FPGA - Fibre Channel 16G

FMAC

Design Specification

January 27, 2014
Revision 1.2

REVISION HISTORY

Revision	Date	Description	By
0.1	9/18/2013	initial draft	Gene Shen
1.0	11/20/2013	Major update – prior to design review	Gene Shen
1.1	11/21/2013	Endian clarification, changes due to design review, change bars	Gene Shen
1.2	1/27/2014	Major update to match RTL edits. Added credit stats chapter	Gene Shen

Table of Contents

ABBREVIATIONS & DEFINITIONS

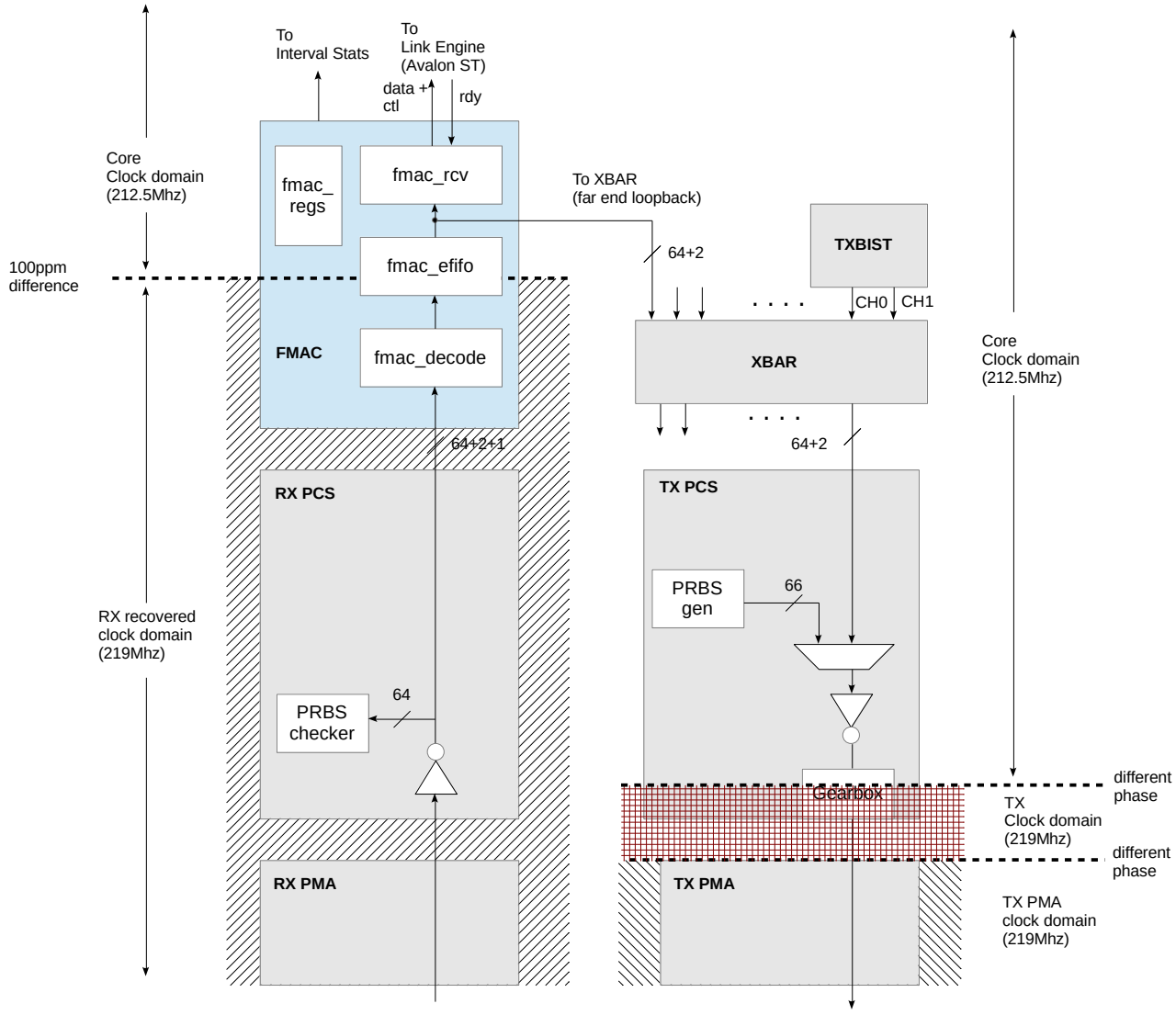
Abbreviation & Words	*Definition*
8b/10b	Channel encoding that encodes 8 bits of data into a 10 bit word used in 8GFC
64/66	Channel encoding used in 10GE & 16GFC
Channel	A term to describe the ingress data path in the FPGA.
CDR	Clock Data Recovery – mechanism for recovering data from a high speed serial data stream
CRC	Cyclic Redundant Check
DAL	Data Acquisition Layer – The PCB with FPGA
DDR	Double Data Rate RAM.
DPL	Data Processing Layer – Motherboard
EOF	End Of Frame. A Primitive that marks the end of a Frame.
FCS	Frame-check-sequence (CRC) for 10G Ethernet
FC-0	The physical Fibre Channel medium.
FC-1	FC encoding and decoding
FC-2	FC frame and signaling
FC-3	FC routing through fabric
FC-4	The ULP layer
FPGA	Field programmable gate array.
Frame	A delimited group of data sent between devices in the Fibre Channel
GB/s	Giga Bytes per second
Gbps	Giga bits per second
Hard IP Block	A specialized piece of logic, implemented in silicon, that can exist in the FPGA such as the SERDES or PCIe blocks.
Interval	The duration that counters increment before being sent to DPL
Link	A term that describes two channels whose frames arrived from the same customer link.
LOSync	Loss of synchronization. Refers to loss of transmission word synchronization.
LOSig	Loss of signal. Indicates SFP transceiver has reported no optical signal. May also indicate the SERDES CDR is no longer locked.
MAC	Media Access Controller.
MFps	Millions of Frames per Second.
Probe	The device that monitors Fibre Channels links.
RAM	Random Access Memory.
SAN	Storage Area Network.
SERDES	A serializer / deserializer that converts data from a parallel interface to a serial interface, and vice – a –versa to provide interconnect to optical SFP's.
SOF	Start Of Frame. A Primitive that marks the beginning of a Fibre Channel Frame.

RX A receive data path of a Link.
TX A transmit data path of a Link.

