Verilog实验

专业：电子科学与技术
学号：171180571
姓名：吴康正

实验源码

限于篇幅原因，实验代码如果附在报告中则会过于臃肿，因此实验代码放在了实验报告最后的附录中，同时也可以在附件的对应目录中或者实验的Git repo中查看。

测试代码

测试代码基本上使用的提供代码，只有秒表的代码进行了一些小修改，增加了百分之一秒的计量以准确测试。

测试结果

对于提供验证文件的模块，测试通过。

对于秒表模块，测试输出（部分）如下：

```
#   out       =  0 0  5 9  9 5
#   out       =  0 0  5 9  9 6
#   out       =  0 0  5 9  9 7
#   out       =  0 0  5 9  9 8
#   out       =  0 0  5 9  9 9
#   out       =  0 1  0 0  0 0
#   out       =  0 1  0 0  0 1
#   out       =  0 1  0 0  0 2
#   out       =  0 1  0 0  0 3
#   out       =  0 1  0 0  0 4
#   out       =  0 1  0 0  0 5
#   out       =  0 1  0 0  0 6
#   out       =  0 1  0 0  0 7
#
# *** REACHED END OF TEST VECTORS ***
```

测试结果表明，模块设计正确，模块工作正常。

附录：源码

4位全加器

```
module adder4(
    output cout,
```

```
    output [3:0] sum.
    input [3:0] ina,
    input [3:0] inb,
    input cin
);

    assign {cout, sum} = ina + inb + cin;

endmodule
```

D触发器

```
module dff(
    output reg q,
    output reg qn,
    input d,
    input clk,
    input set,
    input reset
);

    always @ (set or reset)
    begin
        if (!reset)
        begin
            q = 0;
        end
        else
        begin
            if (!set)
            begin
                q = 1;
            end
        end
    end

    always @ (posedge clk)
    begin
        if (set && reset)
        begin
            q = d;
        end
    end

    always @ *
    begin
        qn = ~q;
    end

endmodule
```

4位计数器

```
```

```
module counter(
    output reg [7:0] out,
    input [7:0] data,
    input load,
    input reset,
    input clk
);

    always @ (posedge clk)
    begin
        if (!reset)
        begin
            out <= 8'b0;
        end
        else
        begin
            if (load)
            begin
                out <= data;
            end
            else
            begin
                out <= out + 1'b1;
            end
        end
    end

endmodule
```

4位串并转换器

```
module serial_pal(
    output reg [3:0] out,
    input clk,
    input reset,
    input en,
    input in
);

    always @ (posedge clk)
    begin
        if (reset)
        begin
            out <= 4'b0;
        end
        else
        begin
            if (en)
            begin
                out <= {out, in};
            end
        end
    end

endmodule
```

8-3优先编码器

```
module code_8_3(
    input [7:0] din,
    output reg [2:0] dout = 0
);

    always @ *
    begin
        casex(din)
            8'b00000001: dout <= 3'd0;
            8'b0000001x: dout <= 3'd1;
            8'b000001xx: dout <= 3'd2;
            8'b00001xxx: dout <= 3'd3;
            8'b0001xxxx: dout <= 3'd4;
            8'b001xxxxx: dout <= 3'd5;
            8'b01xxxxxx: dout <= 3'd6;
            8'b1xxxxxxx: dout <= 3'd7;
            default: dout <= 3'b0;
        endcase
    end

endmodule
```

七段数码管译码器

```
`define out a, b, c, d, e, f, g

module decoder47(
    output reg a,
    output reg b,
    output reg c,
    output reg d,
    output reg e,
    output reg f,
    output reg g,
    input [7:0] din
);

    always @ *
    begin
        case(din)
        // 测试文件中的数码管顺序与word文件中不一致，
        // 为了方便测试，以测试文件为准
        7'd0: {`out} <= 7'b111_1110;
        7'd1: {`out} <= 7'b011_0000;
        7'd2: {`out} <= 7'b110_1101;
        7'd3: {`out} <= 7'b111_1001;
        7'd4: {`out} <= 7'b011_0011;
        7'd5: {`out} <= 7'b101_1011;
        7'd6: {`out} <= 7'b101_1111;
        7'd7: {`out} <= 7'b111_0000;
        7'd8: {`out} <= 7'b111_1111;
        7'd9: {`out} <= 7'b111_1011;
```

```
        endcase
    end

endmodule
```

数字秒表

```
module stopwatch(
    output reg [3:0] msh,
    output reg [3:0] msl,
    output reg [3:0] sh,
    output reg [3:0] sl,
    output reg [3:0] mh,
    output reg [3:0] ml,
    input clk,
    input clr,
    input pause
);

    reg msla, msha, sla, sha, mla, mha;

    always @ (posedge clk or clr or pause)
    begin
        if (clr)
        begin
            msh <= 4'd0;
            msl <= 4'd0;
            sh <= 4'd0;
            sl <= 4'd0;
            mh <= 4'd0;
            ml <= 4'd0;
        end
        else
        begin
            if (!pause)
            begin
                if (msl >= 9)
                begin
                    msl <= 0;
                    msla <= 1;
                end
                else
                begin
                    msl <= msl + 1;
                end
            end
        end
    end

    always @ (posedge msla)
    begin
        if (msh >= 9)
        begin
            msh <= 0;
            msha <= 1;
        end
```

```
        else
        begin
            msh <= msh + 1;
        end
        msla <= 0;
    end

    always @ (posedge msha)
    begin
        if (sl >= 9)
        begin
            sl <= 0;
            sla <= 1;
        end
        else
        begin
            sl <= sl + 1;
        end
        msha <= 0;
    end

    always @ (posedge sla)
    begin
        if (sh >= 5)
        begin
            sh <= 0;
            sha <= 1;
        end
        else
        begin
            sh <= sh + 1;
        end
        sla <= 0;
    end

    always @ (posedge sha)
    begin
        if (ml >= 9)
        begin
            ml <= 0;
            mla <= 1;
        end
        else
        begin
            ml <= ml + 1;
        end
        sha <= 0;
    end

    always @ (posedge mla)
    begin
        if (mh >= 5)
        begin
            mh <= 0;
            mha <= 1;
        end
        else
        begin
```

```
            mh <= mh + 1;
        end
        mla <= 0;
    end

    always @ (posedge mha)
    begin
        msh <= 4'd0;
        msl <= 4'd0;
        sh <= 4'd0;
        sl <= 4'd0;
        mh <= 4'd0;
        ml <= 4'd0;
        mha <= 0;
    end

endmodule
```