

## **Project Specification**

**Project Name:** ATLAS Binary Chip (ABC)

**Version:** V4.04

Includes information about the ABCD chip.

## Table of Contents

<b>1. SCOPE.....</b>	<b>1</b>
1.1. ATLAS BINARY FRONT END READOUT ARCHITECTURE .....	1
<b>2. RELATED PROJECTS AND DOCUMENTS .....</b>	<b>2</b>
<b>3. TECHNICAL ASPECTS.....</b>	<b>3</b>
3.1. REQUIREMENTS .....	3
3.2. SPECIFICATION OF DELIVERABLES .....	5
3.2.1. <i>Input Level Translators</i> .....	5
3.2.2. <i>Input Register and Mask Register</i> .....	7
3.2.2.1 Input Register.....	7
3.2.2.2 Edge Detection Circuitry.....	7
3.2.2.3 Channel Masking Register .....	7
3.2.3. <i>Pipeline</i> .....	8
3.2.3.1 Accumulator Register.....	8
3.2.4. <i>Readout Buffer</i> .....	8
3.2.5. <i>Data compression logic</i> .....	10
3.2.6. <i>Readout Circuitry</i> .....	12
3.2.7. <i>Readout Controller Block</i> .....	13
3.2.7.1 L1 Counter.....	13
3.2.7.2 Bunch Crossing Counter .....	13
3.2.7.3 Event FIFO.....	13
3.2.7.4 Token Generation Logic.....	13
3.2.7.5 Data Formatting Logic .....	14
3.2.7.6 Serial Data Output Driver.....	14
3.2.8. <i>Command Decoding</i> .....	14
Level 1 Trigger Command.....	14
Control Commands .....	15
Slow Control Command.....	15
3.2.9. <i>Configuration Register</i> .....	16
3.2.10. <i>Calibration Logic</i> .....	17
3.2.11. <i>Strobe Delay Register</i> .....	17
3.2.12. <i>DAC Register</i> .....	18
3.2.13 <i>DACs</i> .....	18
3.2.13a <i>Bias DACs</i> .....	19
3.2.14. <i>Clock and Command Inputs</i> .....	19
3.2.15. <i>Chip ID</i> . ....	21
3.2.16. <i>Token Input/Output Circuit</i> .....	21
3.2.17. <i>Test Circuitry</i> .....	22
3.2.18. <i>Readout Protocols</i> .....	23
3.2.19. <i>Control Protocol</i> .....	25
3.2.19.1 Level 1 trigger Command: .....	26
3.2.19.2 Fast Control Command: .....	26
3.2.19.3 Control commands .....	26
3.2.20. <i>Chip Initialisation and Configuration</i> .....	28
3.2.20.1 Send_ID Mode .....	28
3.2.20.2 Data_Taking Mode.....	28
3.2.20.3 Clock Feed Through.....	28
3.2.21. <i>Resets</i> .....	28
3.2.21.1 Power up reset .....	28
3.2.21.2 Soft Reset .....	28
3.2.21.3BC Reset.....	28
3.2.22. <i>Data Readout and Redundancy</i> .....	30
3.2.23. <i>Default Register Values</i> .....	32
3.2.23.1 Master/Slave Selection.....	32
3.2.24. Input/Output Connections .....	33
3.2.25. <i>Electrical Specifications</i> .....	33
3.2.25.1 Supply Voltage.....	33
3.2.25.2 Power Consumption.....	33
3.2.25.3 Power Supply Connections.....	34
3.2.25.3 Input /Output Levels.....	34
3.2.27. <i>Physical Requirements</i> .....	37

## List of Tables and Figures

### FIGURE 1.1

BLOCK DIAGRAM OF THE BINARY READOUT SYSTEM .....	2
--	---

### FIGURE 3.1

BLOCK DIAGRAM OF THE BINARY READOUT CHIP .....	4
--	---

### FIGURE 3.2

INPUT LEVEL TRANSLATORS INPUTS/OUTPUTS .....	5
--	---

### TABLE 3.1

INPUT LEVEL TRANSLATOR I/O SIGNAL DEFINITIONS .....	6
---	---

### TABLE 3.2

TEST MODES .....	6
------------------	---

### FIGURE 3.3

INPUT REGISTER INPUTS/OUTPUTS .....	7
-------------------------------------	---

### TABLE 3.3

INPUT REGISTER I/O SIGNAL DEFINITIONS .....	7
---	---

### TABLE 3.4

MASKING REGISTER MODES OF OPERATION .....	7
---	---

### FIGURE 3.4

PIPELINE INPUT/OUTPUTS .....	8
------------------------------	---

### TABLE 3.5

PIPELINE INPUT/OUTPUT SIGNAL DEFINITIONS .....	8
--	---

### FIGURE 3.5

BLOCK DIAGRAM OF THE READOUT BUFFER.....	9
--	---

### FIGURE 3.6

READOUT BUFFER INPUT/OUTPUTS .....	10
------------------------------------	----

### TABLE 3.6

READOUT BUFFER INPUT/OUTPUT SIGNAL DEFINITIONS .....	10
--	----

### TABLE 3.7A

DATA COMPRESSION CRITERIA .....	10
---------------------------------	----

### TABLE 3.7B

DATA COMPRESSION LOGIC OUTPUT STATES .....	11
--	----

### FIGURE 3.7

DATA COMPRESSION LOGIC INPUT/OUTPUTS .....	11
--	----

### FIGURE 3.8

CONNECTIONS TO READOUT CIRCUITRY .....	12
--	----

### TABLE 3.8

READOUT LOGIC INPUT/OUTPUT SIGNAL DEFINITIONS .....	13
---	----

### FIGURE 3.9

Connections to Readout Controller Circuitry .....	14
---	----

### TABLE 3.9

Readout Controller Input/Output Signal Definitions .....	14
<b>FIGURE 3.10</b>	
COMMAND DECODER INPUTS/OUTPUTS	
<b>TABLE 3.10</b>	
COMMAND DECODER INPUT/OUTPUT SIGNAL DEFINITIONS .....	15
<b>FIGURE 3.11</b>	
CONFIGURATION REGISTER INPUTS/OUTPUTS .....	16
<b>TABLE 3.11A</b>	
CONFIGURATION REGISTER INPUT/OUTPUT SIGNAL DEFINITIONS.....	16
<b>TABLE 3.11B</b>	
CONFIGURATION REGISTER CONTENTS .....	16
<b>TABLE 3.12</b>	
CALIBRATION CODES .....	17
<b>FIGURE 3.12</b>	
STROBE DELAY REGISTER INPUTS/OUTPUTS .....	17
<b>TABLE 3.13A</b>	
STROBE DELAY REGISTER INPUT/OUTPUT SIGNAL DEFINITIONS .....	17
<b>FIGURE 3.13</b>	
DAC REGISTER INPUTS/OUTPUTS .....	18
<b>TABLE 3.14</b>	
DAC REGISTER INPUT/OUTPUT SIGNAL DEFINITIONS .....	18
<b>FIGURE 3.14</b>	
I/O CONNECTIONS TO DACS .....	18
<b>TABLE 3.15A</b>	
CLOCK INPUT/OUTPUT SIGNAL DEFINITIONS .....	20
<b>TABLE 3.15B</b>	
CLOCK INPUT MODES OF OPERATION .....	21
<b>TABLE 3.16A</b>	
GEOGRAPHICAL ADDRESSES (ID(5-0)).....	21
<b>TABLE 3.16B</b>	
ABC GEOGRAPHICAL ADDRESSES (ID3-0).....	21
<b>FIGURE 3.16</b>	
TOKEN AND DATA INPUTS CIRCUIT .....	22
<b>TABLE 3.17A</b>	
TOKEN AND DATA INPUT SIGNAL DEFINITIONS .....	22
<b>FIGURE 3.17</b>	
TOKEN AND DATA OUTPUTS CIRCUIT .....	22
<b>TABLE 3.17B</b>	
TOKEN AND DATA OUTPUT SIGNAL DEFINITIONS .....	22
<b>FIGURE 3.18A</b>	
MODULE DATA FORMAT .....	23

**Figure 3.18b**

ISOLATED HIT DATA FORMAT .....	24
--------------------------------	----

**FIGURE 3.18C**

ADJACENT HIT DATA FORMAT .....	24
--------------------------------	----

**FIGURE 18D**

NULL DATA PACKET .....	24
------------------------	----

**FIGURE 18E**

CONFIGURATION DATA PACKET .....	25
---------------------------------	----

**FIGURE 3.18F**

ERROR DATA FORMAT .....	25
-------------------------	----

**TABLE 3.18**

COMMANDS .....	25
----------------	----

## **1. Scope**

The aim of this project is to develop a chip suitable for the binary readout of the proposed LHC ATLAS Semi-Conductor Tracker (SCT). This chip is to be based on the work already carried out for ATLAS, in particular the DDR2 chip and CDP128 chip. The chip will be designed to interface to the CAFE-M<sup>1</sup> and the DORIC<sup>2</sup> chips. In addition it must also be compatible with the protocols for receiving and sending data as defined for the ATLAS silicon micro-strip detector.<sup>3</sup>

<sup>1</sup> CAFE: A Complementary Bipolar Analog Front End Integrated Circuit for the ATLAS SCT. Issy Kipsis

<sup>2</sup> DORIC. A Front End Clock and L1 Distribution Chip. J.R. Gorbald & P.Seller

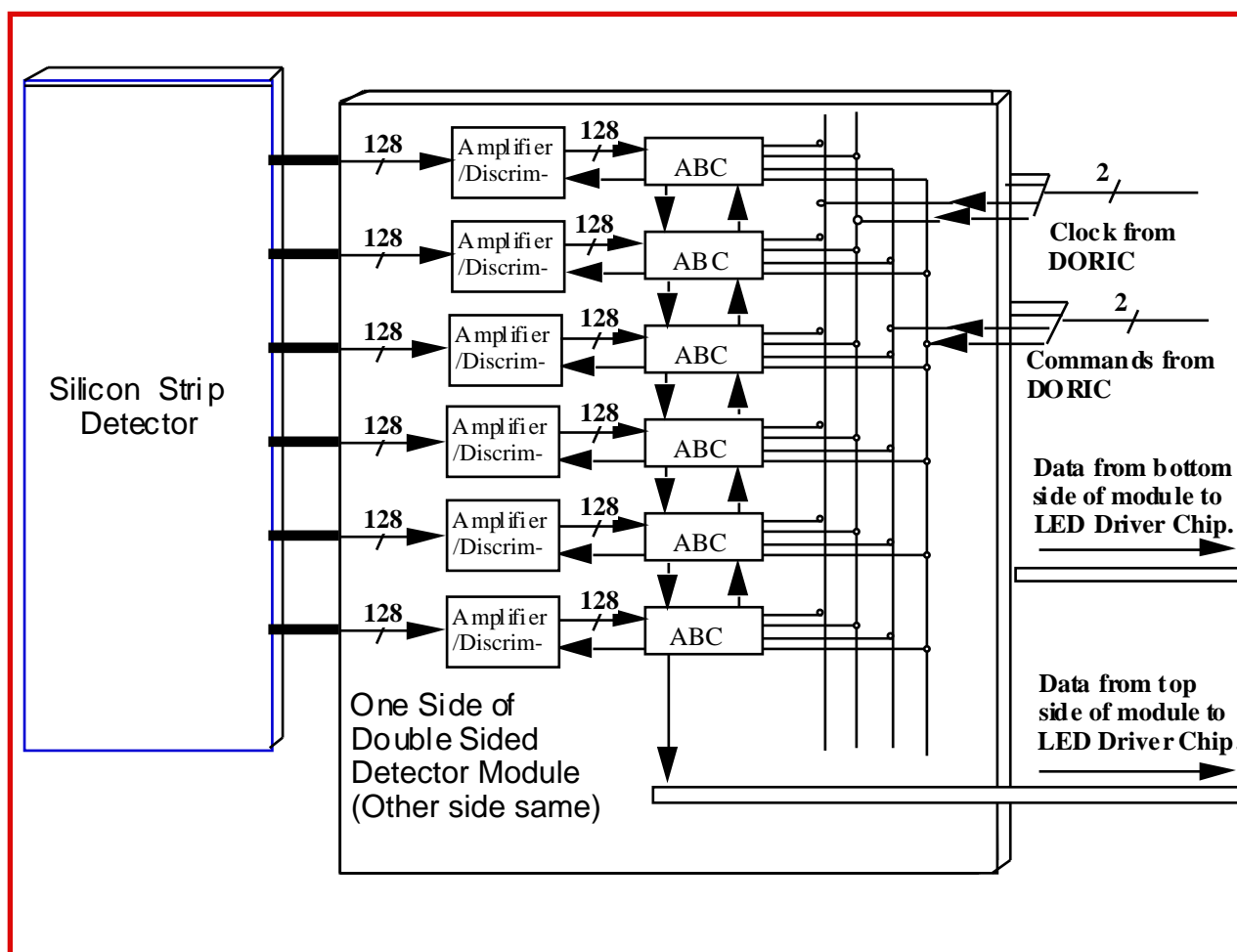
<sup>3</sup> Proposed Protocols for Data Transmission and Control Functions for the ATLAS Silicon Micro-strip Detector

### **1.1. ATLAS Binary Front End Readout Architecture**

The ATLAS binary front-end readout architecture is based on amplifier/discriminator and CMOS pipeline chips. In one configuration this is arranged as two chips, a bipolar amplifier/discriminator chip ( CAFE-M) and a CMOS pipeline chip ("ABC"). The "ABC" chip reads out 128 channels of silicon strip data from a CAFE-M chip. Data are stored for the duration of the level 1 trigger (L1) latency in a 128 deep 1 bit pipeline. In normal data taking, the data corresponding to hit strips are read out ("sparse" read out). Four modes of data read out are possible: in the level mode the decision is based on the data in the BC corresponding to the L1 trigger only but in the edge mode there must be a "0" in the channel for the bunch crossing preceding the trigger. The efficiency is highest for level mode but the occupancy may be up to a factor of two higher. The optimum mode to run will, therefore, depend on luminosity and therefore both modes of operation should be possible. A third mode named hit demands a "1" in any of three bunch crossings, that of the given L1 trigger, the one preceding and the one following. This mode is used for beam tests and diagnostics. The final mode "read\_all" places no requirements on the data. All channels are read out. This mode is used during chip testing.

In order to reduce dead time due to queuing losses all the data from an L1 trigger are transferred to an 8 deep de-randomising buffer. The dead time of the system should be less than 1% for a mean occupancy of 1% and a mean L1 rate of 100 KHz. The data from the 6 ABCs corresponding to one side of an SCT module are readout in a daisy chain via a token passing scheme. The bunch crossing clock, the L1 trigger and the other control information are sent to the chips from the DORIC chip.

