# Debugware

*User Guide*

03/2018

Hercules Microelectronics, Inc.
China

# Contents

# 1 Introduction

This document mainly describes the usage of the debugware IP. The debugware IP is an embedded logic analyzer to help designer inspect the signal transition inside FPGA. The waveform of sampled signals is stored in the embedded memory and the data can be retrieved through JTAG port after trigging events happened. Then, the designers can get a waveform picture according to the conditions configured.

The debugware IP supports the following features:

- Trigger condition can be configured dynamically
- Support real-time capture
- Support trigger type and condition
  - Arithmetic
    - equal
    - not equal
  - Edge
    - rise
    - fall
- Support up to 4 LA cores
- Multiple LA cores management
- Multiple EMB utilized to expand captured data width and depth
  - M7&M5: data width:2~96 depth:512|1024|2048|4096|8192|16384
  - HR3&HR2: data width:2~72 depth:512|1024|2048|4096|8192
  - C1: data width:2~160 depth:512|1024|2048|4096|8192|16384|32768|65536

**Note**:

- ➢ Please make sure that the design meets timing requirements.
- ➢ Do not exceed the total number of EMBs in your design after adding the debugware IP.

Device family support:

HME-M7, HME-M5, HME-HR3, HME-HR2, HME-C1

# 2 Debugware Overview

## 2.1 Pin Description

*Table 2-1* ***debugware interface***

| Interface | Name | Direction | Width | Description |
|---|---|---|---|---|
| User interface | ref_clk_n (n=0,1,2,3) | Input | 1 | Input reference clock, used to sample data |
| | data_in_n (n=0,1,2,3) | Input | Data Width | Input data used to inspect, from FPGA |
| | trig_out_n (n=0,1,2,3) | Output | 1 | Trigger output, active high, indicates trigger condition is met. |

## 2.2 Block Diagram

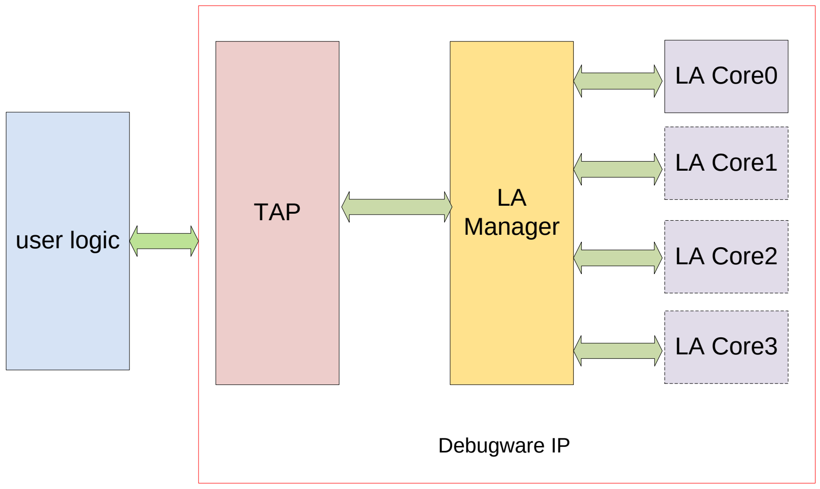


*Figure 2-1* ***debugware block diagram***

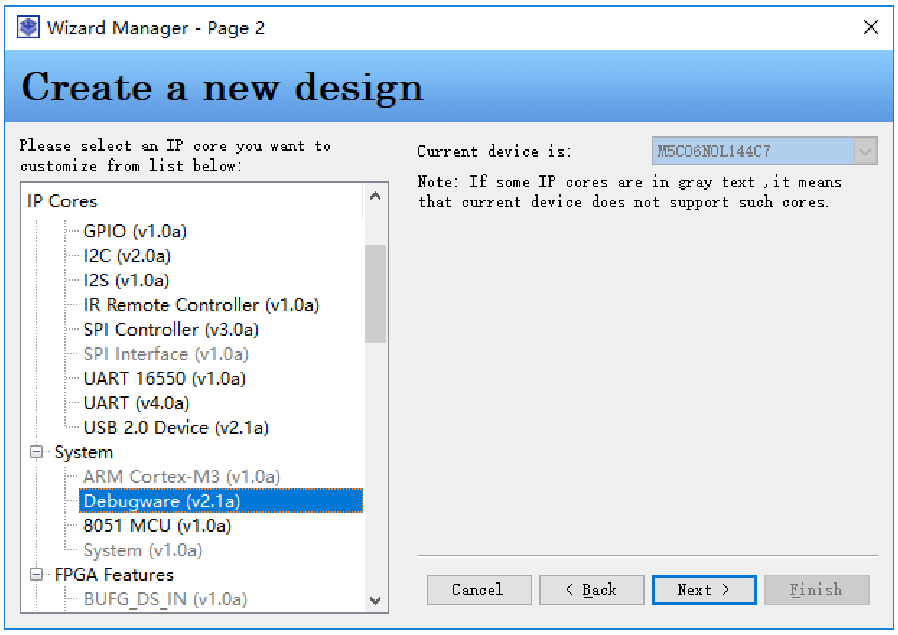
The debugware IP can be divided into three modules: TAP, LA Manager and LA Cores as shown in above figure.

