廣西大學

计算机组成原理课程设计

课题名称 硬连线控制器的常规 CPU 设计

学院： 计算机与电子信息学院

专业： 计算机类

班级： [redacted]

学号： [redacted]

姓名： [redacted]

指导老师： [redacted]

二〇一九年六月二十日

目 录

第一章 课程设计简述

1.1 教学目的

1、融会贯通计算机组成原理与体系结构课程各章教学内容，通过知识的综合运用，加深对 CPU 各模块工作原理及相互联系的认识；
2、掌握硬连线控制器的设计方法；
3、学习运用 EDA 设计工具，掌握用 EDA 设计大规模复杂逻辑电路的方法；
VHDL：超高速集成电路硬件描述语言 QuartusⅡ：
4、培养科学研究能力，取得设计和调试的实践经验。

1.2 课设任务

1、设计一个硬连线控制器，和 TEC-8 模型计算机的数据通路结合在一起，构成一个完整的 CPU，该 CPU 要求：

① 能够完成控制台操作：启动程序运行、读存储器、写存储器、读寄存器和写寄存器。

② 能够执行表 1 中的指令，完成规定的指令功能。

2、在 QuartusⅡ下对硬布线控制器设计方案进行编程和编译。
3、在编译后的硬布线控制器下载到 TEC-8 实验台上的 ISP 器件 EPM7128 中去，使 EPM7128 成为一个硬布线控制器。
4、根据指令系统，编写检测硬连线控制器正确性的测试程序，并用测试程序对硬布线控制器在单拍方式下进行测试，直到成功。
5、在调试成功的基础上，整理出设计文件。

① 硬连线控制器逻辑模块图；
② 硬连线控制器指令周期流程图；
③ 硬连线控制器的硬件描述语言源程序；
④ 测试程序；
⑤ 设计说明书；
⑥ 调试总结。

1.3 实验设备

1、TEC-8 实验系统　　1 台
2、PC 计算机　　1 台
3、双踪示波器　　1 台
4、直流万用表　　1 个
5、逻辑测试笔（在 TEC-8 实验台上）　　1 支

1.4 小组分工

表 1-1 小组分工

学号	姓名	主要完成内容	备注
[redacted]	[redacted]	设计实验、编写代码、绘制流程图	设计
[redacted]	[redacted]	实操上机调试、演示实验	实操
[redacted]	[redacted]	编写测试程序、仿真模拟	模拟
[redacted]	[redacted]	编写小组报告、后期处理	后期
[redacted]	[redacted]	编写测试程序、仿真模拟	模拟

1.5 课程设计的时间安排

表 1-2 时间安排

序号	顺序	完成内容	用时（天）	性质
1	准备阶段	了解实验目的的要求、查阅资料	1	理解
2	设计阶段	硬件、软件的设计，熟悉写存储器、读存储器、写寄存器、读寄存器等过程	7	设计

3	上机阶段	编写微程序、上机运行程序	4	设计
4	编写报告	检查试验效果，总结实验经验、编写实验报告	4	验证

第二章 总体设计思路

2.1 指令系统

为完成课程设计的内容要求、简化控制信号逻辑表达式，使用的指令系统及其相应编码如下：

2.1.1 机器指令

表 2-1 新 CPU 机器指令

名 称	助 记 符	功 能	指 令 格 式		
			IR7 IR6 IR5 IR4	IR3 IR2	IR1 IR0
加法	ADD Rd, Rs	Rd ← Rd + Rs	0001	Rd	Rs
减法	SUB Rd, Rs	Rd ← Rd - Rs	0010	Rd	Rs
逻辑与	AND Rd, Rs	Rd ← Rd and Rs	0011	Rd	Rs
加 1	INC Rd	Rd ← Rd + 1	0100	Rd	XX
取数	LD Rd, [Rs]	Rd ← [Rs]	0101	Rd	Rs
存数	ST Rs, [Rd]	Rs → [Rd]	0110	Rd	Rs
C 条件转移	JC offset	若 C=1，则 PC ← @ + offset	0111	offset	
Z 条件转移	JZ offset	若 Z=1，则 PC ← @ + offset	1000	offset	
无条件转移	JMP Rd	PC ← Rd	1001	Rd	XX

输出	OUT Rs	DBUS ← Rs	1010	XX	Rs
自加	ADD Rd，Rd	Rd←Rd+Rd	1011	Rd	XX
异或	Rd⊕Rs	Rd←Rd⊕Rs	1100	Rd	Rs
或	Rd+Rs	Rd←Rd+Rs	1101	Rd	Rs
同或	Rd⊙Rs	Rd←Rd⊙Rs	1111	Rd	Rs

