

实验平台说明

特征	说明
操作系统版本和系统类型	Windows 7 64 bit enterprise edition
EDA 软件名称和版本	Quartus II 13.1 64 bit
DE2-115 开发板编号	1904615S

自查清单

特征	Part 8/Lab6	Part9/Lab6	Part/Lab	Part ?/Lab?
原理图	√	√	√	
Verilog 代码	√	√	√	
Flow Summary	√	√	×	
RTL 图	√	√	×	
状态机图	×	×	×	
工艺图	√	√	√	
仿真图	×	×	×	
SignalTapII 图	×	×	×	
硬件运行图	√	√	√	

第一部分 Part 8/Lab6

一、实验目的

1. 熟悉 IP 核加法器，IP 核乘法器的使用方法。
2. 熟悉组合逻辑，状态机，选择器，多态电路的组合设计方法。

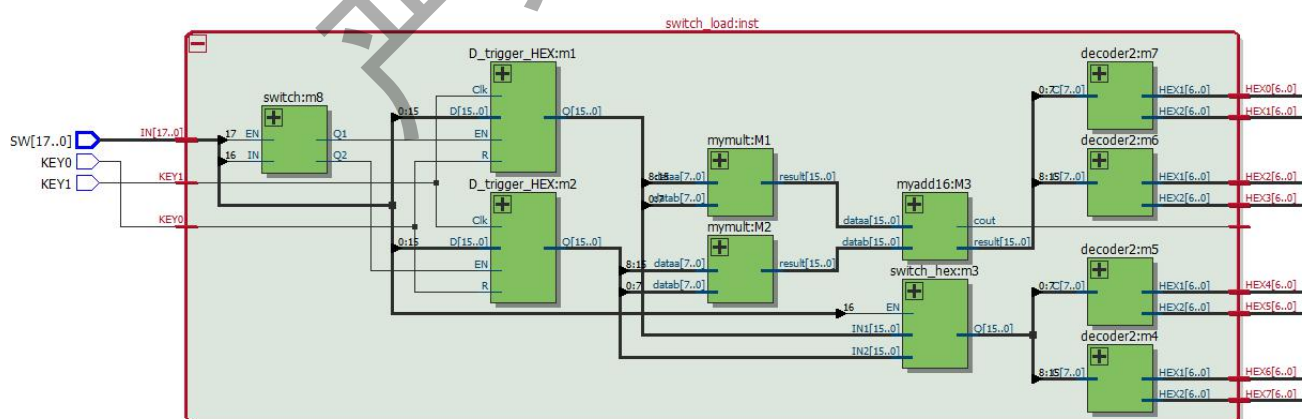
二、实验内容

1. 使用 LAB1 中数码管驱动程序, LAB3 的 D 触发器寄存器, LAB4 的多态选择器。
2. 使用 IP 核中的加法器，乘法器。

三、实验原理

1. 以 17 个开关为输入，以 KEY1 为 CLK，KEY0 为 reset，使用两个寄存器，储存两个值，并且显示在前四个固定的数码管上。
2. 利用触发器可以暂时存储的特点，储存两次输入的值。
3. 将两次触发器暂存的值经过运算，输出给后四个数码管。

RTL 级仿真



四、实验步骤

1) 建立工程，并添加文件。

1. 建立工程，选择常用文件夹，取名为 lab6。

2.添加 Block Diagram/Schematic 文件，取名为 lab6;

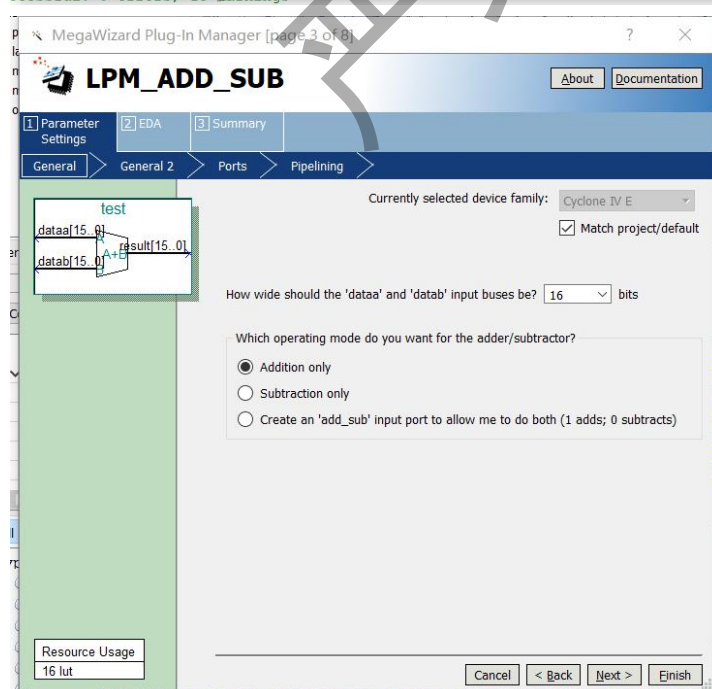
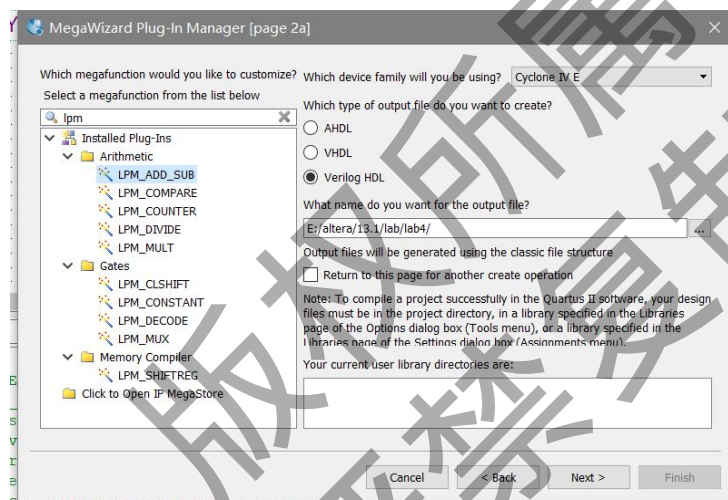
3.添加 Verilog HDL 文件，取名为 PART8