- TAP module is the wrapper logic of debug interface with JTAG controller
- LA Manager is the control unit of debugware IP. It is a bridge between TAP and LA Cores. There is only one LA Manager but it can control up to 4 LA Cores simultaneously.
- LA Core is the executive unit in debugware IP. It includes trigger generator module and storage module.

# 3 Debugware IP Usage

Designer can use the debugware to inspect the internal signals of FPGA following these steps.

**Step 1**. Finish your design and instantiate the debugware IP on fuxi as Figure 3-1.



Wizard Manager - Page 2

Create a new design

Please select an IP core you want to customize from list below:

IP Cores
- GPIO (v1.0a)
- I2C (v2.0a)
- I2S (v1.0a)
- IR Remote Controller (v1.0a)
- SPI Controller (v3.0a)
- SPI Interface (v1.0a)
- UART 16550 (v1.0a)
- UART (v4.0a)
- USB 2.0 Device (v2.1a)

System
- ARM Cortex-M3 (v1.0a)
- Debugware (v2.1a)
- 8051 MCU (v1.0a)
- System (v1.0a)

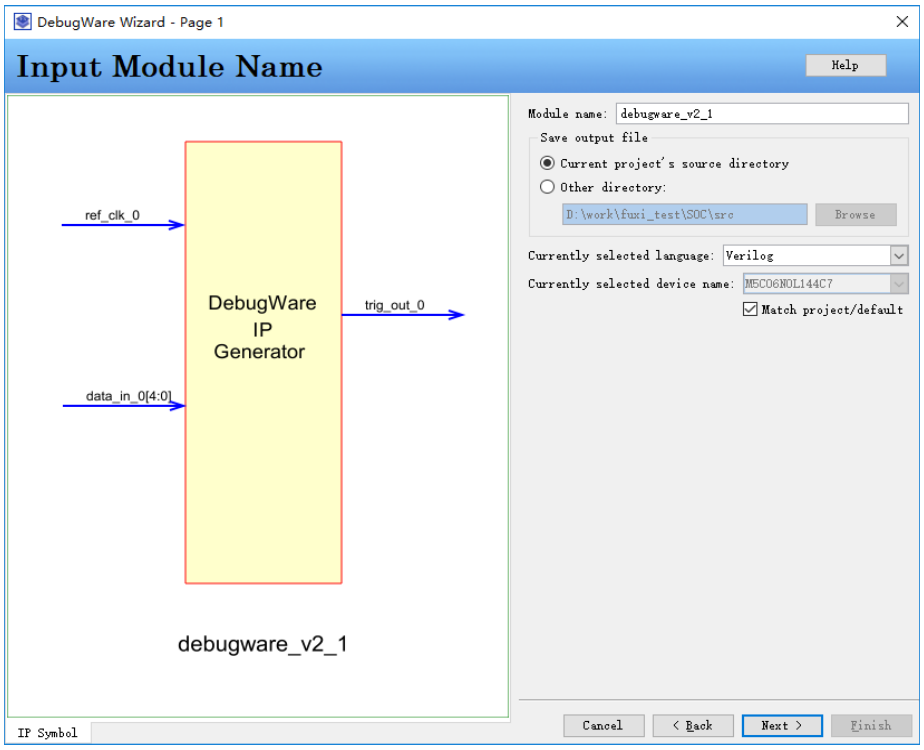
FPGA Features
- BUFG_DS_IN (v1.0a)

Current device is: M5C06N0L144C7

Note: If some IP cores are in gray text ,it means that current device does not support such cores.

*Figure 3-1* ***select the debugware IP***

Then, designer can configure the debugware IP according to the design as Figure 3-2(a), Figure 3-2(b), Figure 3-2(c)