In order for the system to run reliably for many years at the LHC it must be immune to single point failures. If a single ABC on a module fails, then the module can be re-configured so that bypass lines are used to route data and tokens around the bad chip. If a fibre optic data transmission link fails, then the data can be re-routed via the fibre optic link on the other side of the hybrid. If the clock and control link to a module fails, then the clock and control can be taken from a neighbouring module. Soft errors caused by a loss of a bit in data transmission or clock & control are detected in many cases by consistency checks on the data. This is informational since the system will be reset frequently to clear such errors, rather than corrected on error detection. Errors caused by buffer overflows are detected and lost events are labelled as such in the output data stream. Although an excess data rate can cause loss of data, it should never result in the data being read out for the wrong trigger.



**Figure 1.1: Block Diagram of the Binary Readout System**

## **2. RELATED PROJECTS AND DOCUMENTS**

- 1). DORIC. A Front End Clock and L1 Distribution Chip.  
J.R. Gorbald and P.Seller.
- 2). Digital Read-out Chip for Silicon Strip Detectors at SDC  
Kanex Shankar Nikhil Kundu et al
- 3) A 128-Channel Digital Pipeline Chip for Silicon Strip Detector Read-Out  
Joel DeWitt
- 4). CAFE: A Complementary Bipolar Analog Front End Integrated Circuit for the ATLAS SCT.  
Issy Kipsis
- 5). DDR2RH Operating Manual
- 6). A Binary Readout System for Silicon Strip Detectors at the LHC  
A.Cioccio, T.Collins
- 7).AROW Specification for Input Decoding v.11  
J. Gorbald
- 8).Requirements for Wilkinson ADC Based Digitisation and Sparsification Circuits for ATLAS  
Eric Evans
- 9).Proposed Protocols for Data Transmission and Control Functions for the ATLAS Silicon Micro-strip Detector  
A.Grillo
- 10) Dead time calculations for SCT readout architectures. A.R. Weidberg ATLAS-INDET-124.

### **3. Technical ASPECTS**

#### **3.1. Requirements**

- 1) The chip will be designed to accept the 128 output signals from the CAFE-M amplifier and comparator chip
- 2) At the start of each clock cycle the chip must sample the outputs from the CAFE-M and store these values in a pipeline until a decision can be made whether to keep the data.
- 3) Upon receipt of a Level 1 Trigger signal the corresponding set of values together with its neighbours are to be copied into another buffer, the readout buffer.
- 4) The data written into the readout buffer is to be compressed before being transmitted off the chip.
- 5) Transmission of data from the chip will be by means of token passing and must be compatible with the ATLAS protocol.
- 6) The chip will be responsible for supplying the CAFE-M chips with their calibration pulses.
- 7) The chip is required to provide reporting of some of the errors that occur
  - a) Attempt to readout data from the chip when no data is available.
  - b) Readout Buffer Overflow: The readout buffer is full and data from the oldest event/s has been overwritten.
  - c) Readout Buffer Error: The readout buffer is no longer able to keep track of the data held in it.  
(Chip reset required)
  - d) Configuration error (ChipID sent).
- 8) The chip will incorporate such features that will enable it to be tested both at the wafer level and in situ. Tests include but are not restricted to:
  - a) The functionality of input level translators
  - b) Transmission of programmable pattern through the pipeline and readout circuitry.
- 9) The chip shall incorporate features that will enable the system to continue operating in the event of a single chip failure.
- 10) It is a system requirement that the fraction of data which is lost due to the readout buffer on the chip being full is less than 1%. This assumes that on average only 1% of the silicon strip detectors are hit during any particular bunch crossing.
- 11) DACs will be included on the chip to enable the thresholds of the comparators and the calibration amplitude of the CAFE-M chip to be set.



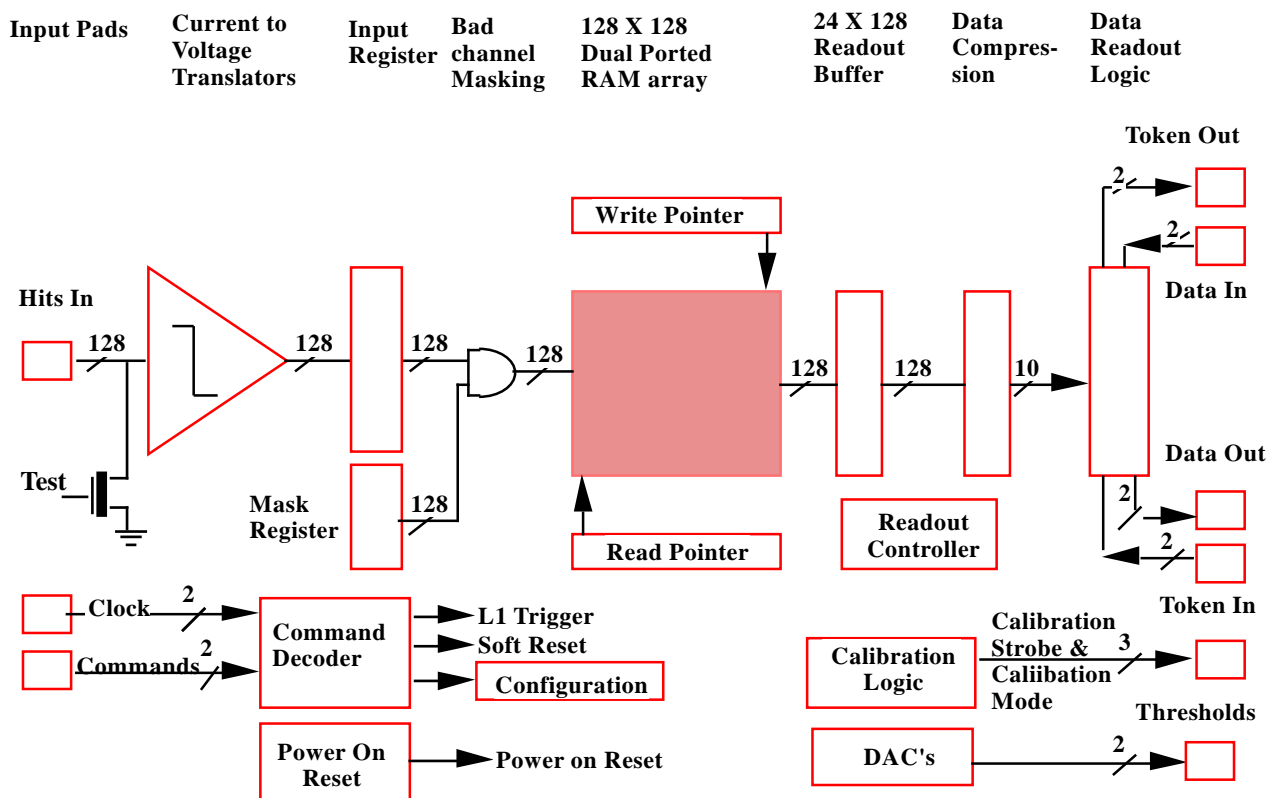


Figure 3.1: Block Diagram of the Binary Readout Chip;

## 3.2. Specification of deliverables

### 3.2.1. Input Level Translators

The outputs from the CAFE-M chip, which are fed to the inputs of the Binary Readout Chip are open collector to enable a low voltage swing current mode interface to be formed between the two chips. The input level translators on the ABC must provide a source of current which can be sunk by the outputs of the CAFE-M chip. The line labelled "return" provides a return path for the current sunk by the CAFE-M outputs to be returned to the ABC to minimise any noise coupling between the two chips. The inputs must also be able to detect the amount of current sunk by the individual outputs of the CAFE-M chip and translate it into the logic levels used inside the Binary Readout Chip. The input impedance and circuit response time must ensure transition times  $< 2\text{nS}$  on both the rising and falling edges. The effective threshold to distinguish between "hit" and "no-hit" signals from the CAFE-M is set by a high and low reference current provided by the CAFE-M chip with threshold  $= (\text{inrh} + \text{inrl})/2$ . The design of these inputs is such that the chip is operational with inputs floating or shorted to ground. Built into the block will be the facility to selectively set a quarter of the input channels at a time to the level of a hit channel. This facility is required to enable the inputs to the chip to be tested without the need to probe all input pads. The signals  $\text{calmode}(1:0)$  are used as a binary address to select one of 4 groups of 32 channels to test. (See Figure 3.25 for the physical location of channel addresses.) The logic signal "testinputs" is used as a test enable. When active, a current equal to  $\text{inrh}$  is injected into the selected channels. The signal test inputs can be activated as a level with a bit in the configuration register (see Table 3.11b) or can be pulsed for one clock cycle by a control command (see Table 3.19). A special input named CA is provided to modify the effective threshold during chip testing. The actual threshold will be  $(\text{inrh} + \text{inrl})/2 + \text{CA}$ . This input will be brought out to a test pad but is intended to be unconnected during normal operation in which case CA will not modify the threshold.

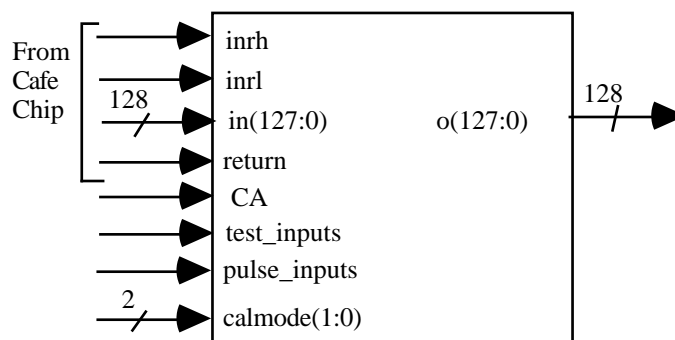


Figure 3. 2: Input Level translators Inputs/Outputs

**Table 3.1: Input Level Translator I/O Signal Definitions**

Signal Name	Active State/Edge	Function
inrh		High ref current (from CAFE-M chip)
inrl		Low ref (from CAFE-M chip)
in(127:0)		Hit Inputs (from CAFE-M chip))
return		Signal Return (to CAFE-M Chip)
CA		Bipolar current to modify threshold during testing
pulse_inputs	High	Pulses test inputs
testinputs	High	Enables testing of inputs
calmode(1:0)		Selects group of channels to be tested
o(127:0)		Data Outputs (to input reg.)

**Table 3.2: Test Modes**

testinputs	calmode(1)	calmode0)	Channels of Chip Tested
0	X	X	Testing disabled
1	0	0	in0, in4 in8,...in124
1	0	1	in1, in5 in9,...in125
1	1	0	in2, in6 in10,...in126
1	1	1	in3, in7 in11,...in127

**N.B.** The calmode bits are also used for selecting the calibration mode of the CAFE-M chip (see section 3.2.10)

### 3.2.2. Input Register and Mask Register

The functions of the input register and mask register will be implemented in a single functional block. The input/output connections to this block are shown in Fig 3.3

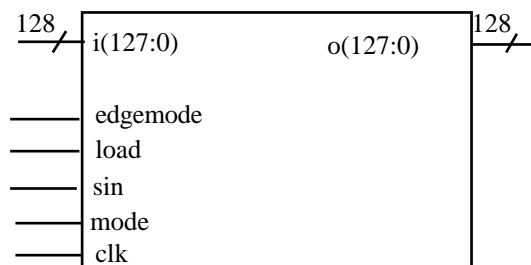


Figure 3.3: Input Register Inputs/Outputs

Table 3.3: Input Register I/O Signal Definitions

Signal Name	Active State/Edge	Function
i(127:0)		Hit Inputs (from input translators)
load	Active High	
sin		Configuration Data Inputs
mode	see Table 3.4	
clk	Pos Edge	
o(127:0)		Data Outputs (to pipeline)
edgemode	High	Enables edge detection logic.

#### 3.2.2.1 Input Register

This register latches the incoming data, delivering a well defined pulse width to the pipeline.

#### 3.2.2.2 Edge Detection Circuitry

The function of this block is to detect a low to high transition in the data entering the chip, and for each of such transition found the circuit block outputs a pulse of duration 1 clock cycle irrespective of the length of the incoming pulse. The effect of this block is that only a single '1' is written into the pipeline for every hit detected regardless of the response time of the CAFE-M chip. This circuitry can be turned on or off by setting the appropriate bit in the configuration register.

#### 3.2.2.3 Channel Masking Register

This register serves a dual purpose. Firstly, the Channel Mask register enables any bad or noisy channels to be turned off thus preventing them from increasing the data rate to a level which would create dead-time from false hits. To turn a bad channel off the corresponding bit of the register should be set to a '0' and to turn a channel on the corresponding bit should be set to a '1'. Secondly, it can be used during chip testing to apply a set of test patterns to the pipeline. The contents of this register can be changed by sending the appropriate control command to the chip.

Table 3.4: Masking Register Modes of Operation

mode	Mode of Operation
0	Normal Data Taking (Contents of register used to "Mask Inputs")
1	Test Mode (Contents of mask register are used to supply test values to pipeline)

### 3.2.3. Pipeline

This consists of a block of dual ported RAM 128-bits wide by 128 locations deep. This must run at 40MHz. The RAM block is addressed by an address pointer. During data taking the chip is instructed to write into the RAM and increment the address pointer every clock cycle. When the address register reaches a count of 127, it automatically resets itself on the following clock cycle. When a level one trigger arrives, the hit-pattern from the appropriate time bin is copied into the readout buffer together with the hit pattern from the previous and next clock cycle. Incorporated into the pipeline is the accumulator register. This function can be enabled and disabled by setting a bit in the configuration register.

#### 3.2.3.1 Accumulator Register

This is a 128-bit wide register used for accumulating hits in the pipeline. This register marks all channels that have been hit since it was last cleared. If the Accumulator Mode is selected in the configuration register, a L1 trigger results in the transfer of the contents of this accumulator into the readout buffer instead of the appropriate time bin of the pipeline. This accumulator column is cleared by a power-up or soft reset command.

