实验报告

实验题目：七段数码管　　姓名：牟真伟　　学号：PB20051061

实验内容

设计一个8位7段数码管的显示电路，每个数码管对应一个锁存器，每个数码管显示的值为为相应锁存器中的值，每个锁存器中的值都可通过外部输入，从而改变数码管显示的值。

设计分析

用8个锁存器存储每位数码管的端选信号，采用一个8进制计数器同步位选信号和段码信号，8进制计数器的输出一方面接到一个38译码器，产生高电平有效的位选信号，另一方面接到一个8到1选择器，将此时位选信号对应的段选信号从寄存器送到数码管段选端，寄存器组输入端采用一个38译码器（低电平有效）进行寄存器的选择，一个使能控制端和一个数据输入端。为使数码管显示亮度不要太暗和减小电路时延对信号同步造成的影响，时钟信号应进行一定分频。

仿真结果记录

由于对整体做仿真较为复杂，仅仿真主要逻辑模块寄存器组

选择信号位sel，数据输入信号位datain，使能信号为en，各寄存器输出信号分别为d0-d7

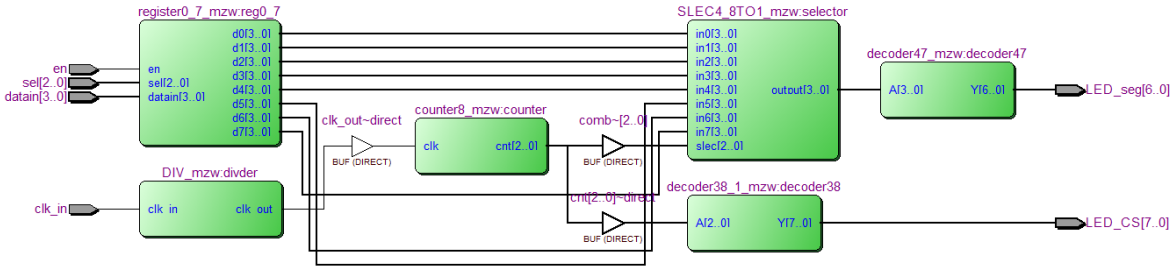
选择信号依次选择0到7号寄存器，数据输入信号依次为0-7，8-15，在每个选择信号后半段，使能信号为1，将当前数据送入相应存储器。因此d0到d7锁存器依次被置为0-7，8-15。