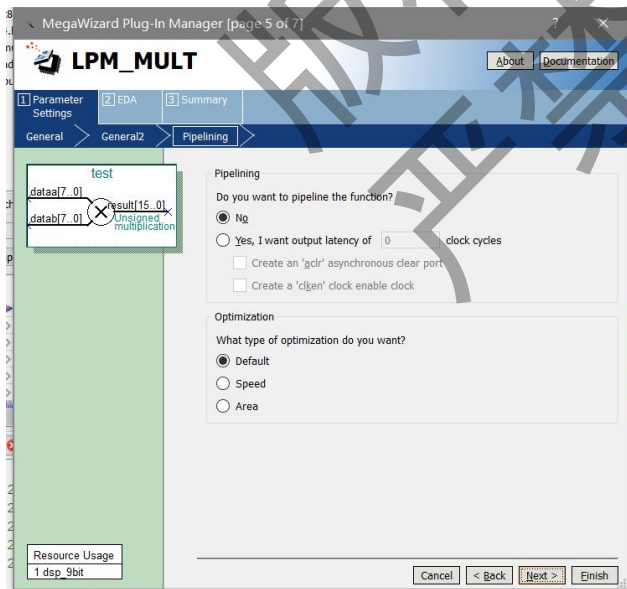
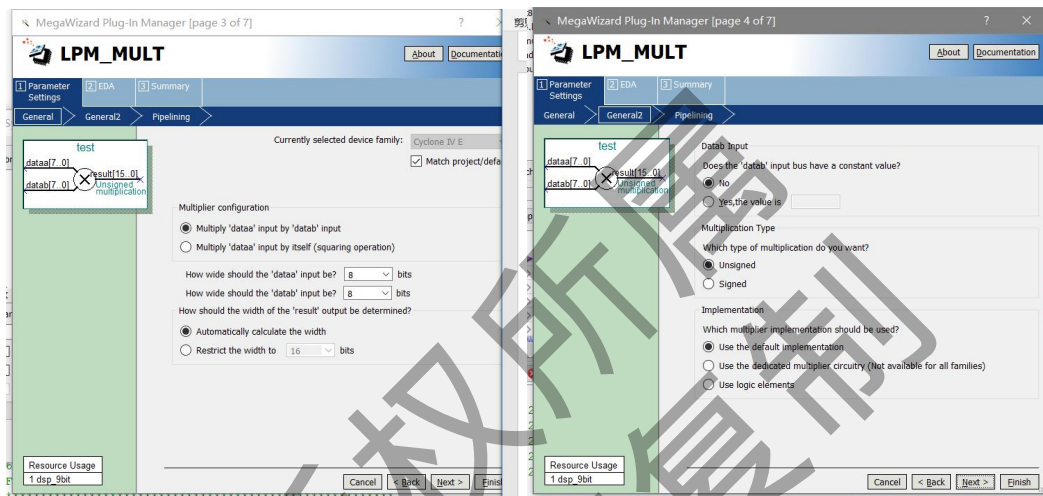
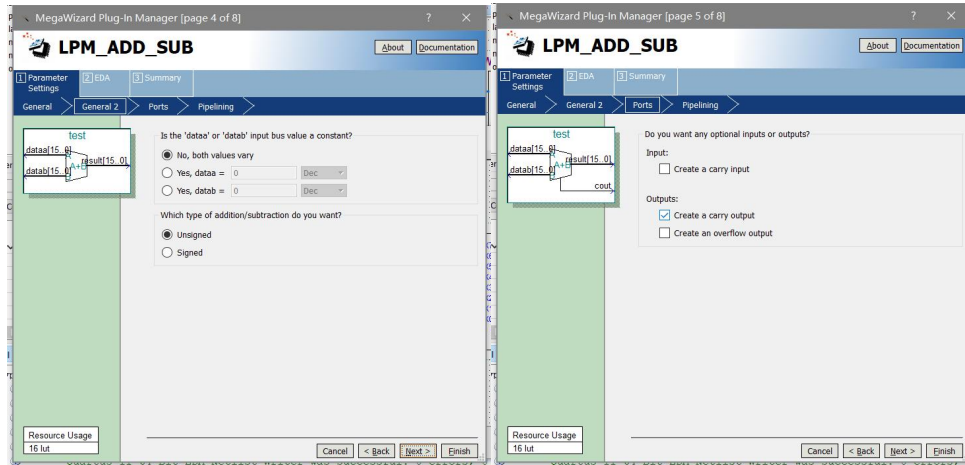
2) 首先理清思路:

在本次实验中，首先通过 SW 和 KEY1,KEY0 对输入进行储存，之后将储存好的数据进行运算调用 IP 核生成加法器乘法器.v 文件，并在模块中调用，采用自上而下的程序写法。

