

RW-WLAN-nX Embedded Logic Analyzer

User Manual

RW-WLAN-nX-EMB-LA-UM/1.04

Version 1.04

2019-01-04

Revision History

Version	Date	Revision Description	Author
1.00	2016-04-19	Initial Revision	Jerome Vanthournout
1.01	2016-06-02	Corrected typo in capture data dump procedure Updated with la_depth field in ID_HIGH register	Jerome Vanthournout
1.02	2016-06-16	Possibility to select the size of the LA memory using parameter Interrupt generation when triggered Added Sw trigger Increase external trigger input up to 4 (each external trigger input can be enabled or disabled)	Jerome Vanthournout
1.03	2016-11-23	Corrected typo.	Jerome Vanthournout
1.04	2017-07-06	Updated register description format	Jerome Vanthournout

Table of Contents

Revision History	2
Table of Contents	3
List of Figures	4
1 Overview	5
2 HW architecture	6
2.1 Block Diagram	6
2.2 Data Capture	6
2.3 Trigger	6
2.3.1 External Trigger	7
2.3.2 Software Trigger	7
2.3.3 Internal trigger	7
2.3.4 Triggered output	7
2.3.5 Interrupt Generation	7
2.4 Memory Organization	7
2.5 Read Captured Data	8
3 Software Control	9
3.1 Register description	9
3.1.1 Register Map	9
3.1.2 Register List	9
3.1.2.1 Register ID_LOW	9
3.1.2.2 Register ID_HIGH	10
3.1.2.3 Register VERSION	10
3.1.2.4 Register CNTRL	10
3.1.2.5 Register STATUS	11
3.1.2.6 Register SAMPLING_MASK_LOW	12
3.1.2.7 Register SAMPLING_MASK_MED	12
3.1.2.8 Register SAMPLING_MASK_HIGH	13
3.1.2.9 Register TRIGGER_MASK_LOW	13
3.1.2.10 Register TRIGGER_MASK_MED	14
3.1.2.11 Register TRIGGER_MASK_HIGH	14
3.1.2.12 Register TRIGGER_VALUE_LOW	14
3.1.2.13 Register TRIGGER_VALUE_MED	15
3.1.2.14 Register TRIGGER_VALUE_HIGH	15
3.1.2.15 Register TRIGGER_POINT	16
3.1.2.16 Register FIRSTSAMPLE	16
3.2 Captured Data Dump	17
4 Integration in RW NX IP	18
4.1 Clocking scheme	18
4.2 Debug Port	18

List of Figures

Figure 1 : Embedded Logic Analyzer Block Diagram	6
--	---

1 Overview

This document describes the Embedded Logic Analyzer, its software interface and some recommendations for its integration inside the RW WLAN NX IP.

