



西安电子科技大学
XIDIAN UNIVERSITY

基于 FPGA 的数字系统设计

实验报告

实验名称：LAB4:n 比特计数器及 RTL 验证实验与
资源分析报告

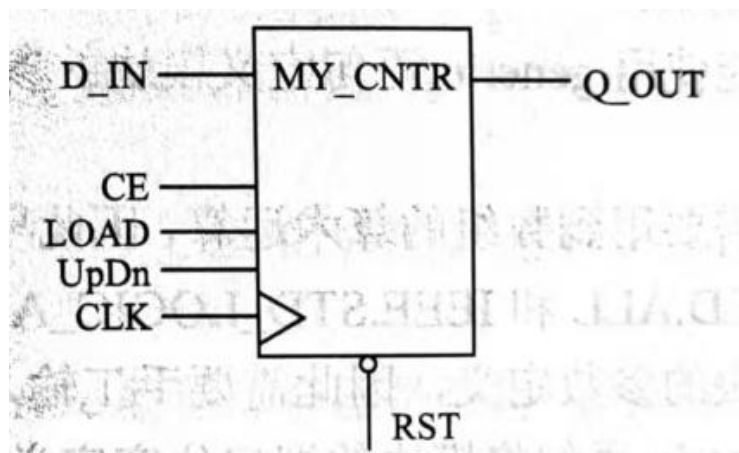
任课教师：沈沛意老师

学号姓名：

提交日期：

一、实验介绍

本实验将完成 MY_CNTR 计数器的 RTL 描述，计数器的位宽用 generic 语句设置为参数。MY_CNTR 是一个 n 比特二进制的计数器，可以向上向下计数，并可设置计数值，计数器用异步的方式进行低电平复位。实验的后半部分将会建立一个测试平台，测试计数器的 RTL 代码是否能够完成既定的功能。



二、实验目标

创建、仿真并验证 n 比特二进制计数器的 RTL 代码

在 ISE 中使用语法模板

使用 VHDL 中的 generic 语句

三、实验过程与步骤

本实验包含三个主要的部分:创建一个新的 ISE 工程:用 generic 语句声明参数;创建一个测试平台验证设计文件的正确性

1. 启动 ISE 创建一个新的工程 LAB64

2. 用 generic 语句声明参数

- 使用 VHDL 中的 generic 语句将模块的端口位宽定义为参数，参数的类型定义为整数类型，默认值为 8。

- 在结构体内，用 if/else 语句描述计数器的功能实现。
- 在描述功能实现时，读者首先需要考虑这些控制信号的优先级，然后考虑在设定的优先级情况下如何描述计数器的功能实现。
- 在结构体中声明一个信号，信号的位宽为 8，用参数声明信号的位宽，信号名称定义为 INT_CNT。在时序进程中，将控制这个信号的值。
- 在进程外，用 INT_CNT 对模块的端口 Q_OUT 赋值，即 $Q_OUT \leq INT_CNT$ 。

- 复位时，用下面的语句对 COUNT 信号赋值：
 $INT_CNT \leq (others \Rightarrow '0');$

使用这种方式赋值时，设计者不需要关注信号的实际位宽。这是增强程序灵活性的一种重要方法，因为在使用 generic 语句时，端口的位宽可以是任意长度。

3. 创建测试平台

- 建立测试平台并验证计数器的功能是否满足设计需求：测试平台生成向导已经为用户搭建好了测试平台的框架，我们利用此框架，只关注测试输入激励。

(1)CLOCK 时钟频率设置为 100 MHz。

(2)在 15 ns 时对计数器进行复位，复位保持时间为 25 ns

(3)CE 信号的初始值设置为 High，300ns 后拉低，并保持 100ns。

(4)LOAD 信号初始值为 Low，500ns 后拉高，并保持一个时钟周期。

(5)UPDN 信号初始值为 High, 750 ns 后拉低。

(6)DIN 信号的值设置为 X"0F"或 00001111

四、实验结果分析

(一)代码及注释

MY_CNTR.v:

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

entity MY_CNTR is

generic (CNT_WIDTH: INTEGER:= 8);
port (CLK: in STD_LOGIC;
      RST: in STD_LOGIC;
      D_IN: in STD_LOGIC_VECTOR(CNT_WIDTH -1 downto 0);
      Q_OUT: out STD_LOGIC_VECTOR(CNT_WIDTH -1 downto 0);
      LOAD: in STD_LOGIC;
      CE: in STD_LOGIC;
      UPDN: in STD_LOGIC);
end MY_CNTR;

architecture RTL of MY_CNTR is
signal INT_CNT: STD_LOGIC_VECTOR(CNT_WIDTH -1 downto 0);
begin
process (CLK, RST)
begin
    if RST='0' then -- 异步复位 (低电平有效)
        INT_CNT <= (others => '0'); -- 复位计数器为全 0

    elsif rising_edge(CLK) then -- 时钟上升沿触发
        if CE='1' then -- 计数使能有效
            if LOAD='1' then -- 同步加载模式
                INT_CNT <= D_IN; -- 加载外部输入值
            else -- 计数模式
                if UPDN = '1' then
                    INT_CNT <= INT_CNT+1; -- 递增
                else
                    INT_CNT <= INT_CNT-1; -- 递减
                end if;
            end if;
        end if;
    end if;
end process;
end RTL;
```

```

        end if;
    end if;
end if;
end if;
end process;

Q_OUT <= INT_CNT;
end RTL;

```

MY_CNTR_TB.vhd:

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY MY_CNTR_TB IS
END MY_CNTR_TB;

ARCHITECTURE behavior OF MY_CNTR_TB IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT MY_CNTR
    PORT(
        CLK : IN  std_logic;
        RST : IN  std_logic;
        D_IN : IN  std_logic_vector(7 downto 0);
        Q_OUT : OUT std_logic_vector(7 downto 0);
        LOAD : IN  std_logic;
        CE : IN  std_logic;
        UPDN : IN  std_logic
    );
    END COMPONENT;

    --Inputs
    signal CLK : std_logic := '0';
    signal RST : std_logic := '1';
    signal D_IN : std_logic_vector(7 downto 0) := X"0F";
    signal LOAD : std_logic := '0';
    signal CE : std_logic := '1';
    signal UPDN : std_logic := '1';

    --Outputs
    signal Q_OUT : std_logic_vector(7 downto 0);

```

```
-- Clock period definitions
constant CLK_period : time := 10 ns;
```

```
BEGIN
```

```
-- Instantiate the Unit Under Test (UUT)
```

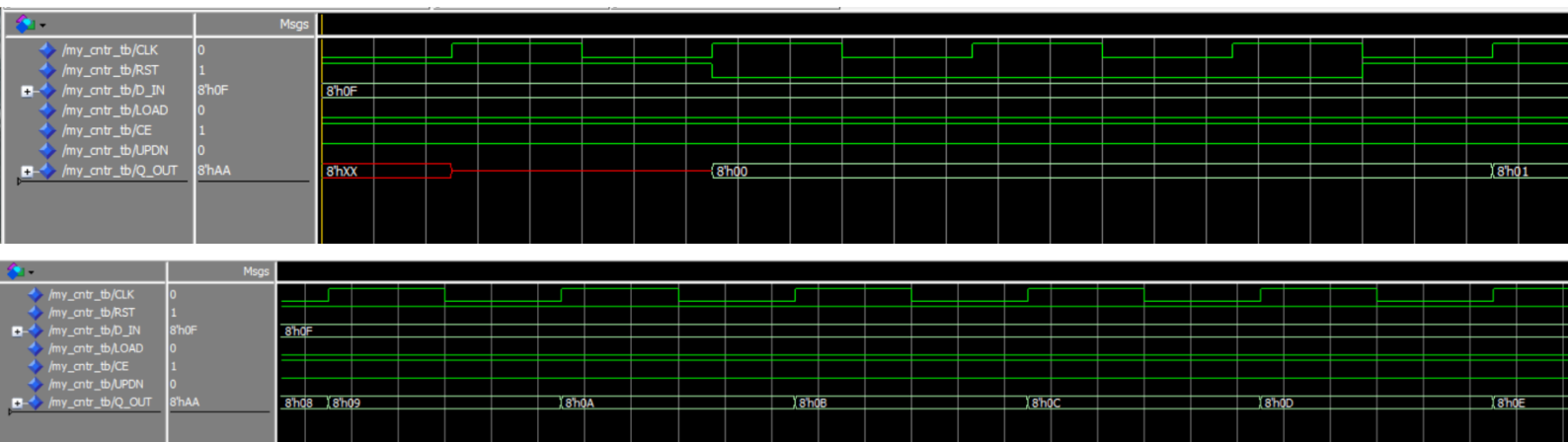
```
uut: MY_CNTR PORT MAP (
    CLK => CLK,
    RST => RST,
    D_IN => D_IN,
    Q_OUT => Q_OUT,
    LOAD => LOAD,
    CE => CE,
    UPDN => UPDN
);
```

```
CLK <= not CLK after 5ns;
CE <= '1','0' after 300ns,'1' after 400ns;
RST <= '0' after 15ns,'1' after 40ns;
UPDN <= '0' after 750ns; -- 750ns 时尝试改变计数方向
LOAD <= '1' after 500ns,'0' after 510ns;
```

```
END;
```

