数字逻辑与计算机组成实验

LAB 04：拓展实验 – 时钟

郑凯琳 205220025
205220025@smail.nju.edu.cn

（一）实验目的

在 Nexys A7-100 开发板上实现一个电子时钟，时钟要求能够显示时、分、 秒；还可以有以下功能：调整时间；闹铃（在特定时间 LED 闪烁）；秒表（提供百分之一秒精度，可以停止重启）等。

（二）实验环境/器材等

硬件器材：Nexys A7-100T 开发板
软件平台：Vivado 开发平台

（三）实验过程

数字抽象

1. 输入：
   - clk —— 时钟信号，与分频器的输出时钟信号连接
   - sel [1:0] —— 功能选择信号（时钟、设置时间、设置闹钟、秒表）
   - adj [2:0] —— 调整时间信号（时、分、秒）
   - stpwtch_en —— 秒表使能信号
   - stpwtch_rst —— 秒表清零信号
   - alm_stp —— 关闭闹钟
2. 输出：
   - alarm —— 闹钟信号，闪烁
   - AN [7:0] —— 七段 LED 数码管
   - hex [6:0] —— 数码管上的 LED

设计思路 & 设计代码

1. 由于时钟、闹钟和秒表是彼此独立的，故将这三组时间在模块内各以 integer 变量存储。另外，设 6 个数据输出的变量来存储：时、分、秒 —— 十位 + 个位。

```
integer h, m, s;                          //内部存储：小时、分钟、秒
integer h_alm, m_alm, s_alm;              //内部存储：闹钟对应的时间
integer h_stpwtch, m_stpwtch, s_stpwtch;  //内部存储：秒表对应的时间

reg [3:0] hr1;  //小时数输出：十位+个位
reg [3:0] hr0;
reg [3:0] min1; //分钟数输出：十位+个位
reg [3:0] min0;
reg [3:0] sec1; //秒数输出：十位+个位
reg [3:0] sec0;

initial
begin
    h = 0; m = 0; s = 0;
    h_alm = 24; m_alm = 0; s_alm = 0;
    h_stpwtch = 0; m_stpwtch = 0; s_stpwtch = 0;
end
```

2. 使用 case 语句来实现功能选择，且时钟的计时应该与其他功能相独立（即在 case 语句外，以免执行其他功能时影响到时钟计时）。

功能选择信号 sel [1:0]	调整时间信号 adj [2:0]	对应功能	备注
00	xxx	时钟	时间 = 闹钟设置的时间，启动闹钟
01	100	设置时间	修改时数
	010		修改分数
	001		修改秒数
10	100	设置闹钟	修改时数
	010		修改分数
	001		修改秒数
11	xxx	秒表	-

```
//时钟计时模块
if (h == 23 && m == 59 && s == 59) begin
    h <= 0; m <= 0; s <= 0;
end
else if (m == 59 && s == 59) begin
    h <= h + 1; m <= 0; s <= 0;
end
else if (s == 59) begin
    m <= m + 1; s <= 0;
end
else s <= s + 1;
```

```
0:
begin //正常计时
    hr1 <= h / 10; hr0 <= h % 10;
    min1 <= m / 10; min0 <= m % 10;
    sec1 <= s / 10; sec0 <= s % 10;

    //检查当前时间与闹钟设置时间是否一致，一致则alarm为1
    if (h == h_alm && m == m_alm && s == s_alm)
        alarm = 1;
    else
        alarm = alarm;
    //闹钟的关闭
    if (!alm_stp)
        alarm = 0;
    else
        alarm = alarm;
end
```

1:

```
begin //调整时间
    if(adj == 3'b001)
        if (s < 59)
            s <= s + 1;
        else
            s <= 0;
    if(adj == 3'b010)
        if (m  < 59)
            m <= m + 1;
        else
            m <= 0;
    if(adj == 3'b100)
        if (h < 23)
            h <= h + 1;
        else
            h <= 0;

    hr1 <= h / 10; hr0 <= h % 10;
    min1 <= m / 10; min0 <= m % 10;
    sec1 <= s / 10; sec0 <= s % 10;
end
```

2:

```
begin //闹钟
 if(adj == 3'b001)
        if (s_alm <= 59)
            s_alm <= s_alm + 1;
        else
            s_alm <= 0;
    if(adj == 3'b010)
        if (m_alm  <= 59)
            m_alm <= m_alm + 1;
        else
            m_alm <= 0;
    if(adj == 3'b100)
        if (h_alm <= 23)
            h_alm <= h_alm + 1;
        else
            h_alm <= 0;


    hr1 <= h_alm / 10; hr0 <= h_alm % 10;
    min1 <= m_alm / 10; min0 <= m_alm % 10;
    sec1 <= s_alm / 10; sec0 <= s_alm % 10;
end
```