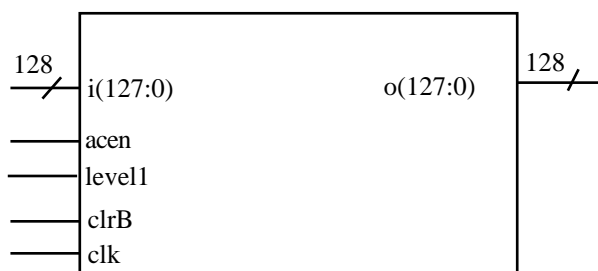


Figure 3.4: Pipeline Input/Outputs

Table 3.5: Pipeline Input/Output Signal Definitions

Signal Name	Active State/Edge	Function
i(127:0)		Data Input
acen	High	Enables Accumulator Register
level1	High	Reads Value out of pipeline
clrB	Low	Initialises pipeline pointers and clears accumulator register
clk	Pos edge	Clock input
o(127:0)		Data Output

### 3.2.4. Readout Buffer

Data corresponding to each L1 trigger will be held in a Readout buffer pending readout. This data buffering is needed to remove the statistical fluctuations in the arrival time of L1 triggers. Data compression and read out will be started only when this buffer is not empty. Three bits of data will be stored in this buffer for each channel per L1 trigger. These bits represents the three bunch crossings centred on the L1 trigger time and are set if the input was above threshold during the corresponding crossings. In the case when the Accumulator Register has been enabled, the contents of this register will be copied into the buffer 3 times resulting in the same amount of data being written into the readout buffer regardless of the operating mode. This buffer will be 128 bits wide by 24 locations deep. This is sufficient to hold the data from eight

L1 triggers. This satisfies the ATLAS specification of maintaining  $\leq 1\%$  data loss at a L1 trigger rate of 100 KHz and a strip occupancy of up to 1% [10].

### RAM and Pointers

This buffer will be implemented as a "barrel store", i.e. it will be addressed by 2 cyclic pointers, a write pointer and a read pointer. Once a pointer has reached the end of the block of RAM, it will return to the beginning of the block of RAM the next time it is incremented.

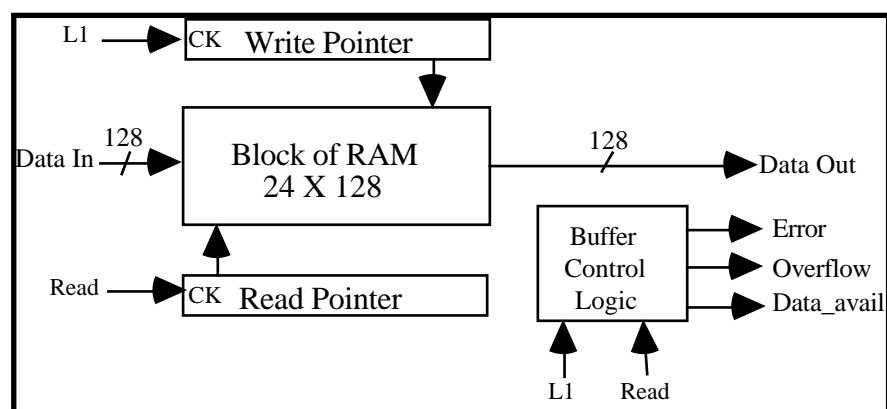
The write pointer will be allowed to go past the read pointer and over-write data that has not yet been read out. However, if this happens the Overflow flag will be set to indicate that data has been over-written. The read pointer will not be allowed to pass the write pointer and if the read pointer should catch up with the write pointer the EMPTY flag will be set. This is to prevent attempts to readout the buffer when there is no data in it.

### Overflow Counter

A counter will be used to track the number of events that have been over-written in the buffer. This counter will be incremented for every time an event is written into the buffer while the buffer is full. The outputs from this counter represents the number of events from which data has been lost. This counter is decremented for every event that is readout of the buffer, until it's value reaches zero. Once its value has reached zero, it is no longer decremented. This is because in this state all the events from which data have been lost will have been cleared and none of the data in the buffer will have been overwritten. Should this counter overflow, the ERROR flag will be set. This will occur after 16 events have been overwritten. This flag will remain set until either a software reset, or power-up resets has been issued to the readout buffer and associated logic.

### Flag Logic

Three signals DATA\_AVAIL, OVERFLOW and ERROR are produced by the readout buffer. DATA\_AVAIL indicates when there is data in the buffer to be readout. This signal is used by the Data Compression logic to determine when to start a readout cycle. OVERFLOW indicates when data in the buffer has been overwritten and hence data lost. OVERFLOW occurs when the buffer contains more than 8 events i.e. 24 samples. This signal is sent to the readout logic which results in the readout logic sending an error message to say that data from the current event being readout has been lost. Finally, the ERROR signal is generated when the buffer has overflowed and it has also lost track of the number of events from which data has been lost. This simulation occurs if the data from more than 16 events have been lost. This flag can only be cleared by issuing a reset to the chip.



**Figure 3.5: Block Diagram of the Readout Buffer**

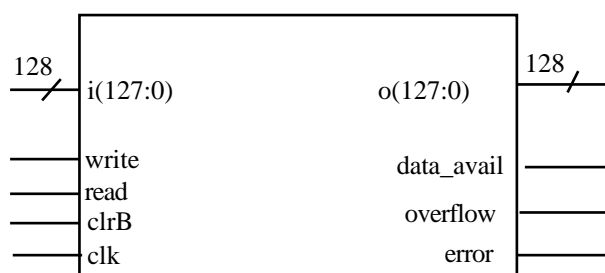


Figure 3.6: Readout Buffer Input/Outputs

Table 3.6: Readout Buffer Input/Output Signal Definitions

Signal Name	Active State/Edge	Function
i(127:0)		Data Input
write	High	Write value into readout buffer
read	High	Reads value from readout buffer
clrB	Low	Resets buffers pointers and counter
clk	Pos edge	Clock input
o(127:0)		Data Output
data_avail	High	Data available in buffer
overflow	High	Buffer Overflow
error	High	Buffer Error

### 3.2.5. Data compression logic

It is anticipated that on any event very few channels will contain hits. This fact can be used to reduce the number of bits of data that have to be read out of the chip for each event. The data compression logic works by examining in turn the 3 bits of data that make up the hit pattern for each channel. Each group of 3 bits is compared against one of 4 selectable criteria. If the pattern meets the criteria, then the hit pattern from that channel is sent to the readout circuitry for transmission, if not, no data is sent from that channel and the hit pattern from the next channel is examined. This process is repeated until the hit patterns from all 128 channels have been examined. The following table shows the 4 selection criteria (currently there are only plans to use 3, the 4th is for chip testing only).

Table 3.7a: Data Compression Criteria

mode(1:0)	Name of Selection Criteria	Hit Pattern (Oldest data bit 1st)	Usage
00	Hit	1XX or X1X or XX1	Detector alignment
01	Level	X1X	Normal Data Taking
10	Edge	01X	Normal Data Taking
11	Test	XXX	Test Mode

**N.B. X = Don't care state.**

This block operates as follows.

As soon as the chip receives a L1 trigger, the three 128-bit words that make up an event are written into the read out buffer. This results in the empty flag on the readout buffer being negated, indicating that there is data to be processed. The data compression logic monitors the state of this flag until it finds that there is data available. Providing that it is not already processing data, it then proceeds to read in the three 128-bit words that make up an event from the readout buffer.

The next thing that happens is that the data compression logic re-arranges the order of the data from being 3 128-bit words into 128 3-bit words. The reason for this is that the data compression algorithm requires all 3 samples of an event to be examined in parallel.

The data compression logic then starts to scan through all the channels in turn until it finds one which has a pattern of hits which matches the data selection criteria. If it finds such a pattern of hits, it asserts the "datavalid" signal and places the pattern of hit bits on the "hit<2:0>" outputs and places the address of the hit channel on the address outputs "ch<6:0>". The logic then waits until the readout logic signals it to proceed by asserting the "next" input. The data compression logic responds to "next" by presenting the address and data for the next hit found if any. If the next hit found is on the next adjacent channel, the "adj" is asserted with the data from the previous channel. If no more hits are found, the "end" signal is asserted.

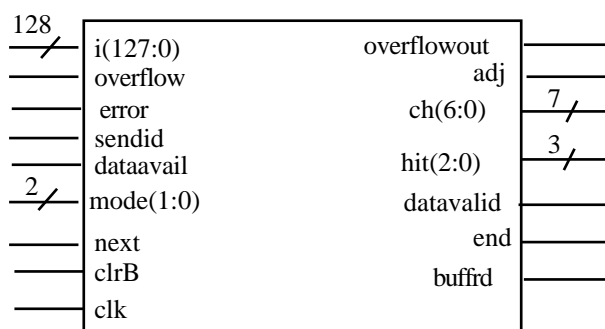
In certain situations, e.g. in the case of a overflow of the readout buffer, it is not necessary for the data compression logic to process the data from the readout buffer but it is still necessary for it to read the 3 values from the buffer in order to flush them from the readout buffer. There are 3 cases when this happens, these are listed below in order of priority.

- 1) When the chip is in its SEND\_ID mode of operation.
- 2) When the Readout Buffer error flag has been set.
- 3) When the Readout Buffer overflow flag has been set.

The following table shows how the datavalid, end and overflowout outputs are used to indicate the status of the data compression logic.

**Table 3.7b: Data Compression Logic Output States**

datavalid	end	overflowout	condition
low	low	low	no events available to be read out i.e readout buffer empty.
high	low	low	data from hit channel waiting to be read out. (not last channel)
high	high	low	data from last hit channel waiting to be read out.
low	high	low	all hits read out or no hits found
low	low	high	data for event lost due to readout buffer overflow



**Figure 3.7: Data Compression Logic Input/Outputs**

**Table 3.7c: Data Compression Logic Input/Output Signal Definitions**

Signal Name	Input/Output	Active State/Edge	Function
i(127:0)	input		Data Input
overflow	input		Overflow output from readout buffer
error	input		Error output from readout buffer
sendid	input		indicates chips mode of operation
dataavail	input	High	Data available to be readout
mode1:0)	input		Selects data compression mode
next	input	High	Find next hit channel.
clrB	input	Low	Resets logic



clk	input	Pos Edge	Clock Input
overflowout	output	High	Overflow output to readout circuitry
adj	output	High	Next Hit found on adjacent channel
ch(6:0)	output		Channel address of Hits
hit(2:0)	output		Hit Data pattern
datavalid	output	High	Hit Data outputs valid
end	output	High	Last Channel scanned
buffrd	output	High	Reads Value out of Readout Buffer

### 3.2.6. Readout Circuitry

The readout circuitry will be responsible for capture and release of the token and outputting data from the chip. The readout circuit always waits until the token arrives. On arrival of the token, it checks if any hits have been found by the data compression logic. If so, it then outputs the appropriate header information. It then proceeds to output the address of the hit channel together with the data from that channel. Once the readout circuitry has finished sending the data from one channel, it proceeds to output the data from then next channel. In the situation where one or more neighbouring channels are to be read out, only the address of the first channel is output., but the data from all hit channels is sent. This process continues until the data compression logic indicates that all channels have been examined by asserting "end". Once all the data corresponding to a single event has been read out, a token is sent out to the next chip in the readout chain. This token will be sent out ahead of the last bit of data sent out. If the chip has no data to be readout, circuitry sends out a "No hit data " code and passes the token on to the next chip in the chain.

If the chip is in "send-id" mode or the readout buffer has overflowed or generated an error condition, the readout circuitry sends the appropriate error packet or configuration data packet. In these cases the readout circuitry is still required to signal to the data compression logic that it has processed an event by asserting the "next" signal. This operation is needed so that a correct count of the number of events waiting to be read out can be maintained.

In the case of an error condition occurring, e.g. attempt to readout data and no data available, the appropriate error code will be sent by the readout logic. If the chip is in the "send\_id" mode of operation, no data or error codes are output from the chip but instead a special packet of data containing information about the chips current configuration is sent.

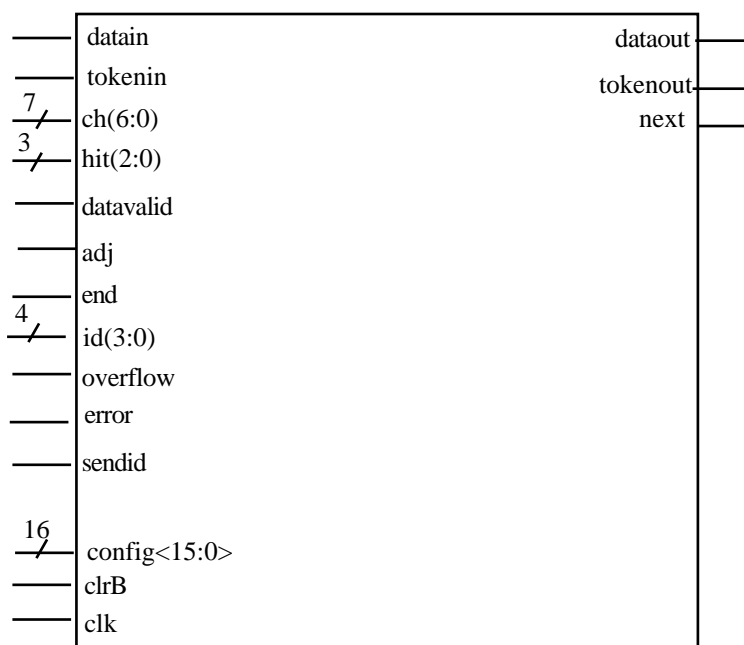


Figure 3.8: Connections to Readout Circuitry

**Table 3.8: Readout Logic Input/Output Signal Definitions**

Signal Name	Input/Output	Active State/Edge	Function
datain	input		Data Input
tokenin	input	High	Token Input
ch(6:0)	input		Address of Hit Channel
hit(2:0)	input		Hit data pattern
datavalid	input	High	Data available for sending
adj	input	High	Hit found on adjacent channel
end	input	High	End of data to be sent
id(3:0)	input		LS 4 bits of chip address
overflow	input	High	Readout buffer Overflow
error	input	High	Readout Buffer Error
sendid	input		Chip mode of operation
config<15:0>	input		Data from config-reg
clrB	input	Low	Resets circuit
clk	input	Pos Edge	Clock input
dataout	output		Data output
tokenout	output		Token Output
next	output	High	Scan Next Channel

### **3.2.7. Readout Controller Block**

This block is to control the readout of data from several ABC chips connected together in a token chain. This block is enabled by placing the chip in "Master Mode". This block has to detect when a L1 trigger has been received, issue a token to all the ABC chips connected to it, collect all the data from the chips and tag the data with the bunch crossing number from which it came and the number of Level 1 Trigger. This block then has to transmit this data serially to the LED driver chip.

#### **3.2.7.1 L1 Counter**

This is a 4-bit binary counter which is incremented every time the chip receives a level 1 trigger. The counter is zeroed by either a hardware reset or a software reset.

