

**ATLAS NSW Electronics Specification  
Component or Facility Name: ROC**

**The Read Out Controller**

**Version: V 0.1**

***Abstract***

The Read Out Controller (ROC) merges hits from up to eight VMMs, reformats the data, adds headers and interfaces to one or more GBT E-links. The chip buffers Level-0 accepted events until a Level-1 Accept trigger further reduces the number of events to transmit.

A brief description of the component being specified and where it will be used.

All items in orange are template descriptions to be replaced in actual specifications. The term “component” will be widely used in these descriptions and that should be taken to be an individual component (e.g. an IC), a card or electronics module (e.g. a power supply), a facility (e.g. DCS) or a service (e.g. grounding and shielding).

If any section does not apply to what is being specified, please do not delete the section, but rather enter “Not Applicable” as the content of the section.

Specifications must be reviewed and approved at a Specification Verification Review (SVR) by a committee formed by the ATLAS review office in collaboration with the NSW project leader and electronics coordinator. Any later changes must be approved by the ATLAS review office who may call for a re-review depending upon the degree of the changes.

In addition to the SVR, three types of design reviews are referenced in this specification template. They are the Preliminary Design Review (PDR), Final Design Review (FDR) and Production Readiness Review (PRR), all of which will be organized by the ATLAS review office in collaboration with the NSW project leader and electronics coordinator. The PDR must be held prior to the fabrication of the first prototype. Further PDRs may be required prior to fabrication of revised prototypes depending upon the degree of changes. The decision for this will be made by the ATLAS review office in collaboration with the NSW project leader and electronics coordinator. The FDR must be held prior to the fabrication of the pre-production units and the PRR must be held prior to start of the production units.



<b><i>Revision History</i></b>				
<b><i>Rev. No.</i></b>	<b><i>Proposed Date</i></b>	<b><i>Description of Changes (Include section numbers or page numbers if appropriate)</i></b>	<b><i>Proposed By: author</i></b>	<b><i>Approved By: reviewer</i></b>
0.1	P: 09/15/2015 A: date	Initial draft	P: Stefan POPA P: Radu COLIBAN P: Traian TULBURE P: Dan NICULA P: Mihai IVANOVICI A: name	
		List only versions that are submitted for approval. If there is more than one author or reviewer, list each with a "P:" or "A:" as appropriate.		

## Table of Contents

<b><u>1 CONVENTIONS AND GLOSSARY.....</u></b>	<b>5</b>
<b><u>2 RELATED DOCUMENTS.....</u></b>	<b>5</b>
<b><u>3 DESCRIPTION OF COMPONENT OR FACILITY .....</u></b>	<b>5</b>
<b><u>4 INTERFACES .....</u></b>	<b>6</b>
<b><u>5 PHYSICAL DESCRIPTION.....</u></b>	<b>6</b>
<b><u>6 MANUFACTURER .....</u></b>	<b>7</b>
<b><u>7 POWER.....</u></b>	<b>7</b>
<b><u>8 COOLING .....</u></b>	<b>7</b>
<b><u>9 INPUT/OUTPUT .....</u></b>	<b>8</b>
<b><u>10 DETAILED FUNCTIONAL DESCRIPTION AND SPECIFICATION.....</u></b>	<b>8</b>
<b><u>10.1 VMM CAPTURE.....</u></b>	<b>8</b>
10.1.1 DES.....	9
10.1.2 COMMA ALIGN .....	10
10.1.3 VMM DEC.....	12
10.1.4 ASSEMBL .....	15
10.1.5 FIFO .....	16
<b><u>10.2 xBAR MODULE .....</u></b>	<b>16</b>
<b><u>10.3 SROC .....</u></b>	<b>20</b>
10.3.1 PACKET BUILDER .....	22
10.3.2 PACKET FIFO.....	26
10.3.3 TTC FIFO.....	26
10.3.4 STREAMER .....	26
10.3.5 SROC ENC.....	28
10.3.6 FEEDER.....	30
10.3.7 E-LINK TX .....	31
<b><u>11 RADIATION TOLERANCE, EMC ISSUES AND OTHER SPECIAL REQUIREMENTS .....</u></b>	<b>32</b>
<b><u>12 TESTING, VALIDATION AND COMMISSIONING .....</u></b>	<b>33</b>
<b><u>13 RELIABILITY MATTERS .....</u></b>	<b>33</b>
<b><u>13.1 CONSEQUENCES OF FAILURES.....</u></b>	<b>33</b>
<b><u>13.2 PRIOR KNOWLEDGE OF EXPECTED RELIABILITY .....</u></b>	<b>33</b>
<b><u>13.3 MEASURES PROPOSED TO INSURE RELIABILITY OF COMPONENT AND/OR SYSTEM .....</u></b>	<b>33</b>
<b><u>13.4 QUALITY ASSURANCE TO VALIDATE RELIABILITY OF DESIGN AND CONSTRUCTION OR MANUFACTURING TECHNIQUES .....</u></b>	<b>33</b>
<b><u>13.5 QUALITY CONTROL TO VALIDATE RELIABILITY SPECIFICATIONS DURING PRODUCTION .....</u></b>	<b>34</b>
<b><u>14 REFERENCES .....</u></b>	<b>34</b>

## 1 Conventions and Glossary

List and describe terms, especially acronyms, used in the specification and any conventions assumed and underlying the details of the specification.

## 2 Related Documents

All documents (other than references) linked to this specification. If this specification is for a component that integrates other NSW components or complex commercial components, include in this list all specifications for these components being integrated.

## 3 Description of Component or Facility

The Readout Controller (ROC) architecture is depicted in Figure 1.

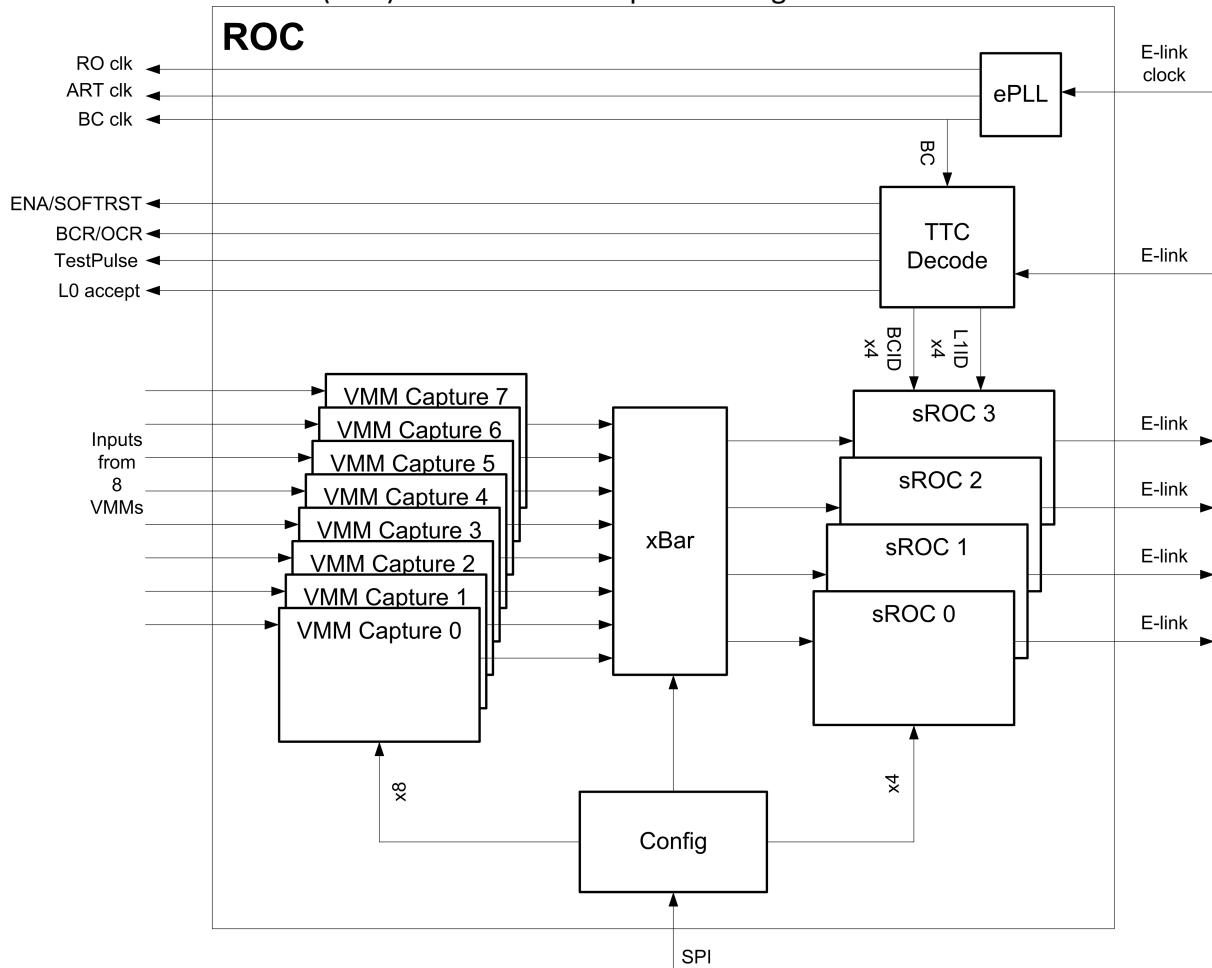


Figure 1: Block diagram of the Readout Controller.

The VMM Capture module receives L0 packets from a VMM on a serial interface and stores them in a FIFO in order to be read by the sROC modules. The xBar module interconnects 8 VMM Capture modules to 4 SROC in a configurable manner. The SROC aggregates the data from the corresponding VMMs and forms packets which are sent via an E-link serial interface. The configuration of the chip and error reporting is done via an SPI interface.

## 4 Interfaces

This component interfaces to other components listed in Table 4.1.

**Table 4.1: Components which Interface to This Component**

Name of Component	Name of Component Specification
List each electrical, mechanical or other component which interfaces to this component.	List the name of the specification for each component in the left column.

## 5 Physical Description

State the physical description of the component: unpackaged IC, packaged IC, PCB, rack mounted module, etc. If specifying a facility, state the area served by the facility, e.g. for the Grounding and Shielding Plan, state the part of the detector for which it applies.

- For an unpackaged IC, include a figure showing its size and pad arrangement and a table with pad assignments.
- For a packaged IC, include the type and size of package, a figure showing the pin arrangement and a table with pin assignments.
- For a PCB, include a figure showing its size and connectors and connection points and a table listing the connection assignments.
- For a rack mounted module or an enclosed unit, include a figure outlining the unit with its size and connections and a table listing the connection assignments.
- For a component that integrates other NSW components or complex commercial components, include a list of all these components being integrated.

The final size may not be known at the time of the first specification review but an estimate should be given. Actual size specified by the time of the PDR, FDR and PRR.



**Figure 5.1: Outline/Floor Plan of Component with Pads, Pins or Connectors Marked**

**Table 5.1: Pad/Pin or Connector Assignments**

Name or Number	Description
Name or number from Figure 5.1	Very brief description of connection.

## 6 Manufacturer

For a custom ASIC, state the foundry and the technology. For an off-the-shelf component, state the manufacturer. For a PCB, state the board fab and assembly vendors. For purchased systems, state the manufacturer. For a custom built system, state the institute responsible for building it.

For ASIC, state whether there are export limitations of this component into some countries as well as the need for including special features such as the PROMPT circuitry.

## 7 Power

List all the voltage and current sources supplied to the component. For each, list the maximum, nominal, and minimum allowed values. For each voltage source, state the expected nominal and maximum current draw. For each current source, state the allowed voltage range. Include any other requirements on the power, e.g. ripple. Some items like current draw may not be known at the time of the first specification review but a good estimate should be included by the time of PDR & FDR with final numbers for PRR.

**Table 7.1: Power Requirements**

Name	Max/Nom/Min V or I Supplied	Nom/Max I for Voltage Source	Max/Min V for Current Source	Other Requirements

## 8 Cooling

Specify the need for cooling if any as well as the maximum power which can be dumped in the environment and the maximum working temperature of the component.