3:

```
begin //秒表
    if (stpwtch_rst) begin //清零
        h_stpwtch <= 0; m_stpwtch <= 0; s_stpwtch <= 0;
    end
    else if (stpwtch_en) begin //计数
        if (h_stpwtch == 23 && m_stpwtch == 59 && s_stpwtch == 59) begin
            h_stpwtch <= 0; m_stpwtch <= 0; s_stpwtch <= 0;
        end
        else if (m_stpwtch == 59 && s_stpwtch == 59) begin
            h_stpwtch <= h_stpwtch + 1; m <= 0; s <= 0;
        end
        else if (s == 59) begin
            m_stpwtch <= m_stpwtch + 1; s_stpwtch <= 0;
        end
        else
            s_stpwtch <= s_stpwtch + 1;
    end
    else begin //暂停
        h_stpwtch <= h_stpwtch; m_stpwtch <= m_stpwtch; s_stpwtch <= s_stpwtch;
    end

    hr1 <= h_stpwtch / 10; hr0 <= h_stpwtch % 10;
    min1 <= m_stpwtch / 10; min0 <= m_stpwtch % 10;
    sec1 <= s_stpwtch / 10; sec0 <= s_stpwtch % 10;
end
```

3. 由于数字要在六个数码管上同时显示，分频及数码管显示的代码也需要注意和调整。

```
//分频
reg [31:0]count_clk2 = 0;
reg [2:0] clk_1ms = 0;
reg [2:0] temp = 0;
always @ (posedge clk)
begin
if(count_clk2 == 4999)
begin
    count_clk2 <= 0;
    temp <= temp + 3'b001;
    clk_1ms <= temp;
end
else
    count_clk2 <= count_clk2 + 1;

if(temp == 6)
    temp <= 3'b000;
end
```

测试代码

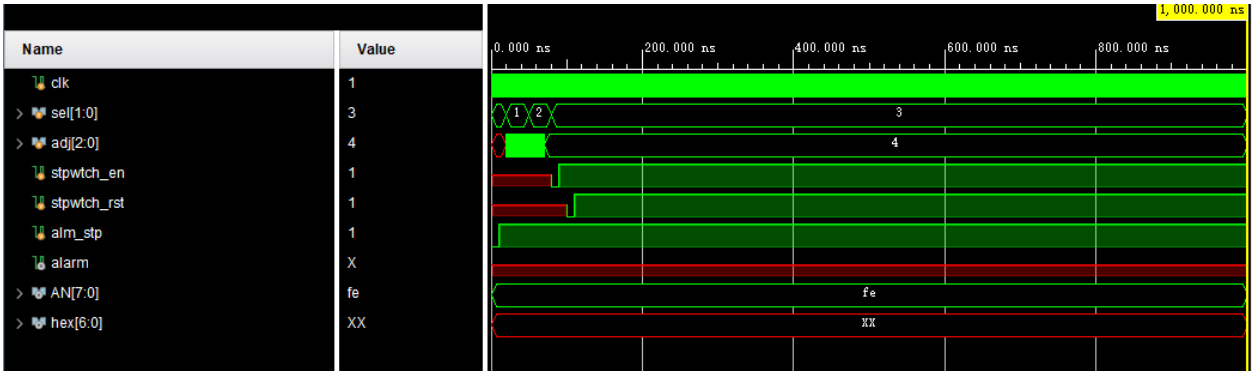
```
initial
begin
    clk = 0;
    sel = 2'b00; alm_stp = 0; #10;
                 alm_stp = 1; #10;
    sel = 2'b01; adj = 3'b001; #10;
                 adj = 3'b010; #10;
                 adj = 3'b100; #10;
    sel = 2'b10; adj = 3'b001; #10;
                 adj = 3'b010; #10;
                 adj = 3'b100; #10;
    sel = 2'b11; stpwtch_en = 0; #10; stpwtch_en = 1; #10;
                 stpwtch_rst = 0; #10; stpwtch_rst = 1; #10;
    $display("Running testbench");
end

always
begin
   clk = ~clk; #1;
end
```

