实验 6 存储器实验 第四次实验报告

22336313 郑鸿鑫 6 班

一． 实验内容

使用 vivado 上的 Block Memory Generator 模拟数据在存储器中的存取过程，初始化 Rom 中的内容通过开关选择相应的地址，将对应的存储器中内容读出来，并通过 7 段数码管显示。

二． 实验过程

根据实验手册创建存储器 IP，命名为 Ins_Rom，编写显示模块的代码如下：（其中关键部分已做注释说明）由于实验板上只有 4 个数码管，所以需要引入一个高位选择信号

```
module display_7seg(
    input CLK,//时钟信号
    input hi,//高位选择信号，用于选择显示高 4 位还是低 4 位
    input [31:0] data,
    output reg[10:0] display_out //7 段数码管显示信号
    );
    parameter T1MS=100000;
    integer count;
    reg [1:0] sel;//用于选择 4 个数码管的其中一个
    reg [3:0] hw;//用于读取 4 个位作为一个要输出的数字 0~f
    initial begin
        sel <= 0;
        count <= 0;//初始化为 0
    end
    always @(posedge CLK) begin
        if (count < T1MS) begin
            count <= count + 1;
        end else begin
            count <= 0;
            case(sel)
                0: display_out[10:7] <= 4'b0111;
                1: display_out[10:7] <= 4'b1110;
                2: display_out[10:7] <= 4'b1101;
                3: display_out[10:7] <= 4'b1011;
```

```
            endcase
            if(hi)
                case(sel)
                    0: hw[3:0] <= data[19:16];
                    1: hw[3:0] <= data[23:20];
                    2: hw[3:0] <= data[27:24];
                    3: hw[3:0] <= data[31:28];
                endcase
            else
                case(sel)
                    0: hw[3:0] <= data[3:0];
                    1: hw[3:0] <= data[7:4];
                    2: hw[3:0] <= data[11:8];
                    3: hw[3:0] <= data[15:12];
                endcase
            case(hw)
                0: display_out[6:0] <= 7'b0000001;
                1: display_out[6:0] <= 7'b1001111;
                2: display_out[6:0] <= 7'b0010010;
                3: display_out[6:0] <= 7'b0000110;
                4: display_out[6:0] <= 7'b1001100;
                5: display_out[6:0] <= 7'b0100100;
                6: display_out[6:0] <= 7'b0100000;
                7: display_out[6:0] <= 7'b0001111;
                8: display_out[6:0] <= 7'b0000000;
                9: display_out[6:0] <= 7'b0000100;
                10: display_out[6:0] <= 7'b0001000;
                11: display_out[6:0] <= 7'b1100000;
                12: display_out[6:0] <= 7'b0110001;
                13: display_out[6:0] <= 7'b1000010;
                14: display_out[6:0] <= 7'b0110000;
                15: display_out[6:0] <= 7'b0111000;//0~f 与 7 段
数码管对应关系，注意是低电平有效
            endcase;
            if (sel == 3)
                sel <= 0;
            else
                sel <= sel + 1;
        end
    end
endmodule
```

接下来编写 top 文件代码如下，用于整合模块

```
module top(
```

```
    input CLK,//时钟信号传入
    input hi,//高位信号传入
    input [7:0] addr,//8 位二进制地址用于选择存储器中对应的地址，由
SW0 到 SW7 手动输入
    output [10:0] display_out//7 段数码管显示输出
    );

    wire [31:0] data;

    Ins_Rom rom (
      .clka(CLK),    // input wire clka
      .addra(addr),  // input wire [7 : 0] addra
      .douta(data)  // output wire [31 : 0] douta
    );//实例化 Rom 模块

    display_7seg display (
        .CLK(CLK),
        .hi(hi),
        .data(data),
        .display_out(display_out)
    );//实例化显示模块
endmodule
```

