The Design of the ATLAS LAr Gain Selector Chip

Stephan Böttcher, John Parsons, Bill Sippach
Nevis Labs, Columbia University

June 26, 2000

Abstract

This note describes the design and implementation of the *Gain Selector* ASIC for the *Front End Board* (FEB) of the LAr calorimeter readout in the ATLAS experiment at the LHC.

The motivations for various design decisions are presented, and the impact of these decisions on the architecture of the FEB.

1 Introduction

In order to meet the large dynamic range required for the ATLAS LAr readout, the signal from each calorimeter channel is split into three overlapping, linear gain scales. The gain scales, denoted LO, MED, and HI, have relative gains of approximately 1, 10 and 100 respectively.

To reduce the event size and required readout bandwidth, typically only one of the gain scales will be digitized and read out for each channel. To optimize the readout precision, one would like to read out the scale which has the highest gain and yet is not saturated. The gain selection needs to be performed individually for each calorimeter channel, and separately for each Level 1 trigger. The Gain Selector chip is designed to perform this gain selection. In addition, it formats and serializes the data before sending it to the interface to the optical link used to transmit the data from the FEB to the Readout Driver (ROD).

Each Gain Selector chip manages the data from 16 calorimeter channels, corresponding to two ADCs. Therefore, 8 Gain Selector chips are required per 128-channel FEB.

The next section describes the Gain Selector functionality and operation in more detail, including a discussion of the the parameters required to configure it in the chosen mode and of the protection provided against corruption due to SEU. This is followed in Section 3 by a discussion of the interfaces between the Gain Selector and the other hardware of the FEB. Finally, Section 4 describes the design and implementation of the Gain Selector as a DMILL ASIC, including details of the design flow, layout, size, estimated power, and pinout. The event format is provided in Appendix A.

2 Gain Selector Functionality and Operation

This Gain Selector implementation offers the following features:

- Operation in *auto-gain*, *fixed-gain*, or *test* mode.
- The gain scale used for the gain selection algorithm is configurable via software.
- Individual 12-bit upper and lower auto-gain thresholds are available for each channel separately, and are configurable via software.
- One, two, or three gains can be readout in fixed-gain mode. The order of the gain scales is software configurable, but all gains must be different.
- In test mode, fixed data out of the thresholds store is sent instead of ADC data.
- Test mode triggers can be selected either from the SCA Controller as usual, or generated by SPAC broadcast events, to allow testing of the readout without an operational SCA Controller.

2.1 Power-up and Configuration

An *enable* input is provided on the Gain Selector. The *enable* signal is connected to the Configuration Controller, and is set upon power-up to automatically disable all Gain Selector output signals, until an explicit request is made via SPAC to enable the outputs. This feature ensures no spurious readout data will be generated while the internal logic of the Gain Selector reaches a stable state. Most importantly, the *sca_read* signal is guaranteed to be inactive while the *enable* input is low, preventing the two SCA chips which share the same ADC from both driving the output after power-up.

After power-up, the parameters needed to configure the Gain Selector can be downloaded via SPAC through the Configuration Controller. The downloaded parameters can be read back for verification. Once the Gain Selectors are correctly configured, the *enable* signal should be set to allow for normal operation.

2.2 Operation During Data Taking

During normal data taking, Level 1 Accepts are received by the SCA Controller, which performs the address bookkeeping required for the operation of the readout. To initiate the digitization of a sample, the SCA Controller must communicate with the Gain Selector. The Gain Selector has no notion of events; it just executes sample commands from the SCA Controller. To form an event, the sample commands must come in a well defined sequence, timed by the *s_ready* signal from the Gain Selector.

As depicted in Figure 1, the sample commands are send on four differential serial signals, at the readout clock frequency of 5 MHz, in a frame of eight serial bits on each signal.

In addition to the SCA cell addresses to be digitized, the Gain Selector receives from the SCA Controller, and forwards to the readout, a BCID (12 bit) and an event number (8 bit), to be send with the event header, and an SCA status word (8 bit), to be send with the event trailer.

The start bit is the first bit on the **frame** line. The next three bits on the **frame** line specify the sample operation to be performed.

bit 7: **1**. Start bit.

bit 6: **F**irst sample of an event, send event header, and perform gain selection algorithm (if in auto-gain mode).

bit 5: **L**ast sample of an event, send event status, and delay further samples to make room for the event header of the next event.

bit 4: **B**ackporch bit. Do not raise the gain for any channel in auto-gain mode.

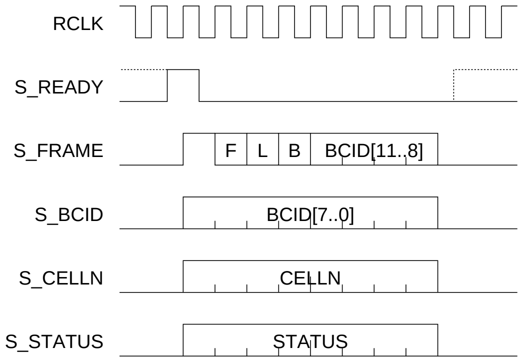

Figure 1: SCA Controller sample command protocol

The remaining four bits in the **frame** line are the upper four bits of the BCID. The eight lower bits of the BCID are transfered on the **bcid** line.

The SCA cell (capacitor) number is transfered on the **celln** line. The SCA cell number is received from the SCA Controller with each sample command and, with a 200 ns delay, serially forwarded to the SCA. The cell number is also sent to the readout in the sample header word. The sample header word further includes the three sample mode flags, and a test data mode bit.

The **status** transfers an eight bit event number with the first sample of an event, to be included in the first event header word. The same line transfers eight status bits in the last sample of an event, to be sent with the event trailer.

The timing between the samples is controlled by the Gain Selector, and transmitted to the SCA Controller via the *s_ready* signal. The SCA Controller must respond promptly to the *s_ready* signal, to prevent gaps between the samples within an event.

In response to the *Last* flag, the Gain selector generates an event trailer, and delays the *s_ready* signal, to ensure space for the next event header (except in auto-gain mode, where the time space is ensured due to the time required for the gain selection algorithm).