#### **3.2.7.2 Bunch Crossing Counter**

This is an 8-bit binary counter which is incremented on every clock cycle. This counter is zeroed by either a hardware reset, a software reset, or a special BC Reset Command

#### **3.2.7.3 Event FIFO**

This is a 24 location deep, 12-bit wide FIFO. Each time the chip receives a L1 trigger, the output of the L1-Counter and the Bunch crossing counter are loaded into the FIFO prior to the counters being incremented. These values are read from the FIFO every time an Event is readout and are used to tag the data with 12-bits of information about which trigger number and bunch crossing number the data came from.

#### **3.2.7.4 Token Generation Logic**

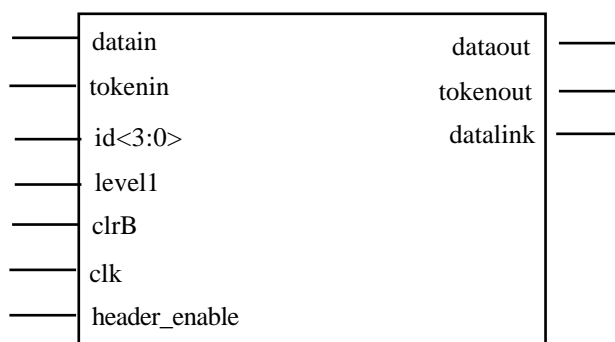
The purpose of the token generation logic is to detect when the chip has received an L1 trigger and to generate a token to initiate the read out of data from that L1 trigger. This logic waits until the Event FIFO becomes not empty and it then issues a token. It then monitors the data passing through it from all the chips in the chain looking for a "Trailer" bit pattern. It waits until this trailer is detected before checking to see if the Event FIFO is empty. If the Event FIFO is still not empty it repeats the cycle.

### 3.2.7.5 Data Formatting Logic

The purpose of this logic is to attach the header information to the packets of data output from the chip on the Serial Data Output.

### 3.2.7.6 Serial Data Output Driver

This circuit block generates the output signals to the data links which send data to the DAQ system.



**Figure: 3.9: Connections to Readout Controller Circuitry**

**Table 3.9: Readout Controller Input/Output Signal Definitions**

Signal Name	Active State/Edge	Function
datain		Serial Data Input
tokenin	High	Token Input
id<5:0>		Address of chip
level1	High	Level 1 Trigger
clrB	Low	Resets block
clk	Pos Edge	Clock input
dataout		Serial data output
tokenout	High	Token Output
header_enable	High	Enables Generation of Packet Header
trailer_enable	High	Enables Generation of Packet Trailer
token_back	High	Input for Token output from ROL
datalink		Serial data out to link driver

### 3.2.8. Command Decoding

The command and control information all comes into the chip on the command input pins. There are two main classes of information which arrive here, Level 1 Triggers Commands and Control Commands. These are distinguished by a 3-bit code. Furthermore two types of Control Commands are possible, Fast Control Commands and Slow Control Commands. Depending on which class arrives, further information may follow. This further information will also need decoding, formatting and sending to the appropriate functional blocks of the chip. More detailed information is contained in 3.2.18 and the actual data fields of the commands are listed in Tables 3.18 and 3.19. The two classes of Commands and two types of Control Commands are:

#### Level 1 Trigger Command

If the 3-bit code indicating this command is received, the control logic writes 3 samples from the pipeline or the accumulator register, into the Readout Buffer.

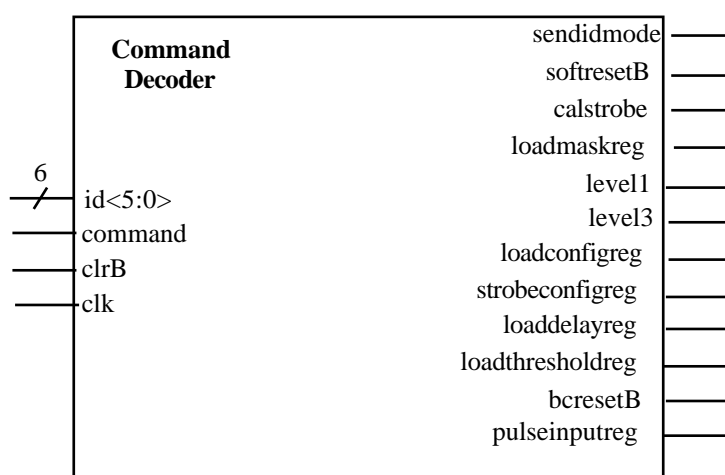
## Control Commands

If the 3-bit code indicating a Control Command is received, the second field of 4 bits is decoded to determine if it is a Fast Control Command or a Slow Control Command. If a Fast Control Command is decoded, the appropriate command is executed. No address or data fields are included in these commands..

### Slow Control Command

If the second field of the command is decoded to be a Slow Control Command, the third, forth, fifth and possibly sixth field is decoded to determine the full action required. These Slow Control Commands are of variable length and the contents of the third field determines the total number of bits to be processed.

The command decoder block is required to decode the command and send the relevant instruction and data to other parts of the chip. The input/output connections to this block are shown in Figure 3.10.



**Figure 3.10: Command Decoder Inputs/Outputs:**

**Table 3.10: Command Decoder Input/Output Signal Definitions**

Signal Name	Active State/Edge	Function
id<5:0>		Chip ID
command		Command Data Input
clrB	Low	Reset Block
clk	Pos Edge	Clock Input
sendidmode		Sets chips mode of operation
softresetB	Low	Software controlled reset
calstrobe	High	Send Calibration pulse to CAFE-M
loadmaskreg	High	Load Mask Register
level1	High	Level_1 Trigger received
level3	High	Same as level1 but High for 3 cycles
loadconfigreg	High	Loads Configuration reg.
strobeconfigreg	High	Strobes data into configuration reg
loaddelayreg	High	Load Strobe Delay Register
loadthresholdreg	High	Load Threshold Register
bcresetB	Low	Bunch crossing reset

### 3.2.9. Configuration Register

This is a 16-bit register which is used to hold information about the chip's current configuration. The following table defines the usage of the bits in this register. The power up value of this register will be zero. The input/output connections to this block are shown in Figure 3.11. Data is shifted into this register MS bit first. The contents of this register are not effected by a software reset command.

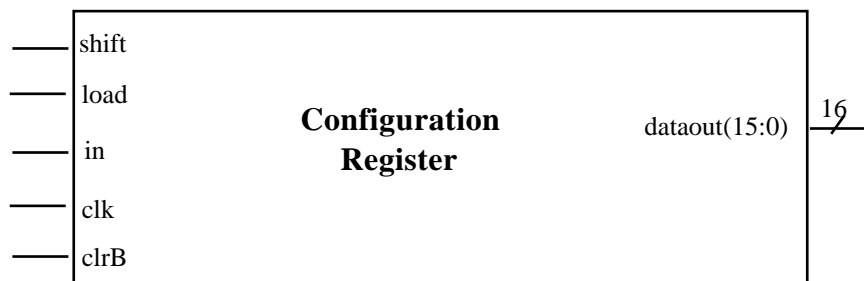


Figure 3.11: Configuration Register Inputs/Outputs

Table 3.11a: Configuration Register Input/Output Signal Definitions

Signal Name	Active State/Edge	Function
load	Pos edge	Transfers data to register outputs
in		Serial Data input
shift	High	Enables data to be shifted into reg
clrB	Low	Resets register to default values
dataout(15:0)		Data Outputs See section 3.2.5

Table 3.11b: Configuration Register Contents

Bit	Name	Function
0-1	Readout Mode	Selects the data compression Criteria (see Table 3.7a)
2-3	Cal_Mode(1:0)	Selects the Calibration code for the CAFE-M chip (see Table 3.12). The state of these two bits also determines which channels are tested when Test Mode is enabled. (see Table 3.2).
4	Cal_Enable	When this bit is set the calibration strobe output to the CAFE-M chip is enabled
5	Test_Mode	When this bit is set test values are applied to the channels defined by bits 2 & 3 of this register.
6	Edge_Detect	When this bit is Set the edge detection circuitry in the input stage is enabled.
7	Mask	When this bit is set the input register is disabled and the contents of the mask register are routed into the L1 pipeline.
8	Accumulate	When this bit is set the Accumulate function is enabled. (see section nnn)
9	Input_Bypass	This bit determines which set of token/data inputs are active.(see section nnnn)
10	Output Bypass	This bit determines which set of token/data outputs are active.(see section nnnn)
11	Master *	When clear the chip acts as a Master providing the masterB input pin has be asserted
12	End	When set this bit configures the chip as the end of a readout chain.
13	Feed_Through	When clear the chip outputs a 20MHz clock signal but only is the chip has been configured as a Master (see above)
14	ABCD	reserved for ABCD (This bit has no effect on the operation of the ABC)
15	Self Destruct	When set this bit prevents data getting into the hands of the bad guys.

\* This bit is "ored" with the value on the "masterB" input. If the result is "0", the chip is placed into master mode. Otherwise, it is placed into slave mode.

### 3.2.10. Calibration Logic

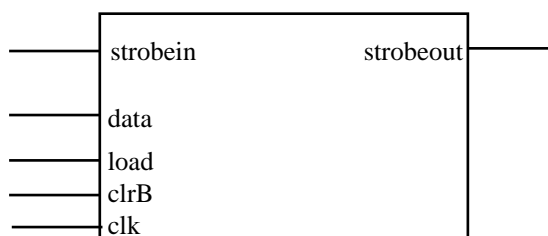
The calibration logic produces a calibration strobe signal for the CAFE-M chip. This strobe is produced in response to a control command when the "Cal Enable" bit is also set in the configuration Register. A two-bit calibration code is also sent to the CAFE-M chip which selects one of the four possible patterns in the CAFE-M chip. The two-bit calibration code outputs are single-ended CMOS levels. The calibration Strobe is a differential current output. The calibration Strobe signal must be sent to the CAFE-M chip a fixed number of clock pulses after receipt of the control command. The delay from the rising edge of the clock signal to the rising edge of the strobe signal is determined by the value loaded into the Strobe Delay Register. This delay can be adjusted in 64 equal steps over a range of values exceeding the length of one clock period.

**Table 3.12: Calibration Codes**

Cal enable Bit	Cal Mode Bit 1	Cal Mode Bit 0	Channels of CAFE-M Chip Pulsed
0	X	X	Calibration disabled
1	0	0	in3, in7 in11,...in127
1	0	1	in2, in6 in10,...in126
1	1	0	in1, in5 in9,...in125
1	1	1	in0, in4 in8,...in124

### 3.2.11. Strobe Delay Register

The Strobe Delay Register is a 8 bit register of which only the least significant 6 bits are used. The value stored in this register determines the relative delay between the rising edge of the Calibration Strobe output to the CAFE-M chip and the rising edge of the clock input. This delay can be set in 64 steps, of approximately  $1.1\text{ns} \pm 0.5\text{ns}$  each. Thus enabling the delay of Calibration Strobe to be swept through at complete clock cycle at 40MHz. Data is shifted into the this register with the MS bit first.



**Figure 3.12: Strobe Delay Register Inputs/Outputs**

**Table 3.13a: Strobe Delay Register Input/Output Signal Definitions**

Signal Name	Active State/Edge	Function
strobein		Strobe input
data		Data input to register
clrB	Low	Resets register
clk	Pos Edge	Clock input
load	High	Loads delay value into register
strobeout		Delayed version of STROBE_IN

The value of delay is determined according to the following formula.

$$\text{delay} = \text{min\_delay} + (\text{register\_value} \times \text{step\_value})$$

Where :

**min\_delay** is the delay produced when the register is set to zero .  
**register\_value** is the value written into the delay register (least significant 6 bits only)  
**step\_value** is the increase in delay produced by incrementing the contents of the delay register (typically 1.13ns.)

### 3.2.12. DAC Register

The DAC register is a 16 bit register which holds 2 8-bit values. A threshold value is held in the MS byte of this register and a calibration amplitude value is held in the LS byte of this register. The outputs of this register are used to control 2 separate 8-bit DACs. The outputs from these DACs supply 2 independent DC current levels to the CAFE-M chip. The MS bit is shifted into this register first.



Figure 3.13: DAC Register Inputs/Outputs

Table 3.14: DAC Register Input/Output Signal Definitions

Signal Name	Active State/Edge	Function
load	Pos Edge	Load Values into register
data		Data input to register
clrB	Low	Resets register
clk		clock input to register
threshold		Threshold Output from 1st DAC
cal amp		Calibration Amplitude Output from 2nd DAC

### 3.2.13 DACs

These are two 8-bit DACs which are used to set a threshold and calibration value supplied by the ABC chip to the CAFE-M chip. These DACs will receive an input reference current "iref" from the CAFE-M chip. This reference current will be scaled at the output of the DAC by a value of 0 to 255 depending on the setting of the DAC register.

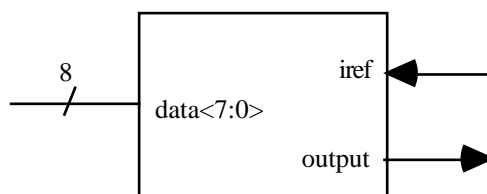


Figure 3.14: I/O connections to DACs

### 3.2.13a Bias DACs

On the ABCD chip there will be 2 additional 5-bit DACs

5-bit DAC for setting the current in the input transistor instead an external reference voltage as implemented in CAFE-M,

5-bit DAC for setting the shaper bias current which will allow to control the shaper gain (not foreseen in CAFE-M).

These DAC s will be set be sending a additional command not implemented on the ABC chip. This command sends a 16-bit value, which is loaded into a 16bit shift register similar to the DAC register. The MS bit will be sent first. Bits 5:0 of the register will be used to set the DAC which controls the current in the input transistor. Bits 14:8 will be used to set the DAC which controls the shaper bias current. These Bit ranges have been chosen to align the data for both DACs on byte boundaries.

### 3.2.14. Clock and Command Inputs

Two sets of clock and command inputs will be provided in order to make the system in which the "ABCs" will be used fault tolerant and to provide an additional method of setting up the timing of the system. Each chip will be supplied with two independent sources of clock and commands. In the event of the fall out of one of these sources, the alternative source can be used. An external input to the chip "select" will be used to determine which pair of inputs will be used by the chip.

