



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：第七章 LAB1

任课教师：沈沛意老师

学号姓名：

提交日期：

一、实验介绍

本实验主要介绍 Xilinx 的开发工具 ISE 的使用

二、实验目标

- 了解 FPGA 的开发流程
- 熟悉 Spartan-3E 开发套件的功能特点
- 清楚 PicoBlaze 8 位控制器的特性

三、实验过程

本实验主要包含五个部分：

- 使用 ISE 集成环境创建一个新的工程
- 添加设计文件到工程中
- 编译设计
- 仿真设计
- 实现设计

四、实验步骤

1. 启动 ISE 创建一个新的工程

- 打开 ISE 集成环境
- 打开创建新工程界面

New Project Wizard

← **Create New Project**
Specify project location and type.

Enter a name, locations, and comment for the project

Name: Flow_Lab

Location: C:\Users\zbw\Desktop\VHDL_LABS\Flow_Lab ...

Working Directory: C:\Users\zbw\Desktop\VHDL_LABS\Flow_Lab ...

Description:

Select the type of top-level source for the project

Top-level source type: HDL

More Info Next > Cancel

命名为 Flow_Lab

➤ 设置参数

New Project Wizard

← **Project Settings**
Specify device and project properties.

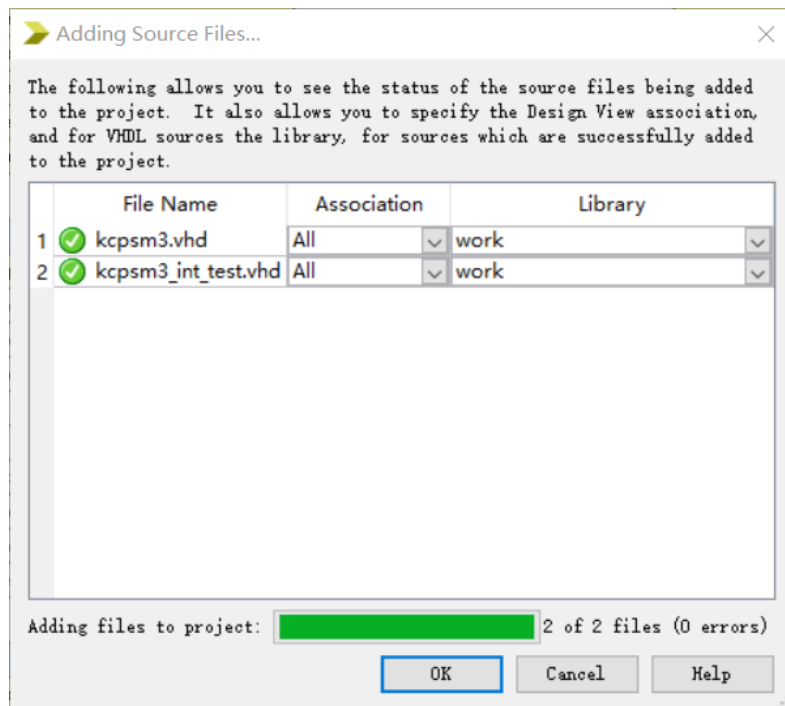
Select the device and design flow for the project

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3E
Device	XC3S500E
Package	FG320
Speed	-4
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

More Info < Back Next > Cancel

2. 添加 HDL 源文件到工程

- 单击 Add Source
- 选择 VHDL/Verilog 文件 kcpsm3_int_test 和 kcpsm3，并单击 Open 按钮
- 单击 Next，然后单击 Finish 按钮，将出现选择源文件类型对话框
- 单击 OK 按钮



3. 编译设计

编写 MY_AND2 实体的 VHDL 代码

- 打开 Windows 搜索，在 KCPSM3 子目录里找到编译器
- 打开文件 int_test.psm，查看代码
- 打开命令窗口
- 使用 cd 命令切换到 Assembler 目录
- 在命令提示符下输入命令(kcpsm3 int_test.psm)，立即编译代码输

