实验八　0809A/D 转换实验

一、 实验目的

1、 了解 ADC0809 的工作原理。

2、 了解用扫描方式驱动七段码管显示的工作原理。

3、 了解时序电路 FPGA 的实现。

4、 学习用 VHDL 语言来描述时序电路的过程。

二、 硬件要求

1、 可变时钟源。

2、 七段码显示。

3、 A/D 转换芯片 ADC0809

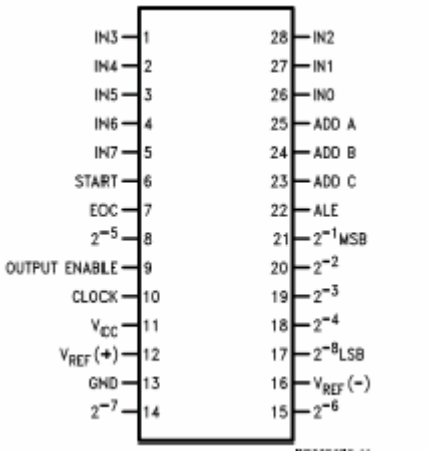
4、 主芯片 EP1K10TC100—3。

5、 三个拨动开关，进行地址选择。

三、 实验原理

该实验是利用 FPGA 控制 ADC0809 的时序，进行 AD 转换，然后将 ADC0809 转换后的数据以十六进制的数据显示出来。

ADC0809 是 8 位 8 通道的逐次比较式 AD 转换芯片。该芯片管脚如右图所示。芯片引脚及其说明如下：



D0－D7（2^{-8}－2^{-1}）：8 位双先三态数据线。

ADDA、ADDB、ADDC：通道选择地址。

OUTPUT ENABLE：输出允许控制。

Clock：ADC 转换时钟。

Vref＋、Vref－：正负参考电压。

IN0－IN7：8 个模拟信号输入通道。

START：AD 转换启动信号。

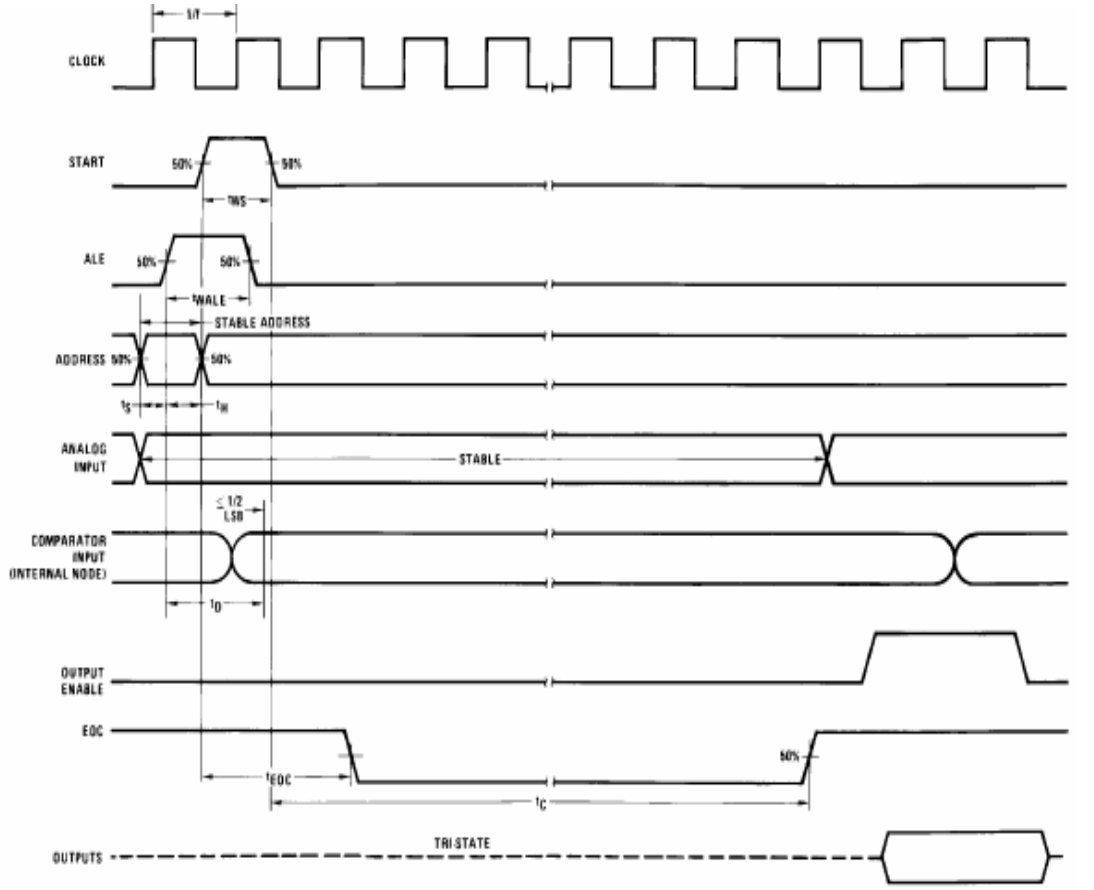
EOC：AD 转换结束信号。

ALE：通道地址锁存信号。

ADC0809 的工作时序如下图所示。其详细工作过程可查阅其他资料。

本实验 FPGA 实现时必须严格遵守 ADC0809 的工作时序，在编写其驱动代码时尤其要注意。ADC0809 的时钟信号从 FPGA 获取，FPGA 的时钟在 500KHz 至 800KHz 都可以选择。现具体介绍代码编写思想：