2 HW architecture

2.1 Block Diagram

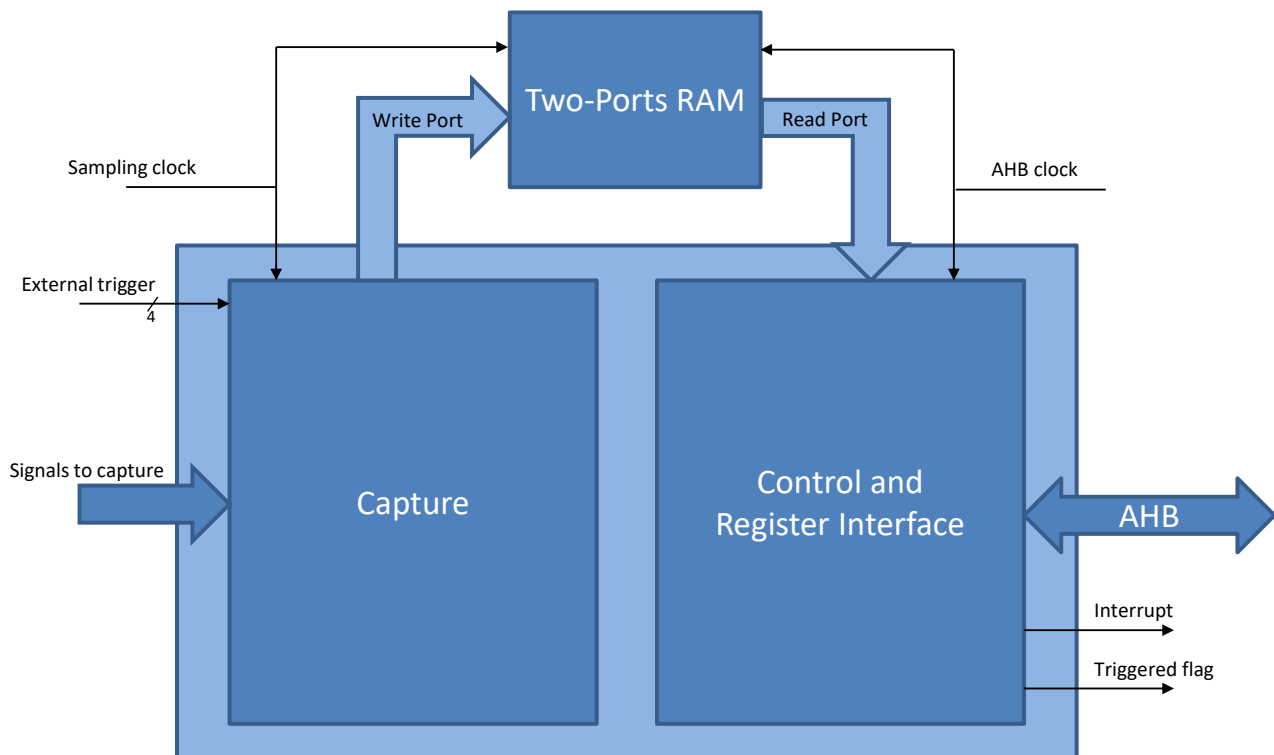


Figure 1 : Embedded Logic Analyzer Block Diagram

2.2 Data Capture

As the signals connected on the Embedded Logic Analyzer may come from different clock domain than the `la_clk`, the sample are resynchronized on `la_clk` clock using a two rising edges FF structure. As a consequence, the `la_clk` frequency shall be higher than all the other signals.

At each `la_clk` clock cycle, the `SAMPLING_MASK` is applied on the captured data and then compared with the data capture at the previous `la_clk` clock cycle. If the captured data changed, the RAM write address is incremented, the capture data are stored in the memory and the delay counter is reset. Otherwise, the RAM write address remains the same and the delay count is incremented and written in the memory. This method named "transitional storage" allows to optimize the memory.

When the LA is enabled, the capture starts at the address 0 and stops either if the LA is stopped (writing 1 in stop field of `CNTRL` register) or if a trigger is generated.

2.3 Trigger

The Embedded Logic Analyzer provides three types of triggers:

2.3.1 External Trigger

A dedicated input, named `ext_trigger[3:0]`, can be used to trig the Embedded Logic Analyzer. When asserted by an external module, it is first resynchronized on the `la_clk` and then stops the capture. Up to four external trigger is supported by the LA and each can be enabled/disable using the register `CNTRL.ext_trigger_en[3:0]`.

2.3.2 Software Trigger

The software can trig the Embedded Logic Analyzer by writing in the `CNTRL.sw_trigger` register. This bit is automatically reset by the HW and behaves a standard trigger.

2.3.3 Internal trigger

The internal trigger is a basic trigger feature which allows to stopping the capture if the captured samples match a given value (`TRIGGER_VALUE_LOW/MED/HIGH` registers). A specific mask can be set for this comparison in order to perform this comparison on a subset of the capture signals. A "1" in the `TRIGGER_MASK_LOW/MED/HIGH` register indicates that the given captured signals will be used for the comparison.

Whatever the trigger source is (internal or external), it is possible to specific the number of sample captured after the trigger before stopping the acquisition. The register `TRIGGER_POINT` is used for this purpose.

2.3.4 Triggered output

When the Embedded Logic Analyzer has been triggered due to one of the previous condition, the output **triggered** is set. It is reset when the Embedded Logic Analyzer is reset or when a new capture starts.