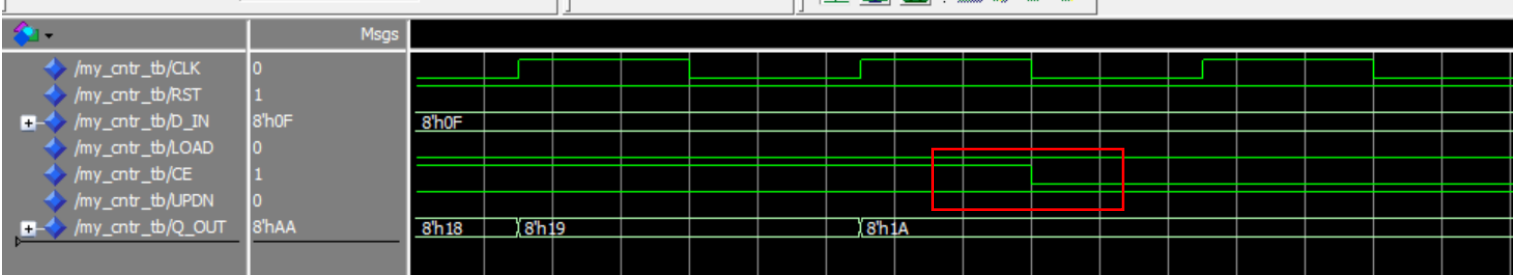
(二)前仿后仿结果以及分析

前仿：

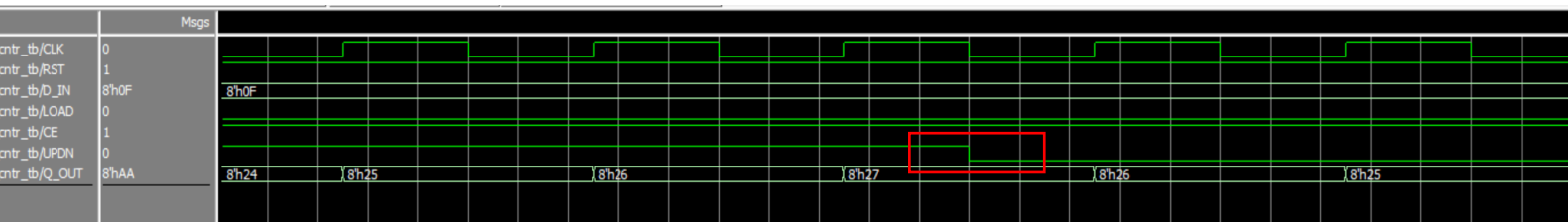


可见，在 CE 使能的状态和 UPDN 为 1 时，计数器递增

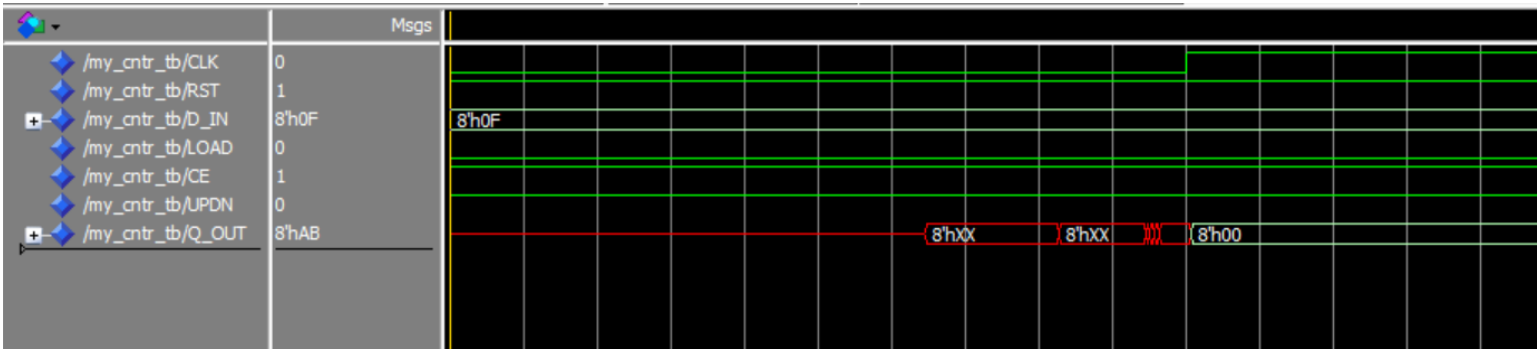
接下来 CE 不使能，计数器停止：



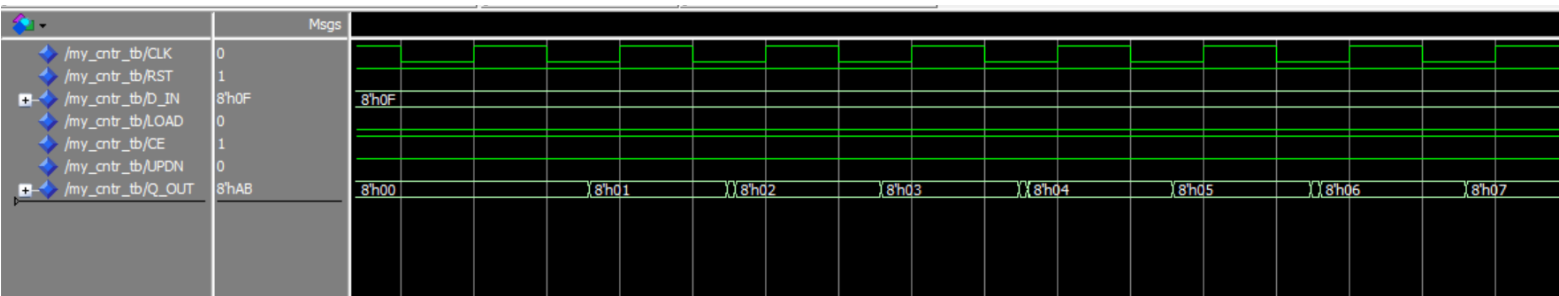
当 UPDN 为 0 时，计数器递减：



后仿：



可见后仿存在一定延迟

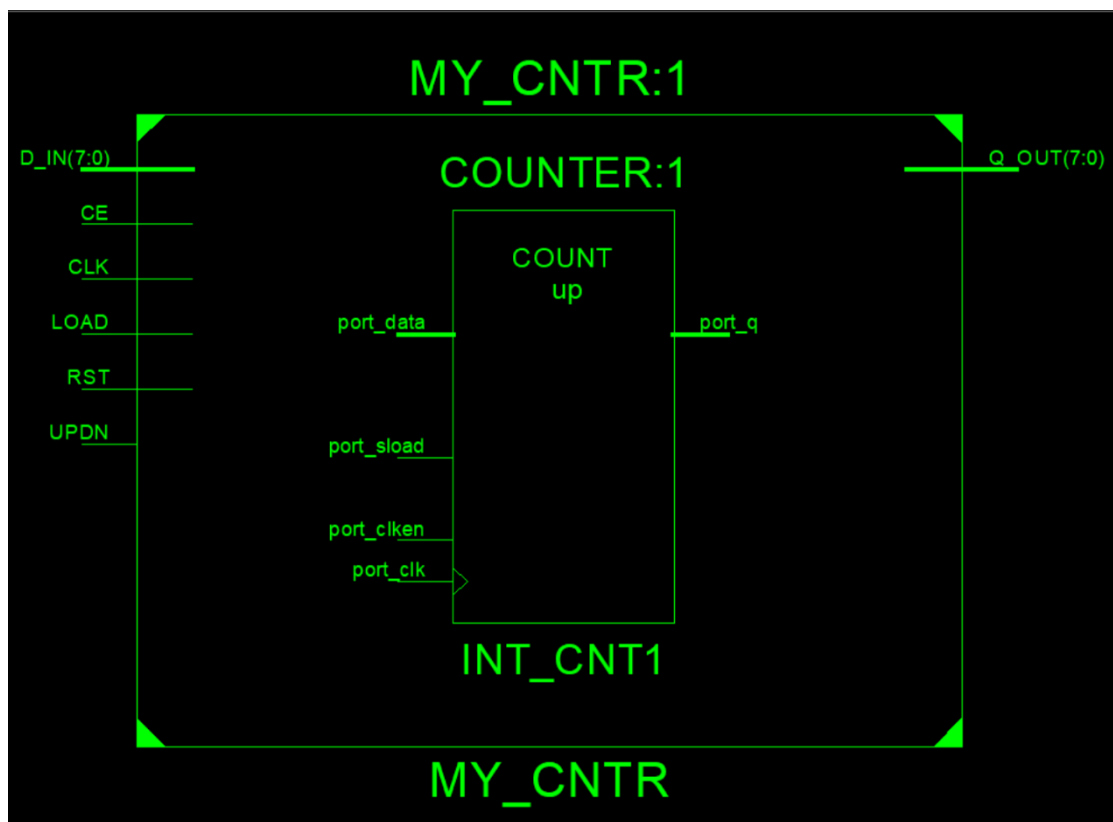


同样的，在 CE 使能的状态和 UPDN 为 1 时，计数器递增；当 UPDN 为 0

时，计数器递减：

	Msgs	
/my_cntr_tb/CLK	0	
/my_cntr_tb/RST	1	
/my_cntr_tb/D_IN	8'h0F	
/my_cntr_tb/LOAD	0	
/my_cntr_tb/CE	1	
/my_cntr_tb/UPDN	0	
/my_cntr_tb/Q_OUT	8'hAB	
