

计算机组成原理实验报告

- 实验题目：数据通路与状态机
- 实验日期：2019年3月29日
- 姓名：张劲喲
- 学号：PB16111485
- 成绩：

实验目的：

1. 排序：s0 ~ s3是x0 ~ x3的排序结果
2. 除法运算：x / y = q ... r

实验设计简述与核心代码：

四数排序单元设计 (SORT.v)

```
1  module SORT(  
2      //=====   
3      input  [3:0] x0,  
4      input  [3:0] x1,  
5      input  [3:0] x2,  
6      input  [3:0] x3,  
7      input  rst,  
8      input  clk,  
9      //=====   
10     output reg [3:0] s0,  
11     output reg [3:0] s1,  
12     output reg [3:0] s2,  
13     output reg [3:0] s3,  
14     output reg done  
15     //=====   
16 );  
17 reg [1:0] cmp_count;  
18 // cmp_count  
19 always@(posedge clk or posedge rst) // 状态机, 3种状态  
20     begin  
21         if(rst)  
22             begin  
23                 cmp_count <= 2'b00;  
24                 done <= 0;
```

```

25         end
26     else if(cmp_count == 2'b10)
27     begin
28         cmp_count <= 2'b00;
29         done <= 1;
30     end
31     else
32     begin
33         cmp_count <= cmp_count + 2'b01;
34     end
35 end
36 //=====
37 // compare
38 always@(posedge clk or posedge rst)
39 begin
40     if(rst)
41     begin
42         {s0[3:0],s1[3:0],s2[3:0],s3[3:0]} <= {x0[3:0],x1[3:0],x2[3:0],x3[3:0]};
43     end
44     else
45     begin
46         case(cmp_count)
47             2'b00:                // 状态一, 1-2比较, 3-4比较
48             begin
49                 {s0[3:0], s1[3:0]} <= (s0[3:0] > s1[3:0]) ? {s0[3:0], s1[3:0]} : {s1[3:0], s0[3:0]};
50                 {s2[3:0], s3[3:0]} <= (s2[3:0] > s3[3:0]) ? {s2[3:0], s3[3:0]} : {s3[3:0], s2[3:0]};
51             end
52             2'b01:                // 状态二, 1-3比较, 2-4比较
53             begin
54                 {s0[3:0], s2[3:0]} <= (s0[3:0] > s2[3:0]) ? {s0[3:0], s2[3:0]} : {s2[3:0], s0[3:0]};
55                 {s1[3:0], s3[3:0]} <= (s1[3:0] > s3[3:0]) ? {s1[3:0], s3[3:0]} : {s3[3:0], s1[3:0]};
56             end
57             2'b10:                // 状态三, 2-3比较
58             begin
59                 {s1[3:0], s2[3:0]} <= (s1[3:0] > s2[3:0]) ? {s1[3:0], s2[3:0]} : {s2[3:0], s1[3:0]};
60             end
61         endcase
62     end
63 end
64 //=====
65 endmodule

```

除法器设计(DIV.v)

```

1 module DIV(
2     input[3:0] x,
3     input[3:0] y,
4     input rst,
5     input clk,

```

```

6     input input_confer,
7     output reg [3:0] q,
8     output reg [3:0] r,
9     output reg error,
10    output reg done
11 );
12    reg [3:0] temp_x;
13    reg [3:0] temp_y;
14    reg [7:0] temp_x;
15    reg [7:0] temp_y;
16    reg [3:0] div_count;    // 状态计数
17    reg div_on_going;      // 运行标志, 对应结束标志done
18    //-----
19    //division vaild
20    always@(*)
21    begin
22        error = ~(|y);
23    end
24    //-----
25    //operator store
26    always@(posedge clk or posedge rst)
27    begin
28        if(rst)                begin    temp_x <= 4'b0;    temp_y <= 4'b0;    end
29        else if(input_confer)    begin    temp_x <= x;        temp_y <= y;        end
30        else                    begin    temp_x <= temp_x;    temp_y <= temp_y;    end
31    end
32    //-----
33    //division on going flag
34    always@(posedge clk or posedge rst)
35    begin
36        if(rst)
37        begin
38            div_on_going <= 1'b0;
39            done <= 1'b0;
40        end
41        else if(input_confer && div_on_going == 1'b0) // 状态机, 八种状态, 四个移位, 四个比较-减法, 交替
出现
42            div_on_going <= 1'b1;
43        else if(div_count == 4'd8)
44        begin
45            div_on_going <= 1'b0;
46            done <= 1'b1;
47        end
48        else
49            div_on_going <= div_on_going;
50    end
51    //-----
52    //division counter
53    always@(posedge clk or posedge rst)
54    begin
55        if(rst)                div_count <= 4'b0;