DebugWare Wizard - Page 1

Input Module Name

Module name: debugware_v2_1

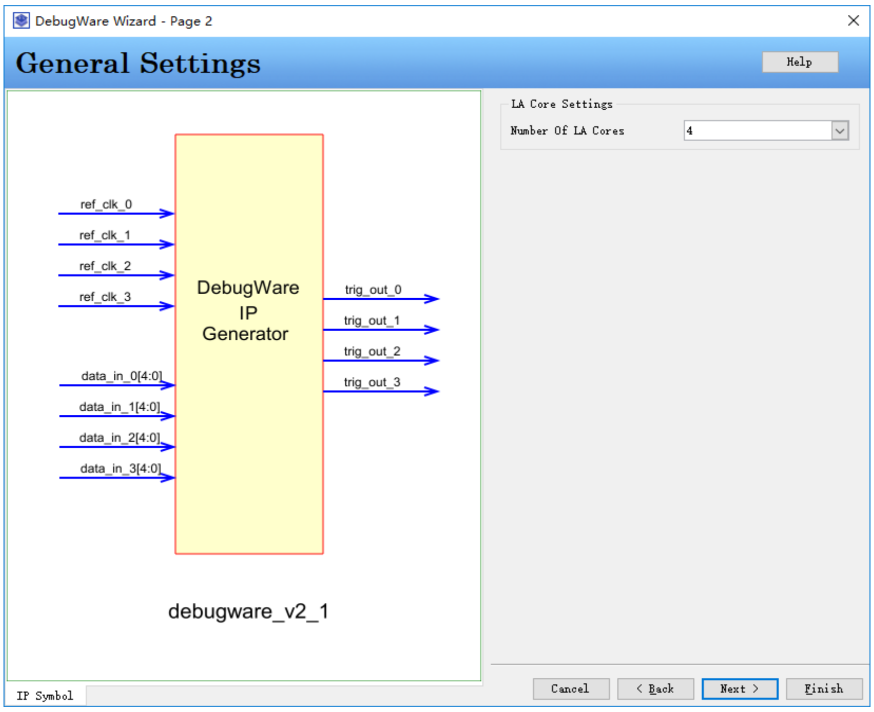
Save output file
- [✓ Current project's source directory]
- Other directory: D:\work\fuxi_test\SOC\src