3) 以 Verilog 代码表示。

4) 调用 IP 核的加法器和乘法器。



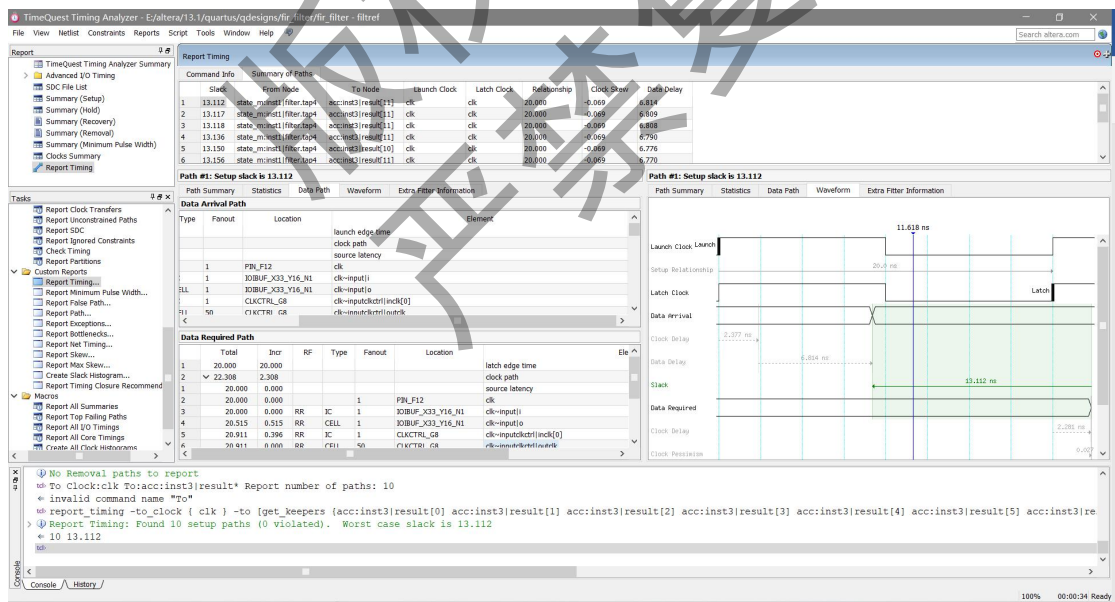


5) 编译并将引脚对应。

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved
HEX0[6]	Output	PIN_H22	6	B6_N0	PIN_H22	2.5 V (default)	
HEX0[5]	Output	PIN_J22	6	B6_N0	PIN_J22	2.5 V (default)	
HEX0[4]	Output	PIN_L25	6	B6_N1	PIN_L25	2.5 V (default)	
HEX0[3]	Output	PIN_L26	6	B6_N1	PIN_L26	2.5 V (default)	
HEX0[2]	Output	PIN_E17	7	B7_N2	PIN_E17	2.5 V (default)	8m
HEX0[1]	Output	PIN_F22	7	B7_N0	PIN_F22	2.5 V (default)	8m
HEX0[0]	Output	PIN_G18	7	B7_N2	PIN_G18	2.5 V (default)	8m
HEX1[6]	Output	PIN_U24	5	B5_N0	PIN_U24	2.5 V (default)	8m
HEX1[5]	Output	PIN_U23	5	B5_N1	PIN_U23	2.5 V (default)	8m
HEX1[4]	Output	PIN_W25	5	B5_N1	PIN_W25	2.5 V (default)	8m
HEX1[3]	Output	PIN_W22	5	B5_N0	PIN_W22	2.5 V (default)	8m
HEX1[2]	Output	PIN_W21	5	B5_N1	PIN_W21	2.5 V (default)	8m
HEX1[1]	Output	PIN_Y22	5	B5_N0	PIN_Y22	2.5 V (default)	8m
HEX1[0]	Output	PIN_M24	6	B6_N2	PIN_M24	2.5 V (default)	8m
HEX2[6]	Output	PIN_W28	5	B5_N1	PIN_W28	2.5 V (default)	8m
HEX2[5]	Output	PIN_W27	5	B5_N1	PIN_W27	2.5 V (default)	8m
HEX2[4]	Output	PIN_Y26	5	B5_N1	PIN_Y26	2.5 V (default)	8m
HEX2[3]	Output	PIN_W26	5	B5_N1	PIN_W26	2.5 V (default)	8m
HEX2[2]	Output	PIN_Y25	5	B5_N1	PIN_Y25	2.5 V (default)	8m
HEX2[1]	Output	PIN_AA26	5	B5_N1	PIN_AA26	2.5 V (default)	8m
HEX2[0]	Output	PIN_AA25	5	B5_N1	PIN_AA25	2.5 V (default)	8m
HEX3[6]	Output	PIN_Y19	4	B4_N0	PIN_Y19	2.5 V (default)	8m
HEX3[5]	Output	PIN_AF23	4	B4_N0	PIN_AF23	2.5 V (default)	8m
HEX3[4]	Output	PIN_AD24	4	B4_N0	PIN_AD24	2.5 V (default)	8m
HEX3[3]	Output	PIN_AA21	4	B4_N0	PIN_AA21	2.5 V (default)	8m
HEX3[2]	Output	PIN_AB20	4	B4_N0	PIN_AB20	2.5 V (default)	8m
HEX3[1]	Output	PIN_U21	5	B5_N0	PIN_U21	2.5 V (default)	8m