## 9 Input/Output

Table 2: Input and outputs of ROC

Name	In, Out or I/O	Dim	Type of Signal or Max/Min	Description
Elnk_BCclk	in	2	SLVS	ROC clk, from GBT
Elnk_TTC	in	2	SLVS	TTC, from GBT
Elnk_data[0:3]	out	8	SLVS	sROC[0...3] L1 data
SPI_CS	in	1	CMOS	SPI Chip Select
SPI_Clk	in	1	CMOS	SPI Clock
SPI_di	in	1	CMOS	SPI data in
SPI_do	out	1	CMOS	SPI data out
reset	in	1	SLVS	from SCA DIO
TDS_BCclk[0:3]	out	8	SLVS	
TDS_BCR[0:3]	out	8	SLVS	
VMM_d0[0:7]	in	16	SLVS	
VMM_d1[0:7]	in	16	SLVS	
VMM_ROclk[0:7]	out	16	SLVS	
VMM_ARTclk[0:7]	out	16	SLVS	
VMM_BCclk[0:7]	out	16	SLVS	
VMM_BCR[0:7]	out	16	SLVS	
VMM_LOA[0:7]	out	16	SLVS	
VMM_TP[0:7]	out	16	SLVS	
VMM_ENA[0:7]	out	16	SLVS	
TEST_MODE	in	1	CMOS	
SEU	out	1	CMOS	
SCAN_IN	in	1	CMOS	
SCAN_OUT	in	1	CMOS	
SCAN_EN	in	1	CMOS	

## 10 Detailed Functional Description and Specification

### 10.1 VMM Capture

VMM Capture architecture is presented in Figure 3, along with the data path. The serial input lines are connected to the deserializer, which outputs 10-bit words, which are synchronized using comma words by the comma align circuit and decoded by the 8b/10b decoder. The resulting 8-bit words are assembled into 33-bit words, which are written in the FIFO.

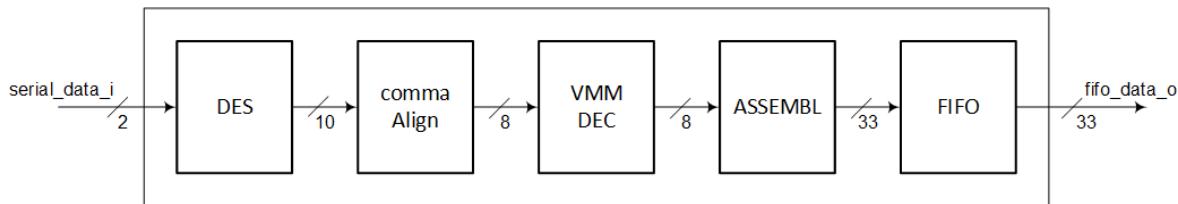


Figure 3: Block diagram and data path of the VMM Capture module.

Table 3: VMM Capture interface

Port	Direction	Dim	Description
clk_40_i	in	1	40 MHz clock input
clk_160_i	in	1	160 MHz clock input
reset_n_i	in	1	Asynchronous reset, active low
serial_data_i	in	2	Serial DDR data input from VMM
fifo_data_o	out	33	Data out from FIFO
fifo_empty_o	out	1	FIFO empty flag
fifo_rd_i	in	1	FIFO read increment signal, connected to xBar
status_o	out	6	Status signals: [5] Error - trying to write in full FIFO [4] Start Bit Error – in received header [3] Coherency error – received packet not complete [2] 8b/10b decoding error [1] Misaligned error – VMM data alignment changed [0] Module is synchronized to VMM via comma alignment

The data inputs are DDR on a clock of 160 MHz, with the odd bits on one line and the even bits on the other giving, the (8b/10b encoded) data being expected to be received LSB-first. For example, for a 10-bit word *abcdefghijkl* (*a* is the LSB), the input format is as follows:

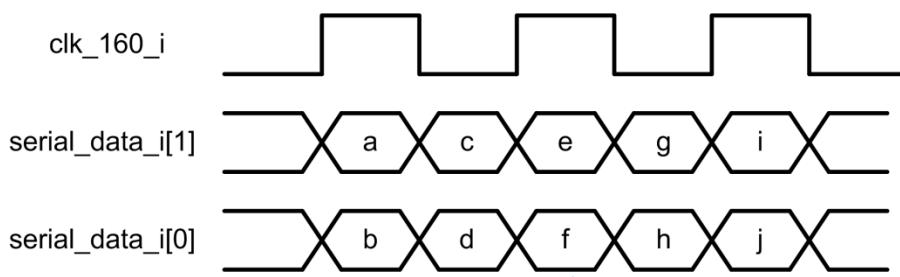


Figure 4. VMM Capture input format.

### 10.1.1 DES

The DES module (Figure 5) captures the serial inputs connected to the VMM and outputs 10-bit data words, encoded using 8b/10b. The interface is presented in Table 4. The output is validated by *data\_valid\_o*. The internal structure of the DES module is presented in Figure 6.

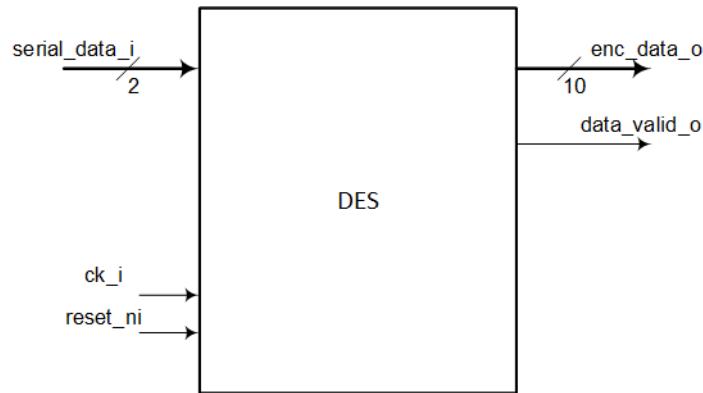


Figure 5: Block diagram of the DES module.

Table 4: DES interface.

Port	Direction	Dim	Description
clk_160_i	in	1	160 MHz clock input
reset_n_i	in	1	Asynchronous reset, active low
serial_data_i	in	2	Serial DDR data input from VMM
enc_data_o	out	10	Output data, which is 8b/10b encoded
data_valid_o	out	1	Signal that indicates that output data is valid

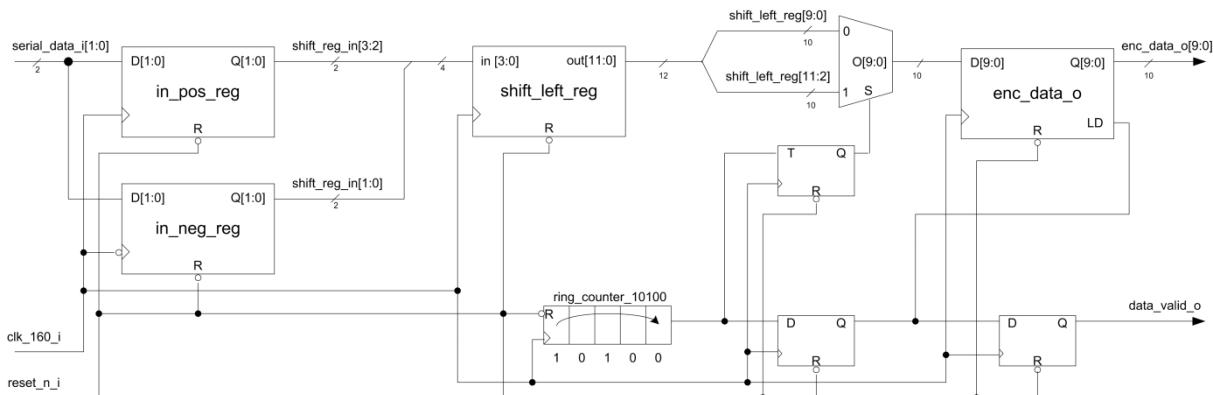


Figure 6: Internal structure of the DES module.

### 10.1.2 COMMA ALIGN

The **commaAlign** module monitors the 10-bit input for the presence of the comma character K28.5. Once this character is identified in the input stream, the 10-bit data is aligned to the proper boundaries, thus achieving synchronization with the VMM.

The input data *enc\_data\_i* does not come continuously (just when *data\_valid\_i*=1).

The output data *enc\_data\_o* is not issued continuously (activated *data\_valid\_o*=1).

The output *aligned\_o* is set when the following events are met sequentially:

- a K28.5 word is determined at any position, inside 2 consecutive 10 bit data
- additional *nok\_p-1* words K28.5 are determined aligned with the 1<sup>st</sup> one

The output *aligned\_o* is reset when during the last *allignTimer\_p* valid input data no K28.5 word was determined.

K28.5 encoding is: 001111 1010 or 110000 0101.

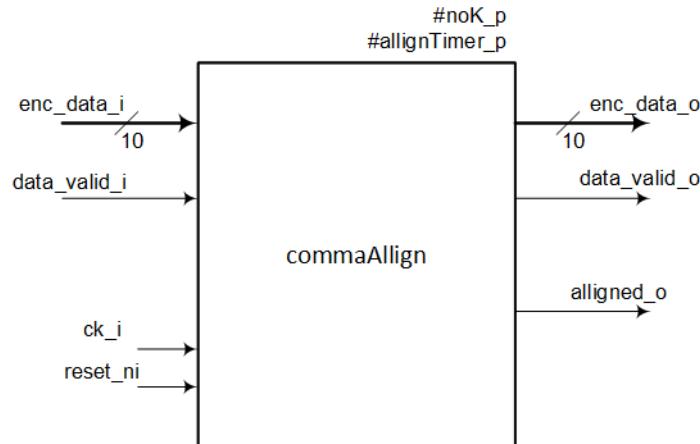


Figure 7: Block diagram of the comma align module.

Table 5: The Comma Align interface

Port	Direction	Dim	Description
ck_i	in	1	160 MHz clock input
reset_ni	in	1	Asynchronous reset, active low
enc_data_i	in	10	Input data that is 8b/10b encoded
data_valid_i	in	1	Signal that indicates that input data is valid
enc_data_o	out	10	Aligned output data that is 8b/10b encoded
data_valid_o	out	33	Signal that indicates that output data is valid
aligned_o	out	1	Signal that indicates that the module is aligned with the data stream

Two consecutive *enc\_data\_i[9:0]* are sampled on *data\_valid\_i* and then propagated through a 5 stages pipeline.

The 1<sup>st</sup> pipe stage store the second sample of *enc\_data\_i[9:0]*.

The 2<sup>nd</sup> pipe stage store first (earliest) sample of *enc\_data\_i[9:0]*.

The 3<sup>rd</sup> pipe stage merges 20 bits from the 1<sup>st</sup> and 2<sup>nd</sup> sample. Each K28 bits (001111 1010 or 110000 0101) are compared with 10 bits from the 20 bits of the two samples (10 phases) as follows:

$$\text{isAligned}[0] = (\text{sample}[9:0] == \text{K28.5a}) \text{ or } (\text{sample}[9:0] == \text{K28.5b})$$

$$\text{isAligned}[1] = (\text{sample}[10:1] == \text{K28.5a}) \text{ or } (\text{sample}[10:1] == \text{K28.5b})$$

...

$$\text{isAligned}[9] = (\text{sample}[18:9] == \text{K28.5a}) \text{ or } (\text{sample}[18:9] == \text{K28.5b})$$

The 4<sup>th</sup> pipe stage propagates the 20 bits date and generates the phase shift (0..9, encoded on 4 bits). If K28.5 was not found in any position (*isAligned[9:0]==10'b0*) then *foundK=0*. If K28.5 was found in a position then *foundK=1*.

The 5<sup>th</sup> pipe stage implements a barrel shifter to aligned the data. Based *phshift* from the 20 bits the 10 bit aligned data is selected.