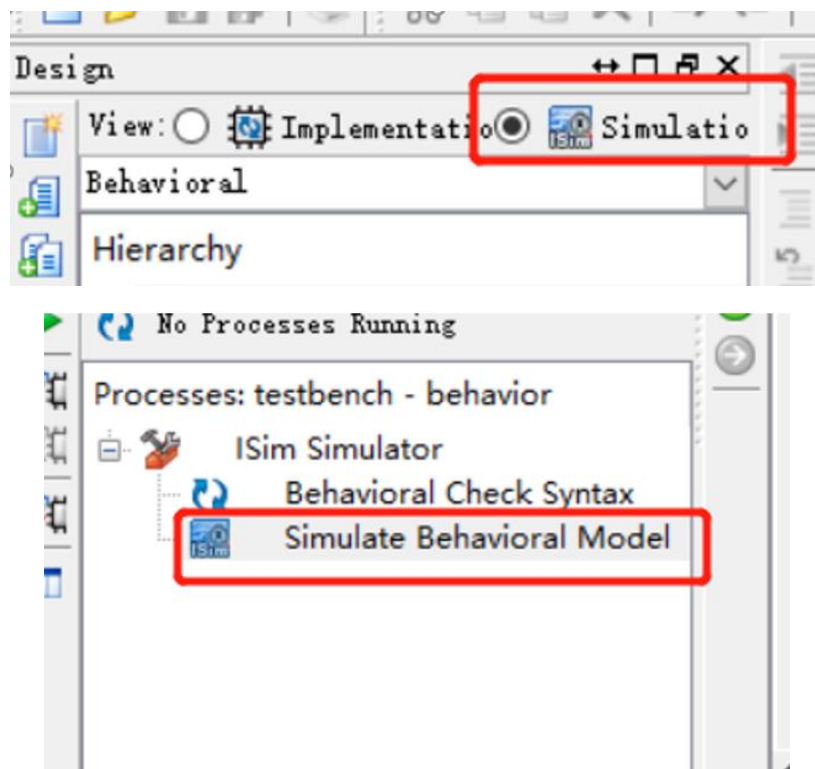
出程序 ROM 文件

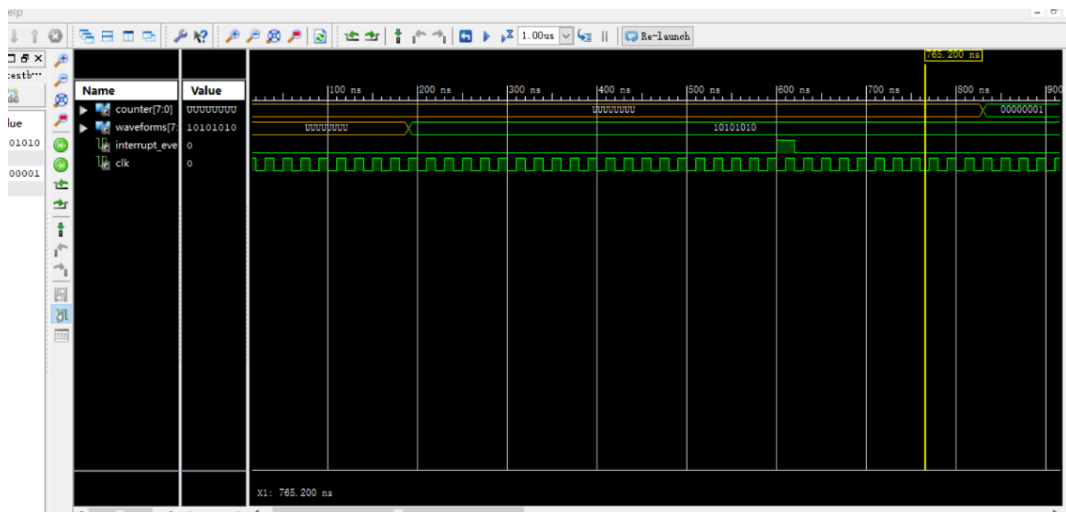
- 在 ISE 中打开 Add Copy of Source 并找到 INIT_TEST.VHD
- 单击 Open 按钮，然后单击 OK 按钮把 INIT_TEST 当做设计文件添加到工程

4. 仿真设计

添加测试文件 testbench.vhd

- 在 ISE 的 Sources 窗口中，选择 Add Copy of Source
- 选择文件 test_bench.vhd 并单击 Open
- 选择 Simulation Only 并单击 OK 按钮，添加测试文件到工程
- 在 Simulation Run Time 选项下输入值 25000ns 并单击 OK 按钮
- 选中 Simulation 模式，双击 Simulate Behavioral Model 仿真设计。





5. 实现设计

实现设计过程中将会生成一些报告

- 在 Sources 窗口中，选中顶层设计文件
- 在 Processes 窗口中双击 Implement Design，ISE 工具会运行所有需要的过程来设计