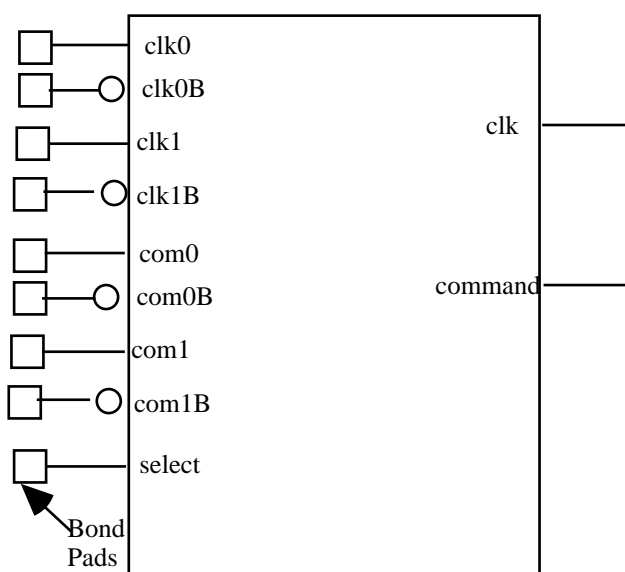


Figure 3.15: Clock & Command Data Inputs



**Table 3.15a: Clock Input/Output Signal Definitions**

Signal Name	Active State/Edge	Function
clk0		default Clock Input
clk0B		Complement of above
clk1		reserve Clock Input
clk1B		Complement of above
com0		default Command Input
com0B		Complement of above
com1		reserve Command Input
com1B		Complement of above
select		Selects which pair of inputs to use
clk		Clock output
command		Command Data Output

**Table 3.15b: Clock Input Modes of Operation**

select	clk	command
Low	clk0	com0
High	clk1	com1

**3.2.15. Chip ID.**

To enable a chip to be individually addressed five inputs (ID(4:0)) will be used to implement a geographical addressing scheme. This is because there will be a total of 12 chips on each module (6 per side), and under certain conditions it may be necessary to address all the chips on 2 modules using the same control line. These inputs will be wire bonded to a unique set of logic levels on each chip mounted on the detector module. This set of logic levels will form a geographical address which will enable individual chips on the module to be addressed. The chips will be bonded according to the following scheme. For historical reasons the command decoder on the chip has been designed to decode a 6-bit geographical address however the MS bit of the chips geographical address is always set to a '1'. The sixth bit ID(5) is not brought out of the chip but instead set to a logic 1 level. Internal pull ups on these inputs will force these inputs to go to a logic '1' level if left unconnected.

**Table 3.16a: Geographical Addresses (ID(5-0))**

id(5:0)	Type of Chip Selected	Odd/Even Module
10aaaa	ABC	Even
11aaaa	ABC	Odd
111111	All ABC chips	Both

N.B aaa is the 4 bit address of the ABC chip on the hybrid see Table 3.16b

**Table 3.16b: ABC Geographical Addresses (ID3-0)**

Chip Position on Hybrid *	ID(3)	ID(2)	ID(1)	ID(0)
M0	LOW	LOW	LOW	LOW
S1	LOW	LOW	LOW	HIGH
S2	LOW	LOW	HIGH	LOW
S3	LOW	LOW	HIGH	HIGH
S4	LOW	HIGH	LOW	LOW
E5	LOW	HIGH	LOW	HIGH
M8	HIGH	LOW	LOW	LOW
S9	HIGH	LOW	LOW	HIGH
S10	HIGH	LOW	HIGH	LOW
S11	HIGH	LOW	HIGH	HIGH
S12	HIGH	HIGH	LOW	LOW
E13	HIGH	HIGH	LOW	HIGH

\* Refer to diagram of Hybrid layout

**3.2.16. Token Input/Output Circuit**

In order to provide some measure of fault tolerance in the system, a token and data bypass circuit will be built into each chip. The purpose of this circuit is to enable a chip to source or send it's token and data to another chip other than it's direct neighbours. Each chip will have 2 token and data inputs. It will also have two token or data outputs. Pairs of inputs and pairs of outputs will be connected to different chips enabling it to send or receive data from one of two chips. In this way, should one of the chip,s neighbours fail, an alternative chip can take its place. Commands are used to direct each chip to use its normal or bypass inputs and outputs.

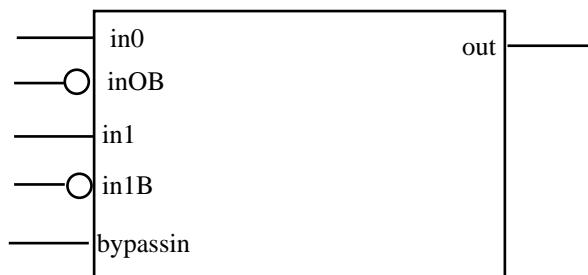


Figure 3.16: Token and Data Inputs circuit

Table 3.17a: Token and Data Input Signal Definitions

Signal Name	Active State/Edge	Function
in0		1st Data/Token Input
in0B		Complement of above
in1		2nd Data/Token Input
in1B		Complement of above
bypassin		When High in1 is selected else in0
out		Token /Data Output

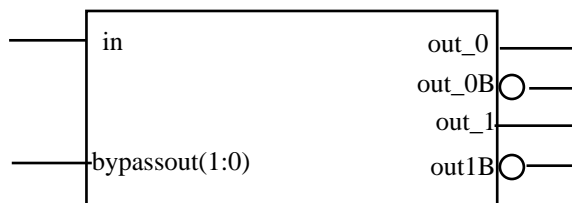


Figure 3.17: Token and Data Outputs circuit

Table 3.17b: Token and Data Output Signal Definitions

Signal Name	Active State/Edge	Function
in		Token/Data in
bypassout(		When High out1 is enabled else out0
out0		1st Data/Token Output
out0B		Complement of above
out1		2nd Data/Token Output
outB		Complement of above

### 3.2.17. Test Circuitry

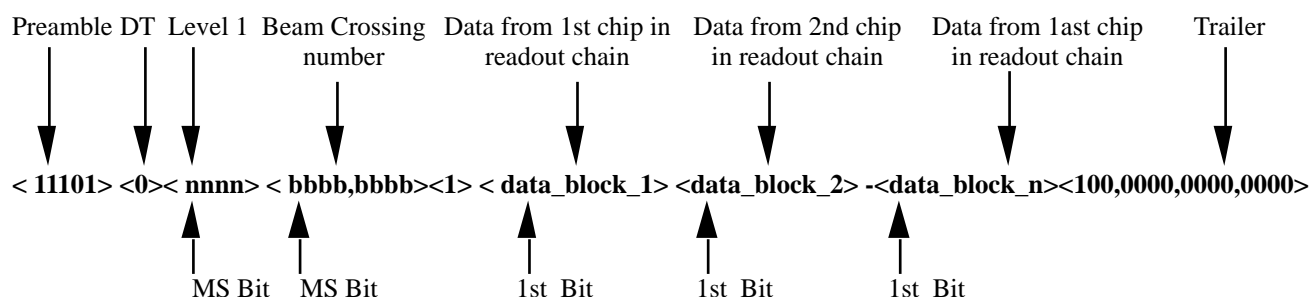
To simplify the testing of the chip during production and additional test pads have been included on the chip to enable selected parts of the chip to be tested easily

### 3.2.18. Readout Protocols

Output data from the ABC can be grouped into one of five classes:

#### 1) Module Data

This type of data packet is only output from chips which have been configured as masters. The packet consists of 2 elements, a 13-bit header generated by the Master ABC chip and a string of physics data packets, from the all ABC chips daisy-chained together including the Master ABC chip.



**Figure 3.18a: Module Data Format**

#### DT(Data Type)

The value of this bit determines the type of data which follows. This can either be Level 1 Trigger Data (DT=0) or Information Data (DT=1). In the case of the ABC chip only Level 1 trigger Data is Sent and hence this field is always set to '0'.

#### Level 1

Current count of Level1 Trigger modulo 16 since the last system reset. This field can be used for event building by the DAQ and also to monitor for lost data.

#### Bunch Crossing Number

Current count of Bunch Crossing modulo 256 since the last system reset or BC Reset command. It is intended to monitor for clock pulses lost by the on-detector electronics and can be used to tag one bunch crossing out of the complete ring of the LHC.

#### Data Block

This is the data packet set from each chip including the master chip. This data block can be any of the four following types, Physics Data, No-Hit Data, Error Data or Configuration Data.

#### 2) Physics Data

This type of data packet is used to send the compressed hit data from the detector. The format of this data is a series of one or more data packets.

`<data_packet_1><data_packet_2> - - - <data_packet_n><data_packet_n+1>`

There are 2 types of data\_packet, isolated hit packet and non-isolated hit packet. A physics data packet can consist of any combination of these 2 types of packet.

#### Isolated Hit Data-Packet.

This type of packet is used to send the hit information from a hit channel on a chip when none of its neighbouring channels have been hit.

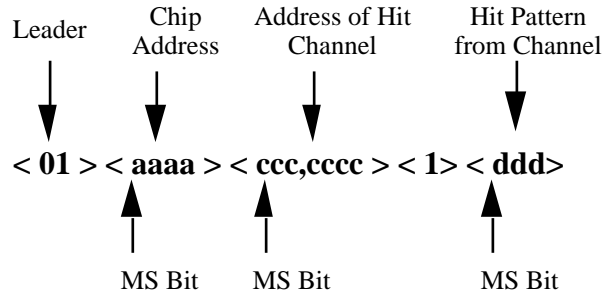


Figure 3.18b: Isolated Hit Data Format

Adjacent Hit Data-Packet

This type of packet is used to send data from a group of 2 or more adjacent channels which have been hit. Only the channel address of the 1st channel in the group is sent. It should be noted that this will also be the lowest numbered channel in the group. The chip address and channel address's of the other channels can be derived from that of the 1st and hence are not sent.

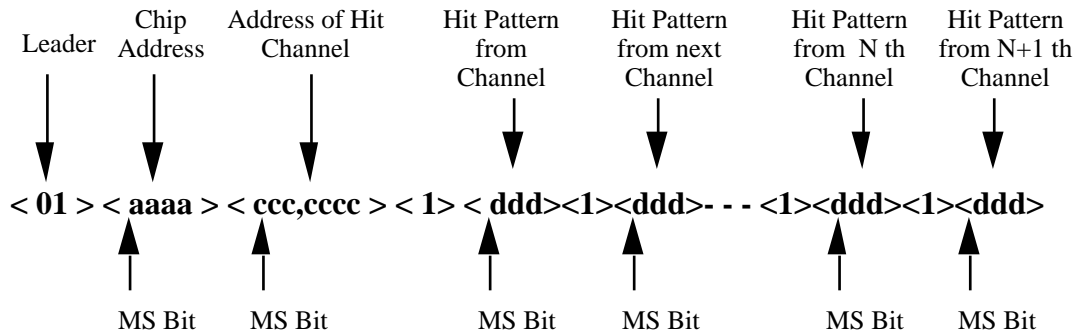


Figure 3.18c: Adjacent Hit Data Format

where

aaaa            LS 4 bits of the chips geographical address  
ccc,cccc       7-bit address of the channel on which the hit or 1st channel in a groups of hits was found  
(See Figure 3.25 for the physical location of channel addresses.)  
ddd            Is the 3 bit hit pattern read out from the hit channel. (Previous, Current, Next)

Example

The following physics data packet would be send out from a chip with a geographical address of 2D<sub>H</sub> and hits on channels 3, 5 and 6.

<01><1101><000,0011><1><ddd3><01><1101><000,0101><1><ddd5><1><ddd6>

where ddd<sub>3</sub> = data from channel 3, ddd<sub>5</sub> = data from channel 5, ddd<sub>6</sub> = data from channel 6

3) Null Data

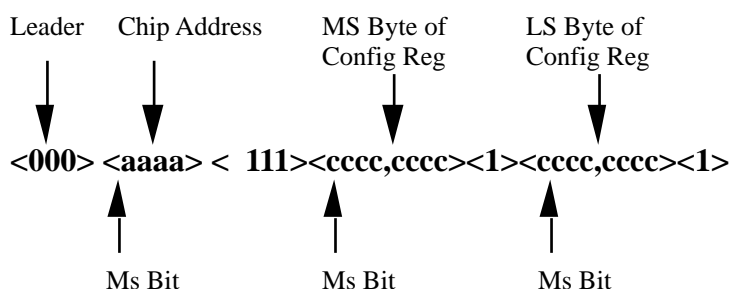
If a chip has received the event currently being read out but has not found any hit channels, it outputs a Null Data Packet.

< 001>

Figure 18d: Null Data Packet

#### 4) Configuration Data

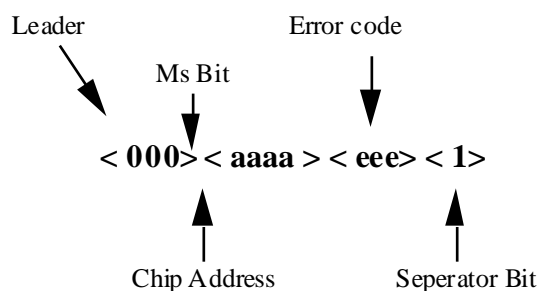
Configuration data is sent by the chip in response to a L1 trigger when the chip is in its Send\_ID mode of operation, i.e. the chip is not sending data. A packet of data is sent from the chip which contains the chips address and the contents of the chips configuration register.



**Figure 18e: Configuration Data Packet**

#### 5) Error Data

Error data is only sent if the chip detects an error, e.g. Buffer overflow. In this cases a data packet of the following format is sent:



**Figure 3.18f: Error Data Format**

#### Error Codes:

3 codes have been defined :

eee = 001      No Data Available (The chip has not received an L1 command)  
 eee = 010      Buffer Overflow  
 eee = 100      Buffer Error (Soft Reset needed)

N.B. Error messages are only sent if the chip is Data\_Taking Mode. (see section 3.2.19 )

#### 3.2.19. Control Protocol

There are two main classes of commands, Level 1 Trigger Commands and Control Commands, and there are two types of Control Commands, Fast Control Commands and Slow Control Commands. It is not expected that the Slow Control Commands will be issued during data taking operation.

**Table 3.18: Commands**

Type	Field 1	Field 2	Field 3	Description
Level 1	110	---	---	Level one Trigger
Fast	101	0100 or 0010		Soft Reset BC Reset
Slow	101	0111	Command	Slow Control Command see Table 3.19

### 3.2.19.1 Level 1 trigger Command:

This is the most frequently issued packet and hence the smallest. All ABC chips that receive this packet act on it. There is no addressing. If this command is received 3 samples are readout out of the pipeline and written into the readout buffer.