2.3.5 Interrupt Generation

When the Embedded Logic Analyzer has been triggered due to one of the previous condition, an interrupt can be generated to the software (**la_interrupt** output). To enable the interrupt generated, the software shall set the `CNTRL.trigger_int_en` register. Writing in `CNTRL.trigger_int_clear` register clears the interrupt. Writing in `CNTRL.trigger_int_set` register generates the interrupt.

The software can also get the interrupt status by reading `CNTRL.trigger_int_set` register.

2.4 Memory Organization

127-122	121-120	119-96	95-0
Triggers status	<i>Reserved</i>	Delay (24bits)	Stored Samples

The memory width is 128bits in order to store the 96bits samples, the delay between two different samples (on 31bits) and the trigger position.

The size of this memory is 65535 samples by default but it can be configured using the `LA_MEMORY_SIZE` parameters during synthesis.

The Trigger status bit indicates the status of the different trigger.

Trigger status[0] : Internal Trigger (pattern match)

Trigger status[1] : Software Trigger

Trigger status[2] : External Trigger (ext_trigger[0])

Trigger status[3] : External Trigger (ext_trigger[1])

Trigger status[4] : External Trigger (ext_trigger[2])

Trigger status[5] : External Trigger (ext_trigger[3])

To simplify the clocking scheme, the memory used for data storage is a Two Port SRAM. The Write port is clocked with la_clk and the Read port is clocked with ahb_clk.

The Delay gives the number of la_clk clock cycle for which the Stored Sample was stable.

2.5 Read Captured Data

The Embedded LA provides an AHB interface which allows reading the captured sampled in the memory. As the AHB bus is 32bits and the memory is 128bits, four AHB accesses are needed to read a sample using the following addressing.

AHB address	Data
0x100000	Captured Sample 1 [31:0]
0x100004	Captured Sample 1 [63:32]
0x100008	Captured Sample 1 [96:64]
0x10000C	{Triggers status, Delay}
0x100010	Captured Sample 2 [31:0]
0x100014	Captured Sample 2 [63:32]
0x100018	Captured Sample 2 [96:64]
0x10001C	{Triggers status, Delay}

3 Software Control

When accessing the Embedded Logic Analyzer through the AHB interface, the Software can either read the captured data from the SRAM or read/write in the register. The selection between register bank or LA memory is done using the haddr[20] as follow :

haddr[20] = 0 : read/write register access

haddr[20] = 1 : read access to the SRAM.

3.1 Register description

3.1.1 Register Map

Address	Register Name
0x0	ID_LOW
0x4	ID_HIGH
0x8	VERSION
0xC	CNTRL
0x10	STATUS
0x14	SAMPLING_MASK_LOW
0x18	SAMPLING_MASK_MED
0x1C	SAMPLING_MASK_HIGH
0x20	TRIGGER_MASK_LOW
0x24	TRIGGER_MASK_MED
0x28	TRIGGER_MASK_HIGH
0x2C	TRIGGER_VALUE_LOW
0x30	TRIGGER_VALUE_MED
0x34	TRIGGER_VALUE_HIGH
0x38	TRIGGER_POINT
0x3C	FIRSTSAMPLE

Table 3.1 - Register Map

3.1.2 Register List

3.1.2.1 Register ID_LOW

This register contains the LA ID LSB

Address	Access		ID_LOW																																	
	HW	SW	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		
+0'H	W	R	id_low[31]	id_low[30]	id_low[29]	id_low[28]	id_low[27]	id_low[26]	id_low[25]	id_low[24]	id_low[23]	id_low[22]	id_low[21]	id_low[20]	id_low[19]	id_low[18]	id_low[17]	id_low[16]	id_low[15]	id_low[14]	id_low[13]	id_low[12]	id_low[11]	id_low[10]	id_low[9]	id_low[8]	id_low[7]	id_low[6]	id_low[5]	id_low[4]	id_low[3]	id_low[2]	id_low[1]	id_low[0]		
Reset Value			0	1	1	0	0	1	0	1	0	1	1	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	1	0	0	0	1	0		
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U			
HW Access			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W		
SW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Table 3.2 - Register ID_LOW