表 2-1 中，XX 代表任意值，Rs 代表源寄存器号，Rs 代表目的寄存器号。在条件转移指令中，@代表当前 PC 的值，offset 是一个四位的有符号数，第三位是符号位，0 代表正数，1 代表负数。注意：@不是当前指令的 PC 值，而是当前指令的 PC 值加 1。

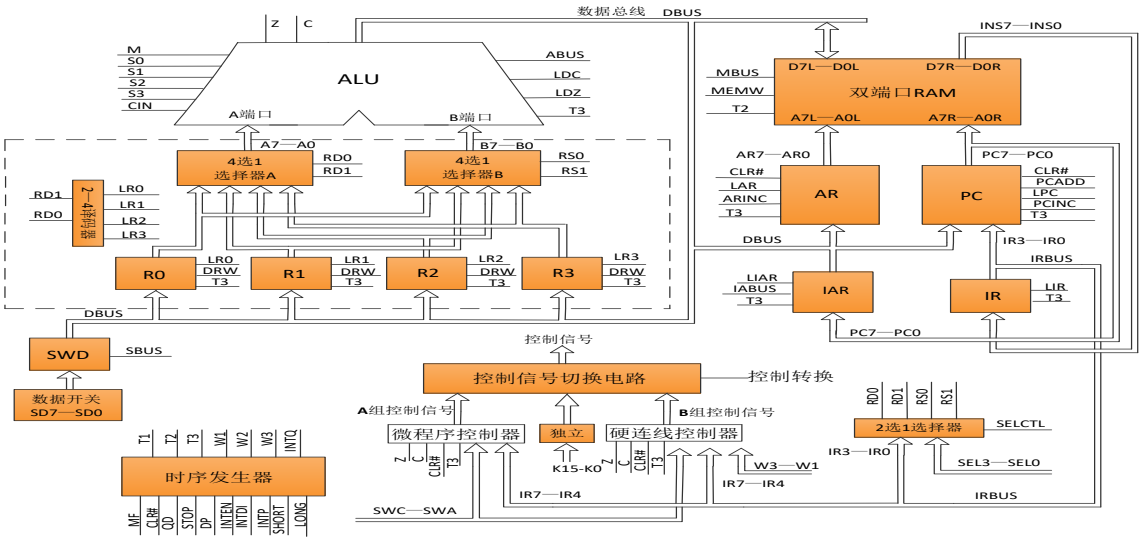
2.1.2 控制台指令

表 2-2 控制台指令

SWC	SWB	SWA	工作方式
0	0	0	PR，启动程序
0	0	1	KRD，读取端口存储器
0	1	0	KWE,写双端口存储器
0	1	1	KLD,加载寄存器堆
1	0	0	KRR,读寄存器堆

2.2 硬连线控制器逻辑模块图

本实验要求设计一个硬连线控制器，和 TEC-8 模型计算机的数据通路结合在一起，构成一个完整的 CPU。我们需要针对 TEC-8 模型计算机的特性来设计硬连线控制器。

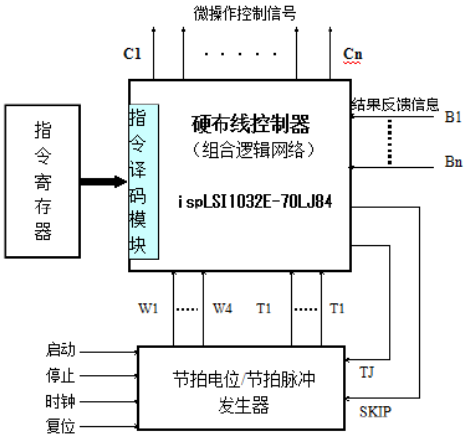


TEC-8

模型计算机框图 2-1

通过观察 TEC-8 的框图可知，硬连线控制器受译码器输出信号（SWC-SWA、IR7-IR4）、节拍电位信号（T3）、状态条件信号（Z、C）、时序发生器产生的节拍脉冲信号（W3-W1）及 CLR#

