**“电子设计自动化(EDA)”实验报告(七)**

姓名 刘宗鑫 学号 201710413046 日期 2020.0619

**一、实验名称：**设计一个时分秒可调的数字钟

**二、实验内容和实验目的**

实验内容：

1．设计一个数字钟,用DE2开发板上的6个数码管(HEX5~HEX0)用来显示时、分、秒。Adjust 按键用来调整时、分、秒。Mode按键用来调整模式，模式的具体含义及数字钟功能要求如表1所示。外部晶振提供50MHz时钟，从PIN\_N2引脚送入FPGA芯片。因此，需要在FPGA内部设计分频器，把50MHz的时钟信号分频成1Hz的时钟供时、分、秒的计数器使用。

表1 数字钟功能

|  |  |
| --- | --- |
| Mode键 | 功能 |
| 按一下mode键 | 此时可调整“时”，每按一下adjust键，增加1小时，增加到23小时，再按一下adjust键，则变为00小时。调整过程中，数码管HEX5，HEX4闪烁，提示用户此时正在调整“小时”。 |
| 再按一下mode键 | 此时可调整“分”，每按一下adjust键，增加1分钟，增加到59分，再按一下adjust键，则变为00分。调整过程中，数码管HEX3，HEX2闪烁，提示用户此时正在调整“分钟”。 |
| 再按一下mode键 | 此时可调整“秒”，每按一下adjust键，增加1秒，增加到59秒，再按一下adjust键，则变为00秒。调整过程中，数码管HEX1，HEX0闪烁，提示用户此时正在调整“秒”。 |
| 再按一下mode键 | 回到正常计时和显示状态 |

实验目的：

深入理解硬件原理图，掌握FPGA硬件开发技能。学会分析和设计一个功能较复杂的综合电路，数字钟，带时分秒调整及显示。在此过程中进一步掌握并熟练使用VHDL的各种语句。