```

```

56         else if(div_on_going)    div_count <= div_count + 4'b1;
57     else
58         div_count <= 4'b0;
59     end
60 //-----
61 //division
62 always@(posedge clk or posedge rst)
63     begin
64         if(rst)
65             begin
66                 temp_x = 8'b0;
67                 temp_y = 8'b0;
68             end
69         else if(div_on_going)
70             begin
71                 if(div_count == 4'b0)                // 四个移位状态
72                     begin
73                         temp_x <= {4'b0,temp_x};
74                         temp_y <= {temp_y,4'b0};
75                     end
76                 else if(div_count[0] == 1'b1)        // 四个比较-减法状态
77                     temp_x <= {temp_x[6:0],1'b0};
78                 else
79                     temp_x <= (temp_x[7:4] >= temp_y[7:4]) ? (temp_x - temp_y + 1) : temp_x;
80                 end
81             end
82         else
83             begin
84                 temp_x = 8'b0;
85                 temp_y = 8'b0;
86             end
87         end
88 //-----
89 //result output
90 always@(posedge done or posedge clk or posedge rst)
91     begin
92         if(rst)
93             begin
94                 q = 4'b0;
95                 r = 4'b0;
96             end
97         else if(done)
98             begin
99                 q = temp_x[3:0];
100                 r = temp_x[7:4];
101             end
102         else
103             begin
104                 q = 4'b0;
105                 r = 4'b0;
106             end
107         end
108     end
109 //-----

```

实验结果：

现场烧录检查：已通过

实现资源消耗与性能统计：

SORT:

Utilization

Post-Synthesis | Post-Implementation

Graph | Table

Resource	Utilization	Available	Utilization %
LUT	117	63400	0.18
FF	35	126800	0.03
IO	35	210	16.67
BUFG	1	32	3.13

Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:

41.487 W (Junction temp exceeded!)

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

125.0°C

Thermal Margin:

-129.3°C (-27.8 W)

Effective θ JA:

4.6°C/W

Power supplied to off-chip devices:

0 W

Confidence level:

Low

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power

98%

Dynamic: 40.690 W (98%)

94%

Signals: 1.278 W (3%)

Logic: 1.300 W (3%)

I/O: 38.112 W (94%)

Device Static: 0.797 W (2%)

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	inf	Worst Hold Slack (WHS):	inf	Worst Pulse Width Slack (WPWS):	NA
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	NA
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	NA
Total Number of Endpoints:	119	Total Number of Endpoints:	119	Total Number of Endpoints:	NA

There are no user specified timing constraints.

DIV:

Utilization

Post-Synthesis | Post-Implementation

Graph | Table

Resource	Utilization	Available	Utilization %
LUT	50	63400	0.08
FF	34	126800	0.03
IO	21	210	10.00
BUFG	1	32	3.13

Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:

1.237 W

Design Power Budget:

Not Specified

Power Budget Margin:

N/A

Junction Temperature:

30.6°C

Thermal Margin:

54.4°C (11.8 W)

Effective θJA:

4.6°C/W

Power supplied to off-chip devices:

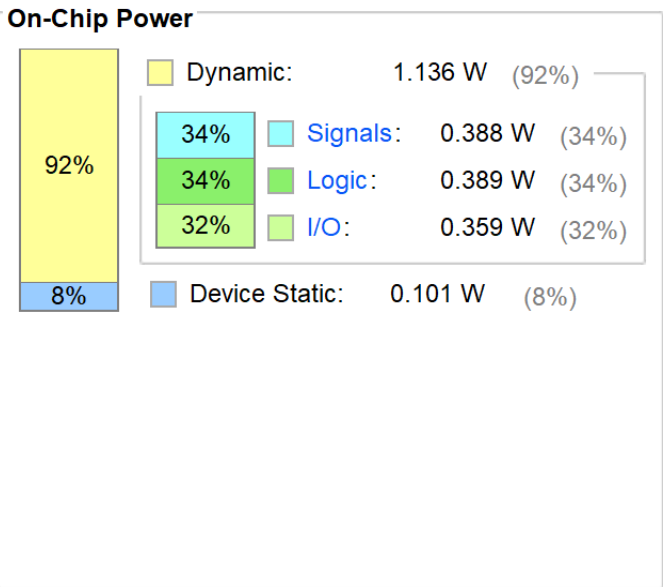
0 W

Confidence level:

Low

Launch Power Constraint Advisor

to find and fix invalid switching activity



Setup	Hold	Pulse Width
Worst Negative Slack (WNS): inf	Worst Hold Slack (WHS): inf	Worst Pulse Width Slack (WPWS): NA
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): NA
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: NA
Total Number of Endpoints: 98	Total Number of Endpoints: 98	Total Number of Endpoints: NA

There are no user specified timing constraints.

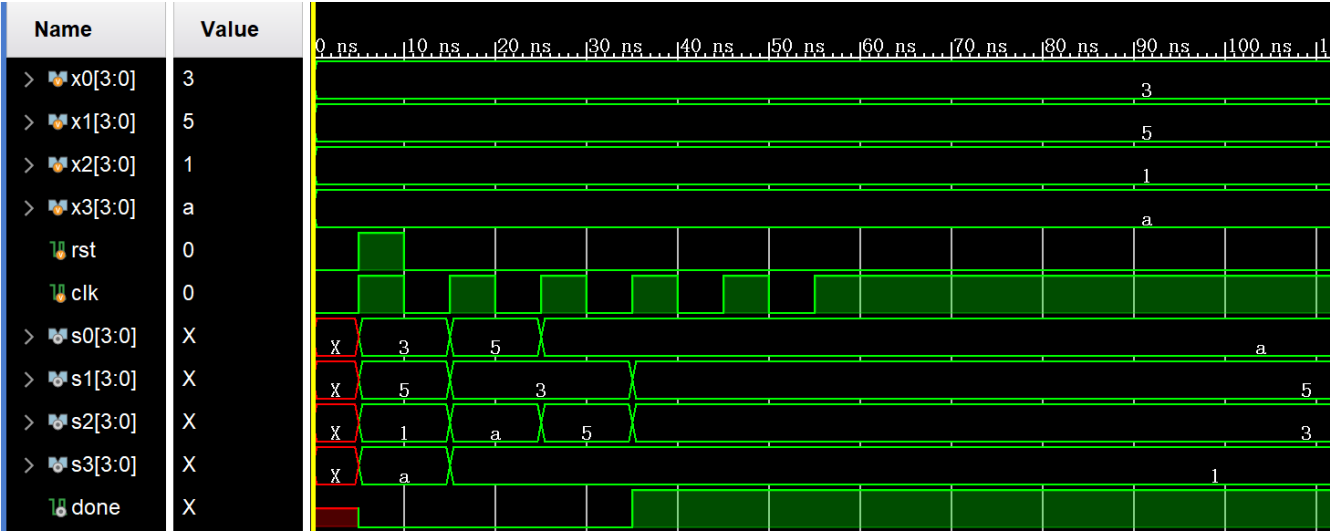
仿真测试结果：

SORT仿真 (test_sort.v)

```
1 module test_sort(  
2  
3     );  
4     reg [3:0] x0,x1,x2,x3;  
5     reg rst,clk;  
6     wire [3:0] s0,s1,s2,s3;  
7     wire done;  
8     SORT  
s(.x0(x0),.x1(x1),.x2(x2),.x3(x3),.rst(rst),.clk(clk),.s0(s0),.s1(s1),.s2(s2),.s3(s3),.done(done));  
9     initial  
10         begin  
11             x0 = 4'd3;  
12             x1 = 4'd5;  
13             x2 = 4'd1;  
14             x3 = 4'd10;  
15             rst = 0;  
16             clk = 0;  
17             #5  
18             rst = 1;  
19             clk = 1;  
20             #5  
21             rst = 0;  
22             clk = 0;  
23             #5  
24             clk = 1;  
25             #5  
26             clk = 0;  
27             #5  
28             clk = 1;  
29             #5  
30             clk = 0;  
31             #5  
32             clk = 1;  
33             #5  
34             clk = 0;  
35             #5  
36             clk = 1;  
37             #5  
38             clk = 0;
```

```
39         #5
40         clk = 1;
41     end
42 endmodule
43
```