*countK* counts the number of consecutive valid data that contains K28.5 symbols. If this counter reaches the *nok\_p*, the *aligned\_o* output is set.

The timer is a down counter, enabled when *data\_valid\_i*, loaded with the parameter *alignTimer\_p* at every *foundK*=1. As a consequence, the timer will reaches 0 only if there were not any K28.5 symbol found during the last *alignTimer\_p* valid symbols. When the times reaches 0, the *aligned\_o* output is reset, and *countK* is cleared.

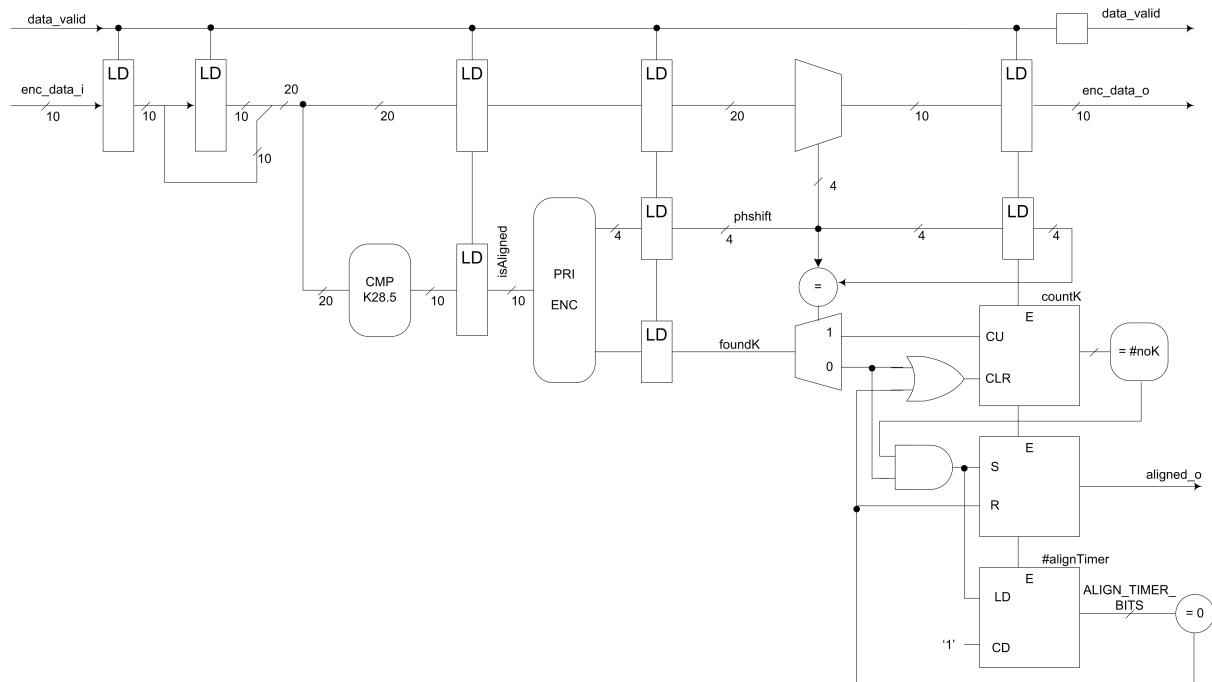


Figure 8: Internal structure of the *commaAlign* module.

### 10.1.3 VMM DEC

The VMM DEC, which includes an 8b/10b decoder, is presented in Figure 9. The 10-bit input is decoded, with the result present on the 8-bit output. The output is validated by *data\_valid\_o*. The input is also validated by *data\_valid\_i*. The *comma* signal is high when the K28.5 comma character was decoded. The *err\_dec* output signals if an illegal character, a running disparity error or decoding error was encountered. The documentation of the 8b/10b encoding can be found at the following link: [http://en.wikipedia.org/wiki/8b/10b\\_encoding](http://en.wikipedia.org/wiki/8b/10b_encoding)

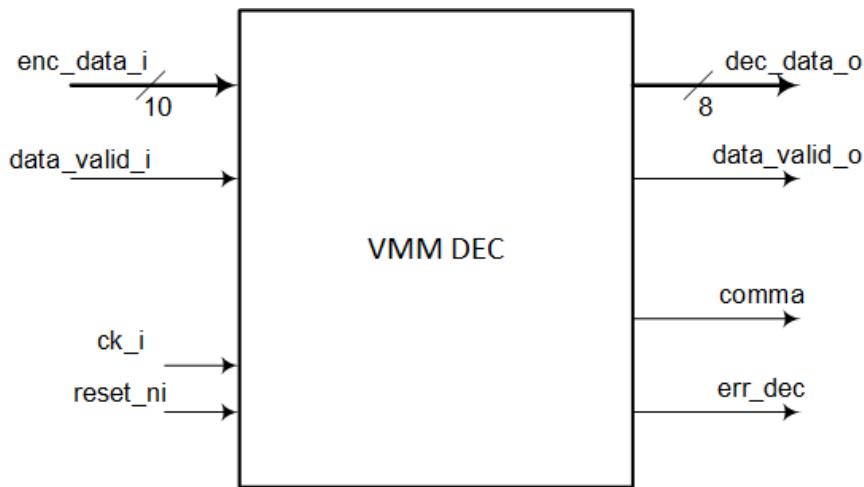


Figure 9: Block diagram of VMM DEC module.

The interface of the VMM DEC module is presented in the following table:

Table 6: VMM DEC interface

Port	Direction	Dim	Description
ck_i	in	1	160 MHz clock input
reset_ni	in	1	Asynchronous reset, active low
enc_data_i	in	10	Input 8b10b encoded character to be decoded (little endian)
data_valid_i	in	1	Signal that indicated that input data is valid
dec_data_o	out	8	The decoded character
data_valid_o	out	1	Signal that indicates that output data is valid
comma	out	1	Signal that indicates that a K28.5 character was decoded.
err_dec	out	1	Signal that indicates that an error was encountered

The decoder module transforms 10 bit symbols into 8 bit data words, checking for errors. This module is required for decoding the data sent to the Read Out Controller by the VMM. The free SGMII-IP-Core contains an 8b/10b encoder and decoder written in Verilog. These can be found at this link: <http://fpga-ipcores.com/8b10b-encoder-decoder/>. Both this modules were tested by reference to simulation models (encoder and decoder) made by Altera. This decoder has the following block diagram:

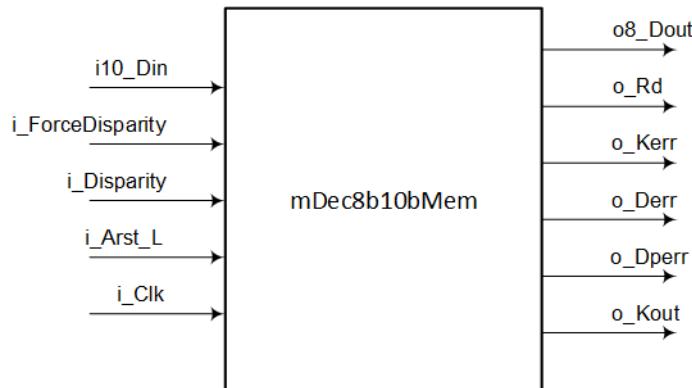


Figure 10: Block diagram of the SGMII-IP-Core 8b/10b decoder

The interface of this decoder module is described in the following table:

Table 7: SGMII-IP-Core 8b/10b decoder interface

Name	Size	Required	Description
i10_Din	10	Y	10-bit data input to be decoded (abcdeifghj)
o_Kout	1	Y	Control character signal
i_Disparity	1	Y	Disparity input, required when i_ForceDisparity is asserted. 1: Even Disparity 0: Odd Disparity
i_ForceDisparity	1	Y	Force disparity to override the internally computed disparity
o8_Dout	8	Y	8-bit data output (HGFEDCBA) Notice that the output is bit reversed, A is the "lsb" Output is available 1 clock after the input
o_Rd	1	N	Running disparity The running disparity will be used to decode the current 10-bit input 1: Even Disparity 0: Odd Disparity
o_Kerr	1	N	Error when decoding K character
o_Derr	1	Y	Error when the 10 bit symbol does not have a corresponding 8 bit data word
o_Dperr	1	Y	Running disparity error
i_Clk	1	Y	Clock input
i_ARst_L	1	Y	Active low reset After reset, the running disparity is initialized to "negative"

The internal structure of the VMM DEC module is depicted in the following figure:

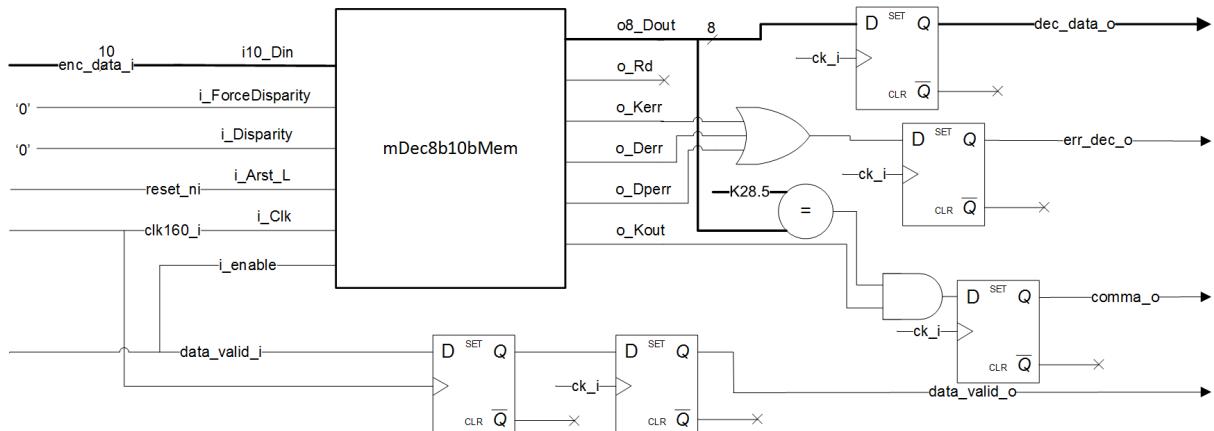


Figure 11: VMM DEC internal structure

### 10.1.4 ASSEMBL

The module, depicted in Figure 12, assembles the 8-bit words received from the decoder into 32-bit words which are written in the FIFO. An additional output bit is present in the data bus, used to indicate the end-of-packet. The module also outputs an error signal, asserted when the module tries to write the data in a full FIFO. The internal structure of the Assembler module is presented in Figure 13.

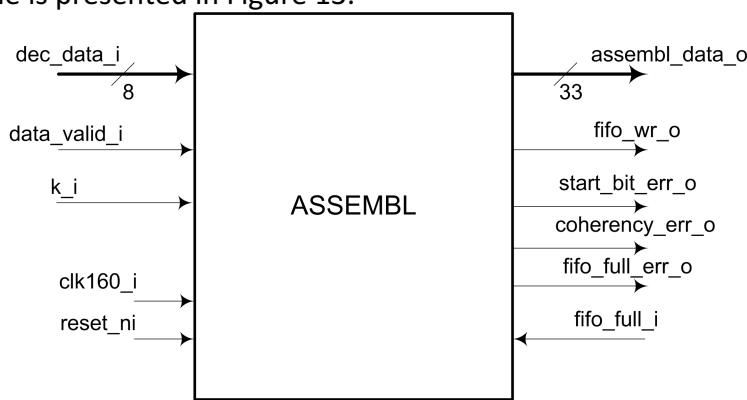


Figure 12: Block diagram of Assembler module.

