

SCA Controller for the Front-End Boards

D.M. Gingrich, J.C. Hewlett, L. Holm, S. Mullin, B. Zhang
University of Alberta, Edmonton, Canada

Version 1.2: August 11, 1999

This document describes the requirements and design of the controller for the ATLAS LAr calorimeter front-end boards. Many issues and choices will only be decided when the front-end board design is finalized. Thus this document is a working document that will undergo considerable revision as the design changes. When necessary, issues relevant to the field programmable gate array implementation of the controller for the module-0 calorimeters are explicitly mentioned.

Contents

1	Functional Requirements	4
1.1	Control of the SCA	4
1.2	Output Information	5
1.3	Control Parameters	5
1.4	Debug Functions	5
1.5	System Issues	5
2	Design Methodology	6
3	Functional Description	7
3.1	FIFO Memories	7
3.2	Address Encoding	7
3.3	Address Sequencing	7
4	Inputs to the Controller	10
4.1	Inputs from the Fast Signal Distribution System	10
4.2	Communication with SPAC	11
4.3	Serial Loading of Programmable Registers	11
5	Modes of Operation	12
6	Configuration Parameters	13
7	Outputs from the Controller	14
7.1	Signals Sent to the SCA	14
7.2	Communication with the Gain Selector Logic	15
8	Status Register	16
9	Environmental Conditions	17
10	Verification of Performance	18
	Appendices	19
A	Configuration and Download	19
A.1	Xilinx Configuration Protocol	19
A.2	Parameter Download and Readback Protocol	19
B	FIFO Memories	22
C	Address Sequencer	27
D	Grey Encoder	30

E	Parameter Download and Readback Circuit	32
E.1	Serial Download	33
E.2	Serial Download Device Select	36
E.3	Serial Download Registers 0 to 9	36
E.4	Serial Readback and Reset	38
E.5	Trigger Diagnostic	38
F	Description of Circuit	43
F.1	Reset Generation	43
F.2	Startup Circuit	43
F.3	Load Write FIFO	46
F.4	Address Deletion	46
F.5	SCA Write Address	48
F.6	Read-Delay FIFO and Read FIFO	51
F.7	Bunch Crossing Counter	51
F.8	Read Trigger	51
F.9	Generate Last Sample	56
F.10	Load Read Buffer RAM	56
F.11	Read Address Sequencer	59
F.12	Event RAM	59
F.13	Readout Control	63
F.14	External Output Signals	63
F.15	Read Control	66
F.16	Event Flag	66
F.17	Read-Complete Latch and FIFO	66
F.18	Error Detection	70
F.19	Inhibit Read and Write	70
F.20	Clocks Circuit	70
G	Important Signals	76
	References	77
	List of Figures	78
	List of Tables	79

1 Functional Requirements

There are three main functions on the front-end board that the controller is responsible for:

1. addressing the switched capacitor arrays (SCA),
2. send information to the gain selector logic and
3. providing data to the output stream.

The main guidelines of the controller requirements are that it be integrated and provided, as near as possible, dead-time free operation of the front-end boards.

1.1 Control of the SCA

The following are the functions that the controller must perform to operate the SCAs.

- Pipeline length: provide the read and write addresses for an SCA with 144 capacitors per channel.
- Write address: provide an 8-bit parallel write address to the SCA every 25 ns.
- Read address: provide five 8-bit serial read addresses at a maximum rate of 100 kHz.
- Grey code: encode the write and read addresses using Grey code.
- Monotonic sequencing: order the write addresses (before Grey encoding) sequentially.
- Provide the proper read-address order for gain selection: provide the address of the time sample at the peak of the pulse, followed by the remaining sample addresses ordered in time.
- Provide an error register: An example of an error condition is when there are no available write addresses left in the SCA.
- Remove SCA addresses: The ability to remove SCA addresses from the pool of available addresses.
- Overlapping events: Triggers will not occur less than 125 ns apart. If events partially overlap, the controller will signal the condition so that the gain can be held fixed by the gain selection logic. Overlapping events are only allowed if the number of samples is less than 9. Reading out more than 8 samples will normally be a diagnostic mode which will not generate overlapping events.

1.2 Output Information

The following is the information the controller must communicate to the gain selector logic and inject into the data stream.

- Address: inject the capacitor address of each time sample into the data stream for each trigger.
- Bunch-crossing number: inject the bunch-crossing number into the data stream for each trigger.

1.3 Control Parameters

The following parameters are downloaded at the beginning of a run, and as such, specify the run conditions.

- Trigger delay: sets the trigger delay time (latency).
- Number of time samples: sets the number of time samples that are digitized per trigger. The maximum number of samples is fixed at 32.
- Select available SCA addresses: the ability to pre-load the available SCA addresses that can be used.

1.4 Debug Functions

The following control functions may be used during debugging.

- The ability to stop the SCA write addressing while reading out.
- The ability to turn the address sequencing on or off.
- The ability to readout the same SCA address each event.
- The ability to trigger and reset from software.

1.5 System Issues

The controller will be loaded from an external serial port (not EPROM). For the FPGA implementation, the program and control parameters will be downloaded separately.

2 Design Methodology

We have chosen to prototype the controller design in FPGA technology. We are currently using the XC4036XL-1HQ240C from Xilinx. The design is written in VHDL. Synthesis and place and route are performed using Xilinx tools.

3 Functional Description

The controller supplies the read and write addresses for the SCAs. The management of the read and write addresses is implemented in a series of four FIFO memories which are described in the Technical Design Report.

A block diagram of the controller is shown in figure 1. A description of the FIFOs and treatment of the addresses follows. A detail description of the input and output signals, as well as, a description of the programmable registers and error conditions are given in subsequent sections.

3.1 FIFO Memories

The FIFO memories are dual ported and synchronous with the 40 MHz clock. The FIFOs are truly dual ported in the sense that when the read and write are enabled, and the FIFO is empty, the data written will simultaneously appear at the output. The data is written and/or read on the rising edge of the clock signal provided the write-enable and/or read-enable are pulled high.

Each FIFO is 8-bit wide. The write FIFO is 144 words deep, which corresponds to the number of capacitors per channel in the SCA. It has an almost-empty flag which will cause an inhibit if the system runs out of available SCA addresses. The almost-empty flag is currently set to 7. The read-delay FIFO is 128 words deep and has no flags since the number of addresses stored at any one time should be constant, which corresponds to the level-1 trigger latency. The read FIFO and read-complete FIFO are each 64 words deep and have empty flags which form part of the readout logic.

An additional 15-bit wide FIFO that is controlled in parallel with the read-delay FIFO is used to store the bunch-crossing number and other event information to be sent to the gain selector logic.

Details about the FIFO memories can be found in appendix B.

3.2 Address Encoding

The controller encodes and orders the SCA capacitor addresses to help reduce the digital noise. The read and write addresses are Grey encoded to minimize the number of bit transitions as the addresses change. A standard binary Grey encoding circuit is used for addresses up to 140. The last 3 addresses are recognized using comparators and are encoded as special cases. Details about the gray encoding can be found in appendix D.

3.3 Address Sequencing

The Grey encoding is only beneficial if the addresses are generated in order before encoding. The address sequencer orders the addresses by choosing the next SCA write address from the output of the write FIFO (WR) or the output of the read-complete FIFO (RC). The read-complete FIFO output becomes the write address if one of the following conditions is satisfied.

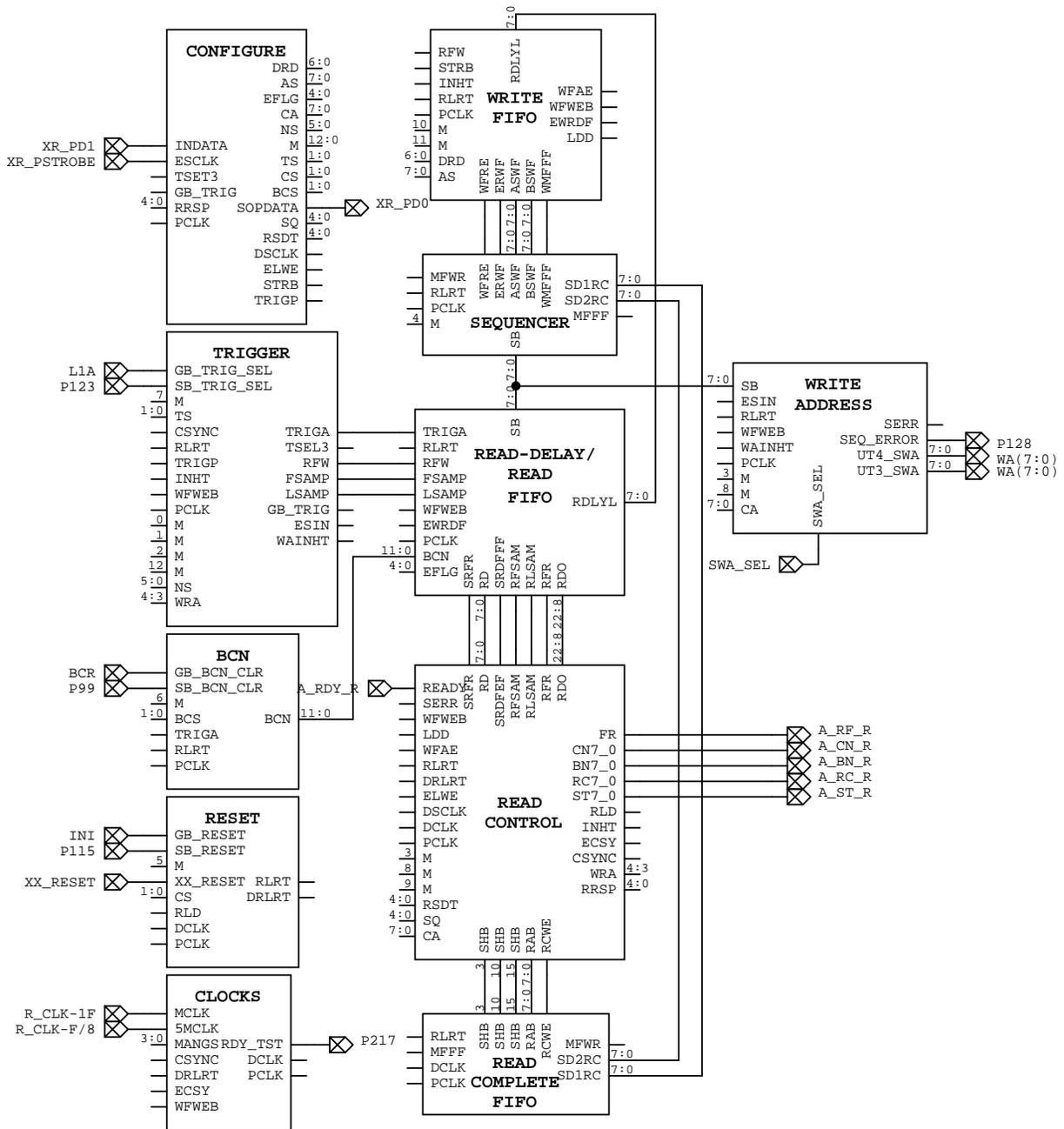


Figure 1: Block diagram of the controller for the front-end boards.

- The write FIFO output is greater than the read-complete FIFO output and the read-complete FIFO output is greater than the last selected address (LS):

$$(WR > RC) \circ (RC > LS).$$

- The write FIFO has wrapped around. The read-complete FIFO output is greater than the last selected address:

$$(RC > LS) \circ (LS > WR).$$

- The write FIFO has wrapped around and the read-complete FIFO has wrapped around, but the selected address has not yet wrapped around. The write FIFO output is greater than the read-complete FIFO output:

$$(WR > RC) \circ (LS > RC) \circ (LS > WR).$$

Three registers and three comparators provide the means of comparing the three addresses and inserting the free address back into the proper sequence of write addresses. Details about the address sequencing can be found in appendix C.

4 Inputs to the Controller

Table 1 lists the inputs to the controller. The pin assignment is for the FPGA implementation. The signal from the gain selector logic will be described in a subsequent section.

Pin	Schematic Label	Description	Net Label
<i>Fast Signals</i>			
124	R_CLK/XILINX-1F	40 MHz clock	mclk
118	R_CLK/XILINX-F/8	5 MHz clock	5mclk
113	INI	initialize	gb_reset
115	<i>not used</i>	board initialize	sb_reset
111	L1A	level 1 accept	gb_trig_sel
123	<i>not used</i>	board level 1 accept	sb_trig_sel
109	BCR	bunch-crossing reset	gb_bcn_clr
99	<i>not used</i>	board bunch-crossing reset	sb_bcn_clr
<i>Gain Selector Logic</i>			
218	A_RDY_R	ready	ready
<i>Parameter Download</i>			
36	XL_PSTROBE	strobe	esclk
32	XL_PDI	data in	indata
34	XL_PDO	data out	sopdata
<i>Configuration Download</i>			
179	X_CCLK_R	clock	CCLK
177	X_DIN/D0	data in	DIN
122	X_/PROGRAM	program	/PROGRAM
120	X_DONE	done	DONE
89	X_/INIT	initialize (10K pullup)	/INT

Table 1: Inputs to the controller. The pin assignment is for the FPGA implementation.

4.1 Inputs from the Fast Signal Distribution System

The controllers require the 40 MHz clock (R_CLK/XILINX-1F), level-1 trigger accept (L1A), bunch-crossing counter reset (BCR) and initialize (INI) signals from the fast signal distribution system. These signals are brought onto the front-end boards and distributed to the controllers. The use of the fast signals are as follows.

- The initialize (INI) signal causes a master clear in the controllers. All FIFOs are cleared, and the write FIFO is loaded with all the available SCA addresses.
- The readout of the SCAs is initiated by the level-1 trigger signal (L1A).
- A bunch-crossing counter is incremented locally and synchronized with the bunch-crossing counter reset signal. For the FPGA implementation, a local 12-bit counter

is incremented in the controller each 40 MHz clock cycle and can be reset with the bunch-crossing counter reset (BCR) signal.

- The 40 MHz and 5 MHz clocks are used as the write and read clocks respectively.

The initialize, level-1 trigger and bunch-crossing counter reset can also be set by using three additional pins, that are not used on the front-end boards, or from software. The selection of the source for these signals are controlled by three programmable registers.

4.2 Communication with SPAC

Table 2 shows the current SPAC commands [1] that communicate with the FPGA implementation of the controller. The SPAC is used to download both the FPGA configuration circuit, as well as, the programmable registers. The configuration circuit is downloaded (appendix A) over SPAC using the Xilinx protocol on dedicated pins.

Command	Bits	R/W	Description of Command
<i>Controller Configuration (booting)</i>			
0x28	D7:0	W	Byte of data to controller with clocks
0x29	X	W	Pulse PROGRAM pin
0x2a	D3:0	W	Not used
0x2b	D0	W	Not used
0x28	D2:0	R	D0: busy from SPAC, D1: DONE, D2: /INIT
<i>Controller Parameter Loading (left side)</i>			
0x10	D1:0	W	D0: data to controller, D1: strobe
0x10	D2	R	Data from controller
<i>Controller Parameter Loading (right side)</i>			
0x14	D1:0	W	D0: data to controller, D1: strobe
0x14	D2	R	Data from controller

Table 2: SPAC commands for the controller.

4.3 Serial Loading of Programmable Registers

All configuration parameters, and some diagnostic information, is sent over a bi-directional serial link to each front-end board using the SPAC protocol. The configuration parameters are downloaded over the serial link by first sending the register address, followed by the contents of the appropriate programmable register. The following signals are currently used.

- A strobe (XL_PSTROBE) loads the configuration data down the serial link.
- The configuration data (XL_PDI) consists of the operating mode and configuration parameters defining the operation of the controller.
- Readback of the registers is also done using the serial link (XL_PDO).

The protocol is described in appendix A.

5 Modes of Operation

Table 3 list the different modes of operation and their location in the control register.

Bit	Level	Description of Control Bit
0	low	Accept triggers
	high	Ignore triggers
1	low	Accept triggers
	high	Ignore triggers until read complete
2	low	Enable triggers and write addressing during read
	high	Disable triggers and write addressing during read
3	low	Grey encode SCA read and write addresses
	high	Bypass Grey encoding of SCA addresses
4	low	Enable address sequencer
	high	Bypass address sequencer
5	low	Do not reset controller and reload write addresses
	high	Reset controller and reload write addresses
6	low	Do not reset bunch-crossing counter
	high	Reset bunch-crossing counter
7	low	Do not issue trigger from software
	high	Issue trigger from software
8	low	Address all capacitors in SCA
	high	Address only a single capacitor in SCA
9	low	Automatically reload if write FIFO almost empty
	high	Inhibit automatic reloading if write FIFO almost empty
10	low	Disable delete/add of capacitor address from address list
	high	Enable delete/add of capacitor address from address list
11	low	Add capacitor address to address list
	high	Delete capacitor address from address list
12	low	Allow overlapping events
	high	Inhibit overlapping events

Table 3: Contents of the control register.

6 Configuration Parameters

Table 4 describes the programmable registers.

Name	Type	Description
M12:0	bit pattern	13-bit control register (see table of modes)
TS1:0	bit pattern	Source of trigger 0 = level 1 accept (L1A) 1 = board trigger (X-RESET) 2 = software trigger (control bit 7) 3 = test triggers
CS1:0	bit pattern	Source of initialize 0 = initialize (INI) 1 = board initialize (X_SPARE) 2 = software initialize (control bit 5) 3 = not used
BCS1:0	bit pattern	Source of bunch-crossing reset 0 = bunch-crossing reset (BCR) 1 = board BCR (P99) 2 = software BCR (control bit 6) 3 = not used
NS5:0	number	Number of samples per event (1-32)
DRD6:0	time	Read delay (25-3200 ns, 25 ns steps) trigger latency = read delay + 150 ns
EFLG4:0	time	Event flag (25-800 ns, 25 ns steps)
CA7:0	address	Address just this capacitor if control bit 8 high
AS7:0	address	Capacitor address to delete or add to address list

Table 4: Description of programmable registers.

The capacitor address to be added/deleted to/from the address list is strobed into/out of the list by accessing register address 1 during the download sequence.

A status bit is set if the time between events is less than the value set in the event flag register EFLG.

By specifying the first sample to be read, the download protocol loads a RAM with the order of the samples to be read out.

7 Outputs from the Controller

Table 5 list the outputs from the controller. The pin assignment is for the FPGA implementation.

Pin	Schematic Label	Description	Net Label
195	WA0	write address bit 0 if bus select low	ut3_swa(0)
194	WA1	write address bit 1 if bus select low	ut3_swa(1)
193	WA2	write address bit 2 if bus select low	ut3_swa(2)
192	WA3	write address bit 3 if bus select low	ut3_swa(3)
190	WA4	write address bit 4 if bus select low	ut3_swa(4)
189	WA5	write address bit 5 if bus select low	ut3_swa(5)
186	WA6	write address bit 6 if bus select low	ut3_swa(6)
185	WA7	write address bit 7 if bus select low	ut3_swa(7)
228	WA0	write address bit 0 if bus select high	ut4_swa(0)
229	WA1	write address bit 1 if bus select high	ut4_swa(1)
230	WA2	write address bit 2 if bus select high	ut4_swa(2)
231	WA3	write address bit 3 if bus select high	ut4_swa(3)
234	WA4	write address bit 4 if bus select high	ut4_swa(4)
235	WA5	write address bit 5 if bus select high	ut4_swa(5)
236	WA6	write address bit 6 if bus select high	ut4_swa(6)
237	WA7	write address bit 7 if bus select high	ut4_swa(7)
70	A_RF_R	frame bit	fr
72	A_RC_R	read control	rc7_0
74	A_BN_R	bunch-crossing number	bn7_0
76	A_CN_R	cell number	cn7_0
77	A_ST_R	status	st7_0

Table 5: Outputs from the controller. The pin assignment is for the FPGA implementation.