HEX3[0]	Output	PIN_V21	5	B5_N1	PIN_V21	2.5 V (default)	
HEX4[6]	Output	PIN_AE18	4	B4_N2	PIN_AE18	2.5 V (default)	
HEX4[5]	Output	PIN_AF19	4	B4_N1	PIN_AF19	2.5 V (default)	
HEX4[4]	Output	PIN_AE19	4	B4_N1	PIN_AE19	2.5 V (default)	
HEX4[3]	Output	PIN_AH21	4	B4_N2	PIN_AH21	2.5 V (default)	8m
HEX4[2]	Output	PIN_AG21	4	B4_N2	PIN_AG21	2.5 V (default)	8m
HEX4[1]	Output	PIN_AA19	4	B4_N0	PIN_AA19	2.5 V (default)	8m
HEX4[0]	Output	PIN_AB19	4	B4_N0	PIN_AB19	2.5 V (default)	8m
HEX5[6]	Output	PIN_AH18	4	B4_N2	PIN_AH18	2.5 V (default)	8m
HEX5[5]	Output	PIN_AF18	4	B4_N1	PIN_AF18	2.5 V (default)	8m
HEX5[4]	Output	PIN_AG19	4	B4_N2	PIN_AG19	2.5 V (default)	8m
HEX5[3]	Output	PIN_AH19	4	B4_N2	PIN_AH19	2.5 V (default)	8m
HEX5[2]	Output	PIN_AB18	4	B4_N0	PIN_AB18	2.5 V (default)	8m
HEX5[1]	Output	PIN_AC18	4	B4_N1	PIN_AC18	2.5 V (default)	8m
HEX5[0]	Output	PIN_AD18	4	B4_N1	PIN_AD18	2.5 V (default)	8m
HEX6[6]	Output	PIN_AC17	4	B4_N2	PIN_AC17	2.5 V (default)	8m
HEX6[5]	Output	PIN_AA15	4	B4_N2	PIN_AA15	2.5 V (default)	8m
HEX6[4]	Output	PIN_AB15	4	B4_N2	PIN_AB15	2.5 V (default)	8m
HEX6[3]	Output	PIN_AB17	4	B4_N1	PIN_AB17	2.5 V (default)	8m
HEX6[2]	Output	PIN_AA16	4	B4_N2	PIN_AA16	2.5 V (default)	8m
HEX6[1]	Output	PIN_AB16	4	B4_N2	PIN_AB16	2.5 V (default)	8m
HEX6[0]	Output	PIN_AA17	4	B4_N1	PIN_AA17	2.5 V (default)	8m
HEX7[6]	Output	PIN_AA14	3	B3_N0	PIN_AA14	2.5 V (default)	8m
HEX7[5]	Output	PIN_AG18	4	B4_N2	PIN_AG18	2.5 V (default)	8m
HEX7[4]	Output	PIN_AF17	4	B4_N2	PIN_AF17	2.5 V (default)	8m
HEX7[3]	Output	PIN_AH17	4	B4_N2	PIN_AH17	2.5 V (default)	8m
HEX7[2]	Output	PIN_AG17	4	B4_N2	PIN_AG17	2.5 V (default)	8m

6) 关于 TIMEQUEST 实在是看不懂，没办法使用，所以暂时放弃。



五、实验代码

```
/**
```

```
* @brief Drive digital tube.
```

```
* @param NULL.
```

```
* @note    null.  
  
* @retval OUTput is the IOs of decoder.  
  
*/
```

```
module decoder_TEST  
  
    (input[3:0] C,  
  
    output reg[6:0] HEX);  
  
    always @(C)  
  
    begin  
  
        case(C)  
  
            4'b0000 : HEX = 7'b1000000;  
  
            4'b0001 : HEX = 7'b1001111;  
  
            4'b0010 : HEX = 7'b0100100;  
  
            4'b0011 : HEX = 7'b0110000;  
  
            4'b0100 : HEX = 7'b0011001;  
  
            4'b0101 : HEX = 7'b0010010;  
  
            4'b0110 : HEX = 7'b0000010;  
  
            4'b0111 : HEX = 7'b1111000;  
  
            4'b1000 : HEX = 7'b0000000;  
  
            4'b1001 : HEX = 7'b0011000;  
  
            4'b1010 : HEX = 7'b1111110;  
  
            4'b1011 : HEX = 7'b1111101;  
  
            4'b1100 : HEX = 7'b1111011;  
  
            4'b1101 : HEX = 7'b1110111;  
  
            4'b1110 : HEX = 7'b1101111;  
  
            4'b1111 : HEX = 7'b1011111;  
  
            default: HEX = 7'b1111111;  
  
        endcase  
  
    end  
  
endmodule
```

```

/**
 * @brief Drive two digital tubes.
 * @param NULL.
 * @note null.
 * @retval OUTput is the IOs of decoder.
 */

module decoder2

    (input[7:0] C,

    output[6:0] HEX2,HEX1);

    decoder_TEST m1 (C[7:4],HEX2);

    decoder_TEST m2 (C[3:0],HEX1);

```

```
endmodule
```

```

/**
 * @brief D flip-flop.
 * @param NULL.
 * @note NULL.
 * @retval The output is the D of D flip-flop.
 */

```

```

module D_trigger_test

    (input Clk,R,D,EN,

    output reg Q);

    always @(posedge Clk or negedge R)

    begin

        if(!R) Q<=1'b0;

        else if(EN)Q<=D;

        // else Q<=1'b0;

    end

endmodule

```

```
/**
```

```

* @brief  Enter 16 signals and latch.

* @param  NULL.

* @note   NULL.

* @retval output is the latch signals.

*/

module D_trigger_HEX

    (input Clk,R,EN,

     input [15:0] D,

     output [15:0]Q);

    D_trigger_test m1 (Clk,R,D[0],EN,Q[0]);
    D_trigger_test m2 (Clk,R,D[1],EN,Q[1]);
    D_trigger_test m3 (Clk,R,D[2],EN,Q[2]);
    D_trigger_test m4 (Clk,R,D[3],EN,Q[3]);
    D_trigger_test m5 (Clk,R,D[4],EN,Q[4]);
    D_trigger_test m6 (Clk,R,D[5],EN,Q[5]);
    D_trigger_test m7 (Clk,R,D[6],EN,Q[6]);
    D_trigger_test m8 (Clk,R,D[7],EN,Q[7]);
    D_trigger_test m9 (Clk,R,D[8],EN,Q[8]);
    D_trigger_test m10 (Clk,R,D[9],EN,Q[9]);
    D_trigger_test m11 (Clk,R,D[10],EN,Q[10]);
    D_trigger_test m12 (Clk,R,D[11],EN,Q[11]);
    D_trigger_test m13 (Clk,R,D[12],EN,Q[12]);
    D_trigger_test m14 (Clk,R,D[13],EN,Q[13]);
    D_trigger_test m15 (Clk,R,D[14],EN,Q[14]);
    D_trigger_test m16 (Clk,R,D[15],EN,Q[15]);

endmodule

/**

* @brief  Switch one hex date from two.

* @param  NULL.