Currently selected language: Verilog

Currently selected device name: M5C06N0L144C7

[✓] Match project/default

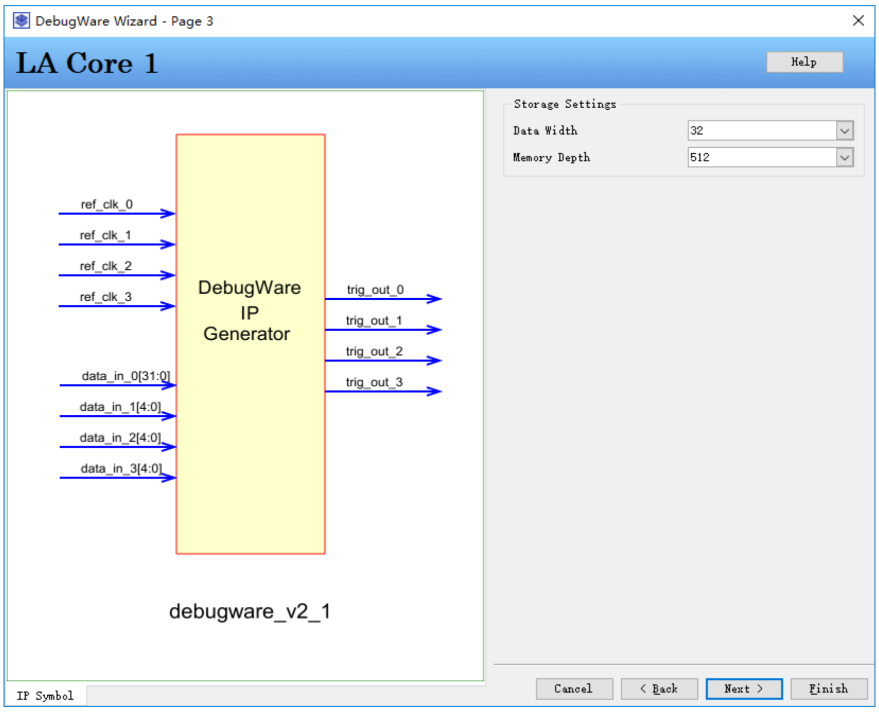
*Figure 3-2(a)* ***set module name***



LA Core Settings

Number Of LA Cores: 4

*Figure 3-2(b)* ***select LA Core number(max = 4)***



Storage Settings

Data Width: 32

Memory Depth: 512

*Figure 3-2(c)* ***select storage size***

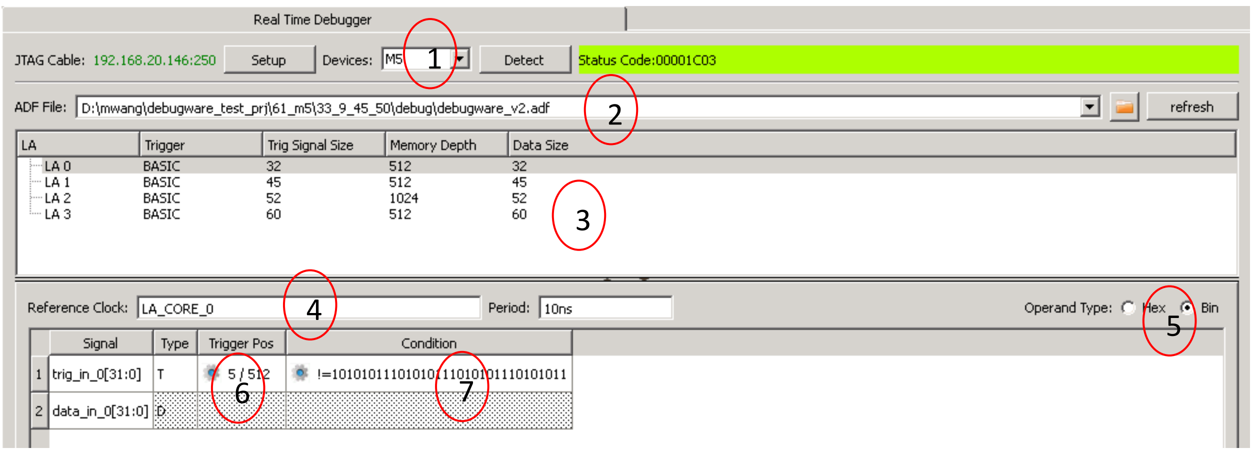
The designer should set the storage parameters for every LA core. After doing all the things, you can click the "finish" button to generate the debugware IP. Then, the debugware IP will be added in the design.

**Step 2**.Finish the instantiation including connecting signals to input and output ports of debugware instance.

**Step 3**. Run all flow and generate ".acf" file.

**Step 4**. Download the ".acf" file to board.

**Step 5**. Open the Real Time Debug UI, click Tools→Real Time Debug. As shown in Figure 3-3.



JTAG Cable: 192.168.20.146:250 Devices: M5 Status Code:00001C03

ADF File: D:\mwang\debugware_test_prj\61_m5\33_9_45_50\debug\debugware_v2.adf

| LA | Trigger | Trig Signal Size | Memory Depth | Data Size |
|---|---|---|---|---|
| LA 0 | BASIC | 32 | 512 | 32 |
| LA 1 | BASIC | 45 | 512 | 45 |
| LA 2 | BASIC | 52 | 1024 | 52 |
| LA 3 | BASIC | 60 | 512 | 60 |

Reference Clock: LA_CORE_0 Period: 10ns Operand Type: Hex [✓ Bin]

| | Signal | Type | Trigger Pos | Condition |
|---|---|---|---|---|
| 1 | trig_in_0[31:0] | T | 5 / 512 | !=10101011101010111010101110101011 |
| 2 | data_in_0[31:0] | D | | |

*Figure 3-3* ***open real time debug UI***

In the Real Time Debug UI, designer can find the information as below.

(1)device information, as label①.

(2)".adf "file path, as label②.

(3)The LA Core information, as label③, including number, trigger data width(equal to data width), memory depth and data width, matching with the configuration of debugware IP in the step 1.

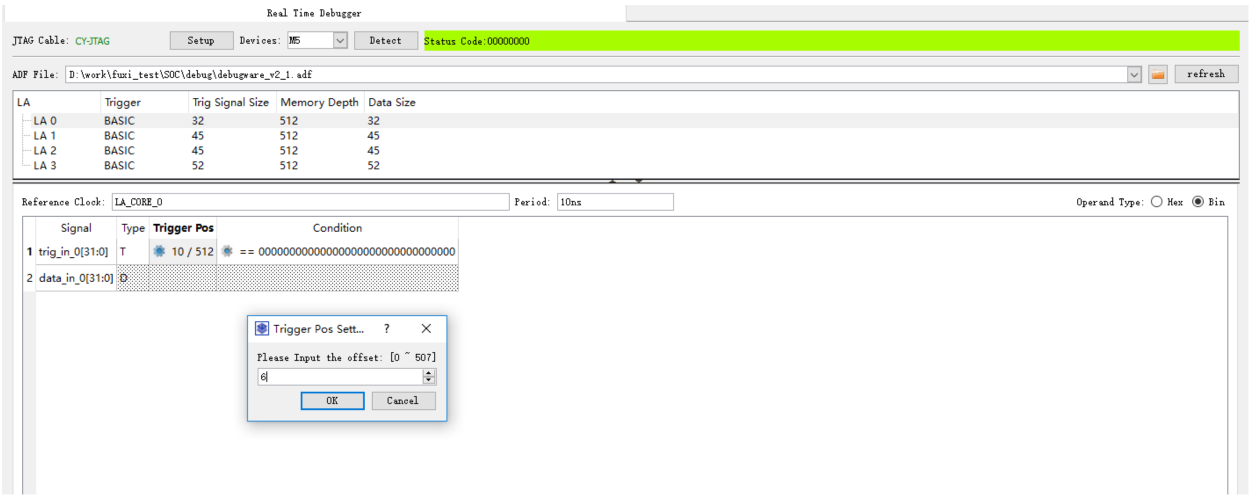
(4)the LA Core currently selected, as label④.

