**山东大学计算机组成原理**

**课程设计实验报告**

**------------简单模型机实现（微程序）**

**（硬布线）**

班级：15.1

目录

[一、总述 3](#_Toc501298654)

[实验目的 3](#_Toc501298655)

[实验软件和硬件环境 3](#_Toc501298656)

[二、微程序控制的运算器设计 3](#_Toc501298657)

[设计目的： 3](#_Toc501298658)

[设计要求： 3](#_Toc501298659)

[微程序控制器的结构： 4](#_Toc501298660)

[微指令的格式： 4](#_Toc501298661)

[主要部件的设计： 4](#_Toc501298662)

[三、微程序控制的存储器读写系统设计 6](#_Toc501298663)

[设计目的： 6](#_Toc501298664)

[设计要求： 6](#_Toc501298665)

[结构与信号索引： 6](#_Toc501298666)

[微指令的格式： 7](#_Toc501298667)

[四、简单模型机设计——微程序实现 7](#_Toc501298668)

[设计目标： 7](#_Toc501298669)

[拟定指令系统 8](#_Toc501298670)

[确定总体结构 8](#_Toc501298671)

[关键部件设计： 9](#_Toc501298672)

[控制方式 10](#_Toc501298673)

[编制指令流程及书写微程序 12](#_Toc501298674)

[调试 14](#_Toc501298675)

[五、模型及设计——硬布线实现 14](#_Toc501298676)

[设计目标： 14](#_Toc501298677)

[总体结构： 14](#_Toc501298678)

[各部件的设计 15](#_Toc501298679)

[六、问题分析 17](#_Toc501298680)

[七、实验感悟 17](#_Toc501298681)

[附录 18](#_Toc501298682)

[1.微程序控制的运算器： 18](#_Toc501298683)

[2.微程序控制的存储器读写系统设计： 18](#_Toc501298684)

[3、微程序实现的模型机 20](#_Toc501298685)

[4、硬布线实现的模型机 26](#_Toc501298686)

# 一、总述

## 实验目的

通过该课程设计的学习，总结计算机组成原理课程的学习内容，运用计算机原理知识，设计一台模型机，从而巩固课堂知识、深化学习内容、完成教学大纲要求，学好这门专业基础课。

## 实验软件和硬件环境

QuartusII， JYS计算机组成实验箱

win10系统，内存8G，CPU i7-5500U

# 二、微程序控制的运算器设计

## 设计目的：

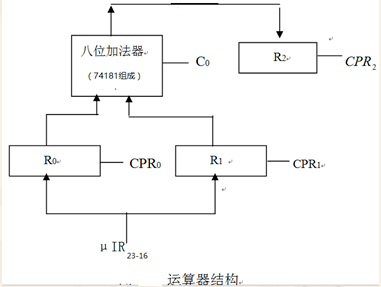
1.熟悉简单运算器的结构。

2.熟悉微命令的产生和时序。

3.熟悉运算器功能测试。

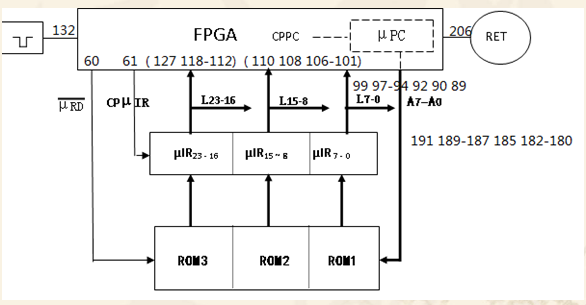
## 设计要求：

运算器结构如下图所示。



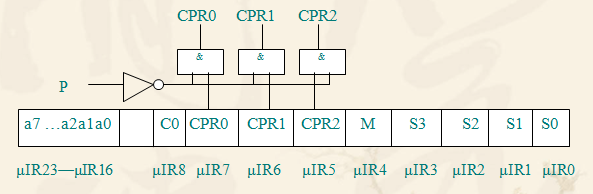
R0、 R1、 R2均为 D 触发器组成的八位寄存器，在打入脉冲 CPRi的作用下，接收数据输入端提供的信息送入 Ri中。R0、R1、R2均为D触发器组成的八位寄存器，在打入脉冲CPRi的作用下，接收数据输入端提供的信息送入Ri中。μIR23-16为微指令寄存器的高八位，可定义为操作数。进位信号C0、打入脉冲CPR0、CPR1、CPR2、M、S0、S1、S2、S3均由微指令寄存器的μIR8和μIR7--μIR0产生。

