DRAFT
ATLAS Level-1 Trigger: Front End ASIC; Users Manual

Alexander Mass
Institut für Hochenergiephysik
Universität Heidelberg

Abstract: *Technical information for integrating the FeAsic into a readout environment is given. After a definition of I/O signals, the sequential programming interface with its programming and control tokens are described. Finally the readout modes of the FeAsic are explained.*

1 Introduction

The Front End Asic (FeAsic) has been designed in the ASIC-Lab at the Institute of High Energy Physics in Heidelberg. The Chip is part of the digitization and readout system for the ATLAS level-1 calorimeter trigger, which is described in detail in [1]. The current version of the FeAsic is a prototype and will be used to demonstrate the technical feasibility of the high speed readout concept. Important tasks of the FeAsic are the preprocessing of the digitized calorimeter data for 1 trigger tower and the readout of level-1 accepted event data. The preprocessing comprises transverse energy calibration, pedestal subtraction and identification of the LHC bunch crossing number.

The design work was based on the following assumptions: The calorimeter energy is measured in trigger towers. An 8 bit FADC digitizes the energy data with an LSB equivalent to 1 GeV. The FADC and the FeAsic is clocked with the bunch crossing frequency of 40 MHz. The preprocessing part was designed using the pipeline technique so that at every time slice a completely set of processed data can be forwarded to the calorimeter processing unit. The readout has to work at level-1 trigger rates up to 100kHz.

The block diagram of the FeAsic is shown in figure 1. The transverse energy and pedestal subtraction is realized by means of a look up table with 8 bit address and 10 bit data bus. The look up table allows to linearize analog compressed signals in order to increase the effective dynamic energy range from 8 to 10 bit. The larger dynamic range is probably required for bunch crossing identification (BCID), which has to work properly on 1 TeV electron signals as well as on small energies in the few GeV range. The BCID circuit determines the time slice with the maximum of the calorimeter pulse. The energy of all other time slices is set to zero. The circuit is implemented as FIR-filter with fixed coefficients combined with a peak finder. The BCID unit is described in more detail in [2]. The dynamic range of the subsequent trigger logic is limited to 8 bit data words. The FeAsic is clipping energies exceeding the 8 bit limit.

A dual port memory is used to store data until the decision of the level-1 trigger. It is used as synchronous FIFO. Two conceptional different readout modes were implemented. In the "stop mode" the level-1 accept signal stops the scrolling memory and the current data as well as the following data words can be directly copied to the readout port. In the fast "readout mode" data from 1, 4 or 8 time slices are copied into a derandomizer buffer. This second buffer is foreseen to compensate statistical fluctuations of level-1 accepts.

A sequential readout concept was chosen to limit the number of I/O Pads on the FeAsic. The sequential port has separate input and output pads, thus allowing to daisy chain several FeAsics. The port is used for readout as well as for controlling and programming.

As an additional feature a playback memory was integrated to serve as data source for tests of the FeAsic. It can also be used as data source for the subsequent trigger logic.