Table 8: The Assembl interface

Port	Direction	Dim	Description
clk160_i	in	1	160 MHz clock input
reset_ni	in	1	Asynchronous reset, active low
dec_data_i	in	8	The decoded character
data_valid_i	in	1	Signal that indicated that input data is valid
k_i	in	1	Indicates that a K28.5 character was decoded
assembl_data_o	out	33	The assembled 32 + 1-bits words which are written into VMM Capture FIFO
fifo_wr_o	out	1	Signal that indicates that output data is valid and it is to be written into VMM Capture FIFO
fifo_full_i	in	1	Signal that indicates that the FIFO is full.
coherency_err_o	in	1	Signal that indicates that the current received packet is not complete
fifo_full_err_o	out	1	Signal that indicated that a word could not be

			written into the FIFO because the FIFO is full.
start_bit_err_o	out	1	Signal that indicates that a start bit error

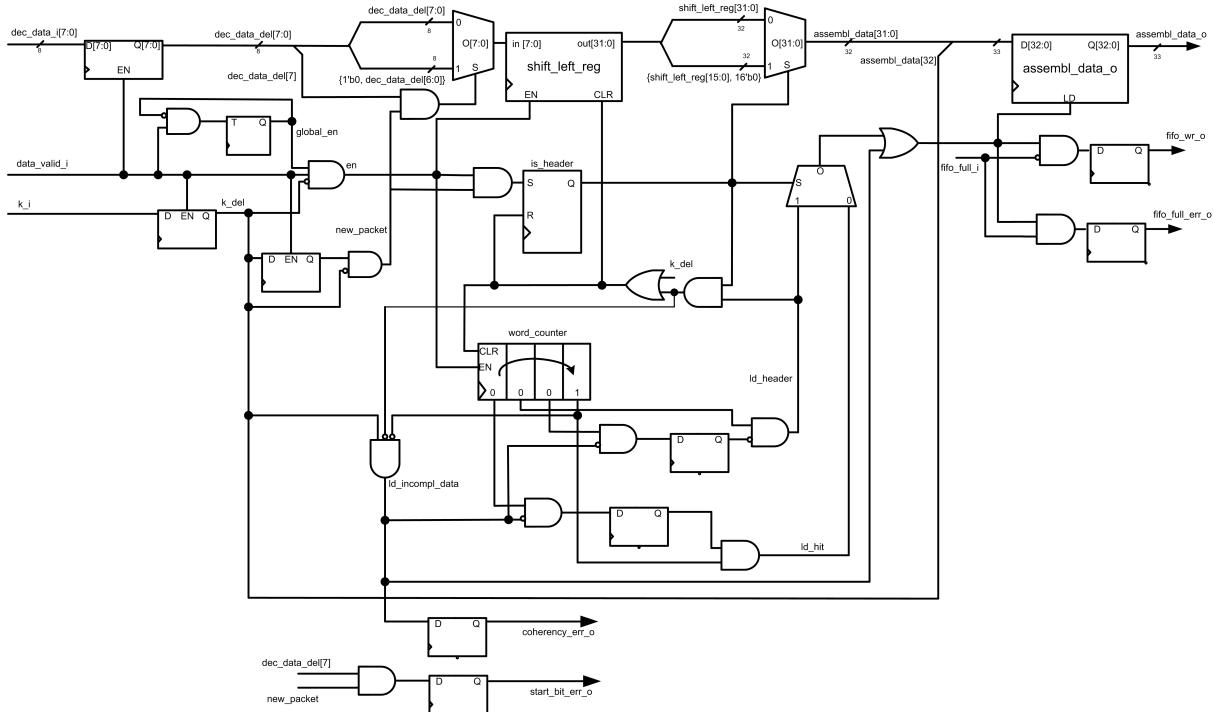


Figure 13: Internal structure of Assembler module.

### 10.1.5 FIFO

The FIFO module stores the 33-bit words and implements the clock domain crossing, between 160 MHz and 40 MHz. The format of the different types of data words stored in the FIFO is presented in Figure 14.

header hit data message	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOP	0	P	orb	BCID (12)												16'b0																	
EOP	1	P	0	T	Chan# (6)						ADC (10)										TDC (8)								N	rel BCID			
1	1	P	1	R	R	msg type	msg ID		msg TBD (20)																								

Figure 14: Format of the data stored in the VMM Capture FIFO

### 10.2 xBar Module

The **xBar** module connects into a fully mesh network 8 x VMM capture modules to 4 x SROC modules.

The data (33 bits) issued by VMM capture modules are directed to SROC together with *empty* and *almostEmpty* control signals.

The *read* signals from SROC are directed to VMM Capture modules.

The selection is based on configuration inputs as follow:

- Each VMM capture (from 8) receives 4 selection bits, corresponding to the destination SROC Total  $4 \times 8 = 32$  configuration bits.
- Each SROC (from 4) receives 3 selection bits, corresponding to the destination VMM Capture index. Total  $3 \times 4 = 14$  configuration bits.

**Table 9: The xBar interface**

<b>Port</b>	<b>Direction</b>	<b>Dim</b>	<b>Description</b>
data_i	in	8x33	Data input from 8 x VMM capture
data_o	out	4x33	Data output to 4 x SROC
fifo_empty_i	in	8	Empty signal from VMM Capture
fifo_almost_empty_i	in	8	almostEmpty signal from VMM Capture
fifo_empty_o	out	4	Empty signal to SROC
fifo_almost_empty_o	out	4	almostEmpty signal to SROC
fifo_read_i	in	4	Read command from SROC
fifo_read_o	out	8	Read command to VMM capture
vmm_ena_i	in	4x8	Configuration bits for 8 x VMM Capture
vmm_sel_i	in	3x4	Configuration bits for 4 x SROC

vmm\_ena\_i comes from the SROC, it is decoded based on Connectivity options for VMM to SROC connection. This input is static.

vmm\_ena\_i = VMM7, VMM6, VMM5, VMM4, VMM3, VMM2, VMM1, VMM0

VMMx = 4'b{SROC index}

vmm\_sel\_i comes from the 4 x SROC modules based on Connectivity options for VMM to SROC connection and on current selected VMM. This input is dynamic based on SROC list.

SROC name	SROC index
1	0001
2	0010
3	0100
4	1000

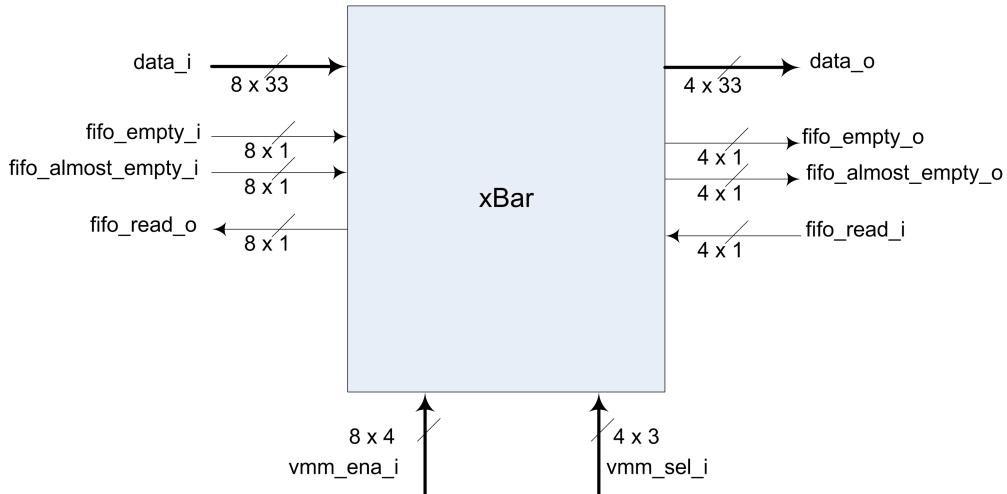


Figure 15: Block diagram of the xBar module

The **xBar** module is fully combinatorial and consist on 4 x MUXes on SROC side (for data, fifoEmpty and fifoAlmostEmpty), and 8 x MUXes on VMM capture side and 4 x DMUXes on SROC side (for fifoRead).

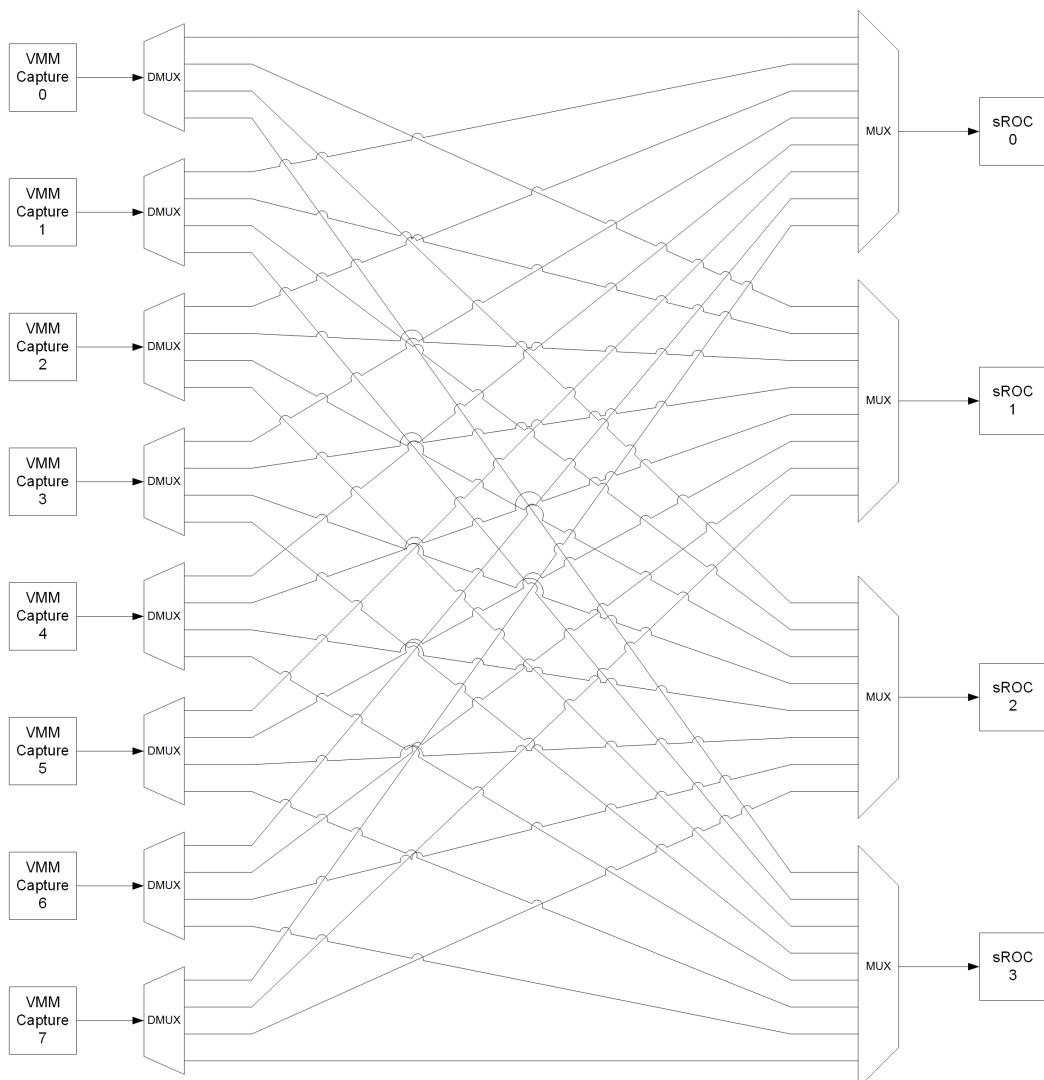
According to table 1 from *Requirements for the NSW VMM3 readout ASIC and the NSW Readout Controller ASIC Design Review Report*, The configuration bits should be set as presented in table XXX. The configuration bit decoding to generate vmmEna\_i and vmmSel\_i inputs to the xBar is performed inside SROC modules.