### 3.2.19.2 Fast Control Command:

This type of command is sent when a command has to be issued to the chip more quickly than can be achieved by sending a slow command to the chip. In the case of the ABC chip, only two commands of this type have been defined, i.e. the Soft Reset and BC Reset commands. It is expected that these commands will be sent to the chip at regular intervals during periods of time when no Level 1 Triggers will be sent to the chip. The purpose of these commands is to perform a limited reset of the chip. (see section 3.2.19 for details)

### 3.2.19.3 Control commands

These are long packets that enable the operation of the chip to be controlled. While they are being sent, it is not possible to send a first level trigger. Only the addressed ABCs will act on the packet, unless the address sent equals '111111', in which case all chips will act on the packet. All chips that receive the packet must decode it, even if they do not act on it. This is to avoid un-addressed ABCs erroneously decoding parts of the data field as the start of packets. To ensure that the chip does not respond to erroneous commands the chip will be placed out of taking mode for any command it receives which effects the configuration of the chip, i.e. all commands in which the 1st bit of field 5 is '0'. Hence it will be necessary to issue a command to the chip to enable data taking after issuing a command to change its configuration. When the chip is not in data taking mode, it will send its ID instead of real data in response to a L1 Trigger. This is the power-on default state.

### 3.2.19.4 Erroneous Commands

If the chip should receive a command that it doesn't recognise it is dealt with in the following ways.

#### Invalid Field 3

If field 1= 101 and field 2=0111, but field 3 is not one of the allowed values, there is some fault in the command. Therefore, the command decoder flushes 15 bits (Field 1, 2, 3) plus however many bits Field 3 designated and starts looking at the next bit for a new command.

#### Mismatched Chip Address of Faulty Field 5

If the chip address (Field 4) does not match the chip address established by bonding pads id<4:0> or Field 5 does not match a valid pattern or the value of field 3 is not appropriate for the command given by field 5, the remaining bits in the input stream indicated by Field 3 are flushed and the command decoder starts at the following bit to look for a new command.

**Table 3.19: Control Commands**

Field 3	Field 4	Field 5	Field 6	Description
0001,1100	aaaaaa	000 000	dddd,dddd,dddd,dddd	Write to Configuration Register
1000,1100	aaaaaa	001 000	d---,---,---,---d	Write to Mask Register
0001,1100	aaaaaa	010 000	dddd,dddd,dddd,dddd	Write to Strobe Delay Register
0001,1100	aaaaaa	011 000	dddd,dddd,dddd,dddd	Write to Threshold Registers
0000,1100	aaaaaa	100 000	-----	Pulse Input_Reg
0000,1100	aaaaaa	101 000	-----	Enable Data taking Mode
0000,1100	aaaaaa	110 000	-----	Issue Calibration Pulse
0000,1100	aaaaaa	111 000	dddd,dddd,dddd,dddd	Load Bias DAC (ABCD only) *

**N.B**

xxx = don't care state.

aaaaaa = 6 bit chip address(MS bit first)

dddd = data value for register (MS bit first except for the data sent to the Mask Register which is sent LS bit first))

\* This instruction only has any effect on the ABCD chip.

Field 3

This is an 8 bit count of the number of bits in the following instruction.

Field 4

This is the 6-bit address of the chip for which the command is intended. (See Section on Geographical Address.)

N.B. To Address an ABC chip the MS bit must always be set.

Field 5

This 6 bit field is used to determine into which register on the chip the data contained in the following field will be written or which command sequence is to be executed.

Field 6

This field holds the data that is to be written into the selected register. With the exception of instructions which load the mask register, this field will be 16-bits long.



### **3.2.20. Chip Initialisation and Configuration**

The chip has 2 modes of operation, "Send\_ID Mode" and "Data\_Taking Mode". After a Power-up reset the chip is placed into Send\_ID mode.

#### **3.2.20.1 Send\_ID Mode**

In this mode of operation the chip sends its ID and Configuration data in response to a L1 trigger. There is no command which explicitly places the chip into this mode of operation, however, any attempt to alter the contents of the chip's various registers automatically results in the chip being placed into Send\_ID mode.

#### **3.2.20.2 Data\_Taking Mode.**

When the chip is not in Send\_ID mode it is in Data\_Taking Mode and visa-versa. In this mode of operation the chip sends out any physics data that it has. The chip may be placed in this mode of operation by sending a command to the chip to enable data taking. The chip may be taken out of this mode of operation and placed into Send\_ID mode by either a Power\_up reset or any attempt to change the contents of the chip's registers.

#### **3.2.20.3 Clock Feed Through**

If the clock feed through bit in the configuration register has been cleared and the chip has been configured as a Master, the chip outputs the chips system clock divided down by 2 from it's data output pins. This feature has been included to simplify system testing.

### **3.2.21. Resets**

There are three kinds of reset in the system.

#### **3.2.21.1 Power up reset**

The power-up reset is an asynchronous (i.e. clock independent) reset that sets the value of the chips command register to zero, it's default value, and clears all the buffers in the chip, thus placing the chip into a well defined state. This type of reset is issued automatically when power is first applied to the chip. Provision will be made to enable this signal to be supplied externally to the chip.

#### **3.2.21.2 Soft Reset**

This type of reset is sent to the chip via a command instruction. Its purpose is to clear all the buffers in the chip, while leaving the configuration of the chip unaffected. This type of reset will be issued to the chip periodically during data taking to eliminate synchronisation errors.

1) Upon receipt of the reset command, the ABC chip resets all internal counters, clears tokens and sets itself to the no-data state. If it was transmitting data, it terminates this immediately.

**N.B.** It should be noted that the off-detector system must then be able to determine the last complete event transmitted before the reset and discard it (Complete in the sense that all read-out chains supply a header and trailer). With either reset it must also be able to recognise and discard partial events since there is no guarantee that different read-out chains will be reading the same event when the periodic reset arrives.

#### **3.2.21.3 BC Reset**

This type of reset is sent to the chip via a command instruction. Its purpose is to zero the Bunch Crossing counter. It has no effect on the operation of any other part of the chip.

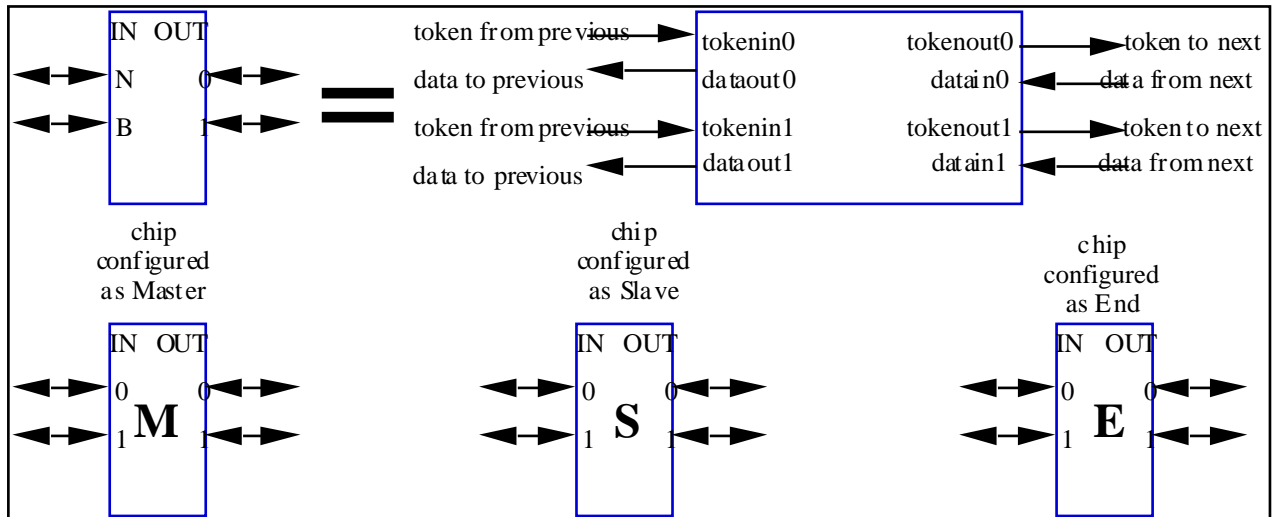
The following sequence of instructions should normally be sent to the chip after power-up

- 1) Send command to load the configuration register with the appropriate settings.
- 2) Send a command to load the mask register
- 3) Send a series of commands to load the DAC register/s and Delay registers
- 4) Send a command to place the chip into data taking mode.

The chip will now be in a state to receive L1 trigger command and send data.

### 3.2.22. Data Readout and Redundancy

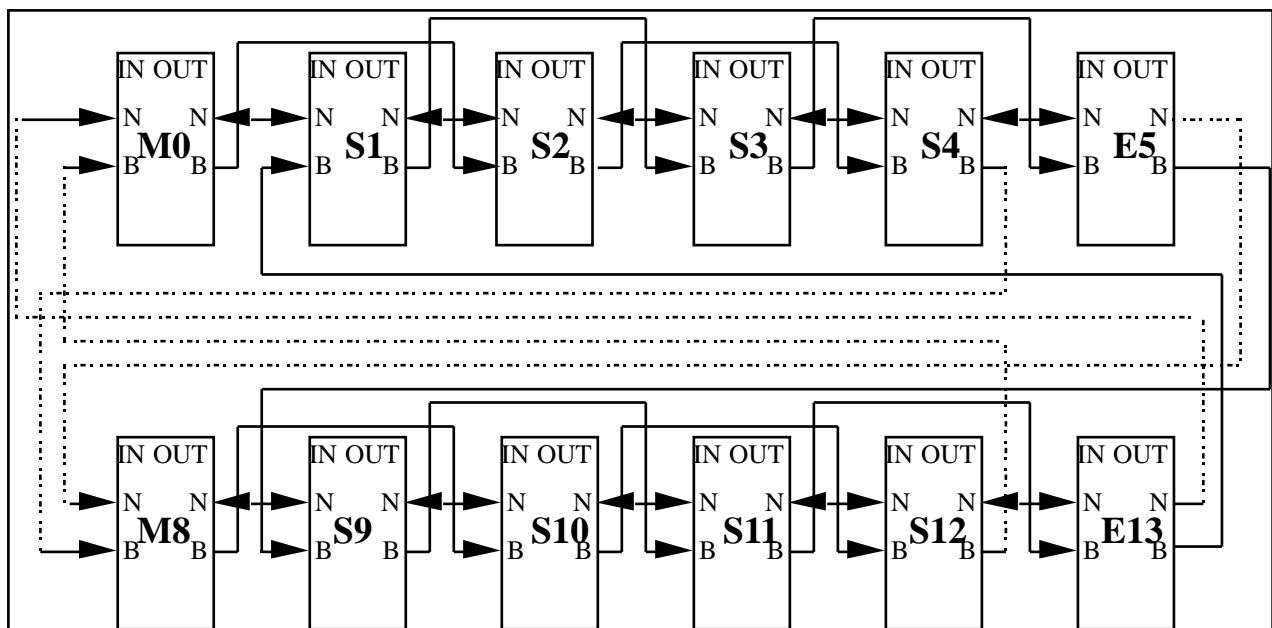
The figure below shows the data and token interconnections on a typical silicon detector module. The module has 6-ABC chips on each side. The LED outputs of 2 of these chips are connected to a fibre-optic interface and are configured to act as Masters in controlling the readout of data from each side of the module. On the diagrams the Master chips are denoted by a “M” and all the other chips are configured to act as slaves as denoted by a “S” or “E” on the diagram.



**Figure 3.19a: Key to symbols used in following Diagrams**

After the receipt of a Level 1 Trigger, the Master chip initiates a readout cycle by sending the pre-amble bits at the start of each data block to the LED driver. It then appends its data bits to the output stream sent to the LED driver. A few clock cycles before the last bit has been sent, it sends a token to the slave chip on it's right. The slave chip on the right responds by sending its data packet to the Master which in turn is appended to the pre-amble and data bits from the Master already sent to the LED driver.

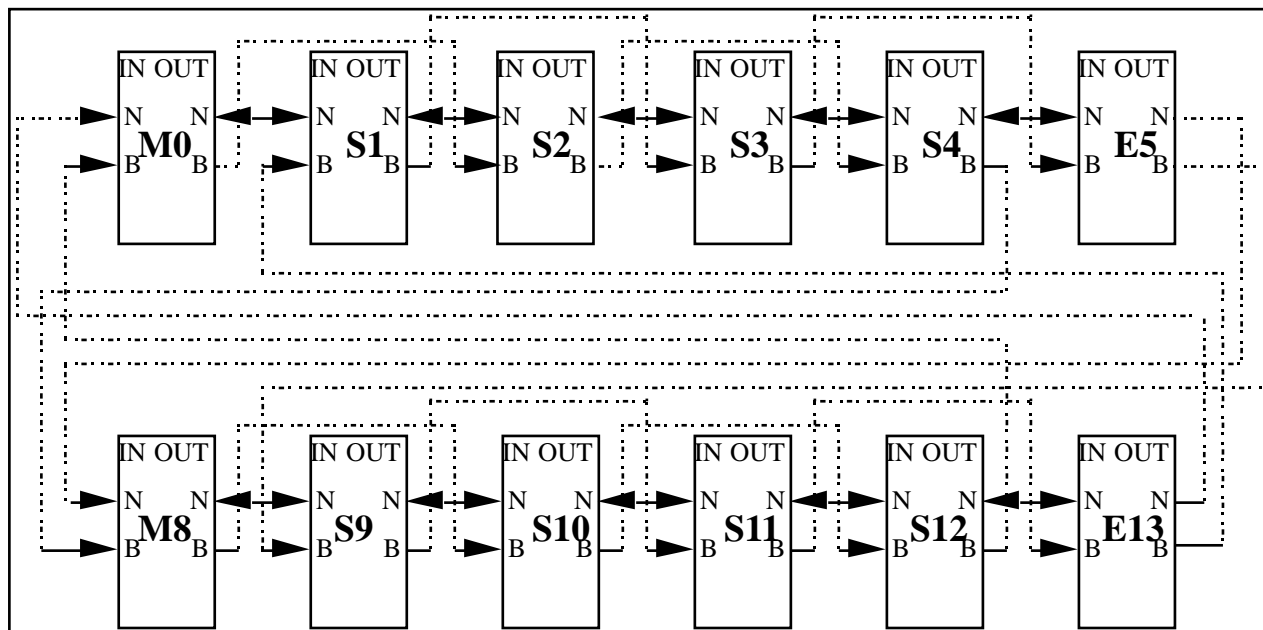
N.B. each chip always sends at least 3 bits of data even if it hasn't found any hit channels in the event being read out.