The sample command frame includes a third flag, called *Backporch*. If this bit is set with the *First* sample, in auto-gain mode, the gain selection algorithm is modified to prevent any channel from using a higher gain than in the previous event. This flag should be set by the SCA Controller with the first samples of events that occurred less than 400 ns (LAr charge collection time) after the previous event. The *Backporch* bit is send together with the *First* and *Last* flag to the readout in each sample header word. The Gain Selector does not interpret the *Backporch* bit in non-peak samples. The SCA Controller may use it to transmit other sample state to the readout, e.g., to cope with overlapping events.

2.3 Operational Parameters

There are a number of parameters that specify the mode of operation of the gain selector. There are three methods to specify these parameters.

1. Option inputs to the chip, hardwired on the FEB layout.
2. Software configuration via SPAC.
3. Sent with the sample frame from the SCA Controller.

2.3.1 Gain mode

The gain selector operates in one of four basic modes: fixed single gain mode, fixed two gains mode, fixed three gain mode, auto gain mode. The fixed gain modes have parameters to specify the gains to use. The auto gain mode has a parameter to specify which gain to use for scanning the peak sample.

All these modes and parameters are encoded in five bits, and there are still codes left for two test data modes. Two gain mode bits:

gain mode	
00	fixed single gain
01	fixed two gains
10	fixed three gains
11	auto gain

Two bits encode the first gain to use in fixed gain mode, or the gain to use for the auto-gain scan. The illegal gain specification 00 enables test data mode.

first gain	
00	test data
01	low gain
10	medium gain
11	high gain

The fifth bit is used only in two (three) fixed gains mode, to tell which of the remaining gains to read (second). In test data mode, this bit disables sample commands from the SCA Controller, and enables masked SPAC triggers instead.

first	**extra gain bit**	
00	0	test data, SCA controller enabled
00	1	test data, SPAC triggered
01	0	low, medium, high
01	1	low, high, medium
10	0	medium, low, high
10	1	medium, high, low
11	0	high, low, medium
11	1	high, medium, low

2.3.2 Test data mode

In test data mode, the ADC data and gain bits are replaced by the output of the configuration store. To properly use this mode, fixed single gain mode should be selected.

Precisely: In any fixed gain mode, in the first gain of the first sample of each event, the lower threshold value replaces the ADC data, and the two LSB of the upper threshold replace the gain bits. All other data words (including the first sample in auto-gain mode) will show whatever happens to be at the output of the gains store (channel 7 thresholds, except during configuration bus activity).

2.3.3 SPAC triggers

The Gain Selector can generate a sample readout frame in response to a command received on the configuration port. The command is encoded in the address of the configuration request, the data is ignored.

The SPAC trigger commands come in two flavors: masked and non-masked. A masked SPAC trigger is accepted only when the Gain Selector is configured in SPAC test mode. A non-masked SPAC trigger is always accepted, and may disrupt normal readout operation.

The SPAC trigger command can specify whether or not the sample frame should have an event header and trailer (both or none). Also, the *Backporch* bit can be set or not.

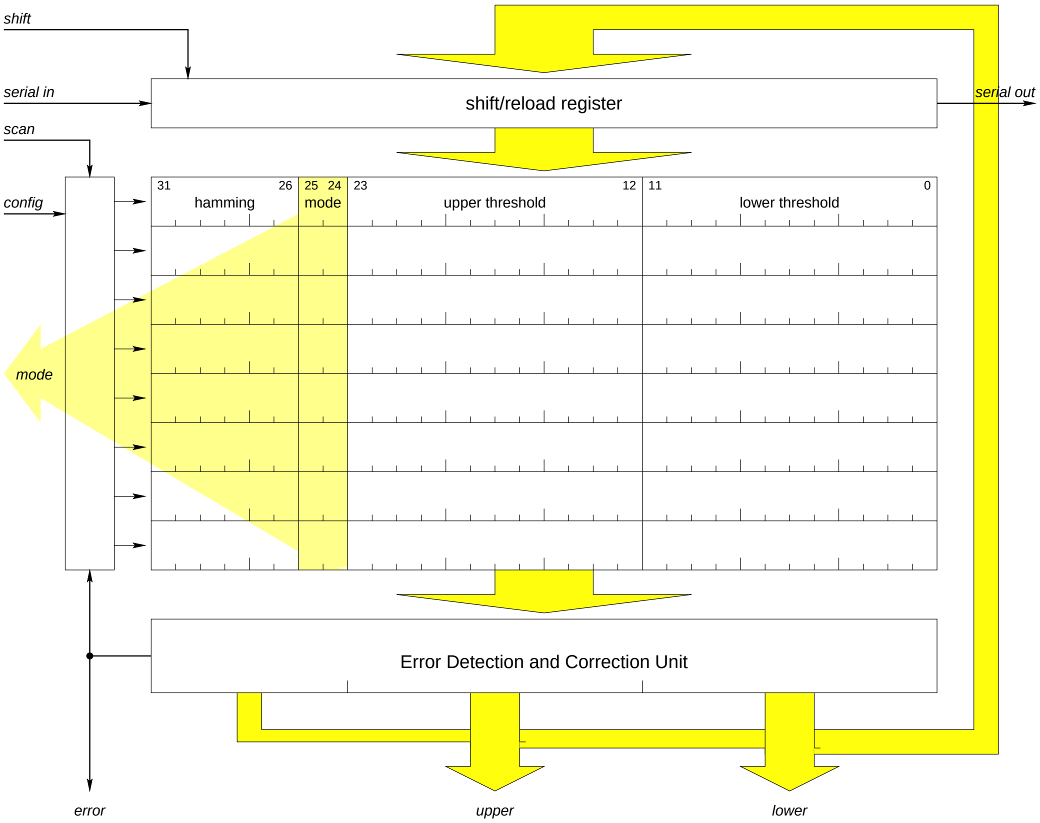

Figure 2: Thresholds and mode storage file.

In SPAC test mode, like in normal test data mode, the ADC data words are replaced by data from the configuration store. Triggers from the SCA Controller are ignored in this mode.