中间结果记录

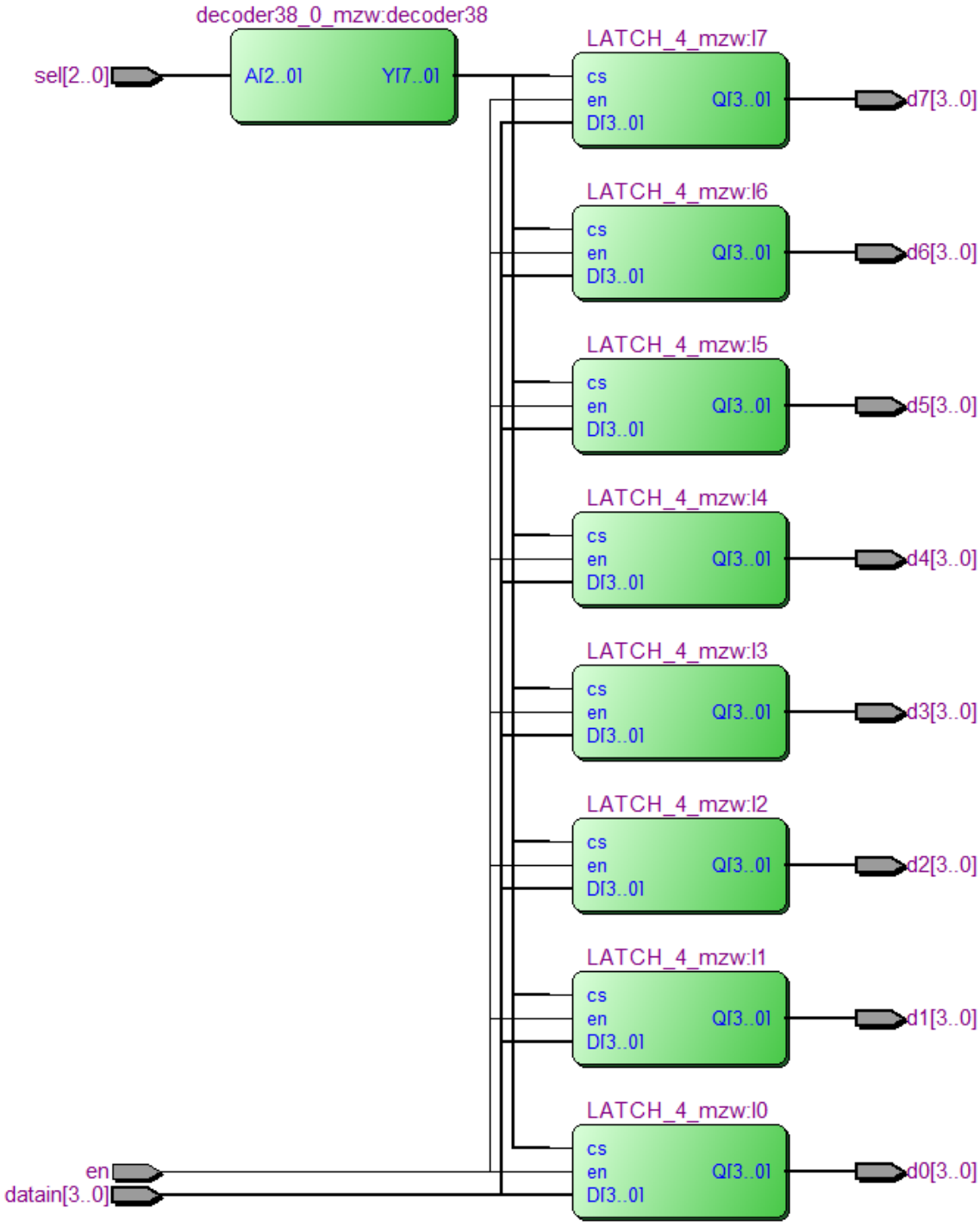
全编译后资源占用情况：

Flow Summary	
Flow Status	Successful - Thu Oct 20 13:38:47 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	FPGA_EXP4_mzw
Top-level Entity Name	FPGA_EXP4_mzw
Family	Cyclone V
Device	5CEFA2F23C8
Timing Models	Final
Logic utilization (in ALMs)	59 / 9,430 (< 1 %)
Total registers	20
Total pins	24 / 224 (11 %)
Total virtual pins	0
Total block memory bits	0 / 1,802,240 (0 %)
Total DSP Blocks	0 / 25 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI PMA RX ATT Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total HSSI PMA TX ATT Serializers	0
Total PLLs	0 / 4 (0 %)
Total DLLs	0 / 4 (0 %)