		8'h20 8'h21 8'h22 8'h23 8'h24 8'h25 8'h26 8'h27 8'h28 8'h29

RTL:



五、资源分析报告与最高工作频率分析

资源分析报告;

Device utilization summary:

Device Utilization Summary					[-]
Logic Utilization	Used	Available	Utilization	Note(s)	
Number of Slice Flip Flops	8	3,840	1%		
Number of 4 input LUTs	10	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	10	3,840	1%		
Number of bonded IOBs	21	141	14%		
Number of BUFMUXs	1	8	12%		
Average Fanout of Non-Clock Nets	2.52				

评估

资源使用极低：所有核心资源利用率均 <15%，设计规模很小
布局质量良好：Logic Distribution 显示所有 Slice 均包含相关
逻辑
时钟网络合理：使用 1 个全局时钟缓冲器 (BUFGMUX)，符合最佳实践

时序报告与相应最高工作频率计算：

```
Release 14.7 Trace (nt64)
Copyright (c) 1995-2013 Xilinx, Inc. All rights reserved.

D:\Xilinx\14.7\ISE_DS\ISE\bin\nt64\unwrapped\trce.exe -
intstyle ise -v 3 -s 5
-n 3 -fastpaths -xml MY_CNTR.twx MY_CNTR.ncd -o MY_CNTR.twr
MY_CNTR.pcf

Design file: MY_CNTR.ncd
Physical constraint file: MY_CNTR.pcf
Device, package, speed: xc3s200, pq208, -5 (PRODUCTION 1.39
2013-10-13)
Report level: verbose report

Environment Variable Effect
-----
NONE No environment variables were set
```

INFO:Timing:2698 - No timing constraints found, doing default enumeration.

