Contents

Contents ................................................................................................................................................. 2

1 Introduction ....................................................................................................................................... 3

2 System Level Structure ..................................................................................................................... 4

3 Example Result .................................................................................................................................. 12

4 Pin and Design Source description .................................................................................................... 14

  4.1 Pin descriptions ............................................................................................................................... 14

  4.2 Pin assignments ............................................................................................................................... 14

  4.3 Design source ................................................................................................................................. 14

5 Revision History ................................................................................................................................. 16
1 Introduction

This document describes how to use the AHB Slave Decoder IP to access the AHB slaves. Following is the detail of the example:

- **Function**
  - In the example, the AHB Slave Decoder works as slave of ARM.
  - ARM accesses five slaves including the AHB interface EMB, AHB interface FIFO, AHB interface GPIO (two, one is as input port and the other is as output port) and EMB (through AHB2APB Bridge) on FPGA.
  - ARM write/read data to/from the five slaves through single and burst operation.

- The example works at
  - FPGA Array logic: 100MHz
  - ARM core: 200MHz

- Device: CME-M7(M7A12N0F484C7)
2 System Level Structure

The general structure of example as follows:

![Diagram of System Level Structure]

This case consists of 4 parts as shown in above figure: PLL, SOC(ARM), AHB Slave Decoder and five slaves.

1. PLL (generated by Wizard):
   a) In the example, this part has an input of 20MHz and provides 200MHz and 100MHz, used by ARM core and FPGA logic. The block can be configured as the figure 2-2 below in the Primace software by Wizard tool.
2. SOC(ARM) (generated by Wizard):
   a) This part is an ARM M3 processor core in the example, which contains two groups of AHB bus, AHB0 and AHB1. In the example, AHB Slave Decoder connects with the AHB0 slave interface.
   b) It is configured as the figure 2-3(a), figure 2-3(b).
The c code in the firmware file “main.c” is used to achieve the function of accessing five slaves on FPGA. The function flow as figure 2-4.

**Figure 2-3(b) set hex file**

**Figure 2-4 firmware flow**
The process of EMB and AHB2APB Bridge are same. In the AHB2APB Bridge test, EMB with common interface works as APB slave. So, ARM access EMB on FPGA through AHB2APB Bridge.

3. AHB Slave Decoder (generated by Wizard):
This IP works as the bridge between ARM and slaves with AHB interface. The number of slaves and start/end address of each slave can be configured. In this example, the configuration is as figure 2-5:

![AHB Slave Decoder configuration](image)

4. Five slaves part (generated by Wizard):

   (1) AHB interface EMB

   The EMB with AHB interface works at SP mode and the Base Address is set as A0000000. ARM access it by single and burst operation.

   The configuration of this IP is as figure 2-6(a), figure 2-6(b)
(2) AHB interface FIFO

The FIFO with AHB interface is software synchronous FIFO and the Base Address is set as A0000800. ARM access it by single operation with interrupt.
The configuration is as figure 2-7.

Figure 2-7 FIFO(AHB interface) configuration

(3)GPIO(output port)
The GPIO with AHB interface works as output port and the output data are used to control the LED on board. The Base Address is set to A0000C00. The configuration is as figure 2-8.

Figure 2-8 gpio_0(AHB interface) configuration
(4) GPIO (input port)

The GPIO with AHB interface works as input port and the input data is from the key SW1 on board. The Base Address is set as A0001000. The configuration is as figure 2-9.

![Figure 2-9 gpio_1(AHB interface) configuration](image)

(5) AHB2APB Bridge

ARM accesses the EMB with common interface through AHB Slave Decoder and AHB2APB Bridge. In the design, EMB works as APB slave.

The AHB2APB Bridge’s Base Address is set as A0001400. The configuration is as figure 2-10.
Figure 2-10 **ahb2apb bridge configuration**
3 Example Result

All the results of data transfer of all slaves will be displayed on PC using Tera Term tool and the GPIO test results are also displayed through LEDs on board.

After you download the acf file, the eight LEDs will be lighted, but when you press the button (SW1) the 8 LEDs will display as follows: LEDx (x is even) will be lighted → No LEDs are lighted → LEDx (x is odd) will be lighted. Meanwhile, other functions test will be displayed on PC.

The Tera Term is configured as figure 3-1 and all the results are as figure 3-2(a), figure 3-2(b), figure 3-2(c), figure 3-2(d).