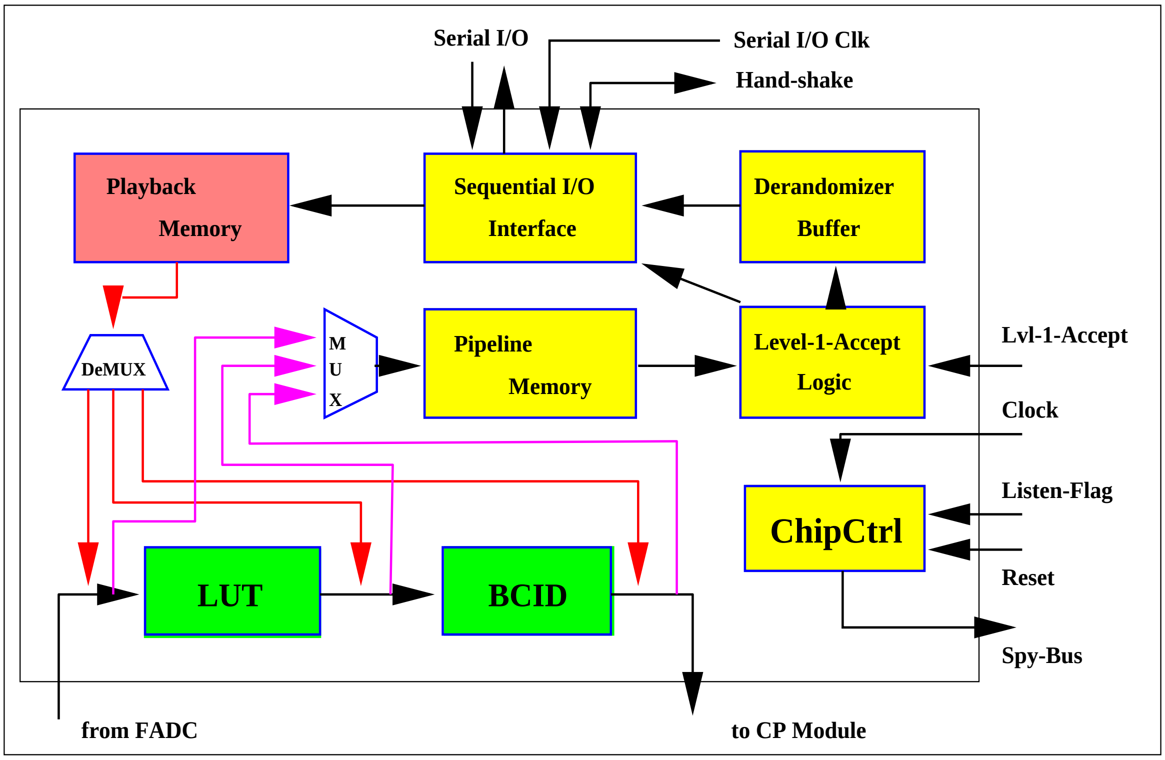


Abbildung 1: Block diagram of the basic components integrated on the FeAsic

The FeAsic design was described in the VERILOG Hardware Description Language. It was implemented in ES2 0.7 μm CMOS process. The ES2 software includes a memory macro cell generator for RAMs, dual ported RAMs and asynchronous FIFOs. The design consists of about 3000 gates and 1.2 kByte of memory and is implemented with 80000 transistors on a 16.4 mm^2 silicon die.

2 I/O signals of the FeAsic

All input signals should be driven with CMOS or TTL voltage levels. The input pads receive signals with a non inverting Schmitt trigger circuit. The input pads have an input capacitance of 2.45 pF. The output pads generate TTL signals and are capable to drive a current of up to 4 mA. ES2 corner power pads are used for the supply of the output pads. The voltage drop across the connections between the supply pads will be less than 30 mV from the full TTL amplitude.

The following table lists all I/O signals. The numbering starts from the upper most pin on the left side proceeding clock wise around the chip.

Pin #	Name	I/O	Function
64	*Lvl-1-Accept*	I	level 1 trigger accept signal
65	*Clock 40MHz*	I	system clock 40 MHz
66-5	*FadcInp[0:7]*	I	input from FADC
17-24	*ToCPM[0:7]*	O	Level-1 output after BCID
63	*ListenFlag*	I	select for programming or readout
15	*SyncReset*	I	reset for address counter
16	*StRegReset*	I	reset for state registers
36	*SerialInp*	I	input to sequential port
35	*SerialOut*	O	output to sequential port
33	*SerialClk*	I	clock for sequential port
34	*SerialAck*	I	acknowledge from controller
32	*SerialRdy*	O	ready for next data
39-37	*SpyBusCtrl[0:2]*	I	select for spy bus page
51-58	*SpyBus[0:7]*	O	spy bus data
30	*BistTest*	I	test pattern input for BIST
31	*BistClk*	I	BIST clock signal
47	*BistResPipeline*	O	BIST result from pipeline
48	*BistResLut*	O	BIST result from lut
49	*BistResPlayb*	O	BIST result from playback
50	*BistResDerand*	O	BIST result from derandomizer
13,14,29,40	*CoreGnd*	pwr	0V for core cells
6,62	*CoreVdd*	pwr	+5V for core cells
46	*Vdd-Macro-1*	pwr	+5V for LUT RAM & Playback RAM
12	*Vdd-Macro-2*	pwr	+5V for Pipeline & Derandomizer
7,45	*PadGnd*	pwr	0V for pad cell
25,61	*PadVdd*	pwr	+5V for pad cells

2.1 40 MHz clock and Level-1 accept signal

In the demonstrator environment the FeAsic will receive trigger related signals from the Trigger Controller Module (TCM). The TCM emulates the ATLAS Trigger Timing Control (TTC) system. It distributes the 40 MHz system clock *SYSCLK* and a level-1 accept flag STOP.