图 2—2



硬布线控制器结构方框图

第三章 设计与调试方案

3.1 设计步骤

3.1.1 硬连线控制器指令周期流程图

硬连线控制器以节拍信号作为电位，时序发生器产生的节拍脉冲信号 W3-W1 为时间单位。根据所要求的指令功能进行硬连线控制器指令周期的设计：

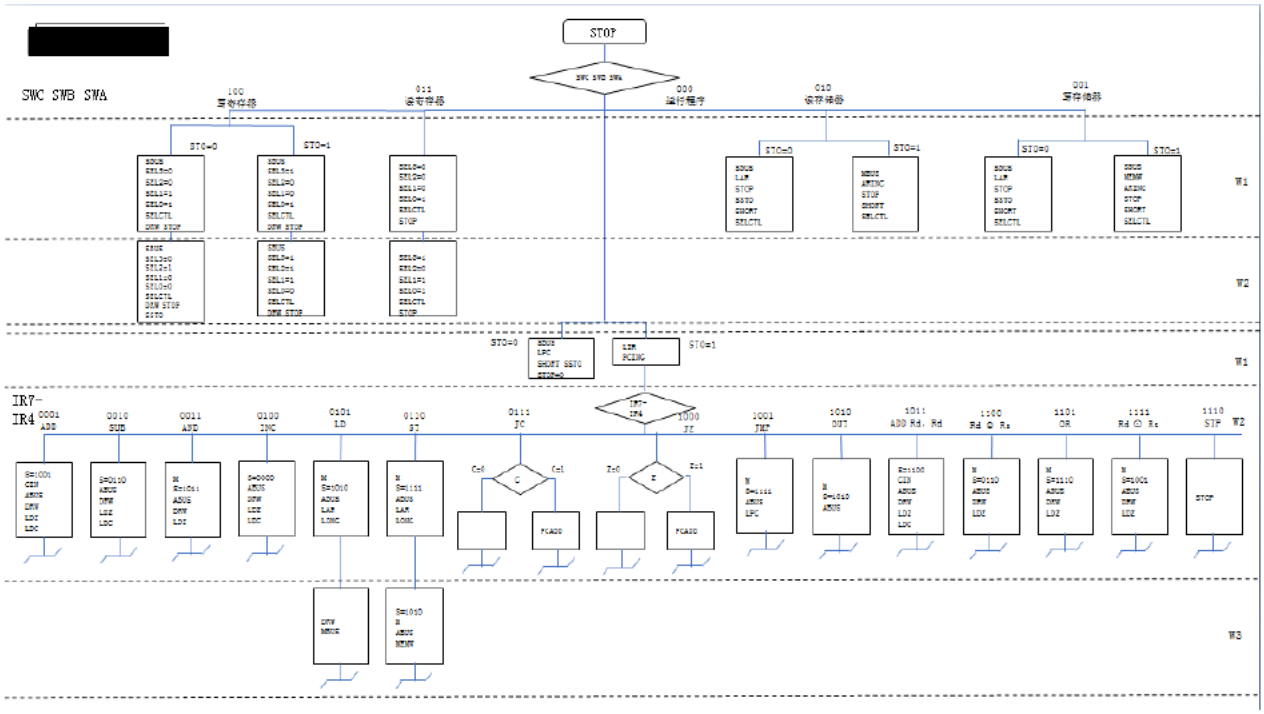


图 3—1

本图中一个执行框代表一个节拍电位时间， TEC-8 实验系统中采用了可变节拍数来执行一条机器指令。

3.1.2 组合逻辑译码表

表 3-1 组合逻辑译码表

IR	ADD	SUB	AND	INC	LD	ST	JC	JZ	JMP	OUT	自加	异或	或	同或
LIR	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1
M			W2		W2	W2/W3	W2	W2	W2	W2	W2	W2	W2	W2
S3	W2		W2		W2	W2/W3			W2		W2		W2	W2
S2		W2				W2			W2	W2		W2	W2	

S1		W2	W2		W2	W2/W3			W2			W2	W2	
S0	W2		W2			W2			W2					W2
CIN	W2										W2			
LDC	W2	W2		W2							W2			
LDZ	W2	W2	W2	W2							W2	W2	W2	W2
DRW	W2	W2	W2	W2	W3	W3					W2	W2	W2	W2
ABUS	W2	W2		W2	W2	W2/W3			W2	W2	W2	W2	W2	W2
LAR					W2	W2								
PCADD							C,W2	Z,W2						
LPC									W2					
MBUS					W3									
MEMW						W3								
LONG					W2	W2								
STOP														
PCING	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1	W1

3.2 硬连线控制器的硬件描述语言源程序

3.2.1 源程序设计思路

使用硬件描述语言书写硬连线控制器源程序时，需根据设计好的指令周期框图进行编写。首先指定调用的程序包，并在程序起始声明控制器的输入输出变量，进行预设计，然后就可依次进行指令的编写了。

3.2.2 源程序代码

见附录

第四章 检验性试验

4.1 基础实验

4.1.1 预制寄存器及其存储器的内容

表 4-1 预制寄存器及其存储器内容

寄存器	R2=60H	R3=FDH	
存储器	[60H]=67H	[61H]=80H	[62H]=0FDH
	[80H]=60H	[0FEH]=03H	[0FFH]=03H