![Figure 3-1 Tera Term configuration](image)

![Figure 3-2(a) FIFO test results](image)
User Guide of AHB Slave Decoder example

Figure 3-2(b) EMB test results

```c
/***********************
EMB testing...
Finish writing(singel) EMB !!!
Finish reading(singel) EMB !!!

Test: EGRAM writing and reading !!!
Finish writing, then reading and check...
EGRAM TEST PASSED !!!

Test: moving data from EGRAM to external EMB through DMA !!!
Finish initiating memory !!!
DMA Transfer Begin!!!
DMA Transfer Finish !!!
DMA data verification is OK.

Test: moving data from external EMB to EGRAM through DMA !!!
Finish initiating memory !!!
DMA Transfer Begin!!!
DMA Transfer Finish !!!
DMA data verification is OK.
END, EMB all functions are OK.
***********************/
```

Figure 3-2(c) AHB2APB Bridge test results

```c
/***********************
AHB2APB Bridge testing...
Finish writing(singel) external FPGA space(AHB2APB Bridge slave) !!!
Finish reading(singel) external FPGA space(AHB2APB Bridge slave) and comparison is ok!!!

Test: EGRAM writing and reading !!!
Finish writing, then reading and check...
EGRAM TEST PASSED !!!

Test: moving data from EGRAM to external EMB(AHB2APB Bridge slave) through DMA !!!
Finish initiating memory !!!
DMA Transfer Begin!!!
DMA Transfer Finish !!!
DMA data verification is OK.

Test: moving data from external EMB(AHB2APB Bridge slave) to EGRAM through DMA !!!
Finish initiating memory !!!
DMA Transfer Begin!!!
DMA Transfer Finish !!!
DMA data verification is OK.
END, AHB2APB Bridge all functions are OK.
***********************/
```

Figure 3-2(d) GPIO test results

```c
/***********************
GPIO testing...
Press key s1,8 LEDS lighting changed...

END, GPIO all functions are OK.
***********************
```

http://www.capital-micro.com
4 Pin and Design Source description

4.1 Pin descriptions

Table 4-1 The AHB Slave Decoder example top module pin description

<table>
<thead>
<tr>
<th>Name</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk_in</td>
<td>Input</td>
<td>1</td>
<td>Clock input (20MHz)</td>
</tr>
<tr>
<td>rst_n</td>
<td>Input</td>
<td>1</td>
<td>Reset input, low active</td>
</tr>
<tr>
<td>p0_t</td>
<td>Inout</td>
<td>32</td>
<td>For led display</td>
</tr>
<tr>
<td>p1_t</td>
<td>Inout</td>
<td>32</td>
<td>For button control</td>
</tr>
</tbody>
</table>

4.2 Pin assignments

The following figure 4-1 shows the detail pin assignments in IO Editor of Primace of M7 example.

![Figure 4-1 IO pin assignment](image)

4.3 Design source

The example RTL source files are shown in table 4-2.
### Table 4-2 The AHB Slave Decoder example’s source files description

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTL</td>
<td></td>
</tr>
<tr>
<td>./src/</td>
<td></td>
</tr>
<tr>
<td>/top.v</td>
<td>The top module, implements the connection of all sub modules.</td>
</tr>
<tr>
<td>/armcm3_v1.v</td>
<td>Arm core module, implemented by Wizard</td>
</tr>
<tr>
<td>/pll_v1.v</td>
<td>PLL module, implemented by Wizard</td>
</tr>
<tr>
<td>/emb_v1.v</td>
<td>AHB interface EMB module, implemented by Wizard</td>
</tr>
<tr>
<td>/fifo_v2.v</td>
<td>AHB interface FIFO module, implemented by Wizard</td>
</tr>
<tr>
<td>/ahb2apb_bridge_v1.v</td>
<td>AHB2APB Bridge module, implemented by Wizard</td>
</tr>
<tr>
<td>/gpio_v1_0.v</td>
<td>GPIO module, implemented by Wizard, as output port</td>
</tr>
<tr>
<td>/gpio_v1_1.v</td>
<td>GPIO module, implemented by Wizard, as input port</td>
</tr>
<tr>
<td>/slave_5_decoder.v</td>
<td>AHB Slave Decoder module, implemented by Wizard</td>
</tr>
<tr>
<td>/apb_emb.v</td>
<td>EMB with common interface module, works as APB slave, implemented by Wizard</td>
</tr>
</tbody>
</table>

Firmware

| /main.c                   | Configure registers of all slaves and write/read data to/from all slaves.  |
# 5 Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2014-12-10</td>
<td>Initial release</td>
</tr>
</tbody>
</table>