```



```

* @note    null.

* @retval OUTput is the hex.

*/

module switch_hex

    (input EN,

     input [15:0] IN1,IN2,

     output reg [15:0]Q);

    always @(EN)

    begin

    if(EN)

        begin Q=IN1;end

    else

        begin Q=IN2;end

    end

endmodule

/**

* @brief   Switch one  date from two.

* @param   NULL.

* @note    null.

* @retval OUTput is the one.

*/

module switch

    (input EN,

     input IN,

     output reg Q1,Q2);

    always @(EN)

    begin

    if(EN)

        begin Q1=IN;Q2=!IN;end

    else

```

```

        begin Q1=1'b0;Q2=1'b0; end

    end

endmodule

/**

* @brief  PART5.

* @param  NULL.

* @note   null.

* @retval NULL.

*/

module PART8

    (input[17:0] IN,

     input KEY0,KEY1,

     output [6:0] HEX7,HEX6,HEX5,HEX4,HEX3,HEX2,HEX1,HEX0,

     output cout);

    wire [7:0]OUT_A,OUT_B,OUT_C,OUT_D;

    wire[15:0]OUT1,OUT2,OUT,temp1,temp2,temp3;

    wire EN1,EN2;

    switch m8(IN[17],IN[16],EN1,EN2);

    D_trigger_HEX m1(KEY1,KEY0,EN1,IN[15:0],OUT1[15:0]);

    D_trigger_HEX m2(KEY1,KEY0,EN2,IN[15:0],OUT2[15:0]);

    switch_hex m3(IN[16],OUT1,OUT2,OUT);

    assign OUT_A = OUT1[15:8];

    assign OUT_B = OUT1[7:0];

    assign OUT_C = OUT2[15:8];

    assign OUT_D = OUT2[7:0];

```

```

mymult M1 (OUT_A,OUT_B,temp1);

mymult M2 (OUT_C,OUT_D,temp2);


myadd16 M3 (temp1,temp2,cout,temp3);


decoder2 m4 (OUT[15:8],HEX7,HEX6);

decoder2 m5 (OUT[7:0],HEX5,HEX4);

decoder2 m6 (temp3[15:8],HEX3,HEX2);

decoder2 m7 (temp3[7:0],HEX1,HEX0);

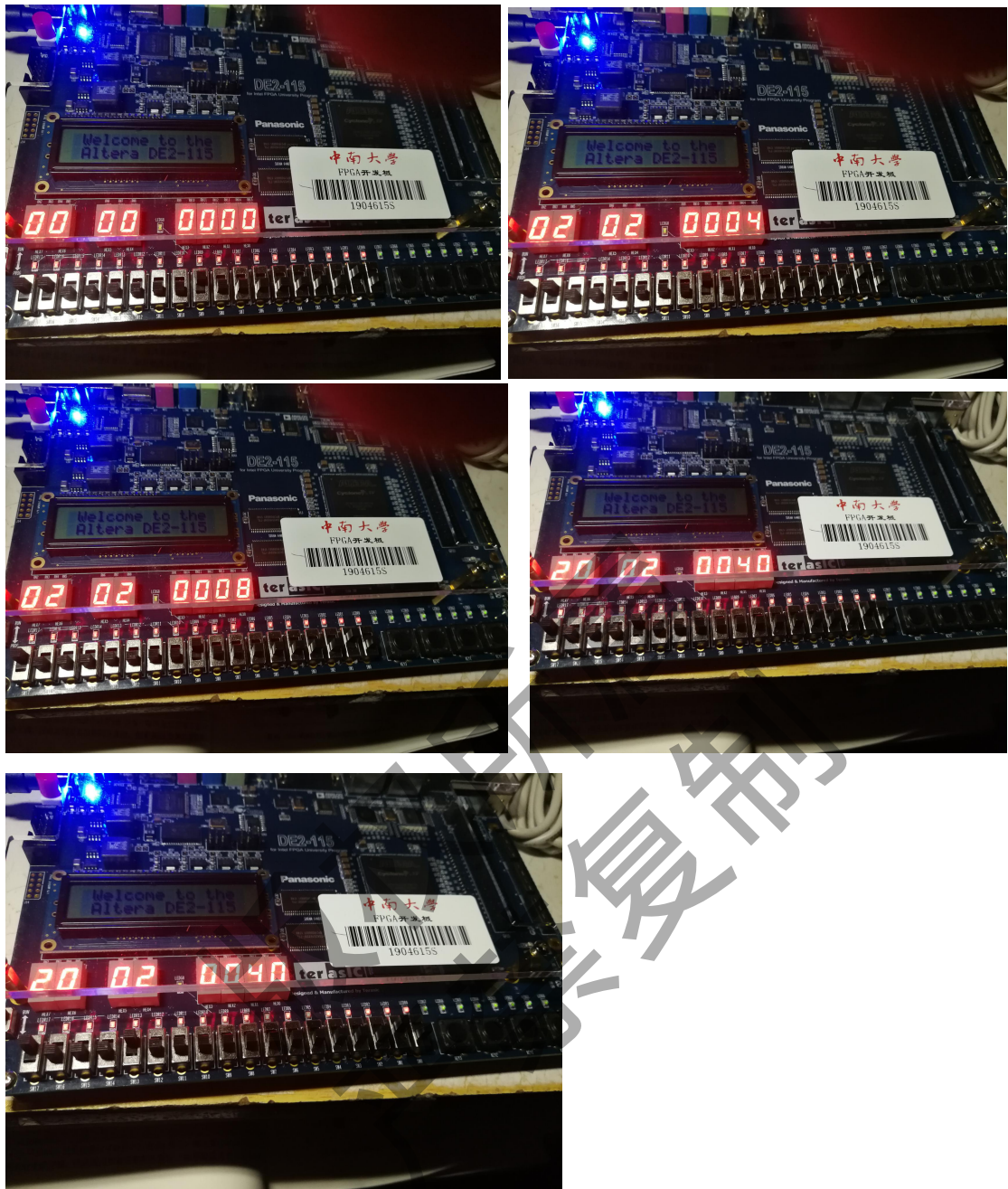
```

endmodule

Flow Summary	
Flow Status	Successful - Tue Nov 26 23:56:46 2019
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Full Version
Revision Name	lab4
Top-level Entity Name	lab4
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	106 / 114,480 (< 1 %)
Total combinational functions	90 / 114,480 (< 1 %)
Dedicated logic registers	32 / 114,480 (< 1 %)
Total registers	32
Total pins	76 / 529 (14 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	2 / 532 (< 1 %)
Total PLLs	0 / 4 (0 %)