4.1.2 程序代码

表 4-2 程序代码

地址	指令	机器码	1	2	3
00H	LD， R0,[R2]	52H	R0=67H		
01H	INC,R2	48H	R2=61H		
02H	LD,R1,[R2]	56H	R1=80H		
03H	ADD,R0,R1	11H	R0=E7H	R0=07H	R0=86H
04H	JC,06H	71H		PC=06H	
05H	AND,R1,R0	34H	R1=80H		R1=82H
06H	SUB,R0,R2	22H	R0=86H	R0=04H	R0=83H
07H	INC,R1	44H	R1=81H	R1=82H	R1=83H
08H	STA,R0,[R1]	64H	[81H]=86H	[82H]=04H	[83H]=83H
09H	INC,R3	4CH	R3=FEH	R3=FFH	R3=00H
0AH	JZ,0DH	82H			PC=0DH
0BH	LD,R2[R3]	5BH	R2=03H	R2=03H	
0CH	JMP,[R2]	98H	PC=03H	PC=03H	
0DH	INC,R3	4CH			R3=01H
0EH	INC,R3	4CH			R3=02H
0FH	SUB,R0,R2	22H			R0=80H
10H	LD,R2,[R0]	58H			R2=60H

11H	ADD,R3,R2	1EH			R3=62H
12H	LD,R3[R3]	5FH			R3=FDH
13H	OUT,RO	OAOH			RO
14H	STP	OEOH			

4.1.3 执行结果

表 4-3 执行结果

读寄存器	RO=80H	R1=83H	R2=60H	R3=FDH
读存储器	[81H]=86H	[82H]=04H	[83H]=83H	

4.1.4 附加功能

表 4-4 附加功能(1)

寄存器 初始值	RO=80H	R1=83H	R2=60H	R3=FDH

表 4-5 附加功能(2)

	指令	机器码	Rd
输出	DBUS ← Rs	10100001	（DBUS）83H
自加	Rd←Rd+Rd	10111000	C0H
异或	Rd←Rd⊕Rs	11000111	7EH
或	Rd←Rd+Rs	11011011	FDH
同或	Rd←Rd⊙Rs	11110110	1CH

4.2 自备检验性试验

注意到若只进行基础实验，将有多个指令的正确性存在争议，因此本组设计了如下检验性试验：

4.2.1 制寄存器及其存储器的内容

表 4-6 预制寄存器及其存储器内容

寄存器	R2=60H	R3=64H	
存储器	[60H]=83H	[64H]=24H	

4.2.2 程序代码

表 4-7 程序代码

地址	指令	机器码	1	2	3
00	LD R0,[R2]	52H	R0=83H		
01	LD R1,[R3]	57H	R1=24H		
02	INC R2	48H	R2=61H		
03	SUB R0,R1	21H	R0=59H		
04	INC R0	40H	R0=60H		
05	STA R1,[R0]	21H	[60H]=24H		
06	ADD R0,R3	13H	R0=48H		
07	JC 09H	71H			
08	SUB R2,R0	28H	R2=13H		
09	JZ 10H	80H			
0A	AND R3,R0		R3=40H		
0B	JMP [R2]	98H	PC=13H		
0C	OUT R0	A0H			
0D	STP	E0H			