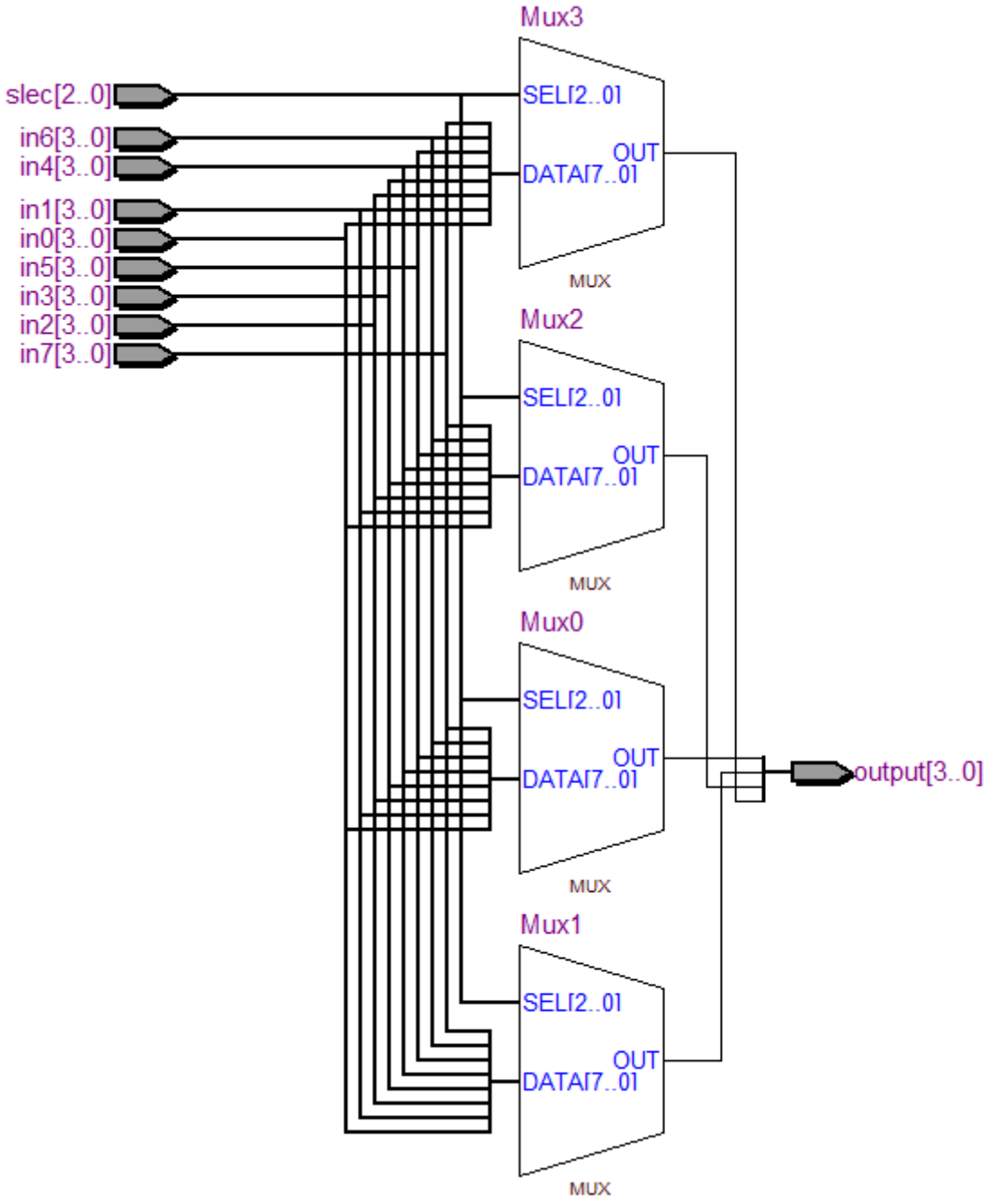
电路总RTL结构图：



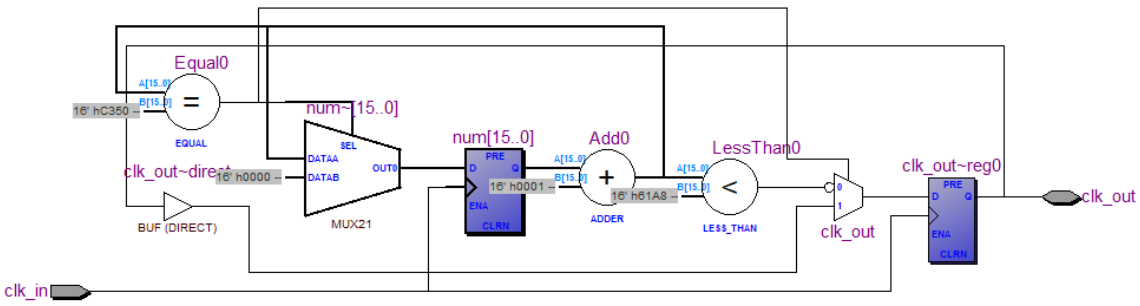
寄存器组RTL结构图：


decoder38_0_mzw:decoder38
sel[2..0]
A[2..0]
Y[7..0]
LATCH_4_mzw:l7
LATCH_4_mzw:l6
LATCH_4_mzw:l5
LATCH_4_mzw:l4
LATCH_4_mzw:l3
LATCH_4_mzw:l2
LATCH_4_mzw:l1
LATCH_4_mzw:l0
cs
en
D[3..0]
Q[3..0]
d7[3..0]
d6[3..0]
d5[3..0]
d4[3..0]
d3[3..0]
d2[3..0]
d1[3..0]
d0[3..0]
en
datain[3..0]

