计算机组成原理课程设计报告

向阳曦 2017211279　张逸群　2017211571
(根据拼音排序)
班级 2017211301

目录

1 课题硬件环境描述

1 硬件: TEC-8 实验电路

2 软件: Windows10, Quartus II 9.0

2 题目分析

2.1 实验目标

1 完成硬连线控制器基础设计

2 在原指令的基础上扩指至少三条

3 修改 PC 指针功能

4 完成流水硬连线控制器基础设计

5 在原指令的基础上扩指至少三条

6 修改 PC 指针功能

2.2 设计思路

1 画出硬连线控制器运行流程图

2 将硬连线控制器运行流程图翻译成硬连线信号

3 将硬连线信号翻译成 VHDL 多分支程序

3 团队分工

向阳曦负责程序指令翻译, 张逸群负责控制台程序.

报告方面, 向阳曦主要完成 tex 矢量图绘画. 两人一起完成指令解释.

合并一起以后, 两人一起完成控制器程序的测试.

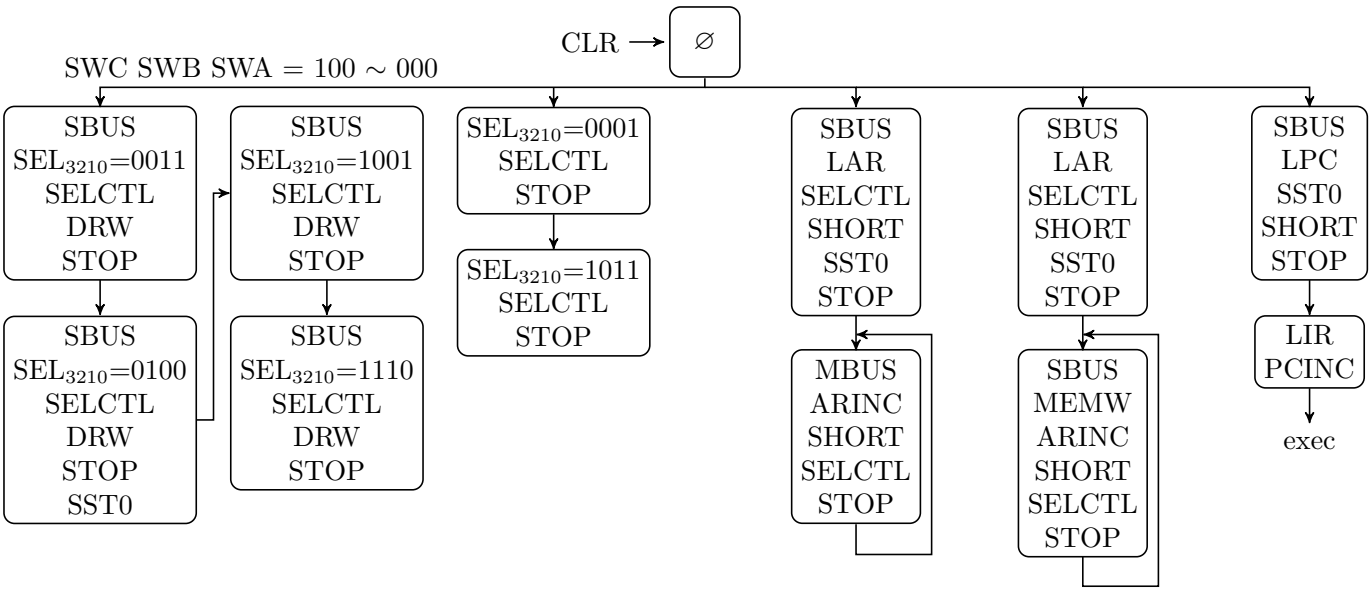
4 设计详解

4.1 硬连线控制器

硬连线控制器的关键是将 μP(IR) 信号作为控制信号的一部分, 根据控制信号的组合逻辑完成下地址转移.

为了设计的方便, 一般先将指令周期流程图画出, 然后根据指令周期流程图写出组合逻辑.VHDL 可以根据组合逻辑的特征, 合并多个相似的控制信号组合为分支条件语句.

4.1.1 硬连线控制器控制台指令流程图



其中 CLR 表示转移到 ∅. 任意没有出边的状态也都转移到 ∅, 任意时刻传入 CLR 信号也都转移到 ∅.

写寄存器 (SWC, SWB, SWA = 100)

写寄存器控制台指令 (100) 一共有四个周期, 第一个周期写 R_0 寄存器.

首先说明 R_0 数据从何进入系统. 在 TEC-8 的控制台上, 有八个并排的数据通路开关. 当 SBUS 信号有效时, 数据通路开关对应的 8 位数据会依次进入 DBUS 上对应的 8 个并排数据线.

其次说明如何将数据从 DBUS 打入 R_0 寄存器. 注意到我们将 SEL_3SEL_2 设置为 00, 表明 ALU 的 A 侧寄存器指针指向了 R_0 寄存器, 同时 SBUS 的 DRW 信号会使 DBUS 上的数据进入 A 侧寄存器指针指向的寄存器中.

剩下两个未说明的信号 SELCTL, STOP 的作用分别为设置当前指令有控制台操作, 指令运行以后暂停. 以后所有指令中这两个信号都是同一含义.

最后是一个无关紧要的细节.SEL_1SEL_0 被设置为 1, 表明 ALU 的 B 侧寄存器指向了 R_3 寄存器, 这样控制台上的 B 侧状态灯会显示 R_3 中的数据状态. 这方便我们检验 R_3 数据是否为我们想要的数据.

接下来 3 个周期分别为写 R_1 寄存器, 写 R_2 寄存器和写 R_3 寄存器. 这三个指令执行完毕以后, 没有下一个指令框, 表明我们会跳转回 $\varnothing$ 指令上, $\varnothing$ 什么也不执行

注意到我们在写 R_1 寄存器时, 有一个 SST0 信号, 这个信号表示 ST0 信号被置为 1, 当 ST0 信号被置为 1 时, 硬连线控制程序不会立即跳回空指令 (这里有两个周期, 为程序的默认指令周期长度. 如果不给 ST0 信号, 程序控制将无法区分同一个指令的一二和三四周期如何给出控制信号), 相当于为写寄存器指令延长两个周期.

读寄存器 (SWC, SWB, SWA = 011)

读存储器没有新的信号需要解释. 第一个指令周期我们将 SEL_{3210} 设为 0001, 这样 A 侧状态灯会显示 R_0 的数据, B 侧状态灯会显示 R_1 的数据. 第二个指令周期我们将 SEL_{3210} 设为 1011, 这样 A 侧状态灯会显示 R_2 的数据, B 侧状态灯会显示 R_3 的数据.

读存储器 (SWC, SWB, SWA = 010)

读存储器的第一个周期是准备操作, LAR 负责将 SBUS 信号送入的数据通路数据写入 AR, 指定了从何处开始读取数据. 第二至第 N 个周期, 是读取操作.MBUS 信号会操作主存打开朝向 DBUS 的总线, 使主存中 AR 寄存器指向的地址的数据得以输出.SHORT 信号操作控制器只发出 W_1 周期信号, 为单周期控制信号.ARINC 负责将 AR 在本周期结束时自增 1. 读取周期会一直重复到控制台接收到使用者发出的 CLR 信号.

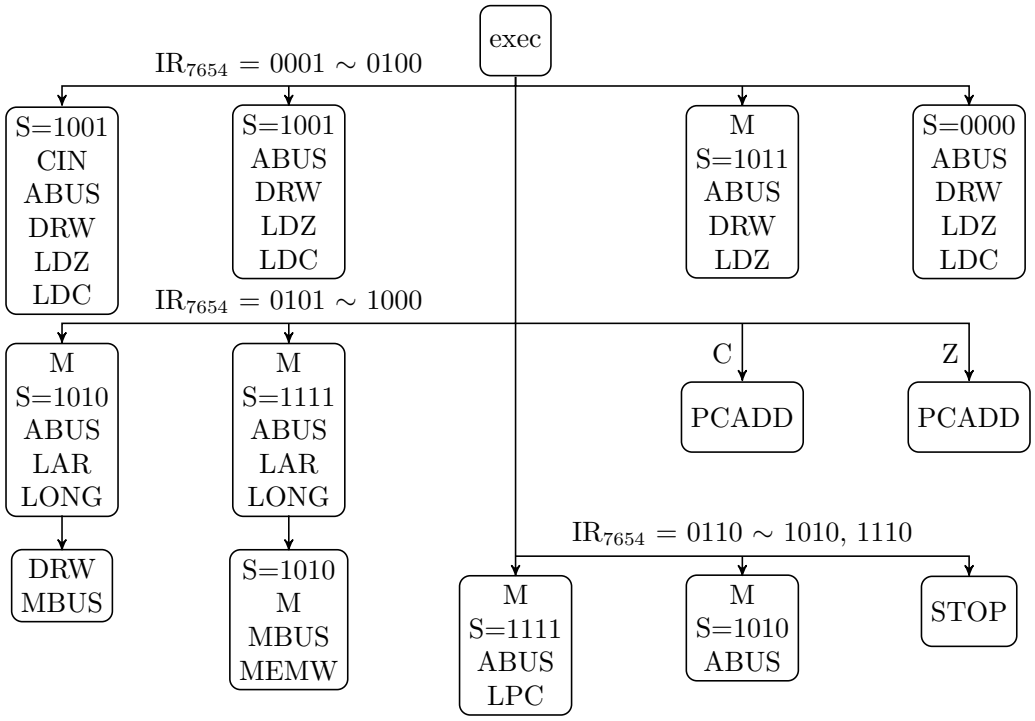
写存储器 (SWC, SWB, SWA = 001)

写存储器的第一个周期是准备操作, LAR 负责将 SBUS 信号送入的数据通路数据写入 AR, 指定了从何处开始读取数据. 第二至第 N 个周期, 是写入操作.SBUS 信号会操作数据通路开关上的数据进入 DBUS, 注意到 MEMW 有效, 因此 DBUS 的数据会同时打入 AR 指定的存储器地址.