硬件实现（引脚分配）

```
## Clock signal
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { clk }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];


##Switches
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { sel[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { sel[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
#set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { adj1[0] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
#set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { adj1[1] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
#set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { adj1[2] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { adj0[0] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
#set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports { adj0[1] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
#set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports { adj0[2] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
#set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS18 } [get_ports { adj0[3] }]; #IO_L24N_T3_34 Sch=sw[8]
#set_property -dict { PACKAGE_PIN U8    IOSTANDARD LVCMOS18 } [get_ports { SW[9] }]; #IO_25_34 Sch=sw[9]
set_property -dict { PACKAGE_PIN R16   IOSTANDARD LVCMOS33 } [get_ports { stpwtch_rst }]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
set_property -dict { PACKAGE_PIN T13   IOSTANDARD LVCMOS33 } [get_ports { stpwtch_en }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
set_property -dict { PACKAGE_PIN H6    IOSTANDARD LVCMOS33 } [get_ports { alm_stp }]; #IO_L24P_T3_35 Sch=sw[12]
set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports { adj[2] }]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports { adj[1] }]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports { adj[0] }]; #IO_L21P_T3_DQS_14 Sch=sw[15]

## LEDs
set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { alarm }]; #IO_L18P_T2_A24_15 Sch=led[0]
#set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { LED[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
#set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { LED[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
#set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports { LED[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
#set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports { LED[4] }]; #IO_L7P_T1_D09_14 Sch=led[4]
#set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { LED[5] }]; #IO_L18N_T2_A11_D27_14 Sch=led[5]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports { LED[6] }]; #IO_L17P_T2_A14_D30_14 Sch=led[6]
#set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { LED[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7]
#set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { LED[8] }]; #IO_L16N_T2_A15_D31_14 Sch=led[8]
#set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports { LED[9] }]; #IO_L14N_T2_SRCC_14 Sch=led[9]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports { LED[10] }]; #IO_L22P_T3_A05_D21_14 Sch=led[10]
#set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports { LED[11] }]; #IO_L15N_T2_DQS_DOUT_CSO_B_14 Sch=led[11]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports { LED[12] }]; #IO_L16P_T2_CSI_B_14 Sch=led[12]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports { LED[13] }]; #IO_L22N_T3_A04_D20_14 Sch=led[13]
#set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports { LED[14] }]; #IO_L20N_T3_A07_D23_14 Sch=led[14]
#set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports { LED[15] }]; #IO_L21N_T3_DQS_A06_D22_14 Sch=led[15]

##7 segment display
set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports { hex[0] }]; #IO_L24N_T3_A00_D16_14 Sch=ca
set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports { hex[1] }]; #IO_25_14 Sch=cb
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports { hex[2] }]; #IO_25_15 Sch=cc
set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports { hex[3] }]; #IO_L17P_T2_A26_15 Sch=cd
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports { hex[4] }]; #IO_L13P_T2_MRCC_14 Sch=ce
set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports { hex[5] }]; #IO_L19P_T3_A10_D26_14 Sch=cf
set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports { hex[6] }]; #IO_L4P_T0_D04_14 Sch=cg
#set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports { DP }]; #IO_L19N_T3_A21_VREF_15 Sch=dp
set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports { AN[0] }]; #IO_L23P_T3_FOE_B_15 Sch=an[0]
set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports { AN[1] }]; #IO_L23N_T3_FWE_B_15 Sch=an[1]
set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports { AN[2] }]; #IO_L24P_T3_A01_D17_14 Sch=an[2]
set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports { AN[3] }]; #IO_L19P_T3_A22_15 Sch=an[3]
set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 } [get_ports { AN[4] }]; #IO_L8N_T1_D12_14 Sch=an[4]
set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports { AN[5] }]; #IO_L14P_T2_SRCC_14 Sch=an[5]
set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVCMOS33 } [get_ports { AN[6] }]; #IO_L23P_T3_35 Sch=an[6]
set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 } [get_ports { AN[7] }]; #IO_L23N_T3_A02_D18_14 Sch=an[7]
```

（四）实验结果

仿真结果:


Name
Value
clk
1
sel[1:0]
3
adj[2:0]
4
stpwtch_en
1
stpwtch_rst
1
alm_stp
1
alarm
X
AN[7:0]
fe
hex[6:0]
XX
1,000.000 ns
0.000 ns
200.000 ns
400.000 ns
600.000 ns
800.000 ns
1
2
3
4
fe
XX