INFO:Timing:3412 - To improve timing, see the Timing Closure User Guide (UG612).

INFO:Timing:2752 - To get complete path coverage, use the unconstrained paths

option. All paths that are not constrained will be reported in the

unconstrained paths section(s) of the report.

INFO:Timing:3339 - The clock-to-out numbers in this timing

report are based on

a 50 Ohm transmission line loading model. For the details of this model,

and for more information on accounting for different loading conditions,

please see the device datasheet.

INFO:Timing:3390 - This architecture does not support a default System Jitter

value, please add SYSTEM_JITTER constraint to the UCF to modify the Clock

Uncertainty calculation.

INFO:Timing:3389 - This architecture does not support 'Discrete Jitter' and

'Phase Error' calculations, these terms will be zero in the Clock

Uncertainty calculation. Please make appropriate modification to

SYSTEM_JITTER to account for the unsupported Discrete Jitter and Phase

Error.

Data Sheet report:

All values displayed in nanoseconds (ns)

Setup/Hold to clock CLK

Source	Max Setup to clk (edge)	Max Hold to clk (edge)	Internal Clock(s)	Clock Phase
CE	1.612 (R)	0.500 (R)	CLK_BUFGP	0.000
D_IN<0>	2.866 (R)	-0.154 (R)	CLK_BUFGP	0.000
D_IN<1>	2.593 (R)	-0.072 (R)	CLK_BUFGP	0.000
D_IN<2>	2.352 (R)	0.168 (R)	CLK_BUFGP	0.000
D_IN<3>	2.647 (R)	-0.204 (R)	CLK_BUFGP	0.000
D_IN<4>	2.242 (R)	0.168 (R)	CLK_BUFGP	0.000
D_IN<5>	2.264 (R)	0.014 (R)	CLK_BUFGP	0.000
D_IN<6>	1.354 (R)	0.637 (R)	CLK_BUFGP	0.000
D_IN<7>	1.284 (R)	0.248 (R)	CLK_BUFGP	0.000
LOAD	4.510 (R)	-0.834 (R)	CLK_BUFGP	0.000
UPDN	3.413 (R)	0.250 (R)	CLK_BUFGP	0.000

Clock CLK to Pad

Destination	clk (edge) to PAD	Internal Clock(s)	Clock Phase
Q_OUT<0>	7.944 (R)	CLK_BUFGP	0.000
Q_OUT<1>	7.339 (R)	CLK_BUFGP	0.000
Q_OUT<2>	8.013 (R)	CLK_BUFGP	0.000
Q_OUT<3>	7.929 (R)	CLK_BUFGP	0.000
Q_OUT<4>	8.232 (R)	CLK_BUFGP	0.000
Q_OUT<5>	8.236 (R)	CLK_BUFGP	0.000
Q_OUT<6>	7.946 (R)	CLK_BUFGP	0.000
Q_OUT<7>	7.979 (R)	CLK_BUFGP	0.000

Clock to Setup on destination clock CLK

Source Clock	Src:Rise	Src:Fall	Src:Rise	Src:Fall
	Dest:Rise	Dest:Rise	Dest:Fall	Dest:Fall
CLK	3.717			

Analysis completed Mon Apr 21 16:46:07 2025

Trace Settings:

Trace Settings

Peak Memory Usage: 4505 MB

最高工作频率计算（理论极限频率）:

$$F_{max} = \frac{1}{T_{clk_min}} = \frac{1}{T_{co} + T_{comb} + T_{su}}$$

取报告中最大组合逻辑延迟 3.717ns

典型寄存器参数 (Spartan-3 -5 速度级):

$$T_{co} \text{ (时钟到输出)} = 0.5\text{ns}$$

$$T_{su} \text{ (建立时间)} = 1.2\text{ns}$$

$$\text{代入公式: } T_{clk_min} = 0.5 + 3.717 + 1.2 = 5.417\text{ns} \Rightarrow F_{max} \approx 184\text{MHz}$$