Table 1: Connectivity options for VMM to SROC connection

SROC1	SROC2	SROC3	SROC4	possible use
"2222"	VMM 0, 7	VMM 1, 6	VMM 2, 5	VMM 3, 4 sTGC inner quad
"422"	VMM 0, 7, 1, 6		VMM 2, 5	VMM 3, 4 sTGC middle quad
"44"	VMM 0, 7, 1, 6		VMM 2, 5, 3, 4	sTGC outer quad
"224"	VMM 0, 7	VMM 1, 6	VMM 2, 5, 3, 4	sTGC middle quad
"8"	VMM 0..7			MM and sTGC outer

**Table 10: Configuration bits set for requested connectivity options for VMM to sROC**

<b>Configuration</b>	<b>vmm_ena_i[15:0]</b>	<b>vmm_sel_i[11:0]</b>
2222	0001_0010_0100_1000_1000_0100_0010_0001	011/100_010/101_001/110_000/111
422	0001_0001_0100_1000_1000_0100_0001_0001	011/100_010/101_xxx_000/111/001/110
44	0001_0001_0100_0100_0100_0100_0001_0001	xxx_010/101/011/100_xxx_000/111/001/110
224	0001_0010_0100_0100_0100_0100_0010_0001	xxx_010/101/011/100_001/110_000/111
8	0001_0001_0001_0001_0001_0001_0001_0001	xxx_xxx_xxx_000/001/010/011/100/101/110/111

**Figure 16: xBar structure.**

### 10.3 sROC

The sROC module reads data from up to 8 VMM capture modules, extracts the hit data which corresponds to the BCID received from the TTC capture module and forms packets which are send over E-link. The block diagram is depicted in Figure 17 and the list of interfaces is presented in Table 11.

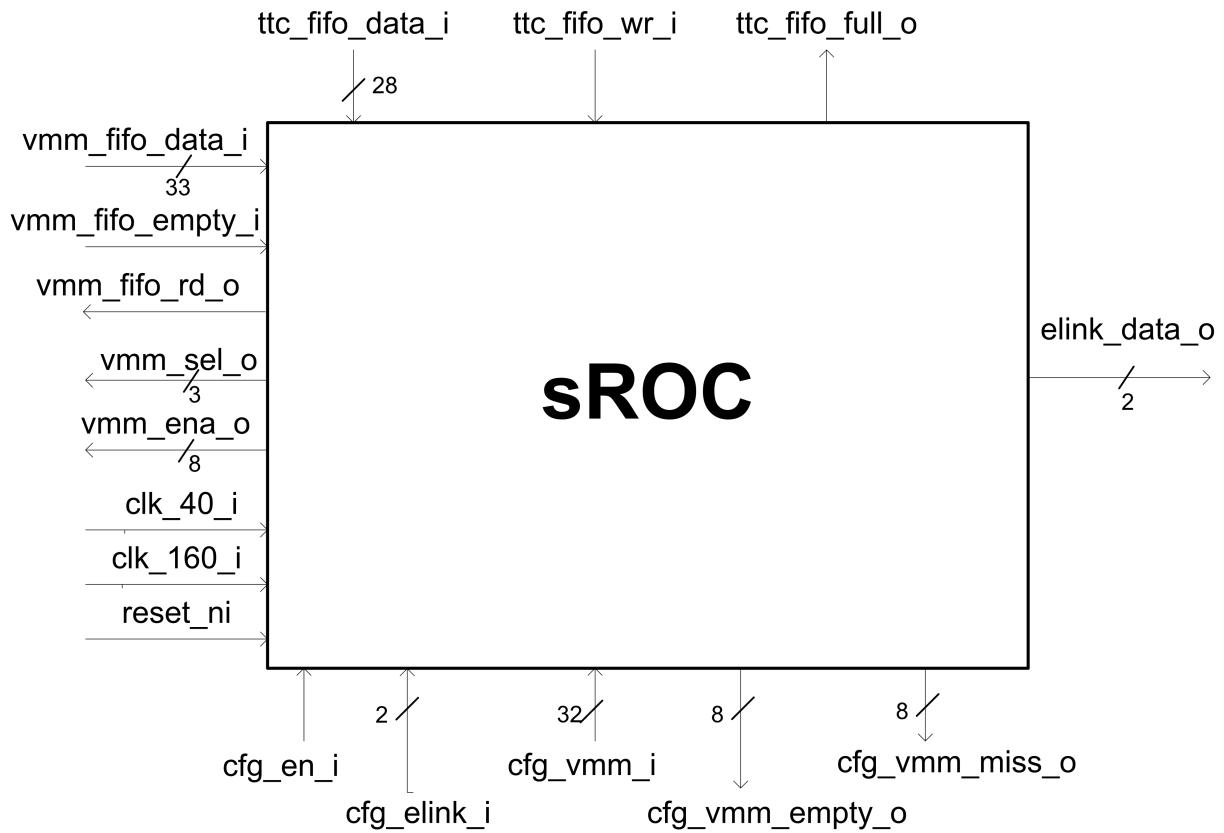


Figure 17: Block diagram of the sROC module.

Table 11: sROC interface.

Port	Dir	Dim	Description
Clock and reset interface:			
clk_40_i	in	1	40 MHz clock
clk_160_i	in	1	160 MHz clock for E-Link
reset_ni	in	1	Asynchronous reset, active low
VMM capture fifo interface:			
vmm_fifo_data_i	in	33	Data from xBar
vmm_fifo_empty_i	in	1	FIFO empty flag from xBar
vmm_fifo_rd_o	out	1	FIFO read signal to xBar
vmm_sel_o	out	3	Select active VMM based on configuration
vmm_ena_o	out	8	Enable access to VMM based on configuration
TTC interface:			
ttc_fifo_data_i	in	28	Data from TTC (BCID - 12 bits and L1ID - 16 bits)

<code>ttc_fifo_wr_i</code>	in	1	Data write signal from TTC
<code>ttc_fifo_full_o</code>	out	1	FIFO full signal to TTC
<b>Configuration interface (SPI):</b>			
<code>cfg_en_i</code>	in	1	Global enable signal for the module
<code>cfg_elink_i</code>	in	2	Configures data frequency for E-Link
<code>cfg_vmm_i</code>	in	32	Specifies which VMM capture modules are connected and in what order they should be read
<code>cfg_vmm_empty_o</code>	out	8	Reports if VMM capture FIFO is empty (set bit)
<code>cfg_vmm_miss_o</code>	out	8	Reports if VMM event miss happened (set bit)
<code>cfg_event_full_o</code>	out	1	Reports that event FIFO is full
<b>Elink serial interface &amp; SEU report bit:</b>			
<code>elink_data_o</code>	out	2	E-link serial data out
<code>seu_err_o</code>	out	1	Error signal, indicates that a Single Event Upset has been detected

After reset, the module waits for a BCID from the TTC capture module. When this TTC BCID is received, the module reads all the connected VMM Capture FIFOs in a round robin fashion, discarding the packets with a smaller BCID than the one received from the TTC. A packet is formed with all the hit data that has the same BCID as the TTC BCID. If no hit data is available for the considered TTC BCID, a null event header is sent. Also, messages from the VMM capture modules are forwarded. After all the VMM capture modules are checked (meaning that there is no header in the FIFOs with a BCID less than or equal to the TTC BCID), the module will read the next BCID received from the TTC module and repeat the process.

Each SROC is configured via I2C with following bits:

- `cfg_en_i` - enable SROC block
- `cfg_elink_i` – select elink speed
- `cfg_vmm_i` - a list of VMM capture blocks connected to VMM. The 32 bits register contains 8 fields of 4 bits each that configure the VMM capture list. Each VMM capture has an index form 0-7 and fields  $\geq 8$  are treated as end of list. Default reset value is 32'h88888888 (empty list). Example for 422 configuration the SROC1 will be programmed to 32'h88886170.

The structure of the sROC module and the main internal signals are presented in Figure 18.

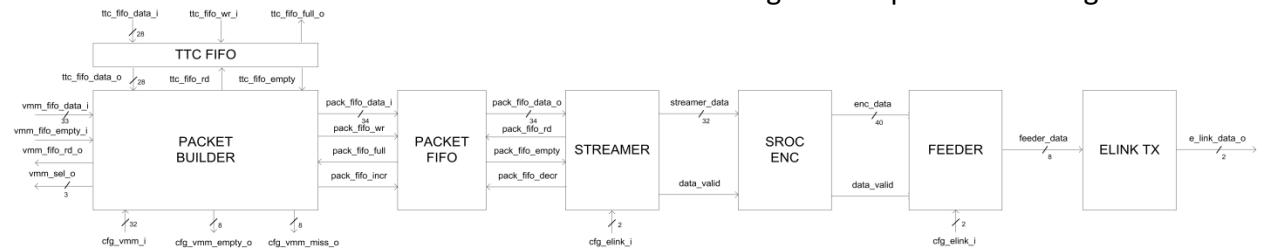


Figure 18: Structure of the sROC module.

### 10.3.1 Packet Builder

The Packet Builder reads data from the VMM Capture modules and formats the data packets which are stored in the Packet FIFO. The structure of the module is depicted in Figure 20; the diagram of the state machine is presented in Figure 19 while a more detailed representation is depicted in Figure 22. The format of the data words at the input of out\_mux is presented in Figure 23.

**Table 12: The Packet Builder interface**

Port	Dir	Dim	Description
<b>Clock and reset interface:</b>			
clk_40_i	in	1	40 MHz clock
reset_ni	in	1	Asynchronous reset, active low
<b>VMM capture fifo interface:</b>			
vmm_fifo_data_i	in	33	Data from xBar
vmm_fifo_empty_i	in	1	FIFO empty flag from xBar
vmm_fifo_rd_o	out	1	FIFO read signal to xBar
vmm_sel_o	out	3	Select active VMM based on configuration
vmm_ena_o	out	8	Enable access to VMM based on configuration
<b>TTC FIFO interface:</b>			
ttc_fifo_data_i	in	28	Data from TTC FIFO(BCID - 12 bits and L1ID - 16 bits)
ttc_fifo_rd_o	out	1	Data read signal to TTC FIFO
ttc_fifo_empty_i	in	1	FIFO empty signal from TTC FIFO
<b>Configuration interface (SPI):</b>			
cfg_en_i	in	1	Global enable signal for the module
cfg_vmm_i	in	32	Specifies which VMM capture modules are connected and in what order they should be read
cfg_vmm_empty_o	out	8	Reports if VMM capture FIFO is empty (set bit)
cfg_vmm_miss_o	out	8	Reports if VMM event miss happened (set bit)
cfg_event_full_o	out	1	Reports that event FIFO is full
<b>PACK FIFO interface:</b>			
pack_fifo_data_o	out	34	Output data to be written into PACK FIFO
pack_fifo_wr_o	out	1	Write signal to PACK FIFO
pack_fifo_full_i	in	1	FIFO full signal from PACK FIFO
pack_fifo_incr_o	out	1	Increment packet counter from PACK FIFO

