field *d* of LV1B(e,d,c) the TTC keeps the LV1 signal for a number 'd' of additional *XCK* clock cycles (a value of 0 generates a LV1 pulse lasting a single clock unit: no contiguous LV1). Contiguous LV1 are truncated if the number of LV1 to be transmitted to any FE chip would become larger than the number set in LV1B(-,-,c). The issuing of MCC-*LV1* to the FE chips is delayed by the number of clock cycles selected by the value stored in the field *e* of the LV1B(e,d,c) register. Finally, when an event is completely reconstructed and transmitted off the MCC, the PendingEventCounter is decremented.

The TTC generates automatically MCC-*SYNC* signals each time the PEND.EV.CNT reaches the value 0. A MCC-*SYNC* signal is also issued if an ERR is generated by one of the FIFO controllers (see Section 2.3.5). In this case LV1 generation is blocked inside the TTC and only when the EVB (EventBuilder) has completed the transmission of all the pending events a SYNC is issued and LV1 generation resumed. LV1.CNT in the EVB block is incremented also when the *LV1* to the FE isdropped. This is necessary to keep synchronization between received triggers and transmitted events by the MCC.

2.3.7 Chip Test Features

Some internal logic and 9 input or output pins have been added to increase MCC testability

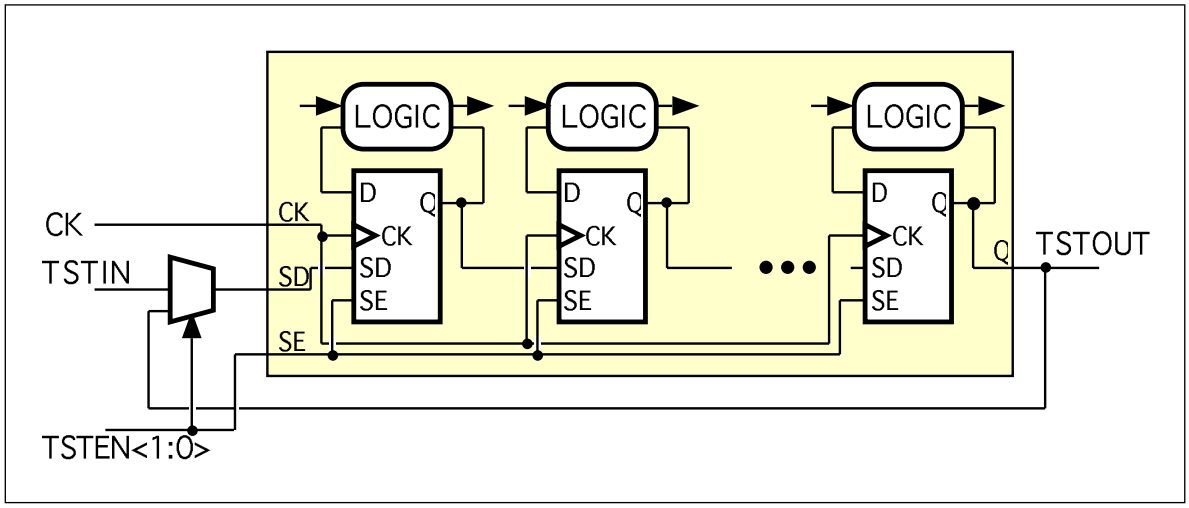

Figure 2-16 Scan Test FF for MCC test.

with a digital wafer tester.

The MCC has 5 LVDS output pins which use current output instead of voltage. They are designed for low currents and low load capacitance. Those requirements make difficult to test the MCC at full speed on wafer. To test the chip functionality with a simpler test machine we have added for each of those outputs a standard CMOS output pin. Additional input pins allows to enable such output pins or to switch them off and use the LVDS outputs as in normal operations. The 5 duplicated output pins are: *DTO*, *LV1*, *STRO*, *SYNC*, *XCK*; they are enabled by MCC-*TSTEN*<1:0> ≠ 0. The parametric test of the 5 LVDS output pins is greatly simplified by the TransparentMode (see Section 2.3.8) where everyone of those lines is directly connected to a corresponding input line.

The other 4 test pins are: *TSTIN*, *TSTOUT* and *TSTEN*<1:0>. These pins are added for the serial scan logic which allows to read or to set to a predefined value each of the MCC internal flip-flops. All the flip-flops inside the MCC are connected into a serial scan chain. The flip-flop chain can be either organized into circular shift registers: the input of the first flip-flop is connected to the output of the last one, being this connected to *TSTOUT* too, or into input-output shift registers: the input of the first flip-flop is connected to *TSTIN* while the output of the last one to *TSTOUT*. In the first case only visibility of internal node is possible and after a number of shifts equal to the number of flip-flops in the chain the internal status of the MCC is restored. In the second case is possible to load any internal configuration while the previous internal status is observed.

The two TSTEN pins allow the programming of the MCC for different test setups:

- TSTEN<1:0> = 0: normal operation;
- TSTEN<1:0> = 1: CMOS output enabled, circular shift (internal observability);
- TSTEN<1:0> = 2: CMOS output enabled, input to output shift (observability and controllability);
- TSTEN<1:0> = 3: CMOS outputs enabled, no shift enabled.

The logics added to improve MCC testability counts for less than 10% of the total number equivalent gates and the principle of its operation is illustrated by the simplified block diagram in Figure 2-16.

2.3.8 FE Test Features: Transparent Mode

The MCC provides additional testing features for the system: it gives direct access to the FE by setting the MCC in ‘transparent’ mode. This behavior is obtained by a special programming pin (*TM* - Transparent Mode) and the duplication of some other input pins which must be connected with their corresponding output pins going to the FE chips (*LDT*, *LV1T* and *CCKT*).

To read back signals from the output of individual FE chips, it would have been necessary to add a pin for each of the 16 FE controlled by the MCC. This would have been too expensive in pin count, especially because the lines are differential. The solution used here is a simple routing of the DTI inputs to the *DTO* output. This data routing is obtained by a 16-bit shift register that enables / disables each individual *DTI* line. The loading of this shift register is done using the *DCI* input data which are clocked in by *CCKT* enabled by TMb = ‘1’ (TransparentMode = False). Figure 2-17 shows a block diagram of the TransparentMode circuitry.

2.4 Multi Chip Module (MCM)

Description needed

2.4.1 Signal Routing and Line Terminations.

Description needed

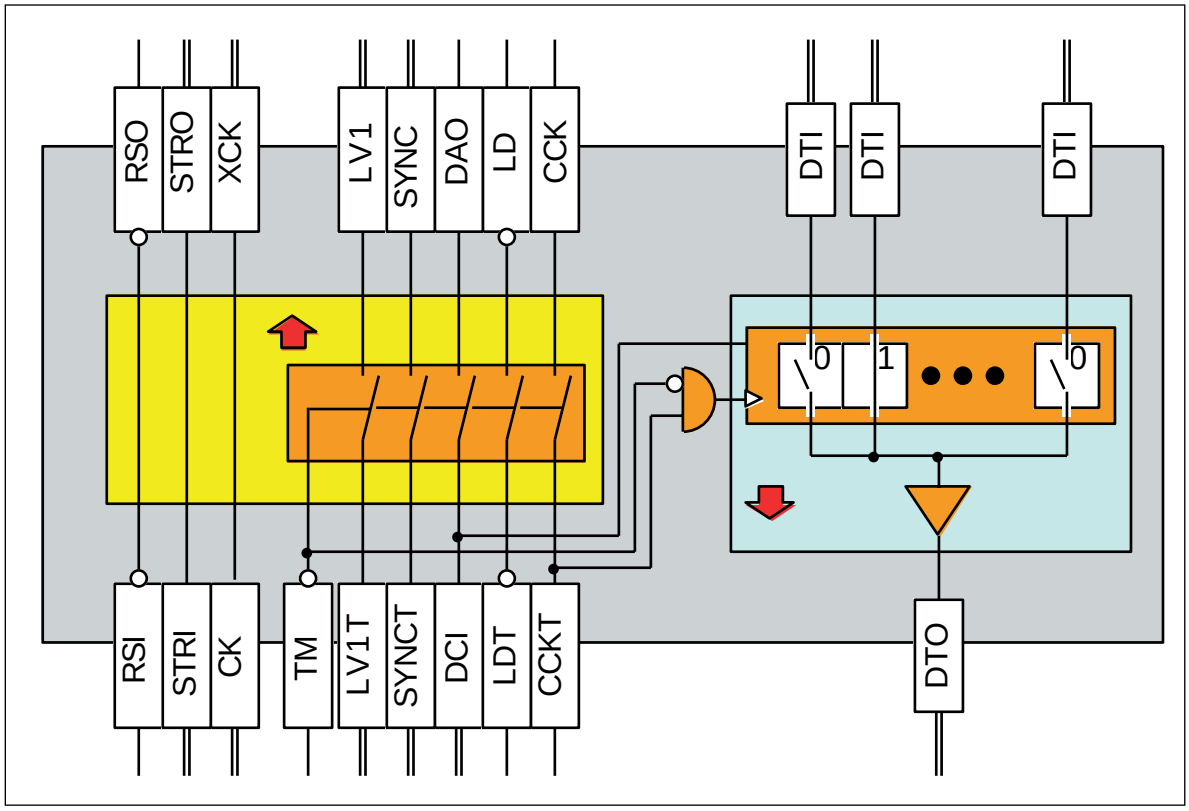

Figure 2-17 MCC System test: Transparent Mode circuitry.

