《FPGA与CPU》课程

设计报告

<姓名>

北京至芯科技FPGA培训课程

2016-6-8

目录

[一、RISC\_SPM 3](#_Toc453235966)

[1.1 顶层设计 3](#_Toc453235967)

[1.2 指令系统设计 3](#_Toc453235968)

[1.3 架构设计 3](#_Toc453235969)

[1.4 控制器设计 4](#_Toc453235970)

[1.4.1 取指周期 4](#_Toc453235971)

[1.4.2 算术运算指令 5](#_Toc453235972)

[1.4.3 数据读指令 6](#_Toc453235973)

[1.4.4 数据写指令 8](#_Toc453235974)

[1.4.5 无条件转移指令 10](#_Toc453235975)

[1.4.6 零标志转移指令 11](#_Toc453235976)

[1.4.7 空操作转移指令 11](#_Toc453235977)

# 一、RISC\_SPM

基本需求：

1. 8位CPU，8位存储器，存储器位于FPGA芯片内部（使用Memory资源）
2. 基本按照Cillte的原架构
3. 扩充一个8位输出端口，用于控制七段数码管的七段码（seg）
4. 扩充一个4位输出端口，用于控制七段数码管的选择（Sel），或者取得LED
5. 通过Testbench验证简单的手编程序（汇编）和其他程序

## 1.1 顶层设计

HALT

clk

risc\_spm

p0[7:0]

rst\_n

p1[3:0]

## 1.2 指令系统设计

## 1.3 架构设计

## 1.4 控制器设计

### 1.4.1 取指周期

路径分析：

TP图：

instr

m\_out

addr

pc

sel\_mux1

sel\_mux2

load\_addr

sel\_mux2

load\_ir

write

pc(+1)

inc\_pc

取指周期

等效节点说明：

1. N1=MUX1+MUX2+ADDR
2. N2=MUX2+IR

SMF：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节拍 | LSM\_1S | LSM\_2S | | | Note |
| Assert | Deassert | Uncycle |
| 复位 | CNT=0 |  | Load\_addr=0  Inc\_pc=0  Load\_ir=0 | Sel\_mux1=`SEL\_PC  Sel\_mux2=`SEL\_BUS1  Write=0 | 取指周期 |
| 0 | CNT+1 | load\_addr=1  inc\_pc=1 |  | Sel\_mux1=`SEL\_PC  Sel\_mux2=`SEL\_BUS1 |
| 1 | CNT+1 |  | load\_addr=0  inc\_pc=0 | Write=0 |
| 2 | CNT+1 | load\_ir=1 |  | sel\_mux2=`SEL\_MEM |
| 3 | CNT+1 |  | load\_ir=0 |  |
| 4 | CNT+1 |  |  |  | 指令执行周期 |

### 1.4.2 算术运算指令

路径分析：

TP图：

算术运算指令的执行周期

y(src)

sel\_mux1

sel\_mux2

load\_y

instr

instr

sel\_mux1

sel\_mux2

load\_<dest>

SMF表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节拍 | LSM\_1S | LSM\_2S | | | Note |
| Assert | Deassert | Uncycle |
| 复位 | CNT=0 |  | Load\_addr=0  Inc\_pc=0  Load\_ir=0 | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1  Write=0 | 取指周期 |
| 0 | CNT+1 | load\_addr=1  inc\_pc=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 |
| 1 | CNT+1 |  | load\_addr=0  inc\_pc=0 | Write=0 |
| 2 | CNT+1 | load\_ir=1 |  | sel\_mux2=`SEL\_MEM |
| 3 | CNT+1 |  | load\_ir=0 |  |
| 4 | CNT+1 | load\_y=1 |  | sel\_mux1=`SEL\_<Src>  sel\_mux2= `SEL\_BUS1 | 算术运算指令  ADD  SUB  AND  NOT |
| 5 | CNT+1 | load\_<dest>=1  load\_z=1 | load\_y=0 | sel\_mux1=`SEL\_<dest>  sel\_mux2=`SEL\_ALU |
| 6 | CNT=0 |  | load\_<dest>=0  load\_z=0 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