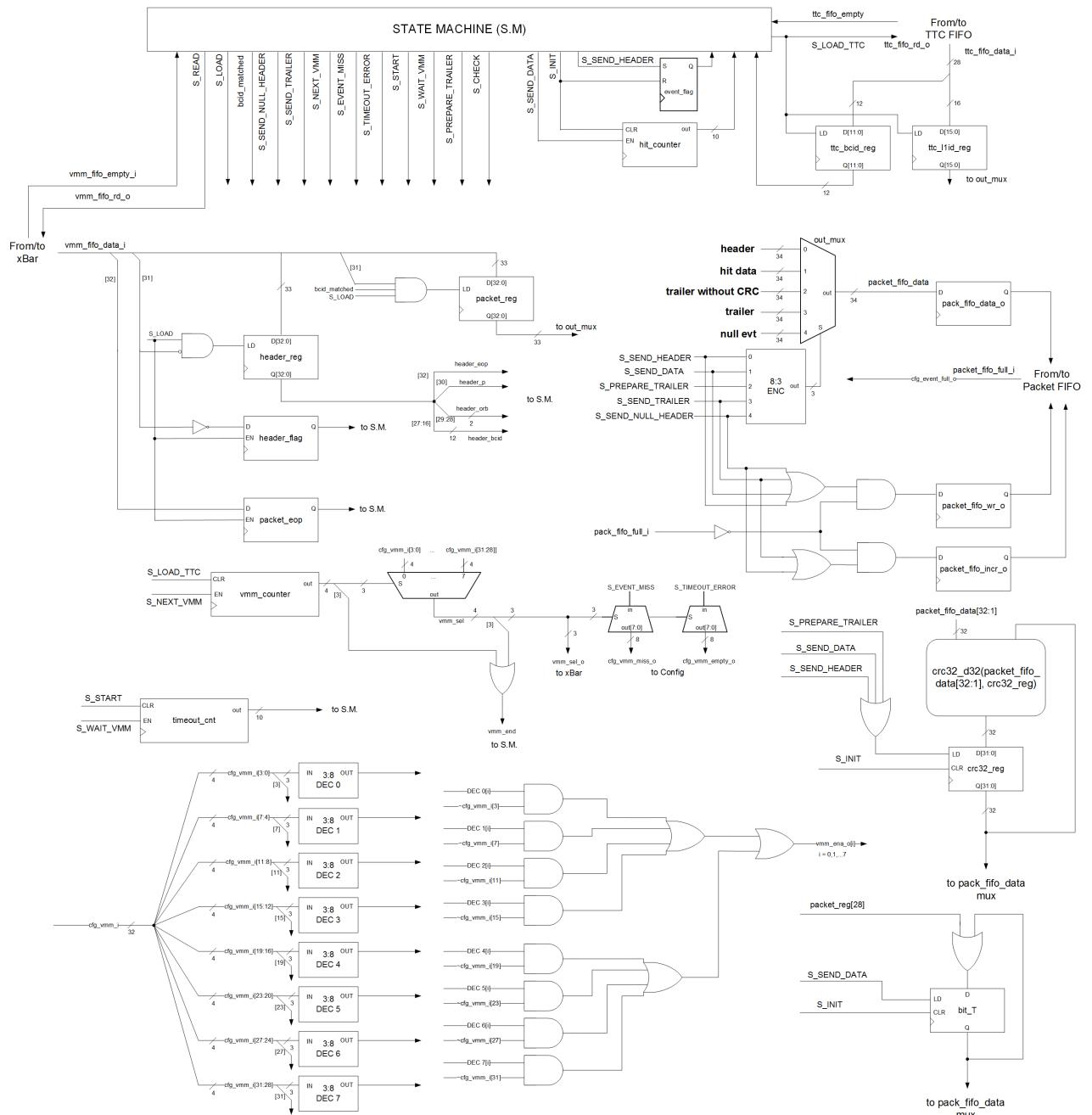


Figure 20: Packet Builder structure.

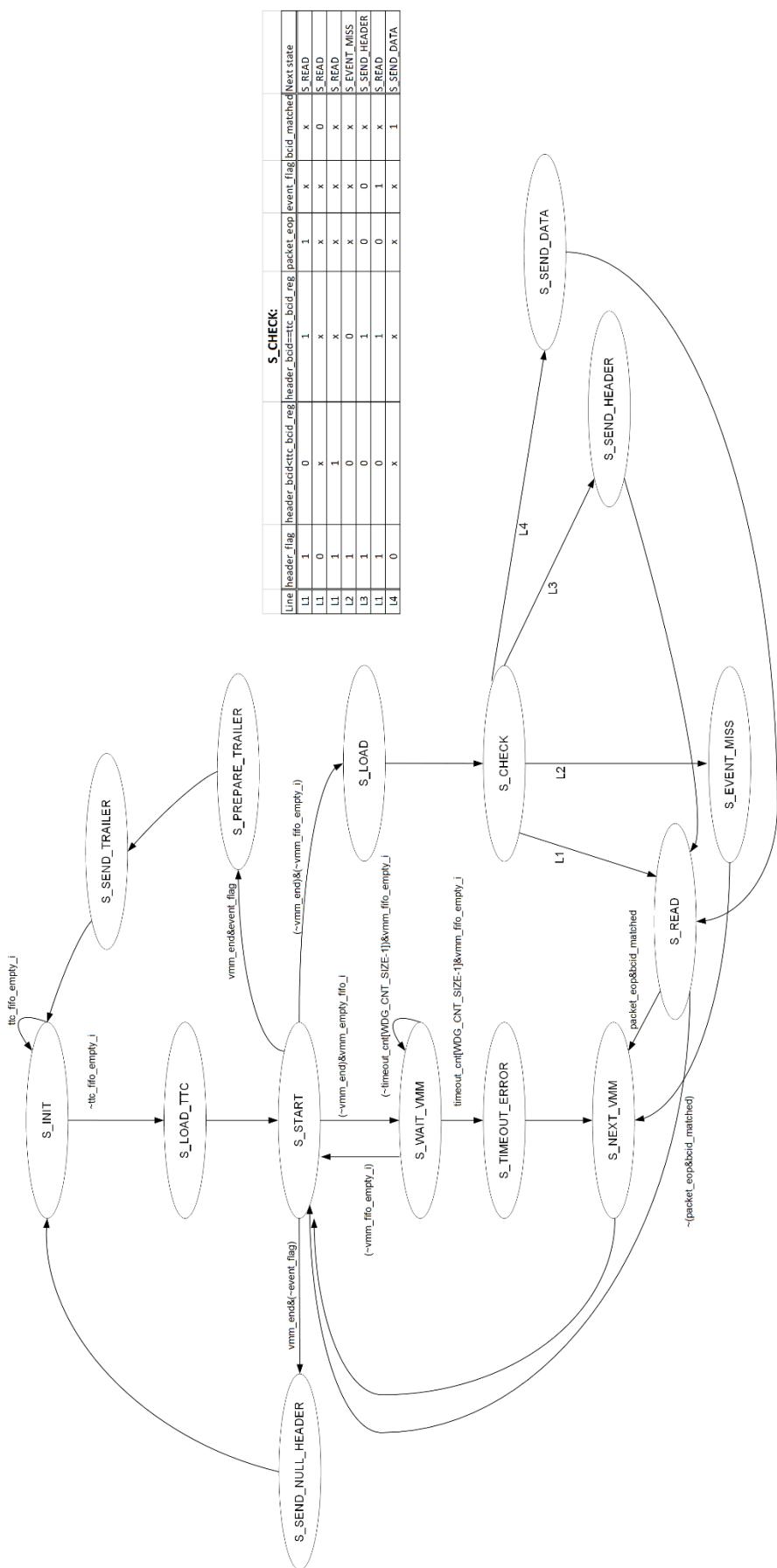
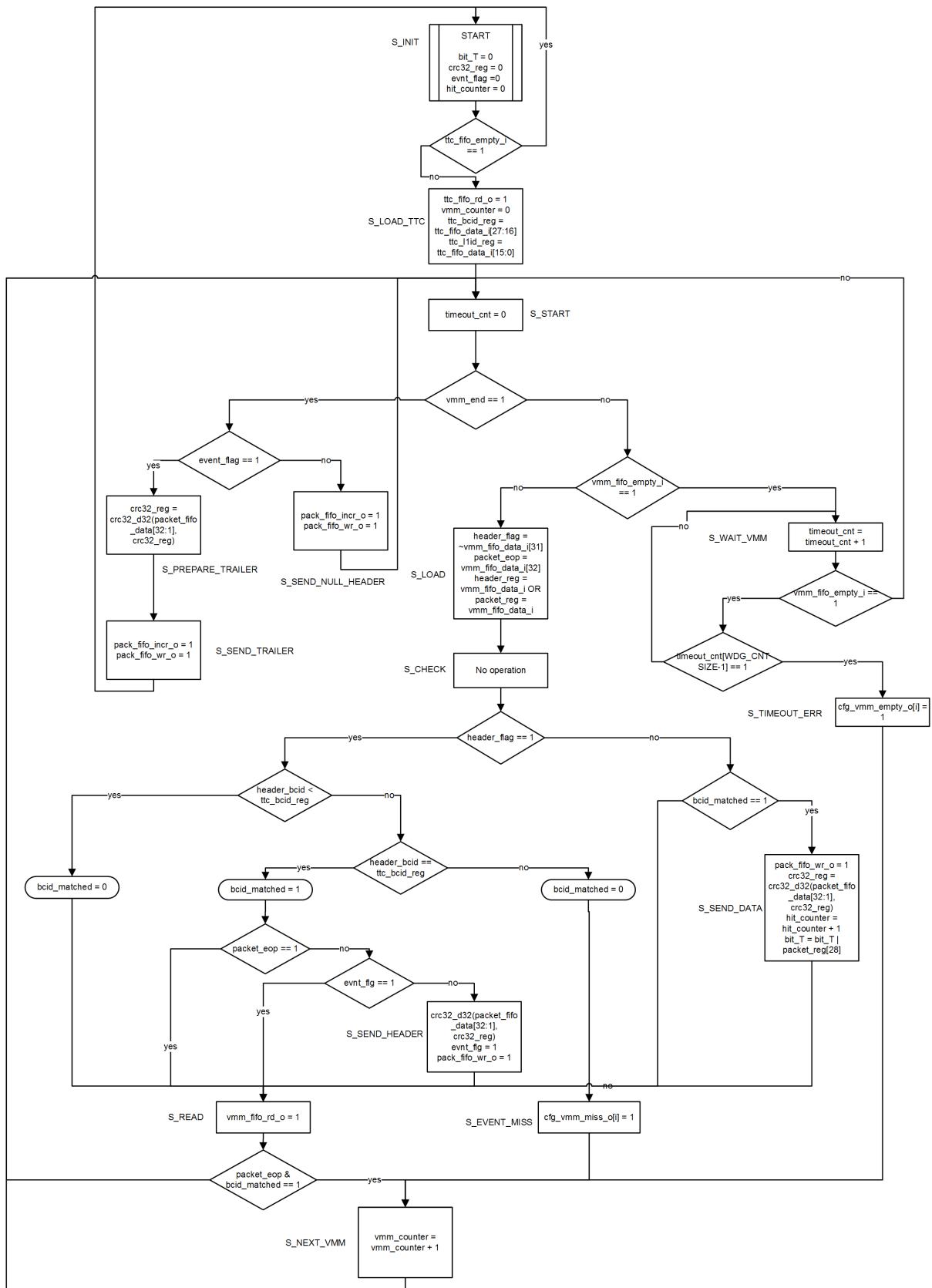


Figure 21: Packet Builder state machine diagram.



**Figure 22: The detailed flow diagram of the Packet Builder**

### 10.3.2 Packet FIFO

The format of the 34-bit data words that are stored in the Packet FIFO is presented in Figure 23.

	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
null evt	1	0	1	Level-1 ID (6)										24'b0										1															
hit header	1	0	0	orbit										BCID (12)										Level-1 ID (16)						0									
hit data	0	P	VMMid										Chan# (6)										ADC (10)										0						
trailer	0	T	R	Level-0 ID (12)										checksum (8)										TDC (8)						N	rel BCID		0						
message	1	1	msg type		RC	VMMid	msg ID (4)		variable length message (20)																				1										

Figure 23: Format of the data in the Packet FIFO.

### 10.3.3 TTC FIFO

The module stores 28-bit words, which represent (BCID, L1ID) pairs received from the TTC Decode module.

### 10.3.4 Streamer

The Streamer module creates the sROC output data stream, which is divided into 32-bit words. The block diagram is depicted in Figure 24. The module reads the data in the Packet FIFO and forwards it, while adding SOP (K28.4) and EOP (K28.6) characters where appropriate. The module also sends comma characters (K28.5) continuously if there are no full packets in the Packet FIFO. The structure of the module is presented in Figure 25, while the flow diagram of the internal state machine is depicted in Figure 26.