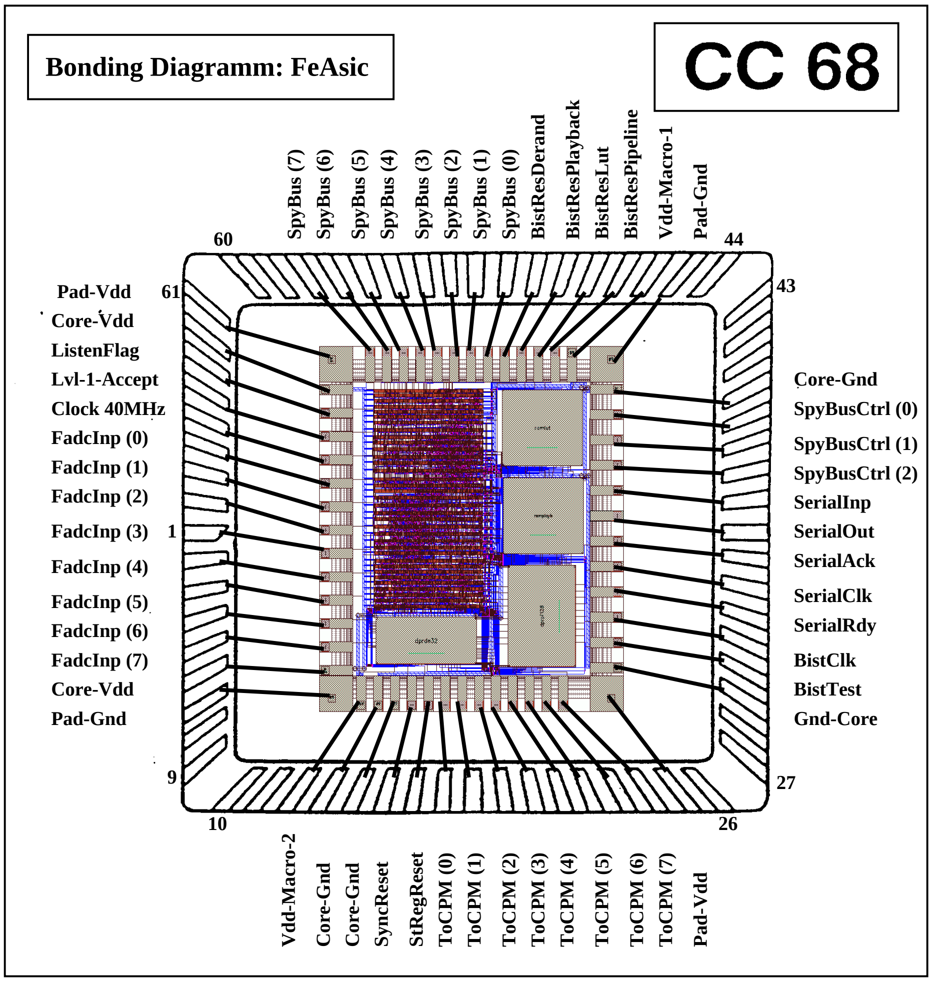
The FeAsic uses positive and negative edge triggered latches. This requires that the duty cycle of the clock signal is close to 1:1. The clock signal is fanned out in a tree structure. Clock signal from every level are used in a circuit design with positive clock skew. Ie. the clock signal for latches come late for devices in the beginning of the processing pipeline and early at the end of the pipeline. Without wire delays the clock signal needs 1.6 nsec to propagate from the input pad to the first latch in the processing pipeline. All input signals (*FadcInp* & *Lvl-1-Accept*) are triggered on the negative edge, they are latched 14.1 nsec after the positive clock edge has arrived.