五、实验结果分析

(一) Int_test.psm 处理流程说明（代码及注释）

Interrupt Service Routine

In the assembler log file for the example, it can be seen that the interrupt service routine has been forced to compile at address '2B0', and that the waveform generation is located in the base addresses. This makes it easier to observe the interrupt in action in the operation waveforms. This program is supplied as 'int_test.psm' for you to assemble yourself.

```

000                                ;Interrupt example
000                                ;
000                                CONSTANT waveform_port, 02                ;bit0 will be data
000                                CONSTANT counter_port, 04
000                                CONSTANT pattern_10101010, AA
000                                NAME REG sA, interrupt_counter
000                                ;
000                                start: LOAD interrupt_counter[sA], 00        ;reset interrupt counter
001                                002AA LOAD s2, pattern_10101010[AA]        ;initial output condition
002                                3C001 ENABLE INTERRUPT
003                                ;
003                                2C202 drive_wave: OUTPUT s2, waveform_port[02]
004                                00007 LOAD s0, 07                        ;delay size
005                                1C001 loop: SUB s0, 01                    ;delay loop
006                                35405 JUMP NZ, loop[005]
007                                0E2FF XOR s2, FF                        ;toggle waveform
008                                34003 JUMP drive_wave[003]
009                                ;
010                                ADDRESS 2B0
011                                2B0 18A01 int_routine: ADD interrupt_counter[sA], 01 ;increment counter
012                                2CA04 OUTPUT interrupt_counter[sA], counter_port[04]
013                                2B2 38001 RETURNI ENABLE
014                                2B3 ;
015                                3FF ADDRESS 3FF
016                                3FF 342B0 JUMP int_routine[2B0]
  
```

Main program delay loop where most time is spent

Interrupt Service Routine (located at address 2B0 onwards)

Interrupt vector set at address 3FF and causing JUMP to service routine

```

000                ;Interrupt example
000                ;
000                CONSTANT waveform_port, 02      ;定义波形输出端口 bit0 为数据
000                CONSTANT counter_port, 04       ; 定义计数器输出端口
000                CONSTANT pattern_10101010, AA   ; 定义波形模式 10101010
000                NAMEREG sA, interrupt_counter ; 将寄存器 sA 命名为
interrupt_counter
000                ;
000 00A00          start: LOAD interrupt_counter, 00      ;复位 interrupt counter 为 0
001 002AA          LOAD s2, pattern_10101010[AA]        ;初始 s2 为波形模式 10101010
002 3C001          ENABLE INTERRUPT                    ; 启用中断
003                ;
003 2C202          drive_wave: OUTPUT s2, waveform_port[02] ; 输出 s2 的值到波形端口
004 00007          LOAD s0, 07                          ;delay size
005 1C001          loop: SUB s0, 01                      ;delay loop
006 35405          JUMP NZ, loop[005]                    ; s0 不为 0 则跳回 loop
007 0E2FF          XOR s2, FF                            ;反转 waveform
008 34003          JUMP drive_wave[003]
009                ;
2B0                ADDRESS 2B0 ; 设置中断处理程序的地址
2B0 18A01          int_routine: ADD interrupt_counter[sA], 01 ;increment counter
2B1 2CA04          OUTPUT interrupt_counter[sA], counter_port[04]
2B2 38001          RETURNI ENABLE ; 返回并启用中断
2B3                ;
3FF                ADDRESS 3FF                          ;设置中断向量的地址
3FF 342B0          JUMP int_routine[2B0]

