基于 fpga 的数字系统设计　　实验报告

评 语	成绩	
教 师: 年　月　日		

学院班级：　171301201

学生学号：　17130120116

学生姓名：　李云水

实验日期：　2020/5/9-2020/5/15

实验题目： 状态机实验

实验环境：联想小新潮 7000 i7 8G ，win10，ISE Design Suite 14.7

一、实验介绍

本次实验完成了书中的状态机实验，完成了 CNTRL_FSM 子模块的描述，同时对 CNTRL_FSM 的 RTL 描述做了进一步的验证。

二、实验目标

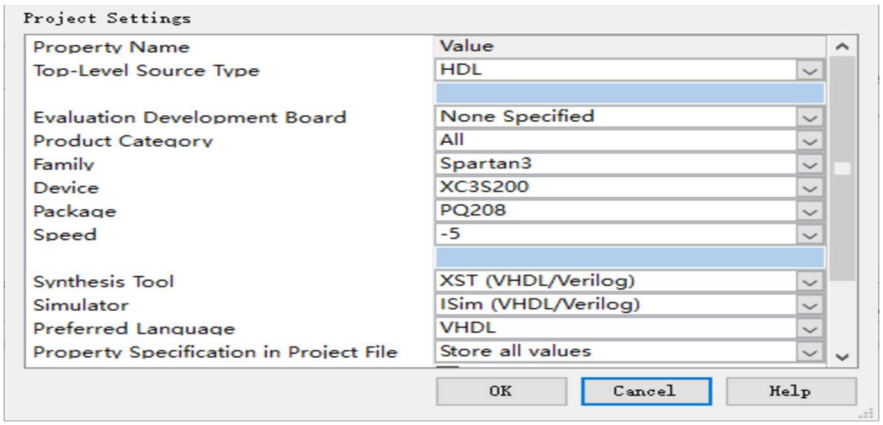
1.学习状态机的 VHDL 语言描述方法

2.学习状态机的单进程和多进程描述方法

三、实验过程

1.创建一个新的工程

配置属性



2.FSM 的多进程描述方式

a.CNTRL_FSM 代码：

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use WORK.CALC1_PACK.ALL;

entity CNTRL_FSM is
    port( DATA_FRAME: in MY_RECORD;
        CLK: in STD_LOGIC;
        RESET: in STD_LOGIC;
        A_IN: out STD_LOGIC_VECTOR(3 downto 0);
        B_IN: out STD_LOGIC_VECTOR(3 downto 0);
        C_IN: out STD_LOGIC;
        OP_CODE: out STD_LOGIC_VECTOR(3 downto 0);
        EXP: out STD_LOGIC_VECTOR(3 downto 0);
        ADDR: out STD_LOGIC_VECTOR(2 downto 0);
        COMP_EN: out STD_LOGIC;
        MEM_EN: out STD_LOGIC;
        ALU_EN: out STD_LOGIC);
end CNTRL_FSM;
```

```
architecture RTL of CNTRL_FSM is
    type CNTRL_STATE is(S0_INIT,S1_FETCH,S2_ALU,S3_COMP,S4_DONE);
    signal CURR_STATE,NEXT_STATE : CNTRL_STATE;

    signal ADDR_I,ADDR_Q : STD_LOGIC_VECTOR(2 downto 0);
begin
    ADDR<=ADDR_Q;

    Sync: process(CLK,RESET)
    begin
        if (RESET ='1')then
        CURR_STATE<=S0_INIT;
        ADDR_Q<=( others =>'0');
        elsif rising_edge ( CLK ) then
        CURR_STATE<=NEXT_STATE;
        ADDR_Q<=ADDR_I;
        end if;
end process;

    COMB: process(CURR_STATE,DATA_FRAME,ADDR_Q)
    begin
        A_IN<=DATA_FRAME.A_IN;
        B_IN<=DATA_FRAME.B_IN;
        C_IN<=DATA_FRAME.C_IN;
        OP_CODE<=DATA_FRAME.OP_CODE;
        EXP<=DATA_FRAME.EXP_OUT;