4.2.3 执行结果

表 4-8 执行结果

读寄存器	R0=48H	R1=24H	R2=13H	R3=40H
读存储器	[60H]=24H			

4.3 程序仿真

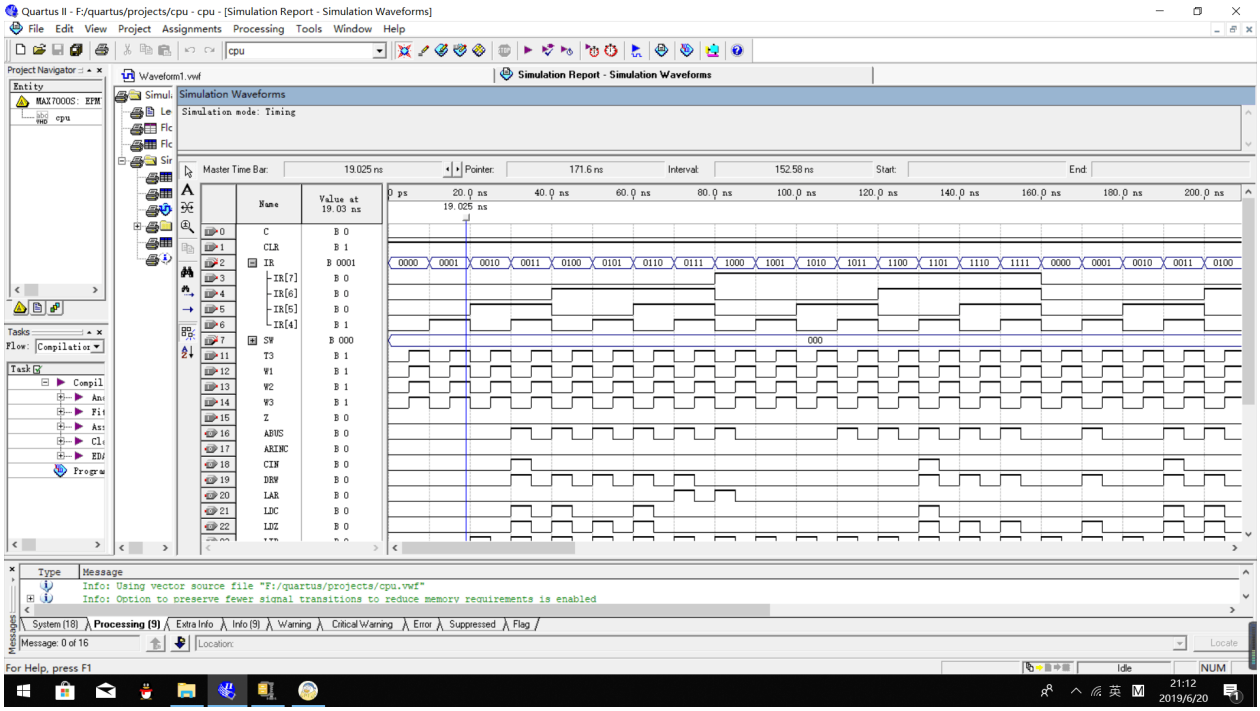


图 4-1

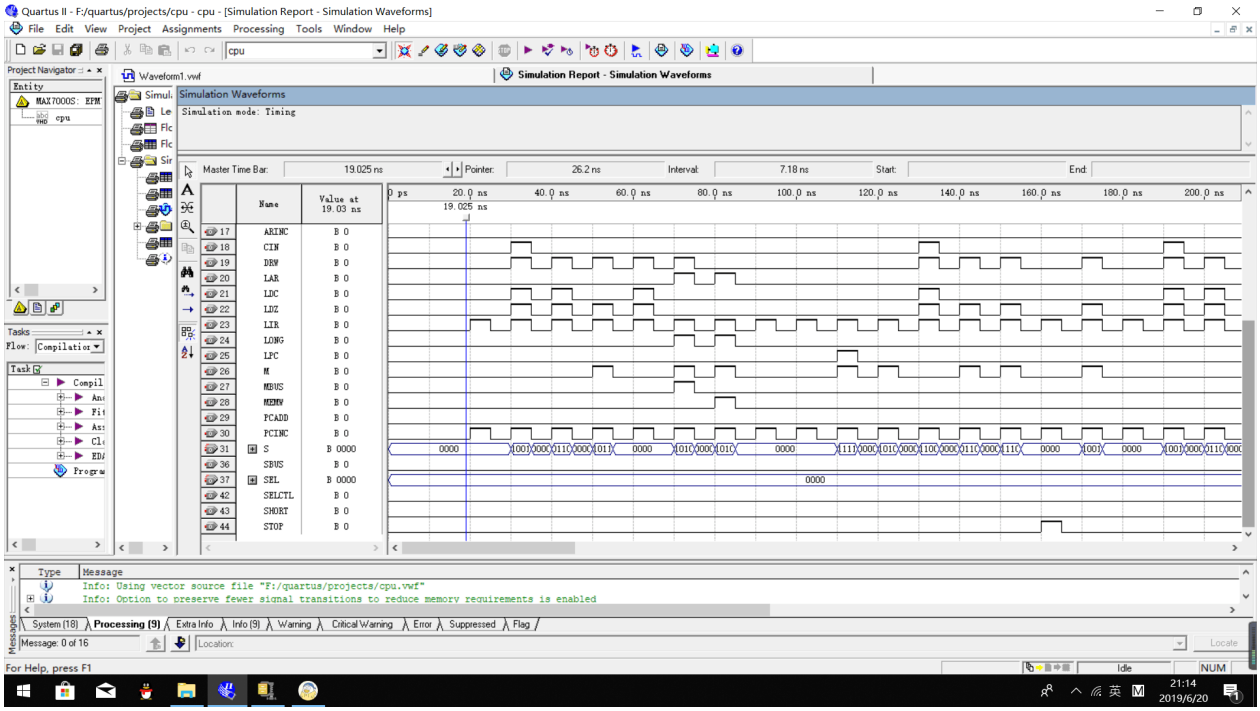


图 4-2

第五章 遇到的问题与体会

5.1 调试日志

在进行硬布线控制器的指令周期设计时，需注意参照已有类似设计进行改进与学习，在实践中完善和改进。

在进行使用 Quartus II 进行代码编写时，曾遇到许多问题：

1. 没有对应的芯片可供选择
2. 项目名和顶部文件名没有正常设置
3. 针脚没有定义

我的解决方案是：

1. 更换设计软件版本，安装元器件库
2. 将项目名称与顶部文件名设置相同
3. 参照实验书进行了芯片针脚定义

5.2 心得体会

进行课程设计不是一蹴而就的，需要大量的时间来学习，来验证。同时也需要合作，需要分工。感谢同组同学对我的大力支持和帮助，没有他们的支持与分工，课设就不能如此顺利的完成。同时感谢老师的指导与答疑。

参考文献

[1] TEC-4 模型计算机介绍.[Z]. 北京

[2] 白中英. 计算机组成原理（第五版·立体化教材）[M].北京.科学出版社，2013

附录

设计程序代码：

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY cpu IS
PORT(SW:IN std_logic_vector(2 Downto 0);                --向量形式简洁易懂
IR:IN std_logic_vector(7 Downto 4);
W1,W2,W3:IN std_logic;
C,Z,CLR,T3:IN std_logic;
ARINC,CIN,DRW,LPC,LAR,LIR,LDZ,LDC,PCINC,PCADD,SELCTL,M,MEMW,STOP:OUT std_logic;
SHORT,LONG,ABUS,SBUS,MBUS:OUT std_logic;
S,SEL:OUT std_logic_vector(3 Downto 0));
END cpu;



ARCHITECTURE cont OF cpu IS
SIGNAL ST0:std_logic;
SIGNAL SST0:std_logic;
BEGIN
PROCESS(ST0,SST0,SW,IR,CLR,T3,W1,W2,W3)     --敏感信号表

BEGIN
IF (CLR=0) THEN                             --clear 信号
      ST0<=0;
      SST0<=0;
      CIN<=0;                                   --初始值
      DRW<=0;
```

```
        LPC<=0;
        LAR<=0;
        LIR<=0;
        LDZ<=0;
        LDC<=0;
        PCINC<=0;
        PCADD<=0;
        SELCTL<=0;
        M<=0;
        S<="0000";
        SEL<="0000";
        MEMW<=0;
        STOP<=0;
        SHORT<=0;
        LONG<=0;
        ABUS<=0;
        SBUS<=0;
        MBUS<=0;
        ARINC<=0;
    ELSE
        CIN<=0;                                    --初始值

        DRW<=0;
        LPC<=0;
        LAR<=0;
        LIR<=0;