7.1 Signals Sent to the SCA

The write address (WA) is an 8-bit parallel address of the capacitor which will store the signal information corresponding to the current write clock. The address is latched in on the falling edge of the SCA write clock. The SCA write clock is not sourced from the controller but is obtained globally from the front-end board. The minimum setup time for WA must be 10 ns.

To make routing of the SCA write addresses on the front-end board easier, the controller sends the write addresses out on a different set of pins depending on the physical location of the controller on the board. The DC level of the write-address bus select (pin 206) is used to make the choice.

7.2 Communication with the Gain Selector Logic

The controller sends 8-bit data packets to the gain selector logic over five pins. The meaning of the bits is shown in table 6

Controller Data	Gain Selector Input	Bits (MSB first)							
		7	6	5	4	3	2	1	0
Frame	FR	1	0	0	0	0	0	0	0
Read Control	RC	OP1:0		C1:0		BN11:8			
Bunch Number	BN	BN7:0							
Cell Number	CN	CN7:0							
Status	ST	ST7:0							

Table 6: Data sent from the controller to the gain selector logic.

The read control bits 7 and 6 are decoded into four kinds of operations:

OP=0: header + cell number + ADC data for one sample + status word
 OP=1: header + cell number + ADC data for one sample
 OP=2: cell number + ADC data for one sample
 OP=3: cell number + ADC data for one sample + status word

Control bit C0 is used to redefine ‘auto gain’ settings for partly overlapping events. It is set for each overlapping sample. Control bit C1 is a spare.

The bunch-crossing number is required only for header operations and comes out only for OP=0 and OP=1. The status word is required only for the last sample but comes out all the time.

The data transmission is controlled by the RDY (table 1) level from the gain selector logic and the 5 MHz read clock. The data is sent MSB first. The frame bit causes the RDY to go low and stay low until the gain selector logic can accept another packet. All signals are sampled with the rising edge of the read clock. The controller does not restore the last SCA address back into the write FIFO for at least eight read clock cycles after RDY goes high. This is to allow time to readout the mirror SCA [2].

8 Status Register

The controller provides a status register which is transmitted to the gain selector logic for insertion into the data stream. Table 7 shows the current possible status conditions.

Bit	Level	Description of Status Bit
0	high	Addresses went out of sequence
1	high	Triggers are being ignored
2	high	Current trigger occurred within EFLG ns of last trigger (EFLG is a programmable register)

Table 7: Status bits.

9 Environmental Conditions

The power consumption of each controller on the front-end board is about 1.2 W which is a small fraction of the total power consumption of the board. Adequate cooling of the board will allow commercial packaging of the controller chip. Accessibility to the front-end boards is limited so reliability must be high.

The radiation levels at the locations of the front-end boards are sufficiently high such that care must be taken to ensure that the electronics is sufficiently radiation tolerant. A series of total ionizing dose tests [3] have been performed on Xilinx FPGAs and indications are that using these types of devices in ATLAS would be inadequate. In addition, because of the large amount of SRAM in the controller, estimates indicate that the rate of single-event upsets in ATLAS will be at the level of a few per minute. We are currently investigating the migration issues of going from the Xilinx FPGA to a radiation-tolerant ASIC and hope to have samples by the end of 1999.

10 Verification of Performance

Since 1993 we have built a series of controller prototypes. Two different discrete-component full-board ECL prototypes were tested using the RD3 calorimeter [4, 5].

The first FPGA version for ATLAS was made in 1997. At that time, the largest FPGA in the Xilinx XC4000 series was the only option with the dual ported RAM that would allow the required FIFOs to be designed. A series of prototypes were made using XC4025E and XC4028E chips. These new FPGAs were rather slow and hence a design using dual interleaving memories operating at half the beam crossing frequency was implemented [6]. This controller was used to control the SCA test chip that was made prior to the module-0 SCA chips.

With the release of the XC4036XL chip in a speed grade of -1, it was possible to implement non-interleaving memories that ran at the full 40 MHz clock speed. We are currently using two XC4036XL-1HQ240C on each front-end board. Speed grades of -09 have now become available, as well as, the possibility to use the Altera 10K Flex series chips.

The design was made using Mentor schematic capture with Xilinx place and route tools. We currently have synthesizable VHDL that can also be placed and routed using Xilinx tools.

Sixteen controllers have been used successfully in two different test beam periods in 1998. The design runs at clock rates of over 60 MHz and has ran stably for over a month on the test bench.

The front-end system was reviewed in 1999. This document is a greatly expanded version of a document submitted to that review [7].

A Configuration and Download

A.1 Xilinx Configuration Protocol

The following pins are used for loading the Xilinx configuration file. We use the Xilinx pin labels. The corresponding front-end board labels are shown in table 1.

CCLK is the configuration clock.

DIN is the serial configuration data input which receives data on the rising edge of CCLK.

$\overline{\text{PROGRAM}}$ is an active low input that forces the FPGA to clear its configuration memory. It is used to initiate a configuration cycle. When $\overline{\text{PROGRAM}}$ goes high, the FPGA finishes the current clear cycle and executes another complete clear cycle, before it goes into a wait state and releases $\overline{\text{INIT}}$. The $\overline{\text{PROGRAM}}$ pin has a permanent weak pull-up, so it need not be externally pulled up to V_{CC} .

$\overline{\text{INIT}}$ is an active-low open-drain output that is held low during the power stabilization and internal clearing of the configuration memory. During configuration, a low on this output indicates that a configuration data error has occurred. An external pull-up resistor is recommended.

DONE indicates the completion of the configuration process.

The following protocol is used to configure the FPGA:

1. Pull $\overline{\text{PROGRAM}}$ low.
2. Make sure $\overline{\text{INIT}}$ goes low in response to this.
3. Release $\overline{\text{PROGRAM}}$ and make sure $\overline{\text{INIT}}$ goes high.
4. Download serial data one frame at a time. A frame is string of characters in the configuration file (rft format) that is terminated by a carriage return. A frame is less than 512-bit.
5. Check for an error at the end of each frame. Make sure $\overline{\text{INIT}}$ stays high and DONE does not go high before the end of the configuration file is reached.
6. When at the end of file and all the frames have been downloaded, check that DONE goes high.

A.2 Parameter Download and Readback Protocol

Before downloading parameters issue 169 1s (clocks) to enable the logic to recognize commands.

There are seven commands that control the download of the data registers and RAM, and control the readback of the data registers and RAM. The commands are shown in table 8

Description	Mnemonic	Code		Signals	
reset all latches	RESET_ALL	1111011111	0x3df	NCLSQ	\overline{EN}
enable write access	SET_ENTER	0011011111	0x0df	ESET	ENT
enable address mode	SET_ENABLE	0111011111	0x1df	ESEN	EN
enable data mode	RESET_ENABLE	1011011111	0x2df	EREN	\overline{EN}
enable read access	SET_RDBK	0101011111	0x15f	ESRB	RBK
enable register access	SET_MUXE	0001011111	0x05f	ESMX	MUXE
enable RAM access	SET_RAME	0110011111	0x19f	ESRM	RMCE

Table 8: Download and readback commands.

A write requires placing the data on the XL_PL1 line, most significant bit first, and strobing in the bits using the XL_PSTROBE¹. Reading requires first strobing XL_PSTROBE 10 times to shift the RESET_ENABLE command out of the readback command shift register. The data can then be obtained by strobing XL_PSTROBE and reading XL_PDO for each bit in the register or RAM. The data comes out least significant bit first.

The four protocol sequences are as follows.

Load a register:

1. RESET_ALL
2. SET_ENTER
3. SET_MUXE
4. SET_ENABLE
5. write register address
6. RESET_ENABLE
7. write register data

Readback a register:

1. RESET_ALL
2. SET_RDBK
3. SET_MUXE
4. SET_ENABLE
5. write register address
6. read register data

¹We use the labels for the controller on the left side of the front-end board. Replace the XL part of the label with XR for the controller on the right side of the front-end board.

Load RAM location (set first sample to read):

1. REST_ALL
2. SET_ENTER
3. SET_RAME
4. SET_ENABLE
5. write RAM address
6. RESET_ENABLE
7. write RAM data

Readback a RAM location:

1. RESET_ALL
2. SET_RDBK
3. SET_RAME
4. SET_ENABLE
5. write RAM address
6. read RAM data

B FIFO Memories

Table 9 shows the size of the RAM used in all the FIFOs in the design. The number of Xilinx configurable logic blocks is also given. All FIFOs are dual ported. The read FIFO and read-complete FIFO have empty flags. The write FIFO has an almost empty flag which is set to 7. Figures 2, 3, 4 and 5 show the schematics of the main FIFO memories.

Function	Depth	Width	Bits	CLBs
write FIFO	144	8	1152	72
read-delay FIFO	128	8	1024	64
read FIFO	64	8	512	32
BCN FIFO	64	15	960	60
read-complete FIFO	64	8	512	32
address deletion RAM	144	1	144	9
load read buffer RAM	32	8	256	16
event RAM	32	15	480	30
read address sequence RAM	32	5	160	5
<i>Total SRAM</i>			5200	320

Table 9: Size of the RAM for the FIFO memories in bits and Xilinx configurable logic blocks (CLB).

In addition to the RAM used in the FIFOs, the RAM shown in table 10 is used.

Function	Depth	Width	Bits	CLBs
add/delete register	144	1	144	9
read buffer	32	8	256	16
sample order	32	5	160	10
event buffer	32	15	480	30
<i>Total SRAM</i>			1040	65

Table 10: Size of the RAM in bits and Xilinx configurable logic blocks (CLB).

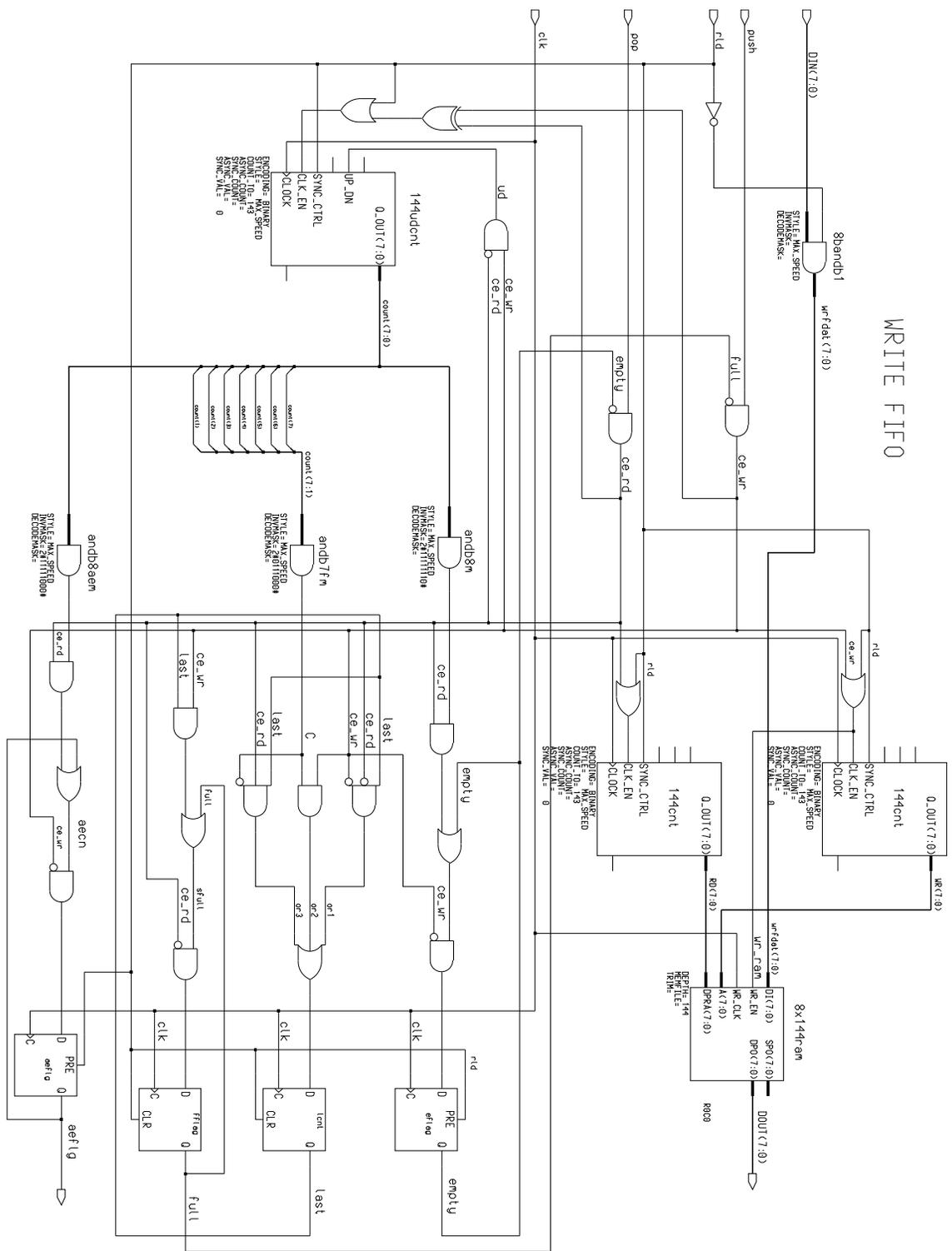


Figure 2: Write FIFO.

C Address Sequencer

This circuit (figure 6) orders the addresses in a monotonic sequence. 8-bit comparators are used to compare the address from the read-complete FIFO (RCBL7:0), the address from the write FIFO (WFOL8:0) and the current address (SB7:0).

The five comparators make the following comparisons:

$$\begin{aligned} \text{ST1} &:= \text{WFOL} > \text{RCBL}, \\ \text{ST2} &:= \text{SB} < \text{RCBL}, \\ \text{ST3} &:= \text{SB} > \text{RCBL}, \\ \text{ST4} &:= \text{WFOL} < \text{SB}, \\ \text{ST5} &:= \text{WFOL} = \text{RCBL}. \end{aligned}$$

If the result of the logic statement

$$\text{M4} + \text{ST5} + (\text{ST1} \circ \text{ST2}) + (\text{ST2} \circ \text{ST4}) + (\text{ST1} \circ \text{ST3} \circ \text{ST4})$$

is true, the address from the read-complete FIFO is chosen. M4 is the fourth bit in the control register, and if it is high, the address sequencing is disabled.

Since we have encountered timing problems in this part of the circuit when targeting for the early Xilinx devices, we have split the address streams into two.

8-bit multiplexers are used to choose between the two streams of addresses from the write FIFO and choose between the two streams of addresses from the read-complete FIFO. The two streams contain the addresses from alternate rising edges of PCLK. The multiplexers contain an 8-bit output latch that has a clock enable.

Toggling latches provide the alternating select signal to choose the correct address stream. 2-bit counters keep track of the number of addresses waiting to be multiplexed, control the multiplexer enable and the clock enable of the toggling latches. The counters can also control the write enable (pop) of the write and read-complete FIFOs. The following describes the operation for the write FIFO addresses. The operation for the read-complete FIFO addresses is similar.

A 2-bit counter keeps track of the number of addresses waiting to be multiplexed out. If there are no addresses to be multiplexed (WMFEF high) the toggling of the latch and multiplexer are disabled (WMFRD low). The counter counts up if the pop of the FIFO is enabled (WFRE high), and the toggle and multiplexer are disabled. The counter counts down if the pop of the FIFO is disabled, and the toggle and multiplexer are enabled; else the counter does not count. If there are two addresses waiting to be multiplexed (WMFFF high) the pop of the FIFO is disabled. The requirement for the multiplexer and toggle to be enabled is that the sequencer logic has chosen it (WFRI low in the case of the write FIFO) and the counter is not at zero.

The multiplexer for the write FIFO normally has a value that can be used by the sequencer selector logic. However, there is only a value for the read-complete FIFO when a read has occurred and there are addresses to go back into the address stream. The first time an address becomes available from the read-complete FIFO the address sequencer select

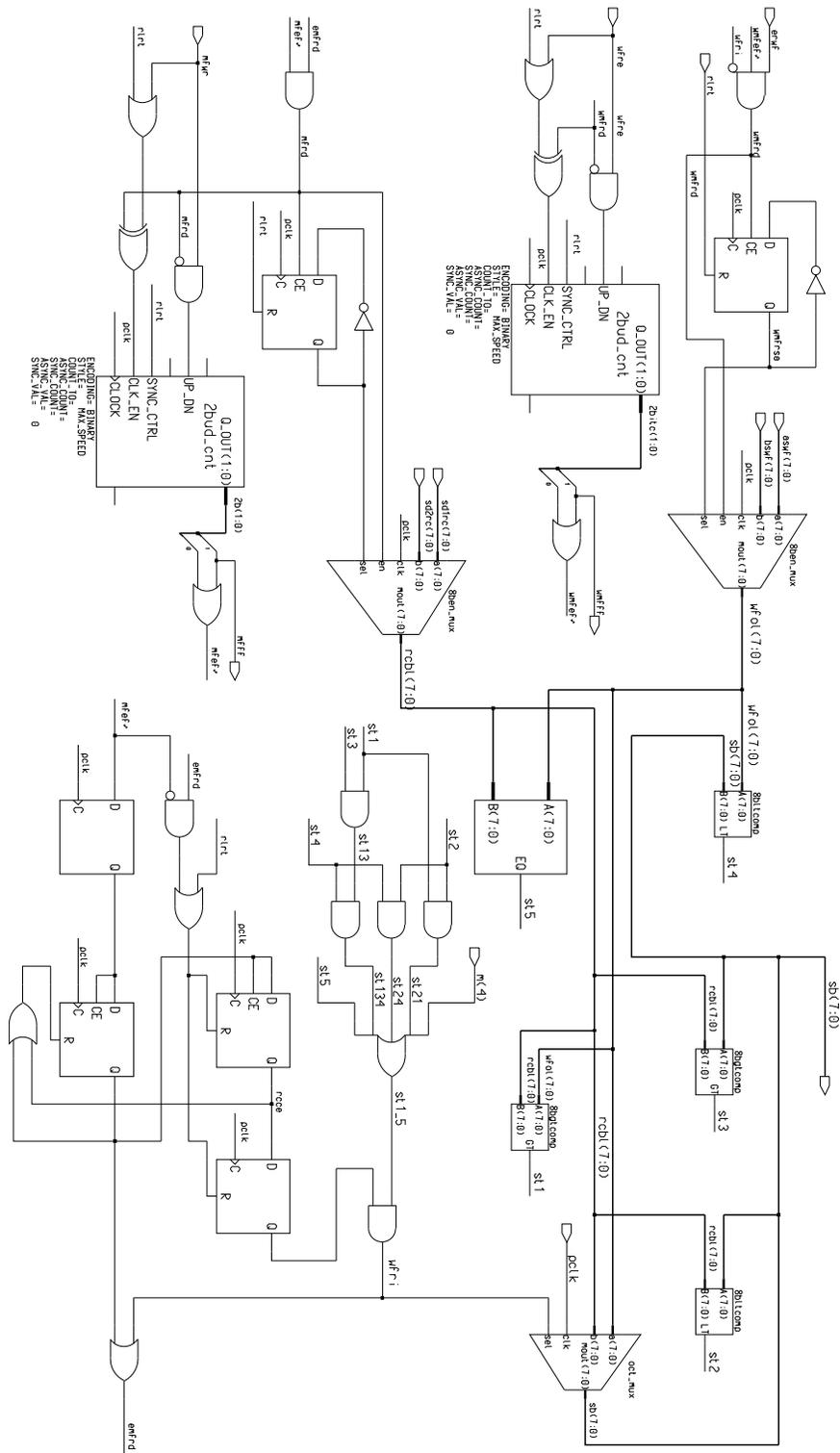


Figure 6: Address sequencer.

logic must be by-passed to allow the address to be loaded into the multiplexer for subsequent comparison. If the pop of the read-complete FIFO is enabled and the counter is at zero, the counter counts up. This cause the signal EMFRD to pulse high, and enable the multiplexer and toggle. It also enables the possibility for WFRI to go high and proper address sequences to be determined. This special logic resets when there are no more addresses from the read-complete FIFO to be compared.