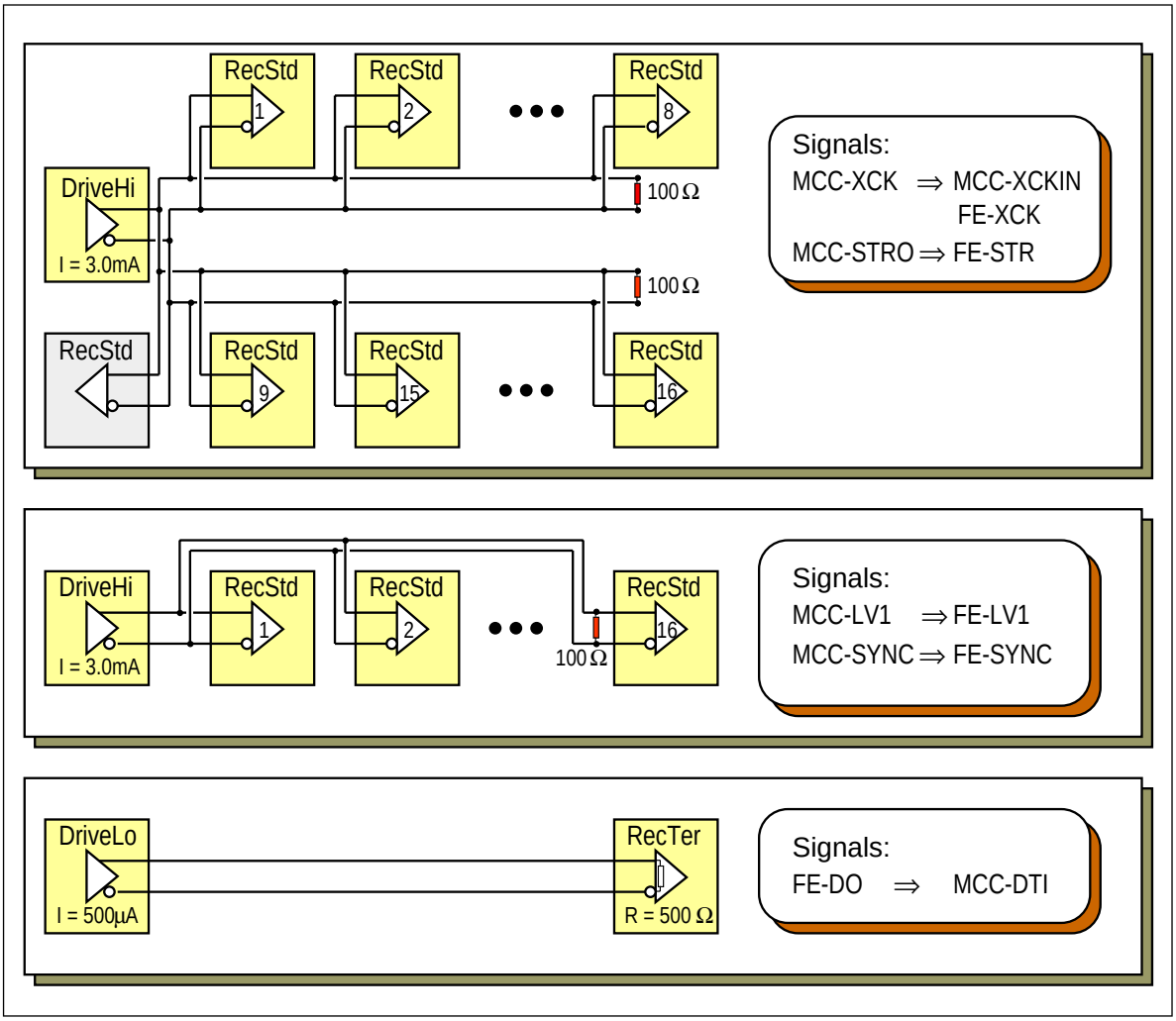

Figure 2-18 Routing and termination for LVDS type signals.

3 Signal Definition & Specifications

3.1 FE, MCC & Module Input / Output pin Definition

3.1.1 FE Chip Pin List and Pad Specifications

For signal names we have used the convention of appending a *"b"* for *ActiveLow* logic, and a *"p"* or *"n"* for the *Positive/Negative* sides of a differential signal. Differential signals have *ActiveHigh* logic. The FE chip has 48 I/O pads in a single row and they are separated by 150 µm.

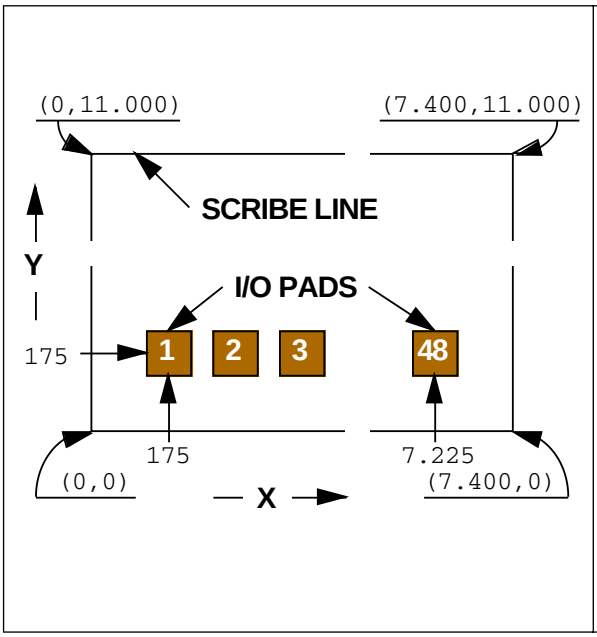

Figure 3-1 FE I/O pad position

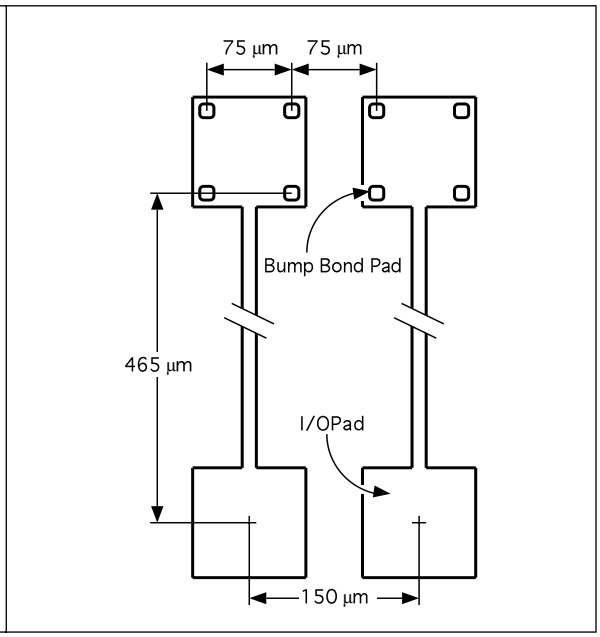

Figure 3-2 Details of layout of combined wire-bond and bump-bond I/O pads.

The pin numbers, pin names and the coordinates of the center of the bonding pad are listed in Table 3-1. Pad coordinates are measured from the left bottom corner of the chip scribe line to the pad center as shown in Figure 3-1. All diagrams of the electronics die assume the reader is looking down on the electronics die from the electronics side (the mirror image of the detector side).

The FE chip area is 7.4×11.0 mm^2. The active detector area is 7.2×8.0 mm^2, while the End-of-Column logic occupies 7.2×2.8 mm^2. A region of 0.1 mm is assigned to the scribe region on all sides of the die, and should contain NO active components. Additional details on the layout of the combined wire-bond and bump-bond I/O pads are shown in Figure 3-2. The bump-bond pads require individual 12µ holes in the passivation for each of the four bumps. The connections between the detector guard-ring and punch-through bias structures and the electronics chips are shown in Figure 3-3. The Guard connection must be made for both Tile1 and Tile2 flavors of detectors, whereas the DGrid connection is only required for Tile2 flavors. All of the connections shown in Figure 3-3 are made via bump-bonding. For the pixel electronics cells, the input metal pad should be a minimum of 20µ diameter with a 12µ passivation opening. For the additional detector contacts for Guard and DGrid, the passivation opening remains 12µ.

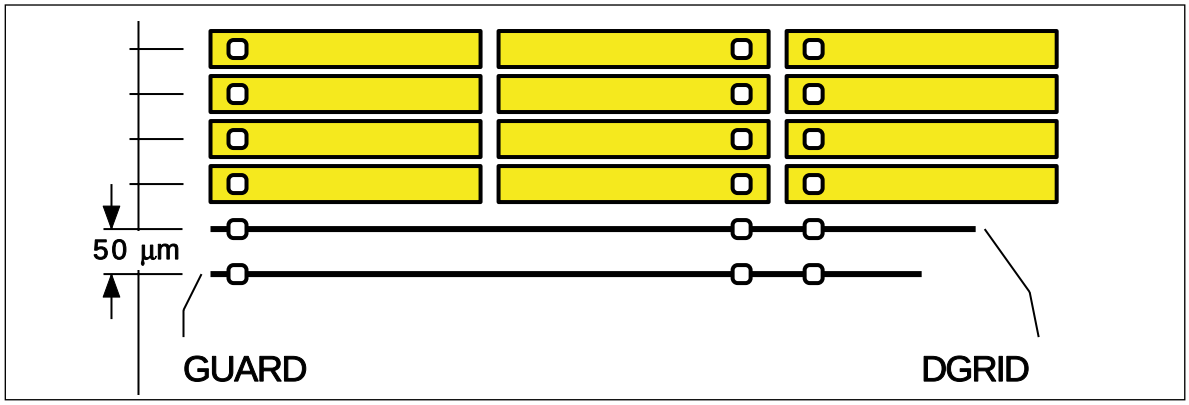

Figure 3-3 Details of interconnections between detector and electronics chip.

Table 3-1 FE chip pin number and coordinates

Pin	NAME	X (mm)	Y (mm)	Pin	Name FE-A	X (mm)	Y (mm)
1	GUARD	0.175	0.175	25	V2	3.775	0.175
2	CCK	0.325	0.175	26	I1	3.925	0.175
3	DI	0.475	0.175	27	I2	4.075	0.175
4	LD	0.625	0.175	28	I3	4.225	0.175
5	STRn	0.775	0.175	29	I4	4.375	0.175
6	STRp	0.925	0.175	30	I5	4.525	0.175
7	GA0	1.075	0.175	31	I6	4.675	0.175
8	GA1	1.225	0.175	32	I7	4.825	0.175
9	GA2	1.375	0.175	33	I8	4.975	0.175
10	GA3	1.525	0.175	34	AVDD	5.125	0.175
11	AVDD	1.675	0.175	35	AVCC	5.275	0.175
12	AVCC	1.825	0.175	36	AGND	5.425	0.175
13	AGND	1.975	0.175	37	DGND	5.575	0.175
14	SHIELD	2.125	0.175	38	DVDD	5.725	0.175
15	DGND	2.275	0.175	39	RSTb	5.875	0.175
16	DVDD	2.425	0.175	40	DOn	6.025	0.175
17	XCKn	2.575	0.175	41	DOp	6.175	0.175
18	XCKp	2.725	0.175	42	MON_HIT1n	6.325	0.175
19	LV1n	2.875	0.175	43	MON_HIT1p	6.475	0.175
20	LV1p	3.025	0.175	44	MON_HIT2n	6.625	0.175
21	SYNCn	3.175	0.175	45	MON_HIT2p	6.775	0.175
22	SYNCp	3.325	0.175	46	MON_AMP	6.925	0.175
23	VCALa	3.475	0.175	47	MON_REF	7.075	0.175
24	V1	3.625	0.175	48	DGRID	7.225	0.175

3.1.1.1 Digital

The definitions for both FE chips are identical, although in practice there may be some slight operational differences (commands implemented, etc.) The pin definitions for both chips *FE-A* and *FE-B* are:

CCK `Control Clock`

Clock to load configuration data (5 MHz)

DI `Data Input`

Serial input for control and configuration data

DO `Data Output`

Serial Data Output. Transmits event hits in data acquisition mode and control / configuration data during system initialization.