        ADDR_I<=ADDR_Q;

        case CURR_STATE is
            when S0_INIT=>
            MEM_EN <='0';
            ALU_EN <='0';
            COMP_EN <='0';
            NEXT_STATE <=S1_FETCH;

            when S1_FETCH=>
            MEM_EN <='1';
            ALU_EN <='0';
            COMP_EN <='0';
            NEXT_STATE <=S2_ALU;

            when S2_ALU=>
            MEM_EN <='0';
            ALU_EN <='1';
            COMP_EN <='0';
            NEXT_STATE <=S3_COMP;
```

```
            when S3_COMP=>
            MEM_EN <='0';
            ALU_EN <='0';
            COMP_EN <='1';
            NEXT_STATE <=S4_DONE;

            when S4_DONE=>
            if ADDR_Q>="101"then
                NEXT_STATE <=S4_DONE;
            else
                NEXT_STATE <=S1_FETCH;
                ADDR_I<=ADDR_Q+1;
            end if;
            MEM_EN <='0';
            ALU_EN <='0';
            COMP_EN <='0';

            when others=>
            NEXT_STATE <=S0_INIT;
            MEM_EN <='0';
            ALU_EN <='0';
            COMP_EN <='0';
        end case;
    end process;
end RTL;
```

b. CALC1_PAK 代码：

```
    --引入库
library IEEE;
use IEEE.STD_LOGIC_1164.all;
--定义包结构
package CALC1_PAK is
--分别定义五个端口
 type MY_RECORD is record
 A_IN : std_logic_vector ( 3 downto 0 );
 B_IN : std_logic_vector ( 3 downto 0 );
 OP_CODE : std_logic_vector ( 3 downto 0 );
 C_IN : std_logic;
 EXP_OUT : std_logic_vector ( 3 downto 0 );
end record MY_RECORD;
end package CALC1_PAK;
```

创建 VHDL Module 写入 CNTRL_FSM 代码，导入 CALC1_PAK 包

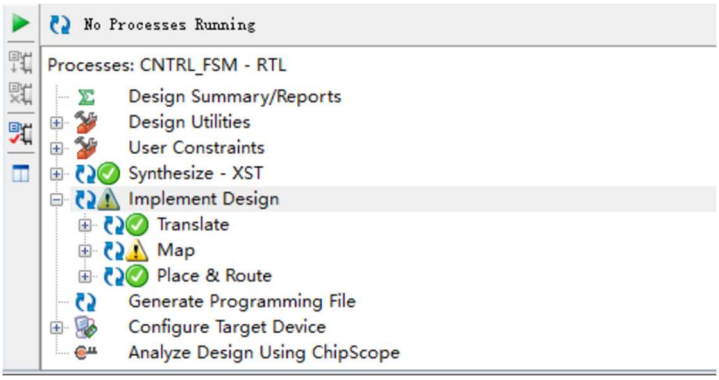
c. 编写测试文件

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

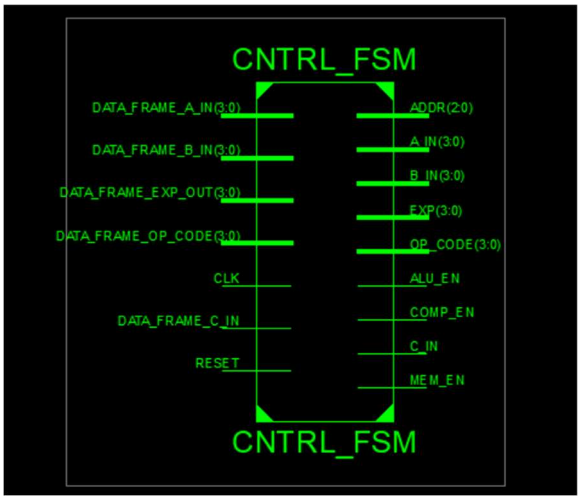
```
use WORK.CALC1_PACK.ALL;

entity CNTRL_FSM_TB_VHD is
end CNTRL_FSM_TB_VHD;