(5)Operand type select, as label⑤,"Hex" and "Bin".

(6)Offset setting, as label⑥.

(7)Condition setting, as label⑦. User sets the value according to operand type selected.

Then, designer can set the offset value, as shown in Figure 3-4



JTAG Cable: CY-JTAG Devices: M5 Status Code:00000000

ADF File: D:\work\fuxi_test\SOC\debug\debugware_v2_1.adf

| LA | Trigger | Trig Signal Size | Memory Depth | Data Size |
|---|---|---|---|---|
| LA 0 | BASIC | 32 | 512 | 32 |
| LA 1 | BASIC | 45 | 512 | 45 |
| LA 2 | BASIC | 45 | 512 | 45 |
| LA 3 | BASIC | 52 | 512 | 52 |

Reference Clock: LA_CORE_0 Period: 10ns Operand Type: Hex [✓ Bin]

| | Signal | Type | Trigger Pos | Condition |
|---|---|---|---|---|
| 1 | trig_in_0[31:0] | T | 10 / 512 | == 00000000000000000000000000000000 |
| 2 | data_in_0[31:0] | D | | |

Please Input the offset: [0 ~ 507]

6

*Figure 3-4* ***set offset***

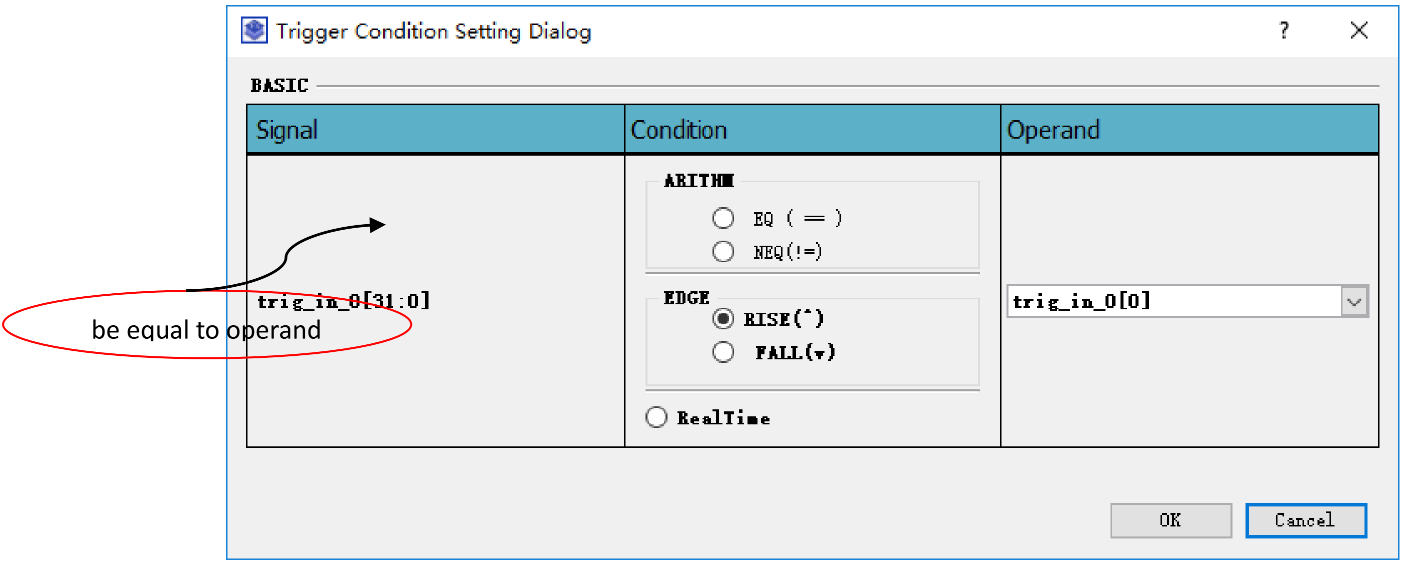
As shown in Figure 3-5, users can set the trigger condition value.

In this operation, two types can be chosen, "hex" and "bin". The bit number you type must be equal to trig_in width if binary and the bit not used as trigger condition can be set" x "instead of "0" or "1". ”x" means that the bit is not used as operand for trigging.

If "bin" is selected and there is enough digit, it will no longer be allowed to enter.

For example, if the width of trig_in is 8, but user does not want to use bit 1, bit3, bit5 and other bits are all 1. So, the value should be entered is "11x1x1x1".