执行程序 (SWC, SWB, SWA = 000)

执行程序的第一个周期是准备操作,SBUS 信号会操作数据通路开关上的数据进入 DBUS, 这时 LPC 信

号有效, 使得 DBUS 数据打入 PC 指针. 给出 SST0 信号以后 ST0 信号一直有效, 程序会一直在 exec 的循环中, 直到控制台接收到使用者发出的 CLR 信号.



上图显示了从左到右, 从上到下依次为 ADD, SUB, AND, INC, LD, ST, JC, JZ, JMP, OUT, STP 的所有指令, 为了画图方便, 没有列入的指令为 DEC(0000), SSP(1011), PUSHS(1100), MOV(1101), POPS(1111).

我们将 NOP 指令 (空指令) 改为了 DEC 指令, 这不影响程序的正确性, 因为 JMP 指令可以代 NOP 完成地址跳转.

需要注意的是, 如果某条指令包含 SHORT, 那么它是一个单拍指令, 如果某条指令包含 LONG, 那么它的下一拍还属于该指令, 计三拍.

关于指令的具体说明, 参考流水硬连线的各指令说明, 因为这两者除了流水的特征没有其他区别.

4.1.2 翻译成硬连线信号

这一步比较简单, 按照 SWC, SWB, SWA, IR_{7654} 将状态编号, 依此对信号进行组合.

SBUS $\leftarrow$

$$(\mathrm{W1} + \mathrm{W2}) \cdot \mathrm{SWC} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} +$$

$$W1 \cdot \overline{SWC} \cdot SWB \cdot \overline{SWA} \cdot \overline{ST0}+$$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot SWA+$$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{ST0}$$

MBUS ←

$$W1 \cdot \overline{SWC} \cdot SWB \cdot \overline{SWA} \cdot ST0+$$

$$W3 \cdot \overline{IR7} \cdot IR6 \cdot \overline{IR5} \cdot IR4 \cdot \overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA}$$

ABUS ←

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{IR7} \cdot \overline{IR6} \cdot \overline{IR5} \cdot IR4 \cdot W2+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{IR7} \cdot \overline{IR6} \cdot \overline{IR5} \cdot \overline{IR4} \cdot W2+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{IR7} \cdot \overline{IR6} \cdot IR5 \cdot \overline{IR4} \cdot W2+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{IR7} \cdot \overline{IR6} \cdot IR5 \cdot IR4 \cdot W2+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{IR7} \cdot IR6 \cdot \overline{IR5} \cdot \overline{IR4} \cdot W2+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{IR7} \cdot IR6 \cdot \overline{IR5} \cdot IR4 \cdot W2+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{IR7} \cdot IR6 \cdot IR5 \cdot \overline{IR4} \cdot (W2 + W3)+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot IR7 \cdot \overline{IR6} \cdot \overline{IR5} \cdot IR4 \cdot W2+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot IR7 \cdot \overline{IR6} \cdot IR5 \cdot \overline{IR4} \cdot W2$$

LAR ←

$$W1 \cdot \overline{SWC} \cdot SWB \cdot \overline{SWA} \cdot \overline{ST0}+$$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot SWA \cdot \overline{ST0}+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} + \cdot\overline{IR7} \cdot IR6 \cdot \overline{IR5} \cdot IR4 \cdot W2+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} + \cdot\overline{IR7} \cdot IR6 \cdot IR5 \cdot \overline{IR4} \cdot W2$$

SST0 $\leftarrow$

$$W1 \cdot \overline{SWC} \cdot SWB \cdot \overline{SWA} \cdot \overline{ST0}+$$

$$W2 \cdot SWC \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{ST0}+$$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot SWA \cdot \overline{ST0}+$$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{ST0}$$

SHORT $\leftarrow$

$$W1 \cdot \overline{SWC} \cdot SWB \cdot \overline{SWA}+$$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot SWA+$$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{ST0}$$

ARINC $\leftarrow$

$$W1 \cdot \overline{SWC} \cdot SWB \cdot \overline{SWA} \cdot ST0+$$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot SWA \cdot ST0$$

MEMW $\leftarrow$

$$W1 \cdot \overline{SWC} \cdot \overline{SWB} \cdot SWA \cdot ST0+$$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot \overline{IR7} \cdot IR6 \cdot IR5 \cdot \overline{IR4} \cdot W3$$

LIR $\leftarrow$

$$\overline{SWC} \cdot \overline{SWB} \cdot \overline{SWA} \cdot ST0 \cdot W1$$

CIN ←

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \overline{\mathrm{IR7}} \cdot \overline{\mathrm{IR6}} \cdot \overline{\mathrm{IR5}} \cdot \overline{\mathrm{IR4}} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \overline{\mathrm{IR7}} \cdot \overline{\mathrm{IR6}} \cdot \overline{\mathrm{IR5}} \cdot \overline{\mathrm{IR4}} \cdot \mathrm{W2}$$

M ←

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \overline{\mathrm{IR7}} \cdot \overline{\mathrm{IR6}} \cdot \mathrm{IR5} \cdot \mathrm{IR4} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \overline{\mathrm{IR7}} \cdot \mathrm{IR6} \cdot \overline{\mathrm{IR5}} \cdot \mathrm{IR4} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \overline{\mathrm{IR7}} \cdot \mathrm{IR6} \cdot \mathrm{IR5} \cdot \overline{\mathrm{IR4}} \cdot (\mathrm{W2} + \mathrm{W3})+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{IR7} \cdot \overline{\mathrm{IR6}} \cdot \overline{\mathrm{IR5}} \cdot \mathrm{IR4} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{IR7} \cdot \overline{\mathrm{IR6}} \cdot \mathrm{IR5} \cdot \overline{\mathrm{IR4}} \cdot \mathrm{W2}$$

SEL0←

$$\mathrm{SWC} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{W1}$$

SEL1←

$$\mathrm{SWC} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{W2} \cdot \mathrm{ST0}$$

$$\mathrm{SWC} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{W1} \cdot \overline{\mathrm{ST0}}$$

SEL2←

$$\mathrm{SWC} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{W2}$$

SEL3←

$$\mathrm{SWC} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{W1} \cdot \mathrm{ST0}+$$

$$\mathrm{SWC} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{W2} \cdot \overline{\mathrm{ST0}}$$

STOP←

$$\mathrm{SWA} + \mathrm{SWB} + \mathrm{SWC}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{IR7} \cdot \mathrm{IR6} \cdot \mathrm{IR5} \cdot \overline{\mathrm{IR4}} \cdot \mathrm{ST0} \cdot \mathrm{W2}$$

SELCTL←

$$\mathrm{SWA} + \mathrm{SWB} + \mathrm{SWC}$$

DRW←

$$\mathrm{SWC} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \overline{\mathrm{IR7}} \cdot \overline{\mathrm{IR6}} \cdot \overline{\mathrm{IR5}} \cdot \overline{\mathrm{IR4}} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{ST0} \cdot \overline{\mathrm{IR7}} \cdot \overline{\mathrm{IR6}} \cdot \overline{\mathrm{IR5}} \cdot \mathrm{IR4} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{ST0} \cdot \overline{\mathrm{IR7}} \cdot \overline{\mathrm{IR6}} \cdot \mathrm{IR5} \cdot \overline{\mathrm{IR4}} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{ST0} \cdot \overline{\mathrm{IR7}} \cdot \overline{\mathrm{IR6}} \cdot \mathrm{IR5} \cdot \mathrm{IR4} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{ST0} \cdot \overline{\mathrm{IR7}} \cdot \mathrm{IR6} \cdot \overline{\mathrm{IR5}} \cdot \overline{\mathrm{IR4}} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{ST0} \cdot \overline{\mathrm{IR7}} \cdot \mathrm{IR6} \cdot \overline{\mathrm{IR5}} \cdot \mathrm{IR4} \cdot \mathrm{W3}$$