Name	Type	Size	Description
id_low[31:0]	U	32	ID of Embedded Logic Analyzer IP (0x656E6562)

Table 3.3 - Register ID_LOW fields description

3.1.2.2 Register ID_HIGH

This register contains the LA ID MSB

Address	Access		ID_HIGH																																			
	HW	SW	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				
+4'H	W	R																	la_depth[15]	la_depth[14]	la_depth[13]	la_depth[12]	la_depth[11]	la_depth[10]	la_depth[9]	la_depth[8]	la_depth[7]	la_depth[6]	la_depth[5]	la_depth[4]	la_depth[3]	la_depth[2]	la_depth[1]	la_depth[0]				
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U				
HW Access																			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W				
SW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R			

Table 3.4 - Register ID_HIGH

Name	Type	Size	Description
la_depth[15:0]	U	16	Depth of the LA Memory (depend on LA_MEMORY_SIZE)

Table 3.5 - Register ID_HIGH fields description

3.1.2.3 Register VERSION

This register contains the LA Version

Address	Access		VERSION																																	
	HW	SW	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		
+8'H	W	R	samplingFreq[7]	samplingFreq[6]	samplingFreq[5]	samplingFreq[4]	samplingFreq[3]	samplingFreq[2]	samplingFreq[1]	samplingFreq[0]	la_version[23]	la_version[22]	la_version[21]	la_version[20]	la_version[19]	la_version[18]	la_version[17]	la_version[16]	la_version[15]	la_version[14]	la_version[13]	la_version[12]	la_version[11]	la_version[10]	la_version[9]	la_version[8]	la_version[7]	la_version[6]	la_version[5]	la_version[4]	la_version[3]	la_version[2]	la_version[1]	la_version[0]		
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	
HW Access			W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
SW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.6 - Register VERSION

Name	Type	Size	Description
samplingFreq[7:0]	U	8	Frequency of la_clk input in MHz.
la_version[23:0]	U	24	Version of the Embedded Logic Analyzer IP

Table 3.7 - Register VERSION fields description

3.1.2.4 Register CNTRL

This register controls the LA

Address	Access		CNTRL																															
	HW	SW	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

[illegible]

Table 3.8 - Register CNTRL

Name	Type	Size	Description
trigger_int_en	U	1	Enable the Embedded Logic Analyzer interrupt generation when a trigger occurs.
trigger_int_clear	U	1	Writing '1' clear the Embedded Logic Analyzer interrupt
trigger_int_set	U	1	Writing '1' generate the Embedded Logic Analyzer interrupt Read this field indicates the current status of the interrupt
ext_trigger_en[3:0]	U	4	Enable the external trigger.
sw_trigger	U	1	Writing '1' trig the Embedded Logic Analyzer
reset	U	1	Writing '1' reset the Embedded Logic Analyzer. Writing '1' de-assert the reset of the Embedded Logic Analyzer.
stop	U	1	Writing '1' stop the capture
start	U	1	Writing '1' start the capture

Table 3.9 - Register CNTRL fields description

3.1.2.5 Register STATUS

This register provides status of the LA

[illegible]

Table 3.10 - Register STATUS

Name	Type	Size	Description
writeaddr[15:0]	U	16	Indicate the location of the latest captured sample.
ext_trigger_status[3:0]	U	4	Indicate the status of the external triggers
sw_trigger_status	U	1	Indicate the status of the software trigger

internal_trigger_status	U	1	Indicate the status of the internal trigger
triggered	U	1	Reading '1' indicates that the capture is stopped due to a trigger
started	U	1	Reading '1' indicates that the capture is on-going