1.0 INTRODUCTION

1.1 Overview

The fibre channel MAC (FMAC) implements FC1(Physical Coding) and FC2 (Signaling protocol) layers for 16G Fibre Channel monitoring. It interfaces with a 64/66 encoded 16G fibre channel PCS using a 64b data / 2b header interface. The PCS interface is qualified by a 2 control signals to indicate data validity. The FMAC sends receive data to the link engine through a 64b FIFO interface and a latching/clearing interval stat interface. It also sends 64b receive data to the XBAR to implement far-end loopback. Below, is a block diagram of the FMAC.



1.2 Clocking

The front-end of the FMAC is clocked by a 219Mhz recovered clock generated by the fibre channel SERDES. The PCS interface, the FMAC decoder, and the FMAC EFIFO write interface are clocked by the recovered clock. The back-end of the FMAC is clocked by a 212.5Mhz core clock. The FMAC EFIFO read interface, the FMAC registers, and the FMAC receive unit are clocked by the core clock.

The FMAC elastic FIFO is responsible for crossing clock domains from the recovered clock to the core clock. These two interfaces are rate matched:

Core : 66b (64b data + 2b hdr) @ 212.5Mhz = 14.0125G
Recovered : 64b @ 219Mhz = 14.0125G

However, the clocks may have up to a 100PPM frequency difference and the elastic FIFO add/inserts IDLES between frames to prevent overflow/underflow conditions.

1.3 Interfaces

1.3.1 PCS Interface

The PCS interface sends recovered data to the FMAC decoder. The PCS interface is qualified by a sync and valid bit. If either sync or valid are negated, the PCS data is ignored.

1.3.2 Link Engine

The FMAC presents two interfaces to the link engine. The first interface provides interval stats which are packaged and delivered to software at periodic intervals. The second interface works through a FIFO and transfers processed fibre channel frames.

1.3.3 Credit Stats

The FMAC tracks buffer to buffer credit stats for packaging through the interval stats interface. An interface to the other channel in the link provides R_RDY primitives. It is important that the two channels in a link are paired correctly for credit stat monitoring.

1.3.4 XBAR

PCS data which has been rate matched to the core clock is sent to the XBAR to be transmitted back out the same channel. This implements a far-end loopback mode using parallel data. This enables us to invert receive data streams to handle board inversions. Loopback modes in the SERDES do not provide access to data inversions.

XBAR data from the FMAC is already rate matched to transmit rates, so there will be no overflow/underflow conditions. However XBAR data may experience initial startup delay before becoming valid for transmit.

1.3.5 Control Register Interface

A standard control register interface accesses the FMAC registers. The FMAC registers live in the core clock domain. When required, synchronizers transfer signals to/from the core clock domain to the recovered clock domain. Synchronization modules are implemented in all critical address/control/data transfers when performing control register reads and writes. However in many cases, synchronizers are not implemented when transferring multi-bit register value to/from control registers to the core. This simplification significantly reduces register counts.

1.4 Key Features

Decoder

- Performs Fibre Channel Spec compliant 16G decode with error checking.
- Supports full 16G wire speed using a 64b interface.
- Converts code violations to idle special functions.
- Generates a detailed error vector to identify decode errors.
- Detects primitive sequences/signals.
- Detects SOF/EOF sequence errors per fibre channel guidelines.
- Monitors linkup events.

Elastic FIFO

- Implements configurable high threshold low threshold, and read thresholds.
- Handles PPM clock differences between write and read clock domains.
- Auto-inserts IDLEs and primitive sequences as required to prevent FIFO overflow/underflow.

Credit Stats

- Tracks and maintains buffer to buffer credit counts.
- Provides credit stats through the interval stats interface.

Receive Unit

- Performs CRC32 checking at wire speed.
- Checks frame minimum and maximum lengths with programmable frame length limits.
- Strips NOS, OLS, LR, LRR primitives from frame.
- Strips SOF, EOF delimiters and IDLEs from frame.
- Strips CRC fields from frame.
- Checks for frame delimiter sequence errors.
- Detects minimum IDLE interpacket gap violations.
- Implements a 64b Link Engine FIFO interface with support for stalls.

- Tracks and maintains buffer to buffer credit stats.
- Tracks and maintains interval stats.
- Re-aligns fibre channel frame payloads to a 64b boundary.

Stats and Registers

- Extensive set of debug and stat counters, including error conditions
- Contains debug register to provide visibility of internal state.
- Complete coverage of all FIFO overflow/underflow conditions.
- Complete coverage of state machine states.

1.5 Related Documents

- FC_FS_2_Framing_Signalling_4_3_2006_rev1_00.pdf (Fibre Channel Framing and Signalling -2, rev 1.00)

- FC_FS_3_Framing_Signalling_10_22_2010_rev1_11.pdf (Fibre Channel Framing and Signalling - 3, rev 1.11)

- FC_FS_4_Framing_Signalling_3_14_2013_rev0_20.pdf (Fibre Channel Framing and Signalling - 4, rev 0.20)

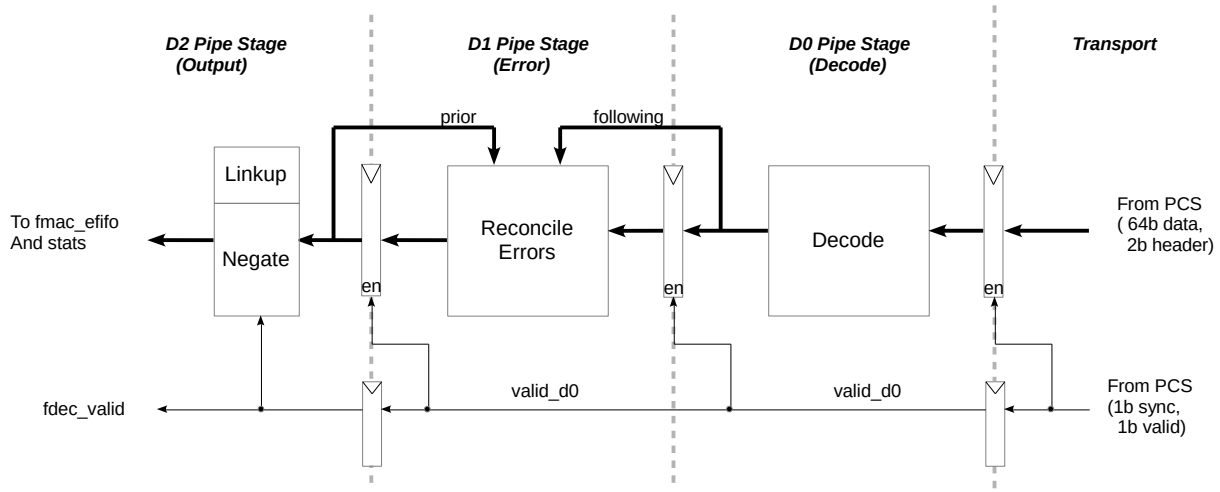
1.6 Endianess

The FMAC is coded in little endian format (first byte transferred is located in lsb – bits [7:0]). The PCS interface supports endianess swap and should be configured little endian. The FMAC interface into the link engine is programmable and transfer data in little endian or bit endian format. The default endian setting from the FMAC to the link engine is big endian.