PCADD←

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{ST0} \cdot \overline{\mathrm{IR7}} \cdot \mathrm{IR6} \cdot \mathrm{IR5} \cdot \mathrm{IR4} \cdot \mathrm{C} \cdot \mathrm{W2}+$$

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{ST0} \cdot \mathrm{IR7} \cdot \overline{\mathrm{IR6}} \cdot \overline{\mathrm{IR5}} \cdot \overline{\mathrm{IR4}} \cdot \mathrm{Z} \cdot \mathrm{W2}$$

PCINC←

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \mathrm{ST0} \cdot \mathrm{W1}$$

LPC←

$$\overline{\mathrm{SWC}} \cdot \overline{\mathrm{SWB}} \cdot \overline{\mathrm{SWA}} \cdot \overline{\mathrm{ST0}} \cdot \mathrm{W2}$$

LONG←

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{W2} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \text{IR4}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{W2} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \overline{\text{IR4}}$

LDC←

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W2}$

LDZ←

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W2}$

4.1.3 翻译成多分支程序

观察硬连线信号, 事实上如果写成 VHDL 语句, 我们可以将 SWCBA 信号合并. 然后再在 $\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}}$ 分支上将 IR_{7654} 合并. 用两个大 switch 语句完成逻辑分类.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
entity CPU is
        port (
                CLR,C,Z,T3,W1,W2,W3: in std_logic;
                IRH:in std_logic_vector(3 downto 0);
```

```
                SWCBA:in std_logic_vector(2 downto 0);
                SELCTL,ABUS,M,SEL1,SEL0,SEL2,SEL3,DRW,SBUS,LIR,MBUS,MEMW,LAR,ARINC,LPC,
                PCINC,PCADD,CIN,LONG,SHORT,STOP,LDC,LDZ: out std_logic;
                S:out std_logic_vector(3 downto 0);
                CP1,CP2,CP3:out std_logic;
                QD:in std_logic
        );
end CPU;

architecture arc of CPU is
signal ST0,ST0_1,ST0_2,STOP_1,STOP_2: std_logic;
begin
        CP1 <= '1';
        CP2 <= '1';
        CP3 <= QD;

        with SWCBA select
                STOP <= '0'                when "000",
                STOP_1 or STOP_2           when others;
        ST0 <= ST0_1;

        process (CLR, T3)
        begin
                -- 任何时候按下CLR, 都会返回
                if (CLR = '0') then
                        ST0_1   <= '0';
                        STOP_1  <= '1';
                -- 如果到节拍电位下降沿T3,ST0_1 |= ST0_2
                elsif (T3'event and T3 = '0') then
                        if (ST0_2 = '1') then
                                ST0_1 <= '1';
                        end if;
                end if;
        end process;

        process (SWCBA, IRH, W1, W2, W3, ST0, C, Z)
        begin
                -- 初始化和状态参数
                SHORT <= '0';
                LONG <= '0';
                -- 设置STOP
                STOP_2 <= '1';
                -- 设置标志ST0
                ST0_2 <= '0';
                -- ALU
                ABUS <= '0';
```

```
                M <= '0';
                CIN <= '0';
                S <= "0000";
                ARINC <= '0';
                -- 保存标志Z
                LDZ <= '0';
                -- 保存标志C
                LDC <= '0';
                SBUS <= '0';
                MBUS <= '0';
                -- 控制台操作标志
                SELCTL <= '0';
                -- RD1~RD0
                SEL3 <= '0';
                SEL2 <= '0';
                -- RS1~RS0
                SEL1 <= '0';
                SEL0 <= '0';
                -- 送指令寄存器标志
                LIR <= '0';
                -- 送地址寄存器标志
                LAR <= '0';
                -- 送程序计数器标志
                LPC <= '0';
                -- (~R)/W
                MEMW <= '0';
                DRW <= '0';
                -- 程序计数器自增标志
                PCINC <= '0';
                -- 程序计数器增量标志
                PCADD <= '0';
                case SWCBA is
                        when "000" =>  --执行程序
                                case ST0 is
                                        when '0' =>
                                                -- load pc
                                                LPC <= W1;
                                                SBUS <= W1;
                                                ST0_2 <= W1;
                                                SHORT <= W1;
                                                STOP_2 <= '0';
                                        when '1' =>
                                                if(W1='1')then
                                                        LIR<=W1;
                                                        PCINC<=W1;
                                                else
```

```
                                    case IRH is
                                        when "0001" =>  --ADD
                                            -- ABUS = W2
                                            ABUS <= W2;
                                            CIN <= W2;
                                            -- 选择加法
                                            -- 选择算术运算, 已经被初始化为M0
                                            S <= "1001";
                                            -- 加法操作
                                            DRW <= W2;
                                            LDZ <= W2;
                                            LDC <= W2;
                                        when "0010" =>  -- SUB
                                            -- 选择算术运算, 选择减法
                                            -- 已经被初始化为M0
                                            S <= "0110";
                                            -- 减法操作
                                            ABUS <= W2;
                                            DRW <= W2;
                                            LDZ <= W2;
                                            LDC <= W2;
                                        when "0011" =>  -- AND
                                            -- 选择逻辑运算, 与运算
                                            M <= W2;
                                            S <= "1011";
                                            ABUS <= W2;
                                            DRW <= W2;
                                            LDZ <= W2;
                                        when "0100" => -- INC
                                            -- 选择算术运算, 与运算
                                            -- 已经被初始化为M0
                                            S <= "0000";
                                            ABUS <= W2;
                                            DRW <= W2;
                                            LDZ <= W2;
                                            LDC <= W2;
                                        when "0101" =>  -- LD
                                            -- 选择算术运算传送,（保留原值）B
                                            M <= W2;
                                            S <= "1010";
                                            ABUS <= W2;
                                            LONG<=W2;
                                            LAR <= W2;
                                            MBUS <= W3;
                                            DRW <= W3;
                                        when "0110" =>  -- ST
```

```
                LONG<=W2;
                -- 设定...
                M <= W2 or W3;
                if(W2='1')then
                        S<="1111";
                else
                        S<="1010";
                end if;
                ABUS <= W2 or W3;
                LAR <= W2;
                MEMW <= W3;
        when "0111" =>  -- JC
                PCADD <= C and W2;
        when "1000" =>  -- JZ
                PCADD <= Z and W2;
        when "1001" =>  -- JMP
                -- 设定算术运算
                M <= W2;
                S <= "1111";
                ABUS <= W2;
                LPC <= W2;
        when "1010" =>  -- OUT
                -- 设定算术运算
                M <= W2;
                S <= "1010";
                ABUS <= W2;
        when "1011" =>  -- SSP
                SEL3<='1';
                LONG<=W2;
                M <= W2 or W3;
                if (W2 = '1') then
                        S <= "1000";
                elsif (W3 = '1') then
                        S <= "1010";
                end if;
                ABUS <= W2 or W3;
                LAR <= W2;
                MEMW <= W3;
        when "1100" =>  -- PUSH
                --无法自减PUSH
                M <= W2 or W3;
                if (W2 = '1') then
                        S <= "1111";
                elsif (W3 = '1') then
                        S <= "1010";
                end if;
```

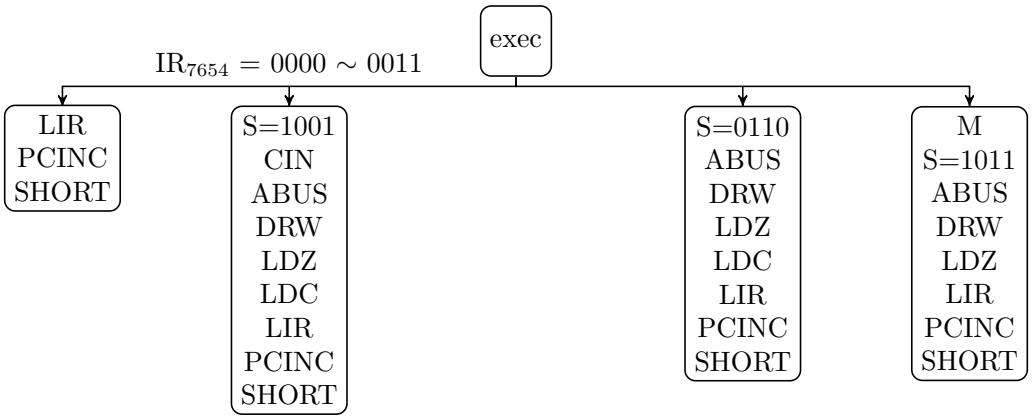
```
                                        ABUS <=W2 or W3;
                                        LAR <= W2;
                                        MEMW <= W3;
                                        LONG <= W2;
                                when "1101" =>  -- MOV B->A
                                        -- 选择逻辑运算, MOV 运算
                                        M <= W2;
                                        S <= "1010";
                                        ABUS <= W2;
                                        DRW <= W2;
                                        LDZ <= W2;
                                when "1110" =>  -- STP
                                        STOP_2 <= W2;
                                when "1111" =>  -- LSP
                                        LONG<=W2;
                                        M <= W2;
                                        S <= "1000";
                                        ABUS <= W2;
                                        LAR <= W2;
                                        MBUS <= W3;
                                        DRW <= W3;
                                when others =>  -- 公操作
                        end case;
                                end if;
                        when others =>
                                        -- 不可能到这吧?
                end case;
        when "001" =>
        --      SEL0<=ST0;
                -- SBUS = (ST0=0 or ST0=1) and W1
                SBUS <= W1;
                -- STOP = (ST0=0 or ST0=1) and W1
                STOP_2 <= W1;
                -- SHORT = (ST0=0 or ST0=1) and W1
                SHORT <= W1;
                -- SELCTL = (ST0=0 or ST0=1) and W1
                SELCTL <= W1;
                -- LAR = (ST0=0) and W1
                LAR <= W1 and (not ST0);
                -- LAR = (ST0=1) and W1
                ARINC <= W1 and ST0;
                -- MEMW = (ST0=1) and W1
                MEMW <= W1 and ST0;
                ST0_2 <= W1;
        when "010" =>
                -- SHORT = (ST0=0 or ST0=1) and W1
```