D Grey Encoder

The circuit used to Grey encode the write and read addresses is shown in figure 7. The Grey encoding is performed using combinatorial logic. A standard algorithm is used for the first 141 addresses. The last three addresses (141, 142, 143) are recognized by using 8-bit comparators and encoding these special cases with fixed signal values. The encoder contains an 8-bit input latch which is clocked by signal PCLK. If the mode bit M3 is set high the Grey encoding logic is by-passed.

GREY CODE

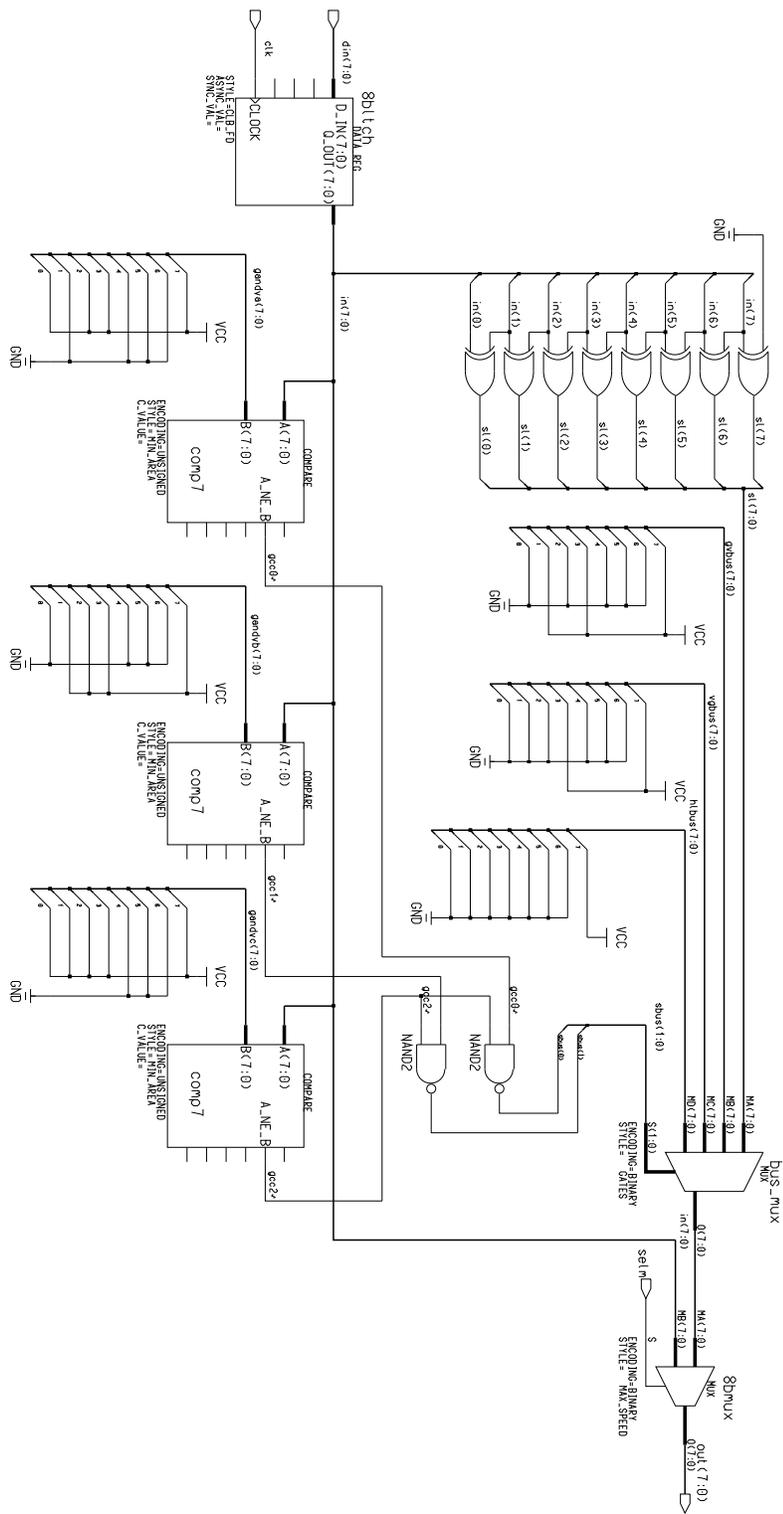


Figure 7: Grey encoder circuit.

E Parameter Download and Readback Circuit

The main unit of the parameter download and readback circuit is a shift register (figure 8). There is one shift register for each configuration parameter and three 10-bit shift registers for storing the download and readback commands. The size of the data shift registers vary from 2-bit to 13-bit, depending on the size of the word that needs to be stored. The 32×5 -bit RAM that stores the sequence of readout samples has an associated 5-bit shift register for its loading and readback.

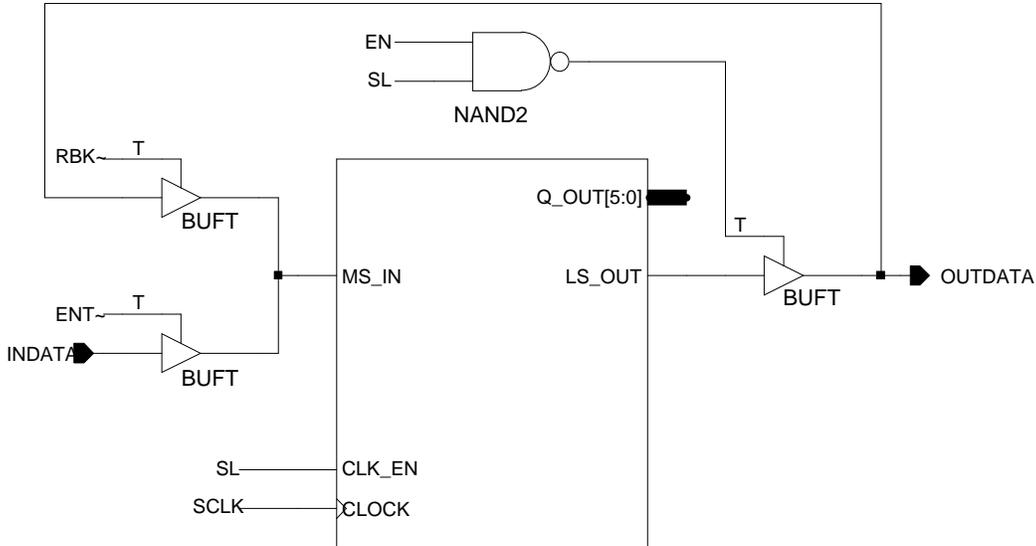


Figure 8: Data register used to hold a parameter.

The register is selected by the SL signal which enables the clock of the register. Data is loaded into the register over the serial line INDATA if ENT is high and RBK is low, i.e. we are in “enter” mode. The data is available to the rest of the circuit on the parallel output of the register. Data is read out of the register if the register is selected, ENT is low and RBK is high, and we are in “enable” mode (EN high). The read out data is looped back round into the register so that at the end of the read operation the data is still properly loaded in the register.

Figure 9 shows the data connections between the two command registers, the readback register and one data register. The first command register can receive the commands (shown in table 8) RESET_ALL, SET_ENTER, SET_ENABLE, SET_RDBK, SET_MUXE and SET_RAME. Logic will respond to the commands RESET_ALL, SET_ENTER, SET_RDBK, SET_MUXE, SET_RAME and generate signals. The first command register also responds to the data register address or RAM address. The second command register can receive all the commands but only responds to commands SET_ENABLE and RESET_ENABLE.

2. Command SET_ENTER is strobed into the first command register and causes the decoder to set ESET high. This causes ENT to go high. ENT high enables the enter stream and will allow data to go into a data register once a data register is selected. The SET_ENABLE command is now in the second command register but it has no effect.
3. Command SET_MUXE is strobed into the first command register and causes the decoder to set ESMX high. This causes MUXE to go high. The SET_ENTER command is now in the second command register but has no effect.
4. Command SET_ENABLE is strobed into the first command register but has no effect. The SET_MUXE command is now in the second command register and also has no effect.
5. The address of the data register we want to write to is strobed into the first command register. The SET_ENABLE command is now in the second command register and causes ESEN to go high and thus latch signal EN. This disables the first command register and enables a 4-to-16 decoder in figure 11 to select the data register with the address that is set in the first command register.
6. Command RESET_ENABLE is strobed into the selected data register, and possibly, out onto line OPDATA, and possibly, into the 10-bit readback register (figure 14).
7. The data for the selected data register is strobed into the register. This puts the data into the register and pushes the RESET_ENABLE command around into the second command register.
8. When done the RESET_ENABLE command causes EREN to go high and to latch signal EN high. This disables the data register input and enables the first command register input for the next sequence.

The following sequence reads back a register:

1. Command SET_ENABLE is strobed into the first command register and causes the 4-to-16 decoder to set NCLSQ high. This resets all the latches.
2. Command SET_RBK is strobed into the first command register and causes the decoder to set ESRB high. This causes RBK to go high. This enables the readback stream and will allow data to loop back around into the register once it is selected and read out. The command SET_ENABLE is now in the second command register and has no effect.
3. Command SET_MUXE is strobed into the first command register and causes the decoder to set ESMX high. This causes MUXE to go high. The command SET_RBK is now in the second command register and has no effect.
4. Command SET_ENABLE is strobed into the first command register and has no effect. The command SET_MUXE is now in the second command register and also has no effect.

5. The address of the data register we want to read from is strobed into the first command register. The command SET_ENABLE is now in the second command register and causes ESEN to go high and latch signal EN. This disables the command register input and enables a 4-to-16 decoder to select the address that is set in the first command register.
6. 10 strobes are issued to bring a hardwired RESET_ENABLE command out of the readback command register into the data register.
7. Data is strobed out and read from the selected data register. This reads out some of the register into the readback register. We then read the data and this loops the register contents back around into the data register. At the same time the data is entering the second command register until at the end of the transfer a RESET_ENABLE command is left in the second command register.
8. When done, the RESET_ENABLE command in the second command register causes EREN to go high and to latch signal EN high. This disables the data register input and enables the first command register input for the next sequence.

Similar steps are performed for loading and reading back the RAM.

E.2 Serial Download Device Select

This is the circuit (figure 11) used to select the desired data register from the register address. It also provides the clock for the latches and the interface to the sample RAM (figure 28).

A 4-to-16 decoder decodes the data register address to select the proper data register. The decoder is enabled when registers are enabled (MUXE) and data registers are enabled (EN). The current circuit allows for an expansion to 16 data registers.

To load the sample RAM, a 5-bit shift register receives the data for the RAM. The parallel out of the shift register makes the data available from the shift register to the RAM. A parallel in of the shift register allows the data to be read from the RAM into the shift register. The serial in and serial out of this register allows it to operate the same way as the other data registers.

An 8-bit shift register delays the strobe SCLK by 4 and 8 PCLKs. The 100 ns delayed strobe DSCLK operates all the latches in figure 10 except for EN. The 200 ns delayed strobe DDCLK operates the EN latch. SCLK is used to shift the instructions into the command registers, DSCLK is used to carry out what is required for the instruction (ie. set or reset a latch, etc.) and DDCLK is used to set or reset the enable (EN) latch, which we do not want to occur for a clear instruction until after the other latches have been cleared.

E.3 Serial Download Registers 0 to 9

Figures 12 and 13 show the nine registers. The shift registers accept serial input, and give serial and parallel output. All shift registers are clocked by SCLK. The input to a register can come from the serial download line PGDATA or be feed back from the readback on line RBKD. Which input is selected, is controlled by tri-state buffers. During readback the

data is read out serially and presented on serial read line OPDATA, as well as, looped back around into the serial input of the register on line RBKD.

The operating mode register M12:0 is special in that all the mode bits are held low during loading or readback of this register. This is because an all-low state is the default mode of operation and this allows the circuit to continue operating during the download and readback of the mode bits. The register address 1 is special and does not correspond to a register but instead generates a strobe which adds or deletes addresses from the pool of addresses (see figures 11 and 19).

E.4 Serial Readback and Reset

The circuit in figure 14 contains the readback command register and reset counter.

The reset counter counts the number of strobes if we are not enabled (EN low). A clear signal (ECLSQ) is generated after 168 strobes. This causes the download control latches (figure 10) to reset.

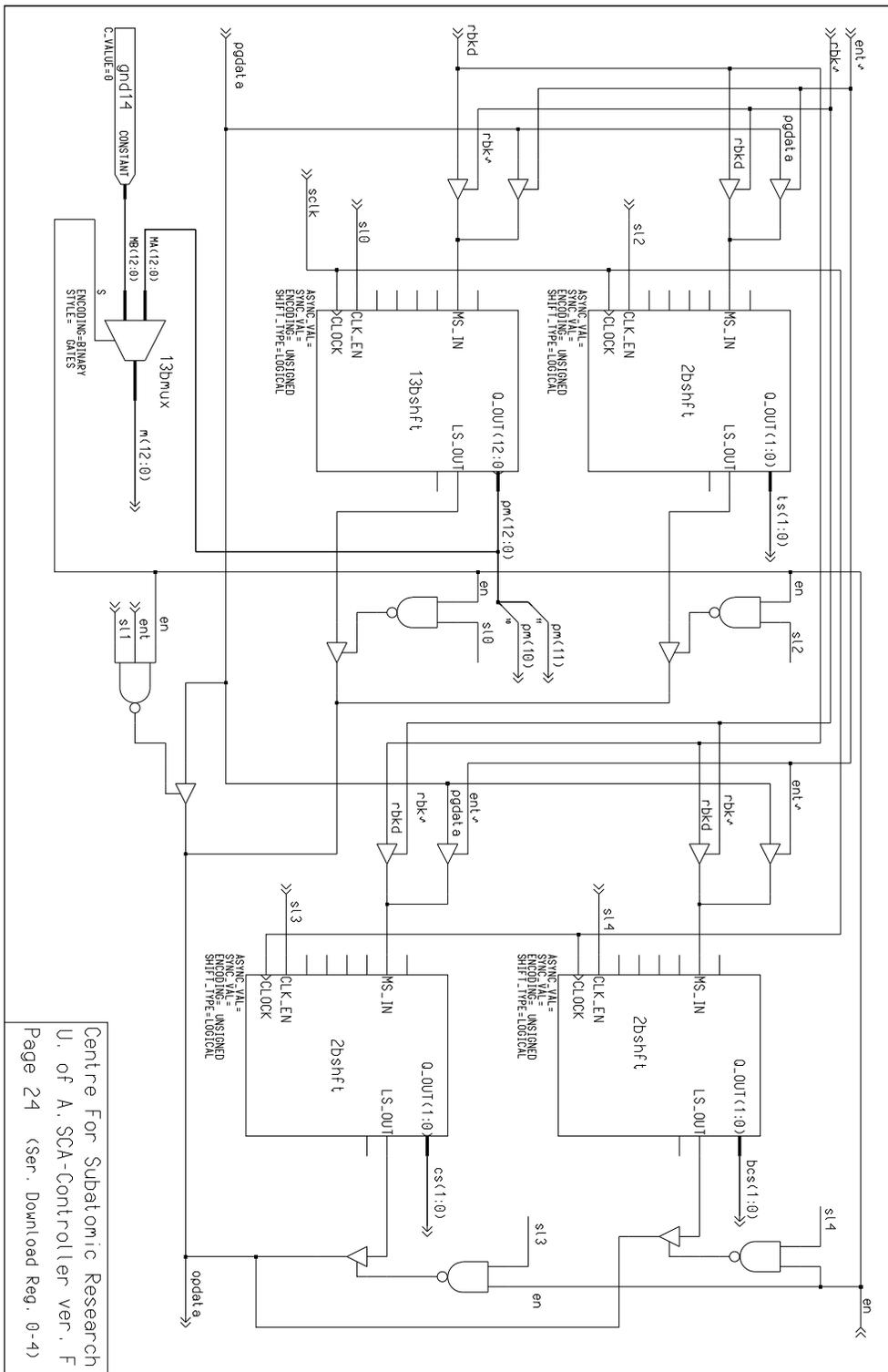
A 10-bit shift register is loaded with the RESET_ENABLE command when we are in readback mode. This command is then allowed to propagate through the data register and eventually into the second command register which will reset the enable after the data register is read back. A 4-bit counter with a maximum count of 10 disables the output buffer while the command is being shifted.

E.5 Trigger Diagnostic

The trigger diagnostic circuit is shown in figure 15. It consists of a 150-bit shift register that is loaded and read back the same as the other parameter registers. The 150-bit register is actually implemented in three 50-bit shift registers strung together. A trigger pattern is stored in this data register. It is used to load a second 150-bit shift register which is connected in parallel to the 150-bit data storage register.

The trigger diagnostic is used by setting the value three in register TS1:0. This cause a 2-to-4 decode (figure 24) to activate the trigger diagnostic circuit with signal TSEL3.

A single external trigger GB_TRIG causes the value in the data register to be loaded into the trigger register. The load is released on the falling edge of the trigger pulse and the shift begins. A bit is shifted out on every rising edge of PCLK. When 150 shifts have occurred the trigger pattern has been shifted out and low states will be constantly shifted out until the next trigger cause a reload.



Centre For Subatomic Research
 U. of A. SCA-Controller ver. F
 Page 24 (Ser. Download Reg. 0-4)

Figure 12: Serial download registers 0 to 4.