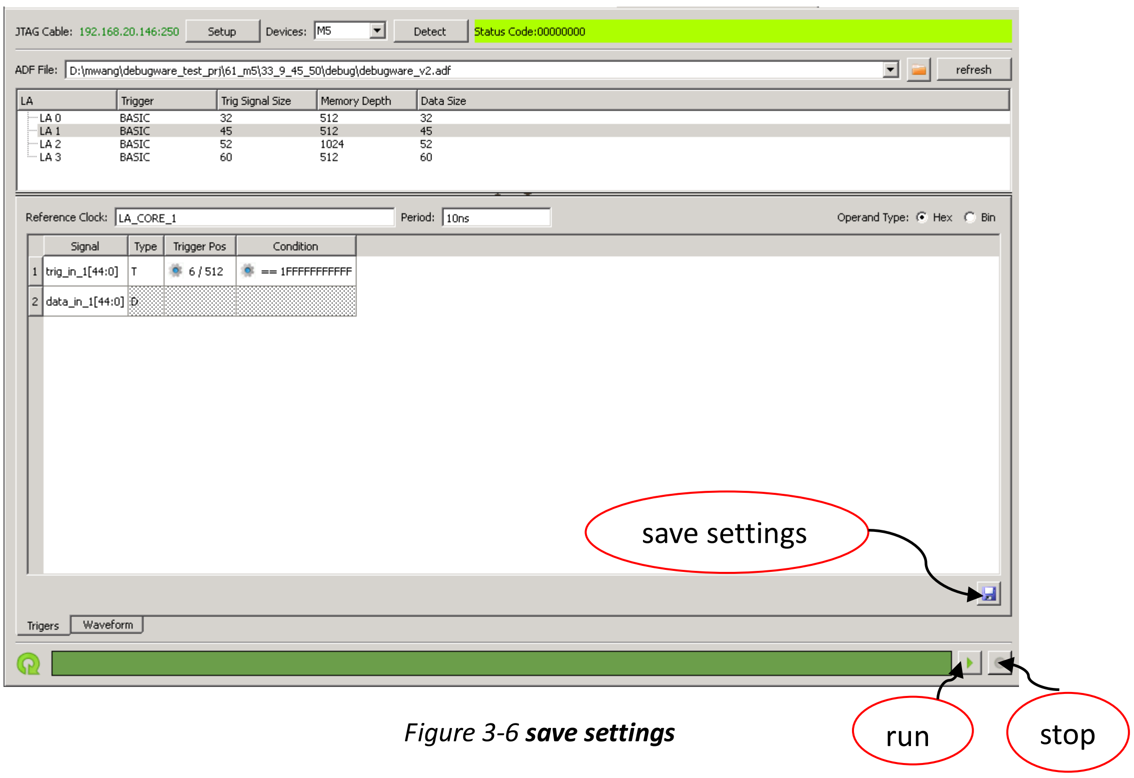
But if Hex is selected, any value less than 2^n (n is width of trig_in) is allowed to type in except "x". So, if masking some bits is needed, please choose "bin".


Trigger Condition Setting Dialog

BASIC

| Signal | Condition | Operand |
|---|---|---|
| trig_in_0[31:0] | ARITHM: ( ) EQ ( == ) ( ) NEQ(!=); EDGE: (●) RISE(↑) ( ) FALL(↓); ( ) RealTime | trig_in_0[0] |