GA<3:0> `Geographical Address`

Configuration pins to assign a ‘hard-wired’ address to each chip.

LD `Load`

Separates address plus command from data to be loaded into the internal configuration registers.

LV1 `Level 1 Trigger`

Level 1 trigger accept signal from MCC.

RST `Reset`

Global chip reset

STR `Strobe`

Fast strobe signal for charge injection

SYNC `Synchronize`

Synchronize beam crossing counter and / or LV1-counter

XCK `System Clock`

40 MHz system clock

The list of FE digital input / output pins is in Table 3-2

Table 3-2 FE chip digital I/O Pins

No.	Name	Direction	Active Lev.	Type	Function	Connect To
2	CCK	In	Rise	CMOS	Timing	MCC-CCK
3	DI	In	High	CMOS	Data / Addr	MCC-DAO
40-41	DOn / DOp	Out	Pos / Neg	LVDS	Data	MCC-DTI
7-10	GA<3:0>	In	High-Pu	CMOS	Programing	Pow / Gnd
4	LD	In	High	CMOS	Control	MCC-LD
19-20	LV1n / LV1p	In	Pos / Neg	LVDS	Timing	MCC-LV1
39	RSTb	In	Low-Pu	CMOS	Control	MCC-RSO
5-6	STRn / STRp	In	Pos / Neg	LVDS	Test	MCC-STRO
21-22	SYNCn / SYNCp	In	Pos / Neg	LVDS	Control	MCC-SYNC
17-18	XCKn / XCKp	In	Rise / Fall	LVDS	Timing	MCC-XCK

Note: Active Logic Level

High: Active level High is True
Low: Active level Low is True
Pos: Positive line in differential connection
Neg: Negative line in differential connection
Rise: Rising edge sensitive input
Fall: Falling edge sensitive input
Pu: Pull up (100 KΩ)

3.1.1.2 Analog

This section consists of eleven pins which contain analog voltages or currents that are used to control the behavior of the analog portions of the FE chips. These inputs are divided into two types. There are “current” inputs, which contain an internal current-mode DAC, nominally controlled through the serial command interface, to regulate the behavior of some part of the FE chip. The input pad in this case is assumed to be connected to the output node of the current-mode DAC, and should simply be connected to an external decoupling capacitor to analog ground. A second type of input is a “voltage” input, in which a voltage-mode external DAC is assumed to exist, and is used to control a voltage. Ideally, the only such input would be the VCal input, where due to the large dynamic range required to perform noise measurements and crosstalk measurements, a 12-bit external DAC is needed. Additional “voltage” inputs may be required, and must be supplied by external DACs, again presumably decoupled to analog ground. The nominal division among the 11 analog pins is to provide 8“current” pins and 3“voltage” pins. The pin definitions are:

ITRa *(FE-A)* `???.` *(Signal type: Current)*

ICOMPHa *(FE-B)* `Higher Comparator Threshold` *(Signal type: Current)*

ICOMPLa *(FE-B)* `Lower Comparator Threshold` *(Signal type: Current)*

IDELAYa *(FE-B)* `Threshold Tuning Delay Adjust` *(Signal type: Current)*

Delay adjustment for internal strobe used for threshold tuning.

IDIFFa *(FE-B)* `Differential Amplifier Bias` *(Signal type: Current)*

IFa *(FE-A/B)* `Feedback` *(Signal type: Current)*

Feedback current for preamplifier.

IPa *(FE-A/B)* `Preamplifier` *(Signal type: Current)*

Bias current for preamplifier input transistor.

IPLa *(FE-A)* `Preamplifier Load` *(Signal type: Current)*

Bias current for preamplifier load.

IPSa *(FE-A)* `Preamplifier Source` *(Signal type: Current)*

Bias current for preamplifier source follower.

ISLEWa *(FE-A)* `???` *(Signal type: Current)*

???

IDa *(FE-A)* `???` *(Signal type: Current)*

???

ISTEPa *(FE-B)* `Threshold Tuning Step Size` *(Signal type: Current)*

Current step size for threshold tuning.

ITHRESHa *(FE-B)* `Global Reference for Diff. Amp.` *(Signal type: Current)*

Adjustment for reference voltage on differential amplifier using replica preamplifier in bias cell.

VCALa *(FE-A/B)* `Calibration` *(Signal type: Voltage)*

Charge injection voltage

V1 *(FE-A)* `Comparator` *(Signal type: Voltage)*

Operation point of comparator, also TOT behavior

V1 *(FE-B)* Spare Input Voltage *(Signal type: Voltage)*

Threshold vernier adjustment voltage

V2 *(FE-A)* `Correction` *(Signal type: Voltage)*

V2 *(FE-B)* Spare Input Voltage *(Signal type: Voltage)*

VTHa *(FE-A)* `Threshold` *(Signal type: Voltage)*

Sets global threshold

Analog I/O pins are listed in Table 3-3 for chip FE-A and in Table 3-4 for chip FE-B. We have used the convention to append an *"a"* to all the analog I/O pins

Table 3-3 FE-A chip analog I/O Pins

No.	Pin	Name	Dir.	Type	Range	Nominal	Connect To
28	I3	ITRa	In/Out	Current	3 ÷ 8 µA	6 µA	Ext.Test
30	I5	IFa	In/Out	Current	1 ÷ 10 nA	3 nA	Ext.Test
31	I6	IPa	In/Out	Current	6 ÷ 13 µA	10 µA	Ext.Test
32	I7	IPLa	In/Out	Current	1 ÷ 5 µA	3 µA	Ext.Test
33	I8	IPSa	In/Out	Current	1 ÷ 5 µA	3 µA	Ext.Test
27	I2	ISLEWa	In/Out	Current			Ext.Test
29	I4	IDa	In/Out	Current			Ext.Test
23		VCALa	In	Voltage			Ext.Test
24	V1	VTHa	In	Voltage	1 ÷ 2 V		Ext.Test
25	V2	VCCDa	In	Voltage			Ext.Test
26	I1	IUNUSEDa			1 .0÷ 1.5 V		Ext.Test

Table 3-4 FE-B chip analog I/O Pins

No.	Pin	Name	Dir.	Type	Range	Nominal	Connect To
27	I2	ICOMPHa	In/Out	Current	18*[0.5,5.0] µA	1.5 µA	Ext.Test
28	I3	ICOMPLa	In/Out	Current	18*[0.5,5.0] µA	1.5 µA	Ext.Test
32	I7	IDELAYa	In/Out	Current			Ext.Test
29	I4	IDIFFa	In/Out	Current	18*[0.5,5.0] µA	1.5 µA	Ext.Test
30	I5	IFa	In/Out	Current	18*[0.5,20] nA	5 nA	Ext.Test
31	I6	IPa	In/Out	Current	18*[1,10] µA	5 µA	Ext.Test
33	I8	ISTEPa	In/Out	Current			Ext.Test
26	I1	ITHRESHa	In/Out	Current	18*[0.5,5.0] µA	1.5 µA	Ext.Test
23		VCALa	In	Voltage	[AVdd,0] V	AVdd	Ext.Test
24	V1		In	Voltage			Ext.Test
25	V2		In	Voltage			Ext.Test

3.1.1.3 Power

The FE chip has two digital and three analog power supply connections. In addition, there is a shield, a detector guard ring and a detector bias pin. These connections are identical for the two

FE chips. The power supply pins, along with power estimates from existing chips, are listed in Table 3-5 .

Table 3-5 FE chip power supply pins

No.	Name	Type	Voltage	Current
11,34	AVDD	Analog	+ 3.0 V	26 mA @ 3V (1)
12,35	AVCC	Analog	+ 1.5 V	37 mA @ 1.5V (2)
13,36	AGND	Analog	0.0 V	63 mA (3)
16,38	DVDD	Digital	+3.0 V	
15,37	DGND	Digital	0.0 V	
1	GUARD	Detector Guard	0.0 V	
48	DGRID	Detector Grid		
14	SHIELD	Shield	0.0 V	

Note:
(1) Formula: I(AVDD) = (IPLa+ICOMPa) x Row# x Col#
(2) Formula: I(AVCC) = (IPSa+IPa) x Row# x Col#
(3) Formula: I(AGND) = (I(AVDD)+ I(AVCC))

3.1.1.4 Test and Monitoring:

There are presently six pins reserved for test and monitoring of the FE chips. A possible assignment for some of these pins, planned for use in *FE-B*, is shown below:

MON_HIT1n HitBus differential output (negative)

MON_HIT1p HitBus differential output (positive)

MON_HIT2n Buffered discriminator differential output from test pixel (negative)

MON_HIT2p Buffered discriminator differential output from test pixel (positive)

MON_AMP Buffered preamplifier output of test pixel

MON_REF free

3.1.2 MCC Pin List

For signal names we have used the convention of appending a *"b"* for *ActiveLow* logic, a *"p"* or *"n"* for *Positive/Negative* in a differential signal. Differential signals have *ActiveHigh* logic. Pin numbers, pin names and coordinates of the center of the bonding pad are listed in Table 3-6.

3.1.2.1 Digital

Pin definition:

CK: `Clock.`

Input clock (40 MHz).