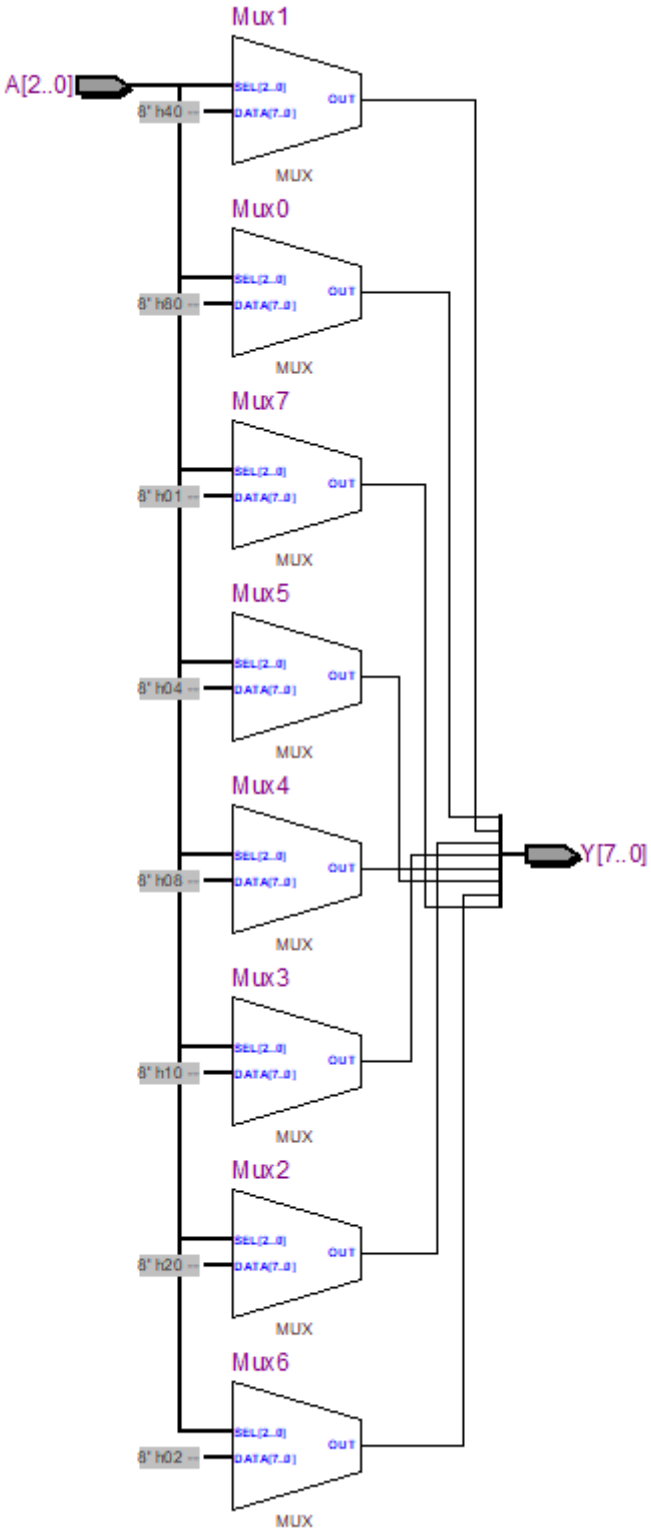
选择器RTL结构图：



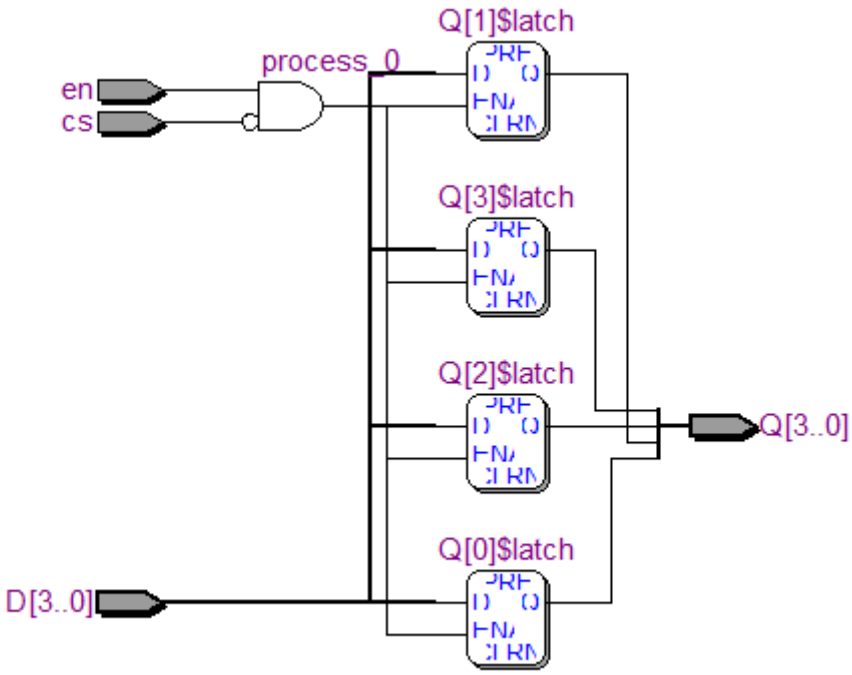
分频器RTL结构图：



38译码器RTL结构图：


Mux1
A[2..0]
8'h40
SEL[2..0]
DATA[7..0]
OUT
MUX
Mux0
8'h80
Mux7
8'h01
Mux5
8'h04
Mux4
8'h08
Mux3
8'h10
Mux2
8'h20
Mux6
8'h02
Y[7..0]

锁存器RTL结构图：



FPGA验证结果记录

实验管脚约束情况如下：

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate
clk_in	Input	PIN_W16	4A	B4A_N0	2.5 V (default)		12mA (default)	
datain[3]	Input	PIN_Y11	3B	B3B_N0	2.5 V (default)		12mA (default)	
datain[2]	Input	PIN_R11	3B	B3B_N0	2.5 V (default)		12mA (default)	
datain[1]	Input	PIN_U12	3B	B3B_N0	2.5 V (default)		12mA (default)	
datain[0]	Input	PIN_V13	4A	B4A_N0	2.5 V (default)		12mA (default)	
en	Input	PIN_T19	5A	B5A_N0	2.5 V (default)		12mA (default)	
LED_CS[7]	Output	PIN_E12	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_CS[6]	Output	PIN_D13	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_CS[5]	Output	PIN_A13	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_CS[4]	Output	PIN_C11	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_CS[3]	Output	PIN_F13	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_CS[2]	Output	PIN_E14	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_CS[1]	Output	PIN_B15	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_CS[0]	