*Figure 3-5* ***set condition***


JTAG Cable: 192.168.20.146:250 Devices: M5 Status Code:00000000

ADF File: D:\mwang\debugware_test_prj\61_m5\33_9_45_50\debug\debugware_v2.adf

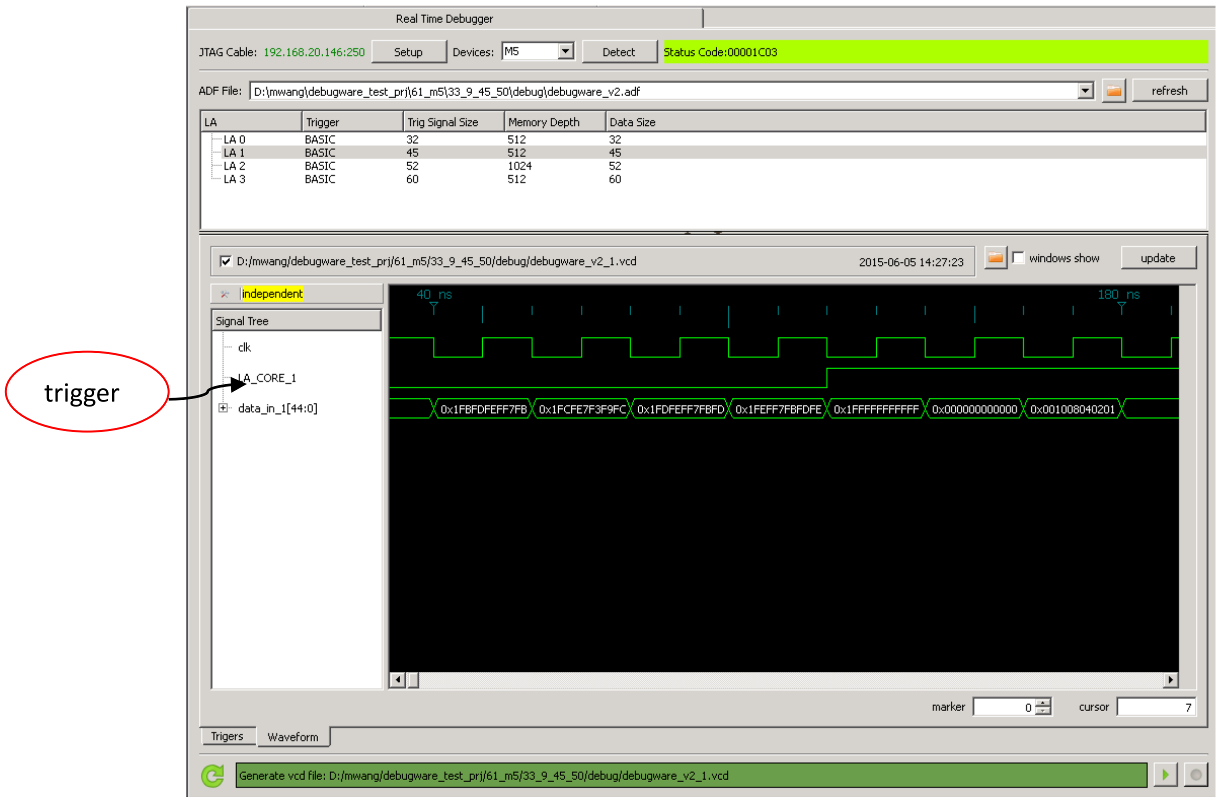
| LA | Trigger | Trig Signal Size | Memory Depth | Data Size |
|---|---|---|---|---|
| LA 0 | BASIC | 32 | 512 | 32 |
| LA 1 | BASIC | 45 | 512 | 45 |
| LA 2 | BASIC | 52 | 1024 | 52 |
| LA 3 | BASIC | 60 | 512 | 60 |

Reference Clock: LA_CORE_1 Period: 10ns Operand Type: (●) Hex ( ) Bin

| | Signal | Type | Trigger Pos | Condition |
|---|---|---|---|---|
| 1 | trig_in_1[44:0] | T | 6 / 512 | == 1FFFFFFFFFFF |
| 2 | data_in_1[44:0] | D | | |

*Figure 3-6* ***save settings***

**Step 6**. After all settings done, click the "run", as Figure 3-6. As shown in Figure 3-7, user can see the waveform of signals need to inspect if trigger events has happened.
Now, users can start the debug work.



JTAG Cable: 192.168.20.146:250 Devices: M5 Status Code:00001C03

ADF File: D:\mwang\debugware_test_prj\61_m5\33_9_45_50\debug\debugware_v2.adf

| LA | Trigger | Trig Signal Size | Memory Depth | Data Size |
|---|---|---|---|---|
| LA 0 | BASIC | 32 | 512 | 32 |
| LA 1 | BASIC | 45 | 512 | 45 |
| LA 2 | BASIC | 52 | 1024 | 52 |
| LA 3 | BASIC | 60 | 512 | 60 |

D:/mwang/debugware_test_prj/61_m5/33_9_45_50/debug/debugware_v2_1.vcd 2015-06-05 14:27:23

independent

40 ns 180 ns

clk

LA_CORE_1

data_in_1[44:0] 0x1FBFDFEFF7FB 0x1FCFE7F3F9FC 0x1FDFEFF7FBFD 0x1FEFF7FBFDFE 0x1FFFFFFFFFFF 0x000000000000 0x001008040201

marker 0 cursor 7

Generate vcd file: D:/mwang/debugware_test_prj/61_m5/33_9_45_50/debug/debugware_v2_1.vcd

*Figure 3-7* ***waveform***

The settings as below can make the debug work more conveniently.