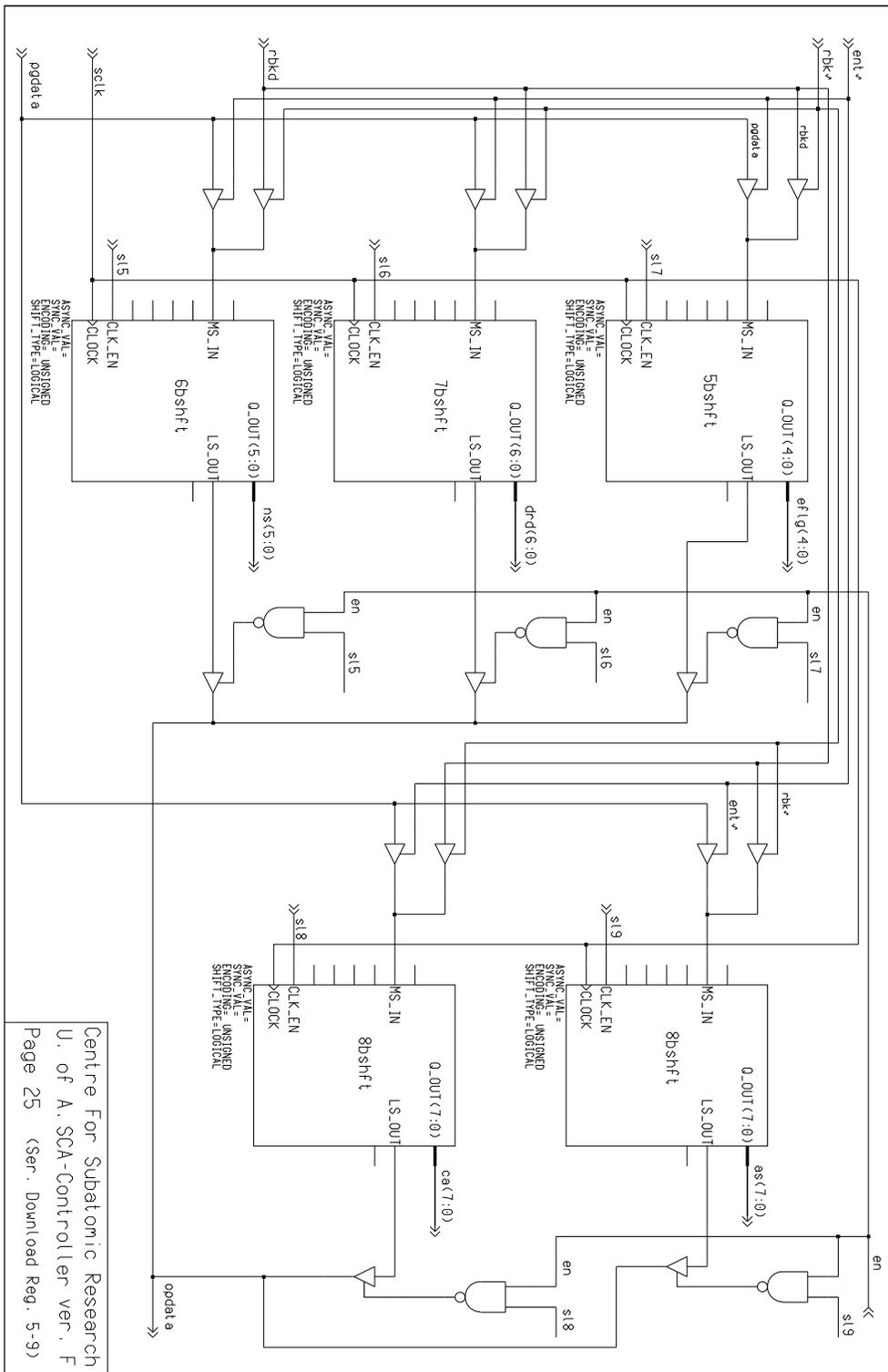
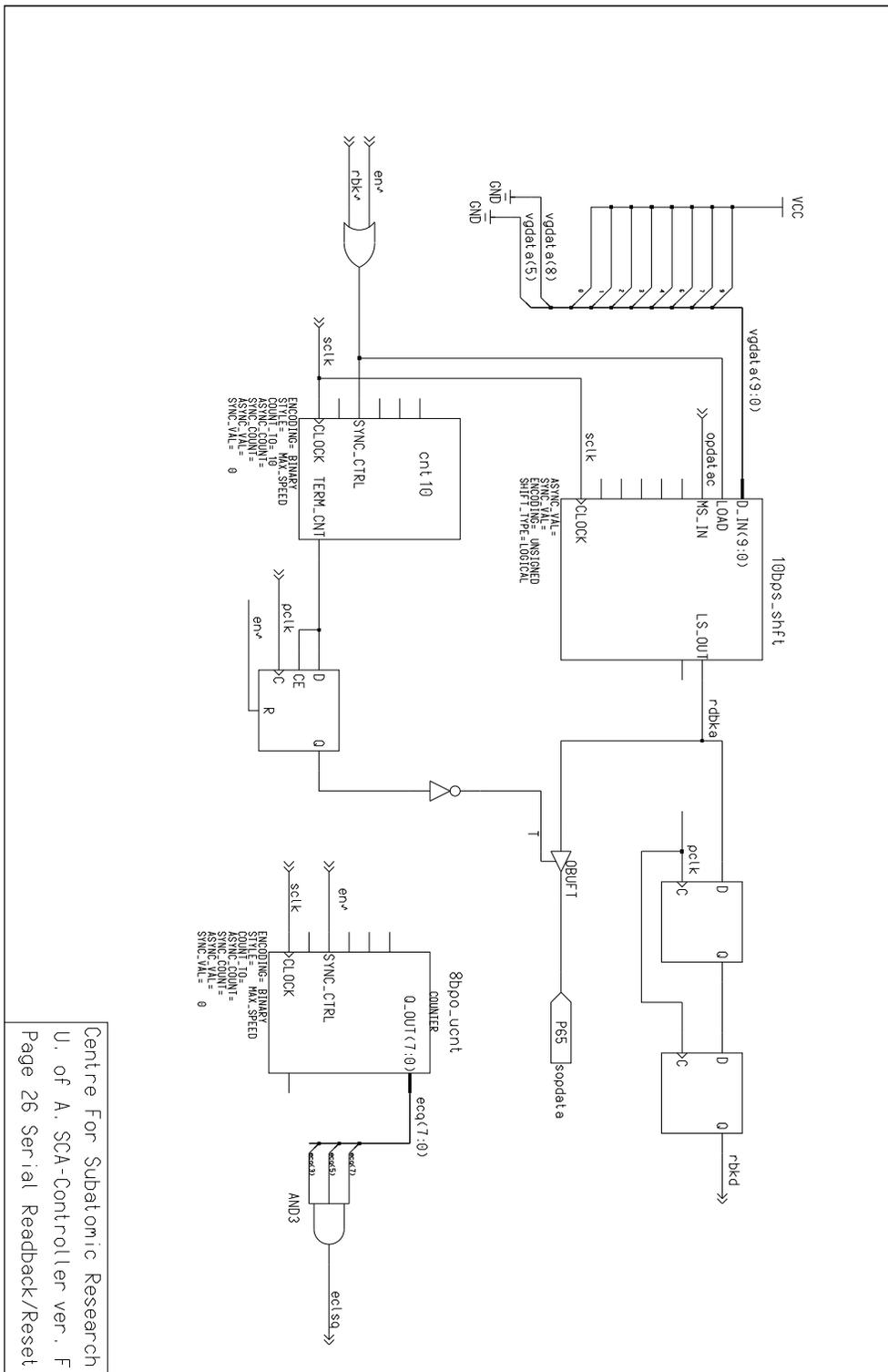


Figure 13: Serial download register 5 to 9.



Centre For Subatomic Research
 U. of A. SCA-Controller ver. F
 Page 26 Serial Readback/Reset

Figure 14: Serial readback and reset.

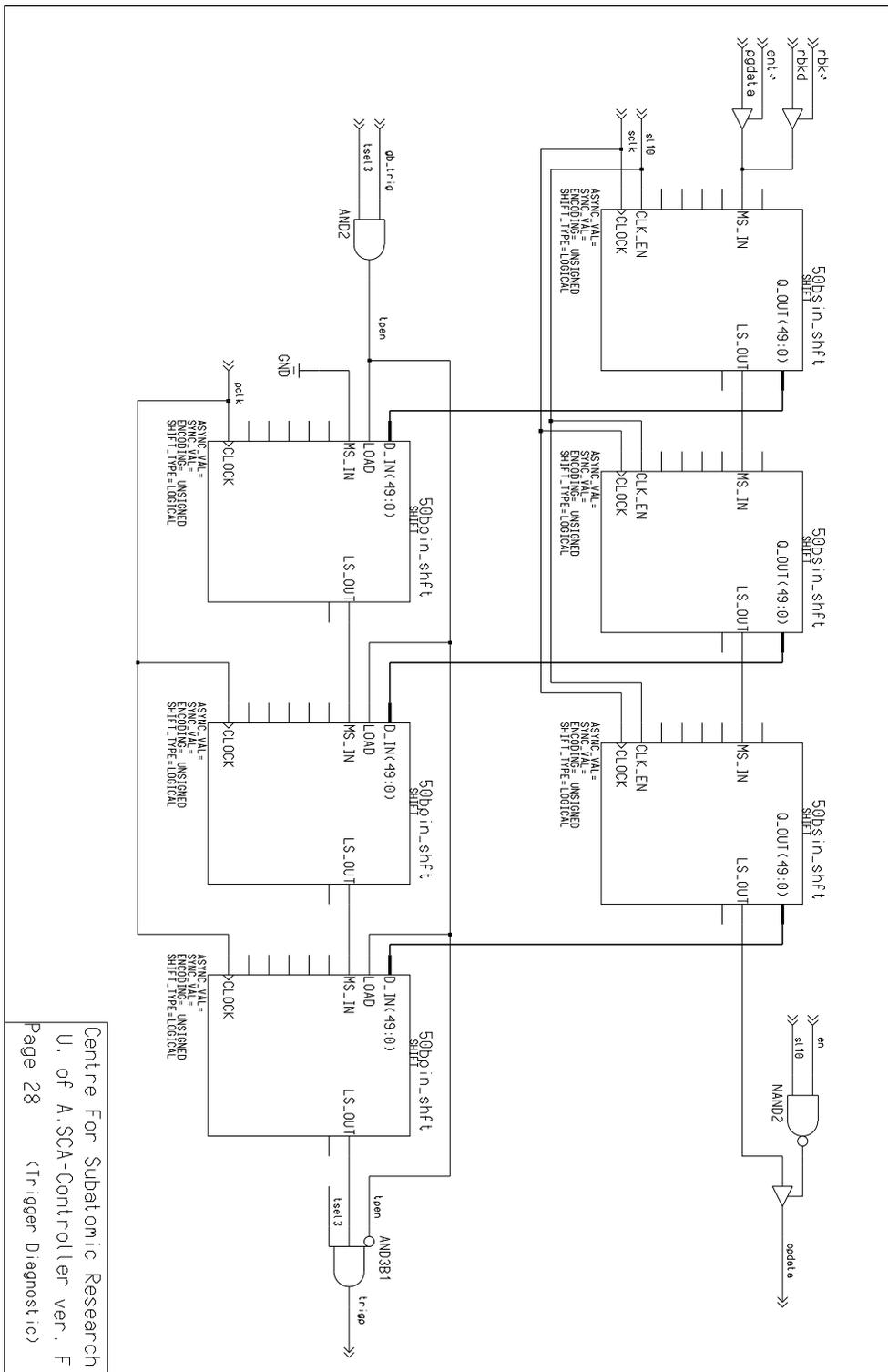


Figure 15: Trigger diagnostic.

F Description of Circuit

F.1 Reset Generation

The reset circuit (figure 16) generates two signals (RLRT and DRLRT) that set the controller into a well defined state and load the FIFOs. Signal RLRT clears the FIFOs.

The reset can be initiated from one of four different sources. The source of the reset is controlled by a 2-to-4 decoder which is set by register CS1:0.

The reset pulse ARST must be high for at least 25 ns and may be longer. The rising edge of ARST is shaped to a 25 ns wide pulse RSCLK which occurs within 25 ns of the rising edge of ARST. The reset and reload signal RLRT goes high on the falling edge of RSCLK and stays high for 200 ns. The delayed reset and reload pulse DRLRT also goes high on the falling edge of RSCLK and stays high for 225 ns to 400 ns.

F.2 Startup Circuit

The startup circuit (figure 17) generates some key signals that allow the loading of the write FIFO and read-delay FIFO. The signals are as follows.

LDC enables the write FIFO load counter.

LCTC disables the write FIFO load counter.

DLDC enables the write FIFO write (push) during loading.

LDD selects the multiplexer so that addresses are input to the write FIFO from the load counter.

ERWF starts the read-delay FIFO load counter.

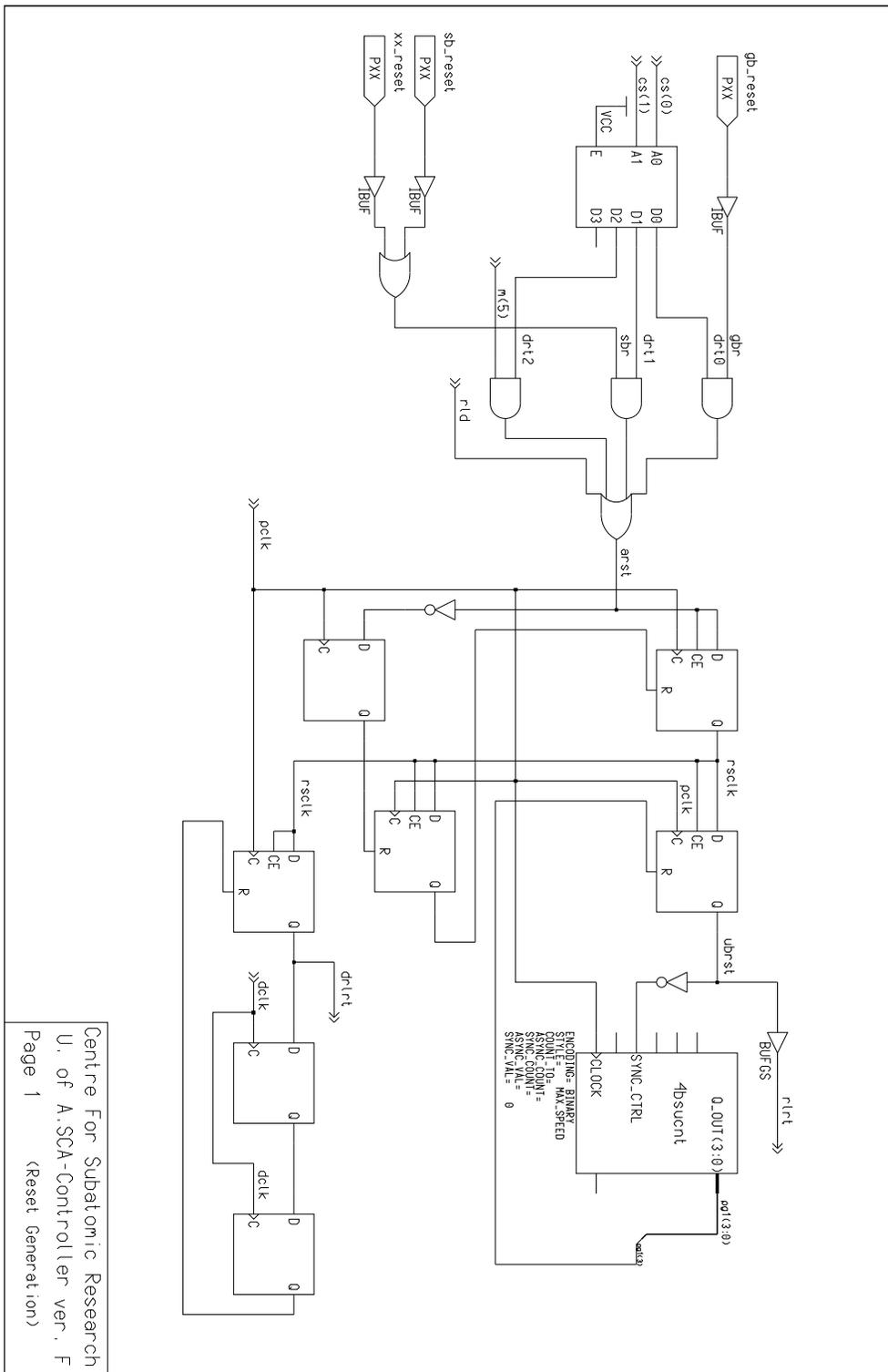
EWRDF enables the write of the read-delay FIFO (push).

DRDTC disables the read-delay FIFO load counter.

WFWEB is the write FIFO write enable after loading (push) and the write enable of the read-delay FIFO (pop).

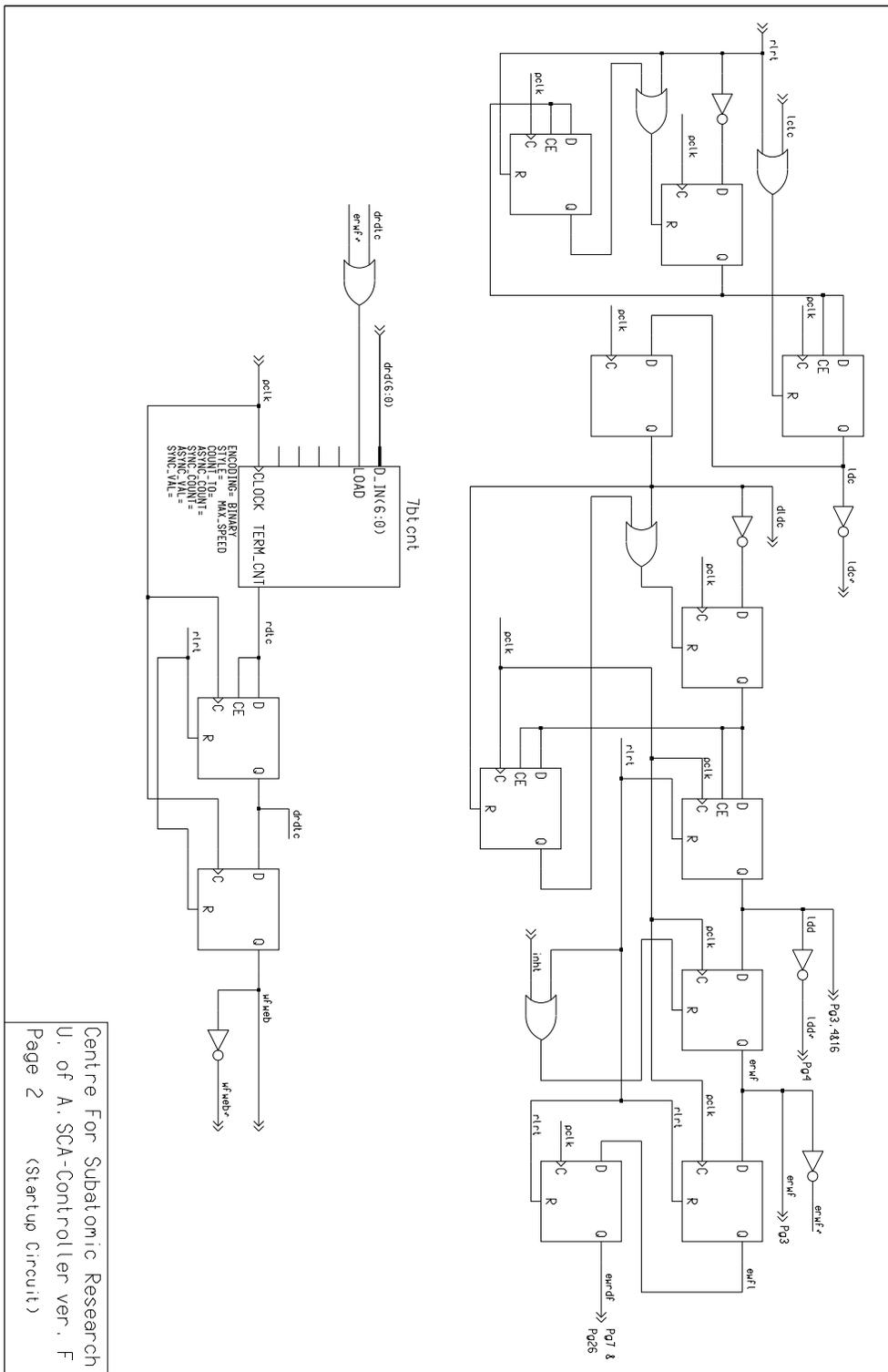
50 ns after the reload and reset pulse RLRT is released, the load counter signal LDC goes high. This causes an 8-bit counter with a maximum count of 144 (figure 18) to start counting off the addresses that are loaded into the write FIFO. 25 ns after the counter starts, the push of the write FIFO is enabled with signal WFWE (figure 19). The write FIFO write enable signal WFWE is derived from a delayed load counter signal DLDC during loading. WFWE will be the same as DLDC during loading if the current address is not marked for deletion.

When the terminal count is reached the load counter terminal count signal (LCTC) goes high and we are finished loading all the addresses into the write FIFO. Signal LDC goes low and the counter is disabled. 25 ns later the push of the write FIFO WFWE goes low and disables further loading.