2.3.4 SEU Protection

As depicted in Figure 2, the design incorporates features to provide protection against SEU corruption of the downloaded parameters. For each of the eight calorimeter channels corresponding to a single ADC, a 32-bit word is assigned to store the relevant parameters. The mapping of these bits is shown below.

[31 : 26]	Six Hamming code bits.
[25 : 24]	Two mode bits.
[23 : 12]	Upper gain threshold.
[11 : 00]	Lower gain threshold.

Five mode bits are distributed in three copies each over the channel configuration words:

ch 0	mode, 00=one, 01=two, 10=three, 11=auto
ch 1	mode
ch 2	mode
ch 3	first gain, 00=test, 01=low, 10=medium, 11=high
ch 4	first gain
ch 5	first gain
ch 6	extra gain mode (two copies)
ch 7	extra gain mode (one bit ignored)

Each of the Hamming code bits enforces even parity on a subset of 16 bits:

[31]	31, 25 ... 11
[30]	30, 25 ... 18, 10 ... 4
[29]	29, 25 ... 22, 17 ... 14, 11 ... 7, 3 ... 1
[28]	28, 25, 24, 21, 20, 17, 16, 13, 12, 10, 9, 6, 5, 3, 2, 0
[27]	27, 25, 23, 21, 19, 17, 15, 13, 11, 10, 8, 6, 4, 3, 1, 0
[26]	26, 24, 23, 21, 18, 17, 14, 12, 11, 10, 7, 5, 4, 2 ... 0

Each configuration word bit contributes to an odd number of Hamming code bits. If any single bit gets flipped, the redundancy of the parity information is sufficient to allow the error to be detected and corrected. If two bits gets flipped, the error can be detected, but not corrected.

For single bit errors, the data is automatically corrected. An error flag is set in the output data, to allow monitoring of the rate of single bit errors. For double bit errors, an error flag is set to indicate the need to download the parameters again via SPAC in order to fix the problem.

3 Gain Selector Interfaces

The FEB readout clock (RCLK) operates at a frequency of 5 MHz, as does most of the Gain Selector logic. The 40 MHz clock is used to serially transmit the readout data.

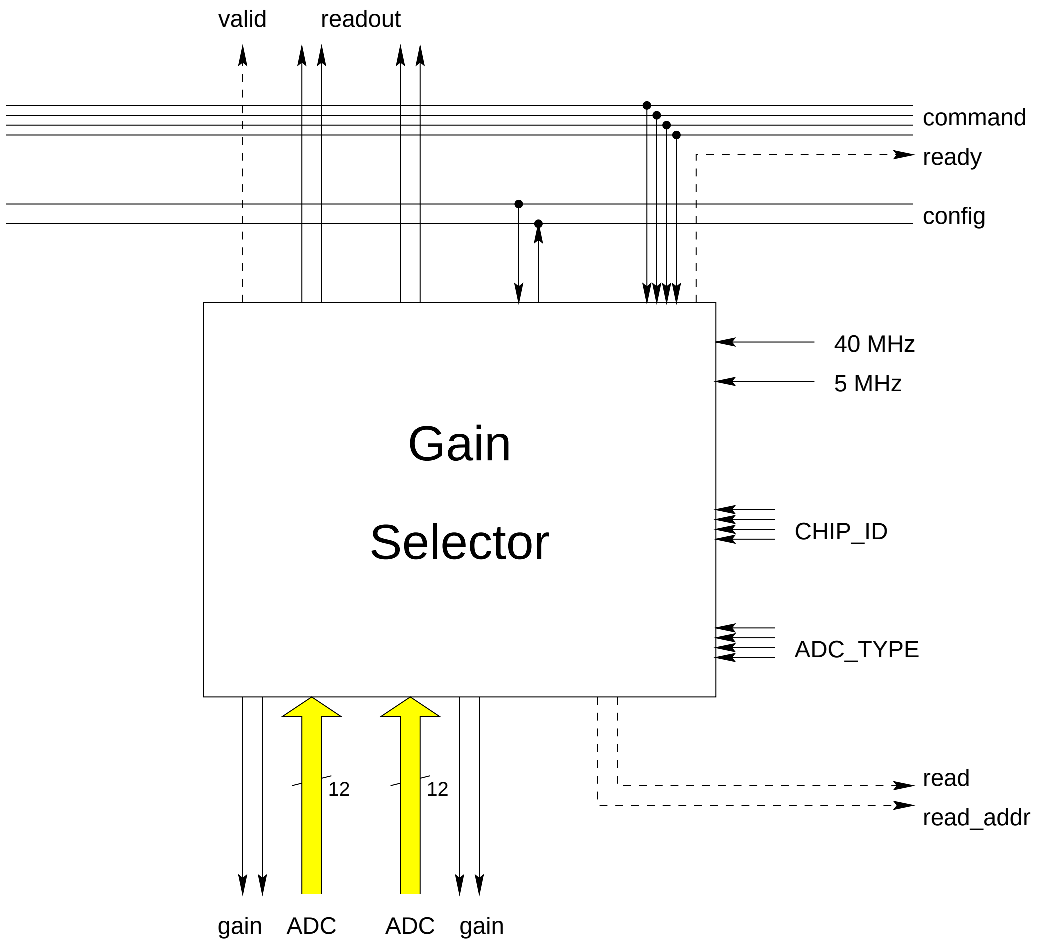

Figure 3: Interface signals of the Gain Selector. Dashed lines denote signals which are required only once per half-FEB.

As depicted in Figure 3, and described in more detail below, the Gain Selector chip needs to interface to five other types of chips on the FEB.

3.1 Interface to Configuration Controller

The *Configuration Controller* chip provides two uni-directional serial links, used for download and readback of the Gain Selector configuration parameters.

A 4-bit CHIP_ID provides the address to which a particular Gain Selector will respond. The CHIP_ID bits are configurable on the FEB layout. Three inputs specify to which settings of address bits $A[6..4]$ the chip must respond for configuration word read or write requests. One bit selects which setting of $A[3]$ selects the left or right Gain Selector on each chip. This bit allows flexibility in the final channel numbering, without constraining the chip placement. The value of the $A[6..3]$ address bits is sent to the readout link in the event header for each ADC.