结果正确:



DIV仿真 (test_div.v)

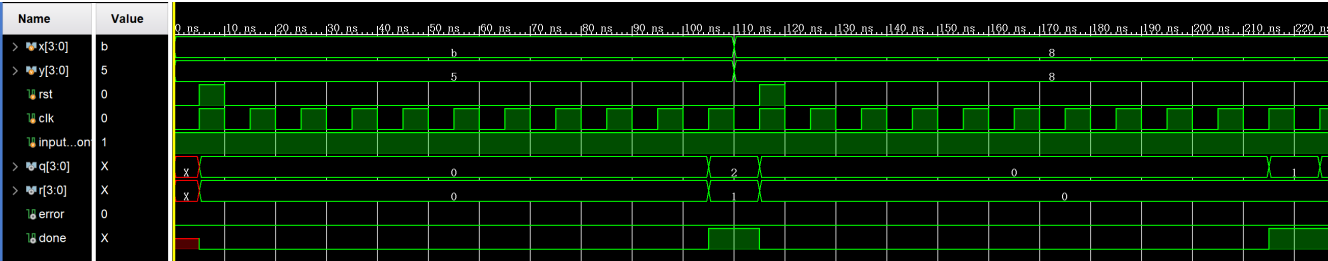
```
1 module test_div(
2
3     );
4     reg [3:0] x,y;
5     reg rst;
6     reg clk;
7     reg input_confer;
8     wire [3:0] q,r;
9     wire error, done;
10    DIV d(.x(x), .y(y), .rst(rst), .clk(clk), .input_confer(input_confer), .q(q), .r(r), .error(error),
11    .done(done));
12    initial
13        begin
14            x = 4'd11; // 测试 11 / 5 = 2 ... 1
15            y = 4'd5;
16            input_confer = 1;
17            clk = 0;
18            rst = 0;
19            #5
20            clk = 1;
21            rst = 1;
22            #5
23            clk = 0;
24            rst = 0;
25            #5
26            clk = 1;
```



```
26      #5
27      clk = 0;
28      #5
29      clk = 1;
30      #5
31      clk = 0;
32      #5
33      clk = 1;
34      #5
35      clk = 0;
36      #5
37      clk = 1;
38      #5
39      clk = 0;
40      #5
41      clk = 1;
42      #5
43      clk = 0;
44      #5
45      clk = 1;
46      #5
47      clk = 0;
48      #5
49      clk = 1;
50      #5
51      clk = 0;
52      #5
53      clk = 1;
54      #5
55      clk = 0;
56      #5
57      clk = 1;
58      #5
59      clk = 0;
60      #5
61      clk = 1;
62      #5
63      //=====
64      x = 4'd8;    // 测试 8 / 8 = 1 ... 0
65      y = 4'd8;
66      clk = 0;
67      rst = 0;
68      #5
69      clk = 1;
70      rst = 1;
71      #5
72      clk = 0;
73      rst = 0;
74      #5
75      clk = 1;
76      #5
```

```
77         clk = 0;
78         #5
79         clk = 1;
80         #5
81         clk = 0;
82         #5
83         clk = 1;
84         #5
85         clk = 0;
86         #5
87         clk = 1;
88         #5
89         clk = 0;
90         #5
91         clk = 1;
92         #5
93         clk = 0;
94         #5
95         clk = 1;
96         #5
97         clk = 0;
98         #5
99         clk = 1;
100        #5
101        clk = 0;
102        #5
103        clk = 1;
104        #5
105        clk = 0;
106        #5
107        clk = 1;
108        #5
109        clk = 0;
110        #5
111        clk = 1;
112        #5
113        clk = 0;
114        #5
115        clk = 1;
116        #5
117        clk = 0;
118    end
119 endmodule
```

结果正确：



实验总结与感想：

- 1. 通过实验了解了数据通路与状态机的设计实现，了解了数据通路与状态机的简单应用。
- 2. 复习了Verilog语法，提高了编程实践能力。