Centre For Subatomic Research
 U. of A.SCA-Controller ver. F
 Page 1 (Reset Generation)

Figure 16: Reset generation.



Centre For Subatomic Research
 U. of A. SCA-Controller ver. F
 Page 2 (Startup Circuit)

Figure 17: Startup circuit.

The terminal count signals LCTC is also latched and causes the write FIFO read enable signal WFRE to go high and addresses to move from the write FIFO into the read-delay FIFO. A further 75 ns later the signal ERWF goes high and starts the 7-bit counter to count down the trigger delay time DRD6:0. This counter is used to count off the number of addresses that are moved from the write FIFO into the read-delay FIFO. Signal EWRDF also goes high 50 ns after the delay counter is enable. This signal is the push of the read-delay FIFO. Addresses are now moving from the write FIFO into the read-delay FIFO.

When the terminal count of the read-delay counter is reached, RDTC pulses high and is latched as signal DRDTC. This signal disables the read-delay counter and causes WFWEB to go high 25 ns later. The WFWEB signal is now used as the signal that enables the pop of the write FIFO and is equal to the write FIFO write enable, provided we are not transferring addresses from the read-delay FIFO into the read FIFO. This means that addresses are being transferred back into the write FIFO from the read-delay FIFO.

The WFWEB signal is the pop of the read-delay FIFO. So now the write FIFO is loaded with all the available addresses and the read-delay FIFO has an amount of the addresses that corresponds to the read delay time. The push and pop of both the write FIFO and read-delay FIFO are now enabled. If no other action occurs, the addresses will just continue to cycle through the two FIFOs.

F.3 Load Write FIFO

This circuit (figure 18) loads the addresses into the write FIFO. When the load control signal LDC (figure 17) goes high an 8-bit counter with a maximum count of 144 starts counting through the addresses. The output of the counter addresses the RAM in the address deletion circuit (figure 19) and is latched for input to the write FIFO. An 8-bit wide 2-to-1 multiplexer is used to chose between the address stored in the latch or the address from the read-delay FIFO. During loading the address from the latch is chosen while during normal running the address from the read-complete FIFO is chosen. The multiplexer is controlled by the signal LDD in figure 17. The output of the multiplexer is input to the write FIFO.

When the counter terminates and the write FIFO is loaded, the terminal count cause the write FIFO read-enable signal WFRE to be latched high. This enables the pop of the write FIFO, provided addresses from the write FIFO are not being inhibited by address sequencer (WMFFF is low).

The addresses flowing from the write FIFO to the address sequencer are split into two streams. This allow the address sequencer to work at effectively half the clock frequency. This was found necessary when the design was targeted for early versions of the Xilinx FPGA.

Two 8-bit latches receive the addresses from the write FIFO and make them available to the address sequencer (figure 6). A toggling latch alternates between enabling the clocks of the two 8-bit latches.

F.4 Address Deletion

This circuit (figure 19) has two pieces. One piece controls the push of the write FIFO and is really part of the startup circuit in figure 17. The write FIFO write enable bit (WFWEB) is

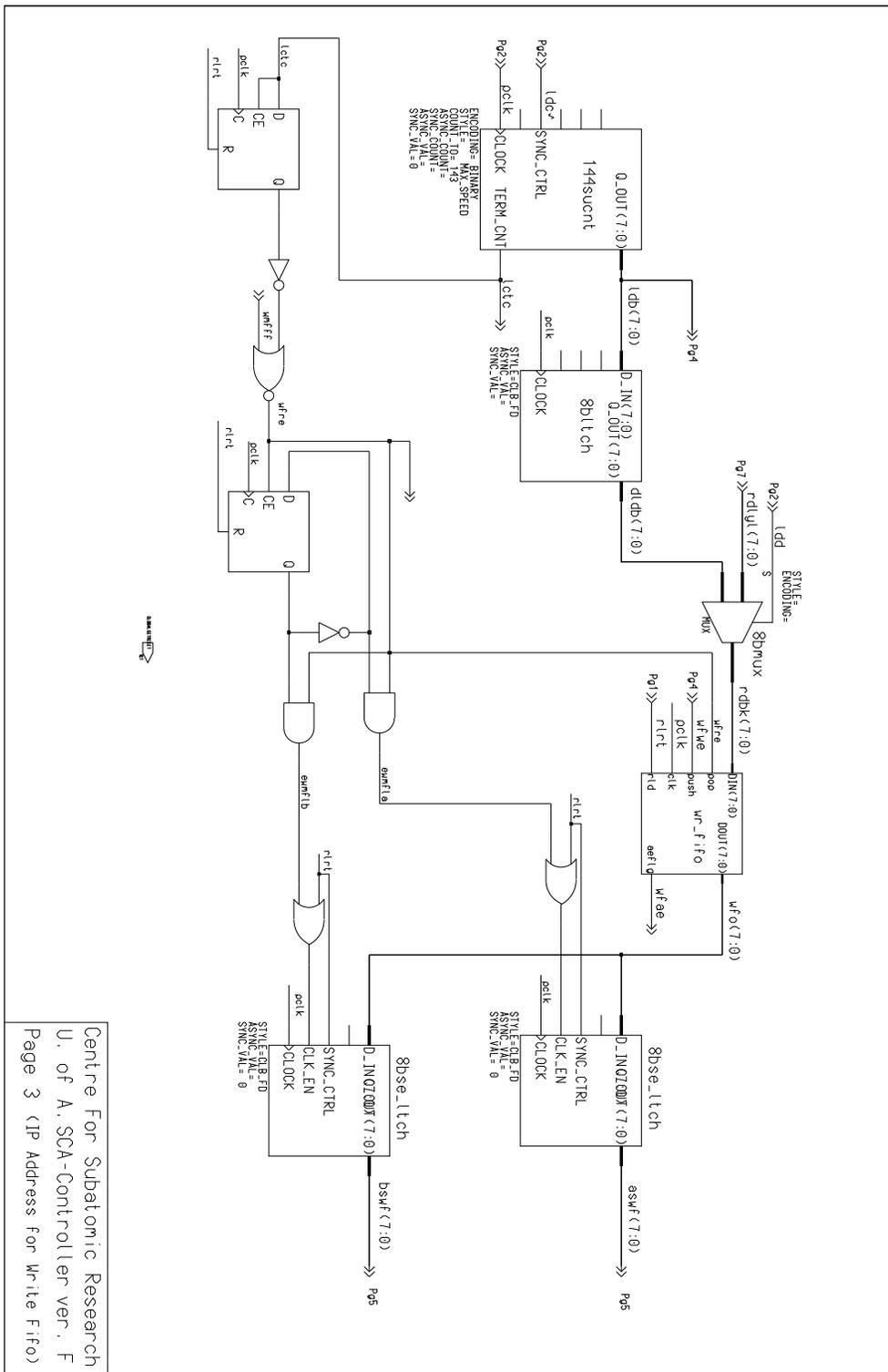


Figure 18: Input address for write FIFO.

latched on the rising edge of PCLK. The output of this latch will generate the write FIFO write enable signal (WFWE), if the read FIFO write signal (RWF) is low. Signal WFWE is the push of the write FIFO and signal RWF is the push of the read FIFO. This part of the circuit is used during normal operation and cycles addresses from the read-delay FIFO back into the write FIFO.

The second piece of the circuit in figure 19 is capable of removing addresses from the pool of available addresses. A 144×1 -bit RAM is used to store a bit for each possible capacitor address. If the bit is high, the address corresponding to that bit will be deleted from the pool of available addresses. The read address of the RAM is controlled by a counter that steps through the possible addresses (figure 18). Writing to the RAM is disabled while the counter is active (signal LDC high). The output of the RAM is used to drive logic to enable or disable the write FIFO write enable. If the current address is to be deleted and the address deletion circuit is enabled (mode bit M10 high) and addresses are being multiplexed from the address counter (LDD low), then the write FIFO write enable signal (WFWE) will be low; else WFWE will be high and the address will be loaded into the write FIFO.

The RAM should be loaded before a circuit reload and reset signal is issued. The address to be deleted is loaded into register AS7:0. This address becomes the write address of the RAM. To delete this address a 1 is loaded into the RAM at this address by setting mode bit M11 high. This value is strobed into the RAM by issue a strobe from the serial download circuit (STRB). The strobe STRB will be shaped to a 25 ns wide pulse and strobe the data at M11 into RAM address AS7:0. This will tag the address for deletion.

The RAM starts up in a cleared state and hence all bits are low and thus all addresses will be available by default. It is also possible to add back an address that has been previously deleted by following the same procedure as above but setting the mode bit M11 low. The protocol for deleting an address is as follows.

1. set address in AS7:0,
2. set M11=1,
3. issue strobe (STRB),
4. set M10=1,
5. issue reset signal.

F.5 SCA Write Address

This circuit (figure 20) makes the write addresses available to the SCAs. An 8-bit wide 2-to-1 multiplexer picks between the write address or a single address stored in register CA7:0. If the mode bit M8 is set high the address stored in register CA7:0 will be selected. A Grey encoder encodes the address if mode bit M3 is not set high. The final address is latched into one of two 8-bit latches on the rising edge of PCLK. The choice of which 8-bit latch will receive the address is set by an external pin (SWA_PORT_SEL). The output latches are disabled if signal WAINHT is high. The output of the 8-bit latches are put onto fast output pins of the FPGA.

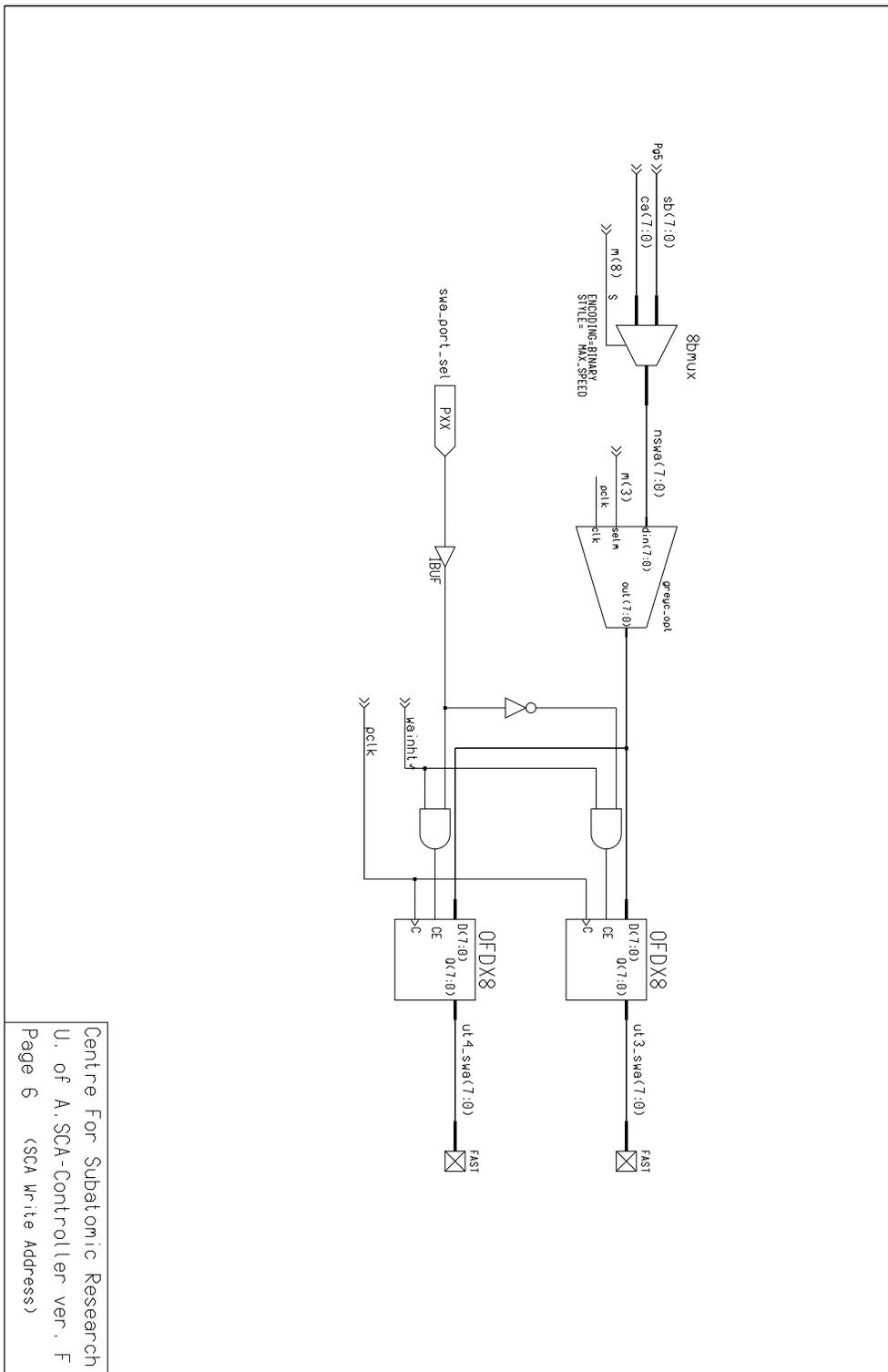


Figure 20: SCA write address.

F.6 Read-Delay FIFO and Read FIFO

This circuit (figure 21) covers the read-delay, read and bunch-crossing number FIFOs. The read-delay FIFO stores the number of addresses equal to the trigger latency. It receives addresses from the selected address SB7:0 from the address sequencer logic. There is an 8-bit latch between the read-delay FIFO and read FIFO that allows sampling of the addresses for placement back into the write FIFO. The read FIFO and bunch-crossing number FIFO are controlled by the same logic and run in parallel. The bunch-crossing number FIFO stores the 12-bit bunch-crossing number BCN11:0 as well as the flags for the first sample FSAMP, the last sample LSAMP and the event flag EVFLAG. The pop of the read FIFO can be inhibited by the empty flag of the FIFO or the multiplexing control logic.

The output of both the read FIFO and bunch-crossing FIFO are each put into pairs of 8-bit latches on alternate rising edges of PCLK. The outputs of all four latches and some control signals are input to the component shown in figure 22. This unit multiplexes the addresses and bunch-crossing numbers back onto single busses. This part of the circuit serves no function and was implemented in a early design to allow a better place and route. The multiplexer select logic has a 2-bit counter and toggling latch, and operates similarly to that of the address sequencer multiplexing logic (figure 6).

F.7 Bunch Crossing Counter

This circuit (figure 23) generates the bunch crossing number. A 12-bit binary counter counts on the rising edge of PCLK. When the trigger is received (signal TRIGA), a 12-bit latch is enabled to receive the value of the counter on the next rising edge of PCLK. The output of the latch is the input to the bunch-crossing number FIFO (figure 5). The latch is disabled during the time that the reset signal RLRT is high. The bunch-crossing counter can be reset by three different sources that are controlled by a 2-to-4 decoder. The decoder selection is set with register BCS1:0. Regardless of the source of the reset, the rising edge is shaped to a 25 ns pulse.

F.8 Read Trigger

This circuit (figure 24) forms the read trigger for the controller. The trigger can be initiated from four different sources: LVL1, M7, board, TRIGP. Which source is active is determined by a 2-to-4 decoder. The decoder is set by register TS1:0. TRIGP is a diagnostic trigger source. Regardless of the source of the trigger, the trigger pulse is shaped to a width of 25 ns when an edge is received. The trigger signal for the circuit is TRIGA.

The trigger signal TRIGA causes the read-FIFO write (RFW) to be enabled. The first sample signal FSAMP is generated 50 ns after the trigger signal TRIGA. A trigger signal TRIGA causes a 5-bit counter to be loaded which starts to count down the number of samples (minus one). 25 ns and 50 ns after the terminal count of the counter is reached, the last sample signal ELSAMP will go high and the read-FIFO write will be disabled, respectively. The counter will cause triggers to be inhibited if overlapping events are not allowed, mode bit M12 high.

Several different sets of conditions can cause the triggers to be inhibited.

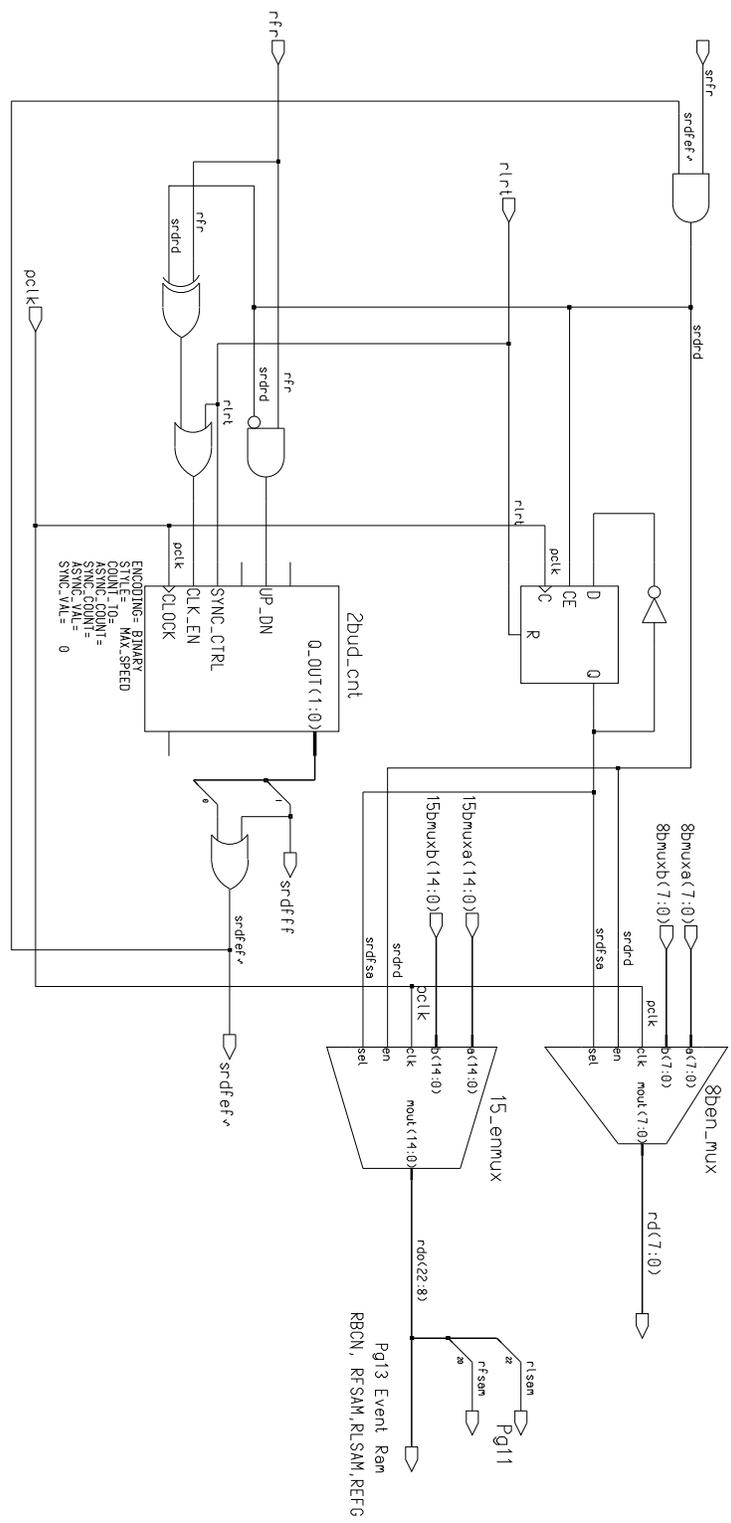


Figure 22: Component U10.