```
                        SHORT<=W1;
                        -- SELCTL = (ST0=0 or ST0=1) and W1
                        SELCTL <= W1;
                        -- STOP = (ST0=0 or ST0=1) and W1
                        STOP_2<=W1;
                        -- SBUS = (ST0=0) and W1
                        SBUS<=W1 and (not ST0);
                        -- LAR = (ST0=0) and W1
                        LAR<=W1 and (not ST0);
                        -- MBUS = (ST0=1) and W1
                        MBUS<=W1 and ST0;
                        -- ARINC = (ST0=1) and W1
                        ARINC<=W1 and ST0;
                        ST0_2<=W1;
                when "011" =>
                        -- SELCTL = W1 or W2
                        SELCTL <= '1';
                        -- STOP = W1 or W2
                        STOP_2 <= W1 or W2;
                        -- SEL0 = W1 or W2
                        SEL0<=W1 or W2;
                        -- SEL1 = W2
                        SEL1<=W2;
                        -- SEL2 = 0
                        -- SEL3 = W2
                        SEL3<=W2;
                when "100" =>
                        -- SELCTL = (ST0=0 or ST0=1) and (W1 or W2)
                        SELCTL <= '1';
                        -- SBUS = (ST0=0 or ST0=1) and (W1 or W2)
                        SBUS <= W1 or W2;
                        -- STOP = (ST0=0 or ST0=1) and (W1 or W2)
                        STOP_2 <= W1 or W2;
                        -- DRW = (ST0=0 or ST0=1) and (W1 or W2)
                        DRW <= W1 or W2;
                        -- SEL0 = (ST0=0 or ST0=1) and W1
                        SEL0 <= W1;
                        -- SEL1 = ((ST0=0) and W1) or ((ST0=1) and W2)
                        SEL1 <= ((not ST0) and W1) or (ST0 and W2);
                        -- SEL2 = (ST0=0 or ST0=1) and W2
                        SEL2 <= W2;
                        -- SEL3 = (ST0=1) and (W1 or W2)
                        SEL3 <= ST0 and (W1 or W2);
                        ST0_2 <= W2;
                when others=>
        end case;
```