2.0 FMAC DECODE UNIT

2.1 Overview

The FMAC decode module is clocked by the RX recovered clock from the fibre channel SERDES. It receives 64b data, 2b header, and 2b of control from the RX PCS. The FMAC decoder is responsible for decoding 64b transmission words and detecting decode errors. Decode attributes are used by the elastic FIFO and the link engine / interval stat interface.



The decoder has a 3 cycle pipeline from decode to valid attributes and error conditions. One (or more) cycles are allocated to transport PCS information to the transmission word decoders. This delay is not included in the 3 cycle pipeline.

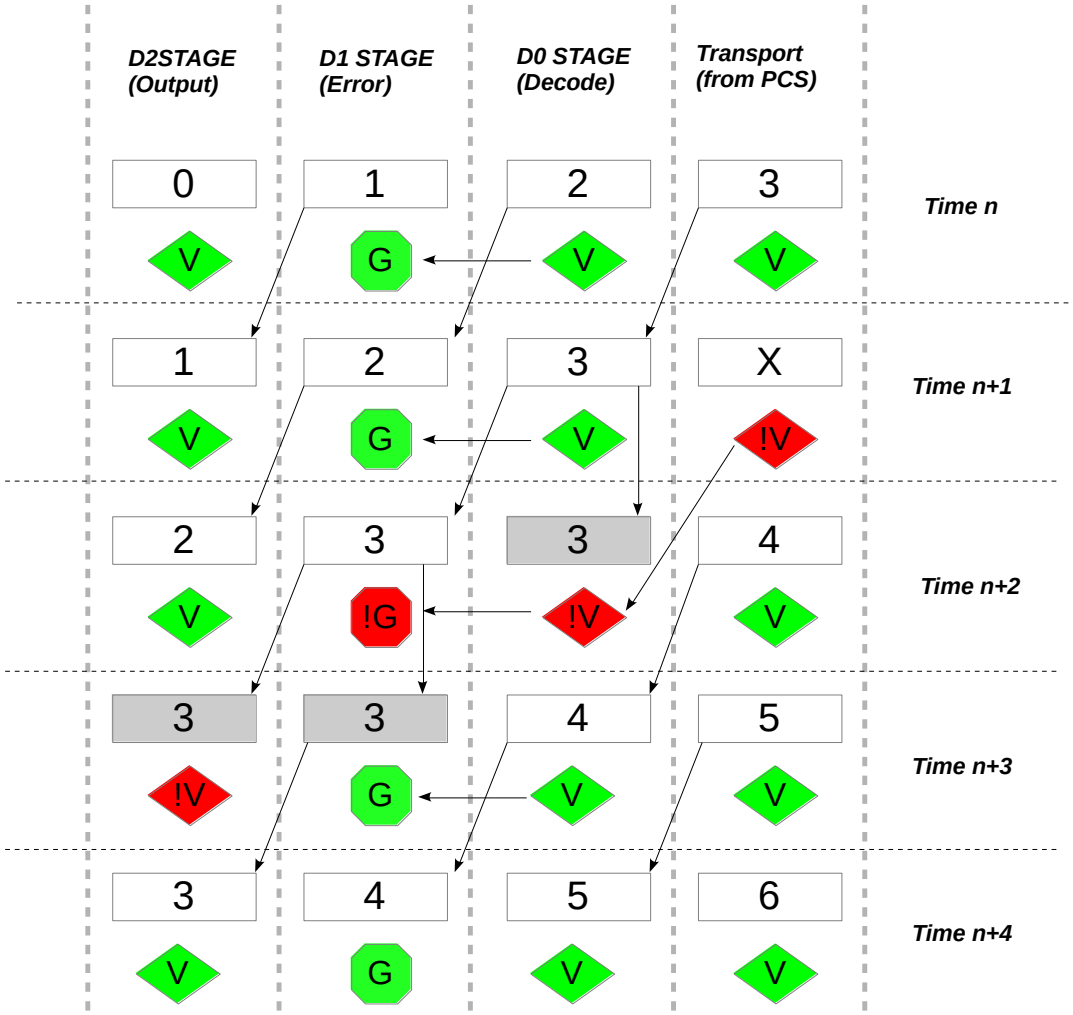
The first cycle performs transmission word decode. During transmission word decode, frame delimiters (SOF, EOF), primitives, and idles are identified from the PCS data. The next cycle reconciles coding violations. In order to handle coding violations, the reconcile errors block peeks at prior and following decode state from the two adjacent pipeline stages. Checking against prior and following decode results follows fibre channel guidelines for decoding SOFs and EOFs. The last cycle outputs data to the fmac_efifo and stats block.

The PCS indicates whether data is valid through a sync and valid control interface. If either of these signals is negated, the decode pipeline is held and eventually, decode outputs are suppressed in the negate block in the D2 pipe stage. Although decoder outputs are qualified by a valid, enables to stats registers must be negated when the decode pipe is held to prevent double counting.

Lastly, the decode unit has a simple linkup state machine which tracks link state. The linkup state machine monitors block sync from the PCS and primitives/idles from the decoder to determine link state.

2.2 Pipeline Sequencing

The below diagram illustrates pipeline sequencing due to invalid PCS rx_data. PCS data packets are represented by blocks. Greyed blocks indicate PCS data which has been held from the previous cycle. The state of each pipeline stage is represented by diamonds or octagons.



Since the D1 stage performs error reconciliation, it decides whether the decode operation it is currently performing is valid. A valid operation is marked as G in the diagram and results in the output stage indicating valid data the next cycle. An invalid operation due to stale decode packets in adjacent stages results in an invalid decode operation, with decode results marked as invalid in the next cycle.

This section provides a step by step description of pipeline operations:

- At time n, there are 4 PCS packets in flight. All packets are valid and packet 0 is output as a valid packet.

- At time n+1, the PCS interface bubbles and the packet in the transport stage is marked invalid. During this cycle, the 3 in flight packets 1-3 continue operation, and packet 1 is output with a valid.

