**COMPHY\_112G\_X4**

**Firmware Eye-Opening Monitor (EOM) Architecture**

(12FFC)

Design R1.0

Firmware Version 0.10.10.85

Doc V0.7

**Marvell Confidential**

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| V0.5 | 05/31/2019 | Xunzhi Wang | Initial version |
| V0.6 | 06/11/2019 | Xunzhi Wang | Update RX\_EOM\_TOP\_CONT\_START\_LANE sequence; ESM\_PHASE and other interfaces |
| V0.7 | 07/18/2019 | Xunzhi Wang | Use set\_dfe\_en; move 2048 when ui alignment; add SW version matching |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Contents

[Revision History 2](#_Toc11745476)

[Contents 3](#_Toc11745477)

[1 Overview 4](#_Toc11745478)

[2 Top State Machine Diagram 4](#_Toc11745479)

[3 Analog Interface (TODO) 5](#_Toc11745480)

[4 Digital Interface (TODO) 5](#_Toc11745481)

[5 User Interface 6](#_Toc11745482)

[5.1 Recommended usage 7](#_Toc11745483)

[5.2 Length of UI 8](#_Toc11745484)

[5.3 SW version matching 8](#_Toc11745485)

[6 Functional Blocks 9](#_Toc11745486)

[6.1 esm\_preparation 9](#_Toc11745487)

[6.1.1 rx\_eom\_cal (expected hardware PI alignment) 10](#_Toc11745488)

[6.1.2 move\_eom\_phase(parameter: delta\_phase\_to\_move) 11](#_Toc11745489)

[6.1.3 check\_eye 11](#_Toc11745490)

[6.1.4 CDR eye check 12](#_Toc11745491)

[6.1.5 eom\_ui\_align (FW UI-PI alignment) 14](#_Toc11745492)

[6.2 esm\_measue 15](#_Toc11745493)

[6.3 esm\_exit 16](#_Toc11745494)

[7 Issues 16](#_Toc11745495)

# Overview

**If you want to know how to interact with FW EOM, go to** [**User Interface**](#_User_Interface)**.**

To support EOM, FW first needs to align EOM clock with data clock, to find the correct phase relationship for data comparision.

Generally, there are two levels of alignment:

1. phase interval (PI, minimum analog phase change unit) alignment: make sure EOM clock’s rising edge aligns with data clock’s rising edge.
2. unit interval (UI, serial clock period) alignment: due to multiple levels of dividers in digital logic, there is delay of a number of UIs between data path and EOM path. The maximum delay is one parallel data clock. Thus FW needs to remove this delay, and let EOM value match data value (illustrated in figure as orange edge).

After alignment, users can change phase and voltage of sampler, and issue commands to gather information for eye drawing.



# Top State Machine Diagram

Firmware has three EOM states:

Esm\_preparation would align EOM clock and data clock to locate the center of eye;

Esm\_measure is for user to check eye status given different phase and voltage settings;

Esm\_exit is to restore EOM settings to default.

By polling user interface, states change upon request.



# Analog Interface (TODO)

|  |  |  |  |
| --- | --- | --- | --- |
| **Port Name** | **Dir** | **Default** | **Description** |
|  |  |  |  |