首先将要转换的 ADC0809 的地址输出，然后产生 ALE 信号的，在该信号的上升沿，地址被打入 ADC0809 的地址锁存器，这样就选中了对应的通道。地址产生结束后，便可产生 START 信号，使 ADC0809 开始进行 AD 转换，需要注意的是，在 ADC0809 转换期间，输入的模拟信号必须稳定，否则可能出现比较大的误差。在地址锁存并且启动转换后，EOC 便会呈现低电平，知道 AD 转换结束，所以 FPGA 在 EOC 从低电平变成高电平之前，不能读取 ADC 的转换数据。在 EOC 变成高电平之后，FPGA 便可将 OUTPUT INPUT 信号拉高，这样 ADC 转换的数据就会呈现在数据线上，FPGA 读入该数据后，在 8 位七段码管上显示出来，这就是整个实验过程的工作流程。



四、 实验内容及步骤

本实验的内容就是用 FPGA 模拟产生 ADC0809 的时序，使其正常工作，对 ADC0809 输入一个模拟量，进行 A/D 转换，然后将读入后的数据进行显示，实验步骤如下：

1、 编写 ADC0809 时序的 VHDL 代码。
2、 用 QuartusII 对其进行编译仿真。
3、 在时序确定无误后，选择芯片 ACEX1K EP1K10TC100－3。
4、 给芯片进行管脚绑定，在此进行编译。
5、 根据自己绑定的管脚，在实验箱上对 ADC0809、显示七段码和 FPGA 之间进行正确连线。
6、 对选定的通道输入一个模拟量,给目标板下载代码,调节电位器改变输入的模拟量，观看实验结果。