architecture TEST of CNTRL_FSM_TB_VHD is
component CNTRL_FSM
port(DATA_FRAME: in MY_RECORD;
CLK: in STD_LOGIC;
RESET: in STD_LOGIC;
A_IN: out STD_LOGIC_VECTOR(3 downto 0);
B_IN: out STD_LOGIC_VECTOR(3 downto 0);
C_IN: out STD_LOGIC;
OP_CODE: out STD_LOGIC_VECTOR(3 downto 0);
EXP: out STD_LOGIC_VECTOR(3 downto 0);
MEM_EN: out STD_LOGIC;
ALU_EN: out STD_LOGIC;
COMP_EN: out STD_LOGIC;
ADDR: out STD_LOGIC_VECTOR(2 downto 0));
end component;

signal DATA_FRAME:MY_RECORD:=("0000","0000","0000",'0',"0000");
signal CLK: STD_LOGIC:='0';
signal RESET: STD_LOGIC:='0';
signal A_IN,B_IN: STD_LOGIC_VECTOR(3 downto 0);
signal C_IN: STD_LOGIC;
signal OP_CODE: STD_LOGIC_VECTOR(3 downto 0);
signal EXP: STD_LOGIC_VECTOR(3 downto 0);
signal ALU_EN,MEM_EN,COMP_EN: STD_LOGIC;
signal ADDR: STD_LOGIC_VECTOR(2 downto 0);
begin
UUT: CNTRL_FSM port map(
DATA_FRAME =>DATA_FRAME,
CLK =>CLK,
RESET =>RESET,
A_IN =>A_IN,
B_IN =>B_IN,
C_IN =>C_IN,
OP_CODE =>OP_CODE,
EXP =>EXP,
ALU_EN =>ALU_EN,
MEM_EN =>MEM_EN,
COMP_EN =>COMP_EN,
ADDR =>ADDR);

CLK <=not CLK after 20 ns;
RESET <='1' after 10 ns ,'0' after 25 ns;
```

```
TB:process
begin
DATA_FRAME<=("1000","0100","0000",'0',"0000");
wait for 100 ns;
DATA_FRAME<=("1000","0100","0101",'0',"0000");
wait for 100 ns;
DATA_FRAME<=("1000","0100","0100",'0',"0000");
wait;
end process;
end test;
```

检查语法是否正确：

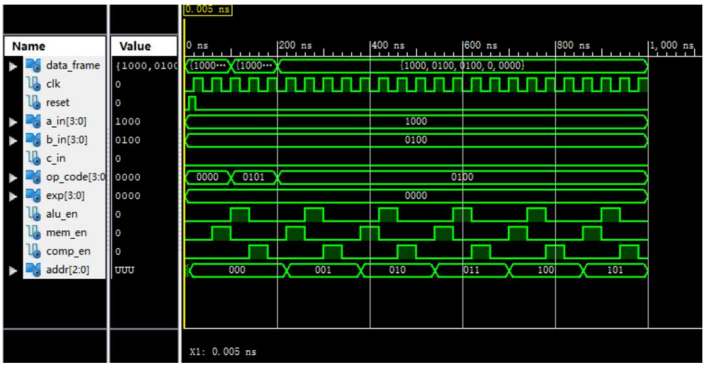


RTL 图：

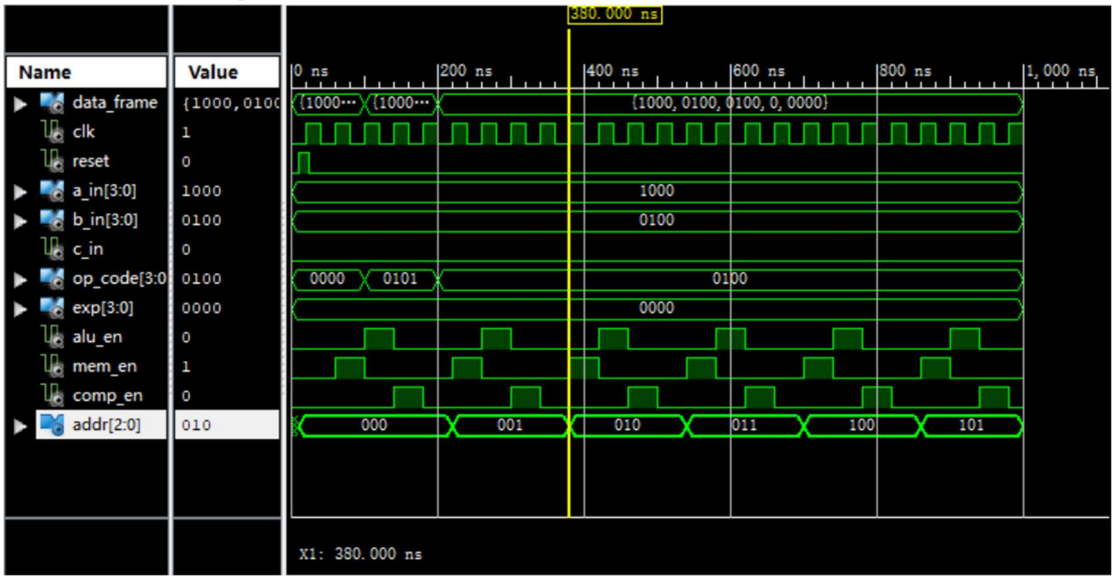


3、仿真

a.将模式设置为 behavioral 打开仿真工具



b. 将模式设置为 post-router 并再次仿真



c. 后仿设计的最高工作频率：