The serial configuration protocol is described in a separate note: `http://www.nevis.columbia.edu/~stephan/lhc/ggains/verilog/sconf/protocol.ps`.

3.1.1 Address mapping

The configuration space addresses recognized by the Gain Selector are:

`0AAA0ccc`	read/write	first ADC, channel `ccc`
`0AAA1ccc`	read/write	second ADC, channel `ccc`
`1A1110fb`	write	SPAC trigger
`1A1111fb`	write	masked SPAC trigger

`AAA` is the chip ID programmed by option inputs.

3.2 Interface to SCA Controller

The *SCA Controller* provides four command signal lines required to initiate the digitization and readout of a sample, and receives a ready signal from one Gain Selector per half-FEB (64 channels), which serves as the handshake to signal that the system is ready for the next sample.

3.3 Interface to the SCA

The Gain Selector delivers two differential signals indicating the chosen gain selects to each of two pairs of *Switched Capacitor Array* (SCA) analog memories. In addition, the Gain Selector chip provides differential outputs for the SCA read enable, and serial SCA read address. These signals will be bussed to the SCAs, so that at most one Gain Selector per half-FEB needs to drive these signals. The final decision as to whether a Gain Selector or the SCA Controller drives these signal awaits completion of the FEB design.

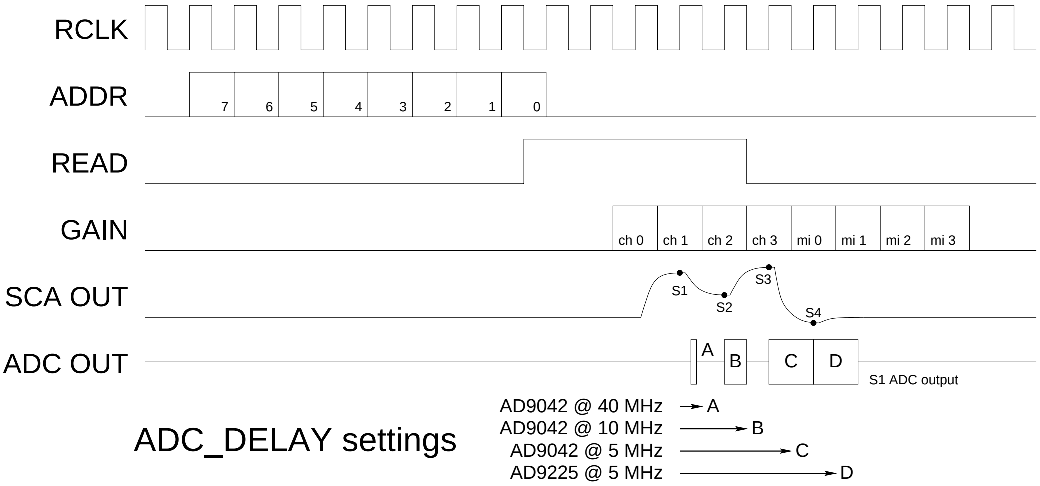

Figure 4: SCA read timing and ADC pipeline latency.

3.4 Interface to the ADC

The Gain Selector reads the digitized samples from two 12-bit ADCs. To allow some flexibility in the final choice of ADC, several pins on the Gain Selector, configurable on the FEB layout, are used to specify some of the properties of the ADC.

The latency of the ADC data with respect to the SCA control timing is set via two pins, allowing four possible choices. The slowest setting would be appropriate for the AD 8220 ADC used on the Module 0 FEBs. The baseline choice for the final FEBs, namely the AD 9042, would require the third setting. Two faster settings could be used to run the AD 9042 at 10, or 40 MHz. See Fig. 4.

Two further option pins select the polarity of the ADC output, and whether the ADC delivers its output in straight binary or two's complement format.

3.5 Interface to the Optical Link

The event data is formatted and serialized according to the event format provided in Appendix A. This format has been slightly updated since that used for Module 0, in order to take into account some changes in the FEB design (e.g., the Gain Selector will provide error flags for single- and double-bit errors detected in its downloaded parameters).

Each Gain Selector instance sends 16-bit words, two bits parallel, eight bits serial, at 40 MHz. The chip also provides a *data_valid* signal, which is active during the transfer of all words belonging to the event data block, excluding the mandatory *all-zeros* word that is inserted between events. The *data_valid* signal from one Gain Selector per half-FEB will be routed on the FEB to the optical link interface chip.

3.6 Interface Timing

The interface protocols to the SCA, SCA Controller, and Configuration Controller, are defined to assert the signals on the opposite edge of the RCLK as they are received. The RCLK is the 5 MHz clock that drives the SCA readout port. This ensures 100 ns setup and 100 ns hold time on every signal. The system is thereby insensitive to board level clock skews, and the clock distribution can be optimized for noise impact without further constraints.

Since the Gain Selector, the SCA Controller, and the Configuration Controller all talk to each other, little freedom remains to assign clock edges to chip operations.

The Gain Selector runs all internal operations, and most of the interface timing on the negative edge of RCLK. This is because the SCA receives the gain select signals and the read command signal on the rising edge of RCLK. The SCA serially receives the read address on the negative edge of RCLK. Thus, the cell address output is the only signal that is driven by the Gain Selector on the positive edge of RCLK.

The serial configuration receiver of the Gain Selector is tightly coupled to its internal operations, and thus also needs to run on the negative edge of RCLK. This mandates that the Configuration Controller runs the configuration busses at the positive edge of RCLK.

The SCA Controller needs to interface to the Gain Selector, driving the serial sample commands on the positive edge of RCLK. However, its interface to the configuration bus must run on the negative edge. These two functions are independent, so that this should not impose a problem for the design of the SCA Controller.

The serial readout runs on the LHC clock frequency (40 MHz). To ensure that the readout serializer can synchronize to the parallel data stream, the rising edge of the RCLK supplied to the Gain Selector must not line up with the falling edge of the provided LHC clock. It is foreseen that the FEB clock distribution will derive the RCLK from the rising edge of the LHC clock.