```
        end process;
end arc;
```

4.2 流水硬连线控制器

4.2.1 指令周期说明



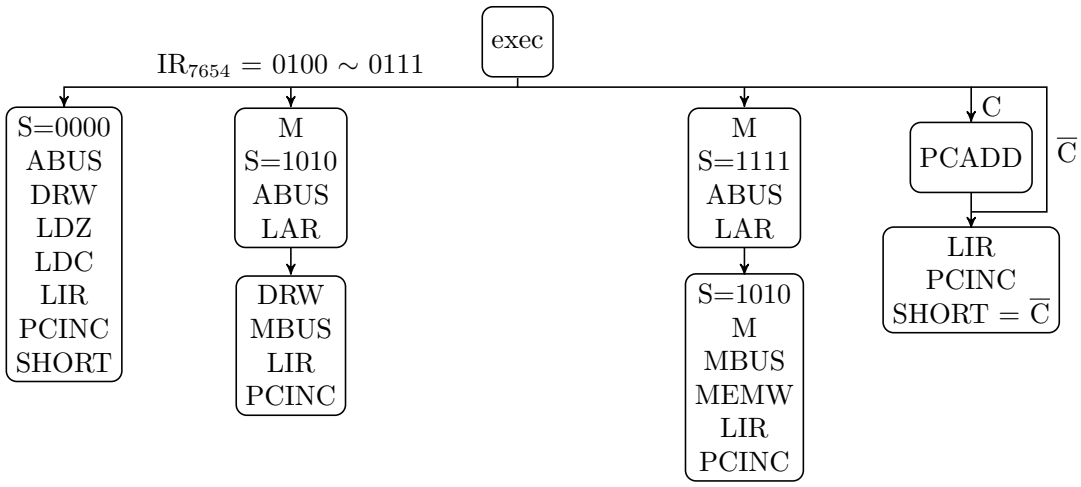
上图给出了第一组四个指令的周期.

第一个是 NOP 空指令, 只加载下地址指令, 并向控制器发送 SHORT 信号表明该指令只有一个周期.

第二个是 ADD 指令.M=0,S=1001 表明 ALU 完成加法运算, CIN=1 表明 ALU 不进 1, 这时候 ALU 将 A 侧寄存器和 B 侧寄存器作为输入,ABUS 将 ALU 的运算结果 A plus B 打入 DBUS.DRW 将 DBUS 数据打入 A 侧寄存器, 因此这个指令完成了 $A \leftarrow A$ plus B.

第三个是 SUB 指令.M=0,S=0110 表明 ALU 完成减法运算, CIN=0 表明 ALU 进 1, 这时候 ALU 将 A 侧寄存器和 B 侧寄存器作为输入,ABUS 将 ALU 的运算结果 A minus B minus 1 plus 1 打入 DBUS.DRW 将 DBUS 数据打入 A 侧寄存器, 因此这个指令完成了 $A \leftarrow A$ minus B.

第四个是 AND 指令.M=1,S=1011 表明 ALU 完成且运算, 这时候 ALU 将 A 侧寄存器和 B 侧寄存器作为输入,ABUS 将 ALU 的运算结果 AB 打入 DBUS.DRW 将 DBUS 数据打入 A 侧寄存器, 因此这个指令完成了 $A \leftarrow AB$.



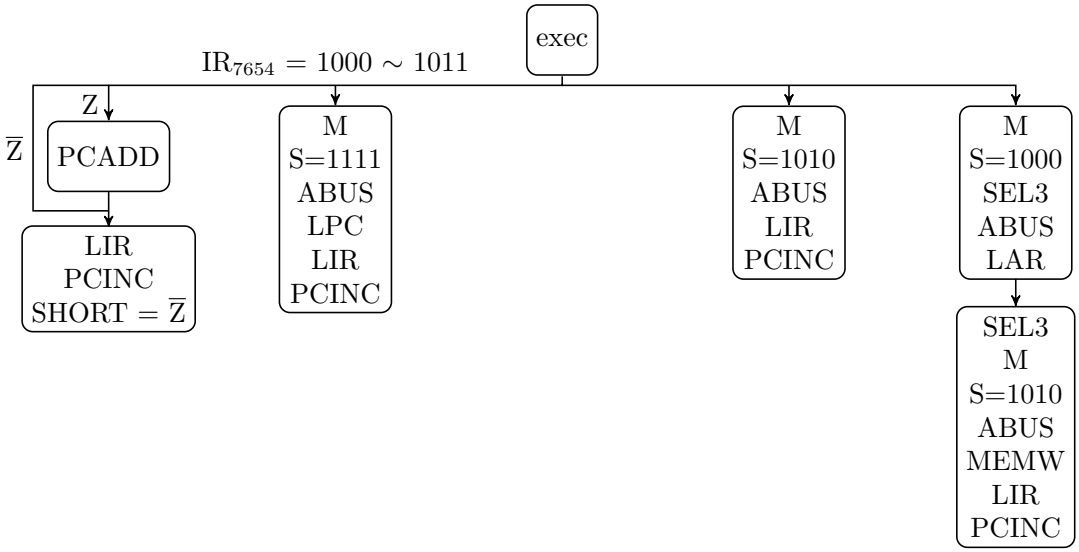
上图给出了第二组四个指令的周期.

第五个指令是 INC 指令.M=0, S=0000 表明 ALU 完成自输出运算,CIN=0 表明 ALU 进 1, 这时候 ALU 将 A 侧寄存器作为输入,ABUS 将 ALU 的运算结果 A 打入 DBUS, 因此 DRW 将 DBUS 数据打入 A 侧寄存器, 因此这个指令完成了 $A \leftarrow A$ plus 1.

第六个指令是 LD 指令. 在指令的第一个周期,M=1,S=1010 表明 ALU 完成自输出.

第七个指令是 ST 指令. 在指令的第一个周期,M=1,S=1111 表示把地址送到地址寄存器, 第二个周期把数据送上数据总线并且写入存储器.

第八个指令是 JC 指令. 如果当前的 C 不为 1, 则直接取下一条的指令结束, 如果当前的 C 为 1, 会调用 PCADD 把当前的 PC 加上偏移的 offset, 最后还是流水机的取指令共操作.



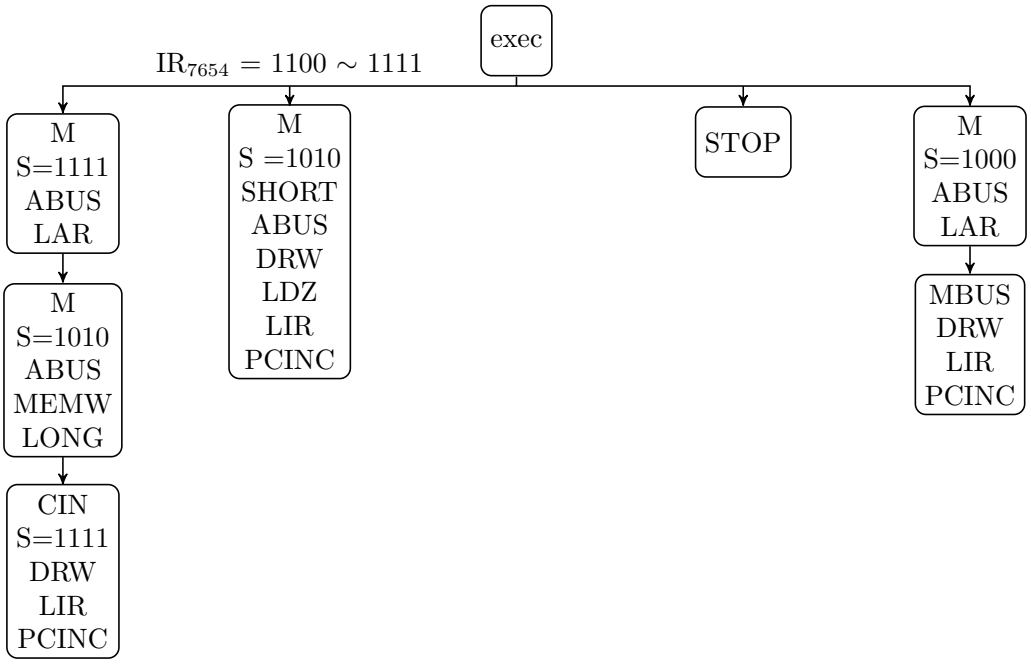
上图给出了第三组四个指令的周期.

第九个指令是 JZ 指令. 如果当前的 Z 不为 1, 则直接取下一条的指令结束, 如果当前的 Z 为 1, 会调用 PCADD 把当前的 PC 加上偏移的 offset, 最后还是流水机的取指令共操作.

第十个指令是 JMP 指令. 第一个周期调用 LPC 并且打开 ABUS 开关, 把地址送上地址总线, 并且写入 PC.

第十一个指令是 OUT 指令. 打开 ABUS, 作用是把数据送上地址总线, 方便观察.

第十二个指令是 SSP 指令. 作用是把保存在寄存器中的栈指针 SP 送回 SP 在存储器中的位置. 这个指令会占用两个周期. 由于栈指针 SP 是隐式给出的, 因此需要先算出 SP 的地址 [FF]. 第一个周期计算 $A|(!A)$ 得到 FF, 同时打开 ABUS 把 FF 送入地址寄存器. 第二个周期把数据写入存储器的 FF 地址.



上图给出了最后一组四个指令的周期.

第十三个指令是 PUSH 指令. 作用是把数据写入栈顶, 并且把栈指针减一. 第一周期是把已经读出到寄存器的 SP 送上地址寄存器, 第二周期把数据写入 [SP], 第三周期把寄存器中的 SP 减一

第十四个指令是 MOV 指令. 只有一个周期, 通过 ALU 实现 A 和 B 计算后输出 A, 然后把输出写入 B.

第十五个指令是 STP 指令. 作用是令 STOP 信号为 1, 终止程序.

第十六个指令是 LSP 指令. 作用是把栈指针 SP 从存储器中读出. 第一个周期和 STP 一样是计算出栈指

针的所在位置 FF, 第二周期从存储器读出 SP 然后存到指定寄存器.

4.2.2 翻译成硬连线信号

这一步与普通硬连线的过程一样。

SBUS ←

$$(\mathrm{W1}+\mathrm{W2})\cdot\mathrm{SWC}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}+$$

$$\mathrm{W1}\cdot\overline{\mathrm{SWC}}\cdot\mathrm{SWB}\cdot\overline{\mathrm{SWA}}\cdot\overline{\mathrm{ST0}}+$$

$$\mathrm{W1}\cdot\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\mathrm{SWA}+$$

$$\mathrm{W1}\cdot\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\overline{\mathrm{ST0}}$$

MBUS ←

$$\mathrm{W1}\cdot\overline{\mathrm{SWC}}\cdot\mathrm{SWB}\cdot\overline{\mathrm{SWA}}\cdot\mathrm{ST0}+$$

$$\mathrm{W2}\cdot\overline{\mathrm{IR7}}\cdot\mathrm{IR6}\cdot\overline{\mathrm{IR5}}\cdot\mathrm{IR4}\cdot\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}$$

ABUS ←

$$\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\overline{\mathrm{IR7}}\cdot\overline{\mathrm{IR6}}\cdot\overline{\mathrm{IR5}}\cdot\mathrm{IR4}\cdot\mathrm{W1}+$$

$$\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\overline{\mathrm{IR7}}\cdot\overline{\mathrm{IR6}}\cdot\mathrm{IR5}\cdot\overline{\mathrm{IR4}}\cdot\mathrm{W1}+$$

$$\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\overline{\mathrm{IR7}}\cdot\overline{\mathrm{IR6}}\cdot\mathrm{IR5}\cdot\mathrm{IR4}\cdot\mathrm{W1}+$$

$$\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\overline{\mathrm{IR7}}\cdot\mathrm{IR6}\cdot\overline{\mathrm{IR5}}\cdot\overline{\mathrm{IR4}}\cdot\mathrm{W1}+$$

$$\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\overline{\mathrm{IR7}}\cdot\mathrm{IR6}\cdot\overline{\mathrm{IR5}}\cdot\mathrm{IR4}\cdot\mathrm{W1}+$$

$$\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\overline{\mathrm{IR7}}\cdot\mathrm{IR6}\cdot\mathrm{IR5}\cdot\overline{\mathrm{IR4}}\cdot(\mathrm{W1}+\mathrm{W2})+$$

$$\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\mathrm{IR7}\cdot\overline{\mathrm{IR6}}\cdot\overline{\mathrm{IR5}}\cdot\mathrm{IR4}\cdot\mathrm{W1}+$$

$$\overline{\mathrm{SWC}}\cdot\overline{\mathrm{SWB}}\cdot\overline{\mathrm{SWA}}\cdot\mathrm{IR7}\cdot\overline{\mathrm{IR6}}\cdot\mathrm{IR5}\cdot\overline{\mathrm{IR4}}\cdot\mathrm{W1}$$

$$\text{LAR} \leftarrow$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \text{SWB} \cdot \overline{\text{SWA}} \cdot \overline{\text{ST0}}+$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \text{SWA} \cdot \overline{\text{ST0}}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W1}$$

$$\text{SST0} \leftarrow$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \text{SWB} \cdot \overline{\text{SWA}} \cdot \overline{\text{ST0}}+$$

$$\text{W2} \cdot \text{SWC} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{ST0}}+$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \text{SWA} \cdot \overline{\text{ST0}}+$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{ST0}}$$

$$\text{SHORT} \leftarrow$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \text{SWB} \cdot \overline{\text{SWA}}$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \text{SWA}$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{ST0}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W1} \cdot \overline{C}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{IR7} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1} \cdot \overline{Z}$$

ARINC $\leftarrow$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \text{SWB} \cdot \overline{\text{SWA}} \cdot \text{ST0}+$$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \text{SWA} \cdot \text{ST0}$$

MEMW $\leftarrow$

$$\text{W1} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \text{SWA} \cdot \text{ST0}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W2}$$

LIR $\leftarrow$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W1} \cdot \overline{C}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{IR7} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1} \cdot \overline{Z}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W2} \cdot C+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{IR7} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W2} \cdot Z+$$

$$\text{W2} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W2}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{ST0}} \cdot \text{W1}$$

CIN ←

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1}$$

M ←

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W2}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W2}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot (\text{W2} + \text{W3})+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{IR7} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W2}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{IR7} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W2}$$

SEL0←

$$\text{SWC} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{W1}$$

SEL1←

$$\text{SWC} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{W2} \cdot \text{ST0}+$$

$$\text{SWC} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{W1} \cdot \overline{\text{ST0}}$$

SEL2←

$$\text{SWC} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{W2}$$

SEL3←

$$\text{SWC} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{W1} \cdot \text{ST0}+$$

$$\text{SWC} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{W2} \cdot \overline{\text{ST0}}$$

STOP ←

$$\text{SWA} + \text{SWB} + \text{SWC}+$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{IR7} \cdot \text{IR6} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{ST0} \cdot \text{W2}$$

SELCTL $\leftarrow$

$$\overline{\text{SWA}} + \overline{\text{SWB}} + \overline{\text{SWC}}$$

DRW $\leftarrow$

$$\text{SWC} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W2} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W2} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W2} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W2} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W3}$$

PCADD $\leftarrow$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{C} \cdot \text{W2} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \text{IR7} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{Z} \cdot \text{W2}$$

PCINC $\leftarrow$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W1} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W1} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W1} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1} +$$

$$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W1} \cdot \overline{C} +$$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{IR7} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1} \cdot \overline{Z}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W2} \cdot C+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{IR7} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W2} \cdot Z+$

$\text{W2} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W2}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{ST0}} \cdot \text{W1}$

LPC $\leftarrow$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \overline{\text{ST0}} \cdot \text{W1}$

LDC $\leftarrow$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W1}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W1}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W1}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1}$

LDZ $\leftarrow$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \overline{\text{IR5}} \cdot \text{IR4} \cdot \text{W1}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \overline{\text{IR4}} \cdot \text{W1}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \overline{\text{IR6}} \cdot \text{IR5} \cdot \text{IR4} \cdot \text{W1}+$

$\overline{\text{SWC}} \cdot \overline{\text{SWB}} \cdot \overline{\text{SWA}} \cdot \text{ST0} \cdot \overline{\text{IR7}} \cdot \text{IR6} \cdot \overline{\text{IR5}} \cdot \overline{\text{IR4}} \cdot \text{W1}$

4.2.3 翻译成多分支程序

这一步与普通硬连线的过程一样。

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
entity CPU is
        port (
                CLR,C,Z,T3,W1,W2,W3: in std_logic;
                IRH:in std_logic_vector(3 downto 0);
                SWCBA:in std_logic_vector(2 downto 0);
                SELCTL,ABUS,M,SEL1,SEL0,SEL2,SEL3,DRW,SBUS,LIR,MBUS,MEMW,LAR,ARINC,LPC,
                PCINC,PCADD,CIN,LONG,SHORT,STOP,LDC,LDZ: out std_logic;
                S:out std_logic_vector(3 downto 0);
                CP1,CP2,CP3:out std_logic;
                QD:in std_logic
        );
end CPU;

architecture arc of CPU is
signal ST0,ST0_1,ST0_2,STOP_1,STOP_2: std_logic;
begin

        CP1 <= '1';
        CP2 <= '1';
        CP3 <= QD;

        with SWCBA select
                STOP <= '0'             when "000",
                STOP_1 or STOP_2        when others;
        ST0 <= ST0_1;

        process (CLR, T3)
        begin
                -- 任何时候按下CLR, 都会返回
                if (CLR = '0') then
                        ST0_1   <= '0';
                        STOP_1  <= '1';
                -- 如果到节拍电位下降沿T3,ST0_1 |= ST0_2
                elsif (T3'event and T3 = '0') then
                        if (ST0_2 = '1') then
                                ST0_1 <= '1';
                        end if;
                end if;
```

```
    end process;

    process (SWCBA, IRH, W1, W2, W3, ST0, C, Z)
    begin
            -- 初始化和状态参数
            SHORT <= '0';
            LONG <= '0';
            -- 设置STOP
            STOP_2 <= '1';
            -- 设置标志ST0
            ST0_2 <= '0';
            -- ALU
            ABUS <= '0';
            M <= '0';
            CIN <= '0';
            S <= "0000";
            ARINC <= '0';
            -- 保存标志Z
            LDZ <= '0';
            -- 保存标志C
            LDC <= '0';
            SBUS <= '0';
            MBUS <= '0';
            -- 控制台操作标志
            SELCTL <= '0';
            -- RD1~RD0
            SEL3 <= '0';
            SEL2 <= '0';
            -- RS1~RS0
            SEL1 <= '0';
            SEL0 <= '0';
            -- 送指令寄存器标志
            LIR <= '0';
            -- 送地址寄存器标志
            LAR <= '0';
            -- 送程序计数器标志
            LPC <= '0';
            -- (~R)/W
            MEMW <= '0';
            DRW <= '0';
            -- 程序计数器自增标志
            PCINC <= '0';
            -- 程序计数器增量标志
            PCADD <= '0';
            case SWCBA is
                    when "000" =>  --执行程序
```

```
                case ST0 is
                        when '0' =>
                                -- load pc
                                LPC <= W1;
                                SBUS <= W1;
                                ST0_2 <= W1;
                                SHORT <= W1;
                                STOP_2 <= '0';
                        when '1' =>
                case IRH is
                        when "0000" =>  -- NOP
                                -- 设定PC
                                LIR <= W1;
                                PCINC <= W1;
                                -- 短周期
                                SHORT <= W1;
                        when "0001" =>  --ADD ()
                                -- 设定PC
                                LIR <= W1;
                                PCINC <= W1;
                                -- 短周期
                                SHORT <= W1;
                                -- ABUS = W1
                                ABUS <= W1;
                                CIN <= W1;
                                -- 选择加法
                                -- 选择算术运算, 传送 B
                                -- 已经被初始化为M0
                                S <= "1001";
                                -- 加法操作
                                DRW <= W1;
                                LDZ <= W1;
                                LDC <= W1;
                        when "0010" =>  -- SUB ()
                                -- 设定PC
                                LIR <= W1;
                                PCINC <= W1;
                                -- 短周期
                                SHORT <= W1;
                                -- 选择算术运算, 选择减法
                                -- 已经被初始化为M0
                                S <= "0110";
                                -- 减法操作
                                ABUS <= W1;
                                DRW <= W1;
                                LDZ <= W1;
```

```
                LDC <= W1;
            when "0011" =>  -- AND ()
                -- 设定PC
                LIR <= W1;
                PCINC <= W1;
                -- 短周期
                SHORT <= W1;
                -- 选择逻辑运算, 与运算
                M <= W1;
                S <= "1011";
                ABUS <= W1;
                DRW <= W1;
                LDZ <= W1;
            when "0100" => -- INC ()
                -- 设定PC
                LIR <= W1;
                PCINC <= W1;
                -- 短周期
                SHORT <= W1;
                -- 选择算术运算, 与运算
                -- 已经被初始化为M0
                S <= "0000";
                ABUS <= W1;
                DRW <= W1;
                LDZ <= W1;
                LDC <= W1;
            when "0101" =>  -- LD
                -- 选择算术运算, 传送 B
                M <= W1;
                S <= "1010";
                ABUS <= W1;
                LAR <= W1;
                -- 设定PC
                LIR <= W2;
                PCINC <= W2;
                MBUS <= W2;
                DRW <= W2;
            when "0110" =>  -- ST
                -- 设定...
                M <= W1 or W2;
                if(W1='1')then
                        S<="1111";
                else
                        S<="1010";
                end if;
                ABUS <= W1 or W2;
```