# Digital Interface (TODO)

|  |  |  |  |
| --- | --- | --- | --- |
| **Port Name** | **FW** | **Default** | **Description** |
| rx\_eom\_pi\_cal\_result\_rd\_lane[10:4] | R | 0 | EOM Dpher: for reading |
| rx\_eom\_pi\_cal\_result\_rd\_lane[3:0] | R | 0 | EOM Dpher: for reading |
| rx\_eom\_pi\_cal\_result\_ext\_lane[10:5] | W | 6'h20 | EOM Dpher: for writing |
| rx\_eom\_pi\_cal\_result\_ext\_lane[4:0] | W | 0 | EOM Dpher: for writing |
| dfe\_adapt\_splr\_en\_lane[7:0] | RW | 8'hff | DFE Adaptation Sampler Selection  1: Enable This Sampler  0: Disable This Sampler  [0]: DP1 sampler  [1]: DP2 sampler  [2]: DP3 sampler  [3]: DP4 sampler  [4]: SP1 sampler  [5]: SP2 sampler  [6]: SP3 sampler  [7]: SP4 sampler |
| eom\_clk\_en\_lane | RW | 0 | Enable EOM Clock |
| eom\_en\_d\_lane | RW | 0 | Enable EOM Clock For Data Sampler |
| eom\_en\_e\_lane | RW | 0 | Enable EOM Clock For Edge Sampler |
| eom\_en\_s\_lane | RW | 0 | Enable EOM Clock For Slicer Sampler |
|  |  |  |  |
| EOM\_VLD\_CNT\_TOP\_P\_LANE[31:0] | R | 0 | Valid Count For Top Positive Eye |
| EOM\_VLD\_CNT\_MID\_P\_LANE[31:0] | R | 0 | Valid Count For Mid Positive Eye |
| EOM\_VLD\_CNT\_BOT\_P\_LANE[31:0] | R | 0 | Valid Count For Bot Positive Eye |
| EOM\_VLD\_CNT\_TOP\_N\_LANE[31:0] | R | 0 | Valid Count For Top Negative Eye |
| EOM\_VLD\_CNT\_MID\_N\_LANE[31:0] | R | 0 | Valid Count For Mid Negative Eye |
| EOM\_VLD\_CNT\_BOT\_N\_LANE[31:0] | R | 0 | Valid Count For Bot Negative Eye |
| EOM\_ERR\_CNT\_TOP\_P\_LANE[31:0] | R | 0 | Error Count For Top Positive Eye |
| EOM\_ERR\_CNT\_MID\_P\_LANE[31:0] | R | 0 | Error Count For Mid Positive Eye |
| EOM\_ERR\_CNT\_BOT\_P\_LANE[31:0] | R | 0 | Error Count For Bot Positive Eye |
| EOM\_ERR\_CNT\_TOP\_N\_LANE[31:0] | R | 0 | Error Count For Top Negative Eye |
| EOM\_ERR\_CNT\_MID\_N\_LANE[31:0] | R | 0 | Error Count For Mid Negative Eye |
| EOM\_ERR\_CNT\_BOT\_N\_LANE[31:0] | R | 0 | Error Count For Bot Negative Eye |
|  |  |  |  |

# User Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Port Name** | **FW** | **User** | **Default** | **Description** |
| ESM\_EN\_LANE | R | RW | 1'h0 | 1-Enable EOM\_EN |
| ESM\_PATH\_SEL\_LANE | R | RW | 1'b0 | 1-Select EOM Slice Path, 0-Select EOM Data Path |
| adapt\_odd\_en\_lane | R | RW | 1 | Adapt Odd Enable: Sampler 1,3 |
| adapt\_even\_en\_lane | R | RW | 1 | Adapt Even Enable: Sampler 2,4 |
| ESM\_DFE\_ADAPT\_SPLR\_EN\_LANE[3:0] | R | RW | 4'h0 | DFE Adapt Sampler Enable When Drawing EOM. Bit x is for sampler x (x=0~3). Overwrite adapt\_odd\_en\_lane/ adapt\_even\_en\_lane config if not 0. Note: would only enable target sampler whose cds\_adapt\_splr\_dis\_lane[7:0] bit is 0. User may check dfe\_adapt\_splr\_en\_lane[7:0] for final setting to digital. |
| ESM\_PHASE\_LANE[10:0] | R | RW | 10'h0 | Eye Shape Monitor Phase Value (-1024 ~ +1023) |
| ESM\_VOLTAGE\_LANE[7:0] | RW | RW | 8'h0 | Eye Shape Monitor Voltage Value(0~127). FW set it to 0 at esm\_preparation state. Let user control it at other states. |
| esm\_lpnum\_lane[15:0] | RW | RW | 16'h0 | DFE Adapt Loop Number. [15:8] for adapt loop number, [7:0] for polarity loop number. After ESM\_EN\_LANE=1, if this reg is still 0, FW set it to 0x2008. |
| eye\_chk\_thresh\_err\_lane[15:0] | RW | RW | 16'h3ff | Eye Check Threshold For Error Count. Recommend to change along with esm\_lpnum\_lane.  For example:  0x3ff for esm\_lpnum\_lane 0xa05;  0x1000 for esm\_lpnum\_lane 0x2008 |
| EOM\_READY\_LANE | RW | RW | 1'b0 | Eye Monitor Drawing Ready: esm\_preparation done. |
| EOM\_DFE\_CALL\_LANE | RW | RW | 1'b0 | DFE Call Enable For Eye Shape Monitor. Set by user, clear by FW. |
| eom\_phase\_ui\_align\_failed\_lane | RW | R | 1'h0 | Debug: UI alignment failed during esm\_preparation. User may check it after EOM\_READY\_LANE=1. |
| cds\_adapt\_splr\_dis\_lane[7:0] | RW | R | 8'h0 | DFE Sampler Adapt Disable; Final guard to dfe\_adapt\_splr\_en\_lane[7:0] setting.  1: CDS disable this Sampler  0: CDS enable This Sampler  [0]: DP1 sampler  [1]: DP2 sampler  [2]: DP3 sampler  [3]: DP4 sampler  [4]: SP1 sampler  [5]: SP2 sampler  [6]: SP3 sampler  [7]: SP4 sampler |
| EOM\_\*\_CNT\_\*\_\*\_LANE[31:0] | R | R | 0 | See [Digital Interface](#_Digital_Interface) |