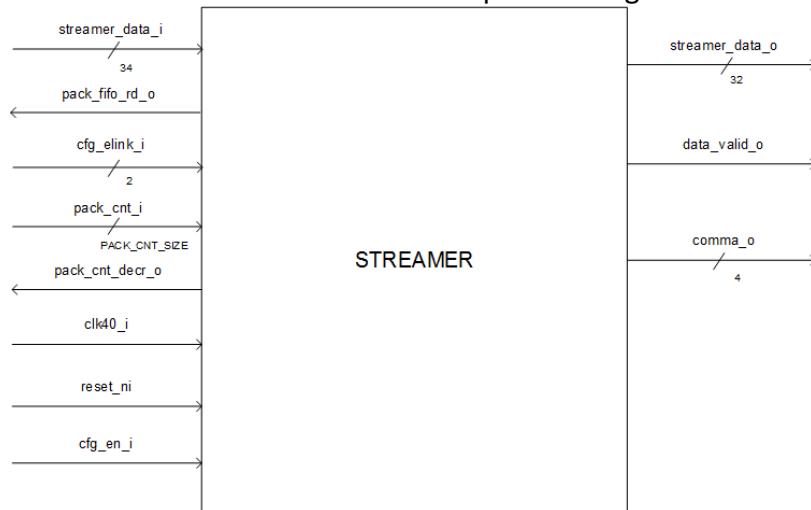


Figure 24: Streamer block diagram.

Table 13: The streamer interface

Port	Direction	Dim	Description
clk40_i	in	1	40 MHz clock signal
reset_ni	in	1	Asynchronous reset, active low
cfg_en_i	in	1	Enable signal
cfg_elink_i	in	2	Configures E-Link output rate
streamer_data_i	in	34	Input data, from Packet FIFO
packet_fifo_rd_o	out	1	Read signal to Packet FIFO
pack_cnt_i	in	PACK_CNT	Number of packets in Packet FIFO

		<u>_SIZE</u>	
pack_cnt_decr_o	out	1	Decrement packet counter in Packet FIFO
streamer_data_o	out	32	Output data (4 concatenated 8-bit words), to sROC ENC
data_valid_o	out	1	Indicates the output data is valid
comma_o	out	4	Indicates which of the 4 output words is a comma

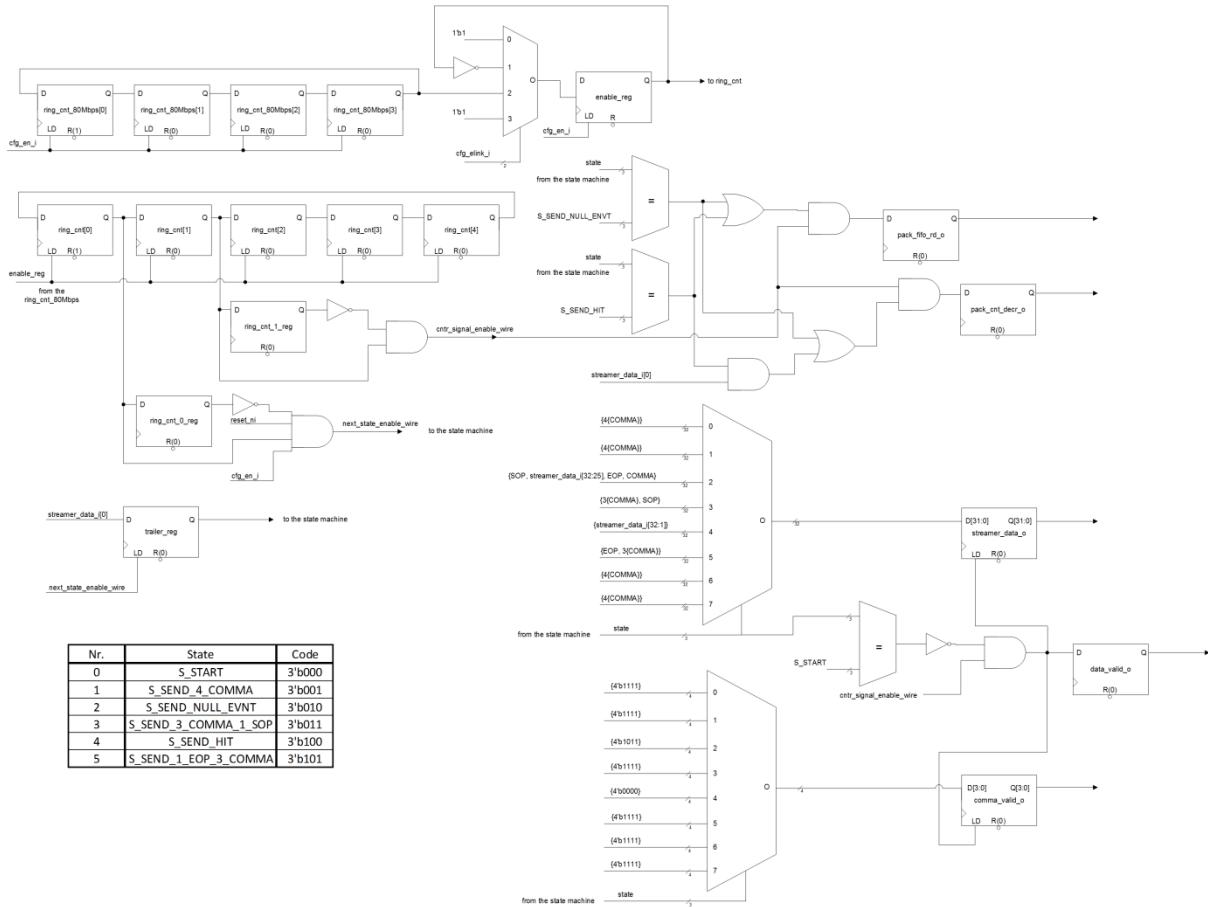


Figure 25: Streamer internal structure.

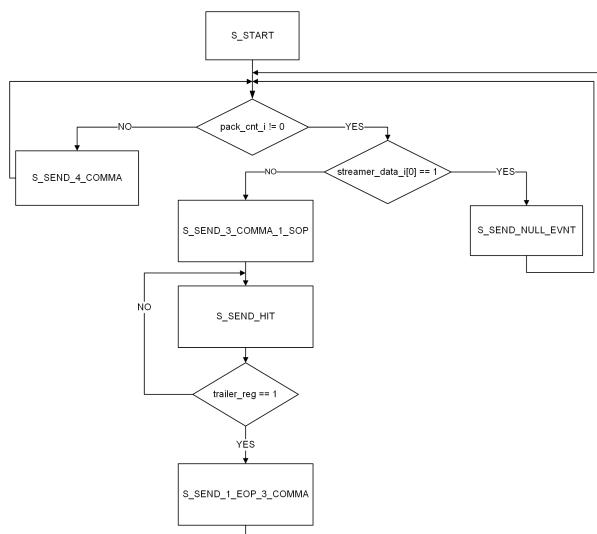


Figure 26: Streamer state machine flow diagram.

### 10.3.5 sROC ENC

The sROC ENC module receives 32-bit data words from the Steamer and outputs 40-bit words, which represent the 8b/10b encoding of the input. The block diagram of the module is presented in Figure 27 and the internal structure in Figure 28.

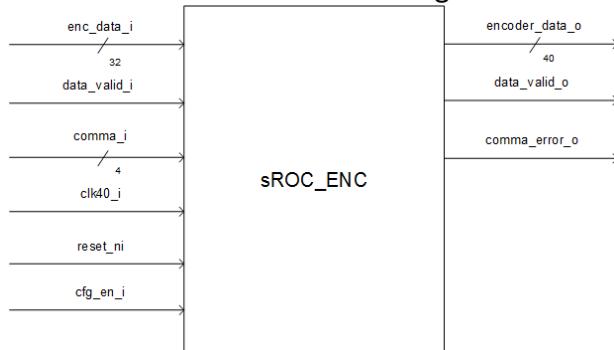


Figure 27: sROC ENC block diagram.

Table 14: The sROC ENC interface

Port	Direction	Dim	Description
clk40_i	in	1	40 MHz clock signal
reset_ni	in	1	Asynchronous reset, active low
cfg_en_i	in	1	Enable signal
enc_data_i	in	32	Input data, 4 concatenated 8-bit words from Steamer
data_valid_i	in	1	Indicates the input data is valid
comma_i	in	4	Indicates which of the 4 input words is a comma
encoded_data_o	out	40	Output data, to Feeder
data_valid_o	out	1	Indicates the output data is valid
comma_error_o	out	1	Indicates an error occurred while encoding a comma

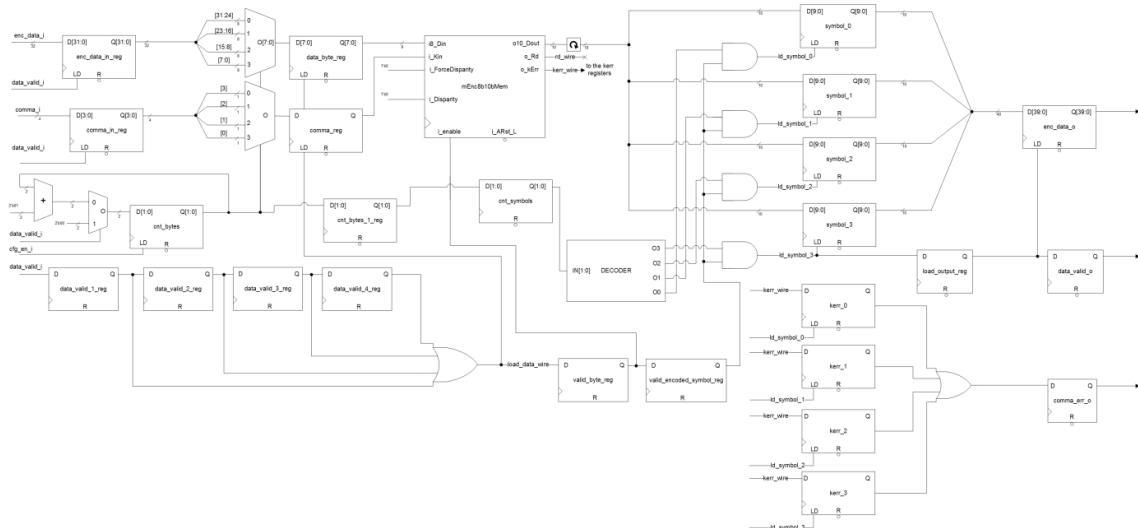


Figure 28: sROC ENC internal structure.

The module uses the free SGMII-IP-Core, which contains an 8b/10b encoder and decoder written in Verilog. These can be found at this link: <http://fpga-ipcores.com/8b10b-encoder-decoder/>.