The FeAsic allows to couple two data sources to the Cluster Processing Module output *ToCPM*; either data from *FadcInp* or a test pattern from the playback memory. The bus switch is causing a delay. The output signal becomes valid 3.4 nsec after the positive edge of the clock signal.

2.2 Control signals

The FeAsic is operated in two different modes which are controlled by the *ListenFlag*. *ListenFlag* low means the chip works in one of the four readout modes. In the other case, the FeAsic waits for programming tokens, which are received via the sequential I/O interface.

The FeAsic responds to reset signals only if the *ListenFlag* is set high. The ASIC has two independent



reset signals. The *StRegReset* signal is responsible for a reset of all state machines and status registers[1]. The *SyncReset* signal synchronizes the operation of all FeAsics in a readout system. It resets the memory address counters for the playback and pipeline memory and initializes the derandomizer buffer.

The *ListenFlag* together with the *Lvl-1-Accept* signal controls also the operation of the playback memory. The playback operation starts on the negative edge of the *ListenFlag*. The Playback memory is designed as scrolling memory. It can only be stopped in the case of stop mode readout. In this case it can be restarted by toggling *ListenFlag*.

2.3 Sequential I/O Interface

The sequential port of the FeAsic serves two purposes. In the readout mode it sends data from level-1 accepted events to the readout controller. In the listen mode it receives tokens for programming of status registers and memories.

The advantage of a sequential interface is the limited number of I/O pads required. The I/O pads should not determined the size and the cost of an ASIC. The following arguments describe the features of the sequential interface which make it suitable for the level-1 digitization system.

- After the level-1 trigger the event rate is reduced to 100 kHz. Assuming a readout system, which

[1] The FeAsic has two types of control registers a.) registers integrated into state machines and b.) registers determining the operation mode of the FeAsic

is based on the 40 MHz bunch crossing clock, 400 clock cycles are available for the readout of an event in average.

- The sequential I/O interface has an independent clock signal *SerialClk*. Therefore it is possible to increase the data throughput by increasing the clock frequency.
- For daisy chaining several FeAsics it is necessary to foresee a separate input and output pads in the interface. This allows a compact readout scheme. Data from sets of FeAsics is shifted into a readout controller device.
- The daisy chain can be used as data output but also as input channel for control purposes. The protocol is explained in chapters 3 & 4.