In the signal tree area, these operations can be done as below:

1. change color to highlight what you want to see
   - shortcut key "t"
   - right mouse button → select "Set Color" → select color
2. show data in different format
   - right mouse button → select "Data Format" → select format
3. create new bus for two or more signals you care for
   - Create: select the signals → right mouse button → select "Create Bus"
   - Rename: right mouse button → select "Add Alias"
   - Show original name: right mouse button → select "Show Original Name"
4. signal value of current time can be shown in the signal tree
   - right mouse button → select "Show Value"

5.export data and generate "dump.txt" in the project directory
   - right mouse button → select "Export Data"

6.search signals by value
   - right mouse button → select "Search"

7.move marked line to previous or next clock edge(rise/fall)
   - on the keyboard:"←": move to previous edge, "→": move to next edge

These settings can help to make the waveform more clearly in waveform area.

1.show grid to differentiate different signals

- right mouse button → select "Show Grid"

2. show the current value on the position of mouse

- right mouse button → select "Fit Here"
- double-click left mouse button

3. zoom function

- overall
  - right mouse button → select "Zoom In" or "Zoom Out"
  - combination key "ctrl" + "+" /"-"
- local
  - right mouse button → select "Zoom Rect"
- fit to all waveform window
  - right mouse button → select "Zoom All"

4.length measure

- right mouse button → select "Length Measure" to set scale ruler to overlay the marked line and dragging the scale ruler to any position by left mouse button to finish the length measure

5.open new waveform window

- right mouse button → select "New Window"

6.change timeline type

- right mouse button → select "Time Line" → select "In Cycle"/"In Timescale"

# 4 Generate File Directory Structure

The debugware IP Wizard generated file includes: source files (src) and related document. The detailed design directory structure is as below.

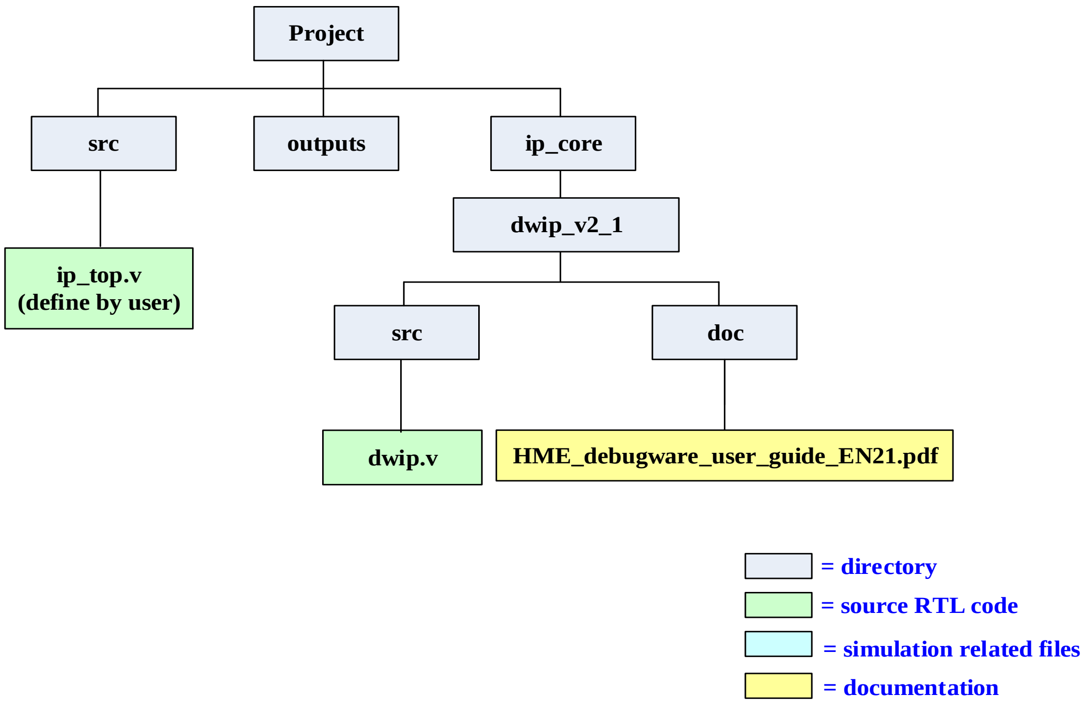


*Figure 4-1* ***IP wizard generated file directory structure***

*Table 4-1* ***Generated File Directory structure***

| Directory | Description |
|---|---|
| src\ | Directory for project source code, including IP wizard generate code. |
| ip_core\ | The directory specially for all IPs |
| \dwip_v2_1 | Directory for debugware IP |
| \doc\HME_debugware_user_guide_EN21.pdf | User guide for debugware IP |
| | |
| \src | IP Design RTL |
| \src\dwip.v | The src of debugware IP (Encrypted) |

## Revision History

| Revision | Date | Comments |
|---|---|---|
| 1.0 | 2013-07-01 | Initial release |
| 2.0 | 2018-03-26 | Do not need to re_run all flow when trigger conditions are changed |
| 2.1 | 2018-03-26 | Support C1 device |
| | | |
| | | |
| | | |
| | | |
| | | |