## Recommended usage

1. Start traffic: recommend PRBS7 and above for PAM2; PRBS31 for PAM4
2. Set ESM\_PATH\_SEL\_LANE=1;

Set adapt\_odd\_en\_lane and adapt\_even\_en\_lane; or ESM\_DFE\_ADAPT\_SPLR\_EN\_LANE[3:0];

Set esm\_lpnum\_lane[15:0] (optional)

1. Set ESM\_EN\_LANE=1
2. Wait for EOM\_READY\_LANE=1; optionally check eom\_phase\_ui\_align\_failed\_lane
3. Start to check eye using the loop:

Set ESM\_PHASE\_LANE

Set ESM\_VOLTAGE\_LANE

Set EOM\_DFE\_CALL\_LANE=1

Wait for EOM\_DFE\_CALL\_LANE=0

Check eom counters

1. Set ESM\_EN\_LANE=0 to finish drawing

## Length of UI

|  |  |
| --- | --- |
| Data rate | Number of PI in 1UI |
| 80G~112G | 64 |
| 40G~56G | 128 |
| 8G~14G | 256 |
| 4G~7G | 512 |
| 2G~3.5G | 1024 |
| 1G~1.5G | 2048 |

## SW version matching

Raptor2 release <R1.20.35.00 does NOT support >=0.10.10.66 FW EOM.

Raptor2 release >=R1.20.35.00 does NOT support <0.10.10.66 FW EOM.

# Functional Blocks

## esm\_preparation



### rx\_eom\_cal (expected hardware PI alignment)

Use unified calibration for rx\_eom\_align90, rx\_eom\_dcc, and rx\_eom\_pi calibration based on settings. This calibration is also done in cal\_top() when power-on. (Sequence under debugging!)

1. lnx\_RX\_EOM\_CAL\_DONE\_LANE = 0; lnx\_RX\_EOM\_CAL\_PASS\_LANE = 0
2. rx\_eom\_cal\_init:

PU\_EOM\_ALIGN90\_DCC\_CMP = 1;

EOM\_CLK\_EN = 1;

EOM\_RESET\_INTP\_EXT = 0;

RXDCC\_HG\_EOMCLK = 1;

RXDCC\_EN\_EOMCLK = 1;

PU\_EOM\_ALIGN90\_DCC\_CMP = 1;

if (reg\_RX\_HALFRATE\_EN\_LANE == 0)