```

(二) Int_test.psm 功能分析

主程序功能：

初始化：

复位中断计数器 (interrupt_counter)，设置初始波形模式为 10101010 (AA)，启用中断。

方波生成：

通过端口 02 循环输出 AA 和 55 (异或 FF 翻转后的值)，形成周期性方波。

使用延时循环 (s0 从 7 递减至 0) 控制波形切换频率，产生固定周期的信号。

中断处理功能：

中断触发：

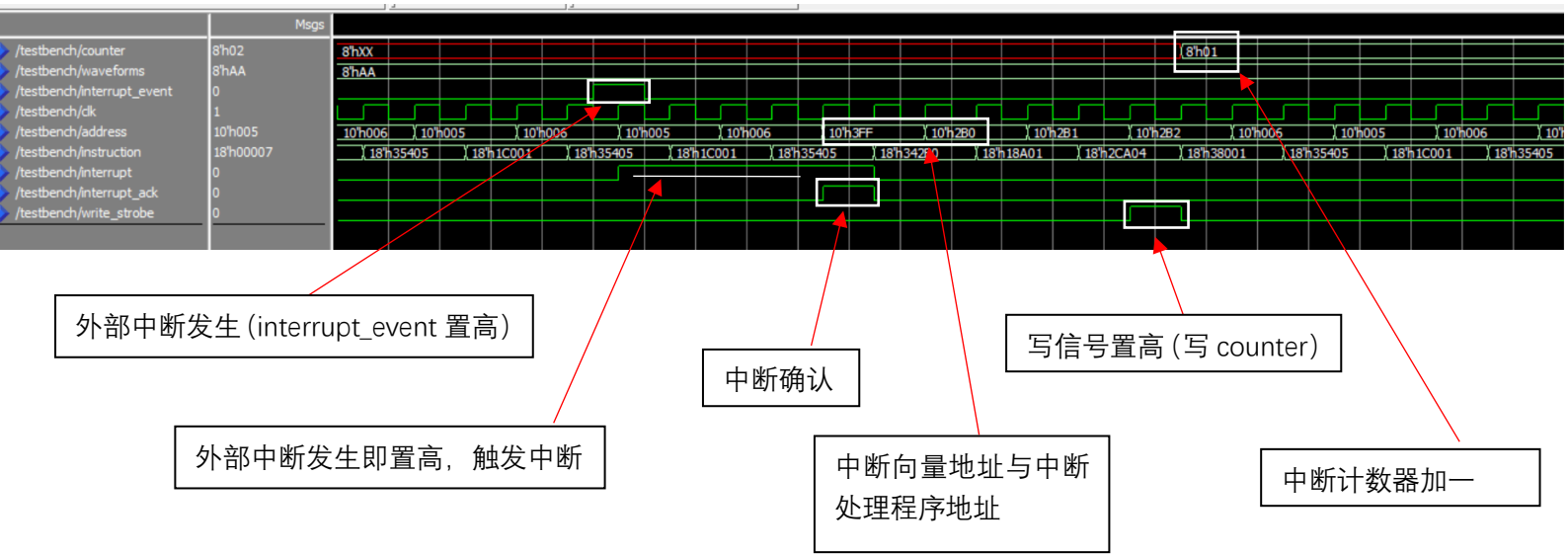
当外部中断 (interrupt_event 置高) 发生时，跳转到地址 2B0 执行中断服务程序。

计数与显示：

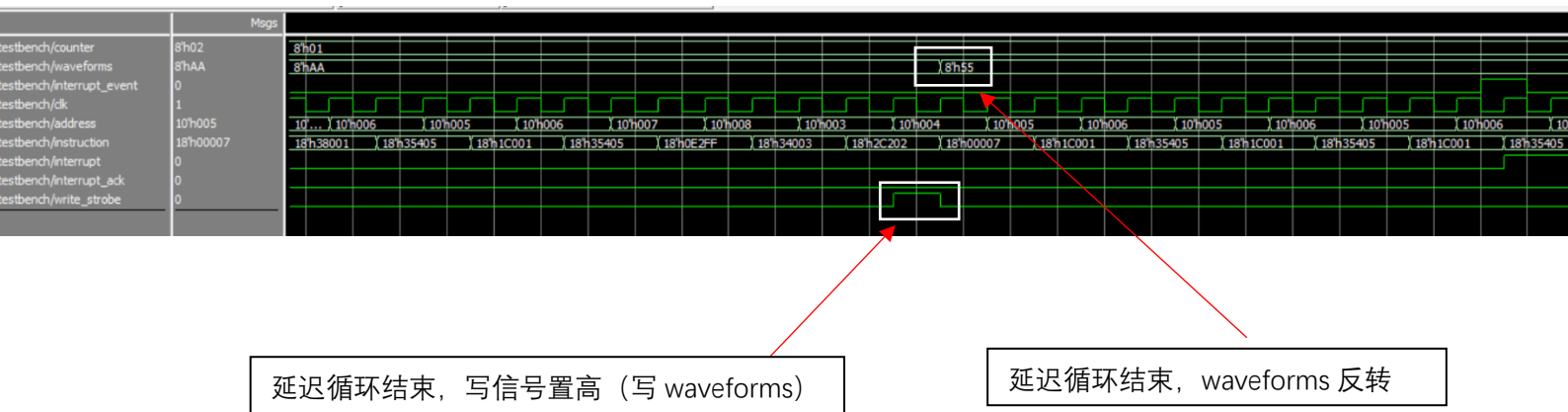
中断计数器 (interrupt_counter) 加 1，记录中断次数。

将当前计数值通过端口 04 实时输出，用于外部显示或监测 (counter)

(三) 中断时波形分析图波形分析图的展示与分析



(四) 延迟循环结束时波形分析图的展示与分析：



六、实验总结

本实验基于 KCPSM3 控制器成功实现了中断功能的验证与信号监控，使得我掌握了 Xilinx ISE 工具链的完整开发流程，包括工程创建、代码编译、仿真调试与设计实现。

验证了外部中断触发后，程序能正确跳转至中断服务程序（地址 2B0），完成计数器累加与端口输出，并实现中断返回。也通过波形仿真确认了信号时序的合理性，如 interrupt_event 触发后 interrupt_ack 的短暂置高、write_strobe 与端口数据更新的同步性。

在此过程中我巩固了 FPGA 开发工具的使用技能，更深化了对中断机制与硬件-软件协同设计的理解。