Table 3.11 - Register STATUS fields description

3.1.2.6 Register SAMPLING_MASK_LOW

This register controls the input signals sampling

Address	Access		SAMPLING_MASK_LOW																																
	HW	SW	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	
+14'H	R	RW	sampling_mask_low[31]	sampling_mask_low[30]	sampling_mask_low[29]	sampling_mask_low[28]	sampling_mask_low[27]	sampling_mask_low[26]	sampling_mask_low[25]	sampling_mask_low[24]	sampling_mask_low[23]	sampling_mask_low[22]	sampling_mask_low[21]	sampling_mask_low[20]	sampling_mask_low[19]	sampling_mask_low[18]	sampling_mask_low[17]	sampling_mask_low[16]	sampling_mask_low[15]	sampling_mask_low[14]	sampling_mask_low[13]	sampling_mask_low[12]	sampling_mask_low[11]	sampling_mask_low[10]	sampling_mask_low[9]	sampling_mask_low[8]	sampling_mask_low[7]	sampling_mask_low[6]	sampling_mask_low[5]	sampling_mask_low[4]	sampling_mask_low[3]	sampling_mask_low[2]	sampling_mask_low[1]	sampling_mask_low[0]	
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 3.12 - Register SAMPLING_MASK_LOW

Name	Type	Size	Description
sampling_mask_low[31:0]	U	32	Define the active signals inside the input signal [31:0]

Table 3.13 - Register SAMPLING_MASK_LOW fields description

3.1.2.7 Register SAMPLING_MASK_MED

This register controls the input signals sampling

Address	Access		SAMPLING_MASK_MED																															
	HW	SW	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
+18'H	R	RW	sampling_mask_med[31]	sampling_mask_med[30]	sampling_mask_med[29]	sampling_mask_med[28]	sampling_mask_med[27]	sampling_mask_med[26]	sampling_mask_med[25]	sampling_mask_med[24]	sampling_mask_med[23]	sampling_mask_med[22]	sampling_mask_med[21]	sampling_mask_med[20]	sampling_mask_med[19]	sampling_mask_med[18]	sampling_mask_med[17]	sampling_mask_med[16]	sampling_mask_med[15]	sampling_mask_med[14]	sampling_mask_med[13]	sampling_mask_med[12]	sampling_mask_med[11]	sampling_mask_med[10]	sampling_mask_med[9]	sampling_mask_med[8]	sampling_mask_med[7]	sampling_mask_med[6]	sampling_mask_med[5]	sampling_mask_med[4]	sampling_mask_med[3]	sampling_mask_med[2]	sampling_mask_med[1]	sampling_mask_med[0]
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 3.14 - Register SAMPLING_MASK_MED

Name	Type	Size	Description
sampling_mask_med[31:0]	U	32	Define the active signals inside the input signal [63:32]

Table 3.15 - Register SAMPLING_MASK_MED fields description

3.1.2.8 Register SAMPLING_MASK_HIGH

This register controls the input signals sampling

Address	Access		SAMPLING_MASK_HIGH																																			
	HW	SW	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				
+1C'H	R	RW	sampling_mask_high[31]	sampling_mask_high[30]	sampling_mask_high[29]	sampling_mask_high[28]	sampling_mask_high[27]	sampling_mask_high[26]	sampling_mask_high[25]	sampling_mask_high[24]	sampling_mask_high[23]	sampling_mask_high[22]	sampling_mask_high[21]	sampling_mask_high[20]	sampling_mask_high[19]	sampling_mask_high[18]	sampling_mask_high[17]	sampling_mask_high[16]	sampling_mask_high[15]	sampling_mask_high[14]	sampling_mask_high[13]	sampling_mask_high[12]	sampling_mask_high[11]	sampling_mask_high[10]	sampling_mask_high[9]	sampling_mask_high[8]	sampling_mask_high[7]	sampling_mask_high[6]	sampling_mask_high[5]	sampling_mask_high[4]	sampling_mask_high[3]	sampling_mask_high[2]	sampling_mask_high[1]	sampling_mask_high[0]				
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U			
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		

Table 3.16 - Register SAMPLING_MASK_HIGH

Name	Type	Size	Description
sampling_mask_high[31:0]	U	32	Define the active signals inside the input signal [95:64]

