



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：第七章实验—int_test.psm 程序流程图
以及功能分析

任课教师：沈沛意老师

学号姓名：

提交日期：

一、 实验环境

win11

ISE14.7, Modelsim SE-64 10.4

二、 实验介绍

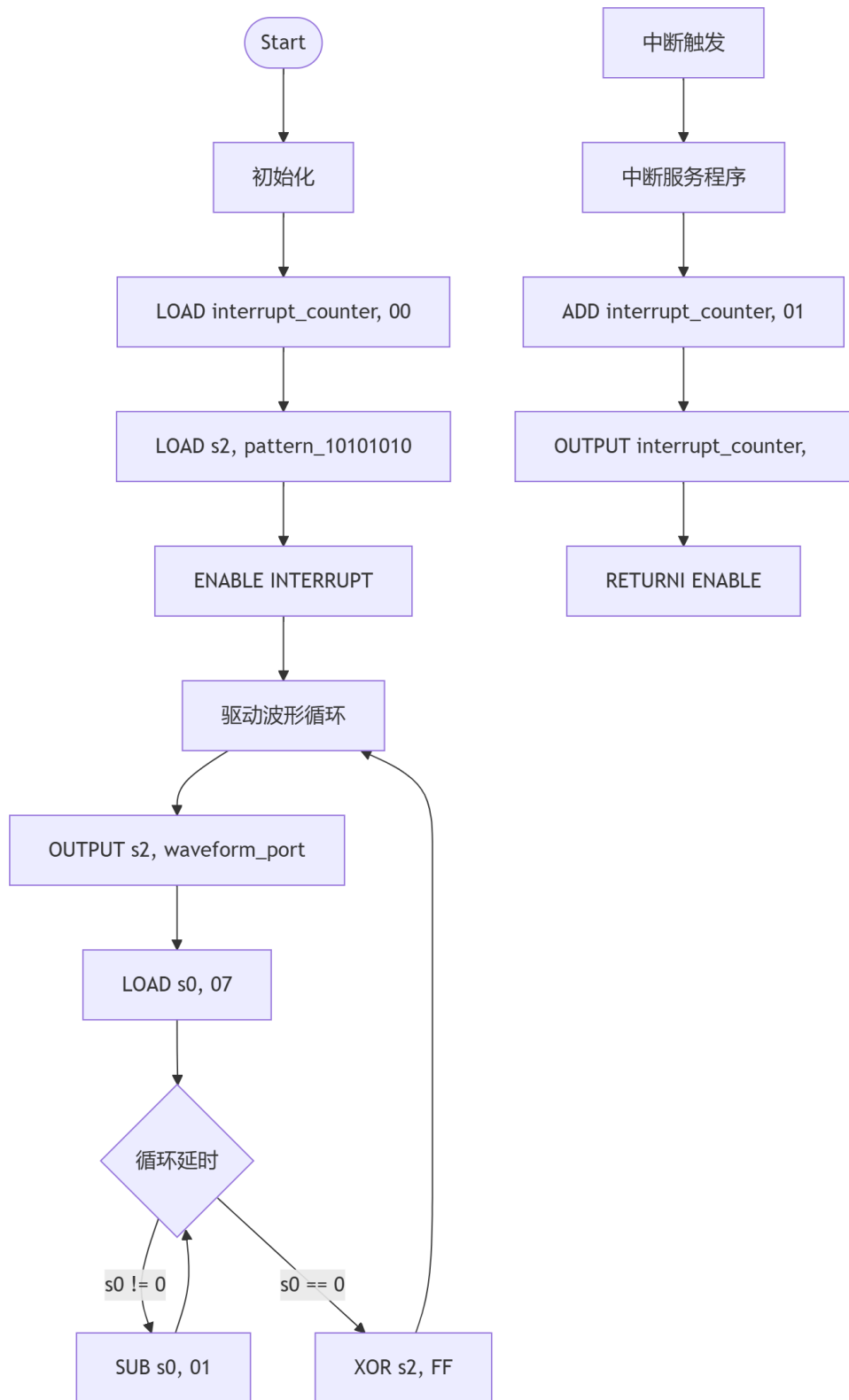
第七章实验一 int_test.psm 程序流程图以及功能分析。

三、 实验目标

- 绘制第七章实验一中 int_test.psm 的程序流程图
- 对此程序做功能分析
- 参考 PicoBlaze 8-bit Embedded Microcontroller User Guide 和 KCPSM3

manual

四、 实验流程图



五、 程序功能分析

1. 硬件交互定义

waveform_port (02): 用于输出方波信号的端口 (bit0 为数据位)。

counter_port (04): 用于输出中断计数器的端口。

pattern_10101010 (AA): 初始波形模式 (二进制 10101010)。

2. 主程序逻辑

初始化阶段:

将中断计数器 interrupt_counter 清零。

寄存器 s2 加载初始波形模式 AA。

启用全局中断 (ENABLE INTERRUPT)。

波形生成循环:

输出波形: 将 s2 的值写入 waveform_port, 生成方波信号。

延时控制: 通过递减寄存器 s0 (初始值 07) 实现简单延时循环。

波形翻转: 通过 XOR s2, FF 翻转 s2 的所有位, 使波形在 AA (10101010) 和 55 (01010101) 之间切换, 形成周期性方波。

3. 中断服务程序

触发条件: 当外部中断 (如定时器溢出或外部信号) 发生时, 程序跳转到中断向量地址 3FF, 执行 int_routine。

中断处理:

中断计数器 interrupt_counter 加 1。

将最新的计数值输出到 counter_port。

通过 RETURNI ENABLE 返回主程序并重新启用中断。

4. 硬件行为

波形输出: 通过 waveform_port 持续输出方波信号, 可用于驱动 LED、

蜂鸣器或其他外设。

中断计数：每次中断发生时，counter_port 更新当前中断次数，可用于监控或调试。

5. 关键设计

异步中断与主循环：主程序在生成波形时，中断可随时插入执行，实现非阻塞式计数。

延时机制：通过软件循环（SUB s0, 01）控制波形切换频率，实际应用中可能替换为硬件定时器。

六、 实验代码

```
    ; 硬件端口/常量定义

    ; 定义波形输出端口地址为 02 (bit0 为数据位)
    CONSTANT waveform_port, 02

    ; 定义中断计数器输出端口地址为 04
    CONSTANT counter_port, 04

    ; 定义初始波形模式为 10101010 (二进制, 对应十六进制 AA)
    CONSTANT pattern_10101010, AA

    ; 将寄存器 sA 命名为 interrupt_counter (中断计数器)
    NAMEREG sA, interrupt_counter

    ;-----

    ; 主程序入口

start:

    ; 初始化中断计数器为 0
    LOAD interrupt_counter, 00

    ; 加载初始波形模式到寄存器 s2 (AA = 10101010)
    LOAD s2, pattern_10101010

    ; 启用全局中断 (允许硬件中断触发)
```

ENABLE INTERRUPT

```
;-----  
  
; 波形生成主循环  
drive_wave:  
    ; 将 s2 的值输出到波形端口 (生成当前波形)  
    OUTPUT s2, waveform_port  
    ; 设置延时循环计数器 s0 初始值为 7 (十进制)  
    LOAD s0, 07  
loop:  
    ; 递减延时计数器 s0  
    SUB s0, 01  
    ; 如果 s0 不为零, 继续循环 (实现延时)  
    JUMP NZ, loop  
    ; 将 s2 的值与 FF 异或 (翻转所有 bit, AA→55 或 55→AA)  
    XOR s2, FF  
    ; 跳回波形生成循环, 持续输出波形  
    JUMP drive_wave  
;-----  
  
; 中断服务程序 (地址 2B0h)  
  
ADDRESS 2B0          ; 设置代码起始地址为 2B0h  
int_routine:  
    ; 中断计数器加 1 (sA += 1)  
    ADD interrupt_counter, 01  
    ; 将当前计数值输出到计数器端口  
    OUTPUT interrupt_counter, counter_port  
    ; 中断返回并重新启用中断 (恢复现场+允许新中断)  
    RETURNI ENABLE  
;-----  
  
; 中断向量设置 (地址 3FFh)
```

ADDRESS 3FF ; 设置中断向量地址为 3FFh

; 当硬件中断发生时, 跳转到 int_routine 执行

JUMP int_routine