



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

电子信息虚拟仿真实验报告

实验 2: 存储器应用电路设计

学院 卓越学院

学号 23040447

姓名 陈文轩

专业 智能硬件与系统(电子信息工程)

2025 年 3 月 24 日

目录

1	顶层电路设计 (verilog)	1
2	仿真结果电路截图	2
3	远程平台操作截图	2
4	实验设计过程简要介绍	3
4.1	实验概述	3
4.2	模块设计	3
4.2.1	顶层模块 chengfaqi	3
4.2.2	主要功能组件	3
4.3	实验内容	4
4.3.1	4×4 乘法器设计与仿真验证	4
4.3.2	数码管显示实现	4
4.4	关键模块说明	5
4.4.1	时钟分频模块	5
4.4.2	乘法器模块	5
4.4.3	数码管控制	5
4.5	实验总结	6

1 顶层电路设计 (verilog)

```

1 module lab201MULMIF(
2     input sys_clk,           // 系统时钟输入
3     input [7:0] multi_input, // 乘法器输入信号, 8位宽
4
5     output [7:0] result,      // 乘法器输出结果, 8位宽
6     output [7:0] sg,         // 数码管段选信号, 8位宽
7     output [7:0] led         // LED显示信号, 8位宽
8 );
9
10 reg [2:0] q = 0;            // 3位计数器, 用于选择当前显示的数码管位
11 wire [31:0] num;            // 32位宽信号, 每4位存储一个十进制数字, 共8位
12 wire [3:0] y;               // 4位宽信号, 存储当前选中的数码管显示值
13
14 // 1Hz时钟生成模块
15 wire clk_1Hz;
16 FrequencyDivider #(.Prescaler(32'd24_999_999)) CLK_1Hz
17     (.rst_n(1'b1),          // 复位信号, 始终为高电平
18      .clk_input(sys_clk),    // 输入系统时钟
19      .clk_output(clk_1Hz));  // 输出1Hz时钟
20
21 // 1000Hz时钟生成模块
22 wire clk_1000Hz;
23 FrequencyDivider #(.Prescaler(32'd24_999)) CLK_1000Hz
24     (.rst_n(1'b1),          // 复位信号, 始终为高电平
25      .clk_input(sys_clk),    // 输入系统时钟
26      .clk_output(clk_1000Hz)); // 输出1000Hz时钟
27
28 // 1000Hz时钟驱动计数器, 用于轮动显示数码管位
29 always@(posedge clk_1000Hz)
30 begin
31     if(q == 3'b111)          // 如果计数器达到最大值(7), 则复位为0
32         q <= 3'd0;
33     else
34         q <= q + 1;          // 否则计数器加1
35 end
36
37 // 查找表乘法器模块
38 multi multi_inst (
39     .address(multi_input), // 输入地址信号
40     .inclock(sys_clk),     // 输入时钟信号
41     .q(result));           // 输出乘法结果
42
43 // 将乘法结果转换为8位十进制数字, 存储到num中
44 assign num = {4'b0000, 4'b0000, 4'b0000, 4'b0000, 4'b0000, 4'b0000,
45              result/100, result%100, result%10};
46
47 // 数码管位选信号生成模块
48 ecode ecode_inst(.cd(q), .sg(sg));
49
50 // 多路选择器模块, 根据q选择num中的4位数字, 输出到y
51 muxa muxa_inst(
52     .A(num[3:0]),           // 第1位数字
53     .B(num[7:4]),           // 第2位数字
54     .C(num[11:8]),          // 第3位数字
55     .D(num[15:12]),         // 第4位数字
56     .E(num[19:16]),         // 第5位数字
57     .F(num[23:20]),         // 第6位数字
58     .G(num[27:24]),         // 第7位数字
59     .H(num[31:28]),         // 第8位数字
60     .S(q),                  // 选择信号
61     .Y(y),                  // 输出选中的4位数字
62     .DP(led[7])             // 小数点信号
63 );
64
65 // BCD码转数码管段选信号模块
66 dec17s dec17s_inst(.A(y), .led7s(led[6:0]));
67
68 endmodule

```

图 1 顶层电路设计 (verilog)