```
--------------------------------------------------------------------------------
Delay:               3.574ns (Levels of Logic = 2)
  Source:            ADDR_Q_2 (FF)
  Destination:       CURR_STATE_FSM_FFd1 (FF)
  Source Clock:      CLK rising
  Destination Clock: CLK rising

  Data Path: ADDR_Q_2 to CURR_STATE_FSM_FFd1
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDCE:C->Q             4    0.626   1.074  ADDR_Q_2 (ADDR_Q_2)
     LUT3_D:I0->O          1    0.479   0.740  CURR_STATE_cmp_ge00001 (CURR_STATI
     LUT3:I2->O            1    0.479   0.000  CURR_STATE_FSM_FFd1-In1 (CURR_STA
     FDC:D                      0.176          CURR_STATE_FSM_FFd1
    ----------------------------------------
    Total                      3.574ns (1.760ns logic, 1.814ns route)
                                       (49.2% logic, 50.8% route)

==========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'CLK'
  Total number of paths / destination ports: 6 / 6
--------------------------------------------------------------------------------
```

d. 资源分析报告

CNTRL_FSM Project Status (05/09/2020 - 13:31:36)			
Project File:	lab7.xise	**Parser Errors:**	No Errors
Module Name:	CNTRL_FSM	**Implementation State:**	Placed and Routed
Target Device:	xc3s200-5pq208	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	No Warnings
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	All Constraints Met
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary [-]				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	8	3,840	1%	
Number of 4 input LUTs	6	3,840	1%	
Number of occupied Slices	5	1,920	1%	
Number of Slices containing only related logic	5	5	100%	
Number of Slices containing unrelated logic	0	5	0%	
Total Number of 4 input LUTs	6	3,840	1%	
Number of bonded IOBs	42	141	29%	
Number of BUFGMUXs	1	8	12%	
Average Fanout of Non-Clock Nets	1.96			

四、实验总结

总体而言，这次实验是一次比较难的实验，以至于我花了一周多来消化这些东西，它不像以往只需要鼠标的操作就能基本完成实验，对于后仿真，需要了解到它的编译文件，并对其进行修改，而且进行一些额外的以前没有过的操作，尽管如此，我们还是无法得到我们预想的结果，后仿波形中，并无时延产生，于是依葫芦画瓢，把 myrecord 的内容拆解等一些操作才接近时延的完成。感谢潘同学的无私奉献，从他的文字里的一些理解，让我对 FPGA 的理解也逐渐加深。