CCK: `Control Clock.`

Control clock to write configuration parameters into the FE chips

Table 3-6 MCC pin number and coordinates

Pin No.	Name	X (mm)	Y (mm)	Pin No.	Name	X (mm)	Y (mm)
1	GND_1	10.500	3.175	43	VDD_4	0.105	3.225
2	VDD_1	10.500	3.425	44	XCK	0.105	2.975
3	TSTOUT	10.500	3.675	45	XCKn	0.105	2.725
4	DCIn	10.500	3.925	46	XCKp	0.105	2.575
5	DCIp	10.500	4.075	47	XCKINn	0.105	2.325
6	CKn	10.500	4.325	48	XCKINp	0.105	2.175
7	CKp	10.500	4.475	49	SYNCn	0.105	1.925
8	LDT	10.500	4.725	50	SYNCp	0.105	1.775
9	SYNCTn	10.500	4.975	51	SYNC	0.105	1.525
10	SYNCTp	10.500	5.125	52	GND_5	0.105	1.275
11	TSTEN0	10.500	5.375	53	VDD_5	0.105	1.025
12	TSTEN1	9.175	6.195	54	DTI8n	1.425	0.105
13	RSOb	8.625	6.195	55	DTI8p	1.575	0.105
14	DAO	8.075	6.195	56	DTI9n	2.125	0.105
15	DTI0n	7.525	6.195	57	DTI9p	2.275	0.105
16	DTI0p	7.375	6.195	58	DTI10n	2.825	0.105
17	DTI1n	6.825	6.195	59	DTI10p	2.975	0.105
18	DTI1p	6.675	6.195	60	DTI11n	3.525	0.105
19	GND_2	6.125	6.195	61	DTI11p	3.675	0.105
20	VDD_2	5.575	6.195	62	DTI12n	4.225	0.105
21	DTI2n	5.025	6.195	63	DTI12p	4.375	0.105
22	DTI2p	4.875	6.195	64	DTI13n	4.925	0.105
23	DTI3n	4.325	6.195	65	DTI13p	5.075	0.105
24	DTI3p	4.175	6.195	66	GND_0	5.625	0.105
25	DTI4n	3.625	6.195	67	VDD_0	6.175	0.105
26	DTI4p	3.475	6.195	68	DTI14n	6.725	0.105
27	DTI5n	2.925	6.195	69	DTI14p	6.875	0.105
28	DTI5p	2.775	6.195	70	DTI15n	7.425	0.105
29	DTI6n	2.225	6.195	71	DTI15p	7.575	0.105
30	DTI6p	2.075	6.195	72	LD	8.125	0.105
31	DTI7n	1.525	6.195	73	CCK	8.675	0.105
32	DTI7p	1.375	6.195	74	TSTIN	9.175	0.105
33	GND_3	0.105	5.275	75	LV1Tn	10.500	0.975
34	VDD_3	0.105	5.025	76	LV1Tp	10.500	1.125
35	n/c	---	---	77	CCKT	10.500	1.375
36	LV1	0.105	4.775	78	RSIb	10.500	1.625
37	LV1n	0.105	4.525	79	TMb	10.500	1.875
38	LV1p	0.105	4.375	80	STRIn	10.500	2.125
39	STROn	0.105	4.125	81	STRIp	10.500	2.275
40	STROp	0.105	3.975	82	DTOn	10.500	2.525
41	STRO	0.105	3.725	83	DTOp	10.500	2.675
42	GND_4	0.105	3.475	84	DTO	10.500	2.925

CCKT `Control Clock Transparent.`

Control clock transmitted to the *CCK* output pin when the MCC is in TransparentMode.

DAO: `Data/Address Output.`

When *LD* is *Low*, FE `address` plus `command` is transmitted, when *LD* is *High*, data are transmitted.

DCI: `Data/Command Input.`

MCC receives data, commands and timing signals from this pin.

DTI: `Data Input.`

Data Input from FE chips

DTO: `Data Output.`

Data Output to LCC hybrid

LD: `Load.`

Load line used to separate address plus command from data sent to the FE chips.

LDT: `Load Transparent.`

Load signal which is fanned out to *LD* pin in TransparentMode.

LV1: `Level 1 Trigger.`

Level 1 Trigger pulse transmitted to the FE chips.

LV1T: `Level 1 Trigger Transparent Mode.`

Level 1 Trigger pulse received and retransmitted to FE chips when the MCC is put in TransparentMode.

RSI: `Reset Input.`

Input master reset.

RSO: `Reset Output.`

Fanout of *RSI* to the FE chips.

SYNC: `Synchronize.`

Signal to synchronize FE's and MCC LV1 counters.

SYNCT: `Synchronize Transparent.`

Synchronize pulse fanout to FE chips in TransparentMode.

STRI: `Test Charge Input Strobe.`

Strobe to inject test charge. Fanned out to *STRO* pin.

STRO: `Strobe Output.`

Strobe to inject test charge at the FE pixel input amplifiers

TM: `Transparent Mode.`

Special test mode that put the MCC in TransparentMode.

TSTEN `Test Enable.`

Two bits to enable the MCC internal test mode.

TSTIN `Test Input.`

Serial scan data input.

TSTOUT `Test Output.`

Serial scan data output.

XCK `System Clock.`

Fanout of the 40 MHz input clock.

XCKIN System Clock Input.

System Clock used by the MCC.

The list of MCC digital input / output pins is in Table 3-7,

Table 3-7 MCC Digital I/O Pins

No.	Name	Direction	Active Lev.	Type	Function	Connect To
1-2	CKp / CKn	In	Pos / Neg	LVDS	Timing	External
3	CCK	Out	Rise	CMOS	Timing	FE-CCK
4	CCKT	Input	Rise-Pd	CMOS	Test / Timing	Ext. Test
5	DAO	Out	High	CMOS	Data / Addr	FE-DI
6-7	DCIp / DCIn	In	Pos / Neg	LVDS	Data / Cmd	External
8-39	DTIp<15:0> / DTIn<15:0>	In	Pos / Neg	LVDS	Data	FE-DO
40	DTO	Out	High	CMOS	Chip Test	Chip Test
41-42	DTOp / DTOn	Out	Pos / Neg	LVDS	Data	External
43	LD	Out	High	CMOS	Control	FE-LD
44	LDT	In	High-Pd	CMOS	Test / Control	External
45	LV1	Out	High	CMOS	Chip Test	Chip Test
46-47	LV1p / LV1n	Out	Pos / Neg	LVDS	Control	FE-LV1
48-49	LV1Tp / LV1Tn	In	Pos / Neg	LVDS	Test / Control	Ext. Test
50	RSIb	In	Low-Pu	CMOS	Control	External
51	RSOb	Out	Low	CMOS	Control	FE-RST
52	SYNC	Out	High	CMOS	Chip Test	Chip Test
53-54	SYNCp / SYNCn	Out	Pos / Neg	LVDS	Control	FE-SYNC
55-56	SYNCTp / SYNCTn	In	Pos / Neg	LVDS	Test / Control	Ext. Test
57-58	STRIp / STRIn	In	Pos / Neg	LVDS	Test / Timing	External
59	STRO	Out	High	CMOS	Chip Test	Chip Test
60-61	STROp / STROn	Out	Pos / Neg	LVDS	Timing	FE-STR
62	TMb	In	Low-Pu	CMOS	Test / Control	Ext. Test
63-64	TSTEN<1:0>	In	High-Pd	CMOS	Chip Test	Chip Test
65	TSTIN	In	High-Pd	CMOS	Chip Test	Chip Test
66	TSTOUT	Out	High	CMOS	Chip Test	Chip Test
67	XCK	Out	Rise	CMOS	Chip Test	Chip Test
68-69	XCKp / XCKn	Out	Pos / Neg	LVDS	Timing	FE-XCK
70-71	XCKINp / XCKIN / n	In	Pos / Neg	LVDS	Timing	MCC-XCK

Note: Active Logic Level

High: Active level High is True
Low: Active level Low is True
Pos: Positive line in differential connection
Neg: Negative line in differential connection
Rise: Rising edge sensitive input
Fall: Falling edge sensitive input
Pd: Pull Down (100 KΩ)
Pu: Pull Up (100 KΩ)

3.1.2.2 Power

The MCC has only two power supplies. They are listed in Table 3-8.

Table 3-8 MCC power supply pins

No.	Name	Type	Voltage	Current
1-5	VDD	Digital	+ 3.0 ÷ +5.0 V	0 mA @ 3V
6-10	VSS	Digital	0.0 V	-0 mA @ 3 V

3.1.3 MCM Signal List

3.1.3.1 Digital

(In this paragraph there will be a description of all digital signals in the Multi Chip Module)

3.1.3.2 Analog

(In this paragraph there will be a description of all analog signals in the Multi Chip Module)

3.1.3.3 Power

(In this paragraph there will be a description of all power signals in the Multi Chip Module)

3.2 Input/Output Pin Electrical Specifications

3.2.1 CMOS Type Pins

CMOS Output buffers provide 2 mA output current. Input buffers can have either a pull-up (“Pu”) or a pull-down (“Pd”) resistance. Such resistance is to keep a ‘1’ (“Pu”) or a ‘0’ (“Pd”) logical value if the input is unconnected. Inputs without “Pu” or “Pd” must be always driven.

3.2.2 LVDS Type Pins

Input or output pads inproperly called “LVDS” (Low Voltage Differential Signalling) are differential sense voltage receivers or current transmitters. The set of LVDS drivers includes 3 transmitters and 2 receivers:

DriveLo: Low Current Differential Driver

Type:	Output
Output Current:	I_O = 250 µA
Output Voltage:	V_{Od} = 125 mV differential @ R_T = 500 Ω
Max Load Cap:	C_{Lmax} = 10 pF
Propag. Delay:	$t_{dl} \leq$ 5 nS (?) @ 10 pF
Termination:	R_T = 500 Ω differential
Use:	point-to-point links between MCC and FE

DriveHi: High Current Differential Driver

Type:	Output
Output Current:	I_O = 3.0 mA
Output Voltage:	V_{Od} = 300 mV differential @ R_T = 100 Ω
Propag. Delay:	$t_{dl} \leq$ 5 nS (?) @ $R_L = Z_0$
Termination:	R_T = 100 ÷ 120 Ω differential
Use:	point-to-multipoint from MCC to FE, MCC to LCC links, Clock (XCK) and strobe (STRO) distribution from MCC to FE's.