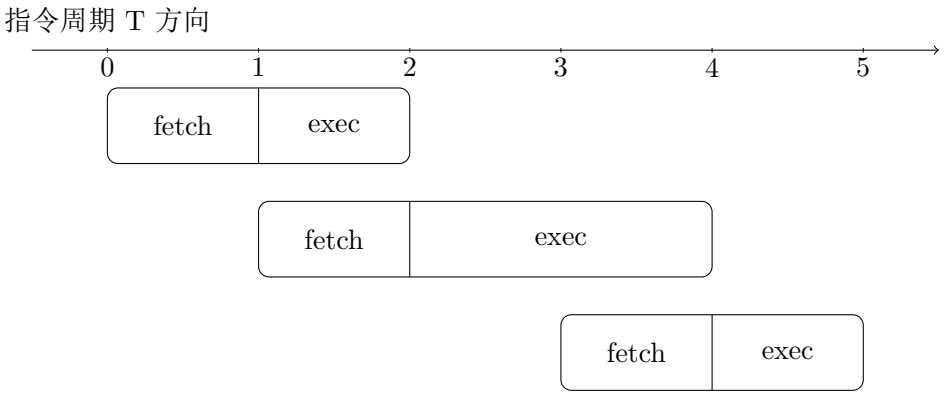
```
                LAR <= W1;
                MEMW <= W2;
                -- 设定PC
                LIR <= W2;
                PCINC <= W2;
            when "0111" =>  -- JC
                -- 设定PC
                LIR <= (W1 and (not C)) or (W2 and C);
                PCINC <= (W1 and (not C)) or (W2 and C);
                PCADD <= C and W1;
                SHORT <= W1 and (not C);
            when "1000" =>  -- JZ
                -- 设定PC
                LIR <= (W1 and (not Z)) or (W2 and Z);
                PCINC <= (W1 and (not Z)) or (W2 and Z);
                PCADD <= Z and W1;
                SHORT <= W1 and (not Z);
            when "1001" =>  -- JMP
                -- 设定算术运算
                M <= W1;
                S <= "1111";
                ABUS <= W1;
                LPC <= W1;
                -- 设定PC
                LIR <= W2;
                PCINC <= W2;
            when "1010" =>  -- OUT
                -- 设定PC
                LIR <= W1;
                PCINC <= W1;
                -- 短周期
                SHORT <= W1;
                -- 设定算术运算
                M <= W1;
                S <= "1010";
                ABUS <= W1;
            when "1011" =>  -- SSP
                SEL3<='1';
                M <= W1 or W2;
                if (W1 = '1') then
                        S <= "1000";
                elsif (W2 = '1') then
                        -- or S <= "1111"
                        S <= "1010";
                end if;
                ABUS <= W1 or W2;
```

```
                LAR <= W1;
                MEMW <= W2;
                -- 设定PC
                LIR <= W2;
                PCINC <= W2;
            when "1100" =>  -- PUSH
                M <= W1 or W2;
                CIN <= W3;
                if (W1 = '1') then
                        S <= "1111";
                elsif (W2 = '1') then
                        S <= "1010";
                elsif (W3 = '1') then
                        S <= "1111";
                end if;
                ABUS <= W1 or W2 or W3;
                LAR <= W1;
                MEMW <= W2;
                LONG <= W2;
                DRW <= W3;
                LIR <= W3;
                PCINC <= W3;
            when "1101" =>  -- MOV B->A
                -- 设定PC
                LIR <= W1;
                PCINC <= W1;
                -- 短周期
                SHORT <= W1;
                -- 选择逻辑运算, MOV 运算
                M <= W1;
                S <= "1010";
                ABUS <= W1;
                DRW <= W1;
                LDZ <= W1;
            when "1110" =>  -- STP
                STOP_2 <= W1;
            when "1111" =>  -- LSP
                M <= W1;
                S <= "1000";
                ABUS <= W1;
                LAR <= W1;
                MBUS <= W2;
                DRW <= W2;
                -- 设定PC
                LIR <= W2;
                PCINC <= W2;
```

```
                        when others =>  -- 公操作
                                -- 设定PC
                                LIR <= W1;
                                PCINC <= W1;
                end case;
                        when others =>
                                -- 不可能到这吧?
                end case;
        when "001" =>
--              SEL0<=ST0;
                -- SBUS = (ST0=0 or ST0=1) and W1
                SBUS <= W1;
                -- STOP = (ST0=0 or ST0=1) and W1
                STOP_2 <= W1;
                -- SHORT = (ST0=0 or ST0=1) and W1
                SHORT <= W1;
                -- SELCTL = (ST0=0 or ST0=1) and W1
                SELCTL <= W1;
                -- LAR = (ST0=0) and W1
                LAR <= W1 and (not ST0);
                -- LAR = (ST0=1) and W1
                ARINC <= W1 and ST0;
                -- MEMW = (ST0=1) and W1
                MEMW <= W1 and ST0;
                ST0_2 <= W1;
        when "010" =>
                -- SHORT = (ST0=0 or ST0=1) and W1
                SHORT<=W1;
                -- SELCTL = (ST0=0 or ST0=1) and W1
                SELCTL <= W1;
                -- STOP = (ST0=0 or ST0=1) and W1
                STOP_2<=W1;
                -- SBUS = (ST0=0) and W1
                SBUS<=W1 and (not ST0);
                -- LAR = (ST0=0) and W1
                LAR<=W1 and (not ST0);
                -- MBUS = (ST0=1) and W1
                MBUS<=W1 and ST0;
                -- ARINC = (ST0=1) and W1
                ARINC<=W1 and ST0;
                ST0_2<=W1;
        when "011" =>
                -- SELCTL = W1 or W2
                SELCTL <= '1';
                -- STOP = W1 or W2
                STOP_2 <= W1 or W2;
```

```
                                -- SEL0 = W1 or W2
                                SEL0<=W1 or W2;
                                -- SEL1 = W2
                                SEL1<=W2;
                                -- SEL2 = 0
                                -- SEL3 = W2
                                SEL3<=W2;
                        when "100" =>
                                -- SELCTL = (ST0=0 or ST0=1) and (W1 or W2)
                                SELCTL <= '1';
                                -- SBUS = (ST0=0 or ST0=1) and (W1 or W2)
                                SBUS <= W1 or W2;
                                -- STOP = (ST0=0 or ST0=1) and (W1 or W2)
                                STOP_2 <= W1 or W2;
                                -- DRW = (ST0=0 or ST0=1) and (W1 or W2)
                                DRW <= W1 or W2;
                                -- SEL0 = (ST0=0 or ST0=1) and W1
                                SEL0 <= W1;
                                -- SEL1 = ((ST0=0) and W1) or ((ST0=1) and W2)
                                SEL1 <= ((not ST0) and W1) or (ST0 and W2);
                                -- SEL2 = (ST0=0 or ST0=1) and W2
                                SEL2 <= W2;
                                -- SEL3 = (ST0=1) and (W1 or W2)
                                SEL3 <= ST0 and (W1 or W2);
                                ST0_2 <= W2;
                        when others=>
                end case;
        end process;
end arc;
```