Table 3.17 - Register SAMPLING_MASK_HIGH fields description

3.1.2.9 Register TRIGGER_MASK_LOW

This register controls the trigger mask

Address	Access		TRIGGER_MASK_LOW																																	
	HW	SW	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		
+20'H	R	RW	trigger_mask_low[31]	trigger_mask_low[30]	trigger_mask_low[29]	trigger_mask_low[28]	trigger_mask_low[27]	trigger_mask_low[26]	trigger_mask_low[25]	trigger_mask_low[24]	trigger_mask_low[23]	trigger_mask_low[22]	trigger_mask_low[21]	trigger_mask_low[20]	trigger_mask_low[19]	trigger_mask_low[18]	trigger_mask_low[17]	trigger_mask_low[16]	trigger_mask_low[15]	trigger_mask_low[14]	trigger_mask_low[13]	trigger_mask_low[12]	trigger_mask_low[11]	trigger_mask_low[10]	trigger_mask_low[9]	trigger_mask_low[8]	trigger_mask_low[7]	trigger_mask_low[6]	trigger_mask_low[5]	trigger_mask_low[4]	trigger_mask_low[3]	trigger_mask_low[2]	trigger_mask_low[1]	trigger_mask_low[0]		
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 3.18 - Register TRIGGER_MASK_LOW

Name	Type	Size	Description
trigger_mask_low[31:0]	U	32	Define the signals inside the input signal [31:0] used for trigger comparison with trigger_value.

Table 3.19 - Register TRIGGER_MASK_LOW fields description

3.1.2.10 Register TRIGGER_MASK_MED

This register controls the trigger mask

Address	Access		TRIGGER_MASK_MED																																								
	HW	SW	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									
+24'H	R	RW	trigger_mask_med[31]	trigger_mask_med[30]	trigger_mask_med[29]	trigger_mask_med[28]	trigger_mask_med[27]	trigger_mask_med[26]	trigger_mask_med[25]	trigger_mask_med[24]	trigger_mask_med[23]	trigger_mask_med[22]	trigger_mask_med[21]	trigger_mask_med[20]	trigger_mask_med[19]	trigger_mask_med[18]	trigger_mask_med[17]	trigger_mask_med[16]	trigger_mask_med[15]	trigger_mask_med[14]	trigger_mask_med[13]	trigger_mask_med[12]	trigger_mask_med[11]	trigger_mask_med[10]	trigger_mask_med[9]	trigger_mask_med[8]	trigger_mask_med[7]	trigger_mask_med[6]	trigger_mask_med[5]	trigger_mask_med[4]	trigger_mask_med[3]	trigger_mask_med[2]	trigger_mask_med[1]	trigger_mask_med[0]									
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U		
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R		
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 3.20 - Register TRIGGER_MASK_MED

Name	Type	Size	Description
trigger_mask_med[31:0]	U	32	Define the active signals inside the input signal [63:32] used for trigger comparison with trigger_value.

Table 3.21 - Register TRIGGER_MASK_MED fields description

3.1.2.11 Register TRIGGER_MASK_HIGH

This register controls the trigger mask

Address	Access		TRIGGER_MASK_HIGH																																					
	HW	SW	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						
+28'H	R	RW	trigger_mask_high[31]	trigger_mask_high[30]	trigger_mask_high[29]	trigger_mask_high[28]	trigger_mask_high[27]	trigger_mask_high[26]	trigger_mask_high[25]	trigger_mask_high[24]	trigger_mask_high[23]	trigger_mask_high[22]	trigger_mask_high[21]	trigger_mask_high[20]	trigger_mask_high[19]	trigger_mask_high[18]	trigger_mask_high[17]	trigger_mask_high[16]	trigger_mask_high[15]	trigger_mask_high[14]	trigger_mask_high[13]	trigger_mask_high[12]	trigger_mask_high[11]	trigger_mask_high[10]	trigger_mask_high[9]	trigger_mask_high[8]	trigger_mask_high[7]	trigger_mask_high[6]	trigger_mask_high[5]	trigger_mask_high[4]	trigger_mask_high[3]	trigger_mask_high[2]	trigger_mask_high[1]	trigger_mask_high[0]						
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U		
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 3.22 - Register TRIGGER_MASK_HIGH

Name	Type	Size	Description
trigger_mask_high[31:0]	U	32	Define the active signals inside the input signal [95:64] used for trigger comparison with trigger_value.

