

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： 2020/5/15

## 实验题目：存储器与记录类型实验

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

### 一、实验介绍

本实验将完成图 6.15 所示的结构中 MEM 部分的 moban 。实验内容主要涉及记录（record）数据类型的使用方法，以及如何以此为基础，用二维数组描述一个存储器。

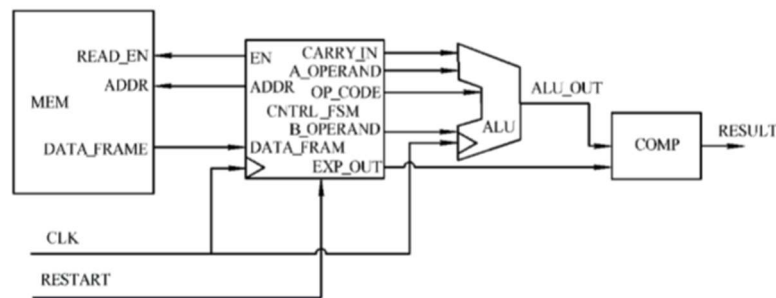


图 6.15 简单的计数器

### 二、实验目标

1. 用记录（record）数据类型创建一个用户自定义的 RAM 结构
2. 创建计数器的第一个文件
3. 多种 VHDL 数据类型的复合使用
4. 验证逻辑结构

### 三、实验过程

1. 创建一个新的工程

➤ New Project Wizard

← Create New Project  
Specify project location and type.

Enter a name, locations, and comment for the project

Name: lab3  
Location: E:\FPGALab\lab3\lab3  
Working Directory: E:\FPGALab\lab3\lab3

#### 配置属性

Property Name	Value
Top-Level Source Type	HDL
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3
Device	XC3S200
Package	PQ208
Speed	-5
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values

OK Cancel Help

## 2、创建一个包含记录（record）数据类型的包集合

创建一个包集合，并在包集合中包含用户自定义的记录(record)数据类型。

(1) 选择菜单栏的 Project→New Source。

(2) 在 Select Source Type 窗口中，选择左侧的 VHDL Package，在右侧 File Name 栏中填入文件名 CALC1\_PACKAGE。

(3) 单击 Next 按钮，文本编辑框中即显示已创建好的包集合模板。

这个包集合模板中包含了各种常用的声明语句，在使用时可能感觉比较繁杂，由于本实验只定义一个记录(record)数据类型，读者可以只保留包集合模板中 package 和 end package 的声明语句，其他语句都删掉。

(4) 将包集合命名为 CALC1\_PAK。

(5) 创建一个记录，命名为 MY\_RECORD，MY\_RECORD 包含以下元素：

```
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 MYRECORD;
```

## 3、创建一个二维存储器数组

用已经声明的 MY\_RECORD 数据类型声明一个二维存储器数组

(1) 选择菜单栏中的 Project→New Source;

(2) 在 Select Source Type 窗口中，选择左侧的 VHDL Module，在右侧 File Name 栏中填入文件名 MEM，单击 Next 按钮进入端口定义窗口 Define Module。

(3) 在 Define Module 窗口中输入下面的端口定义：

```
ADDR : in STD_LOGIC_VECTOR(2 downto 0);
EN : in STD_LOGIC;
DATA_FRAME : out MY_RECORD;
```

注意：在 Define Module 窗口中是无法定义 MY\_RECORD 类型的，窗口中的端口默认为 STD\_LOGIC 和 STD\_LOGIC\_VECTOR 类型的数据。DATA\_FRAME 的数据类型需要在文件生成好后，在文本编辑器中修改。