等效节点说明：

N1=MUX1+MUX2+Y

N2的等效参考《设计与实践V17》113页，关于等效算法参考《同步电路》的相关课程

### 1.4.3 数据读指令

路径分析:

N1节点：

pc

addr

bus2

ADDR

bus1

MUX2

MUX1

PC

sel\_mux1

sel\_mux2

N1

load\_addr

N2节点：

addr

bus2

ADDR

MUX2

m\_out

MEM

ADDR

addr

sel\_mux2

N2

load\_addr

N3节点：

<dest>

<dest>

MUX2

bus2

MEM

m\_out

ADDR

addr

sel\_mux2

N3

load\_<dest>

TP图（控制信号的潜伏期分析）：

addr

pc

instr

addr

m\_out

sel\_mxu1

sel\_mux2

load\_addr

sel\_mux2

load\_addr

write

addr

m\_out

sel\_mux2

load\_<dest>

write

<dest>

TP图（程序指针修正的潜伏期分析）：

inc\_pc

dest

instr

pc(已经被修正)

pc修正的游程

SMF表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节拍 | LSM\_1S | LSM\_2S | | | Note |
| Assert | Deassert | Uncycle |
| 复位 | CNT=0 |  | Load\_addr=0  Inc\_pc=0  Load\_ir=0 | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1  Write=0 | 取指周期 |
| 0 | CNT+1 | load\_addr=1  inc\_pc=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 |
| 1 | CNT+1 |  | load\_addr=0  inc\_pc=0 | Write=0 |
| 2 | CNT+1 | load\_ir=1 |  | sel\_mux2=`SEL\_MEM |
| 3 | CNT+1 |  | load\_ir=0 |  |
| 4 | CNT+1 | load\_addr=1  inc\_pc=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 | 读指令  RD |
| 5 | CNT+1 |  | load\_addr=0  write=0  inc\_pc=0 |  |
| 6 | CNT+1 | load\_addr=1 |  | sel\_mux2=`SEL\_MEM |
| 7 | CNT+1 |  | load\_addr=0  write=0 |  |
| 8 | CNT+1 | load\_<dest>=1 |  | sel\_mux2=`SEL\_MEM |
| 9 | CNT=0 |  | load\_<dest>=0 |  |

### 1.4.4 数据写指令

路径分析：

N1节点：(将指令低字节的地址送至地址寄存器)

load\_addr

sel\_mux2

sel\_mux1

addr

bus2

ADDR

bus1

MUX2

pc

MUX1

PC

N1

N2节点：（指令低地址中的数据地址送至地址寄存器）

addr

bus2

ADDR

MUX2

m\_out

MEM

ADDR

addr

sel\_mux2

N2

load\_addr

N3节点

m\_out

MEM

mem\_addr

ADDR

addr

<src>

MUX1

<src>

bus1

mem\_data

sel\_mux1

write

N3

TP图：（控制信号的潜伏期分析）

sel\_mux1

sel\_mux2

load\_addr

instr

m\_out

addr

m\_out

addr

sel\_mux1

write=1

sel\_mux2

load\_addr

write=0

TP图：（程序指针修正的潜伏期分析）

m\_out

inc\_pc

instr

pc(已经被修正)

下一指令取指周期

PC修正的游程(当前指针执行周期)