RecStd: Standard Receiver

Type:	Input
Input Voltage	V_{Imin} < 25 mV differential
Termination:	none
Use:	point-to-multipoint from MCC to FE, MCC to LCC links.

RecTer: Terminated Receiver

Type:	Input
Input Voltage	V_{Imin} < 25 mV differential
Termination:	R_T = 500 Ω differential
Use:	point-to-point links from FE to MCC.

FE-A/B and MCC signals of LVDS type are listed in Table 3-9. Termination of LVDS lines are shown in Figure 2-18 of Section 2.4.1.

3.3 Signal Formats & Protocols

3.3.1 FE to MCC: Event Data Format and Protocol

The FE chip encodes and transmits hits from an event using fixed length packets. Each packet contains a 1-bit header, followed by a 4-bit LV1, an 8-bit row number, a 5-bit column number, and an optional 8-bit TOT value. The encoding uses a NRZ coding synchronous with the 40

Table 3-9 LVDS type drivers and receivers in the FE-A/B and MCC chips.

Chip	Pin Name	Dir	LVDS Type	Connect to	Line Termination
FE	DOn / DOp	Out	DriveLo	RecTer	500 Ω internal
	LV1n / LV1p	In	RecStd	DriveHi	100 Ω external
	STRn / STRp	In	RecStd	DriveHi	100 Ω + 100 Ω external
	SYNCn / SYNCp	In	RecStd	DriveHi	100 Ω external
	XCKn / XCKp	In	RecStd	DriveHi	100 Ω + 100 Ω external
MCC	CKn / CKp	In	RecStd	1.5 ÷ 4.0 mA	100 Ω external
	DCIn / DCIp	In	RecStd	1.5 ÷ 4.0 mA	100 Ω external
	DTIn / DTIp	In	RecTer	DriveLo	500 Ω internal
	DTOn / DTOp	Out	DriveHi	RecStd	100 Ω external
	LV1n / LV1p	Out	DriveHi	RecStd	100 Ω external
	LV1Tn / LV1Tp	In	RecStd	1.5 ÷ 4.0 mA	100 Ω external
	SYNCn / SYNCp	Out	DriveHi	RecStd	100 Ω external
	SYNCTn / SYNCTp	In	RecStd	1.5 ÷ 4.0 mA	100 Ω external
	STRIn / STRIp	In	RecStd	1.5 ÷ 4.0 mA	100 Ω external
	STROn / STROp	Out	DriveHi	RecStd	100 Ω + 100 Ω external
	XCKn / XCKp	Out	DriveHi	RecStd	100 Ω + 100 Ω external
	XCKINn / XCKINp	Out	DriveHi	RecStd	100 Ω + 100 Ω external

MHz clock coming from the MCC (MCC-*CKO*). The FE encodes EoE (End-of-Event), WNG#0 (Warning) or ERR (Error, not presently used by the demonstrator chips) using invalid row numbers (from 224 up).

The WNG message is transmitted at the end of the event and replaces the EoE message, which

Table 3-10 Packet format for event read-out used in transmission from FE to MCC.

	Header (Bin)	LV1# (Hex)	Row# (Hex)	Column# (Dec)	ToT (1)
Data hit	1	0÷F	00 ÷ DF	0 ÷ 23	0 ÷ FF
EoE / WNG#n	1	0÷F	En	xx	xx
EoE	1	0÷F	Fn	xx	xx

Note:
(1) Time over Threshold (ToT) if option is selected, missing field if option is deselected
(2) x = don't care

is suppressed in such cases. Table 3-10 summarizes the format for data hits and messages transmitted from FE chips to the MCC, while Figure 3-4 shows the transmission format in the time domain.

3.3.2 MCC / FE: Protocol to Configure FE Parameters

Initialization parameters are loaded into the FE chips by the MCC with a protocol using *CCK* (5 MHz clock) to latch the input data and *LD* (load) to separate `data` from `address` and `control`. The rising edge of the clock is used to latch the incoming data. The `data` registers inside the FE are chained together to form shift registers. These structures can be loaded with new data values while the old ones are shifted out and read back into the MCC through the FE-*DO* lines.

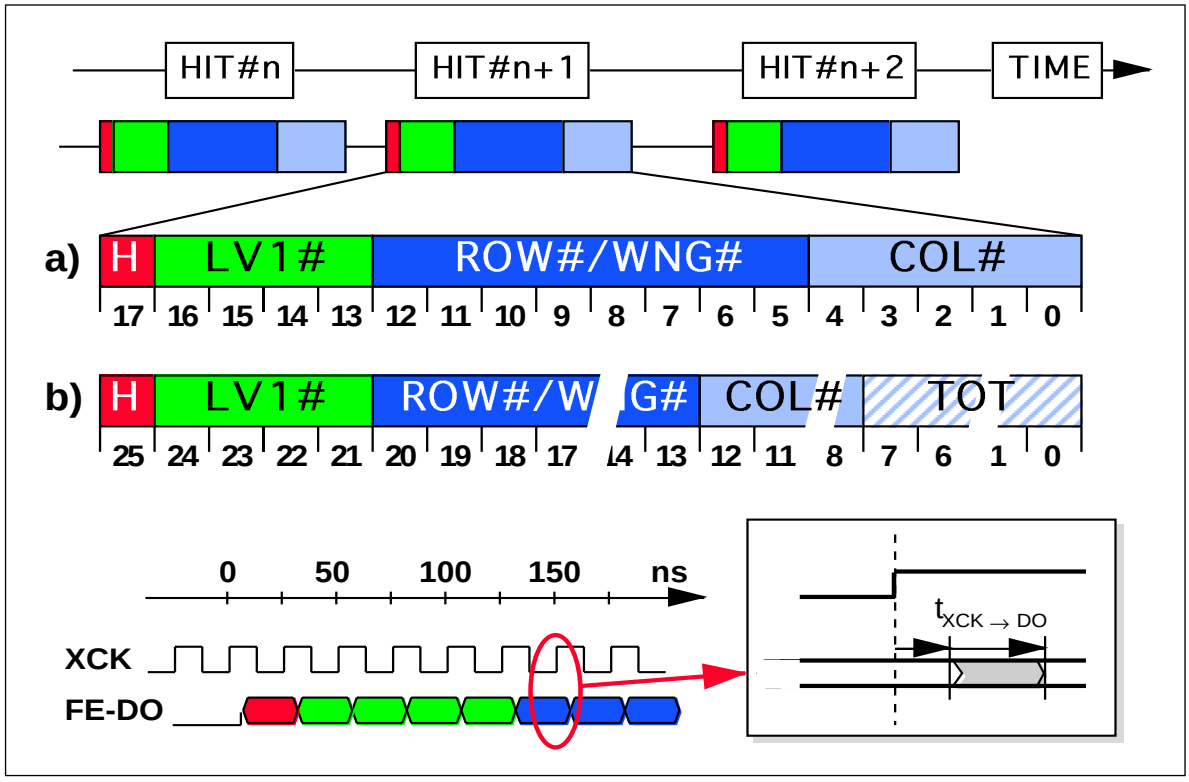

Figure 3-4 Data format and timing for event transmission from FE to MCC. Hit frame length is 18-bit long if Time over Threshold (ToT) is not provided (*a*), while it is 26-bit long if ToT is generated (*b*).

Before transmitting the data train, the `geographical address` together with the `command` are shifted in (see Figure 3-5). `Geographical address` and `command` (which could either be a real command like *SoftReset* or a way of defining a secondary address for incoming data) are transferred by the first 29-bits transmitted when *LD* is low. Then *LD* is raised and the transfer of the `data` starts. Data transmission is terminated by the falling edge of *LD* and the absence of *CCK*. *CCK* is generated by the MCC dividing its input clock (*CK*) by a factor 8 (i.e. *CCK* = 5 MHz). *CCK* is only generated when a bit carrying `address`, `command` or `data` information is present at MCC-*DAO* pin. The waveforms used by this protocol are shown in Figure 3-5.

3.3.3 Event Format at MCC Output

Figure 3-6 is a graphic example of event encoding from the MCC output.

The format of the event which comes out from the MCC has been defined considering that:

- Events are ordered by LV1 arrival time.

  Event building is done by grouping all hits belonging to the same LV1 number. The first event which has been triggered is the first to be sent out. Every event is entirely transmitted before the next event is considered for transmission: no event interleaving is allowed.

- Data are sorted and grouped by FE chips.

  The event builder sorts pixel hits by FE chip order. This permits data compression by “clustering” for either the case of a non-uniform hit distribution (jets) or a large pixel occupancy (B-layer) since every hit does not need to extend the address information to include the FE chip number with its row and column address.