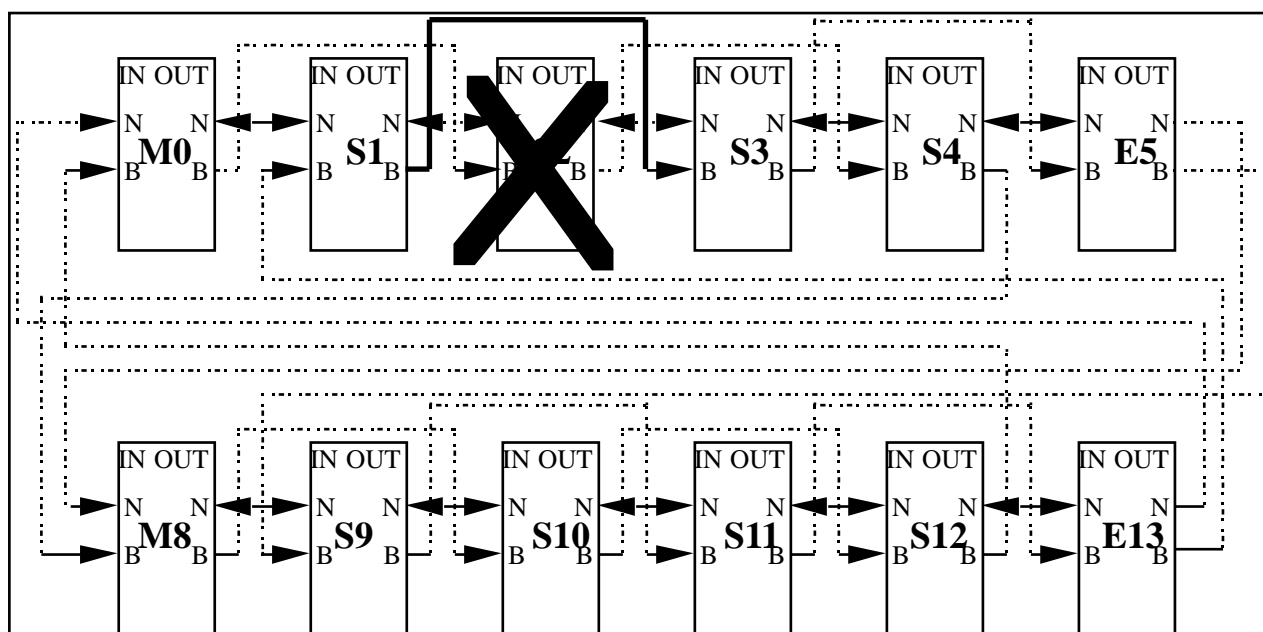
**Figure 3.19b: Diagram Showing the Interconnection of ABC chips on a Silicon Detector Module.**

Once this slave chip has finished sending its data, it also passes on the token to the next chip on the right. The next chip on the right passes its data onto the previous chip on the left which in turn passes it back to the Master chip for transmission to the LED driver. This process continues until the last chip in the chain has sent its data.

A bit is set in the last chip in the chain to inform it that it is the last chip (these chips are shown as 'E' on the diagrams). When this chip has sent its data it appends a trailer to the end of the data stream. While the Master chip is outputting data, it is constantly looking for the trailer pattern which has been carefully chosen to be distinct from the data. Once it finds the trailer pattern, it knows that all the data from the event has been sent and it can start processing the next event.



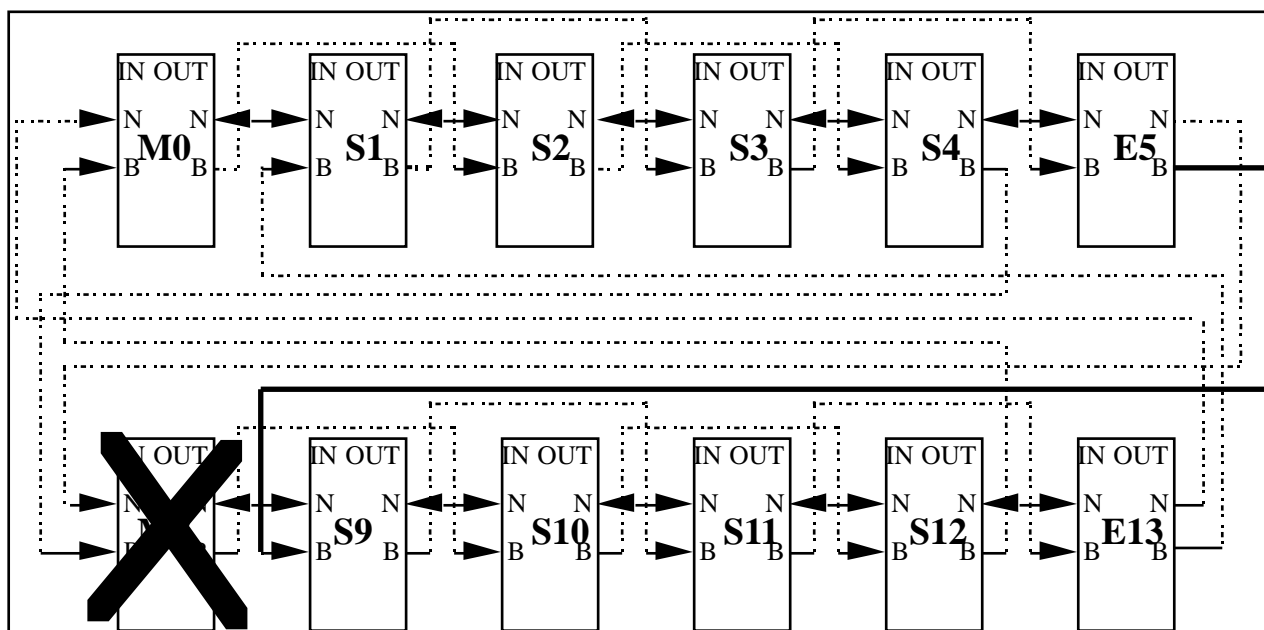
**Figure 3.19c: Diagram Showing the normal flow of Data and Tokens between chips.**  
(Active links are highlighted with solid lines.)



**Figure 3.19d: Diagram Showing the flow of Tokens and Data in the event of the failure of a Slave ABC chip**

In the event of the failure of one of the Slave chips, the previous and next slave chips in the chain are programmed to route their data and tokens around the failed chip. If the last chip in the chain should fail, then the penultimate chip in the chain is programmed to perform the operation of the "End chip".

In the event of the failure of a Master chip in the chain, the data and tokens from the chain with the failed Master chip are routed to the working master chip as shown in the next diagram .



**Figure 3.19e: Diagram Showing the flow of Tokens and Data in the Event of the failure of a Master ABC chip**

### 3.2.23. Default Register Values

On power up, the contents of the configuration register will be set to zero. This results in the following configuration.

Read Out Mode is set to Detector alignment mode

Calibration Mode is disabled

Send\_ID mode is enabled.

tokenin0 and datain0 inputs are enabled

tokenout0 and dataout0 outputs are enabled.

Input test mode is disabled.

Edge Detection Mode disabled

Clock Feed Through Mode is Enabled if the chip is acting as a Master.

Chip will not be configured as the end of a readout chain.

The chip will be configured as a Master if masterB is asserted, else it will be configured as a slave.

#### 3.2.23.1 Master/Slave Selection

The default state of the chip on power up is determined by the state on the masterB input pin. If this pin has been left unconnected or tied high, the chip will power-up as a Slave. If this pin has been tied to ground, the chip will power up as a Master. If the chip is configured as a Master on power up it may be re-configured as a slave. However if the chip has been configured as a slave on power up it may not be re configured as a master.

### 3.2.24. Input/Output Connections

The following the following tables describes the names and function of the various Input/Output connections to the chip.

**Table 3.20: Input Signals**

Name	Function	Type
in<0:127>	Signal Inputs	Current Mode
clk0 & clk1	Clock input	LVDS
clk0B & clk1B	Complement of above signal	LVDS
com0 & com1	Command Input	LVDS
com0B & com1B	Complement of above signal	LVDS
tokenin0 & tokenin1	Token Input	Current Mode
tokenin0B & tokenin1B	Complement of above signal	Current Mode
datain0 & datain1	Data Input	Current Mode
datain0B & datain1B	Complement of above signal	Current Mode
id<5:0>	Geographical address of chip	CMOS
masterB	Sets chip default to master	CMOS
select	Selects clock/command inputs	CMOS
resetB	Resets Chip	CMOS
IDAR	Current refernece for DAC	analog
CA	Input level adjustment	analog
INRH	High level input reference	analog
INRL	Low Level Input reference	analog

**Table 3.21: Output Signals;**

Name	Function	Type
tokenout0 tokenout1	Token Output	Current Mode
tokenout0B tokenout1B	Complement of above	Current Mode
dataout0 dataout1	Data Output	Current Mode
dataout0B dataout1B	Complement of above	Current Mode
datalink	Data Output to data link driver	LVDS
datalinkB	Complement of above	LVDS
CALD0	Calibration Mode Output to CAFE-M Chip	CMOS
CALD1	Calibration Mode Output to CAFE-M Chip	CMOS
CALSP	Positive Calibration Strobe Output to CAFE-M Chip	Open drain
CALSN	Complement of above	Open drain
ITH	Input threshold output to CAFE-M chip	Analogue
CALI	Calibration Threshold output to CAFE-M chip	Analogue

### 3.2.25. Electrical Specifications

#### 3.2.25.1 Supply Voltage

4.0 volts  $\pm$  5%.

For the on chip power-on reset circuitry to operate correctly the power supply must be ramped up to 90% of its final value in less than 10ms.

#### 3.2.25.2 Power Consumption

This depends upon he mode of operation of the chip . In slave mode it is expected to be less than or equal to 64mW when operating at a 1% occupancy and 100Khz Level 1 trigger rate. In Master the power consumption will be approximately 164mW.

### 3.2.25.3 Power Supply Connections

The chip has a total of 8 power supply connection pins. Two of these pins are on the row of pads next to the chips input pads. These pads are intended to provide a return path for the switching currents in the input stage of the chip. These pads should be connected directly to the corresponding pads on the CAFE-M chip. The set of 2 pairs of power connections on the right hand side of the chip provide all the power to the digital sections of the chip. These pads should be connected to the digital power supply for the chip. The remaining pair of pads are intended to supply power just to the analogue sections of the chip. These pads should be connected to the analogue supply to the chip.

### 3.2.25.3 Input /Output Levels

**Table 3.22: Input Levels for Hit Inputs**

Parameter	Minimum	Typical	Maximum
Low Level Input Current $I_{IL}$		50 $\mu$	
High Level Input Current $I_{IH}$		200 $\mu$	

**Table 3.23: Output Levels for Calibration Code Outputs**

Parameter	Minimum	Typical	Maximum
Low Level Voltage $V_{OL}$	VSS		0.5 V
High Level Voltage $V_{OH}$	VDD - 0.5V		VDD

**Table 3.24: Output Levels for Calibration Strobe Outputs**

Parameter	Minimum	Typical	Maximum
Low Level Output Current $I_{OL}$		11 $\mu$ A	18 $\mu$ A
High Level Output Current $I_{OH}$	180 $\mu$	242 $\mu$	

Load capacitance for all cases is less than 1pF.

**Table 3.25: Input Levels for LVDS Inputs (Clock, Control)**

Parameter	Conditions	Minimum	Maximum	Units
Input Voltage Range $V_i$	$V_{gpd} \leq 950\text{mV}$	0	2400	mV
Input Voltage Common mode $V_{icm}$	$V_{gpd} \leq 950\text{mV}$	50	2350	mV
Differential high input threshold $+V_{idth}$	$V_{gpd} \leq 950\text{mV}$		100	mV
Differential high input threshold $-V_{idth}$	$R_{load} = 100 \pm 1\%$	-100		mV
Threshold hysteresis	$(+V_{id}) - (V_{id})$	25		mV
Receiver input impedance		100 k		

N.B. No internal terminating resistor is built into these inputs and consequently an external resistor terminated resistor is required.

**Table 3.26: Input Levels for Token and Data Inputs (Token\_in, Data\_in)**

---

Parameter	Minimum	Typical	Maximum
Low Level Input Voltage $V_{IL}$		$V_{DD}/2 - 100\text{mV}$	
High Level Input Voltage $V_{IH}$		$V_{DD}/2 + 100\text{mV}$	
Receiver input impedance		120 ohms	



**Table 3.27: Output Levels for Token and Data Outputs (Token\_Out, Data\_out)**

Parameter	Conditions	Minimum	Typical	Maximum
Output Voltage Low VOL	$C_L=50\text{pf}$ $R_{load}=125\text{ohms}$		$V_{DD}/2 - 100\text{mV}$	
Output Voltage High VOH	$C_L=50\text{pf}$ $R_{load}=125\text{ohms}$		$V_{DD}/2 - 100\text{mV}$	
Output Differential Voltage	$C_L=50\text{pf}$ $R_{load}=125\text{ohms}$		200mV	

**Table 3.28: Output Levels for LED Outputs**

Parameter		Minimum	Maximum	Units
Output Voltage low VOL	$R_{load} = 100 \pm 1\%$	1000		mV
Output Voltage High VOH	$R_{load} = 100 \pm 1\%$		1400	mV
Output offset Voltage	$R_{load} = 100 \pm 1\%$	1125	1275	mV
Output Differential Voltage	$R_{load} = 100 \pm 1\%$	250	400	mV
Output impedance	$I_{load} = 2\text{mA to } 3\text{mA}$	40	280	