Table 3.23 - Register TRIGGER_MASK_HIGH fields description

3.1.2.12 Register TRIGGER_VALUE_LOW

This register controls the trigger value

Address	Access		TRIGGER_VALUE_LOW																															
	HW	SW	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
+2C'H	R	RW	trigger_value_low[31]	trigger_value_low[30]	trigger_value_low[29]	trigger_value_low[28]	trigger_value_low[27]	trigger_value_low[26]	trigger_value_low[25]	trigger_value_low[24]	trigger_value_low[23]	trigger_value_low[22]	trigger_value_low[21]	trigger_value_low[20]	trigger_value_low[19]	trigger_value_low[18]	trigger_value_low[17]	trigger_value_low[16]	trigger_value_low[15]	trigger_value_low[14]	trigger_value_low[13]	trigger_value_low[12]	trigger_value_low[11]	trigger_value_low[10]	trigger_value_low[9]	trigger_value_low[8]	trigger_value_low[7]	trigger_value_low[6]	trigger_value_low[5]	trigger_value_low[4]	trigger_value_low[3]	trigger_value_low[2]	trigger_value_low[1]	trigger_value_low[0]
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 3.24 - Register TRIGGER_VALUE_LOW

Name	Type	Size	Description
trigger_value_low[31:0]	U	32	Define the value to be compared with input signal [31:0] for trigger generation.

Table 3.25 - Register TRIGGER_VALUE_LOW fields description

3.1.2.13 Register TRIGGER_VALUE_MED

This register controls the trigger value

Address	Access		TRIGGER_VALUE_MED																															
	HW	SW	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
+30'H	R	RW	trigger_value_med[31]	trigger_value_med[30]	trigger_value_med[29]	trigger_value_med[28]	trigger_value_med[27]	trigger_value_med[26]	trigger_value_med[25]	trigger_value_med[24]	trigger_value_med[23]	trigger_value_med[22]	trigger_value_med[21]	trigger_value_med[20]	trigger_value_med[19]	trigger_value_med[18]	trigger_value_med[17]	trigger_value_med[16]	trigger_value_med[15]	trigger_value_med[14]	trigger_value_med[13]	trigger_value_med[12]	trigger_value_med[11]	trigger_value_med[10]	trigger_value_med[9]	trigger_value_med[8]	trigger_value_med[7]	trigger_value_med[6]	trigger_value_med[5]	trigger_value_med[4]	trigger_value_med[3]	trigger_value_med[2]	trigger_value_med[1]	trigger_value_med[0]
Reset Value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
HW Access			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
SW Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 3.26 - Register TRIGGER_VALUE_MED

Name	Type	Size	Description
trigger_value_med[31:0]	U	32	Define the value to be compared with input signal [63:32] for trigger generation.

Table 3.27 - Register TRIGGER_VALUE_MED fields description

3.1.2.14 Register TRIGGER_VALUE_HIGH

This register controls the trigger value

Address	Access		TRIGGER_VALUE_HIGH																															
	HW	SW	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

+34'H																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			</
-------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	----

Table 3.28 - Register TRIGGER_VALUE_HIGH

Name	Type	Size	Description
trigger_value_high[31:0]	U	32	Define the value to be compared with input signal [95:64] for trigger generation.

Table 3.29 - Register TRIGGER_VALUE_HIGH fields description

3.1.2.15 Register TRIGGER_POINT

This register controls the trigger value

Address	Access		TRIGGER_POINT																																
	HW	SW	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	
+38'H	R	RW																	trigger_point[15]	trigger_point[14]	trigger_point[13]	trigger_point[12]	trigger_point[11]	trigger_point[10]	trigger_point[9]	trigger_point[8]	trigger_point[7]	trigger_point[6]	trigger_point[5]	trigger_point[4]	trigger_point[3]	trigger_point[2]	trigger_point[1]	trigger_point[0]	
Reset Value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Type																			U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	
HW Access																			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
SW Access																			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Table 3.30 - Register TRIGGER_POINT

Name	Type	Size	Description
trigger_point[15:0]	U	16	Number of samples stored in the memory after the trigger.