Output	PIN_B16	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_seq[6]	Output	PIN_K19	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_seq[5]	Output	PIN_H18	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_seq[4]	Output	PIN_K20	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_seq[3]	Output	PIN_M18	5B	B5B_N0	2.5 V (default)		12mA (default)	1 (default)
LED_seq[2]	Output	PIN_E16	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_seq[1]	Output	PIN_G13	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
LED_seq[0]	Output	PIN_G17	7A	B7A_N0	2.5 V (default)		12mA (default)	1 (default)
sel[2]	Input	PIN_Y14	4A	B4A_N0	2.5 V (default)		12mA (default)	
sel[1]	Input	PIN_Y15	4A	B4A_N0	2.5 V (default)		12mA (default)	
sel[0]	Input	PIN_AA15	4A	B4A_N0	2.5 V (default)		12mA (default)	

数据输入，选择信号，使能信号均采用拨码开关输入，数码管的位选端和段选端分别接到对应电路，时钟信号由50Mhz进行50000分频得到5kHz时钟信号。

实验现象：

上电后，各数码管均显示随机，通过选择信号选定一个数码管，再通过数据输入端设置特定值，打开使能开关，即可让该数码管显示特定值。使能无效时，数码管显示不会随选择信号和数据输入信号改变，使能有效时，当前选择信号所选定的数码管的值随当前数据输入信号的值改变。

实验总结

本次实验通过对数码管动态显示电路的设计，学习了数码管动态扫描，在较少的引脚的条件下，利用人眼的视觉暂留效应，达到数码管多位显示的效果。该方法的关键是要做好位选信号和段选信号的同步，否侧二则错位会导致每位数码管会显示原本全部数码管的内容，导致数码管近似显示8。在实验过程中，时钟信号的选择也较为关键，太快会影响信号同步，需经过调试选出一个满足要求的时钟信号。本实验还通过存储电路存储数码管显示值，学习了存储器电路的知识。