{

RX\_EOM\_DCC\_CAL\_BYPASS\_EN = 1;//0;//1;

reg\_RX\_EOM\_DCC\_CAL\_SINGLE\_EN\_LANE = 1;

reg\_RX\_EOM\_DCC\_CAL\_CONT\_EN\_LANE = 1;

reg\_RX\_EOM\_DCC\_CAL\_SINGLE\_MODE\_STEPSIZE\_LANE\_2\_0 = 0;

reg\_RX\_EOM\_DCC\_CAL\_CONT\_MODE\_STEPSIZE\_LANE\_2\_0 = 1;

reg\_RX\_EOM\_PI\_CAL\_BYPASS\_EN\_LANE = 0;

reg\_RX\_EOM\_PI\_CAL\_SINGLE\_EN\_LANE = 1;

//RX\_EOM\_PI\_CAL\_CONT\_EN=0

reg\_RX\_EOM\_PI\_CAL\_SINGLE\_MODE\_STEPSIZE\_LANE\_2\_0 = 4;

//RX\_EOM\_PI\_CAL\_CONT\_MODE\_STEPSIZE[2:0] =1

reg\_RX\_EOM\_ALIGN90\_CAL\_BYPASS\_EN\_LANE = 1;//0;//1;

reg\_RX\_EOM\_ALIGN90\_CAL\_SINGLE\_EN\_LANE = 1;

reg\_RX\_EOM\_ALIGN90\_CAL\_CONT\_EN\_LANE = 1;

reg\_RX\_EOM\_ALIGN90\_CAL\_SINGLE\_MODE\_STEPSIZE\_LANE\_2\_0 = 0;

reg\_RX\_EOM\_ALIGN90\_CAL\_CONT\_MODE\_STEPSIZE\_LANE\_2\_0 = 1;

}

else if (reg\_RX\_HALFRATE\_EN\_LANE == 1)

{

reg\_RX\_EOM\_PI\_CAL\_SETTING\_LANE\_6\_0 = (0X294 & 0x7f);

reg\_RX\_EOM\_PI\_CAL\_SETTING\_LANE\_12\_7 = (0X294 & (0x3f << 7));

reg\_RX\_EOM\_ALIGN90\_CAL\_BYPASS\_EN\_LANE = 1;

RX\_EOM\_DCC\_CAL\_BYPASS\_EN = 0;//1;

reg\_RX\_EOM\_DCC\_CAL\_SINGLE\_EN\_LANE = 1;

reg\_RX\_EOM\_DCC\_CAL\_CONT\_EN\_LANE = 0;

reg\_RX\_EOM\_DCC\_CAL\_SINGLE\_MODE\_STEPSIZE\_LANE\_2\_0 = 0;

reg\_RX\_EOM\_DCC\_CAL\_CONT\_MODE\_STEPSIZE\_LANE\_2\_0 = 1;

reg\_RX\_EOM\_PI\_CAL\_BYPASS\_EN\_LANE = 0;

reg\_RX\_EOM\_PI\_CAL\_SINGLE\_EN\_LANE = 1;

//RX\_EOM\_PI\_CAL\_CONT\_EN=0

reg\_RX\_EOM\_PI\_CAL\_SINGLE\_MODE\_STEPSIZE\_LANE\_2\_0 = 4;

//RX\_EOM\_PI\_CAL\_CONT\_MODE\_STEPSIZE[2:0] =1

reg\_RX\_EOM\_ALIGN90\_CAL\_BYPASS\_EN\_LANE = 0;//1;

reg\_RX\_EOM\_ALIGN90\_CAL\_SINGLE\_EN\_LANE = 1;

reg\_RX\_EOM\_ALIGN90\_CAL\_CONT\_EN\_LANE = 0;

reg\_RX\_EOM\_ALIGN90\_CAL\_SINGLE\_MODE\_STEPSIZE\_LANE\_2\_0 = 0;

reg\_RX\_EOM\_ALIGN90\_CAL\_CONT\_MODE\_STEPSIZE\_LANE\_2\_0 = 1;