The Gain Selector drives the readout outputs (including *data_valid*) on the rising edge of the LHC clock. The Readout Optics Multiplexer is supposed to receive these signals on the rising edge of LHC clock. The hold time requirement of the Readout Optics Multiplexer will be met by propagation delays in the data path.

4 Gain Selector ASIC Design

The Gain Selector ASIC design was driven by the following objectives:

- Fit two Gain Selector instances per chip, in order to handle the data from two ADCs (i.e., 16 calorimeter channels).
- Provide robust interface timing (setup and hold times).
- Minimize the noise impact on the FEB. Use low voltage swing differential signals where applicable.
- Protect all data against SEU corruption which cannot be easily detected on the *ReadOut Driver* (ROD). Protect other stored state as much as affordable, to reduce the rate of detectable errors.
- Ensure that the design cannot lock up. Whatever the state of the circuit, it must, by itself after a few clock cycles, return into a state where it can be reconfigured, without the use of a reset pin.

A block diagram showing the architecture of the Gain Selector is shown in Figure 5.

4.1 Design Flow

The Gain Selector chip design is expressed in Verilog Hardware Description Language.

The design is targeted on the 0.8 μm DMILL process, using the standard cell library of the DMILL Design Kit (DDK) Version 5.1. Synthesis is performed by Synopsys Version 1999.10. The chip layout uses Cadence 4.4.3, employing Silicon Ensemble for automatic Place and Route. Simulation is

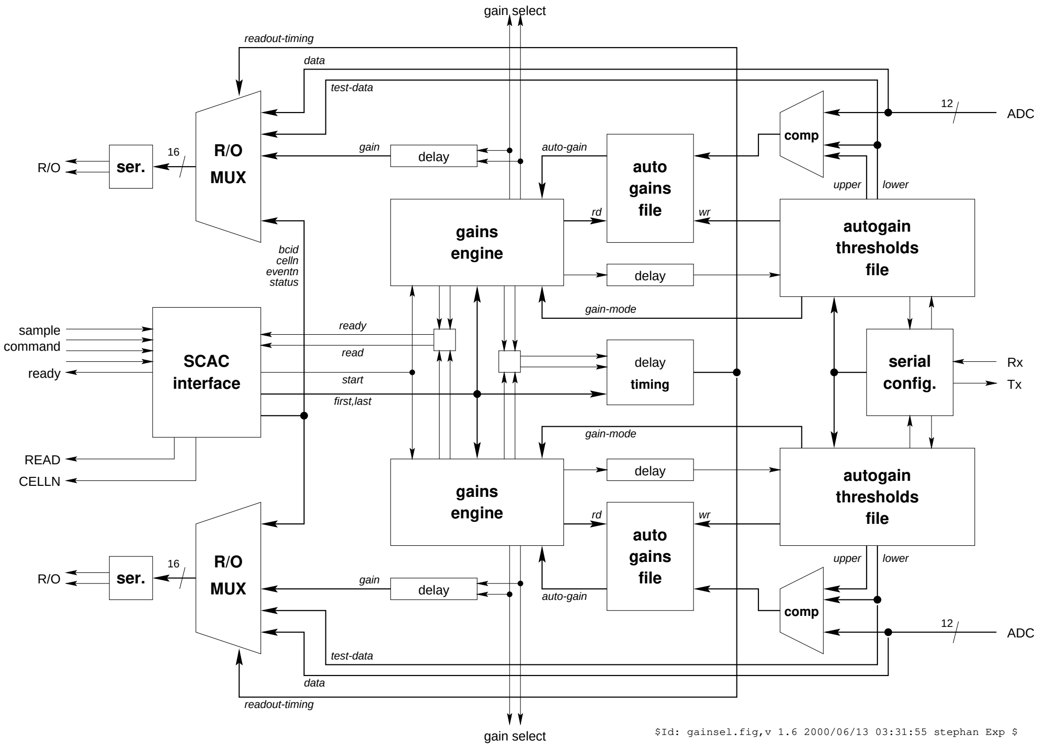

Figure 5: Block diagram depicting the basic architecture of the Gain Selector.

done with Icarus Verilog [`http://icarus.com/eda/verilog`] (pre-synthesis only) and Verilog-XL (pre-synthesis, post-synthesis, and with back-annotated post-layout timing). Design rule checking is performed with Diva.

4.2 Verilog Model

The structure of the design is expressed by separating the Verilog code into modules, as reflected in Figure 6. The Verilog files are available on the WWW at `http://www.nevis.columbia.edu/~stephan/lhc/ggains/verilog`.

Part of the complexity of the design arises from the fact that different parts of the circuit must operate in one of two distinct time domains, the SCA read control timing, and the ADC output timing. Careful attention had to be paid to synchronizing between these two domains. A brief description of the main Verilog modules is provided below.