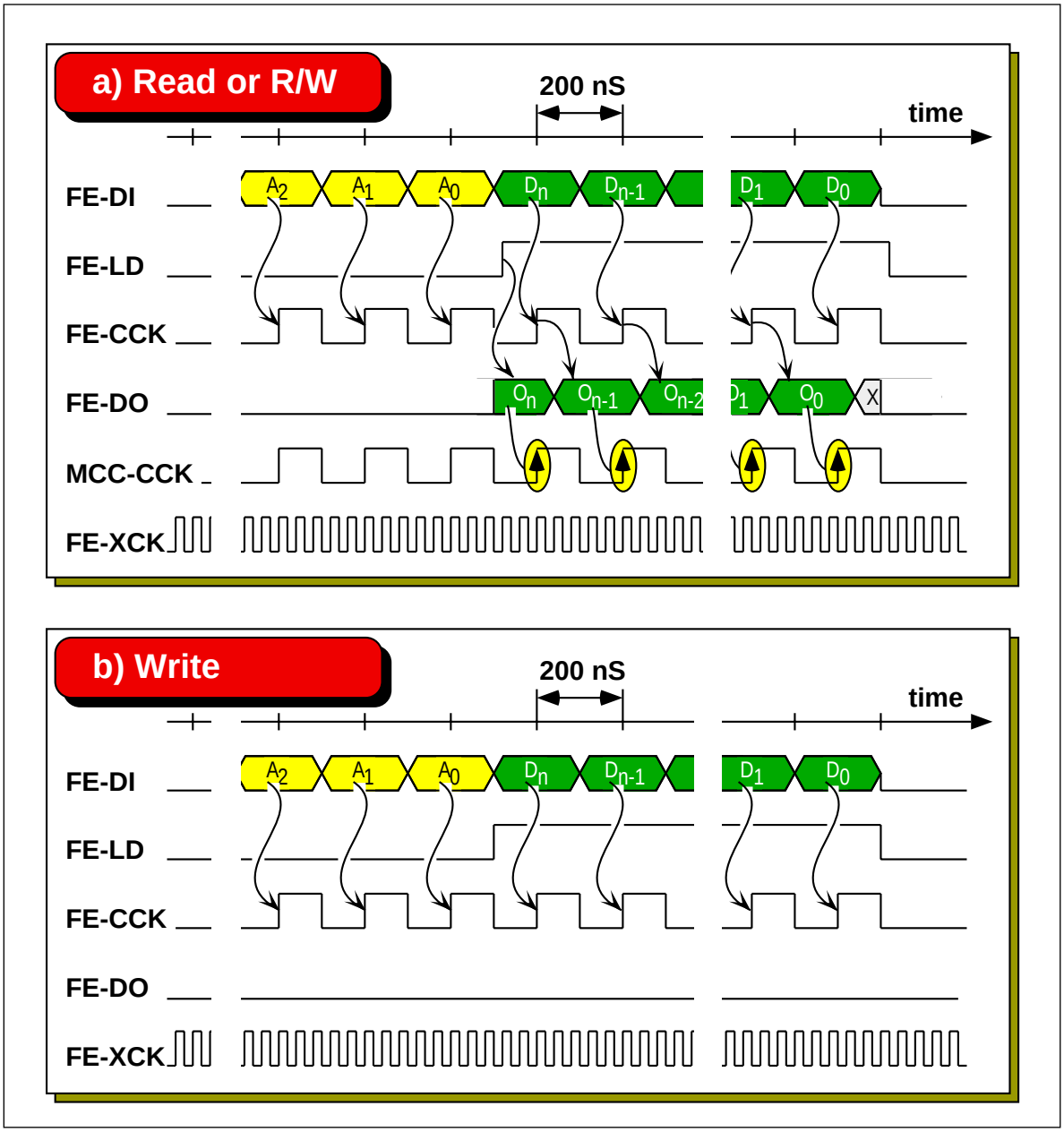

Figure 3-5 Protocol between MCC and FE chips to load initial configuration. Read/Write operations can be executed at the same time if new values are loaded into the FE while present values are readout (a). On low-to-high transition of the FE-LD signal only FE chips which recognize the transmitted address operate on data.

- Event length is not known until the whole event is transmitted.

  The event builder does not know until the event is completely built up what the total number of hits is, or what the number of hits per FE chip is. This makes the event builder simpler, but forbids the usage of forward word counters in the event frame. Event data fields must be recognized by their content.

- Data must be compressed due to bandwidth limit.

  Since we need to keep the number of transmission cables as small as possible, and since a rather large amount of data is transmitted from the MCC and even more from the LCC, event coding which compresses data is specially important.

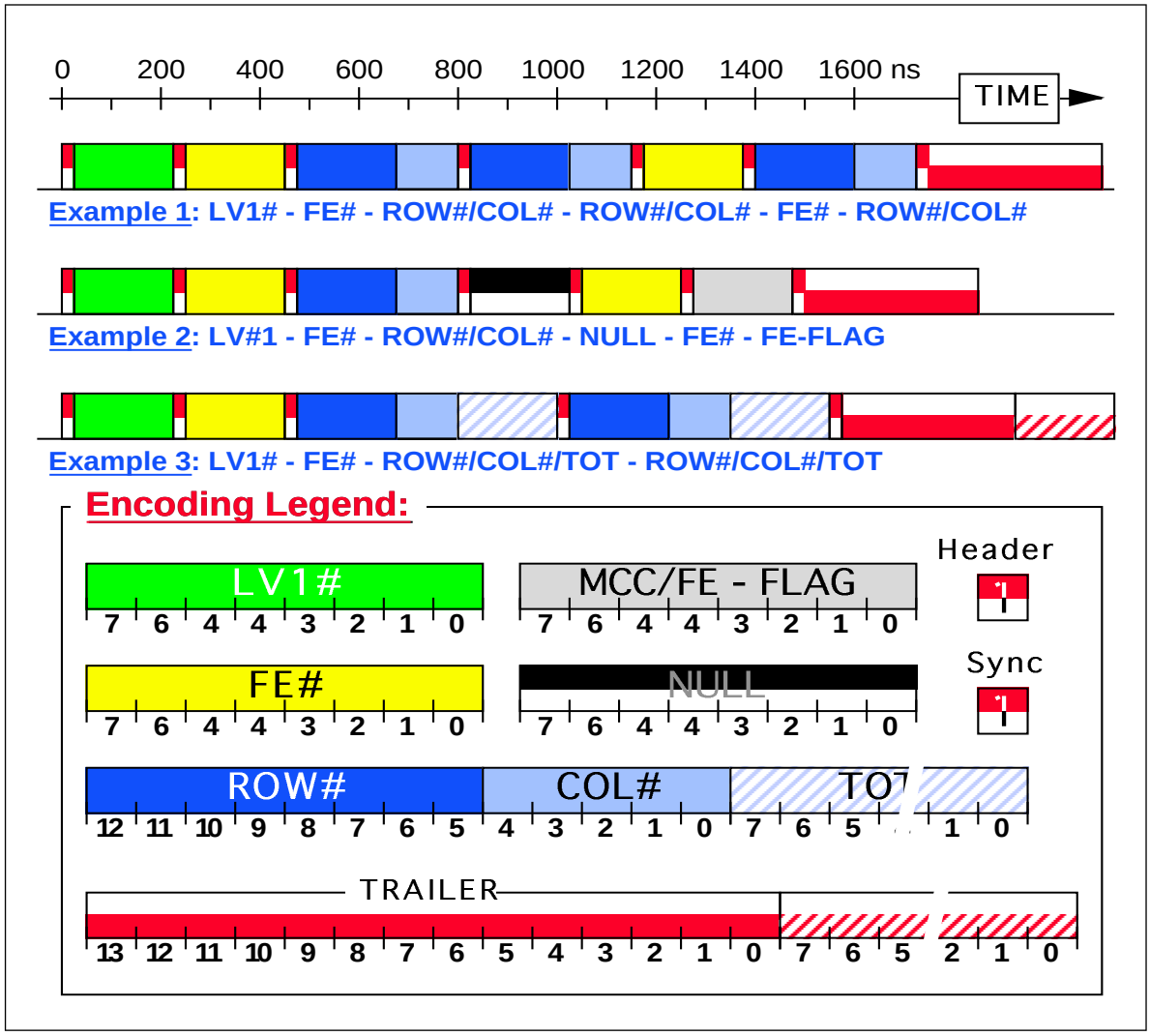

Figure 3-6 Event Data format at the MCC output. The example 3 illustrates the case of Time over Threshold encoding.

- Recovery from transmission errors.

  Any transmission error must be recovered by the next event data. The chosen mechanism is to use a Trailer whose value can never occur in the data. Since data fields are allowed to take any value we decided to add a Sync bit (bit set to '1') after every data field and a Trailer containing a fixed number of zeroes.

- Null data (wait).

  In the coding we have included the possibility of informing the receiver that the event is not finished yet in case the MCC cannot supply new data. This is optional for the MCC, but the design of the receiver must include this option for generality.

An event frame at the output of the MCC is an ordered stream of data fields separated by synchronization bits. The syntax for the event data structure is as follow:

<Event>

::= <Header> LV1# <MccFlag>? <FrontEnd>* <Trailer>

<Header>

::= 1

<Trailer>

::= <Sync> 00 0000 0000 0000

::= <Sync> 00 0000 0000 0000 0000 0000 (if opt. ToT selected)

<FrontEnd>

::= <Sync> MCC-FE# <Hit>* <FeFlag>?

<Hit>

::= <Sync> ROW# COL#

::= <Sync> ROW# COL# TOT (if opt. ToT selcted)

<MccFlag>

::= <Sync> MCC-FLAG

<FeFlag>

::= <Sync> FE-FLAG

<Sync>

::= 1

||= 1 NULL <1 NULL>*

The values of the keywords which appear in the event syntax are in Table 3-11. We have left some Reserved encoding space for possible extensions needed by the LCC.

Table 3-11 Value for the keywords in the event syntax.

Keyword	Low Value	High Value	Description
COL#	0 0000	1 0111	Column number
FE-FLAG	1111 0000	1111 1110	Error / Warning or Message associated to FE no. 0 to 15
LV1#	0000 0000	1111 1111	Level 1 Trigger number from 0 to 255
MCC-FE#	1110 0000	1110 1111	FE number in the module from 0 to 15
MCC-FLAG	0000 0000	1101 1111	Error / Warning or Message from the MCC
NULL	1111 1111		Null data
ROW#	0000 0000	1101 1111	Row number from 0 to 223
TOT	0000 0000	1111 1111	Time over Threshold (if opt. ToT is selected)

Data at the MCC-*DTO* pin are in phase with MCC-*CK* and are stable at the clock rising edge.

3.3.4 Configuration Data Format at MCC Output

In response to a serial command (such as Slow:Read) the MCC or the selected FE sends back a stream of bits (usually the contents of some register). To such bits the MCC adds a Header (a bit

‘1’) immediately before the first data bit. It is the receiver which issued the read command that knows the length of the data stream following the Header.