**三、实验实现方法(或设计思想)及实验结果**

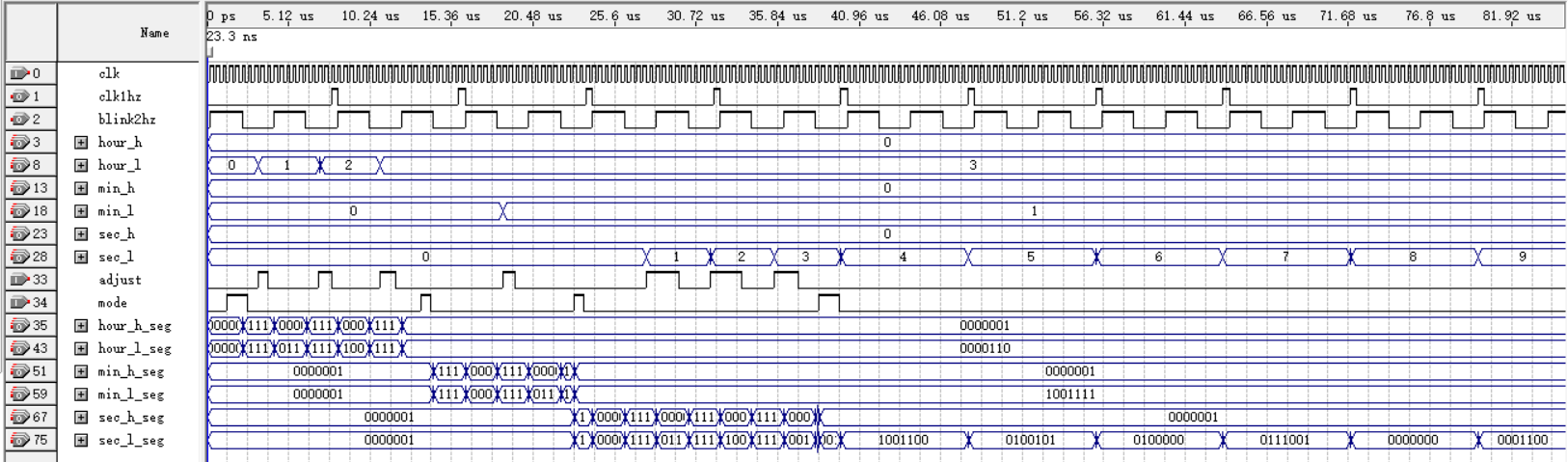
实现方法：

|  |  |
| --- | --- |
| Fdiv | library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  entity fdiv is      port(          clk: in std\_logic;          clk1hz, blink2hz: out std\_logic      );  end fdiv;  architecture arc of fdiv is      signal cnt : integer range 0 to 50;  begin      process(clk)      begin          if clk'event and clk = '1' then              cnt <= cnt + 1;              if 0 <= cnt and cnt <5 then                  blink2hz <= '1';              elsif 10 <= cnt and cnt < 15 then                  blink2hz <= '1';              else                  blink2hz <= '0';              end if;                if cnt = 19 then                  clk1hz <= '1';                  cnt <= 0;              else                  clk1hz <= '0';              end if;          end if;      end process;  end arc;  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  entity fdiv is      port(          clk: in std\_logic;          clk1hz, blink2hz: out std\_logic      );  end fdiv;  architecture arc of fdiv is      signal cnt : integer range 0 to 50;  begin      process(clk)      begin          if clk'event and clk = '1' then              cnt <= cnt + 1;              if 0 <= cnt and cnt <5 then                  blink2hz <= '1';              elsif 10 <= cnt and cnt < 15 then                  blink2hz <= '1';              else                  blink2hz <= '0';              end if;                if cnt = 19 then                  clk1hz <= '1';                  cnt <= 0;              else                  clk1hz <= '0';              end if;          end if;      end process;  end arc; |
| cunter24 | library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  entity counter24 is      port(          clk, en: in std\_logic;          cout: out std\_logic;          hh, hl: buffer std\_logic\_vector( 3 downto 0 )      );  end counter24;  architecture arc of counter24 is  begin      process( clk )      begin          if clk'event and clk = '1' then              if en = '1' then                  if hh = "0010" and hl = "0011" then                      hh <= "0000";                      hl <= "0000";                  else                      hl <= hl + "0001";                      if hl = "1001" then                          hl <= "0000";                          hh <= hh + "0001";                      end if;                  end if;              end if;          end if;      end process;      cout <= '1' when (hh = "0010" and hl = "0011") else '0';  end arc;  library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  entity counter24 is      port(          clk, en: in std\_logic;          cout: out std\_logic;          hh, hl: buffer std\_logic\_vector( 3 downto 0 )      );  end counter24;  architecture arc of counter24 is  begin      process( clk )      begin          if clk'event and clk = '1' then              if en = '1' then                  if hh = "0010" and hl = "0011" then                      hh <= "0000";                      hl <= "0000";                  else                      hl <= hl + "0001";                      if hl = "1001" then                          hl <= "0000";                          hh <= hh + "0001";                      end if;                  end if;              end if;          end if;      end process;      cout <= '1' when (hh = "0010" and hl = "0011") else '0';  end arc; |
| counter60 | library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  entity counter60 is      port(          clk, en: in std\_logic;          cout: out std\_logic;          hh, hl: buffer std\_logic\_vector( 3 downto 0 )      );  end counter60;  architecture arc of counter60 is  begin      process( clk )      begin          if clk'event and clk = '1' then              if en = '1' then                  if hh = "0101" and hl = "1001" then                      hh <= "0000";                      hl <= "0000";                  else                      hl <= hl + "0001";                      if hl = "1001" then                          hl <= "0000";                          hh <= hh + "0001";                      end if;                  end if;              end if;          end if;      end process;      cout <= '1' when (hh = "0101" and hl = "1001") else '0';  end arc; |
| segment4to7 | library ieee;  use ieee.std\_logic\_1164.all;  entity segment4to7 is  port(       data:in std\_logic\_vector( 3 downto 0 );       output :out std\_logic\_vector(6 downto 0));  end segment4to7;  architecture rtc of segment4to7 is  begin      process(data)      begin          output <= "1111111";          if data = "0000" then              output <= "0000001";          elsif data = "0001" then              output <= "1001111";          elsif data = "0010" then              output <= "0010010";          elsif data = "0011" then              output <= "0000110";          elsif data = "0100" then              output <= "1001100";          elsif data = "0101" then              output <= "0100101";          elsif data = "0110" then              output <= "0100000";          elsif data = "0111" then              output <= "0111001";          elsif data = "1000" then              output <= "0000000";          elsif data = "1001" then              output <= "0001100";          else              output <= "1111111";          end if;      end process;  end rtc; |