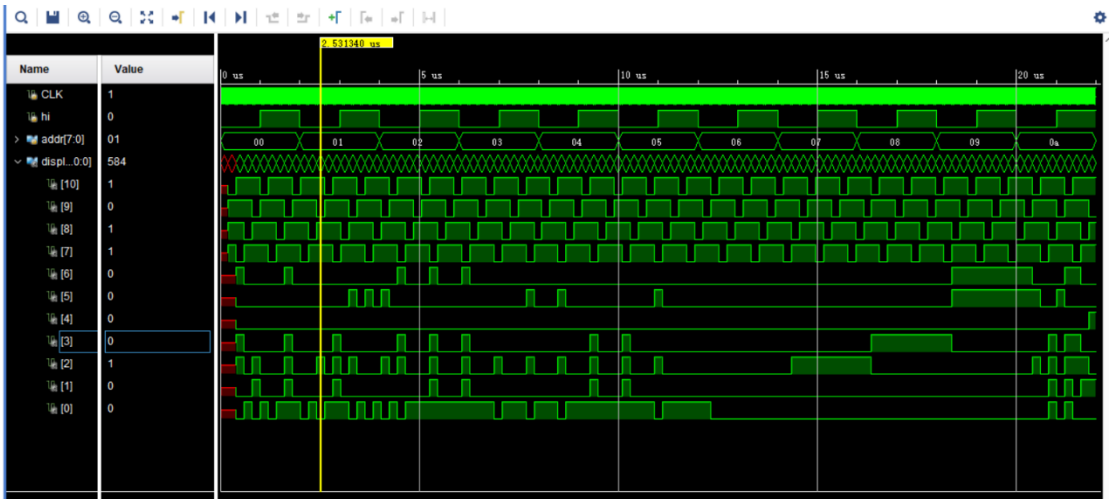
三． 实验结果分析

Rom 中的数据由.coe 文件给出，其中初始化数据如下：

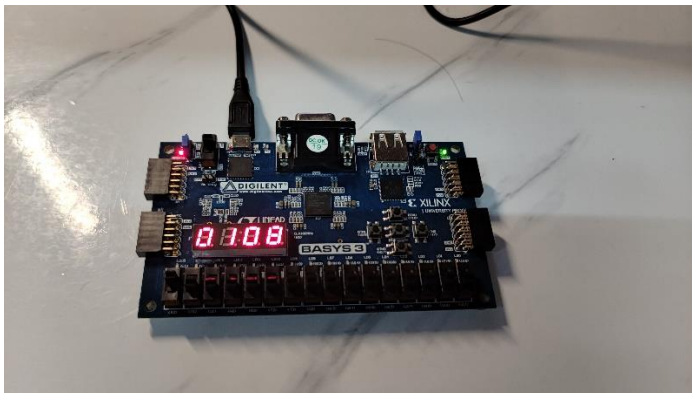
C: > Users > 26618 > Desktop > ROM > prgmip16.coe

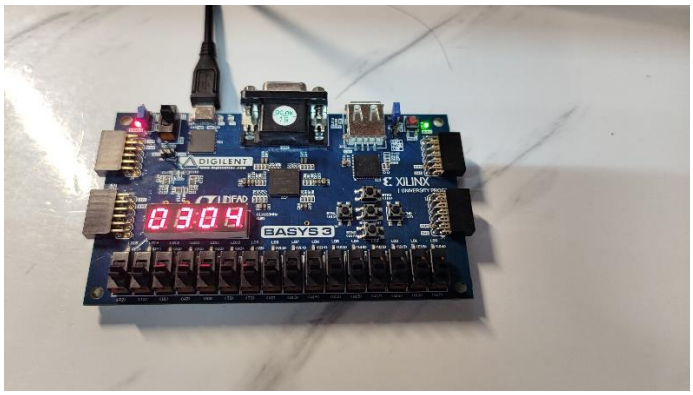
```
memory_initialization_radix=16;
memory_initialization_vector=
01080304,
05060907,
00010004,
00050009,
00070005,
00000005,
88888888,
99999999,
aaaaaaaa,
abbbbbbb,
12345678,
12abcdef;
```

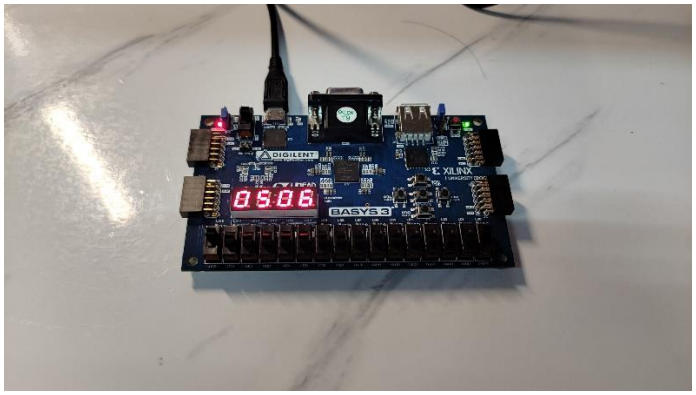
仿真过程如下，经检验符合预期：

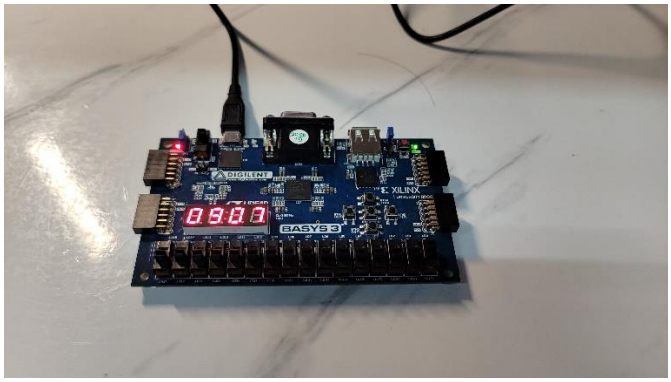


上板结果如下：（选取部分作为示例）









四． 实验总结

Block Memory Generator 是 Vivado 中的存储器模块 IP 核，可用于产生 RAM 和 ROM，ROM 核通过.coe 文件载入。

实验过程中遇到的问题：

在上板进行运行程序时，需注意 xdc 约束文件的编写，对应好端口和自己文件中的模块实输入输出端口的名称。