- At time n+2, pipe stage D0 (decode) is held due to invalid PCS data in the previous cycle. Pipe stage D1 (error) marks its decode operation as !Good since the decode information in pipe stage D0 is invalid. Since the decode operation in pipe stage D1 (error) was good in time n+1, output data in pipe stage D2 (output) is marked valid. Finally, PCS data in the transport stages becomes valid.

- At time n+3, pipe stages D1 (error) is held due to a !G decode operation in pipe stage D1 and D0 the previous cycle. Pipe stage D0 (decode) is updated with incoming PCS packet 4. Since packet 3 in pipe stage D1 (error) now has correct adjacent packets, the decode operation is marked Good. Lastly, decode data in pipe stage D2 is marked invalid due to a !Good operation in D1 (error) the previous cycle , and stat register enables are suppressed.

- At time n+4, the decode pipeline is back in full operation with all in flight packets valid.

Because the decode pipeline interlocks and holds – it is able to sequence any number of PCS invalid packets. Architecturally however, the PCS interface should only bubble the data once every 32 cycles.

2.3 64/66 Transmission Word Synchronization

A fibre channel port has two mutually exclusive states:

- Word synchronization acquired (Sync)
- Loss of word synchronization (LOSync)

While synced, the FMAC decodes, checks for errors, and passes information to the receive unit and link engine. During LOS, the decoder does not perform decoding or error checking, and negates all attributes.

With 64b/66b encoded frames, synchronization is acquired under the 3 following conditions:

1. Block synchronization from PCS block lock state machine (or FEC block synchronization)
2. BER monitor is below HIGH threshold.
3. Receiver does not report loss-of-signal

The PCS module is responsible for tracking 64/66 transmission word synchronization according to Fibre Channel framing specifications. The term "SYNC" indicates transmission word synchronization has been acquired under the 3 previous conditions. Transmission word synchronization is maintained regardless of the presence of coding violations – the two conditions are not related.

2.4 Decoder – Pipe Stage D0

The FMAC decoder receives block synchronized 64b transmissions words and decodes them into fibre channel special functions and primitives. The decoder waits for 64/66 transmission word synchronization before starting decode. If transmission word synchronization has not been established, decode attributes are negated and error checking is suppressed. The decoder performs all decode in a single cycle.

2.4.1 Control Transmission Words

64b transmission words are either all data or control. Data transmission words contain 64b of data. Control transmission words contain two special functions, or a special function and data. The encodings in fibre channel are a subset of the valid encodings specified in the 10GBASE-R specifications.

Table 2-1: Table of control groups from 10GBASE-R – invalid encodings for fibre channel are marked in grey.

Input Data	**Sync**	**Block Payload**							
Bit Position:	0 1	2							65
Data Block Format:									
$D_0\ D_1\ D_2\ D_3/D_4\ D_5\ D_6\ D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
Control Block Formats:		**Block Type Field**							
$C_0\ C_1\ C_2\ C_3/C_4\ C_5\ C_6\ C_7$	10	0x1e	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$C_0\ C_1\ C_2\ C_3/O_4\ D_5\ D_6\ D_7$	10	0x2d	C_0	C_1	C_2	C_3	O_4	D_5	D_6	D_7
$C_0\ C_1\ C_2\ C_3/S_4\ D_5\ D_6\ D_7$	10	0x33	C_0	C_1	C_2	C_3		D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3/S_4\ D_5\ D_6\ D_7$	10	0x66	D_1	D_2	D_3	O_0		D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3/O_4\ D_5\ D_6\ D_7$	10	0x55	D_1	D_2	D_3	O_0	O_4	D_5	D_6	D_7
$S_0\ D_1\ D_2\ D_3/D_4\ D_5\ D_6\ D_7$	10	0x78	D_1	D_2	D_3	D_4		D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3/C_4\ C_5\ C_6\ C_7$	10	0x4b	D_1	D_2	D_3	O_0	C_4	C_5	C_6	C_7
$T_0\ C_1\ C_2\ C_3/C_4\ C_5\ C_6\ C_7$	10	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0\ T_1\ C_2\ C_3/C_4\ C_5\ C_6\ C_7$	10	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7

$D_0 D_1 T_2 C_3/C_4 C_5 C_6 C_7$	10	0xaa	D_0	D_1	C_3	C_4	C_5	C_6	C_7
$D_0 D_1 D_2 T_3/C_4 C_5 C_6 C_7$	10	0xb4	D_0	D_1	D_2	C_4	C_5	C_6	C_7
$D_0 D_1 D_2 D_3/T_4 C_5 C_6 C_7$	10	0xcc	D_0	D_1	D_2	D_3	C_5	C_6	C_7
$D_0 D_1 D_2 D_3/D_4 T_5 C_6 C_7$	10	0xd2	D_0	D_1	D_2	D_3	D_4	C_6	C_7
$D_0 D_1 D_2 D_3/D_4 D_5 T_6 C_7$	10	0xe1	D_0	D_1	D_2	D_3	D_4	D_5	C_7
$D_0 D_1 D_2 D_3/D_4 D_5 D_6 T_7$	10	0xff	D_0	D_1	D_2	D_3	D_4	D_5	D_6

2.4.2 Transmission Word Type

Data transmission words do not have a type field. The most significant byte in the control transmission word specifies the type. Valid types are listed in the below table. Invalid transmission word types are considered a coding violation and are decoded as two idles per fibre channel requirements.

Table 2-2: Table of valid transmission word types [1]

Transmission Word type value	Transmission Word content	Reference
1Eh	Idle Special Function followed by Idle Special Function; or Receiver Error	5.3.6.1 5.3.6.10
33h	Idle Special Function followed by SOF Special Function	5.3.6.2
B4h	EOF Special Function followed by Idle Special Function	5.3.6.3
2Dh	Idle Special Function followed by other Special Function	5.3.6.4
4Bh	Other Special Function followed by Idle Special Function	5.3.6.5
55h	Other Special Function followed by other Special Function	5.3.6.6
66h	Other Special Function followed by SOF Special Function	5.3.6.7
78h	SOF Special Function followed by word of data	5.3.6.8
FFh	Word of data followed by EOF Special Function	5.3.6.9
any other value	Restricted for IEEE 802.3-2008, shall not be transmitted	IEEE 802.3-2008

2.4.3 Order Code Decode

[1] From "Fibre Channel Framing Signaling – 3"

EOF/SOF/other special functions are qualified by a 4b order code. The 4b order code contains two valid encodings. The valid encodings correspond with the control word type – please refer to the fibre channel specification for details. Any other value results in a code violation and the special function is decoded as an idle special function.