2 仿真结果电路截图

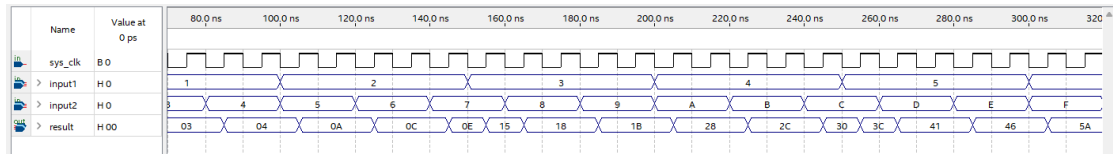


图 2 VWF 仿真结果

3 远程平台操作截图

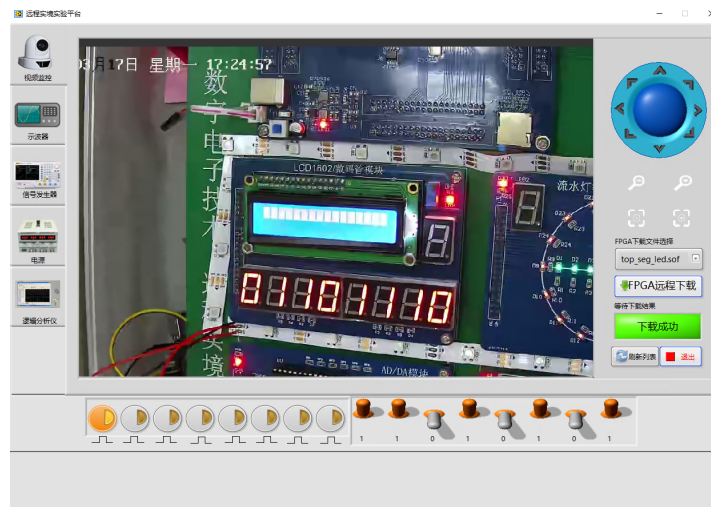


图 3 操作截图:4b1010*4'b0101=8b0110_1110

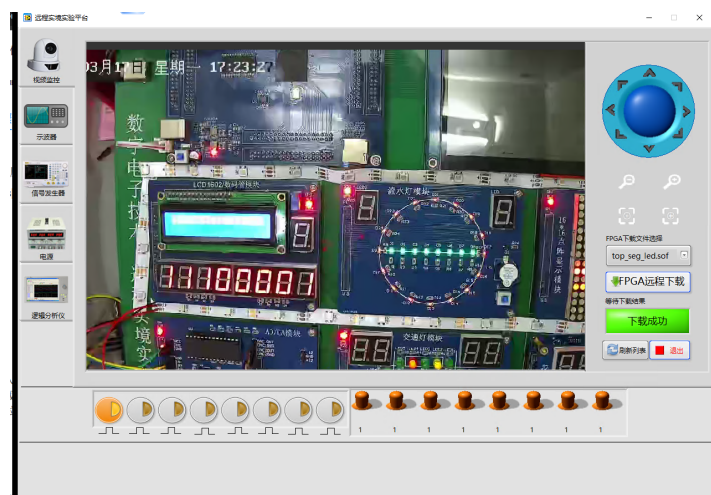


图 4 操作截图:4b1111*4b1111=8b1110_0001

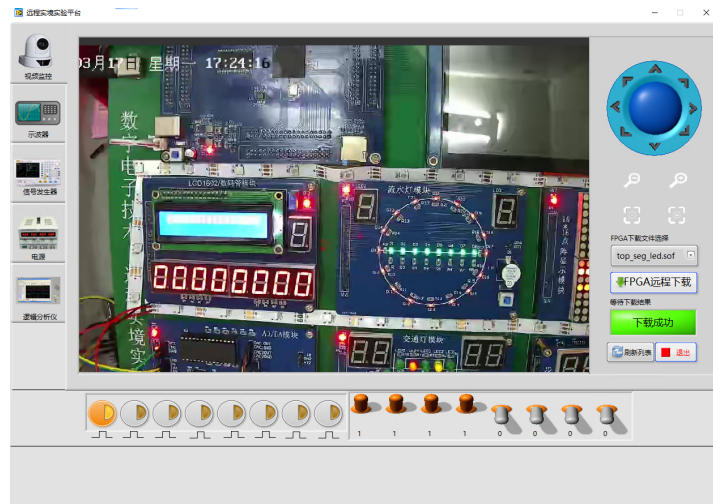


图 5 操作截图:4'b1111*4'b0000=8'd0

4 实验设计过程简要介绍

4.1 实验概述

本实验设计了一个 4×4 乘法器，并通过数码管显示模块展示乘法结果。实验分为两部分：仿真验证和硬件平台实现。

4.2 模块设计

4.2.1 顶层模块 chengfaqi

```
1 module chengfaqi(  
2     input sys_clk,           // 系统时钟输入  
3     input [7:0] multi_input, // 乘法器输入信号, 8位宽  
4     output [7:0] result,     // 乘法器输出结果, 8位宽  
5     output [7:0] sg,         // 数码管段选信号, 8位宽  
6     output [7:0] led         // LED显示信号, 8位宽  
7 );
```

4.2.2 主要功能组件

1. 时钟分频模块

- 生成 1Hz 和 1000Hz 时钟信号
- 用于控制数码管扫描频率

2. 查找表乘法器模块

- 实现 4×4 乘法运算
- 使用预存储的结果表提高运算速度

3. 数码管显示控制

- 8 位数码管动态扫描显示
- BCD 码转换与段选控制

4.3 实验内容

4.3.1 4×4 乘法器设计与仿真验证

乘法器采用查找表 (LUT) 方式实现, 将 4 位输入 (0-15) 的所有可能组合结果预先存储, 通过输入地址直接查找结果。

数学表达式:

$$\text{result} = A \times B$$

其中 A 和 B 各为 4 位输入, 组合为 8 位输入信号 `multi_input`。

仿真测试应覆盖所有可能的输入组合 (0×0 到 15×15), 验证乘法器功能正确性。典型测试向量包括:

- 边界值测试: 0×0, 15×15
- 随机组合测试: 如 7×8, 12×5 等
- 特殊组合测试: 如 10×10, 8×4 等