六 实验体会

组合逻辑电路常常让人自闭。



第一部分 Part 9/Lab6

六、实验目的

- 1.在 part8 的基础上使用 ALTMULT_ADD 核进行电路优化。

七、实验内容

- 1.改变程序结构，优化使 PART8 高效。

八、实验原理

1. 以 17 个开关为输入，以 KEY1 为 CLK，KEY0 为 reset，使用两个寄存器，储存两个值，并且显示在前四个固定的数码管上。
2. 利用触发器可以暂时存储的特点，储存两次输入的值。
3. 将两次触发器暂存的值经过运算，输出给后四个数码管。

RTL 级仿真

九、实验步骤

6) 建立工程，并添加文件。

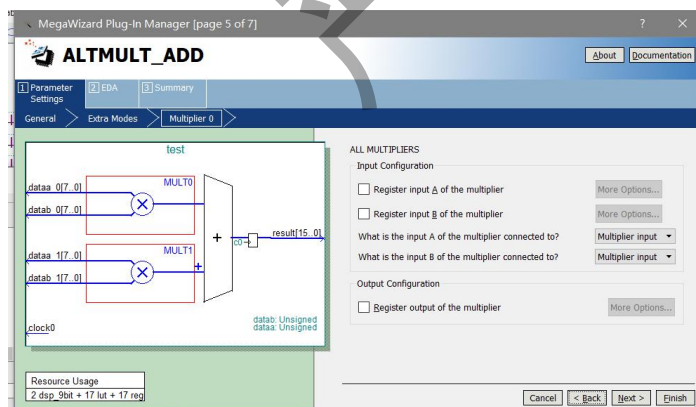
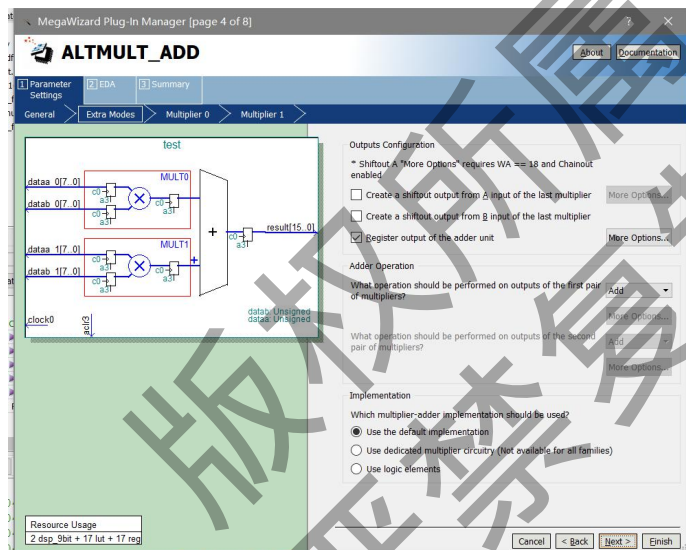
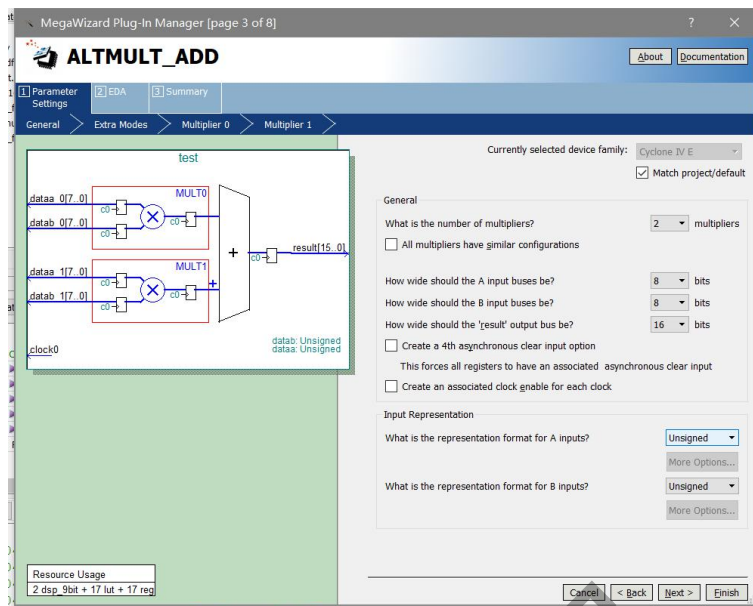
1. 添加 Verilog HDL 文件，取名为 PART9

7) 首先理清思路：

在本次实验中，首先通过 SW 和 KEY1,KEY0 对输入进行储存，之后将储存好的数据进行运算调用 IP 核生成加法器与乘法器的综合.v 文件，并在模块中调用，采用自上而下的程序写法。