Table 2-3: Table of valid order fields[2]

Value	Ordered Set	Reference
0h	Primitive Sequence	5.3.7.3
Fh	Primitive Signal	5.3.7.2
any other value	Restricted for IEEE 802.3, shall not be transmitted	IEEE 802.3-2008

2.4.4 Idle Special Functions

Idle special functions are qualified by four 7b control fields. The only valid control code is `00h` or Idle. All other control codes are considered a coding violation and the invalid control code is decoded as an idle. The FC16 PCS layer does not generate RCV_ERROR control codes (`1Eh`) and the appearance of this control code is also considered a coding violation.

Table 2-4: Table of valid control codes[2]

Value (least significant seven bits)	Meaning	Reference
00h	Idle	5.3.7.2
1Eh	Error. This code shall be used only for receiver error reporting (see 5.3.6.10)	5.3.6.10
any other value	Restricted for IEEE 802.3-2008, shall not be transmitted	IEEE 802.3-2008

2.4.5 Brocade ARBFF Idle Special Function

Brocade implements a proprietary IDLE fill word. This fill word may be used to specify virtual channel assignments in ISL (Inter Switch Links). The FMAC does not currently support the Brocade IDLE fill word as the format is unknown.

2.4.6 SOF/EOF Special Functions

Frame delimiters SOF and EOF are qualified by a 4b order code and three modifier bytes. The 4b order code must always be 0. The three modifier bytes are encoded as shown in the below

[2] From "Fibre Channel Framing Signaling – 3"

table. Invalid modifier bytes are decoded as an EOFa per fibre channel requirements. All other encodings are invalid and considered coding violations.

Table 2-5: Table of valid frame delimiter codes[4]

Abbr.	Frame delimiter	Reference	Modifier Byte 1	Modifier Byte 2	Modifier Byte 3
SOF_{i2}	SOF Initiate Class 2	10.3.7.2.2	B5h	55h	55h
SOF_{n2}	SOF Normal Class 2	10.3.7.3.2	B5h	35h	35h
SOF_{i3}	SOF Initiate Class 3	10.3.7.2.3	B5h	56h	56h
SOF_{n3}	SOF Normal Class 3	10.3.7.3.3	B5h	36h	36h
SOF_{f}	SOF Fabric	FC-SW-5	B5h	58h	58h
EOF_{t}	EOF Terminate	10.3.8.2.2	95h	75h	75h
EOF_{a}	EOF Abort	10.3.8.3.2	95h	F5h	F5h
EOF_{n}	EOF Normal	10.3.8.2.1	95h	D5h	D5h
EOF_{ni}	EOF Normal-Invalid	10.3.8.3.3	8Ah	D5h	D5h

2.4.7 Other Special Functions

2.4.7.1 Overview

Other special functions are qualified by a 4b order code and 3 modifier bytes. Invalid codes are processed as an idle special function and marked as a coding violation.

Table 2-6: Table of other special function codes[3]

Abbr.	Primitive Signal	Reference	Order code	Modifier Byte 1	Modifier Byte 2	Modifier Byte 3
R_RDY	Receiver_Ready	19.4	Fh	95h	4Ah	4Ah
VC_RDY	Virtual Circuit Ready	FC-SW-5	Fh	F5h	VC_ID	VC_ID
BB_SCs	Buffer-to-Buffer State Change (SOF)	19.4.9	Fh	95h	96h	96h
BB_SCr	Buffer-to-Buffer State Change (R_RDY)	19.4.9	Fh	95h	D6h	D6h

Abbr.	Primitive Sequence	Reference	Order code	Modifier Byte 1	Modifier Byte 2	Modifier Byte 3

[3] From "Fibre Channel Framing Signaling – 3"

NOS (see NOTE)	Not_Operational	clause 7	0h	55h	BFh	45h
OLS	Offline	clause 7	0h	35h	8Ah	55h
LR	Link_Reset	clause 7	0h	49h	BFh	49h
LRR	Link_Reset_Response	clause 7	0h	35h	BFh	49h
NOTE	The representation of NOS used in this standard is consistent with the 8B/10B representation, and differs from that used in 10GFC (i.e., a REMOTE FAULT Primitive Sequence)					

2.4.7.2 Primitive Sequences and Signals

Primitive signals (or just primitives) are special functions for which each instance is relevant and independent of other special functions and primitives. Each 64b transmission word can have 0, 1 or 2 primitive signals each cycle. Primitive signals are:

1. R_RDY
2. VC_RDY
3. BB_SCS
4. BB_SCR

VC_RDY primitives are further qualified by a 16b VC_ID field for use in ISLs which support virtual channels. The 16b VC_ID is not currently used and is not generated by the decoder to limit RAM storage requirements.

Virtual Channel Number	VC_ID Value
00h	D0.0
01h	D1.0
02h	D2.0
...	see FC-FS-3
FDh	D29.7
FEh	D30.7
FFh	D31.7

Primitive sequences are transmitted continuously by a port while the condition exists. The fibre channel guide requires 3 consecutive instances of a primitive sequence without any intervening data before recognition by the receiver. The FMAC decode module matches this requirement and only signals a primitive sequence after 3 consecutive instances. Primitive sequences are:

1. NOS
2. OLS

3. LR
4. LRR

2.5 Reconcile Errors – Pipe State D1

2.5.1 Introduction

The FMAC decoder identifies invalid sync header and transmission word values, and marks them as code violations. These errors are only signaled when PCS data is synced and valid, otherwise errors are suppressed. Decode errors follow requirements specified by the Fibre Channel Framing and Signaling Guide – 3 (FC-FS-3).

There are two types of decode errors.

1. **Invalid encodings:** The first type of errors is performed in pipe stage D0, and consists of invalid transmission word encodings or sync header values. These decode errors are identified in a single cycle.

2. **Sequence Errors:** The second type of errors is performed in pipe stage D1 and requires peeking at decode results from prior or following cycles. These decode errors are called sequence errors because they are due to incorrect sequences of transmission words.

In the event of invalid transmission word types, the FMAC layer does **not** recode the transmission word type into the `rcv_error` transmission type. This preserves the contents of the 64b transmission word for later debug and inspection.

2.5.2 Decode Error Vector

All decode errors are reported in the `fdec_err_vec` status vector. Similar to decode attributes, the error vector is available 3 cycles after arrival of PCS hdr/data.