| mode\_adjust | library ieee;  use ieee.std\_logic\_1164.all;  use ieee.std\_logic\_unsigned.all;  entity mode\_adjust is      port(          clk, adjust, mode: in std\_logic;          clk1hz, blink2hz: buffer std\_logic;          min\_h, min\_l : buffer std\_logic\_vector( 3 downto 0 );          hour\_h, hour\_l : buffer std\_logic\_vector( 3 downto 0 );          sec\_h, sec\_l : buffer std\_logic\_vector( 3 downto 0 );          min\_h\_seg, min\_l\_seg : out std\_logic\_vector( 6 downto 0 );          hour\_h\_seg, hour\_l\_seg : out std\_logic\_vector( 6 downto 0 );          sec\_h\_seg, sec\_l\_seg : out std\_logic\_vector( 6 downto 0 )      );  end mode\_adjust;  architecture arc of mode\_adjust is      component fdiv      port(          clk: in std\_logic;          clk1hz, blink2hz: out std\_logic      );      end component;      component counter24          port(              clk, en: in std\_logic;              cout: out std\_logic;              hh, hl: buffer std\_logic\_vector( 3 downto 0 )          );      end component;      component counter60          port(              clk, en: in std\_logic;              cout: out std\_logic;              hh, hl: buffer std\_logic\_vector( 3 downto 0 )          );      end component;      component segment4to7          port(          data:in std\_logic\_vector( 3 downto 0 );          output :out std\_logic\_vector(6 downto 0));      end component;      signal en\_h, en\_m, ens, enh, enm, cout\_hour, cout\_min, cout\_sec: std\_logic;      signal min\_h\_out, min\_l\_out, hour\_h\_out, hour\_l\_out,sec\_h\_out, sec\_l\_out: std\_logic\_vector( 3 downto 0 );      signal blink\_min, blink\_hour, blink\_sec: std\_logic;      signal clk\_hour, clk\_min, clk\_sec: std\_logic;      type state\_type is ( timing, set\_hour, set\_min, set\_sec );      signal cur\_state: state\_type;  begin      clock: fdiv port map( clk, clk1hz, blink2hz );        process( mode )      begin          if mode'event and mode = '1' then              case cur\_state is                  when timing =>                      cur\_state <= set\_hour;                  when set\_hour =>                      cur\_state <= set\_min;                  when set\_min =>                      cur\_state <= set\_sec;                  when set\_sec =>                      cur\_state <= timing;              end case;          end if;      end process;        enh <= '1' when cur\_state = set\_hour else '0';      enm <= '1' when cur\_state = set\_min else '0';      ens <= '1' when cur\_state = set\_sec or cur\_state = timing else '0';        clk\_hour <= adjust when cur\_state = set\_hour else clk1hz;      clk\_min <= adjust when cur\_state = set\_min else clk1hz;      clk\_sec <= adjust when cur\_state = set\_sec else clk1hz;        hour: counter24 port map( clk\_hour, cout\_min or enh, cout\_hour, hour\_h, hour\_l );      min: counter60 port map( clk\_min, cout\_sec or enm, cout\_min, min\_h, min\_l );      sec: counter60 port map( clk\_sec, ens, cout\_sec, sec\_h, sec\_l);          blink\_hour <= '1' when cur\_state = set\_hour else '0';      blink\_min <= '1' when cur\_state = set\_min else '0';      blink\_sec <= '1' when cur\_state = set\_sec else '0';        hour\_h\_out <= "1111" when (blink2hz = '0' and blink\_hour = '1') else hour\_h;      hour\_l\_out <= "1111" when (blink2hz = '0' and blink\_hour = '1') else hour\_l;      min\_h\_out <= "1111" when (blink2hz = '0' and blink\_min = '1') else min\_h;      min\_l\_out <= "1111" when (blink2hz = '0' and blink\_min = '1') else min\_l;      sec\_h\_out <= "1111" when (blink2hz = '0' and blink\_sec = '1') else sec\_h;      sec\_l\_out <= "1111" when (blink2hz = '0' and blink\_sec = '1') else sec\_l;          hourhout: segment4to7 port map( hour\_h\_out, hour\_h\_seg );      hourlout: segment4to7 port map( hour\_l\_out, hour\_l\_seg );      minhout: segment4to7 port map( min\_h\_out, min\_h\_seg );      minlout: segment4to7 port map( min\_l\_out, min\_l\_seg );      sechout: segment4to7 port map( sec\_h\_out, sec\_h\_seg );      seclout: segment4to7 port map( sec\_l\_out, sec\_l\_seg );    end arc; |

设计思路：

1. 分频器
   1. 利用一个计数器，在计数满时clk1hz输出1；在相应区间时，blink2hz输出1，即可实现clk1hz，以及占空比为50%的2hz时钟
2. 计数器
   1. 分成十位和各位，按照计数器规则实现即可。
3. 4-7译码器
   1. 根据数字对应的编码进行输出
   2. 如果超过了0-9的范围则不进行显示。
4. 模式转换和顶层设计
   1. 利用状态机进行状态转换
   2. 根据状态设置信号和标志为
   3. 根据时钟和标志位控制信号的明与灭。

实验结果：



可见第一次位调整小时，第二次调整分，第三次调整秒，第四次之后为正常计时为1时显示数字，为0时灭掉。此外，当调整到相应的模式时，时钟变为adjust。

**四、总结**

**本次实验将多个模块整合到一起。联系了复杂系统的设计方式。先做出整体的系统框架，之后一块一块的进行实现，最后将单个模块综合在一起。**

**相比较连线的方式，使用component语句更加方便。**

（本次实验的简短总结或体会）