VHDL源代码

```
-- FPGA_EXP4_mzw @ PB20051061

library ieee;
use ieee.std_logic_1164.all;

entity FPGA_EXP4_mzw is
    port(
        sel :IN std_logic_vector(2 downto 0);
        datain :IN std_logic_vector(3 downto 0);
        en :IN std_logic;
        clk_in :IN std_logic;
        LED_CS :OUT std_logic_vector(7 downto 0);
        LED_seg :OUT std_logic_vector(6 downto 0) -- 6543210 -> abcdefg
    );

end FPGA_EXP4_mzw;

architecture arch_FPGA_EXP4 of FPGA_EXP4_mzw is
    component register0_7_mzw port(
        sel :IN std_logic_vector(2 downto 0);
        datain :IN std_logic_vector(3 downto 0);
        en :IN std_logic;
        d0 :OUT std_logic_vector(3 downto 0);
        d1 :OUT std_logic_vector(3 downto 0);
        d2 :OUT std_logic_vector(3 downto 0);
        d3 :OUT std_logic_vector(3 downto 0);
        d4 :OUT std_logic_vector(3 downto 0);
        d5 :OUT std_logic_vector(3 downto 0);
        d6 :OUT std_logic_vector(3 downto 0);
        d7 :OUT std_logic_vector(3 downto 0)
    );
    end component;

    component DIV_mzw port(
        clk_in :IN std_logic;
        clk_out :INOUT std_logic
    );
    end component;

    component decoder38_1_mzw port(
```

```
        A :IN std_logic_vector(2 downto 0);
        Y :OUT std_logic_vector(7 downto 0)
    );
    end component;

    component counter8_mzw port(
        clk :IN std_logic;
        cnt :INOUT std_logic_vector(2 downto 0)
    );
    end component;

    component SLEC4_8TO1_mzw port(
        in0 :IN std_logic_vector(3 downto 0);
        in1 :IN std_logic_vector(3 downto 0);
        in2 :IN std_logic_vector(3 downto 0);
        in3 :IN std_logic_vector(3 downto 0);
        in4 :IN std_logic_vector(3 downto 0);
        in5 :IN std_logic_vector(3 downto 0);
        in6 :IN std_logic_vector(3 downto 0);
        in7 :IN std_logic_vector(3 downto 0);
        slec :IN std_logic_vector(2 downto 0);
        output :OUT std_logic_vector(3 downto 0)
    );
    end component;

    component decoder47_mzw port(
        A :IN std_logic_vector(3 downto 0);
        Y :OUT std_logic_vector(6 downto 0) -- 6543210 -> abcdefg
    );
    end component;

    signal clk_out :std_logic;
    signal cnt :std_logic_vector(2 downto 0);
    signal  d0 :std_logic_vector(3 downto 0);
    signal  d1 :std_logic_vector(3 downto 0);
    signal  d2 :std_logic_vector(3 downto 0);
    signal  d3 :std_logic_vector(3 downto 0);
    signal  d4 :std_logic_vector(3 downto 0);
    signal  d5 :std_logic_vector(3 downto 0);
    signal  d6 :std_logic_vector(3 downto 0);
    signal  d7 :std_logic_vector(3 downto 0);
    signal  bcd :std_logic_vector(3 downto 0);
begin
    reg0_7 :register0_7_mzw port map(sel,datain,en,d0,d1,d2,d3,d4,d5,d6,d7);
    divder :DIV_mzw port map(clk_in,clk_out);
    counter :counter8_mzw port map(clk_out,cnt);
    decoder38 :decoder38_1_mzw port map(cnt,LED_CS);
    selector :SLEC4_8TO1_mzw port map(d0,d1,d2,d3,d4,d5,d6,d7,cnt,bcd);
    decoder47 :decoder47_mzw port map(bcd,LED_seg);
end arch_FPGA_EXP4;
```

```vhdl
-- 四位锁存器latch @ PB20051061 牟真伟

library ieee;
use ieee.std_logic_1164.all;

entity LATCH_4_mzw is
    port(
        D :IN std_logic_vector(3 downto 0);
        Q :OUT std_logic_vector(3 downto 0);
        en :IN std_logic;
        cs :IN std_logic
    );
end LATCH_4_mzw;

architecture arch_LATCH_4 of LATCH_4_mzw is
begin
    process(cs,en,D)
    begin
        if(cs = '0' and en = '1')   then
            Q <= D;
        end if;
    end process;

end arch_LATCH_4;

-- FPGA_EXP3 2to1选择器 @PB20051061牟真伟

library ieee;
use ieee.std_logic_1164.all;

entity SLEC4_8TO1_mzw is
    port(
        in0 :IN std_logic_vector(3 downto 0);
        in1 :IN std_logic_vector(3 downto 0);
        in2 :IN std_logic_vector(3 downto 0);
        in3 :IN std_logic_vector(3 downto 0);
        in4 :IN std_logic_vector(3 downto 0);
        in5 :IN std_logic_vector(3 downto 0);
        in6 :IN std_logic_vector(3 downto 0);
        in7 :IN std_logic_vector(3 downto 0);

        slec :IN std_logic_vector(2 downto 0);
        output :OUT std_logic_vector(3 downto 0)
    );
end SLEC4_8TO1_mzw;

architecture arch_SLEC4_8TO1 of SLEC4_8TO1_mzw is

begin
```

```
process(slec,in0,in1,in2,in3,in4,in5,in6,in7)
    begin
        case slec is
            when "000" => output<=in0;
            when "001" => output<=in1;
            when "010" => output<=in2;
            when "011" => output<=in3;
            when "100" => output<=in4;
            when "101" => output<=in5;
            when "110" => output<=in6;
            when "111" => output<=in7;
            when others => NULL;
        end case;
    end process;
end arch_SLEC4_8TO1;


-- decoder47_mzw
library ieee;
use ieee.std_logic_1164.all;

entity decoder47_mzw is
    port(
        A :IN std_logic_vector(3 downto 0);
        Y :OUT std_logic_vector(6 downto 0) -- 6543210 -> abcdefg
    );
end decoder47_mzw;

architecture arch_decoder47 of decoder47_mzw is

begin
    process(A)
    begin
        case A is
            when "0000" => Y <= "0000001";
            when "0001" => Y <= "1001111";
            when "0010" => Y <= "0010010";
            when "0011" => Y <= "0000110";
            when "0100" => Y <= "1001100";
            when "0101" => Y <= "0100100";
            when "0110" => Y <= "0100000";
            when "0111" => Y <= "0001111";
            when "1000" => Y <= "0000000";
            when "1001" => Y <= "0000100";
            when others => null;
        end case;
    end process;
end arch_decoder47;

-- decoder38_mzw
```