Table 3.31 - Register TRIGGER_POINT fields description

3.1.2.16 Register FIRSTSAMPLE

This register provides the location of the first sample in the LA memory

Address	Access		FIRSTSAMPLE																															
	HW	SW	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
+3C'H	W	R																	firstsample[15]	firstsample[14]	firstsample[13]	firstsample[12]	firstsample[11]	firstsample[10]	firstsample[9]	firstsample[8]	firstsample[7]	firstsample[6]	firstsample[5]	firstsample[4]	firstsample[3]	firstsample[2]	firstsample[1]	firstsample[0]

[illegible]

Table 3.32 - Register FIRSTSAMPLE

Name	Type	Size	Description
firstsample[15:0]	U	16	Location in the memory of the first sample.

Table 3.33 - Register FIRSTSAMPLE fields description

3.2 Captured Data Dump

When the LA capture is completed, indicated by **started** = 0 and **triggered** = 1 in STATUS register, the SW can dump the content of the Embedded Logic Analyzer memory.

To dump the content of the memory, the SW shall first check the value of **writeaddr** in STATUS register which indicates if the capture as completely filled or not the Embedded LA memory.

If the value of **writeaddr** is the size of the memory, the SRAM has been completely filled. In such case, the SW shall dump the memory in two steps

```
from 0x100000 + firstsample<<4 to 0x100000 + writeaddr<<4 - 4
```

and from 0x100000 to 0x100000 + **firstsample**<<4 – 4

If the value of **writeaddr** is lower than the size of the memory, the SRAM has been partially filled. In such case, the SW shall dump from 0x100000 to 0x100000 + **writeaddr**<<4

As example with a SRAM width of 65535, and **firstsample** = 8192. From AHB point of view, the first sample is located at the address 0x100000 + **firstsample**<<4 = 0x120000

If **writeaddr** = 65535, then the dump shall be done in two steps

from 0x120000 to 0x1FFFFC

and from 0x100000 to 0x11FFFC

If **writeaddr** != 65535, then the dump shall be done in one step

from 0x100000 to 0x11FFFC

4 Integration in RW NX IP

4.1 Clocking scheme

To ease the analysis, it is recommended to clock the Embedded Logic Analyzer with a clock aligned with the MAC-PHY Clock which is faster than the Platform and MAC Core Clocks. This allows capturing the MAC-PHY signals synchronously without missing pulses coming from the Platform or MAC Core clock domains.

As an example, if the `mpif_clk` is running at 60MHz, the `mac_core_clk` at 80MHz and the `plf_clk` at 100MHz, it is recommended to generate the `la_clk` at 120MHz aligned with the `mpif_clk`.

Except for the MAC-PHY signals which are captured synchronously, all the others input signals are considered as asynchronous.

4.2 Debug Port

It is recommended to connect the MAC-PHY interface, the Software Debug Port and the MAC Core debug signals to the Embedded LA as following:

input_data	Signal name
95-94	0
93	<code>mpif_rfshutdown</code>
92	<code>mpif_txreq</code>
91-84	<code>mpif_txdata[7:0]</code>
83	<code>mpif_macdatavalid</code>
82	<code>mpif_phyrdy</code>
81	<code>mpif_txend_p</code>
80	<code>mpif_rxreq</code>
79-72	<code>mpif_rxdata</code>
71	<code>mpif_cca_pri20</code>
70	<code>mpif_cca_sec20</code>
69	<code>mpif_rxendfortiming_p</code>
68	<code>mpif_rxerr_p</code>
67	<code>mpif_rxend_p</code>
66	<code>mpif_phyerr_p</code>
65	<code>mpif_keeprfon</code>
64	<code>mpif_rifsrxdetected</code>
63-32	<code>diag_mac_hw</code>
31-0	<code>diag_mac_sw</code>