8) 以 Verilog 代码表示。

9) 调用 IP 核的 altmult_add。



10) 编译并将引脚对应。

Node Name	Direction	Location	I/O Bank	VREF Group	Filter Location	I/O Standard	Reserved
HEX0[6]	Output	PIN_H22	6	B6_N0	PIN_H22	2.5 V (default)	
HEX0[5]	Output	PIN_J22	6	B6_N0	PIN_J22	2.5 V (default)	
HEX0[4]	Output	PIN_L25	6	B6_N1	PIN_L25	2.5 V (default)	
HEX0[3]	Output	PIN_L26	6	B6_N1	PIN_L26	2.5 V (default)	
HEX0[2]	Output	PIN_E17	7	B7_N2	PIN_E17	2.5 V (default)	8m
HEX0[1]	Output	PIN_F22	7	B7_N0	PIN_F22	2.5 V (default)	8m
HEX0[0]	Output	PIN_G18	7	B7_N2	PIN_G18	2.5 V (default)	8m
HEX1[6]	Output	PIN_U24	5	B5_N0	PIN_U24	2.5 V (default)	8m
HEX1[5]	Output	PIN_U23	5	B5_N1	PIN_U23	2.5 V (default)	8m
HEX1[4]	Output	PIN_W25	5	B5_N1	PIN_W25	2.5 V (default)	8m
HEX1[3]	Output	PIN_W22	5	B5_N0	PIN_W22	2.5 V (default)	8m
HEX1[2]	Output	PIN_W21	5	B5_N1	PIN_W21	2.5 V (default)	8m
HEX1[1]	Output	PIN_Y22	5	B5_N0	PIN_Y22	2.5 V (default)	8m
HEX1[0]	Output	PIN_M24	6	B6_N2	PIN_M24	2.5 V (default)	8m
HEX2[6]	Output	PIN_W28	5	B5_N1	PIN_W28	2.5 V (default)	8m
HEX2[5]	Output	PIN_W27	5	B5_N1	PIN_W27	2.5 V (default)	8m
HEX2[4]	Output	PIN_Y26	5	B5_N1	PIN_Y26	2.5 V (default)	8m
HEX2[3]	Output	PIN_W26	5	B5_N1	PIN_W26	2.5 V (default)	8m
HEX2[2]	Output	PIN_Y25	5	B5_N1	PIN_Y25	2.5 V (default)	8m
HEX2[1]	Output	PIN_AA26	5	B5_N1	PIN_AA26	2.5 V (default)	8m
HEX2[0]	Output	PIN_AA25	5	B5_N1	PIN_AA25	2.5 V (default)	8m
HEX3[6]	Output	PIN_Y19	4	B4_N0	PIN_Y19	2.5 V (default)	8m
HEX3[5]	Output	PIN_AF23	4	B4_N0	PIN_AF23	2.5 V (default)	8m
HEX3[4]	Output	PIN_AD24	4	B4_N0	PIN_AD24	2.5 V (default)	8m
HEX3[3]	Output	PIN_AA21	4	B4_N0	PIN_AA21	2.5 V (default)	8m
HEX3[2]	Output	PIN_AB20	4	B4_N0	PIN_AB20	2.5 V (default)	8m
HEX3[1]	Output	PIN_U21	5	B5_N0	PIN_U21	2.5 V (default)	8m

HEX3[0]	Output	PIN_V21	5	B5_N1	PIN_V21	2.5 V (default)	
HEX4[6]	Output	PIN_AE18	4	B4_N2	PIN_AE18	2.5 V (default)	
HEX4[5]	Output	PIN_AF19	4	B4_N1	PIN_AF19	2.5 V (default)	
HEX4[4]	Output	PIN_AE19	4	B4_N1	PIN_AE19	2.5 V (default)	
HEX4[3]	Output	PIN_AH21	4	B4_N2	PIN_AH21	2.5 V (default)	8m
HEX4[2]	Output	PIN_AG21	4	B4_N2	PIN_AG21	2.5 V (default)	8m
HEX4[1]	Output	PIN_AA19	4	B4_N0	PIN_AA19	2.5 V (default)	8m
HEX4[0]	Output	PIN_AB19	4	B4_N0	PIN_AB19	2.5 V (default)	8m
HEX5[6]	Output	PIN_AH18	4	B4_N2	PIN_AH18	2.5 V (default)	8m
HEX5[5]	Output	PIN_AF18	4	B4_N1	PIN_AF18	2.5 V (default)	8m
HEX5[4]	Output	PIN_AG19	4	B4_N2	PIN_AG19	2.5 V (default)	8m
HEX5[3]	Output	PIN_AH19	4	B4_N2	PIN_AH19	2.5 V (default)	8m
HEX5[2]	Output	PIN_AB18	4	B4_N0	PIN_AB18	2.5 V (default)	8m
HEX5[1]	Output	PIN_AC18	4	B4_N1	PIN_AC18	2.5 V (default)	8m
HEX5[0]	Output	PIN_AD18	4	B4_N1	PIN_AD18	2.5 V (default)	8m
HEX6[6]	Output	PIN_AC17	4	B4_N2	PIN_AC17	2.5 V (default)	8m
HEX6[5]	Output	PIN_AA15	4	B4_N2	PIN_AA15	2.5 V (default)	8m
HEX6[4]	Output	PIN_AB15	4	B4_N2	PIN_AB15	2.5 V (default)	8m
HEX6[3]	Output	PIN_AB17	4	B4_N1	PIN_AB17	2.5 V (default)	8m
HEX6[2]	Output	PIN_AA16	4	B4_N2	PIN_AA16	2.5 V (default)	8m
HEX6[1]	Output	PIN_AB16	4	B4_N2	PIN_AB16	2.5 V (default)	8m
HEX6[0]	Output	PIN_AA17	4	B4_N1	PIN_AA17	2.5 V (default)	8m
HEX7[6]	Output	PIN_AA14	3	B3_N0	PIN_AA14	2.5 V (default)	8m
HEX7[5]	Output	PIN_AG18	4	B4_N2	PIN_AG18	2.5 V (default)	8m
HEX7[4]	Output	PIN_AF17	4	B4_N2	PIN_AF17	2.5 V (default)	8m
HEX7[3]	Output	PIN_AH17	4	B4_N2	PIN_AH17	2.5 V (default)	8m
HEX7[2]	Output	PIN_AG17	4	B4_N2	PIN_AG17	2.5 V (default)	8m