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity decoder38_0_mzw is
    port(
        A :IN std_logic_vector(2 downto 0);
        Y :OUT std_logic_vector(7 downto 0)
    );
end decoder38_0_mzw;

architecture arch_decoder38 of decoder38_0_mzw is

begin
    process(A)
    begin
        case A is
            when "000" => Y <= "11111110";
            when "001" => Y <= "11111101";
            when "010" => Y <= "11111011";
            when "011" => Y <= "11110111";
            when "100" => Y <= "11101111";
            when "101" => Y <= "11011111";
            when "110" => Y <= "10111111";
            when "111" => Y <= "01111111";
            when others => null;
        end case;
    end process;
end arch_decoder38;

library ieee;
use ieee.std_logic_1164.all;

entity decoder38_1_mzw is
    port(
        A :IN std_logic_vector(2 downto 0);
        Y :OUT std_logic_vector(7 downto 0)
    );
end decoder38_1_mzw;

architecture arch_decoder38 of decoder38_1_mzw is

begin
    process(A)
    begin
        case A is
            when "000" => Y <= "00000001";
            when "001" => Y <= "00000010";
            when "010" => Y <= "00000100";
            when "011" => Y <= "00001000";
            when "100" => Y <= "00010000";
```

```
            when "101" => Y <= "00100000";
            when "110" => Y <= "01000000";
            when "111" => Y <= "10000000";
            when others => null;
        end case;
    end process;
end arch_decoder38;

-- 寄存器组

library ieee;
use ieee.std_logic_1164.all;

entity register0_7_mzw is
    port(
        sel :IN std_logic_vector(2 downto 0);
        datain :IN std_logic_vector(3 downto 0);
        en :IN std_logic;
        d0 :OUT std_logic_vector(3 downto 0);
        d1 :OUT std_logic_vector(3 downto 0);
        d2 :OUT std_logic_vector(3 downto 0);
        d3 :OUT std_logic_vector(3 downto 0);
        d4 :OUT std_logic_vector(3 downto 0);
        d5 :OUT std_logic_vector(3 downto 0);
        d6 :OUT std_logic_vector(3 downto 0);
        d7 :OUT std_logic_vector(3 downto 0)
    );
end register0_7_mzw;

architecture arch_register0_7_mzw of register0_7_mzw is
    component LATCH_4_mzw port(
        D :IN std_logic_vector(3 downto 0);
        Q :OUT std_logic_vector(3 downto 0);
        en :IN std_logic;
        cs :IN std_logic
    );
    end component;

    component decoder38_0_mzw port(
        A :IN std_logic_vector(2 downto 0);
        Y :OUT std_logic_vector(7 downto 0)
    );
    end component;

    signal cs7_0 :std_logic_vector(7 downto 0);

begin
    l0 :LATCH_4_mzw port map(datain,d0,en,cs7_0(0));
    l1 :LATCH_4_mzw port map(datain,d1,en,cs7_0(1));
    l2 :LATCH_4_mzw port map(datain,d2,en,cs7_0(2));
```