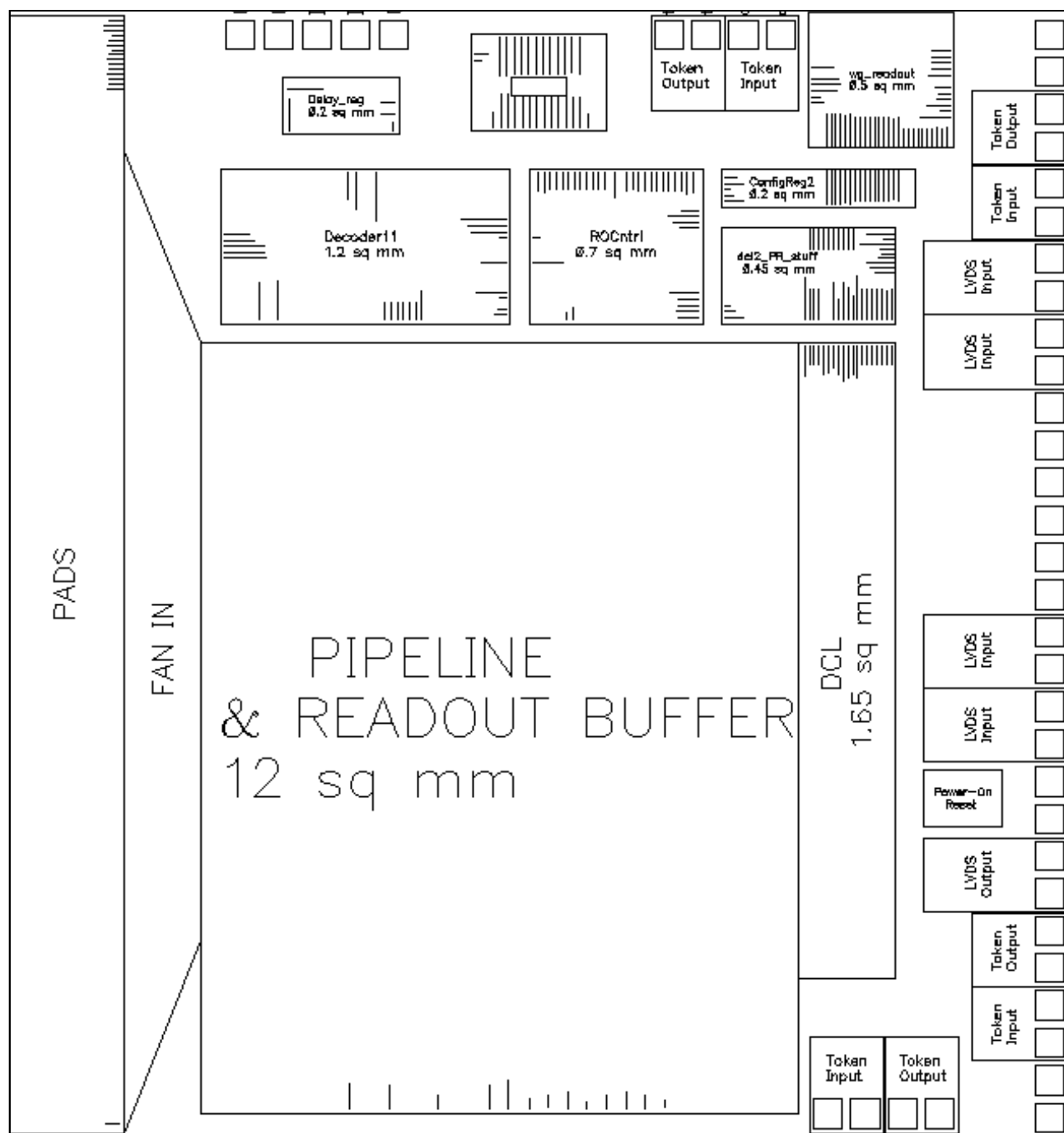
**Table 3.29: DAC Input and Output levels**

Parameter	Minimum	Typical	Maximum
IREF DAC Reference current	-270 $\mu$	-300 $\mu$	-330 $\mu$
CALDI	$-(i_{ref}/256) * num - 10\%$	$-(i_{ref}/256) * num$	$-(i_{ref}/256) * num + 10\%$
IDAC	$(i_{ref}/256) * num - 10\%$	$(i_{ref}/256) * num$	$(i_{ref}/256) * num + 10\%$

Num is the value loaded into the DAC register

### 3.2.27. Physical Requirements

The die size is 5.5 mm x 6.2 mm..

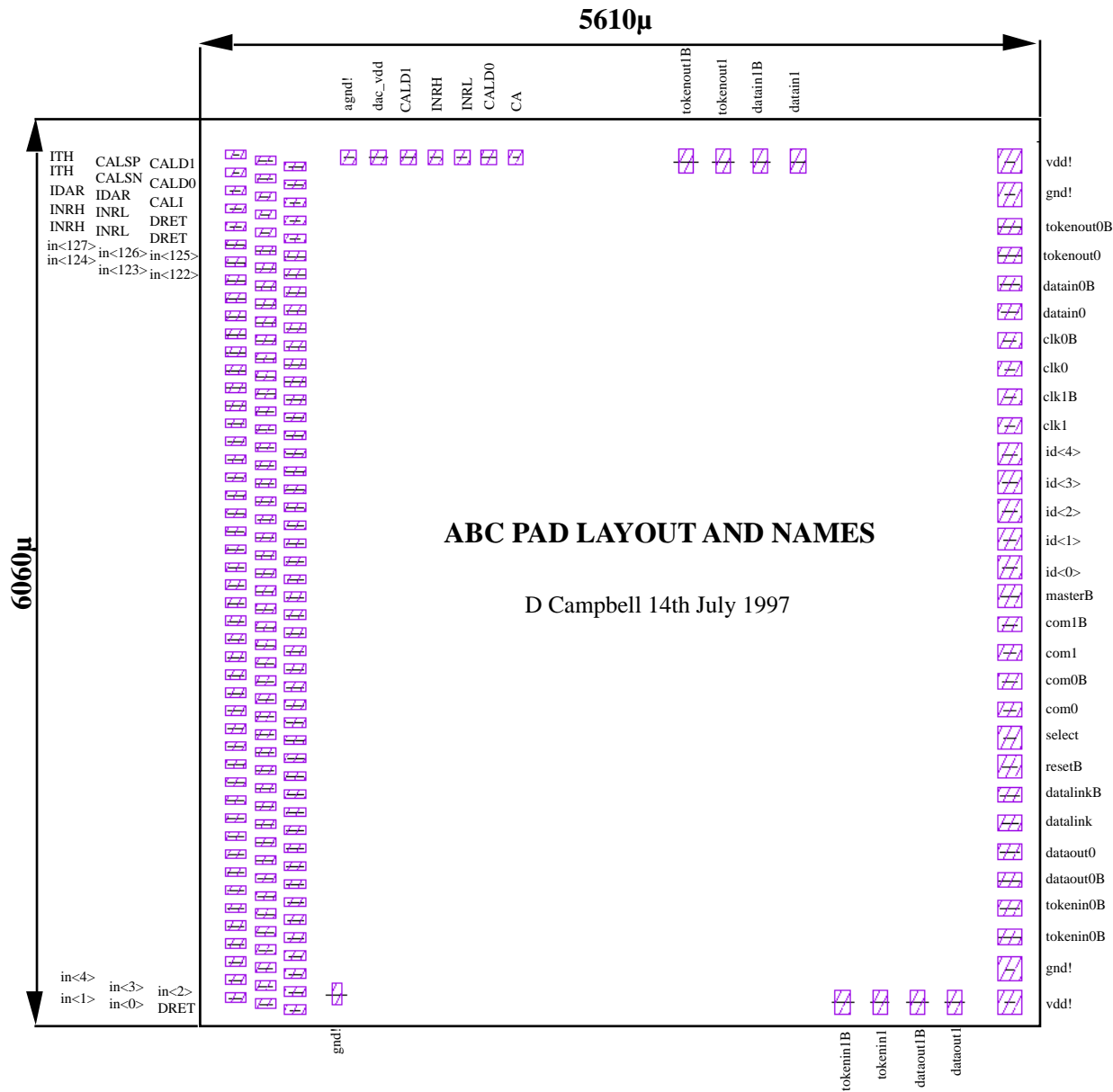


**Figure 3.20: Chip Layout**

Note Channel 0 is at the lower edge of this figure. The other channels, addressed 1 through 127, are located consecutively from bottom to top. This numbering convention affects which channels are tested by the binary encoded channel address lines, calmode(1:0), to the Input Level Translators (See section 3.2.1.) and the channel addresses which become part of the Physics Data reported by the chip (See section 3.2.17.). This also corresponds to CAFE-M pad layout.

3.2.25.1 Bond Pad Arrangement

The input pads to the chip and all others which interface to the CAFE-M chip have been arranged on a pitch so as to enable the Binary Readout Chip to be wire bonded directly to the CAFE-M chip.



Notes: Pad side view of chip shown, origin is at bottom left hand corner.  
Dimensions shown are from the edge of the scribe-lane ("street")  
Chip thicknes is 525 μ

Table 3.30: Bond Pad Connections

Pad Name	Pad Centre (x)	Pad Centre (y)	Pad Size (x)	Pad Size (y)	Pad Name	Pad Centre (x)	Pad Centre (y)	Pad Size (x)	Pad Size (y)
ITH	200	5720	140	60	CALD1	600	5640	140	60
ITH	200	5600	140	60	CALD0	600	5520	140	60
IDAR	200	5480	140	60	CALI	600	5400	140	60
INRH	200	5360	140	60	DRET	600	5280	140	60
INRH	200	5240	140	60	DRET	600	5160	140	60
in<127>	200	5120	140	60	in<125>	600	5040	140	60
in<124>	200	5000	140	60	in<122>	600	4920	140	60
in<121>	200	4880	140	60	in<119>	600	4800	140	60
in<118>	200	4760	140	60	in<116>	600	4680	140	60
in<115>	200	4640	140	60	in<113>	600	4560	140	60
in<112>	200	4520	140	60	in<110>	600	4440	140	60
in<109>	200	4400	140	60	in<107>	600	4320	140	60
in<106>	200	4280	140	60	in<104>	600	4200	140	60
in<103>	200	4160	140	60	in<101>	600	4080	140	60
in<100>	200	4040	140	60	in<98>	600	3960	140	60
in<97>	200	3920	140	60	in<95>	600	3840	140	60
in<94>	200	3800	140	60	in<92>	600	3720	140	60
in<91>	200	3680	140	60	in<89>	600	3600	140	60
in<88>	200	3560	140	60	in<86>	600	3480	140	60
in<85>	200	3440	140	60	in<83>	600	3360	140	60
in<82>	200	3320	140	60	in<80>	600	3240	140	60
in<79>	200	3200	140	60	in<77>	600	3120	140	60
in<76>	200	3080	140	60	in<74>	600	3000	140	60
in<73>	200	2960	140	60	in<71>	600	2880	140	60
in<70>	200	2840	140	60	in<68>	600	2760	140	60
in<67>	200	2720	140	60	in<65>	600	2640	140	60
in<64>	200	2600	140	60	in<62>	600	2520	140	60
in<61>	200	2480	140	60	in<59>	600	2400	140	60
in<58>	200	2360	140	60	in<56>	600	2280	140	60
in<55>	200	2240	140	60	in<53>	600	2160	140	60
in<52>	200	2120	140	60	in<50>	600	2040	140	60
in<49>	200	2000	140	60	in<47>	600	1920	140	60
in<46>	200	1880	140	60	in<44>	600	1800	140	60
in<43>	200	1760	140	60	in<41>	600	1680	140	60
in<40>	200	1640	140	60	in<38>	600	1560	140	60
in<37>	200	1520	140	60	in<35>	600	1440	140	60
in<34>	200	1400	140	60	in<32>	600	1320	140	60
in<31>	200	1280	140	60	in<29>	600	1200	140	60
in<28>	200	1160	140	60	in<26>	600	1080	140	60
in<25>	200	1040	140	60	in<23>	600	960	140	60
in<22>	200	920	140	60	in<20>	600	840	140	60
in<19>	200	800	140	60	in<17>	600	720	140	60
in<16>	200	680	140	60	in<14>	600	600	140	60
in<13>	200	560	140	60	in<11>	600	480	140	60
in<10>	200	440	140	60	in<8>	600	360	140	60
in<7>	200	320	140	60	in<5>	600	240	140	60
in<4>	200	200	140	60	in<2>	600	120	140	60
in<1>	200	80	140	60	DRET	600	0	140	60

Pad Name	Pad Centre (x)	Pad Centre (y)	Pad Size (x)	Pad Size (y)	Pad Name	Pad Centre (x)	Pad Centre (y)	Pad Size (x)	Pad Size (y)
CALSP	400	5680	140	60	gnd!	877	103	60	140
CALSN	400	5560	140	60	agnd!	957	5702	100	100
IDAR	400	5440	140	60	dac_vdd	1157	5702	100	100
INRL	400	5320	140	60	CALD1	1357	5702	100	100
INRL	400	5200	140	60	INRH	1537	5702	100	100
in<126>	400	5080	140	60	INRL	1717	5702	100	100
in<123>	400	4960	140	60	CALD0	1897	5702	100	100
in<120>	400	4840	140	60	CA	2077	5702	100	100
in<117>	400	4720	140	60	tokenout1B	3214	5674	100	160
in<114>	400	4600	140	60	tokenout1	3464	5674	100	160
in<111>	400	4480	140	60	datain1B	3714	5674	100	160
in<108>	400	4360	140	60	datain1	3964	5674	100	160
in<105>	400	4240	140	60	tokenin1B	4264	50	100	160
in<102>	400	4120	140	60	tokenin1	4514	50	100	160
in<99>	400	4000	140	60	dataout1B	4764	50	100	160
in<96>	400	3880	140	60	dataout1	5014	50	100	160
in<93>	400	3760	140	60	vdd!	5384	5674	160	160
in<90>	400	3640	140	60	gnd!	5384	5454	160	160
in<87>	400	3520	140	60	tokenout0B	5384	5237	160	100
in<84>	400	3400	140	60	tokenout0	5384	5047	160	100
in<81>	400	3280	140	60	datain0B	5384	4857	160	100
in<78>	400	3160	140	60	datain0	5384	4667	160	100
in<75>	400	3040	140	60	clk0B	5384	4477	160	100
in<72>	400	2920	140	60	clk0	5384	4287	160	100
in<69>	400	2800	140	60	clk1B	5384	4097	160	100
in<66>	400	2680	140	60	clk1	5384	3907	160	100
in<63>	400	2560	140	60	id<4>	5384	3717	160	150
in<60>	400	2440	140	60	id<3>	5384	3527	160	150
in<57>	400	2320	140	60	id<2>	5384	3337	160	150
in<54>	400	2200	140	60	id<1>	5384	3147	160	150
in<51>	400	2080	140	60	id<0>	5384	2957	160	150
in<48>	400	1960	140	60	masterB	5384	2767	160	150
in<45>	400	1840	140	60	com1B	5384	2577	160	100
in<42>	400	1720	140	60	com1	5384	2387	160	100
in<39>	400	1600	140	60	com0B	5384	2197	160	100
in<36>	400	1480	140	60	com0	5384	2007	160	100
in<33>	400	1360	140	60	select	5384	1817	160	150
in<30>	400	1240	140	60	resetB	5384	1627	160	150
in<27>	400	1120	140	60	datalinkB	5384	1437	160	100
in<24>	400	1000	140	60	datalink	5384	1247	160	100
in<21>	400	880	140	60	dataout0	5384	1057	160	100
in<18>	400	760	140	60	dataout0B	5384	867	160	100
in<15>	400	640	140	60	tokenin0	5384	677	160	100
in<12>	400	520	140	60	tokenin0B	5384	487	160	100
in<9>	400	400	140	60	gnd!	5384	270	160	160
in<6>	400	280	140	60	vdd!	5384	50	160	160
in<3>	400	160	140	60					
in<0>	400	40	140	60					