(4) 在 MEM 的库和包集合使用声明中，加入对 CALC1\_PAK 的使用声明。

(5) 在结构体中定义一个二维存储器类型 ROM\_ARRAY，ROM\_ARRAY 类型中包含 6 个 MY\_RECORD 类型的数据。

(6) 声明一个 ROM\_ARRAY 类型的常量 MY\_ROM，并对其进行初始化。

(7) 在结构体的进程中，用 MY\_ROM 中的数据对输出端口赋值，读者可以参考本实验后面的实验代码完成此步骤。

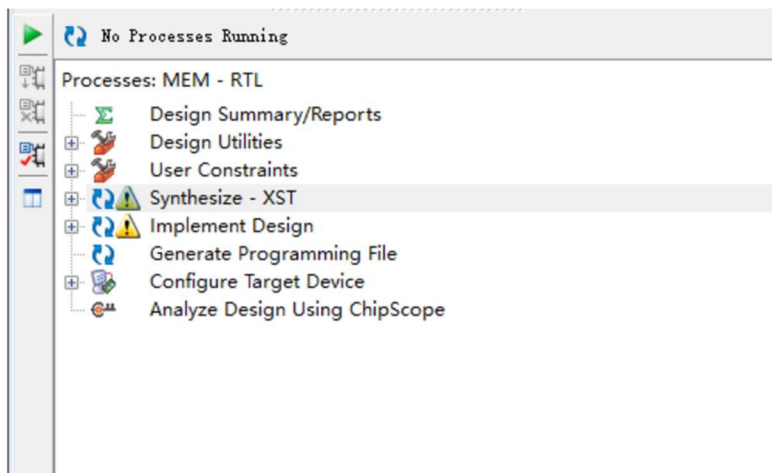
语法检查：

(1) 在 Sources 窗口中选择 MEM.VHD，在 Processes 窗口中双击 Check Syntax。

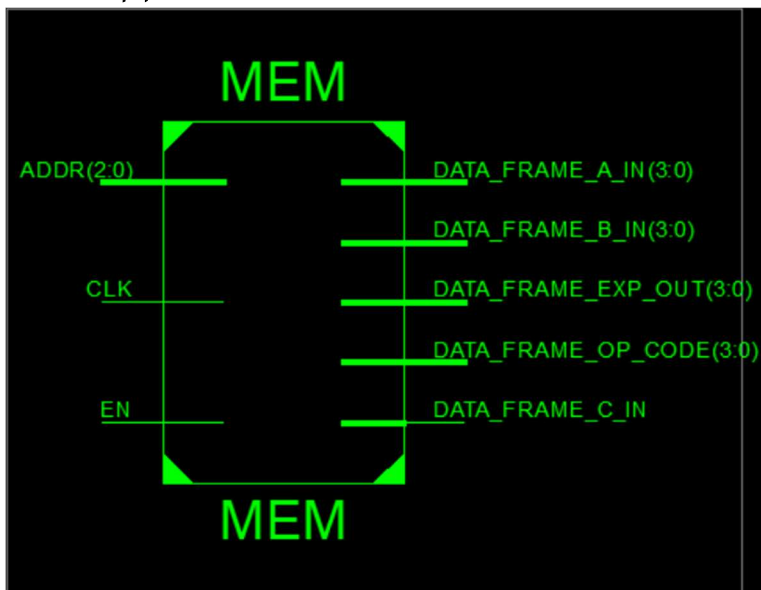
(2) 如果语法检查有错误，请修正错误后继续进行下面的步骤。

注意：如果此时查看 MEM 对应的 RTL 原理图，读者会看到这个原理图并不完整，这是由于存储器中全部的值都是常量，而且没有做任何的处理，综合器在综合时将电路做了优化。

其中语法检查中出现了一些 warning，其中之一就是存储器中的值为常量，且没有做处理。其他 warning 是一些版本问题。



RTL 图:



#### 四、实验总结

在本实验中，我们首先创建了一个包集合，包集合中的内容将在后续的设计中经常用到；之后，创建了一个用户自定义的记录类型，并以此为基础定义了一个二维存储数组类型。存储器内部的值设置为常量，我们会在后续的实验中修改存储器的值。

#### 五、实验代码及详细注释

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;

```

MEM 代码:

```

--引入库
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--引入自建包
use work.CALC1_PAK.ALL;
--定义实体
entity MEM is
    --定义端口, 在 RTL 图中可清楚看到
    Port ( CLK, EN: in std_logic ;
          ADDR : in std_logic_vector(2 downto 0):="000";  -- initial-
ize for simulation
          DATA_FRAME : out MY_RECORD );
end entity MEM;

architecture RTL of MEM is
    --定义二维存储器类型, 内含 6 个 MY_RECORD 数据
    type ROM_ARRAY is array ( 0 to 5 ) of MY_RECORD;
    --初始化 MY_ROM
    constant MY_ROM : ROM_ARRAY :=
        ( 0 => ( A_IN => "1000", B_IN => "0010", OP_CODE => "0001", C_IN => '
0', EXP_OUT => "1010" ),
          1 => ( A_IN => "0100", B_IN => "0010", OP_CODE => "0001", C_IN => '
0', EXP_OUT => "0110" ),
          2 => ( A_IN => "0010", B_IN => "0010", OP_CODE => "0001", C_IN => '
0', EXP_OUT => "0100" ),
          3 => ( A_IN => "0001", B_IN => "0010", OP_CODE => "0001", C_IN => '
0', EXP_OUT => "0011" ), -- induce error
          4 => ( A_IN => "0011", B_IN => "0010", OP_CODE => "0001", C_IN => '
0', EXP_OUT => "0101" ),
          5 => ( A_IN => "0111", B_IN => "0010", OP_CODE => "0001", C_IN => '
0', EXP_OUT => "1001" ));

    --

begin

process ( CLK )
    begin
        if rising_edge ( CLK ) then
            if ( EN = '1' ) then

```

```
        DATA_FRAME <= MY_ROM(conv_integer(ADDR)); --根据地址输出数据
    end if;
end if;
end process ;

end RTL;
```

## 六、实验中的问题

这次的实验我学习了创建新的包集合，以及二维存储器数组代码的创建，对端口、过程、实体等概念有了进一步的理解，目前 Testbench 还在 coding 中，其他部分由于实验本身比较基础所以没遇到什么问题。