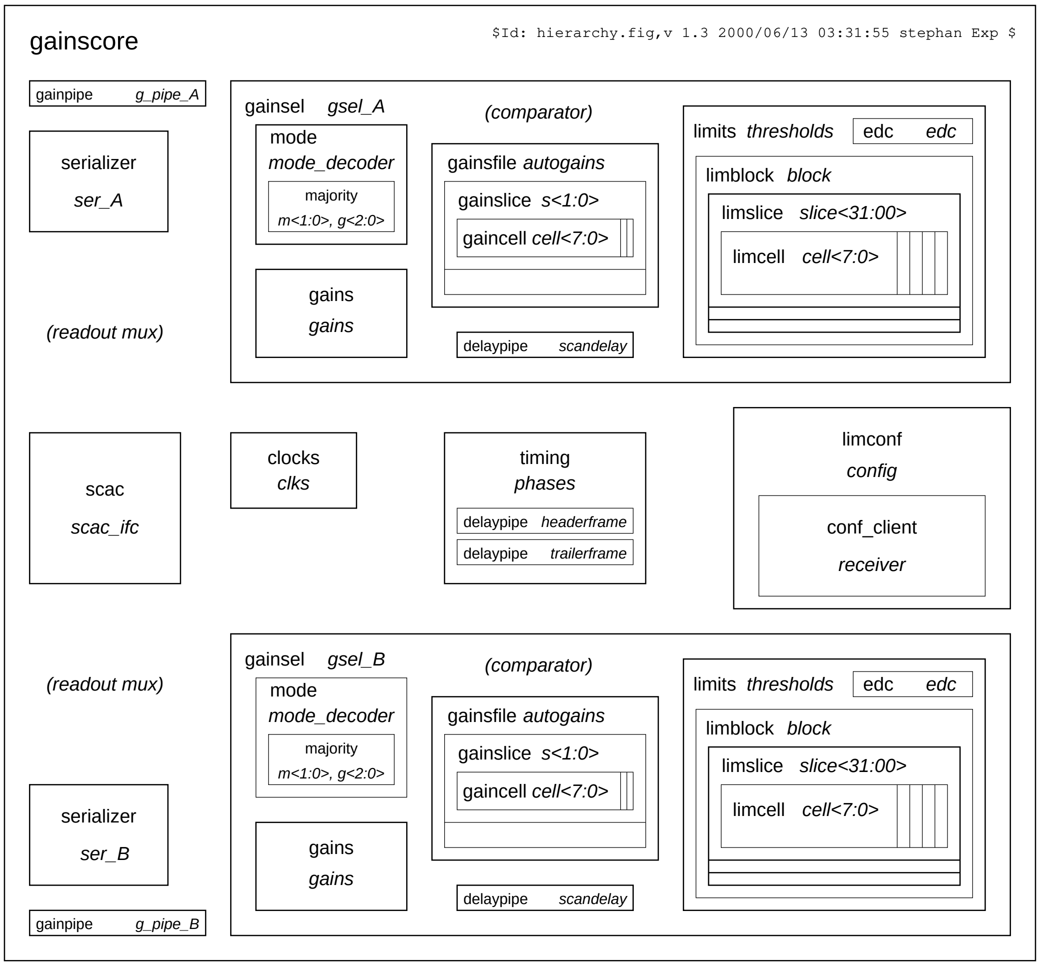

Figure 6: Verilog design entry module structure. Every box denotes a module instance, or an array of module instances. The module *(instance)* names are written in upright *(italic)* script into the boxes.

4.2.1 SCA Controller interface

The SCA Controller interface receives the sample frame commands from the SCA Controller, provides all sample/event parameter, and triggers the gains engine to do a sample frame. This module operates in the SCA control time domain.

4.2.2 GAINS engines

The GAINS engine drives the SCA *gain* selects, and triggers the readout timing (*go*, *done*) and the auto-gain algorithm (*scan*). This module operates in the SCA control time domain.

4.2.3 Timing and delays

The delaypipe and gainpipe modules contain variable delay shift registers, to adapt for the ADC pipeline delay. The signals that are delayed are: *sample_start*, to provide the event/sample header readout timing; *sample_done*, to provide the event trailer timing; *scan*, to trigger the gain selection algorithm; and the *gain* selects as provided to the readout.

These modules define the time shift between the SCA and ADC time domains.

4.2.4 GAINS file

The GAINS file stores the results of the gain selection algorithm. The read and write operations are independent, with reading operating in the SCA time domain, and writing operating in the ADC time frame.

Integrated into the GAINS file is a part of the gain selection algorithm that prevents the gain of any channel to be higher than in the previous event if the previous event occurred less than 400 ns earlier. This condition if flagged by the B-bit, which comes as part of the sample command from the SCA Controller.

4.2.5 Auto-gain thresholds files

The thresholds file stores all soft configuration information, mainly the auto-gain thresholds, but also the gain mode. The thresholds are provided in response to a *scan* trigger from the gains engine. During the auto-gain scan cycle, the data is fed through an *Error Detection and Correction* (EDC) circuit, to cope with SEU. The corrected data is written back into the latches whenever a single bit error was detected.

The thresholds file provides 16 bits of mode configuration. These bits are corrected for single-bit errors, but only after they would have been used once in a corrupted state. To avoid this, the required 5 mode control bits are stored three times each (requiring 15 of the 16 bits), and majority logic used to ensure the correct mode is used until the EDC unit can correct the stored bits (see Section 2.3.4). In case of single- or double-bit errors, the appropriate error flags are set in the output data.

This module operates in the ADC time domain.

4.2.6 Comparators

The comparator, operating in the ADC time frame, compares the ADC data with two thresholds, to determine the gain to use for each channel.

4.2.7 Serial configuration

The serial configuration module receives configuration commands from the Configuration Controller. To access the thresholds file, the serial configuration module temporarily disables EDC operation, and uses the shift register internal to the thresholds file to serialize/deserialize a data word. In case of a readback operation, the serial configuration module may need to wait for a gain selection process to finish, since the read port is shared, and the auto-gain operation has priority.

4.2.8 Multiplexer

The output multiplexer generates the event/sample data frame. The header data is driven by the SCA Controller interface. The data part comes from the ADC and the delayed gain selects. The event trailer comes in part from the SCA Controller interface. Since the event parameters live in the SCA time domain, the status bits for the event railer need to be latched at *sample_done*, to be available in the ADC time trailer. The readout operates in the ADC time domain.

4.2.9 Serializer

The readout data stream (16 bits at 5 MHz per ADC) is serialized on the 40 MHz LHC clock, two bits parallel, 8 bit serial. A *data_valid* signal is provided, which is active for all output words which are non-zero.

4.3 Gain Selector Layout

The synthesized netlist includes 3333 DDK standard cells (including standard and custom IO pads, excluding Power pads and clock tree). The chip has 88 pads (including 16 power pads). The chip dimensions are $4.6 \times 4.6\,\mathrm{mm}^2$. The logic cells occupy 45 % of the core area.

4.4 Input and Output Signals

All outputs are differential signals, driven by TTL output cells provided by the DDK library (jbot1). Each pair of output cells is driven by the inverting and non-inverting outputs of a flip-flop cell (jfd), to ensure small skew. The driving flip-flops are not further loaded (except *data_valid*), and pre-placed within the chip core, near to the corresponding output pad cells. The serial configuration readback output employs tri-state output cells (jbto1).