        LDZ<=0;
        LDC<=0 ;
        PCINC<=0;
        PCADD<=0;
        SELCTL<=0;
        M<=0;
        S<="0000";
        SEL<="0000";
```

```
MEMW<=0;
STOP<=0;
SHORT<=0;
LONG<=0;
ABUS<=0;
SBUS<=0;
MBUS<=0;
ARINC<=0;

SST0<=(NOT ST0)AND ((SW(2)AND(NOT(SW(1)))AND(NOT SW(0))AND W2)OR((NOT SW(2))AND
SW(1)AND(NOT SW(0))AND W1)OR((NOT SW(2))
        AND (NOT SW(1)) AND SW(0) AND W1));
--用译码形式表示 SST0

IF(SST0=1 ) THEN
    IF (T3EVENT AND (T3=0))THEN   --T3 下降沿
        ST0<=1;
        END IF;
        END IF;
        CASE SW IS
        --根据硬连线控制器参考流程图采用流程的形式

        WHEN "100" =>     --写寄存器
            IF (ST0=0) THEN
                IF (W1=1) THEN
                    SBUS<=1;
                    SEL<="0011";
                    SELCTL<=1;
                    DRW<=1;
                    STOP<=1;
                    END IF;
                IF (W2=1) THEN
                    SBUS<=1;
                    SEL<="0100";
```

```
                SELCTL<=1;
                DRW<=1;
                STOP<=1;
                SST0<=1;
            END IF;
        END IF;
        IF (ST0=1) THEN
            IF (W1=1) THEN
                SBUS<=1;
                SEL<="1001";
                SELCTL<=1;
                DRW<=1;
                STOP<=1;
            END IF;
            IF (W2=1) THEN
                SBUS<=1;
                SEL<="1110";
                SELCTL<=1;
                DRW<=1;
                STOP<=1;
            END IF;
        END IF;
    WHEN "011" =>                   --读寄存器
        IF (W1=1) THEN
            SEL<="0001";
            SELCTL<=1;
            STOP<=1;
        END IF;
        IF (W2=1) THEN
            SEL<="1011";
            SELCTL<=1;
            STOP<=1;
        END IF;
    WHEN "010"=> --读存储器
```