This encoder has the following block diagram:

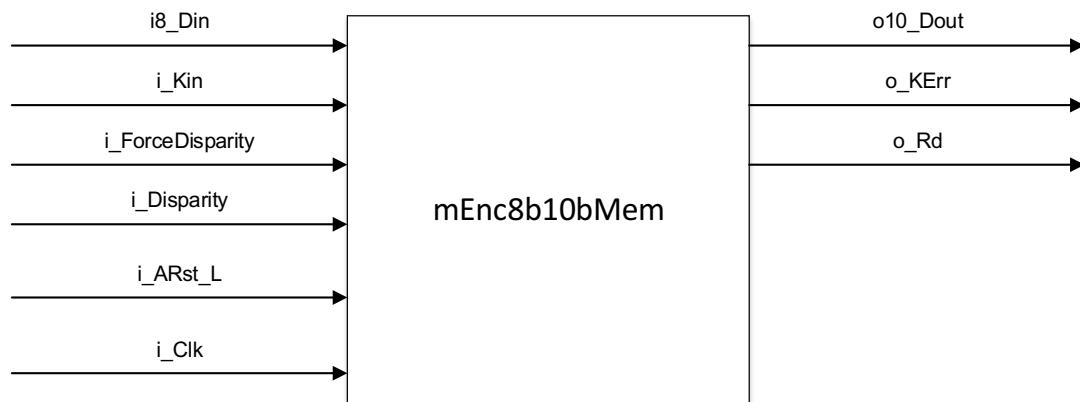


Figure 29: Block diagram of the SGMII-IP-Core 8b/10b encoder

The interface of this encoder module is described in the following table:

Table 15: SGMII-IP-Core 8b/10b encoder interface

Name	Size	Required	Description
i8_Din	8	Y	8-bit data input to be encoded (HGFEDCBA)
i_Kin	1	Y	Control character
i_Disparity	1	Y	Disparity input, required when i_ForceDisparity is asserted. 1: Even Disparity 0: Odd Disparity

i_ForceDisparity	1	Y	Force disparity to override the internally computed disparity
o10_Dout	10	Y	10-bit data output (abcdeifghj) Notice that the output is bit reversed, a is the “lsb” and should be transmitted first by the serializer Output is available 1 clock after the input
o_Rd	1	N	Running disparity The running disparity will be used to decode the current 10-bit input 1: Even Disparity 0: Odd Disparity
o_Kerr	1	N	Error when encoding K character
i_Clk	1	Y	Clock input
i_ARst_L	1	Y	Active low reset After reset, the running disparity is initialized to “negative”

### 10.3.6 Feeder

The Feeder receives 40-bit data words from the sROC Encoder and outputs an 8-bit word every clock cycle to the E-Link TX. The block diagram and the internal structure of the module are presented in Figure 30 and Figure 31, respectively. The output rate of the sROC, which determines the format of *feeder\_data\_o* is configured via the *cfg\_elink\_i* input, as follows:

<i>cfg_elink_i</i>	Output rate [Mb/s]
00	320
01	160
10	80

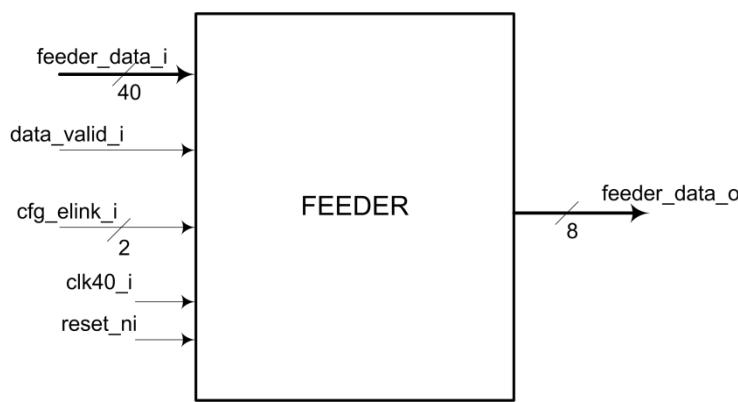
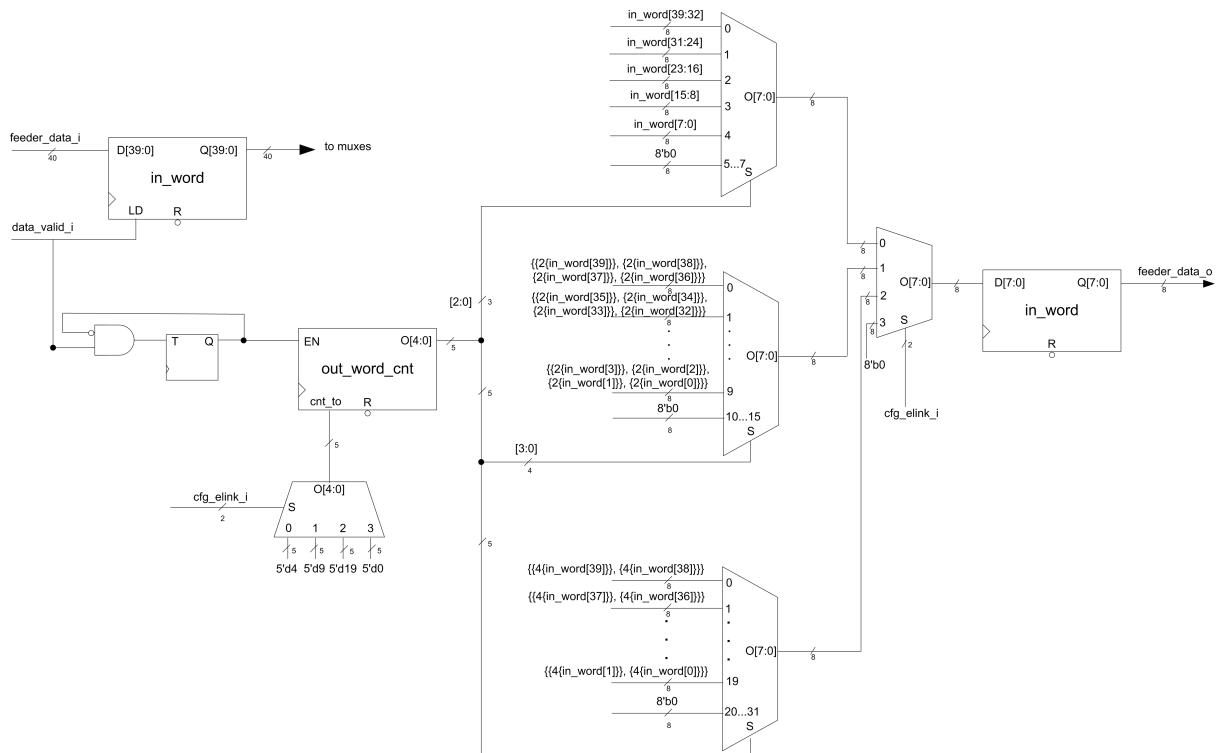


Figure 30: Feeder block diagram.

**Table 16: The Feeder interface**

<b>Port</b>	<b>Direction</b>	<b>Dim</b>	<b>Description</b>
clk40_i	in	1	40 MHz clock signal
reset_ni	in	1	Asynchronous reset, active low
cfg_elink_i	in	2	Configures E-Link output rate
feeder_data_i	in	40	Input data, from sROC ENC
data_valid_i	in	1	Indicates the input data is valid
feeder_data_o	out	8	Output data, to E-Link TX

**Figure 31: Feeder internal structure.**

### 10.3.7 E-Link TX

The E-Link TX serializes the data received from the Feeder and sends it via the differential output *elink\_data\_o*. The (8b/10b) encoded data is output LSB-first. The block diagram is presented in Figure 32, while the structure of the module is depicted in Figure 33.

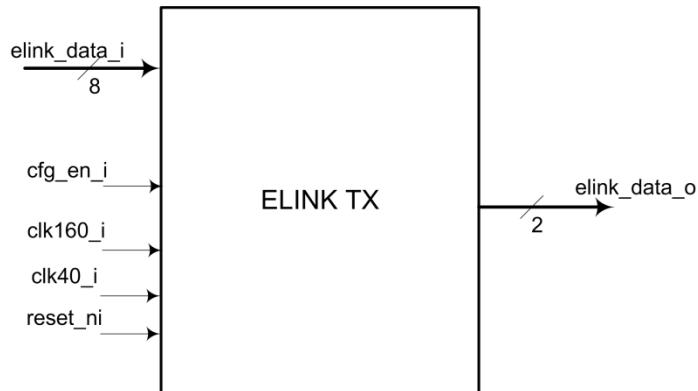


Figure 32: E-Link TX block diagram.

Table 17: The E-link TX interface

Port	Direction	Dim	Description
clk40_i	in	1	40 MHz clock signal, for receiving data from Feeder
clk160_i	in	1	160 MHz clock signal, for data transmission
reset_ni	in	1	Asynchronous reset, active low
cfg_en_i	in	1	Enable signal
elink_data_i	in	8	Input data, from Feeder
elink_data_o	out	2	Differential output signal

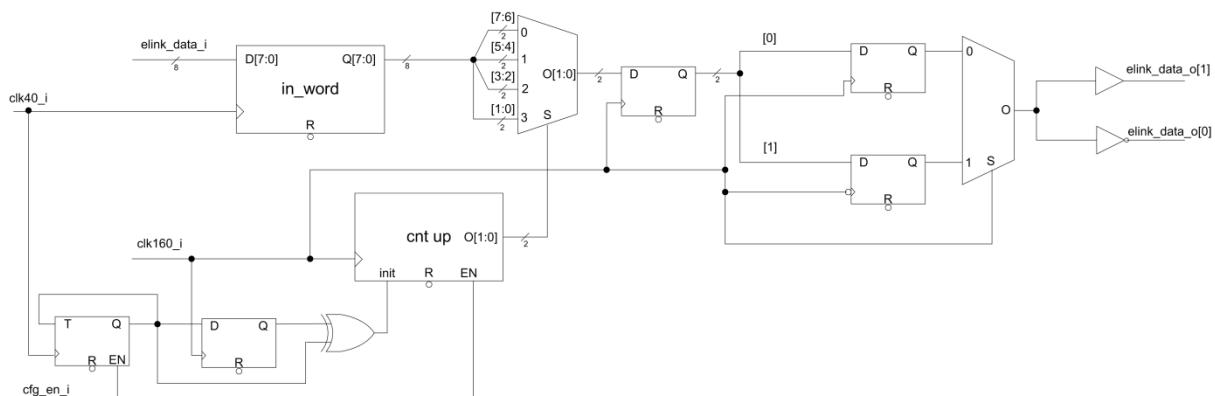


Figure 33: E-Link TX internal structure.

## 11 Radiation Tolerance, EMC Issues and Other Special Requirements

If some level of radiation tolerance is required, state the minimum total ionizing dose (TID) and non-ionizing fluence after which the component must remain functional and meet all other specifications. Also, describe the allowable single event upset (SEU) rate, possibly by sections of the component if the allowable rate is not the same across the whole component. If there are other special requirements beyond those covered in other sections describe them here.

If the component is to be used in presence of magnetic field, state the maximum field value and any special requirement such as the need for using non magnetic components.

If there are no radiation tolerance or other special requirements, then state "Not Applicable" for this section.

EMC issues must be documented (requirements, implementation, etc.).

Special safety requirements (e.g. need for halogen free cables) are to be included here.

## 12 Testing, Validation and Commissioning

Describe testing procedures that will be used to demonstrate that the fabricated component meets the specifications. This should include radiation testing if radiation tolerance is one of the requirements. Specify the testing features included in the system in view of easily test and qualify the component during production (e.g. BIST included in ASICs or systems).

It is not required to complete this section prior to the first specification review but it should be completed prior to the PDR. Prior to the FDR, the production testing must be described, and prior to the PRR, commissioning plans must be included.

## 13 Reliability Matters

### 13.1 Consequences of Failures

Describe the consequences to the detector of a failure of one unit of this component, e.g. x% of the sub-detector channels will be lost, or one stave or petal could overheat causing delamination of its component parts. The severity of the consequences will determine the level of reliability required and the level to be validated by QA and QC procedures defined in sections 12.4 and 12.5.

### 13.2 Prior Knowledge of Expected Reliability

Based upon industry experience, collaboration experience or personal experience, give an estimate of the reliability of this component.

### 13.3 Measures Proposed to Insure Reliability of Component and/or System

Include such measures as conservative design techniques (give specific examples), redundancy and possibilities to replace failed part. If failed part could be replaced, estimate the difficulty and time involved for installing replacements,

### 13.4 Quality Assurance to Validate Reliability of Design and Construction or Manufacturing Techniques

Describe what stress tests will be applied during the development period to validate the reliability of this component. Give a brief outline of any appropriate reliability theory being used. These tests could involve destructive tests.

It is not required to complete this section prior to the first specification review but it must be completed prior to the PDR. It is strongly recommended that these plans be reviewed and approved prior to the actual PDR to avoid the possibility of failing the PDR and thus delaying the fabrication or construction of the prototype parts,

### **13.5 Quality Control to Validate Reliability Specifications during Production**

Describe what stress tests will be applied during production, possibly on a sampling basis, to validate the reliability of production units. These could likely be destructive tests. Specify the required sampling percentage of production units.

It is not required to complete this section prior to the first specification review but it must be completed prior to the FDR. It is strongly recommended that these plans be reviewed and approved prior to the actual FDR to avoid the possibility of failing the FDR and thus delaying the fabrication or construction of the pre-production parts,

## **14 References**

If there are any references pertaining to the development of this component, list them here.