Two clock inputs and the 24 ADC inputs are single ended TTL signals, received by TTL input cells of the DDK library (jbit1). The clocks are distributed single-ended to ensure well defined edges, and to minimize insertion delay (jitter, skew) into the chip core.

The *enable* input employs a Schmitt trigger input cell (jbis1). This input serves as an active low reset, and may be driven by an *RC* network.

Eight board level configuration pins are equipped with TTL input cells with internal pullup resistors.

All other signal inputs are received by differential *receiver* cells, which are not part of the DDK library, but they are based on generic IO pads from the DDK library. The differential amplifier is taken from the SCA design. The version used on the Gain Selector employs an NMOS input stage, which works best in the lower part of the voltage range. This ensures compatibility with the LVDS standard and allows to interface to circuits operating at 3.3 or 2.5 V. A *bias* input cell provides the reference for the *receiver* cells bias current.

4.4.1 Low Voltage, Differential Signals

To reduce noise induced by digital signal on the FEB, most signals are transmitted differentially with low voltage swings. Instead of using custom on-chip LVDS drivers, the design uses standard TTL output drivers, and an external resistor network (see Figure 7). Two series resistors at the sending end set

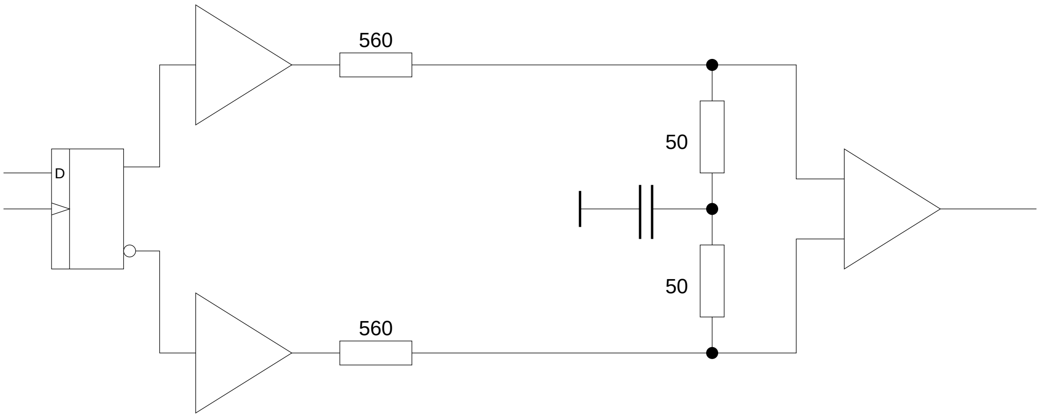

Figure 7: Arrangement used on the FEB for transmission of low voltage, differential signals.

the current to about 4 mA. Two 50 Ω termination registers and a center tap capacitor provide a differential input voltage of 400 mV.

4.5 Power consumption

The power supply voltage is 5 V. The power consumption of the logic core was estimated to be 16 mA [`http://www.nevis.columbia.edu/~stephan/lhc/ggains/verilog/floorplan.txt`].

The IO pads will draw 40 mA for ten low voltage, differential drivers (see section 4.4.1 below). This power is dissipated in the resistor network external to the chip.

The DDK IO cells provide four power rings, for analog and digital V_{CC} and V_{SS}. Since the Gain Selector does not include analog circuits, the analog power rings were connected to the digital counterparts. There are *no* separate I/O and core power pins. An additional pair of power rings (50 μm wide) were placed around the chip core, to facilitate the power distribution into the core. In addition to the power stripes associated with the DDK standard cells, there were four vertical power stripes added to the core layout.