}

1. reg\_RX\_EOM\_TOP\_START\_LANE = 1; delay 0.5us
2. Wait for reg\_RX\_EOM\_TOP\_DONE\_LANE=1
3. lnx\_RX\_EOM\_CAL\_PASS\_LANE = 1 if no timeout happened
4. reg\_RX\_EOM\_TOP\_START\_LANE = 0
5. Store results:

lnx\_cal\_rx\_eom\_dcc\_result\_lane = reg\_RX\_EOM\_DCC\_CAL\_RESULT\_RD\_LANE\_6\_0;

lnx\_cal\_rx\_eom\_pi\_result\_lane = reg\_RX\_EOM\_PI\_CAL\_RESULT\_RD\_LANE\_10\_4;

lnx\_cal\_rx\_eom\_align90\_result\_msb\_lane = reg\_RX\_EOM\_ALIGN90\_CAL\_RESULT\_MSB\_RD\_LANE\_2\_0;

lnx\_cal\_rx\_eom\_align90\_result\_lsb\_lane = reg\_RX\_EOM\_ALIGN90\_CAL\_RESULT\_LSB\_RD\_LANE\_6\_0;

1. lnx\_RX\_EOM\_CAL\_DONE\_LANE = 1
2. PU\_EOM\_ALIGN90\_DCC\_CMP = 0

### move\_eom\_phase(parameter: delta\_phase\_to\_move)

Called in eom\_ui\_align and esm\_measure.

1. reg\_RX\_EOM\_TOP\_CONT\_START\_LANE = 0
2. reg\_RX\_EOM\_COMN\_EXT\_EN\_LANE = 1
3. Move phase tag\_CDR\_OS\_PH\_JMP\_STEP (1) at a time until delta\_phase\_to\_move are moved. Loop as follows:

c.1 Read current phase reg\_RX\_EOM\_PI\_CAL\_RESULT\_RD\_LANE

c.2 Add or substract tag\_CDR\_OS\_PH\_JMP\_STEP

c.3 Write to reg\_RX\_EOM\_PI\_CAL\_RESULT\_EXT\_LANE

c.4 Toggle reg\_RX\_EOM\_AUTO\_ZERO\_CLK\_EXT\_LANE

1. reg\_RX\_EOM\_COMN\_EXT\_EN\_LANE = 0

### check\_eye

Called in CDR eye check.

1. Find the max value of vld counters and err counters
2. Return 1 (pass) if vld\_max>=eye\_chk\_thresh\_vld\_lane[7:0] (default 255) and err\_max<=eye\_chk\_thresh\_err\_lane[15:0] (default 1023 for loop number 0xa05). Specially for esm\_preparation state, the latter criterion is err\_max<=4096 for loop number 0x2008.

### CDR eye check

Called in eom\_ui\_align and training functions: cdr\_dfe\_scheme(cds\_table[CDS\_EN\_EYECHECK])

This is part of DSP driver cdr\_dfe\_scheme.

Only focus on the ESM\_EN=1 path,



### eom\_ui\_align (FW UI-PI alignment)

As currently rx\_eom\_cal cannot do PI alignment as expected, we use the function to handle all alignments.

FW UI-PI alignment process can be illustrated by the EOM clock moving diagram below.



Notes:

1. Move x\*[y,z] PI: move\_eom\_phase(x) at a time and check\_eye(); loop up to z times or until an eye check pass/failure happened.
2. dec\_num = 2^(RXSPEED\_DIV\_LANE&0x3)\*((RXSPEED\_DIV\_LANE>>2&0x1)\*2), this is related to [Length of UI](#_Length_of_UI) to move more aggressive on lower speeds.
3. Width = (high\_bound-low\_bound)/2

## esm\_measue



## esm\_exit



# Issues

1. At certain phases, the number of eye check errors varies over time. It causes wrong FW UI alignment, and leaves eye uncentered.
2. Is the HW voltage range (6 bits) enough for PAM2?

In PAM2, reg\_DFE\_F0\_RES\_DOUBLE\_LANE is 1.

If the height is still out of range, user can set dfe\_res\_f0\_lane[1:0] to 3, and redo txtrain.

1. rx\_eom\_cal cannot work as expected.
2. The recommended loop numbers for user to choose.

1G issue: for 0x2008 setting, the valid bits are sometimes 140000, sometimes 280000.