The below table outlines transmission word decode errors and consequences.

Table 2-7: Table of transmission word decode errors

Error	err_vec	Description	Consequence
Invalid synchronization header	[0]	- 2b header is not a control or data sync value.	- Decoded as two idles - marked as a coding violation
Invalid control word type	[1]	- Invalid and unrecognized type field in control transmission word. - rcv_error types are considered invalid. They are not expected from our PCS layer.	- Decoded as two idles - marked as coding violation
Invalid SOF	[2]	- SOF special function with invalid modifier byte or order code	- Decoded as an idle - SOF decode attribute negated

			- marked as a coding violation
Invalid EOF	[3]	- EOF special function with invalid modifier byte or order code	- Decoded as an idle - EOF decode attribute negated - marked as a coding violation
Invalid Other	[4]	- OTHER special function with invalid modifier byte or order code	- Decoded as an idle - OTHER decode attributes negated - marked as a coding violation
Invalid Idle	[5]	- IDLE special function with invalid control code	- Decoded as an idle - marked as a coding violation
SOF Sequence error	[6]	- SOF special function AND prior transmission word: a) was a data transmission word b) was any transmission word containing an SOF c) caused a coding violation to be reported	- Decoded as an idle - SOF decode attribute negated - marked as a coding violation
EOF Sequence error	[7]	- SOF special function AND prior transmission word: a) was a data transmission word b) was any transmission word containing an SOPF c) caused a coding violation to be reported	- Decoded as an idle - SOF decode attribute negated - err_vec[6] (SOF sequence error) asserted - marked as a coding violation

2.6 Linkup State Machine

The linkup state machine tracks link state. It basically has 2 states:

1. LINK_DOWN :
   - Default state on reset.
   - The link is considered down.
   - After 1 valid IDLE special function without error, go to LINK_UP and output a link up event, otherwise stay in LINK_DOWN.

2. LINK_UP :
   - The link is considered up.
   - On a NOS/OLS/LR/LRR primitive or PCS loss of sync, go to LINK_DOWN and output a link down event, otherwise stay in LINK_UP.

Current State	Input Condition	Next State	Output
LINK_DOWN	1 valid IDLE special function (two IDLEs) without error	LINK_UP	Linkup event
LINK_UP	NOS primitive OLS primitive LR primitive LRR primitive PCS loss of sync	LINK_DOWN	Linkdown event

2.7 Loss of Sync

In Dominica, loss of sync was generated by the More Than IP FC1/FC2 block. MTIP implemented a loss of sync state machine which tracked coding errors (8b10b not in table, and disparity errors). High rates of coding errors resulted in loss of sync which marked the link down. The loss of sync mechanism is an architectural requirement for 8G Fibre Channel. The RX datapath was only enabled when sync was acquired.

In 16G Fibre Channel, loss of sync is due to 64b transmission word synchronization and is implemented according to 16G requirements (BER state machine, block alignment). Transmission word synchronization is not affected by coding violations – which is different from 8G FC where loss of sync is dependent on 8b10b coding errors.

On loss of sync, PCS data is marked invalid and not written into the FMAC elastic FIFO. A frame which experiences loss of sync while being written into the EFIFO will be corrupted and fail CRC checks. If the loss of block sync occurs on frame delimiters, the frame will generate a sequence error. Lastly, if the loss of block sync occurs in between frames on fill words, the missing IDLEs would have been dropped by the frame processor anyway.

3.0 FMAC ELASTIC FIFO UNIT

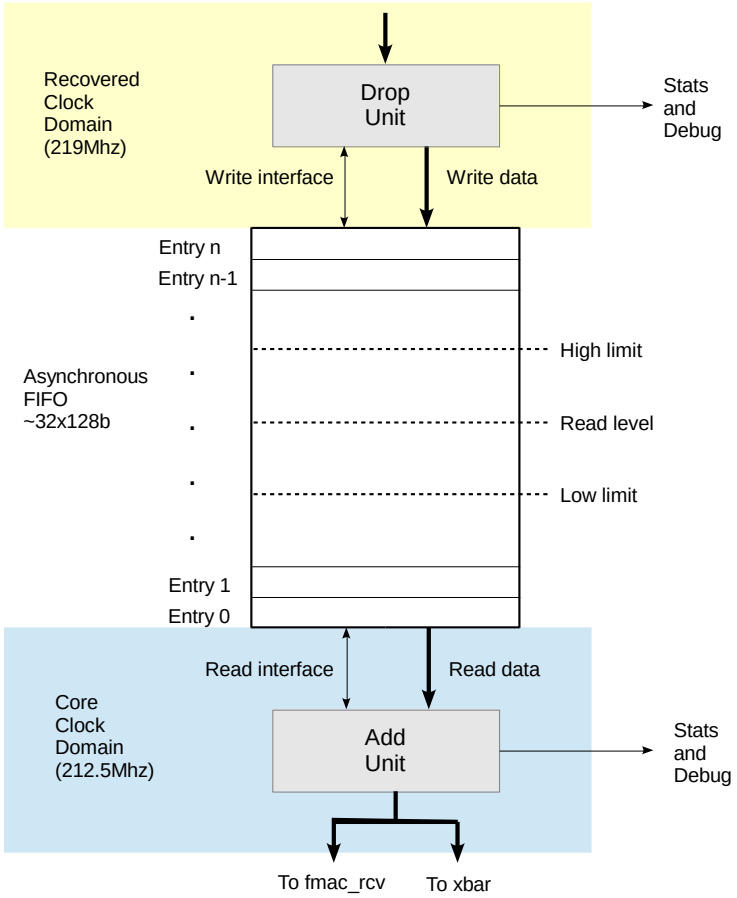
3.1 Rationale

High speed serial interfaces recover clocks from incoming data streams. Because of variations in PLLs and reference clock generators, the recovered clock will not be exactly the same frequency as the local clock. Fibre channel specifies a maximum 100PPM (1 in 10,000) frequency difference between recovered and local clocks.

The FMAC elastic FIFO is responsible for transferring receive data clocked by the recovered clock into the local clock domain. The goal of the elastic FIFO is to transfer data with minimal impact to the data stream.

3.2 Implementation

A block diagram of the FMAC Elastic FIFO is shown in the below diagram.



The elastic FIFO is built as a 32 entry asynchronous FIFO with independent write and read clocks. The write interface is clocked by the recovered clock domain (219Mhz) while the read interface is clocked by the core clock domain (212.5Mhz). Although these clocks are at different frequencies, the data rate on both read and write interfaces are architecturally matched at 14.025G bits per second – except for 100PPM difference in the clocks.