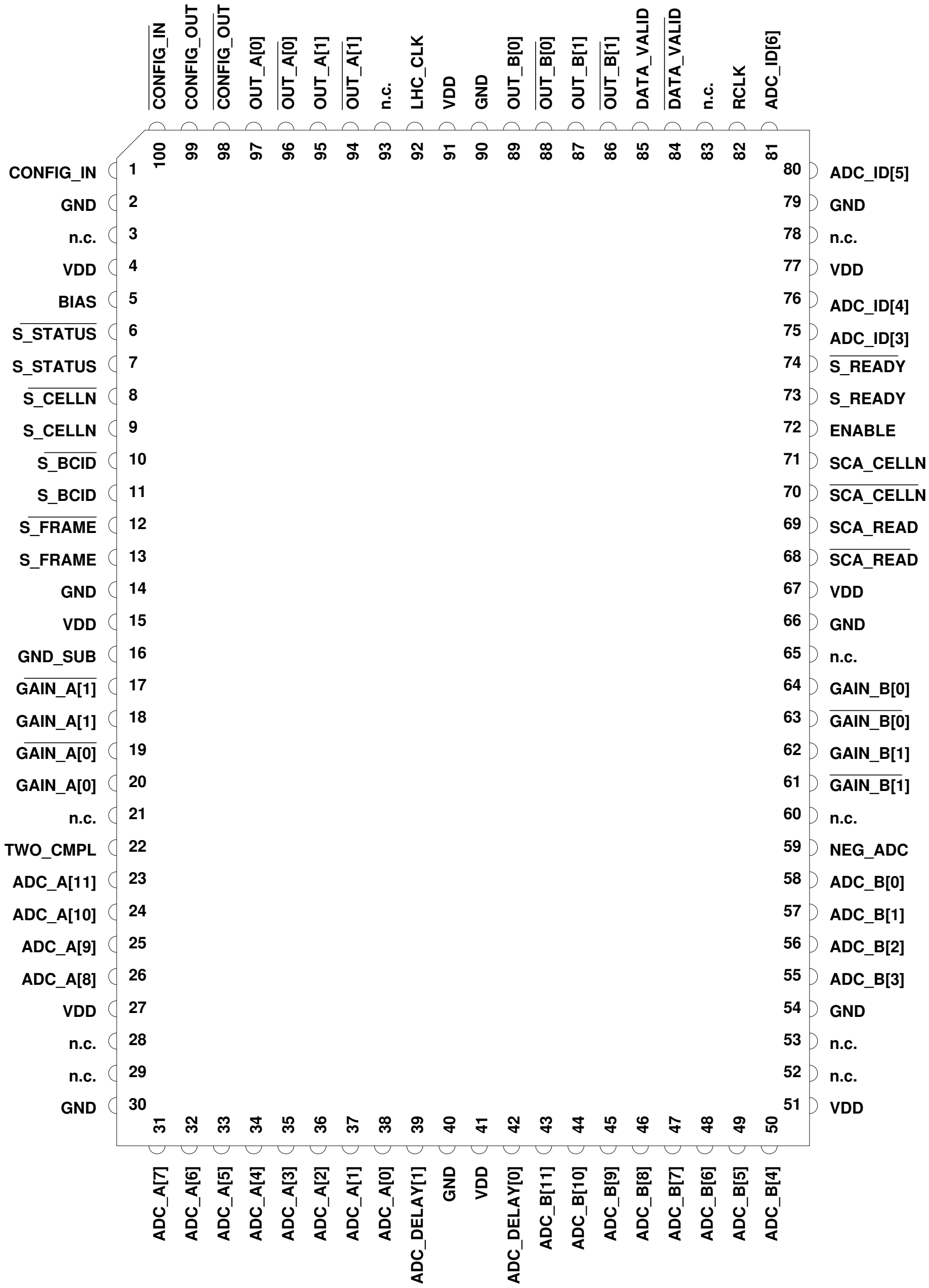

Figure 8: Pin assignment of the Gain Selector chip.

5 Simulation Results

The Gain Selector design was simulated with Verilog-XL at various stages in the design process, using the primary Verilog source for functional verification, the synthesized netlist for simulation including gate delays, and finally the netlist with clock tree, with back-annotated timing extracted from the layout.

The simulation tests parameter download, sample processing in all modes, SEU error correction, and parameter readback.

The simulations were run with DDK best case, nominal, and worst case timing. The back-annotated, worst case timing simulations succeed at elevated operating frequencies up to 140 MHz LHC clock.

Best case is (0.5, 0 °C, 5.5 V), and worst case is (1.5, 85 °C, 4.5 V), (process, temperature, supply voltage).

6 Pinout and Package

The Gain Selector has 88 bond pads, 22 on each edge. The chip will be packaged in a 100 pin PQFP package, the same one used for the Shaper, SCA, and Configuration Controller. Fig.8 shows the pinout (preliminary).

A Readout event format

The event data format consists of an frame start word, two event header words, one sample header and $n \times 8$ data words per sample, an event trailer word and at least one frame end word to separate from the next event.

When the readout link is idle, words of all zeros are sent. The event start word is all ones. Both of these violate the odd parity imposed on the header and data words. Bit 15 of all words is zero, except for the start word. Bit 14 is used as a Parity bit to set odd parity for all header and data words.

Bits 12 and 13 are zero for header and trailer words, and encode the gain for data words, 01: low, 10: medium, and 11: high gain.

The first event header word carries the ADC-id (4 bits) and the bits sent by the SCA Controller on the **status** line with the first sample. The idea is to make it an event number.

The second event header word contains the BCID, as received from the SCA Controller with the first sample command.

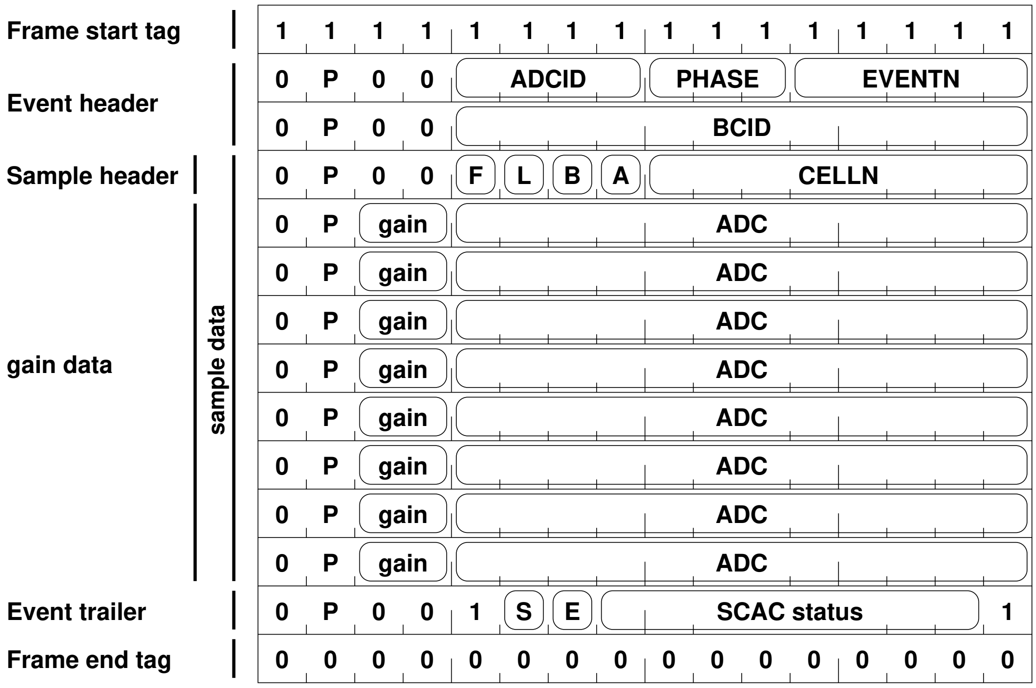

Figure 9: Readout event format

The sample header word contains the SCA cell (capacitor) number (8 bits), the three sample mode bits from the SCA Controller, and the test mode bit.

The data words contain the ADC data and gain bits.

The event trailer (status) word contains

Event status bits	
Bit 0	Forced high.
Bits 1...8	SCA status word.
Bits 9...10	Single/Double bit errors in EDC unit.
Bit 11	Forced high.