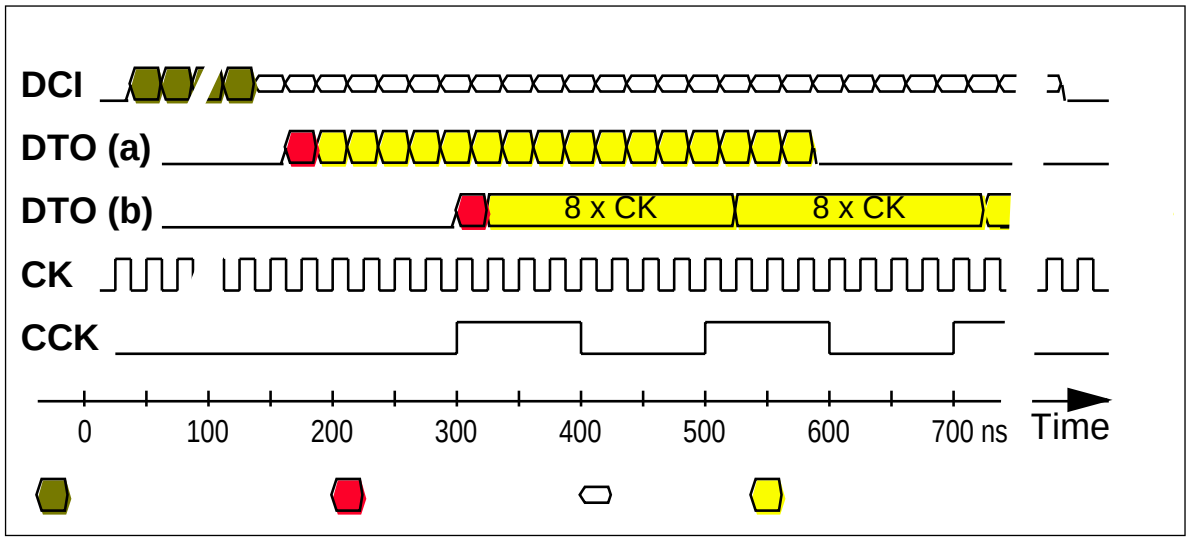

Figure 3-7 Data format and protocol at the MCC-DTO output pin. The waveform (a) is for data produced by the MCC, while (b) is for data originating from the FE chip.

If the data, read back from the MCC-*DTO* pin, originate from an FE chip, every bit after the Header is transmitted at a rate of 5 MHz, i.e. output values are kept constant for 8 *CK* clock cycles (see Figure 3-7)

3.3.5 Control/Data Format and Protocol at MCC Input

The serial command protocol defined for the MCC is largely copied from the ABC[1] chip command set. An explanation of the command set is given in Section 2.3.2, "Command Set".

The 3 groups of serial commands:

1. Trigger command is a 3-bit long word so that a new LV1 can be issued every 75 ns (3 CK clock cycles, i.e. 3 bunch crossing) as required by ATLAS.
2. Fast commands are 7-bit long words, and they can be issued in the absence of LV1 commands without blocking the data acquisition.
3. Slow commands are long from eleven to thousands of bits and they block event data acquisition (if running) when issued.

Trigger and Fast commands can be interleaved, while Slow commands stop the event readout (RUN bit in the CSR register is reset). Command encoding is given in Table 3-12 and Table 3-13. The most significant bit of all the commands is ‘1’ which can be interpreted as a Header bit.

The Control / Data protocol uses MCC-*DCI* as a data line and MCC-*CK* as a data validation line. Data must be stable on the rising edge of MCC-*CK*. Data are allowed to change at each clock cycle unless they transport information going to the FE chips. In this last case each information bit must be kept stable for 8 clock cycles.

1. Atlas Binary Chip.

Table 3-12 Control Commands

Type	Name	Field 1	Field 2	Field 3	Description
Trigger	LV1	110			Level 1 Trigger
Fast	SyncFE	101	0100		Re-synchronize FE chips and MCC
Fast	SyncMCC	101	0010		Re-synchronize MCC LV1 counter
Fast	PushEvent	101	0101		Push next event out
Slow	Command	101	0111	Command	Slow command (see Table 3-13)

Note:
The MSB of Field 1 is always '1' being the header which synchronize the serial data stream

Table 3-13 Slow Commands

Name	Field 3	Field 4 [1]	Field 5	Bits [2]	Description
WrRegister	0000	Address	Data	16	Write to MCC register
RdRegister	0001	Address	----	16	Read from MCC register
WrFifo	0010	Address	Data	25	Write a word to the enabled FIFO(s)
RdFifo	0011	Address	----	25	Read a word from the addressed FIFO
WrFrontEnd	0100	----	Data	Var [3]	Write to the FE chips.
RdFrontEnd	0101	----	----	Var [3]	Read (or R/W) from the Enabled FE chips
WrReceiver	0110	----	Data	Var [4]	Write to Enabled Receiver Channels
	0111	----			Not defined
EnDataTake	1000	----		0	Enable Data Taking
SoftReset	1001	----		0	Software Reset

Note:
1) Field 4 is always 4 bits long.
2) Length of Field 5 is variable and it is specified in this column
3) Var = <CCNT> × 8 + <DCNT> × 8 CK units = <CCNT> + <DCNT> bits
4) Var = <CCNT> × 8 + <DCNT> × 8 CK units = <CCNT> × 8 + <DCNT> × 8 bits

3.4 Signal Switching and Timing Relationship

3.4.1 FE Signal Switching

This section contains only the most significant signal relationships and switching times. For a more complete description the reader is referred to "Pixel Front-End Chip Specification, K. Einsweiler and P. Fischer".

3.4.1.1 Signals in Data Taking Operation.

Signals in this section refer to the signals at the FE chip inputs or outputs and which are active during data taking. Signal timing characteristics are in Table 3-14 with the parameter definitions being in Figure 3-8.

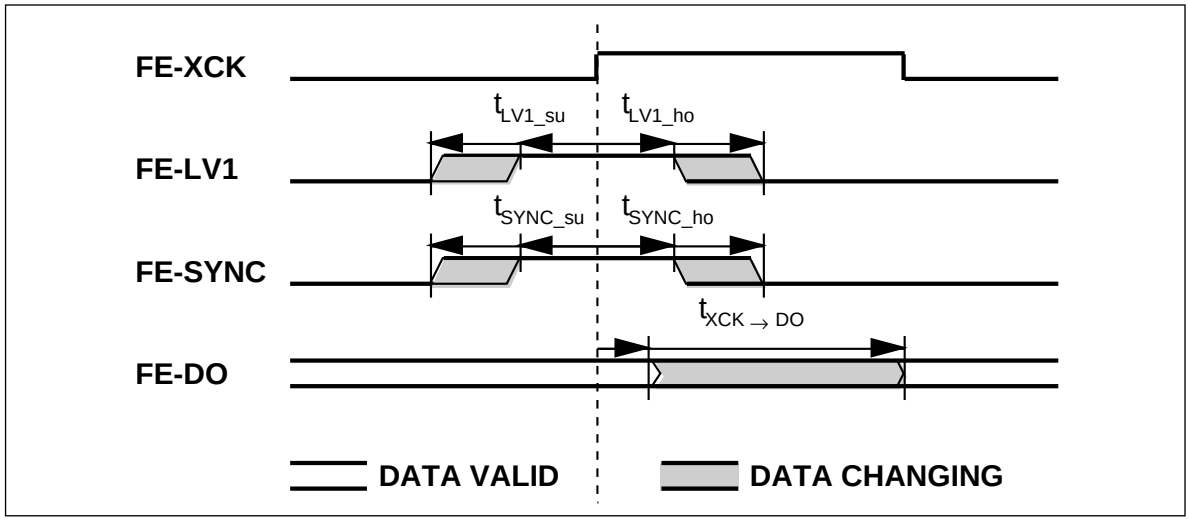

Figure 3-8 FE signal switching: parameter definition for FE signals in “Data Taking” operation.

Table 3-14 FE signal switching: Switching characteristics for FE signals in “Data Taking” operation.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{XCK}	XCK frequency			40		MHz
t_{LV1_su}	LV1 setup time		5			ns
t_{LV1_ho}	LV1 hold time		2			ns
t_{SYNC_su}	SYNC setup time		5			ns
t_{SYNC_ho}	SYNC hold time		2			ns
$t_{XCK \rightarrow DO}$	From XCK rising to Data Output	C_L = 10 pF	2		20	ns

3.4.1.2 Signals in Initialization and Control Operation

Signals in this section refer to the signals at the FE chip inputs or outputs and which are active during system initialization. Signal timing characteristics are in Table 3-15 with the parameter definitions being in Figure 3-9.

3.4.2 MCC Signal Switching

This section contains only the most significant signal relationships and switching times. For a more complete description the reader is referred to “Module Controller Chip Specification, G. Darbo and G. Meddeler”.

3.4.2.1 MCC to FE Timing: Data Taking Operation.

Signals in this section refer to the signals at the MCC chip inputs or outputs and which are active during data taking. Signal timing characteristics are in Table 3-16 with the parameter definitions being in Figure 3-10

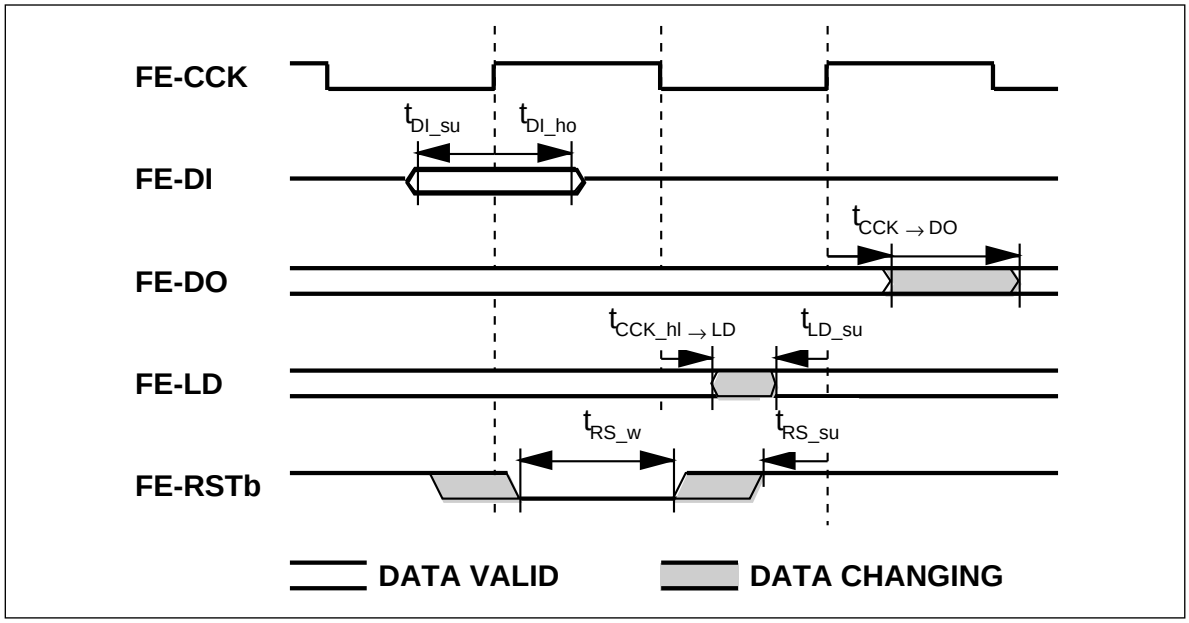

Figure 3-9 FE signal switching: parameter definitions for FE signals in “System Configuration” operation