```
    l3 :LATCH_4_mzw port map(datain,d3,en,cs7_0(3));
    l4 :LATCH_4_mzw port map(datain,d4,en,cs7_0(4));
    l5 :LATCH_4_mzw port map(datain,d5,en,cs7_0(5));
    l6 :LATCH_4_mzw port map(datain,d6,en,cs7_0(6));
    l7 :LATCH_4_mzw port map(datain,d7,en,cs7_0(7));
    decoder38 :decoder38_0_mzw port map(sel,cs7_0);

end arch_register0_7_mzw;

-- 计数器 @PB20051061牟真伟

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity counter8_mzw is
    port(
        clk :IN std_logic;
        cnt :INOUT std_logic_vector(2 downto 0):="000"
    );
end counter8_mzw;

architecture arch_counter8 of counter8_mzw is

begin
    process(clk)
    begin
        if (clk'event and clk='1') then
            cnt <= cnt + 1;
        end if;
    end process;
end arch_counter8;

-- register0_7_tb  @ PB20051061

library ieee;
use ieee.std_logic_1164.all;

entity register0_7_tb is
end register0_7_tb;

architecture arch_register0_7_tb of register0_7_tb is
    component register0_7_mzw port(
        sel :IN std_logic_vector(2 downto 0);
        datain :IN std_logic_vector(3 downto 0);
        en :IN std_logic;
        d0 :OUT std_logic_vector(3 downto 0);
        d1 :OUT std_logic_vector(3 downto 0);
        d2 :OUT std_logic_vector(3 downto 0);
        d3 :OUT std_logic_vector(3 downto 0);
```

```
        d4 :OUT std_logic_vector(3 downto 0);
        d5 :OUT std_logic_vector(3 downto 0);
        d6 :OUT std_logic_vector(3 downto 0);
        d7 :OUT std_logic_vector(3 downto 0)
    );
    end component;
    signal  sel :std_logic_vector(2 downto 0);
    signal  datain :std_logic_vector(3 downto 0);
    signal  en :std_logic;
    signal  d0 :std_logic_vector(3 downto 0);
    signal  d1 :std_logic_vector(3 downto 0);
    signal  d2 :std_logic_vector(3 downto 0);
    signal  d3 :std_logic_vector(3 downto 0);
    signal  d4 :std_logic_vector(3 downto 0);
    signal  d5 :std_logic_vector(3 downto 0);
    signal  d6 :std_logic_vector(3 downto 0);
    signal  d7 :std_logic_vector(3 downto 0);
begin
    reg :register0_7_mzw port map(sel,datain,en,d0,d1,d2,d3,d4,d5,d6,d7);

    process
    begin
        en <= '0';
        wait for 10 ns;
        en <= '1';
        wait for 10 ns;
    end process;

    process
    begin
        datain <= "0000";
        wait for 20 ns;
        datain <= "0001";
        wait for 20 ns;
        datain <= "0010";
        wait for 20 ns;
        datain <= "0011";
        wait for 20 ns;
        datain <= "0100";
        wait for 20 ns;
        datain <= "0101";
        wait for 20 ns;
        datain <= "0110";
        wait for 20 ns;
        datain <= "0111";
        wait for 20 ns;
        datain <= "1000";
        wait for 20 ns;
        datain <= "1001";
        wait for 20 ns;
```

```
        datain <= "1010";
        wait for 20 ns;
        datain <= "1011";
        wait for 20 ns;
        datain <= "1100";
        wait for 20 ns;
        datain <= "1101";
        wait for 20 ns;
        datain <= "1110";
        wait for 20 ns;
        datain <= "1111";
        wait for 20 ns;
    end process;

        process
    begin
        sel <= "000";
        wait for 20 ns;
        sel <= "001";
        wait for 20 ns;
        sel <= "010";
        wait for 20 ns;
        sel <= "011";
        wait for 20 ns;
        sel <= "100";
        wait for 20 ns;
        sel <= "101";
        wait for 20 ns;
        sel <= "110";
        wait for 20 ns;
        sel <= "111";
        wait for 20 ns;

    end process;

end arch_register0_7_tb;
```