五、 实验连线

如果是调用的本书提供的 VHDL 代码，则实验连线如下：

Clk：时钟输入信号，500KHz 至 800KHz 之间均可。

EOC：输出信号，接 ADC0809 的 EOC 信号。

Din：数据输入，接 ADC0809 的数据总线 D0－D7；

Start：输出信号，接 ADC0809 的 START 信号。

Ale：地址锁存，接 ADC0809 的 ALE 信号。

OE： 输出允许，接 ADC0809 的 OUTPUT ENABLE 信号。

Sa、Sb、Sc：七段显示选通信号，接七段码显示 SEL0、SEL1 和 SEL2。

A、B、C、D、E、F、G：分别连接至七段码显示的 a、b、c、d、e、f、g。

ADC0809 的地址选择信号 A、B 和 C 分别到三个拨挡开关。

通过 A、B、C 选取输入通道 CH0～CH7 其中的相应通道输入一个可变模拟量。

六、 实验部分 VHDL 代码

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
-----------------------------------------------------------------
entity adc is
  port(  clk,eoc     :  in      std_logic;        --Clock Signal
         din         :  in      std_logic_vector(7 downto 0); --data bus
         clock,start  :  out     std_logic;       --clock of adc0809
         ale         :  out     std_logic;       --ale signal of adc0809
         oe          :  buffer  std_logic;       --out enable signal
         a,b,c,d,e,f,g    :  out     std_logic;       --7 segment driver
         sa,sb,sc       :  out     std_logic);      --Display Select
end adc;
-----------------------------------------------------------------
architecture behave of adc is
  signal  dcount        : std_logic_vector(2 downto 0);
  signal  adh,adl       : std_logic_vector(6 downto 0);
  signal  display       : std_logic_vector(6 downto 0);
  signal  adcount       : std_logic_vector(19 downto 0);
  signal  din_h,din_l   : std_logic_vector(3 downto 0);
  signal  disp_flag     : std_logic;
  begin
    process(clk)
      begin
        clock<=clk;
    end process;
    process(clk)   --accumulate adcount
      begin
        if(clk'event and clk='1') then
           adcount<=adcount+1;
        end if;
    end process;
    process(clk)  --start ad0809 convert
      begin
        if(clk'event and clk='1') then
           if(adcount=0) then
              ale<='1';
              start<='0';
           elsif(adcount=1) then
              ale<='1';
              start<='1';
           elsif(adcount=2) then
              ale<='0';
              start<='1';
           else
              ale<='0';
              start<='0';
           end if;
        end if;
    end process;
    process(clk)  --out enable signal
      begin
        if(clk'event and clk='1') then
           if(adcount=1000000 and eoc='1') then
              oe<='1';
           else
              oe<='0';
           end if;
        end if;
    end process;
    process(clk)  --rd the adc data
      begin
        if(clk'event and clk='1') then
           if(oe='1') then
              din_h<=din(7 downto 4);
              din_l<=din(3 downto 0);
              disp_flag<='1';
           else
```

```
            disp_flag<='0';
          end if;
       end if;
    end process;
    process(clk)
      begin
        if(clk'event and clk='1') then
           if(disp_flag='1') then
              case din_h is
                when "0000"=>adh<="0111111";  --display 0
                                ……
                when "1111"=>adh<="1110001";  --display f
                when others=>adh<=adh;         --no change
              end case;
              case din_l is
                when "0000"=>adl<="0111111";  --display 0
                              ……
                when "1111"=>adl<="1110001";  --display f
                when others=>adl<=adl;         --no change
              end case;
           end if;
       end if;
    end process;
    process(clk)  --display process
      begin
        if(clk'event and clk='1') then
           dcount<=dcount+1;
           sa<=dcount(0);
           sb<=dcount(1);
           sc<=dcount(2);
           case dcount is
             when "111"=>display<="1110111";  --display A
             when "000"=>display<="1011110";  --display d
             when "001"=>display<="0111001";  --display C
             when "010"=>display<="1000000";  --display -
             when "011"=>display<="1000000";  --display -
             when "100"=>display<=adh;
             when "101"=>display<=adl;
             when "110"=>display<="1110100";  --display h
             when others=>display<="0000000";
           end case;
         end if;
    end process;
    process(clk)   --In this process, a,b,c,d,e,f,g and dot will output
      begin
        if(clk'event and clk='1') then
           a<=display(0);
              ……
           g<=display(6);
        end if;
    end process;
end behave;
```

七、 实验报告要求

对于外部模拟信号 Vtest 范围超出 0～5V 的情况下，应如何修改设计和显示模块？

请学生思考：为什么引入 CLK 信号？用与不用 CLK 信号对显示可能产生什么影响？