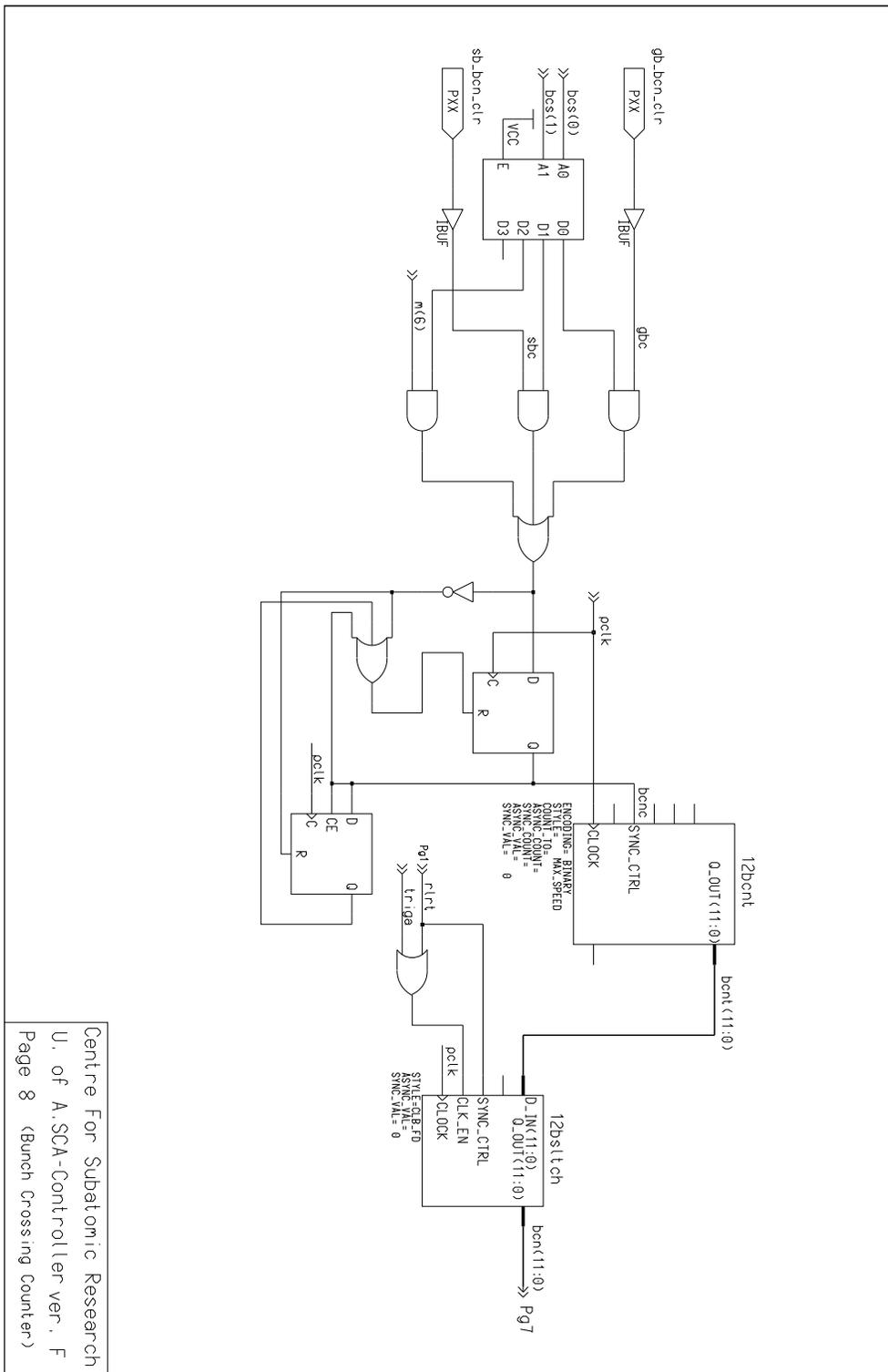


Figure 23: Bunch crossing counter.

NSTL The sample register is set for more than 32 samples: NSTL low

INHT The write FIFO is almost empty and we are not loading the FIFO.

WFWEB We have not yet load the read-delay FIFO.

M0 Bit 0 in the mode register is set.

M12 Bit 12 in the mode register is set and not all the samples have been transferred to the read FIFO: RFWA high.

M1 Bit 1 in the mode register is set and the readout is not complete.

M2 Bit 2 in the mode register is set and we are reading.

TMOLE The number of samples is set below 9 and we have more 24 samples pending readout because of multiple triggers.

Overlapping events are only allowed if the number of samples is less than 9. Reading out more than 8 samples will normally be a diagnostic mode which will not generate overlapping events. However, if we are allowing overlapping events, we want to avoid a steady stream of them. After the number of samples pending readout reaches 16 we inhibit triggers until the current number of samples drops below 16.

F.9 Generate Last Sample

This circuit (figure 25) generates the last sample flag (LSAMP). The number of samples stored in register NS5:0 is compared with the value of 32. If the number of samples is greater than 32 the triggers are inhibited with signal NSTL. For a valid number of samples, a value of one is subtracted from the number of samples and this number (ENS4:0) is used to load the sample count-down counter in the read trigger logic (figure 24).

If a second trigger does not occur before the samples are counted down, or the number of samples is greater than 8, then the last sample flag is simply given by ELSAMP from the sample count-down counter. If a second trigger occurs before the samples are counted down, and the number of samples is less 9, then the last sample flag is generated from the trigger signal TRIGA propagating through the shift register. This represents the last sample from each trigger. A last sample signal is also generated from the sample count-down counter, which has been reload when the second trigger occurred. Thus for overlapping events, the last sample of each event is tagged.

F.10 Load Read Buffer RAM

This circuit (figure 26) loads a RAM with the read addresses and thus allows the addresses to be re-ordering. The RAM is dual ported and is 8-bit wide by 32-bit deep to allow the maximum number of samples to be stored. The write addressing of the RAM is controlled by the read-capture logic shown in figure 27. The read addressing of the RAM is controlled

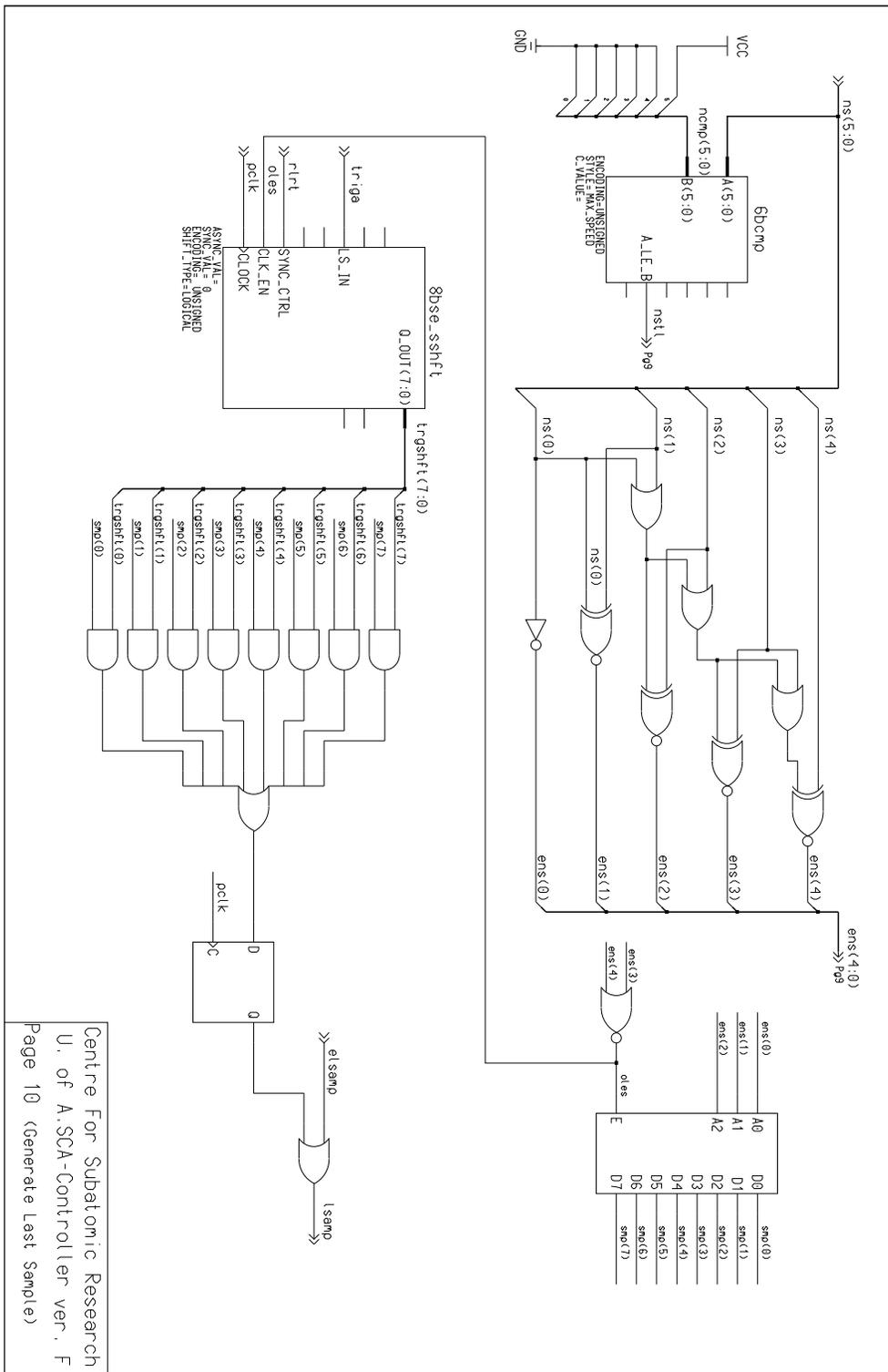


Figure 25: Generate last sample.

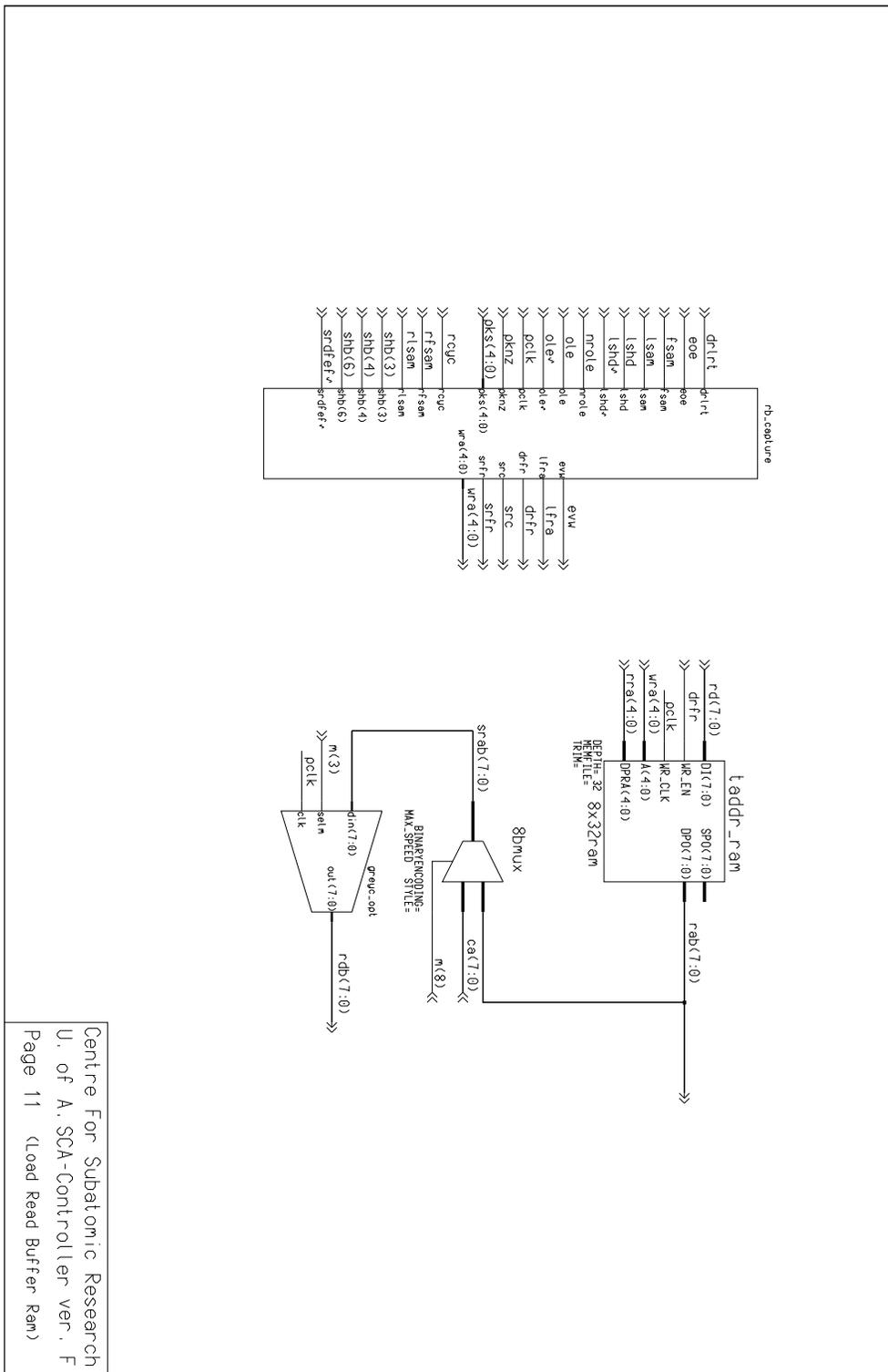


Figure 26: Load read buffer RAM.

by the RAM that is loaded with the read sequence at the time when the parameter registers are downloaded (figure 28).

The output of the RAM goes to the read-complete FIFO (RAB7:0) and is presented as external output signals. If mode bit M8 is set, the single address stored in register CA7:0 will be output. The external outputs will be Grey encoded if mode bit M3 not set.

F.11 Read Address Sequencer

The read address sequencer (figure 28) orders the read addresses. A 5-bit wide by 32-bit deep RAM is load with the read sequence before the run. The loading is done over the input port RSDT4:0 and the readback over one of the output ports RRSP4:0. The other output port RRA4:0 provides the read addresses for the read buffer RAM in figure 26. If the address from the read sequence RAM is not zero, signal PKNZ goes high.

The outputs signals are generated from the following logic:

$$\mathbf{LSAM4} = (\mathbf{LSAM} \circ \overline{\mathbf{FSAM}} \circ \mathbf{SHB4}) + (\mathbf{FSAM} \circ \mathbf{LSAM} \circ \overline{\mathbf{OLE}} \circ \mathbf{SHB5}),$$

$$\mathbf{LSHD} = \mathbf{LSAMP4} \circ \overline{\mathbf{OLE}},$$

$$\mathbf{EOE} = \mathbf{LSHD} \circ \mathbf{SHB7},$$

$$\mathbf{GSHB} = \overline{\mathbf{LSHD}} \circ \mathbf{SHB6} \text{ and}$$

$$\mathbf{NXTS} = \overline{\mathbf{LSHD}} + (\mathbf{LSHD} \circ \mathbf{EVW}).$$

A 5-bit counter steps through the read sequence RAM and event RAM. The synchronous control of the counter is set with logic

$$\mathbf{CLRR} + (\overline{\mathbf{OLE}} \circ \mathbf{LSAM4}),$$

and the clock enable of the counter is set with logic

$$\mathbf{CLRR} + (\mathbf{FSAM} \circ \mathbf{LSAM} \circ \overline{\mathbf{OLE}} \circ \mathbf{SHB5}) + \overline{(\mathbf{FSAM} \circ \mathbf{LSAM} \circ \overline{\mathbf{OLE}})} \circ \mathbf{SHB4},$$

where $\mathbf{CLRR} = \mathbf{EOE} + \overline{\mathbf{SRC}} \circ \overline{\mathbf{RCYC}}$.

F.12 Event RAM

This circuit (figure 29) reorders the event information to correspond to the order of the samples read out. A 15-bit wide by 32-bit deep RAM stores the information to be sent to the gain selector logic. The RAM holds the signals LSAM, RAEFLG, FSAM and the 12-bit bunch-crossing number. The event data is loaded from the bunch-crossing number FIFO. The data is written in a non-sequential order given by the read-capture circuit (figure 27) which issues the RAM write address WRA4:0 and write enable signal DRFR. The data is read out from the RAM sequentially with address RSA4:0 which is generated by a counter from the read-address sequencer circuit (figure 28). The RAM is controlled in parallel with the load read buffer RAM in figure 26.

If FSAM is low all outputs for the bunch-crossing number are grounded. This is because the brunch-crossing number is only output for header operations (FSAM high).

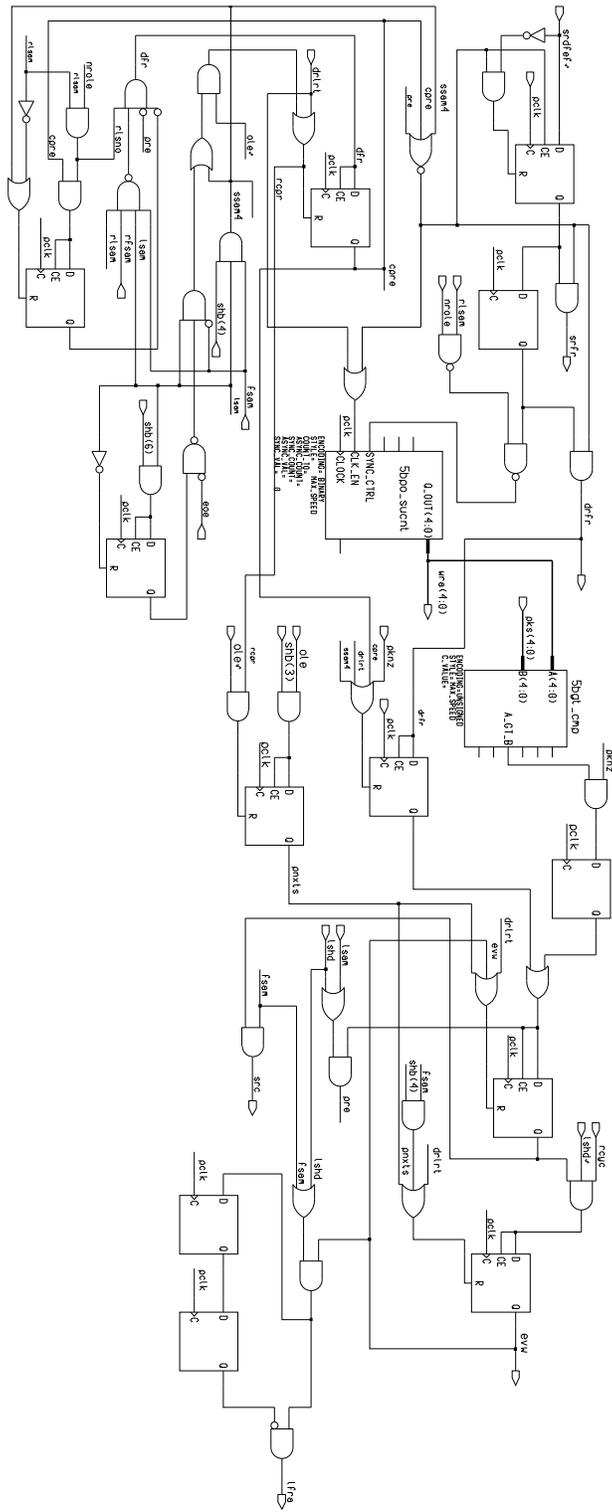


Figure 27: rb_capture.