The elastic FIFO initially starts at empty and reads are suppressed until the FIFO has reached its read level (nominally set at the mid-way point). This initial buffering allows sufficient entries in the FIFO to enable add/drop mechanisms to prevent FIFO overflow/underflow. Once the read-level has been met – the read interface is enabled (and stays enabled). The read interface continually reads and monitors the number of FIFO entries. If the FIFO occupancy drops below the low level, the read interface identifies an "idle" period between frames, suppresses a FIFO read, and auto-inserts an idle. The read interface qualifies read data with a valid – in case the elastic FIFO becomes empty.

Conversely, the write interface also monitors the FIFO occupancy, and if the number of FIFO entries exceeds the high threshold, the write interface identifies an "idle" period, and drops an incoming idle. There is no bubbling of the write interface in this case.

3.3 Clock Correction Transmission Words

The following transmission words are considered clock correction words which may be inserted/deleted for purposes of elastic FIFO management:

- IDLE transmission word
- Brocade ARBFF transmission word
- NOS primitive
- OLS primitive
- LR primitive
- LRR primitive

The add unit monitors the current transmission word state – and auto-inserts the correct transmission word to match the current state (*e.g.* – if the receive stream consist of NOS primitives, it auto-inserts a NOS).

Counts of dropped and added transmission words are tracked in fmac_regs.

3.4 Error Cases

The drop unit will always drop a clock correction word if possible. 16G Fibre channel specifies a minimum of four 32b IDLEs between frames, which guarantees at least one 64b IDLE/IDLE transmission word - so it should always be possible to drop one 64b fill word. In case there is a

single fill word, the drop unit will drop the single fill word leaving no fill words between frames. In error cases when there are no fill words or no 64b IDLE/IDLE transmission words, the drop unit will not drop words corresponding with actual fibre channel frames. The FIFO is generated with underflow and overflow protection. Reads and writes which underflow or overflow the FIFO are ignored to prevent FIFO pointer corruption. However, the contents of the frame may become corrupted in these cases.

The elastic FIFO is designed to auto-insert on back to back clock cycles - in order to handle cases when the PCS does not provide valid data for an extended period of time.

It is possible for the PCS to not provide data for extended periods of time (in excess of a few clocks). This can occur on loss of block sync or other error cases. When this happens, elastic FIFO writes do not occur, and the elastic FIFO can become empty. Read logic checks elastic FIFO empty state, and qualifies data with a valid.

3.5 Sizing

The maximum fibre channel frame is ~2KB or 32 entries of the elastic FIFO. Combined with a data rate difference of $1/10{,}000^{th}$, there is more than enough time to find clock correction words to add/drop between frames.

3.6 Startup

On startup and during reset or reset sequencing, the fibre channel SERDES and PCS needs to guarantee that the PCS interface marks all data as invalid and not synced. This prevents bad data from being sent through the FMAC.

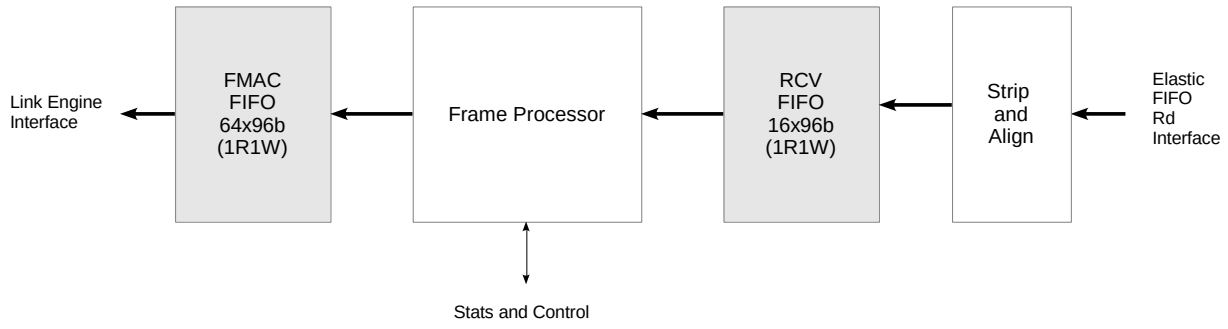
4.0 FMAC RCV UNIT

4.1 Overview

4.1.1 Operations

The FMAC receive unit is responsible for processing and stripping fibre channel data streams. It performs the following actions:

1. Identifies and removes IDLEs.
2. Identifies and removes SOF/EOF frame delimiters.
3. Identifies and removes NOS/OLS/LR/LRR primitives.
4. Identifies and removes 4B CRC fields from a fibre channel frame.
5. Performs data integrity CRC checks.
6. Identifies invalid or corrupted frames. Terminates invalid frames with an EOF it doesn't exist. Marks all invalid frames with an error code.
7. Performs minimum interpacket fill word checks.
8. Interfaces with the Link Engine through a FIFO interface.
9. Supports endianess swap to the Link Engine.
10. Tracks and maintains interval stats.
11. Extracts stats and debug information.



The FMAC receive unit operates in the core clock domain. All FIFOs are constructed with overflow/underflow protection and will ignore reads/writes to prevent FIFO corruption. The FMAC receive unit operates in cut-through mode to minimize latency and storage requirements.

4.1.2 Frame Format

The contents of a fibre channel frame consist of a SOF, an optional extended header, frame header, frame data, a CRC checksum, and an EOF.

Bits Word	31 .. 24	23 .. 16	15 .. 08	07 .. 00
SOF	K28.5 A0	Dxx.x A1	Dxx.x A2	Dxx.x A3
0	R_CTL B0	D_ID B1	D_ID B2	D_ID B3
1	CS_CTL/ Priority B4	S_ID B5	S_ID B6	S_ID B7
2	TYPE B8	F_CTL B9	F_CTL B10	F_CTL B11
3	SEQ_ID B12	DF_CTL B13	SEQ_CNT B14	SEQ_CNT B15
4	OX_ID B16	OX_ID B17	RX_ID B18	RX_ID B19
5	Parameter B20	Parameter B21	Parameter B22	Parameter B23
6	Data_Field B24	Data_Field B25	Data_Field B26	Data_Field B27
7	Data_Field B28	Data_Field B29	Data_Field B30	Data_Field B31
n-1	CRC B32	CRC B33	CRC B34	CRC B35
EOF	K28.5 C0	Dxx.x C1	Dxx.x C2	Dxx.x C3

4.2 Strip and Align Unit

The strip unit receives data from the elastic FIFO. It decodes the incoming data and removes:

- NOS, OLS, LR, LRR primitives
- SOF, EOF delimiters
- IDLES

It also performs minimum interpacket fill word checks. It counts the number of IDLEs in between frames and checks that it meets minimum programmable limits.