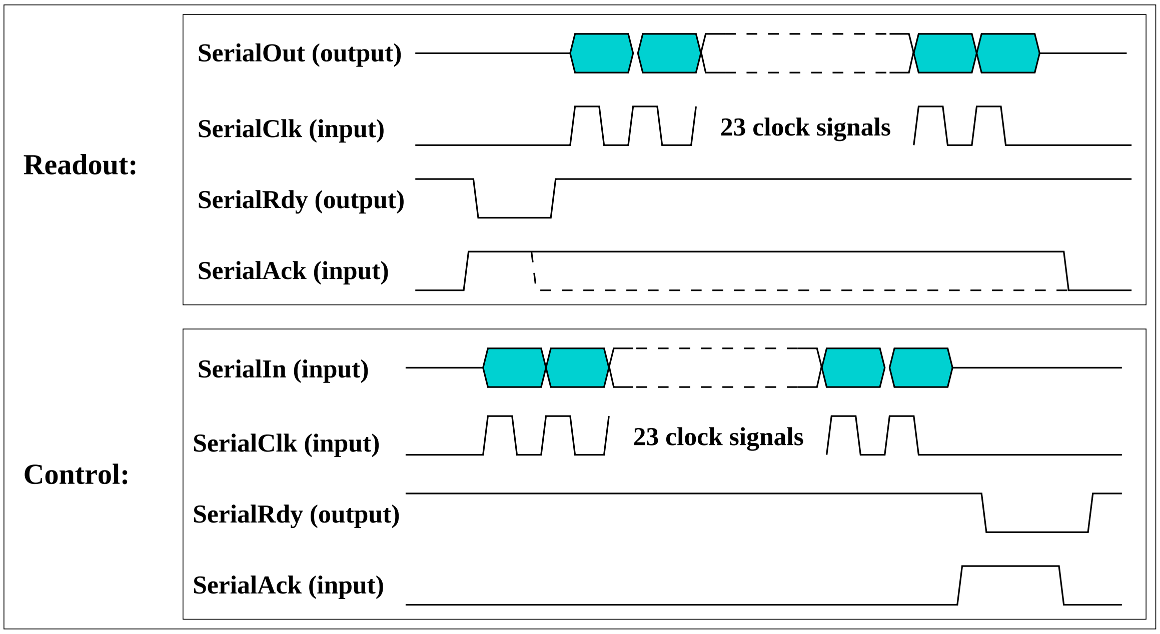


Abbildung 2: Timing diagram of the sequential interface

The level-1 accept signals are asynchronous and it is not excluded that a second accept signal appears during a sequential readout cycle. It is therefore mandatory to include a second buffer to compensate fluctuations of the level-1 accept signals. The derandomizer buffer is realized as asynchronous FIFO with 24 data slots. The size of the derandomizer was chosen to be sufficient for a dead time free readout, if only 1 data sample per event has to be read out.

Two handshake signals are required for controlling the sequential I/O interface. Figure 2 shows an output and an input sequence. A 23 bit shift register is used as buffer between the *SerialInp* and the *SerialOut* port. This shift register is accessible in two ways. In the serial mode the data is shifted with every *SerialClk* signal. In the parallel mode the FeAsic can read to or write from the register. A state machine controls the access to the register. In case of *SerialRdy* low the FeAsic has parallel access to the register. A sequential operation is possible if *SerialRdy* is high and *SerialAck* is low. In the readout mode the FeAsic waits for the *SerialAck* signal. After a positive edge of *SerialAck* the FeAsic either inspects the control token or copies data to be read out to the shift register. As soon as the internal operations are finished the *SerialRdy* signal is set high.

In the continuous readout mode data words are only copied to sequential I/O interface if data is stored in the derandomizer buffer. In the *Listen Mode* control tokens can be shifted into the I/O register at any time. In this mode the data is interpreted by the control unit after setting of *SerialAck*. The *SerialAck* signal has to be reseted before the FeAsic is ready for the next I/O operation.

2.4 Spy Bus System

In case of unsatisfactory operation of the FeAsic it is desirable to look directly on important signals and how they change with the system clock. This excludes the sequential I/O interface for such operations. An 8 bit wide bus *SpyBus*[0:7] was integrated for this purpose. In order to increase the number of accessible signals 8 different pages can be multiplexed to the spy bus output. A dedicated page can be selected by *SpyBusCtrl [0:2]*. The goal was to make most of the state machine, status registers and memory address pointers visible to the exterior.

It follows a list and short explanation of the accessible signals.

Page	Bits	Signal Name	Meaning
0	(0:2)	*DerandState*	state of derandomizer buffer
	(3:4)	*LossCounter*	counter to monitor overflows of the derandomizer buffer
	(5)	*PipelineSyncFlag*	checks if Pipeline memory address are running synchronous
	(6)	*StopTrigger*	shows status of the Stop readout mode
	(7)	*PipelineWrtAddr*(6)	part of the write address of the pipeline memory
1	(0:7)	*Serial2SpyBus*	the first 8 bits of the sequential I/O interface
2	(0:7)	*PipelineReadAddr*	read pointer for the pipeline memory
3	(0:1)	*PipelineWrtAddr*(0:1)	2 bits of the write pointer for the pipeline memory
	(2:5)	*DataSetCounter*	counter for # of time slices to be copied to the derandomizer buffer
	(6:7)	*StateReadout*	state machine register for controlling the readout
4	(0:7)	*PlaybackReadAddr*	address pointer of the playback memory
5	(0:7)	*PipelineDelay*	status register for the pipeline delay
6	(0:1)	*BcidPeakCtrl*	status register for BCID peak finder
	(2:4)	*StateSerialIo*	state machine register of sequential I/O interface
	(5)	*LutOperation*	status register for the LUT by pass flag
	(6:7)	*StateInput*	state machine register for the interpretation of input data
7	(0:1)	*SourceMux*	status register for data source (FADC or playback memory)
	(2:3)	*CaptureMux*	status register for controlling the input to the pipeline
	(4:5)	*ReadoutMode*	status register that determines the readout mode
	(6:7)	*BcidSlowCtrl*	status register for the BCID operation