```
        IF (ST0=0) THEN
            IF (W1=1) THEN
                SBUS<=1;
                LAR<=1;
                STOP<=1;
                SHORT<=1;
                SELCTL<=1;
                SST0<=1;
            END IF;
        END IF;
        IF (ST0=1) THEN
            IF (W1=1) THEN
                MBUS<=1;
                ARINC<=1;
                STOP<=1;
                SHORT<=1;
                SELCTL<=1;
            END IF;
        END IF;
    WHEN "001"=> --写存储器
        IF (ST0=0) THEN
            IF (W1=1) THEN
                SBUS<=1;
                LAR<=1;
                STOP<=1;
                SHORT<=1;
                SELCTL<=1;
                SST0<=1;
            END IF;
        END IF;
        IF (ST0=1) THEN
            IF (W1=1) THEN
                SBUS<=1;
                MEMW<=1;
```

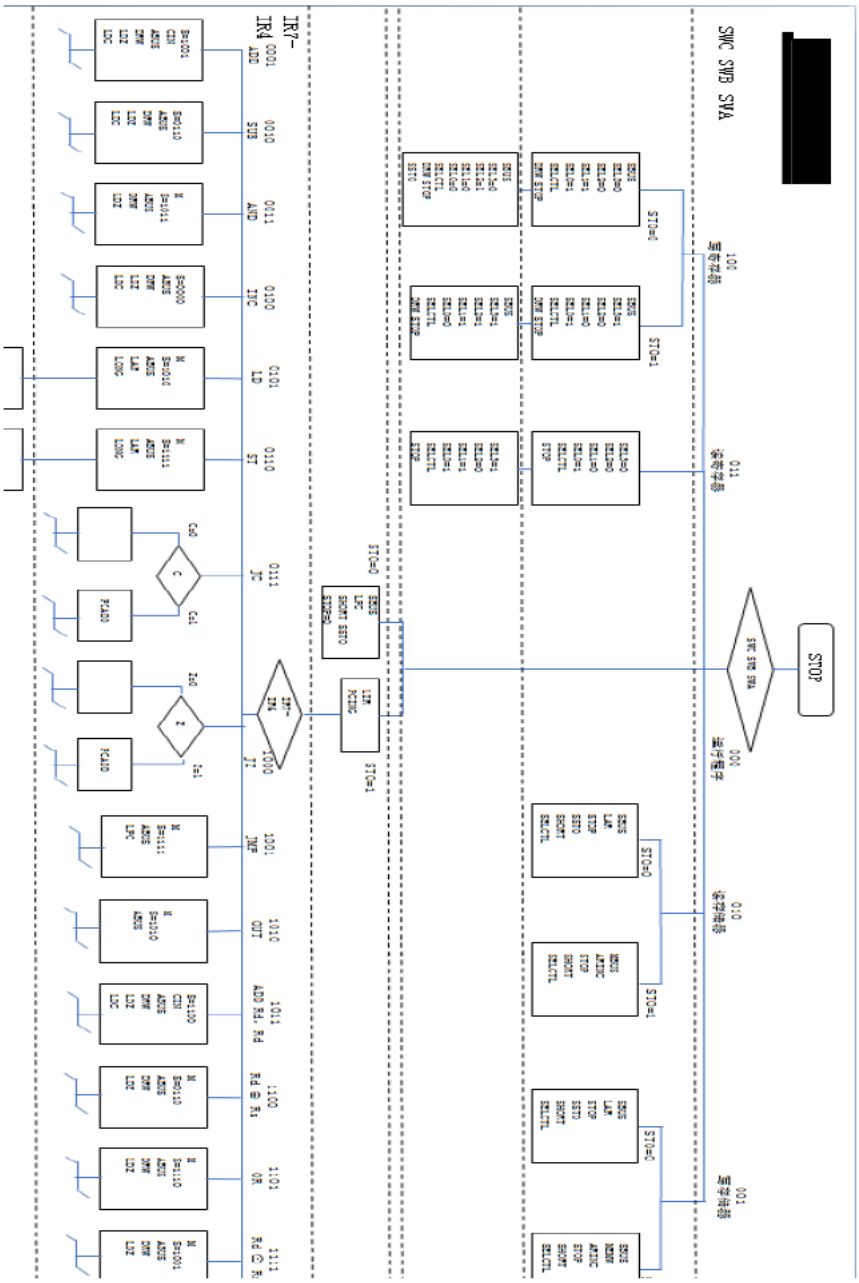
```
                ARINC<=1;
                STOP<=1;
                SHORT<=1;
                SELCTL<=1;
            END IF;
        END IF;
    WHEN "000"=> --取指
        IF (W1=1) THEN
            LIR<=1;
            PCINC<=1;
        END IF;
    CASE IR IS
    WHEN "0001"=> --ADD
        IF (W2=1) THEN
            S<="1001";
            CIN<=1;
            ABUS<=1;
            DRW<=1;
            LDZ<=1;
            LDC<=1;
        END IF;