Fibre channel is 4B (32b) aligned, however the width of the receive datapath is 64b. The strip unit also aligns the fibre channel frame so that the SOF is located in the first word.

4.3 Frame Processor RCV FIFO

The frame processor receive FIFO decouples the non-stalling elastic FIFO read interface into the frame processor. Since the frame processor needs to insert EOFs on invalid frame conditions, it needs to stall its internal pipeline which is not possible with the elastic FIFO.

The front-end write interface into the receive FIFO strips and removes IDLES/NOS/OLS/LRR/LRRs. This creates occasional bubbles on the write interface. The read interface is controlled by the frame processor which checks for FIFO empty before performing reads. Generally, the receive FIFO should operate at near empty since the write interface should bubble much more often than the read interface stalls due to error conditions. Only under invalid conditions (frames separated by no fill words) would the receive FIFO overflow.

4.4 Frame Processor

4.4.1 CRC

The Cyclic Redundancy Check (CRC) is a 4B field that immediately precedes the EOF delimiter. It covers the fibre channel frame from SOF to EOF (not including the delimiters) and is used to verify data integrity. Refer to the FDDI-MAC FCS and FC-FS-3 Fibre Channel Framing and Signaling specifications for more information.

The fibre channel CRC implements the following polynomial:

$$G(x) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

It is pipelined to make timing.

4.4.2 Length Checks

The receive unit performs frame minimum and maximum length checks. For purposes of length checks, the frame length consists of the frame header and frame data. It does not consist of the SOF/EOF delimiters and 4B CRC.

4.4.3 Sequence Errors

The frame processor checks for improper sequences of delimiters:

- Missing EOF (e.g. SOF followed by a SOF without a preceding EOF)
- Missing SOF (e.g. EOF followed by an EOF without a preceding SOF).

Note that since the decoder converts code violations to IDLEs (per fibre channel requirements), code violations may manifest themselves as sequence errors.

4.5 Extended Header and ISL Support

The FMAC receive unit does not currently decode extended and optional headers. It also does not support ISL interfaces and virtual fabrics/channels. Both features can and will be added in future versions of the FMAC.

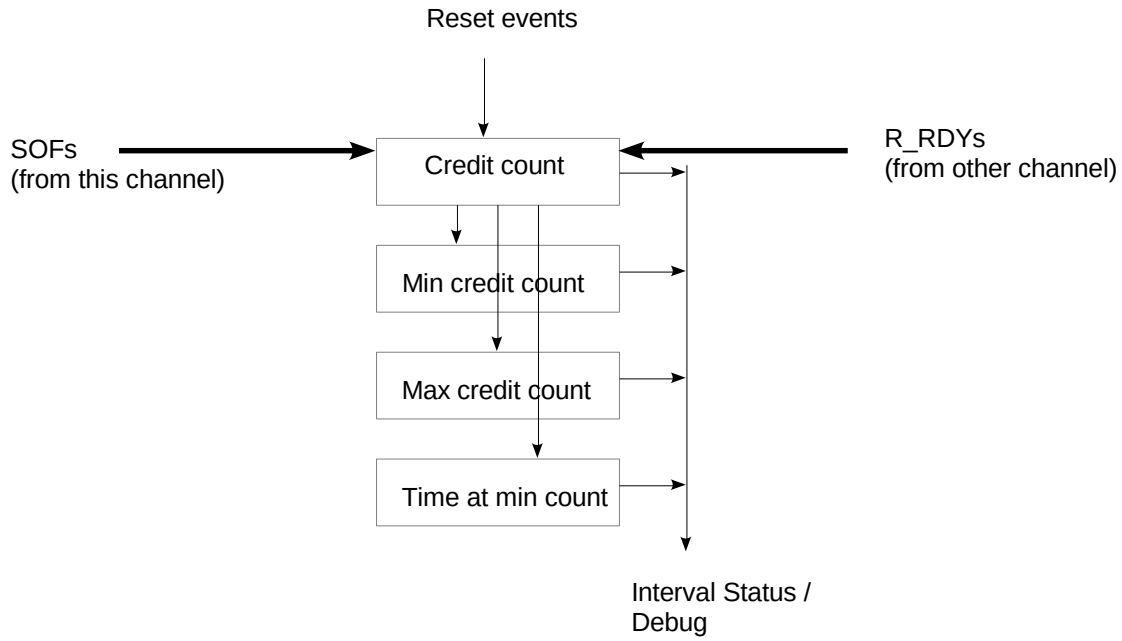
4.6 FMAC FIFO

The FMAC FIFO buffers processed frames between the FMAC and link engine. The FMAC FIFO is organized as 64 entries x 96b. It is written continuously by the frame processor based on frame arrival rates. The write interface is non-stalling. The read interface pushes data into the link engine. The read data is qualified with a valid. The link engine is not able to stall the read interface. In Dominica, a stall interface existed but was not used and has been removed for simplicity.

The FMAC FIFO is implemented with overflow/underflow protection. Read and write commands are ignored to prevent FIFO corruption.

5.0 CREDIT STATS

The FMAC receive unit maintains link level buffer to buffer credit counts. The credit information is synchronized with intervals events and sent to the link engine as part of the interval stats interface.



The credit count is initialized to a programmable starting value. The default starting value is the half-way point. Since our equipment can be inserted into live traffic, we do not know the current credit balance. The credit count can increase or decrease from the half-way point depending on prior history.

Certain events (reset, link_up, new counter initialization value) reset the counter back to the starting value.

Each link contains two channels to monitor transmit and receive streams. The credit stat module in each FMAC is paired with the other credit stat module in the link. SOFs from this channel decrement the counter, and R_RDYs from the other channel decrement the counter. The counter does not overflow or underflow and saturates or floors at either its maximum or minimum value.

Interval and debug stats are generated from the credit counter. Please check the FMAC register documentation for more details.

The credit stats module does not currently support virtual channels.

6.0 FMAC REGISTERS

FMAC registers are documented in the SVN repository.

7.0 CONCLUSION

7.1 Issues

- Brocade ARBFF encoding?
  - Not needed (for now).
- Endianess on interfaces
  - Resolved : little endian internal in FMAC, little endian PCS interface, big endian link engine interface