SMF表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节拍 | LSM\_1S | LSM\_2S | | | Note |
| Assert | Deassert | Uncycle |
| 复位 | CNT=0 |  | Load\_addr=0  Inc\_pc=0  Load\_ir=0 | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1  Write=0 | 取指周期 |
| 0 | CNT+1 | load\_addr=1  inc\_pc=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 |
| 1 | CNT+1 |  | load\_addr=0  inc\_pc=0 | Write=0 |
| 2 | CNT+1 | load\_ir=1 |  | sel\_mux2=`SEL\_MEM |
| 3 | CNT+1 |  | load\_ir=0 |  |
| 4 | CNT+1 | load\_addr=1  inc\_pc=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 | 写指令  WR |
| 5 | CNT+1 | write=0 | load\_addr=0  inc\_pc=0 |  |
| 6 | CNT+1 | load\_addr=1 |  | sel\_mux2=`SEL\_MEM |
| 7 | CNT+1 | write=1 | load\_addr=0 | sel\_mux1=<src> |
| 8 | CNT=0 |  | write=0 |  |
|  |  |  |  |  |  |

### 1.4.5 无条件转移指令

等效节点分析：

N1节点：（将指令低字节地址送地址寄存器）

sel\_mux2

sel\_mux1

addr

bus2

ADDR

bus1

MUX2

pc

MUX1

PC

N1

load\_addr

N2节点：（将位于指令低字节单元中的转移地址取出，送至PC）

bus2

PC

MUX2

pc

m\_out

MEM

ADDR

addr

load\_pc

sel\_mux2

N2

TP图（控制信号潜伏期分析）：

inc\_pc

instr

sel\_mux1

sel\_mux2

load\_addr

m\_out

addr

sel\_mux2

load\_pc

write=0

SMF表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节拍 | LSM\_1S | LSM\_2S | | | Note |
| Assert | Deassert | Uncycle |
| 复位 | CNT=0 |  | Load\_addr=0  Inc\_pc=0  Load\_ir=0 | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1  Write=0 | 取指周期 |
| 0 | CNT+1 | load\_addr=1  inc\_pc=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 |
| 1 | CNT+1 |  | load\_addr=0  inc\_pc=0 | Write=0 |
| 2 | CNT+1 | load\_ir=1 |  | sel\_mux2=`SEL\_MEM |
| 3 | CNT+1 |  | load\_ir=0 |  |
| 4 | CNT+1 | load\_addr=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 | 无条件转移指令  BR |
| 5 | CNT+1 | write=0 | load\_addr=0 |  |
| 6 | CNT+1 | load\_pc=1 |  | sel\_mux2=`SEL\_MEM |
| 7 | CNT=0 |  | load\_pc=0 |  |

### 1.4.6 零标志转移指令

当z\_flag为真执行BR指令的SMF表

当z\_flag为假执行下表（PC修正）：

SMF表（z\_flag=0）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节拍 | LSM\_1S | LSM\_2S | | | Note |
| Assert | Deassert | Uncycle |
| 复位 | CNT=0 |  | Load\_addr=0  Inc\_pc=0  Load\_ir=0 | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1  Write=0 | 取指周期 |
| 0 | CNT+1 | load\_addr=1  inc\_pc=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 |
| 1 | CNT+1 |  | load\_addr=0  inc\_pc=0 | Write=0 |
| 2 | CNT+1 | load\_ir=1 |  | sel\_mux2=`SEL\_MEM |
| 3 | CNT+1 |  | load\_ir=0 |  |
| 4 | CNT+1 | inc\_pc=1 |  |  | 零标志转移指令  BRZ |
| 5 | CNT=0 |  | inc\_pc=0 |  |

### 1.4.7 空操作转移指令

SMF表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 节拍 | LSM\_1S | LSM\_2S | | | Note |
| Assert | Deassert | Uncycle |
| 复位 | CNT=0 |  | Load\_addr=0  Inc\_pc=0  Load\_ir=0 | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1  Write=0 | 取指周期 |
| 0 | CNT+1 | load\_addr=1  inc\_pc=1 |  | sel\_mux1=`SEL\_PC  sel\_mux2=`SEL\_BUS1 |
| 1 | CNT+1 |  | load\_addr=0  inc\_pc=0 | Write=0 |
| 2 | CNT+1 | load\_ir=1 |  | sel\_mux2=`SEL\_MEM |
| 3 | CNT+1 |  | load\_ir=0 |  |
| 4 | CNT=0 |  |  |  | 空操作指令 |