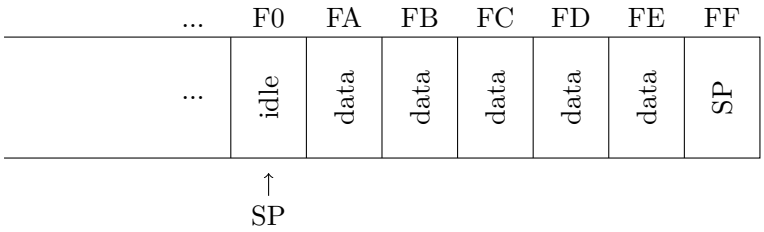
4.3 流水 CPU 的实现关键



流水 CPU 设计的关键是重叠无冲突的指令，在这里，取指操作 LIR 和指针加一操作 PCINC 与其他指

令不是互斥的，所以可以将取指和执行重合，从而缩短 cpu 执行指令的周期消耗。

4.4 新增的栈指令



本次实验在原指令集的基础上扩增了三条指令压栈指令 PUSHS, 存栈指针指令 SSP, 取栈指针指令 LSP. 同时复用了退栈指令 POPS(INC).

执行 LSP 指令以后，栈操作 PUSHS，POPS 有效，可用寄存器个数减少为 3 个。

执行 SSP 指令以后，存储器中的 SP 指针被更新，可用寄存器个数增加至 4 个。

新增栈以后，允许程序有更强大的功能，例如函数传参，天然拥有上下文无关文法的分析和编程功能等。

5 指令格式

名称	助记符	功能	指令格式		
			IR7 ∼4	IR3 IR2	IR1 IR0
加法	ADD R_d R_s	$R_d \leftarrow R_d$ and R_s	0001	R_d	R_s
减法	SUB R_d R_s	$R_d \leftarrow R_d$ minus R_s	0010	R_d	R_s
逻辑与	AND R_d R_s	$R_d \leftarrow R_d$ and R_s	0011	R_d	R_s
加 1	INC R_d	$R_d \leftarrow R_d$ and 1	0100	R_d	XX
取数	LD R_d $[R_s]$	$R_d \leftarrow [R_s]$	0101	R_d	R_s
存数	ST R_s $[R_d]$	$R_s \rightarrow [R_d]$	0110	R_d	R_s
C 条件转移	JC addr	$PC \leftarrow @ + C \cdot$ ofs	0111	ofs	
Z 条件转移	JZ addr	$PC \leftarrow @ + Z \cdot$ ofs	1000	ofs	
无条件转移	JMP $[R_d]$	$PC \rightarrow R_d$	1001	R_d	XX
输出	OUT R_s	$DBUS \leftarrow R_s$	1010	XX	R_s
取栈指针	SSP R_3	$R_3 \leftarrow [0xFF/SP]$	1011	1111	
压栈	PUSH R_s	$[0xFF/SP] \leftarrow R_s$	1100	11	R_s
移动	MOV R_d R_s	$R_d \leftarrow R_s$	1101	R_d	R_s
停机	STP	STOP	1110	XX	XX
存栈指针	LSP	$[0xFF/SP] \rightarrow R_s$	1111	1111	