Table 3-15 FE signal switching: Switching characteristics for FE signals in “System Configuration” operation

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{CCK}	CCK frequency			5		MHz
t_{DI_su}	Data Input setup time		5			ns
t_{DI_ho}	Data Input hold time		2			ns
$t_{CCK \rightarrow DO}$	From CCK rising to Data Output	C_L = 10 pF	2		20	ns
$t_{CCK\_hl \rightarrow LD}$	From CCK falling to LD changing		2			ns
t_{LD_su}	LOAD setup time		20			ns
t_{RS_w}	Reset pulse width		50			ns
t_{RS_su}	Reset setup time		5			ns

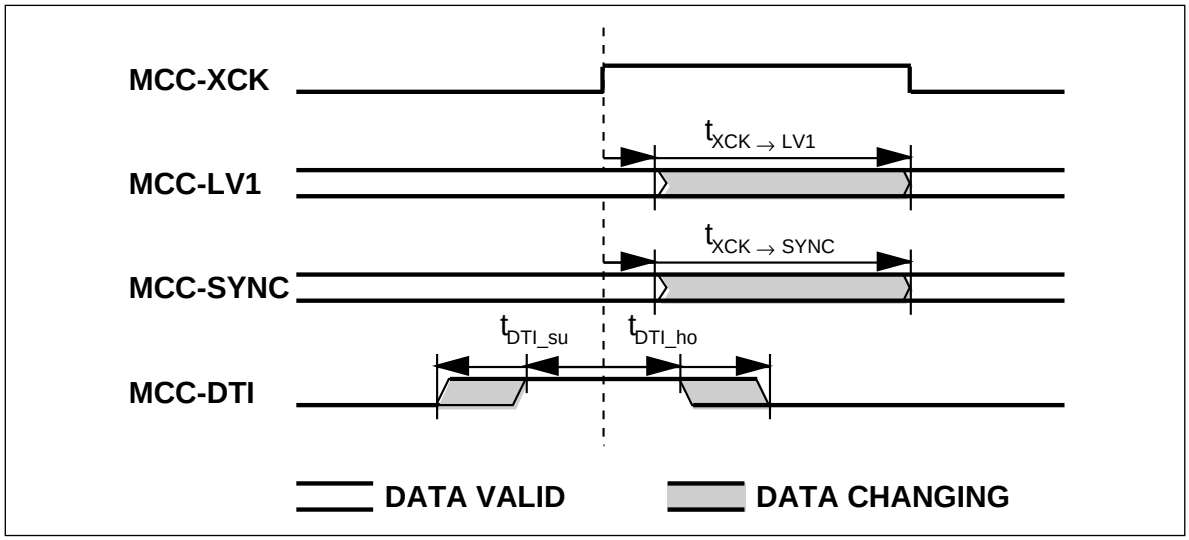

Figure 3-10 MCC signal switching: parameter definitions for MCC to FE signals in “Data Taking” operation

Table 3-16 MCC signal switching: Switching characteristics for MCC signals in “Data Taking” operation.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{XCK}	XCK frequency			40		MHz
$t_{XCK\rightarrow LV1}$	From XCK rising to LV1 changing		2		20	ns
$t_{XCK\rightarrow SYNC}$	From XCK rising to SYNC changing		2		20	ns
t_{DTI_su}	DTI setup time		5			ns
t_{DTI_ho}	DTI hold time		2			ns

3.4.2.2 MCC to FE Timing: Control Signals

Signals in this section refer to the signals at the MCC chip inputs or outputs and which are active during system configuration. Signal timing characteristics are in Table 3-17 with the parameter definitions being in Figure 3-11

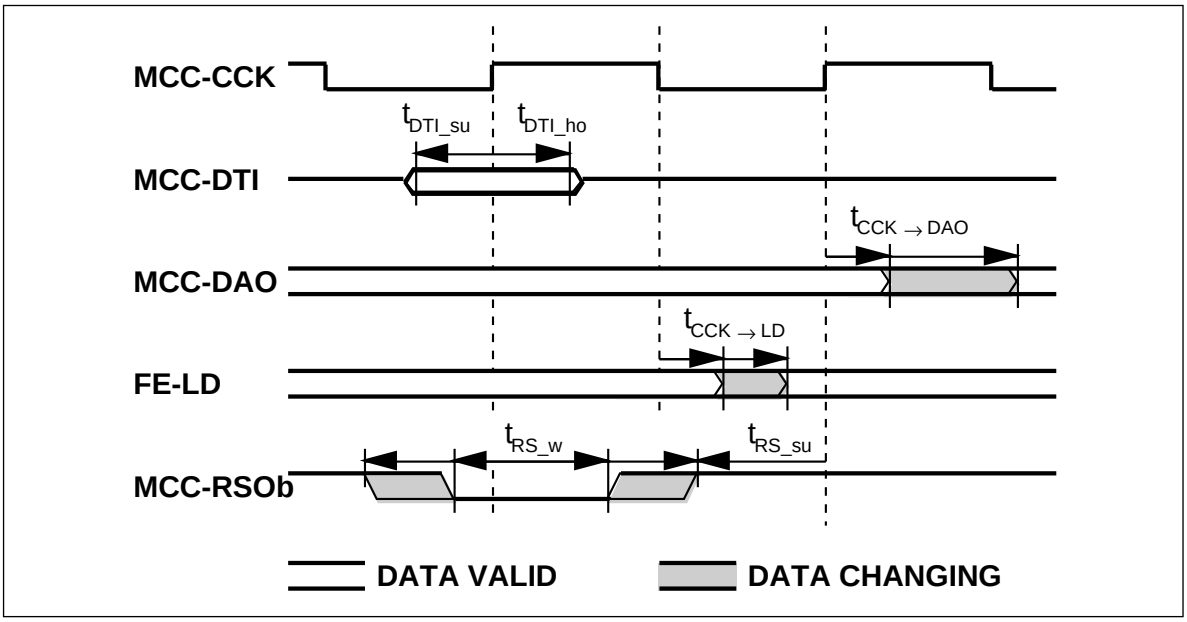

Figure 3-11 MCC signal switching: parameter definitions for MCC to FE signals in “System Configuration” operation

Table 3-17 MCC signal switching: Switching characteristics for MCC signals in “System Configuration” operation.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{CCK}	CCK frequency			5		MHz
t_{DTI_su}	Data Input setup time		5			ns
t_{DI_ho}	Data Input hold time		2			ns
$t_{CCK\rightarrow DAO}$	From CCK rising to Dt/Add Output	C_L = 40 pF	2		30	ns
$t_{CCK\rightarrow LD}$	From CCK falling to LOAD Output	C_L = 40 pF	2		30	ns
t_{RS_w}	Reset pulse width	C_L = 40 pF	50			ns
t_{RS_su}	Reset setup time	C_L = 40 pF	20			ns

3.4.2.3 MCC to LCC Timing: Control and Command-Data Signals

Signals in this section refer to the signals at the MCC chip inputs or outputs and which are active during system configuration. Signal timing characteristics are in Table 3-17 with the parameter definitions being in Figure 3-11

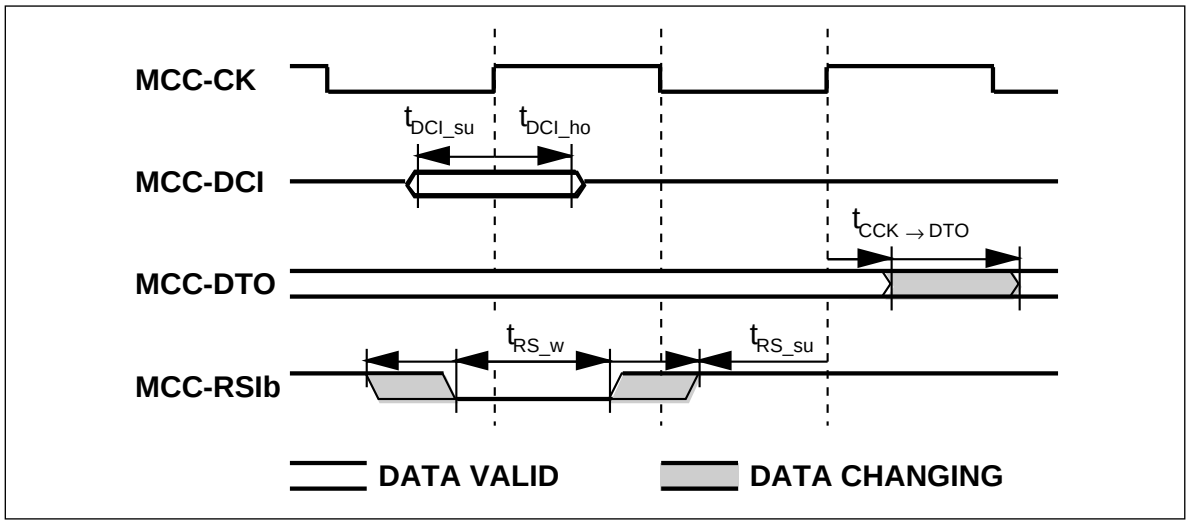

Figure 3-12 MCC signal switching: parameter definitions for MCC to LCC signals.

Table 3-18 MCC signal switching: Switching characteristics for MCC to LCC signals.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
f_{CK}	XCK frequency			40		MHz
t_{DCI_su}	Data Command setup time		5			ns
t_{DCI_ho}	Data Command hold time		2			ns
$t_{CK\rightarrow DTO}$	From CK rising to Data Output	C_L = 50 pF	2		20	ns
t_{RS_w}	Reset pulse width		50			ns
t_{RS_su}	Reset setup time		0			ns