2.5 Build in Self Test for the Memories

Two aspects render the implementation of memories on the FeAsic as the most delicate item. The memories use by far the most transistors on the die. This results in a large risk of failures during the production process. A second risk is caused by our application for the memories, which are accessed every clock cycle. These two problems are common for many designs. To improve the diagnostic opportunities ES2-library has an option to include a self test circuit into the memories. The self test circuit provide an independent path from the exterior to the memory and back to an output pad. Testing the memory with the self test circuit allows to distinguish between design and production faults.

The build in self test (BIST) requires 2 input and 1 output pad. The memories are clocked during the test with the signal *BistClk*. ES2 delivers a test pattern generator program which allows to generate a sequential bit stream, which can be fed into the test input channel *BistTest*. The *BistClk* and *BistTest* signals are fanned out to the 4 memories. Each memory has a separate BIST result output: *BistResPipeline*, *BistResLut*, *BistResPipeline* and *BistResDerand*.

3 Controlling the FeAsic

The FeAsic has a variety of features that has to be configured for the different run or debugging modes. The commands are received via the sequential I/O interface. Each command consists of a command token (the first 5 bits) and additional data if required. The FeAsic expects these commands as soon as the *ListenFlag* is set. The *SerialAck* signal triggers the interpretation of the commands. The *SerialRdy* signal is kept low as long as the interpretation last.

3.1 Load Operation of Look up Table and Playback Memory

Both memories have 8 bit address and 10 bit data words. The format of the sequential data word is as follows: The control token is ’b10010 (LSB last) for the loading of the look up table and ’b10011 for loading of the playback memory. The next 8 bits are reserved for the address followed by the 10 bit data word.

3.2 Configuration of the Level-1 processing pipeline

Four status registers define the operation of the level-1 trigger processing unit. After a state register reset (*StRegReset*) those registers are loaded with default values. The following table lists the tokens to program the control registers and gives a short description.

The FeAsic can work with external or internal data source, as shown in figure 1. The external data source is the default setting and may be selected by sending the token *SrcFadc*. The internal data source is the playback memory. The user can select between 3 bus switches to couple data into the processing pipeline. The bus switches are located before the look up table, before the BCID unit and after the BCID unit. In the last mode the playback memory serves directly as data source for the output and can be used as data server for the subsequent trigger logic.

The look up table is bypassed by default. It should be loaded before the Bypass mode is switched off.

The BCID circuit stores the digitized data in a shift register until a FIR filter and a peak finder have detected a maximum of in the amplitude sequence. The BCID unit has 4 operation modes. Besides the standard operation mode in which the BCID unit extracts the maximum of a pulse train, it is possible to let all data words pass the filter. It is also possible to exchange the stored amplitude values with the weighted sum of the FIR-filter.

Token	Default	Name	Meaning
’b00100	x	*SrcFadc*	data from FADC
’b00101		*SrcPbBeforeLut*	data from playback memory before look up table
’b00110		*SrcPbBeforeBcid*	data from playback memory before BCID
’b00111		*SrcPbAfterBcid*	data from playback memory after BCID
’b11100	x	*LutByPassOn*	enable bypass of look up table
’b11101		*LutByPassOff*	disable bypass of look up table
’b10100	x	*BcidOperation*	standard operation of BCID unit
’b10101		*BcidByPass*	bypass of BCID unit
’b10110		*BcidFirMax*	output of maximum FIR filter value
’b10111		*BcidFirFilter*	output of FIR filter values
’b11000		*BcidSynNoEqual*	peak finder Fir(t+1)<Fir(t)>Fir(t-1)
’b11001		*BcidAsymGEG*	peak finder Fir(t+1)<=Fir(t)>Fir(t-1)
’b11010	x	*BcidAsymGGE*	peak finder Fir(t+1)<Fir(t)=>Fir(t-1)
’b11011		*BcidSymEqual*	peak finder Fir(t+1)<=Fir(t)=>Fir(t-1)

It is not clear until yet which peak finder condition is most suitable for our application. All four possible relations are integrated and can be selected.