6 上机调试通用流程

在每次实验之前, 我们都会先做下面几个工作。

6.1 写寄存器 (100)

按照之前的指令流程进行指令操作. 具体是：

1 SWCBA $\leftarrow$ 0b100

2 QD

3 $R_0 \leftarrow$ 0b10101010(0xA5)

4 QD

5 $R_1 \leftarrow$ 0b01010101(0x5A)

6 QD

7 $R_2 \leftarrow$ 0b10101010(0xA5)

8 QD

9 $R_3 \leftarrow$ 0b01010101(0x5A)

10 QD

11 CLR

6.2 读寄存器 (011)

按照之前的指令流程进行指令操作. 具体是：

1 SWCBA $\leftarrow$ 0b011

2 QD

3 QD

4 QD

5 CLR

如果读出 R_0, R_1, R_2, R_3 无问题, 则寄存器信号是可靠的,DBUS,SBUS 是可靠的.

6.3 运行代码 (000)

在寄存器中写入 $R_0 \leftarrow$ A5, $R_1 \leftarrow$ 5A, $R_2 \leftarrow$ 5B. 在存储器中写入指令

ADD R_0 R_1, 0001 0001

SUB R_0 R_2, 0010 0010

SUB R_1 R_2, 0010 0110

INC R_0, 0100 00XX

STOP, 1110 XXXX

检查结果 $R_0 =$ A5, $R_1 = 00, R_2 =$ 5B. 并且注意信号灯 C,Z 是否正常. 从此可以看出 ALU 部分功能有效,C,Z 标志有效.

6.4 进一步执行其他代码

至此, 花费 10 分钟就可以初步相信箱子正常工作.

至此可以自己写其他指令测试其他指令是否有效.

这里展示其中一个检测栈指令有效性的代码.

在寄存器中写入 $R_0 \leftarrow$ FA, $R_1 \leftarrow$ 0C, $R_2 \leftarrow 00, R_2 \leftarrow 55$.

在存储器中写入指令:

LSP, 1111 1111,

PUSH R_0, 1100 1100,

PUSH R_1, 1100 1101,

SSP, 1011 1111

检查存储器 [FF] = FC, [FE] = FA, [FD] = 0C. 检查 $R_3 = \mathrm{FC}$.

7 调试过程中的问题及讨论

7.1 代码无法导入箱子或者代码导入箱子以后测试程序仍然发生以前的问题

这里暂且不考虑箱子信号的问题, 具体原因可能有几个.

1 串口损坏, 导致写入箱子失败.

2 箱子板子识别损坏, 导致电脑无法针对这个芯片写入程序.

3 电脑只会写入一个文件项目, 如果在测试过程中换了文件, 需要在写入前确认是否写入目标文件是否是自己想要的文件.

7.2 程序运行失败

考虑信号问题

1 dp 单拍未设定, 导致程序直接运行到 stop 或者因为没有 stop, 在主存中持续运行.(表现行为是, 程序一闪而过或者 W1,W2,W3 信号灯在同一时间发亮.)

2 PCINC 信号失效.(表现行为是 PCINC 信号灯不亮或者 PCINC 信号灯已经亮但内部电路失效, 这种情况对于所有信号都是一样的, 所以下略, 只说明失效时的电路行为. 此时 PC 灯未能自增)

3 LIR 信号失效 (此时 IR 灯始终为初始状态).

4 AR 损坏, 表现行为是计算预期地址和实际显示 AR 不一致.

5 信号冲突, 并行信号如果有先后关系, 则有很大概率失败, 比如 STOP/SST0.

6 当上述情况均为发生, 应当考虑是否是程序代码有问题.

7.3 隐含敏感指令问题

设计的时候主要考虑每个信号什么时候应该为 1, 然后在进程里面相应的地方给输出赋值即可. 然后 VDHL 的进程会在敏感参数表发生改变的时候唤醒. 所以不能直接在进程里面修改敏感信号, 要新建一个 SIGNAL 记录有没有修改输出, 然后在每个时钟的下降沿修改输出, 这样就不会一直唤醒自己了.

8 设计调试小结

8.1 向阳曦的总结

在调试硬件及其编程语言时, 一定要注意硬件的不可靠性.

所以在所有测试开始之前一定要先做如下步骤:

1 测试数据通路是否有问题. 这个可以使用读写寄存器判断, 可以迅速推断问题.

2 如果已知之前有程序能够运行, 先跑这两个简单指令的程序, 如果无错, 则推测是在程序指令中有信号错误.

3 根据已知信号无错, 缩小以后的信号调试错误推测范围. 最后能够确定该系统在独立情况下信号无错的结论.

4 根据程序使用的信号无错, 推断程序的指令是否有问题.

5 根据程序的指令是否在表面上有问题, 推测是否是深层逻辑问题.

得到这些经验还是要拜托实验室里成群不能用的箱子. 里面的箱子坏的各有特色, 不带重样, 因此积累了大量经验. 关于程序部分, 我和张研究了一会如何设计指令. 从 x86 中挑选了大约十条指令, 并设计了指令格式. 等到开始编写时, 才发现 IR_{3210} 虽然和 SEL_{3210} 的值相同, 但是因为 SEL_{3210} 只能用于控制台读写寄存器的输出, 所以 IR_{3210} 是无法读出的. 未果, 寻弃.

最终我们选择完整地把 PUSHS,POPS 一整套栈操作完成. 但是按照正常流程 PUSHS, 应当是:

1. 得到 SP;2. 写入 SP;3.SP 下推.

但是 SP 在硬件中没有专门存储器, 寄存器只有 4 个, 占用其中一个的代价太高. 如果每次从主存取, 周期占用太多. 因此设计了新的 LSP 和 SSP. 采用一般数据库 prepare 的思路. 这时又遇到问题了.LSP 的流程应当是:

1. 得到 SP 在主存地址;2. 读入 SP 到寄存器中.

这里我们如何去硬编码 SP 的地址是个问题.

深思熟虑以后. 我们将 SP 地址从寄存挪到主存, 再到最后敲定一个特殊的地址. 我们选择了 FF. 因为这个根据任意 8 位 2 进制数字 A 或上 $\overline{A}$ 永远等于 FF. 所以限定 A, B 为相同寄存器, 就能在一个周期得到硬编

码的 FF 写入 AR. 最后实际的设计为:

1. 要求 A,B 选择的寄存器一致;2. 第一个周期将 $A|\overline{B}$ 送入 AR.3. 取出 FF 上的值到 A 寄存器上.

这样只需要额外一条指令就能临时得到 SP, 额外另一条指令就能存储 SP.

最后, 我们将栈设计为向下生长, 将 POPS 合并到 INC SP 中. 又节约了一条指令空间, 将这个指令编码给了另一个指令 MOV.

8.2 张逸群的总结

课程设计让我对计算机组成原理有了更深刻的了解.

就课程设计的要求来说, 流水机器和非流水机的差别不大, 无非是什么时候取指令的问题. 非流水机需要一个额外的时钟来取值, 由于 TEC8 最多只有 3 个时钟, 也就是说其他操作要在 2 个时钟内完成, 这个给设计带来了很大的不便. 流水机在执行的时候顺便取指令, 这样能节约出来一个周期.

设计的时候主要考虑每个信号什么时候应该为 1, 然后在进程里面相应的地方给输出赋值即可. 然后 VDHL 的进程会在敏感参数表发生改变的时候唤醒. 所以不能直接在进程里面修改敏感信号, 要新建一个 SIGNAL 记录有没有修改输出, 然后在每个时钟的下降沿修改输出, 这样就不会一直唤醒自己了.

还有个问题是指令的问题, 由于 TEC8 的模式是前四位是指令, 后四位是参数, 由于后四位只能读入不能输出, 所以参数必须要手动给出, 而且用不了立即数的形式, 这样的结果是一些原本很简单的操作会变得非常麻烦, 需要用寄存器中转一下. 有些原本可以隐式给出的参数也必须需要手动给出, 不过有个例外是 FF, 这个可以通过 ALU 算出来, 因为 $A|(!A) = FF$, 同理 $A\&(!A) = 0$, 也可以不用手动给出. 不过这样的代价是浪费了一个周期来计算隐式给出的参数.

剩下的问题都不是太多了, 按照求出来的表达式写一下就可以了.