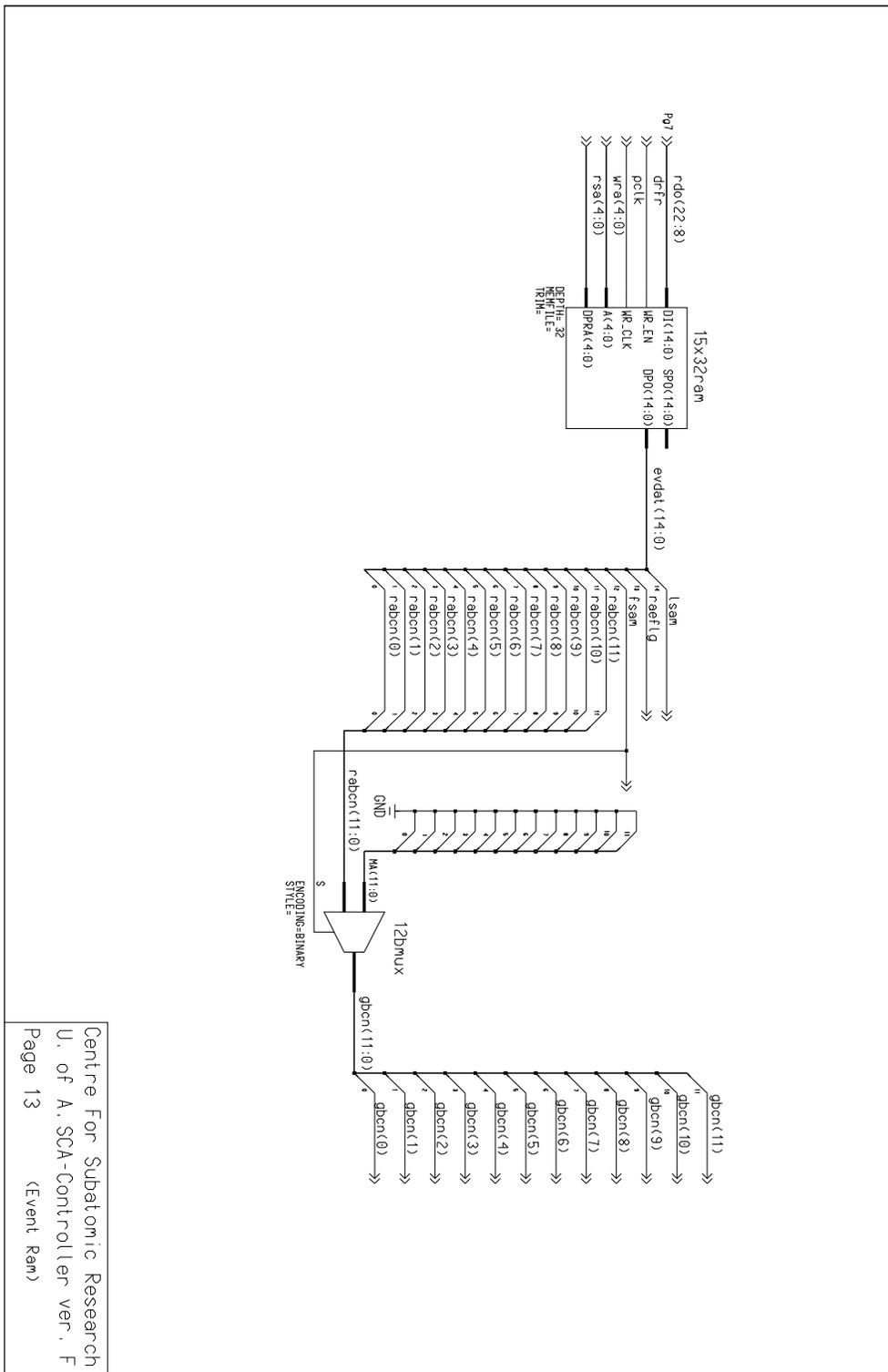


Figure 29: Event RAM.

F.13 Readout Control

This circuit (figure 30) receives the ready signal from the gain selector logic and outputs the frame bit FR to the gain selector logic, enables the output latches to the gain selector logic, pulses the push of the read-complete FIFO (RCWE) and generates the timing signals SHB15:0.

READY goes high when the gain selector logic can accept another packet. If the logic $(ECSY \circ \overline{CSYNC} \circ RCYC) + (NXTS \circ SHB7 \circ RCYC)$ is satisfied, the frame bit FR will be output on the falling edge of DCLK. The FR bit is formed from a pulse that is 100 ns wide.

A 16-bit shift register generates some key timing signals. If we have an event that is not overlapping and no other event pending, a 1 is shifted into the register. This causes SHB0 to go high on the next clock and causes the clock enable of the shift register to go low until we receive a READY. A set bit in the shift register moves through until it gets to SHB7. If there is another sample, SHB0 will set again, and we will wait for a READY to shift that bit through. This will continue until NXTS is low, which occurs if EVW is low.

Signals SHB3, SHB4, SHB5 and SHB6 are used to determine whether we have an overlapping event, whether there is another event waiting, and to update the sequence and RAM counters.

SSHE latches high when SHB0 is set. This disables the external output shift registers from loading and enables the output latches in figure 31. The signal SSHE is disabled when we reach the end of the shift and signal SHB7 goes high.

The READY signal also causes a low to shift through a 12-bit shift register which will cause a pulse on the falling edge of the shift register output, 12 DCLKS later. This falling edge will pulse signal RCWE which will enable the read-complete FIFO write enable.

F.14 External Output Signals

This circuit (figure 31) presents the serial outputs cn7_0, bn7_0 and st7_0 to the gain selector logic. The capacitor address cn7_0 corresponds to internal signals RDB7:0 from the load buffer RAM (figure 26). The lowest order eight bits of the bunch-crossing number bn7_0 corresponds to internal signals GBCN7:0 from the event RAM (figure 29). The status bits st7_0 correspond to the internal signals STAT7:0. Currently only the three lowest order status bits are used. In order of increasing significance, they correspond to signals SERR, INHT and RAEFLG. SERR indicates that the addresses have gone out of order and comes from the error detection logic (figure 36). INHT indicates that the controller is inhibiting triggers because the write FIFO has gone almost empty. RAEFLG indicates that the current trigger occurred within EFLAG ns of the last trigger.

The three latches receiving the data are clocked on the rising edge of PCLK. The clock-enable of the latches are generated by signal LDDL = LFRA + GSHB6. The outputs from the three latches are loaded into the 8-bit shift registers on the rising edge of the read clock DCLK. The load of the shift registers is enabled by signal LDSH = SHB7 + \overline{SSHE} . The falling edge of DCLK latches the outputs from the shift registers and makes them available to the output pins. The output latch only latches the data if signal SSHE is high, else only low states are present on the output pins.

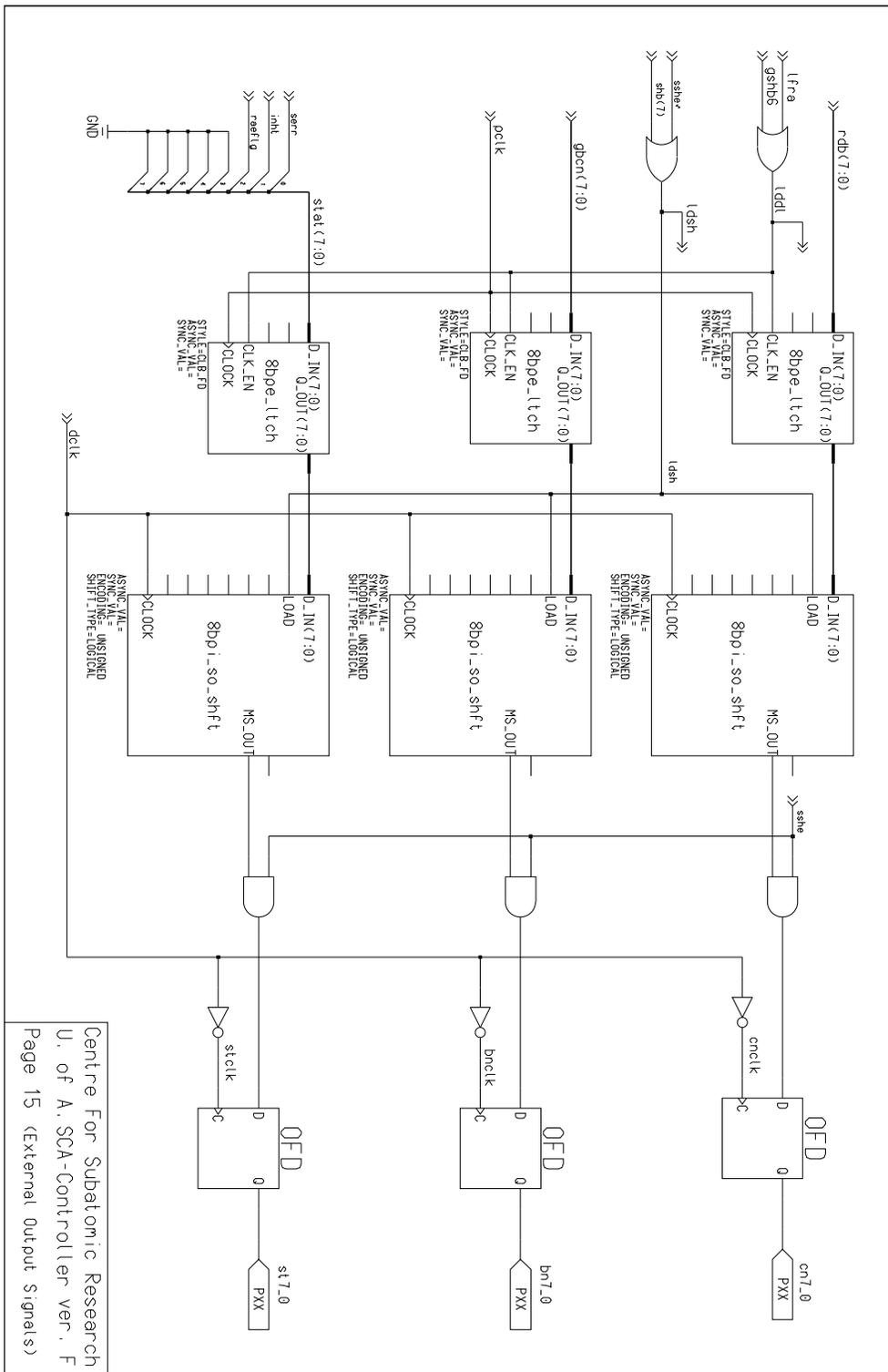


Figure 31: External output signals.

F.15 Read Control

The circuit in figure 32 shifts out the external signal rc7_0. The four most significant bits of the 12-bit bunch crossing number are put into the lower four bits of an 8-bit latch. The control bit C0 will be described in a moment. The control bit C1 is not used and ground. The operation code OP1:0 is formed from the signals FSAM and LSAM according to table 6. The data are latched, loaded and shifted out with the same signals as the circuit in figure 31.

The signal C0 (or OLE) is high when the current sample is shared between two events. Logic recognizes the first sample and causes a latch to latch high. If a sample comes with both the first and last sample flags set, the OLE signal goes high. If a sample comes with the first sample flag set and the latch indicates that this is not the first sample, then the OLE flag is set. If a last sample flag occurs, the latches are reset and OLE goes low.

The signals RFSAM, RLSAM and RFR are used to generate the signal NROLE. If the read FIFO read signal RFR is high, the signal NROLE will respond to signals RFSAM and RLSAM. NROLE is normally high and will go low if only the last sample flag (RLSAM) goes low. It will remain low until only the first sample flag goes high (RFSAM).

The circuit in figure 33 generates the read control signals RCYC, ECSY and CSYNC. It also recognizes if the write FIFO is almost empty and can cause an inhibit or reload.

Signal SRC from figure 27 is latched to give signal RCYC. The signal RCYC is cleared with logic $\text{EOE} \circ \overline{\text{NXTS}}$. The signal LFRA from figure 27 is latched and then synchronized with the read clock DCLK to give signal ECSY. One DCLK cycle later CSYNC is also latched high. Signal CSYNC resets signal ECSY. The falling edge of signal RCYC resets signal CSYNC.

If the write FIFO almost empty flag goes high (we are almost out of available addresses) and we are not loading the write FIFO, an inhibit signal (INHT) is generated. This inhibit signal will cause the address sequencer to not accept any more addresses from the write FIFO and will disable the push of the read-delay FIFO. The inhibit can only be cleared by a circuit reload and reset signal RLRT.

F.16 Event Flag

This circuit (figure 34) causes an external signal (EVFLGA) to be generated if two triggers occur within a predefined time period. The first trigger signal TRIGA is latched on the rising edge of PCLK. TRIGA causes a 5-bit counter to be loaded with the event flag time stored in register EFLG4:0. On the next rising edge of PCLK the counter begins to count down. If a second trigger signal TRIGA occurs before the counter has terminated, a latch is set and the signal EVFLAG goes high. Both latches are reset when the counter terminates or when a system reset RLRT is issued. The absence of the first trigger causes the counter to continuously load.

F.17 Read-Complete Latch and FIFO

This circuit (figure 35) stores the addresses waiting to go back into the pool of available addresses. The addresses from the read buffer RAM (RAB7:0) are latched three times before they are placed into the read-complete FIFO. These latches delay the data in order

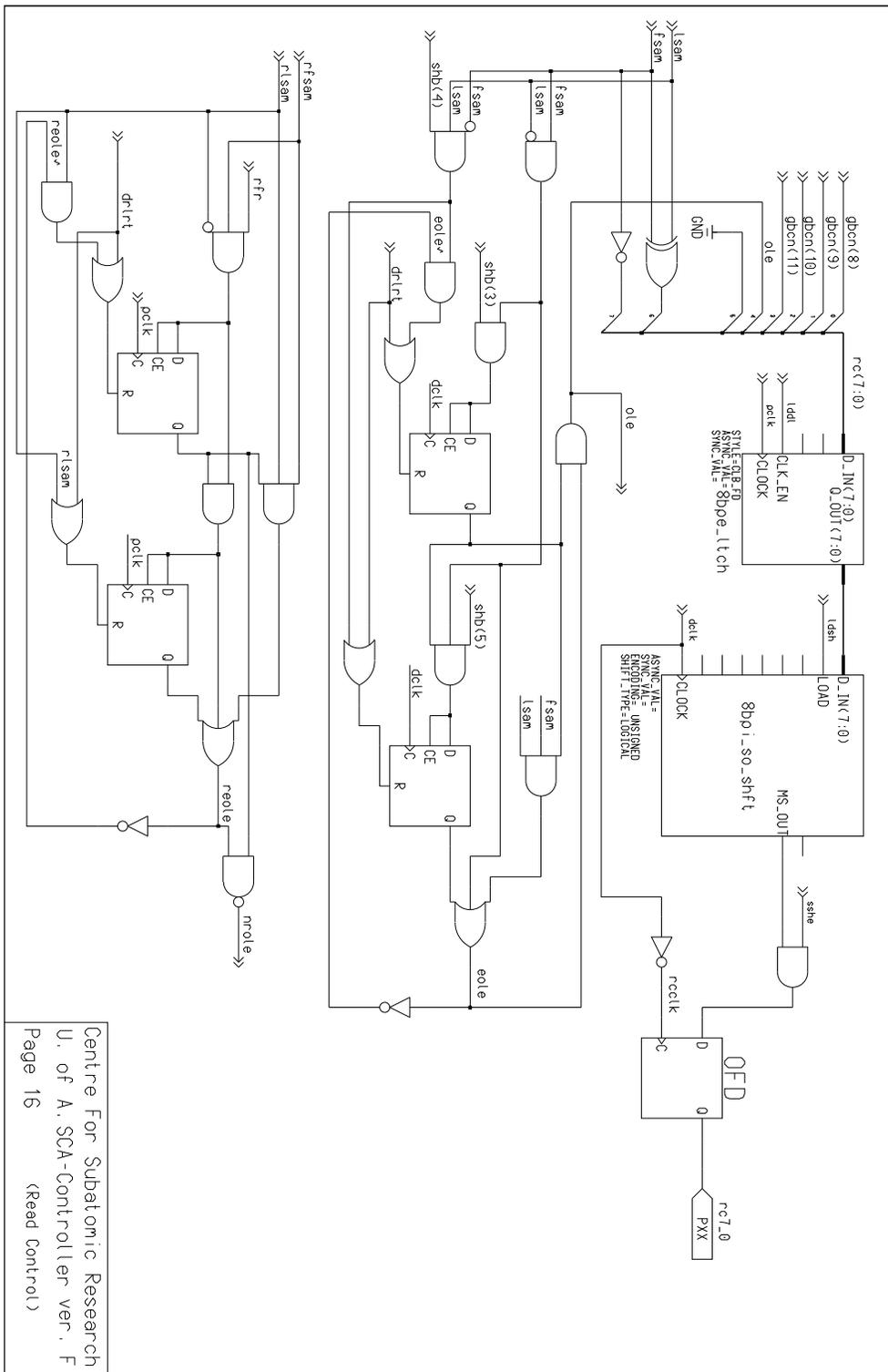


Figure 32: Read control: sheet 1 of 2.

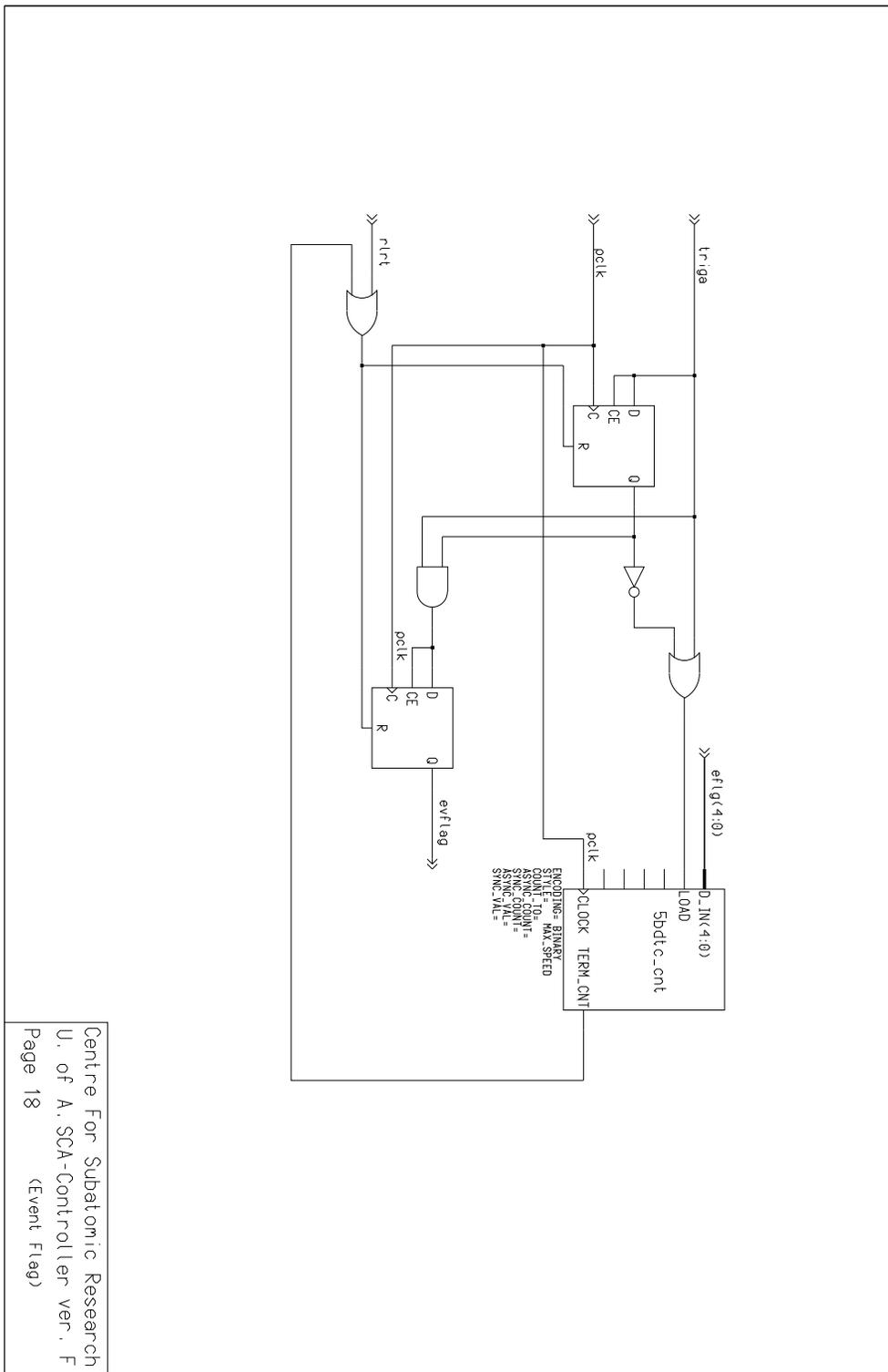


Figure 34: Event flag.

to coordinate it with the read-complete FIFO push signal (RCWE). The addresses coming out of the read-complete FIFO are latched into a pair of latches which operate on alternate transitions of PCLK. The data from these latches enters the address sequencer (figure 6) for insertion into the pool of available addresses. Signals MFFF and MFWR are communicated with the address sequencer to determine when addresses are available and when they can be output from the read-complete FIFO. The input latches before the FIFO and the FIFO push are controlled by the shift signals SH3, SH10, SH15 and RCWE in the read control logic in figure 30.