3.3 Configuration of the Readout Components

The FeAsic has four readout modes which are explained in more detail in chapter 4. The control tokens are again listed in the table below.

The pipeline memory is implemented as scrolling memory using a dual ported RAM with 7 bit address and 10 bit data words. The read and write address are running synchronously. The difference between the 2 address pointers determines the temporal duration of data words in the pipeline. The pointer difference is programmed with the *LoadPipelineDelay* token. The token is followed by a 7 bit value for the difference. On a *SyncReset* the read address is set to pipeline address 0 while the write address is set to value defined by *LoadPipelineDelay*. The default value is a delay of 80 clock cycles.

A multiplexer is used to store data from 1 of 3 locations of the processing pipeline in the pipeline memory, see again figure 1. Under default condition data are sent to the digital pipeline.

Token	Default	Name	Meaning
'b01100	x	*ReadoutStop*	readout in stop mode
'b01101		*ReadoutDera1*	copy 1 BC to derandomizer buffer
'b01110		*ReadoutDera4*	copy 4 BC to derandomizer buffer
'b01111		*ReadoutDera8*	copy 8 BC to derandomizer buffer
'b10001	80	*LoadPipelineDelay*	defines pipeline delay, followed by delay value
'b01000	x	*CaptBeforeLut*	pipeline input from before the look up table
'b01001		*CaptBeforeBcid*	pipeline input from before BCID
'b01010		*CaptAfterBcid*	pipeline input from after BCID

3.4 Compact loading of Status Register

It might be more convenient to set all status registers by a single command token. The *LoadStatusRegister* token 'b10000 allows such operations. It is followed by an 11 bit data word. The meaning of the bits in this data word are explained in the following table.

Bits	extract from token:	Meaning
(0:1)	'b001xx	input from FADC or playback memory
(2:3)	'b010xx	capture mode for pipeline memory
(4:5)	'b011xx	readout mode
(6:7)	'b101xx	BCID operation mode
(8:9)	'b110xx	BCID peak finder mode
(11)	'b1110x	look up table bypass Mode

4 Readout Mode

The FeAsic has four different readout modes, which start to operate as soon as the *ListenFlag* is switched off. The Level-1 Trigger determines the Level-1 accept flag for every bunch crossing. The FeAsic receives the Level-1 accept flag and initiates the readout process for accepted events.

In stop mode the pipeline memory is stopped and the current data value is copied to the sequential I/O interface on request. Each request increments the read and write address of the pipeline memory. In this mode the whole pipeline memory is accessible.

In the high speed readout modes data from 1, 4 or 8 bunch crossings is copied to the derandomizer buffer. Those data words are copied on request to the sequential I/O interface. A problem can appear if the number of accepted events leads to an overflow of the derandomizer buffer. The derandomizer buffer has 32 addresses. If more than 32 data words are stored the address pointers for the derandomizer become corrupt. This problem is especially imminent, if 4 or 8 time slices per event have to be stored in the derandomizer. To exclude overflows new set of data words can only be copied to the derandomizer if this buffer is less than 75% full. Ie. the derandomizer buffer can store 24, 6 or 3 data sets depending on the number of time slices to be readout out per event. Every additional data set is rejected and the number of lost data sets is counted in a 2 bit counter (*LossCounter*).

The format for the 23 bit readout data word looks as follows. The first 2 bits identify the current readout mode ('b00 = stop mode, 'b01 = copy 1 BC, 'b10 copy 4 BC, 'b11 copy 8 BC). The 3'rd bit is unused. The next 2 bits show the current value of *LossCounter*. The *LossCounter* is copied together with the data into the derandomizer buffer and set to zero afterwards. The following 8 bit word holds the read address of the pipeline memory. The last 10 bits are reserved for the data word.

Literatur

[1] C. Geweniger et al.; Proposal of a Front End Digitization and Readout System for the ATLAS Level-1 Calorimeter Trigger, ATLAS DAQ-NO-96-xx

[2] I. Brawn et al.; Bunch Crossing Identification for the ATLAS Level-1 Trigger, CERN/LHCC/95-56