十、实验代码

module PART9

```

(input[17:0] IN,

input KEY0,KEY1,

output [6:0] HEX7,HEX6,HEX5,HEX4,HEX3,HEX2,HEX1,HEX0,

output cout);

wire [7:0]OUT_A,OUT_B,OUT_C,OUT_D;

wire[15:0]OUT1,OUT2,OUT,temp1,temp2,temp3;

wire EN1,EN2;

switch m8(IN[17],IN[16],EN1,EN2);

D_trigger_HEX m1(KEY1,KEY0,EN1,IN[15:0],OUT1[15:0]);

D_trigger_HEX m2(KEY1,KEY0,EN2,IN[15:0],OUT2[15:0]);

switch_hex m3(IN[16],OUT1,OUT2,OUT);

```

```
assign OUT_A = OUT1[15:8];

assign OUT_B = OUT1[7:0];

assign OUT_C = OUT2[15:8];

assign OUT_D = OUT2[7:0];


/* mymult M1 (OUT_A,OUT_B,temp1);

mymult M2 (OUT_C,OUT_D,temp2);


myadd16 M3 (temp1,temp2,cout,temp3);*/

myaltnmultadd M1 (OUT_A,OUT_B,OUT_C,OUT_D,temp3);


decoder2 m4 (OUT[15:8],HEX7,HEX6);

decoder2 m5 (OUT[7:0],HEX5,HEX4);

decoder2 m6 (temp3[15:8],HEX3,HEX2);

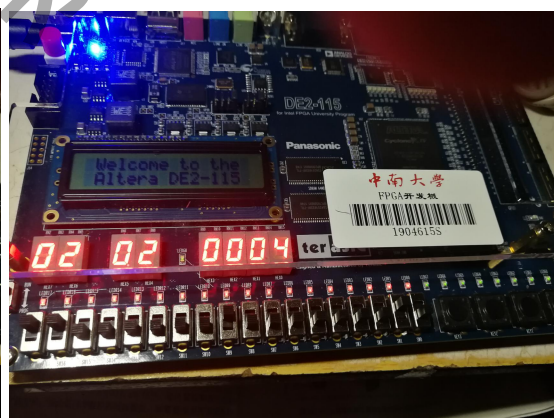
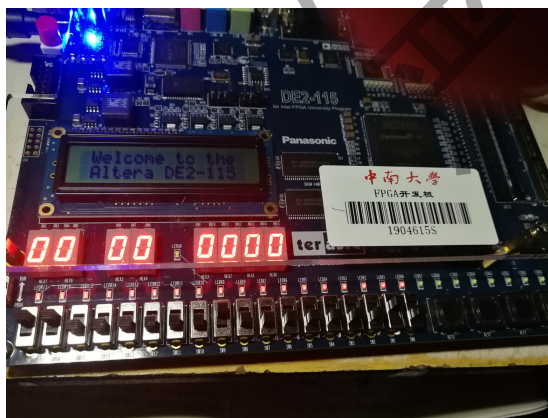
decoder2 m7 (temp3[7:0],HEX1,HEX0);

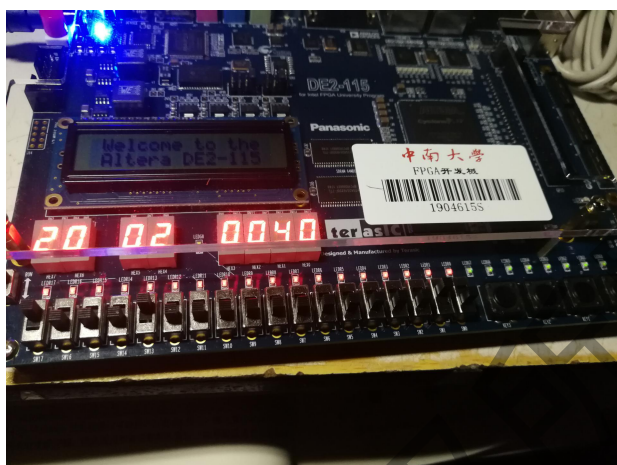
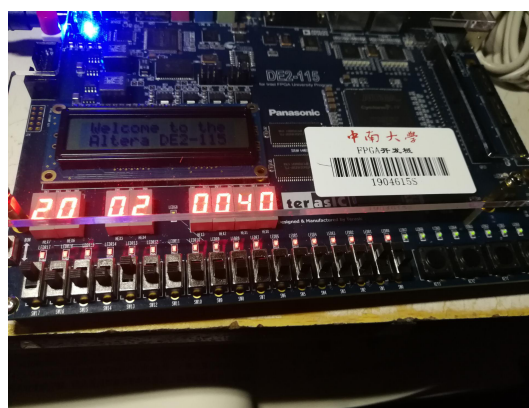
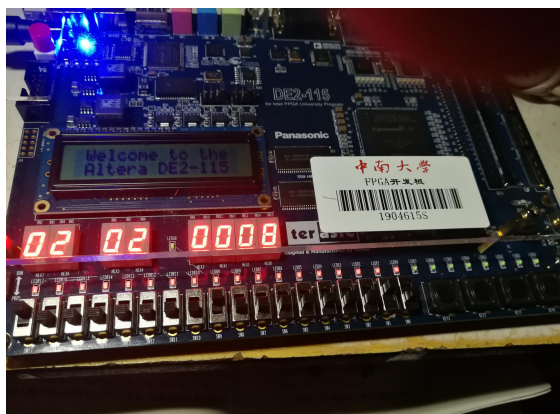
endmodule
```


Flow Summary	
Flow Status	Successful - Wed Nov 27 09:01:16 2019
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Full Version
Revision Name	lab4
Top-level Entity Name	lab4
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	62 / 114,480 (< 1 %)
Total combinational functions	46 / 114,480 (< 1 %)
Dedicated logic registers	32 / 114,480 (< 1 %)
Total registers	32
Total pins	76 / 529 (14 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

六 实验体会

调用 IP 核可以使复杂的内核变得简单，只要写好外部逻辑接口即可。





版权、
严禁复制