F.18 Error Detection

This circuit (figure 36) checks to see that the addresses are increasing properly and generates an error if they are not. The output address from the address sequencer is compared to the last address and checked to see if it is incrementing. Wrap around is handled. If the addresses are out of order, a latch is set and an 8-bit counter starts counting. If a second sequence error does not arrive before the counter has terminated, the latch is reset and no error condition is signaled on the output pin. If a second sequence error arrives before the counter has terminated, an error condition is signaled on the output pin. The output pin state will be held high until a circuit reset is issued (RLRT). An error signal can not be generated during a reload and reset (RLRT high or WFWEB low) or during the readout (ESIN high).

F.19 Inhibit Read and Write

This circuit (figure 37) is logic to inhibit the controller operations. Different inhibit signals can be issued.

If the read FIFO write signal RFW goes high signal ESIN will be latched high. The ESIN signal inhibits the sequence error signal (figure 36). If mode bit M1 is set, triggers will be inhibited with signal RTIN1. If mode bit M2 is set, triggers will be inhibited with signal RTIN2 and the write addresses will be inhibited with signal WAINHT. For these inhibits to be released the read FIFO write signal must be first removed. The end of read cycle signal CSYNC (which is synchronized to the 5 MHz clock) goes low and causes signal EBP to pulse. The pulse enables two counters. A 9-bit counter counts down until its terminal count is reached. This counter allows time for the addresses to be read out and put back into the write FIFO. At this point the counter is disabled and the inhibit signal RTIN2 (and ESIN) is released, if set. The second counter is enabled if at least one of mode bits M1 or M2 are set. An 8-bit counter counts down until its terminal count is reached. This counter allows time for the addresses to be read out. At this point the counter is disabled and the inhibit signals RTIN1 and WAINHT are released, if set.

F.20 Clocks Circuit

The clock circuit (figure 38) simply receives the 40 MHz and 5 MHz external clocks and makes them available to the controller as PCLK and DCLK respectively.

The remaining circuit in figure 38 is not meant to be used on a front-end board. The circuit is used for standalone running without the gain selector. It is used for testing the

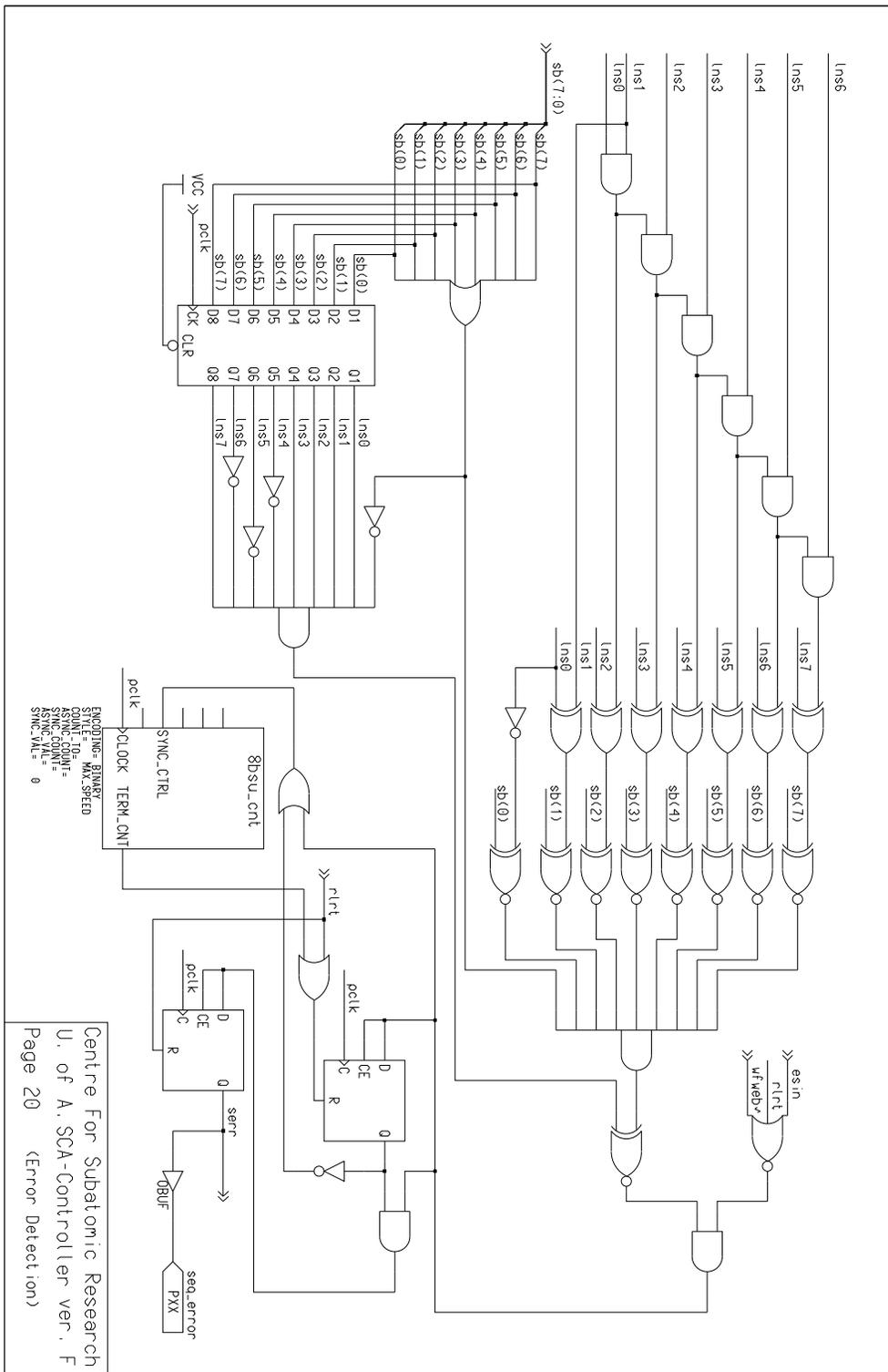


Figure 36: Error detection.

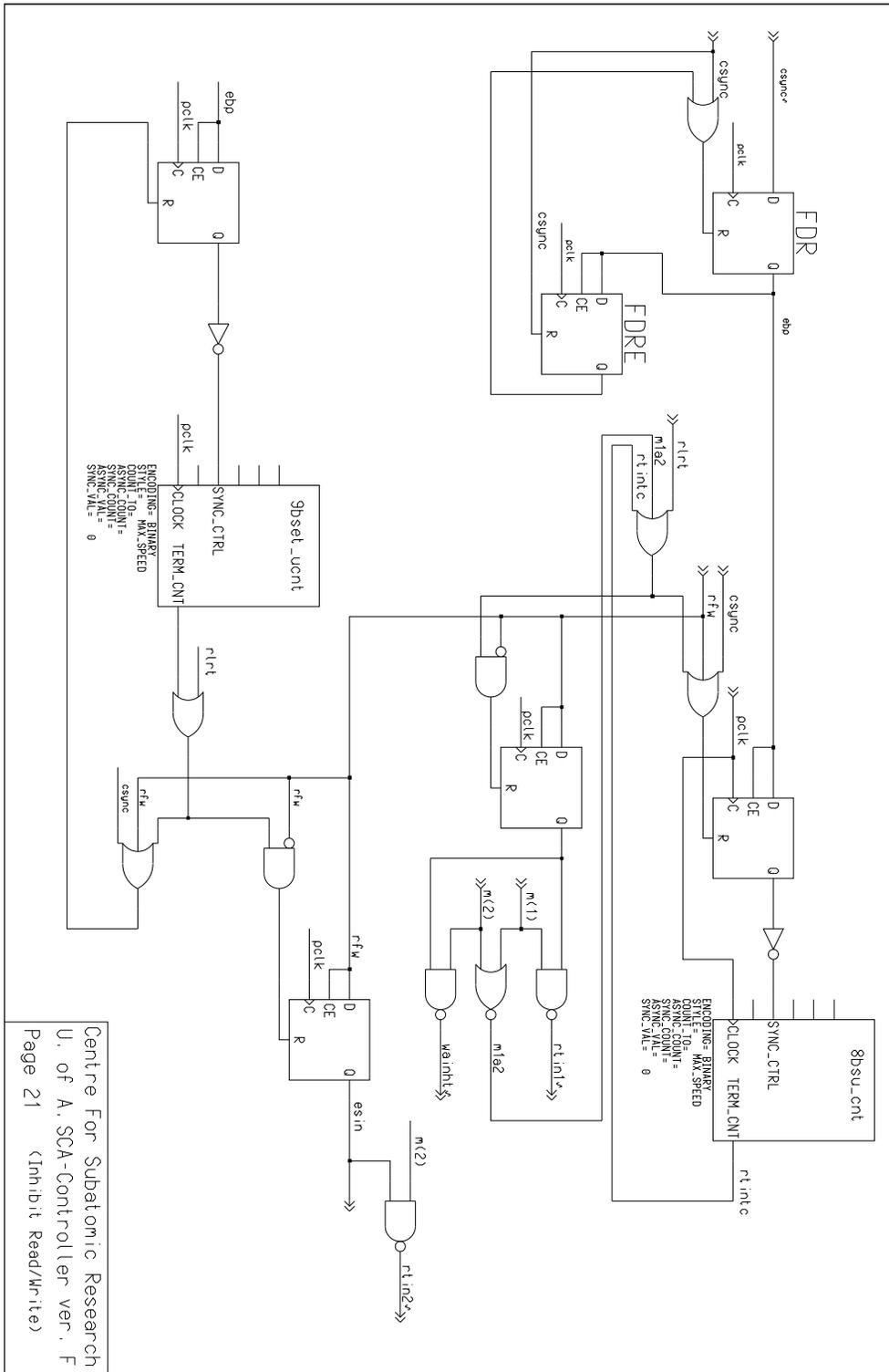


Figure 37: Inhibit read and write.

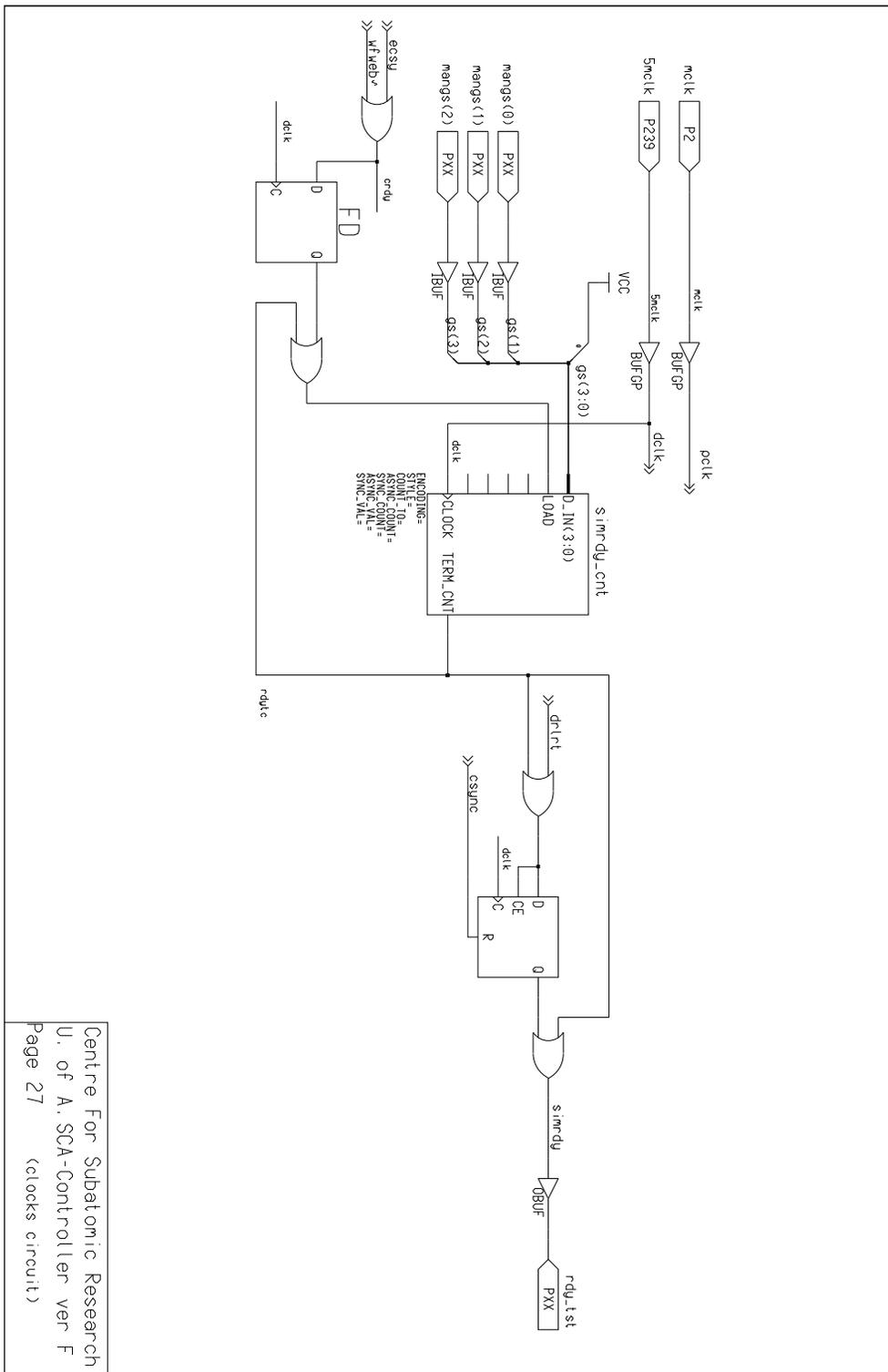


Figure 38: Clocks circuit.

controller on a test-board in Alberta.

G Important Signals

CSYNC data is read from the read RAM.

DCLK internal 5 MHz clock.

DRLRT delay reload and reset.

DLDC enables push of write FIFO while loading addresses.

EWDF enable write of read-delay FIFO (push).

FSAMP first sample signal.

LDC disable load of write FIFO counter.

LDD multiplex address from counter into write FIFO.

LSAMP last sample signal.

MFWR master FIFO write (read-complete FIFO pop).

PCLK internal 40 MHz (25 ns) primary clock.

RCWE read-complete FIFO write enable (push).

RFR read-FIFO read (pop).

RFW read-FIFO write (push).

RLRT reload and reset.

SB7:0 selected address from address sequencer.

SCLK strobe used for parameter loading and readback.

TRIGA trigger.

WAINHT inhibits write addresses to SCA if high.

WFRE write FIFO read enable (pop).

WFWE write FIFO write enable (push).

WFWEB write-FIFO write enable bit (pop of read-delay FIFO).

References

- [1] *SPAC: Serial Protocol for the ATLAS Calorimeter*, R. Bernier *et al.*, ATL-AL-LAL-ES-9.0.
- [2] *Design of the ATLAS LAr Front End Board*, ATL-AL-EN-0009.
- [3] *Total Ionizing Dose Effects in a Xilinx FPGA*, N.J. Buchanan *et al.*, ATL-LARG-99-003.
- [4] *An analog pipeline readout module for calorimetry at LHC*, D.M. Gingrich *et al.*, Proceedings of the International Conference on Electronics for Future Colliders, 11-12 May 1994, LeCroy Corp., Chestnut Ridge, New York, 189-201.
- [5] *A Pipeline Controller for the ATLAS Calorimeter*, D.M. Gingrich *et al.*, Proceedings of the First Workshop on Electronics for LHC Experiments, Lisbon, 1995, CERN/LHCC/95-56, 270-272.
- [6] *Design of the FPGA Controller for the SCA Test Chip*, D.M. Gingrich, 1997, <http://www.phys.ualberta.ca/~electronics/dual/index.html>
- [7] *Design of the SCA Controller for the Front-End Boards of the LAr Calorimeters*, D.M. Gingrich *et al.*, ATL-AL-EN-0008.

List of Figures

1	Block diagram of the controller for the front-end boards.	8
2	Write FIFO.	23
3	Read-delay FIFO.	24
4	Read address FIFO and read-complete FIFO.	25
5	Bunch-crossing number FIFO.	26
6	Address sequencer.	28
7	Grey encoder circuit.	31
8	Data register used to hold a parameter.	32
9	Command, data and readback registers.	33
10	Serial download.	34
11	Serial download device select.	37
12	Serial download registers 0 to 4.	39
13	Serial download register 5 to 9.	40
14	Serial readback and reset.	41
15	Trigger diagnostic.	42
16	Reset generation.	44
17	Startup circuit.	45
18	Input address for write FIFO.	47
19	SCA address deletion.	49
20	SCA write address.	50
21	Read delay and read FIFO.	52
22	Component U10.	53
23	Bunch crossing counter.	54
24	Read trigger.	55
25	Generate last sample.	57
26	Load read buffer RAM.	58
27	rb_capture.	60
28	Read address sequencer.	61
29	Event RAM.	62
30	Readout control.	64
31	External output signals.	65
32	Read control: sheet 1 of 2.	67
33	Read control: sheet 2 of 2.	68
34	Event flag.	69
35	Read-complete latch and FIFO.	71
36	Error detection.	72
37	Inhibit read and write.	73
38	Clocks circuit.	74

List of Tables

1	Inputs to the controller. The pin assignment is for the FPGA implementation.	10
2	SPAC commands for the controller.	11
3	Contents of the control register.	12
4	Description of programmable registers.	13
5	Outputs from the controller. The pin assignment is for the FPGA implementation.	14
6	Data sent from the controller to the gain selector logic.	15
7	Status bits.	16
8	Download and readback commands.	20
9	Size of the RAM for the FIFO memories in bits and Xilinx configurable logic blocks (CLB).	22
10	Size of the RAM in bits and Xilinx configurable logic blocks (CLB).	22