预期输出应符合乘法运算规则, 例如:

$$7 \times 8 = 56$$

$$15 \times 15 = 225$$

4.3.2 数码管显示实现

采用动态扫描方式显示 8 位十进制数字:

1. 数字分解:

$$\text{num} = \{\text{高位 0, 高位 0, 高位 0, 高位 0, 高位 0, 百位, 十位, 个位}\}$$

2. 扫描控制:

- 使用 3 位计数器 q 循环选择当前显示位 (0-7)
- 扫描频率为 1000Hz, 确保无闪烁

3. 段选信号生成:

- 通过 dec17s 模块将 BCD 码转换为 7 段显示码
- 小数点由 muxa 模块控制

对于乘法结果 R , 显示格式为:

$$\text{显示} = 00000 \parallel \left\lfloor \frac{R}{100} \right\rfloor \parallel \left\lfloor \frac{R\%100}{10} \right\rfloor \parallel R\%10$$

例如, 56 显示为"00000056", 225 显示为"00000225".

4.4 关键模块说明

4.4.1 时钟分频模块

```
1 FrequencyDivider #(.Prescaler(32'd24_999_999)) CLK_1Hz
2 FrequencyDivider #(.Prescaler(32'd24_999)) CLK_1000Hz
```

分频公式:

$$f_{out} = \frac{f_{in}}{2 \times (\text{Prescaler} + 1)}$$

4.4.2 乘法器模块

```
1 multi multi_inst (
2     .address(multi_input),
3     .inclock(sys_clk),
4     .q(result)
5 );
```

实现 $16 \times 16 = 256$ 种可能的乘法结果查找。

4.4.3 数码管控制

1. 位选控制:

```
1 ecode ecode_inst(.cd(q), .sg(sg));
```

2. 数字选择:

```
1 muxa muxa_inst(.A(num[3:0]), ..., .S(q), .Y(y));
```

3. 段码转换:

```
1 decl7s decl7s_inst(.A(y), .led7s(led[6:0]));
```

4.5 实验总结

本实验成功实现了:

1. 基于查找表的 4×4 乘法器设计
2. 乘法结果的数码管动态扫描显示

这种设计方法既保证了计算准确性，又实现了良好的显示效果。