    WHEN "0010"=> --SUB
        IF (W2=1) THEN
            S<="0110";
            ABUS<=1;
            DRW<=1;
            LDZ<=1;
            LDC<=1;
        END IF;
    WHEN "0011"=> --AND
        IF (W2=1) THEN
            M<=1;
            S<="1011";
```

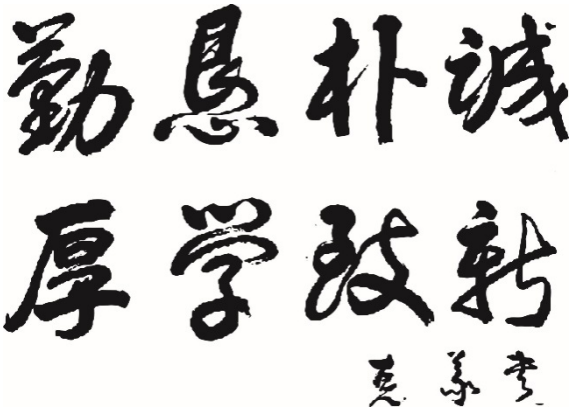
```
            ABUS<=1;
            DRW<=1;
            LDZ<=1;
        END IF;
    WHEN "0100"=> --INC
        IF (W2=1) THEN
            S<="0000";
            ABUS<=1;
            DRW<=1;
            LDZ<=1;
            LDC<=1;
        END IF;
    WHEN "0101"=> --LD
        IF (W2=1) THEN
            M<=1;
            S<="1010";
            ABUS<=1;
            LAR<=1;
            LONG<=1;
        END IF;
        IF (W3=1) THEN
            DRW<=1;
            MBUS<=1;
        END IF;
    WHEN "0110"=> --ST
        IF (W2=1) THEN
            M<=1;
            S<="1111";
            ABUS<=1;
            LAR<=1;
            LONG<=1;
        END IF;
        IF (W3=1) THEN
            S<="1010";
```

```
            M<=1;
            ABUS<=1;
            MEMW<=1;
        END IF;
    WHEN "0111"=> --JC
        IF (W2=1) THEN
            IF (C=1) THEN
                PCADD<=1;
            END IF;
        END IF;
    WHEN "1000"=> --JZ
        IF (W2=1) THEN
            IF (Z=1) THEN
                PCADD<=1;
            END IF;
        END IF;
    WHEN "1001"=> --JMP
        IF (W2=1) THE
            M<=1;
            S<="1111";
            ABUS<=1;
            LPC<=1;
        END IF;
    WHEN "1010"=> --OUT
        IF (W2=1) THEN
            M<=1;
            S<="1010";
            ABUS<=1;
        END IF;
    WHEN "1011"=> --ADD ITEMSELF
        IF (W2=1) THEN
            S<="1100";
            CIN<=1;
            ABUS<=1;
```

```
                DRW<=1;
                LDC<=1;
                LDZ<=1;
            END IF;
        WHEN "1100"=> --XOR
            IF (W2=1) THEN
                M<=1;
                S<="0110";
                ABUS<=1;
                DRW<=1;
                LDZ<=1;
            END IF;
        WHEN "1101"=> --OR
            IF (W2=1) THEN
                M<=1
                S<="1110";
                ABUS<=1;
                DRW<=1;
                LDZ<=1;
            END IF;
        WHEN "1111"=> --WOR
            IF (W2=1) THEN
                M<=1;
                S<="1001";
                ABUS<=1;
                DRW<=1;
                LDZ<=1;
            END IF;
        WHEN "1110"=> --STP
            IF (W2=1) THEN
                STOP<=1;
            END IF;
    WHEN OTHERS=>SBUS<=0;
    END CASE;
```

```
    WHEN OTHERS=>SBUS<=0;
    END CASE;
    END IF;
END PROCESS;
END cont;
```


SWC SWB SWA
STOP
000
001
010
011
100
101
IR7~IR4
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
STOP=0
STOP=1


勤恳朴诚
厚学改新