## 微程序控制器的结构：



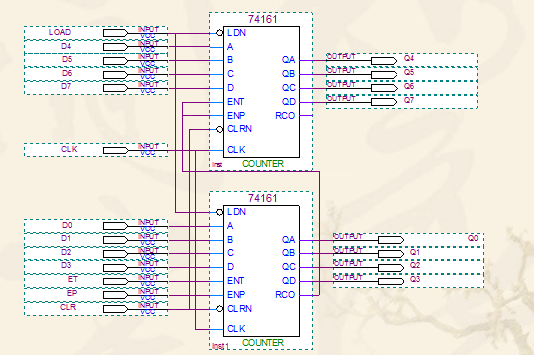
控制存储器 24 位, ROM3# 、 ROM2#、 ROM1#、 μ IR23-16、 μ IR15-8、 μIR7-0、L23-16、 L15-8、 L7-0、 μRD 、 CPμIR、 A0～A7均连入 FPGA 系统中。

## 微指令的格式：

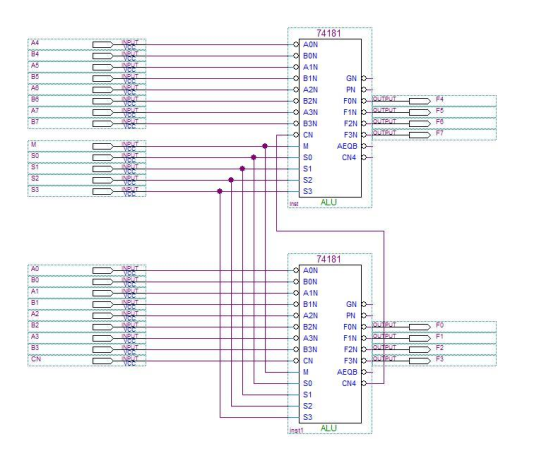


## 主要部件的设计：

**upc：**



**ALU的设计：**



总电路图见附录。

# 三、微程序控制的存储器读写系统设计

## 设计目的：

1.熟悉随机存储器读写系统结构设计。

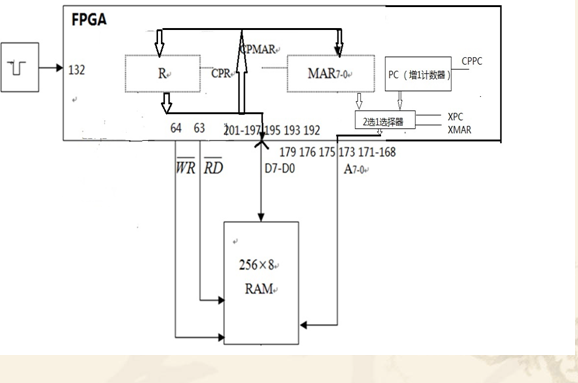
2.熟悉随机存储器的读写时序。

3.熟悉随机存储器的读写操作的微程序实现。

4.熟悉随机存储器的功能测试

## 设计要求：

根据pc访问内存，取出地址Ad1，根据Ad1访问取出数据X，将X保存在Ad2地址单元。结构框图如下所示。

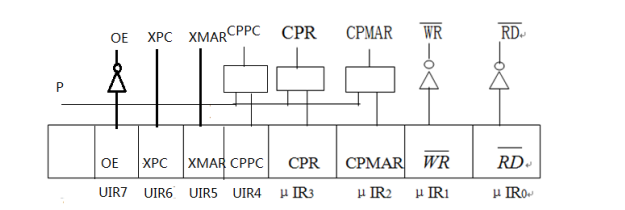


## 结构与信号索引：

MAR为地址寄存器。R寄存器必须为具有三态输出的寄存器，直接调用74374即可。PC为带清零端的8位增1计数器（在上一部分的运算器设计中已经实现过）。FPGA引脚信号如下：

①A7～A0 地址信号，方向指向 RAM。  
②D7～D0 数据信号，双向， RAM 的数据通过该数据线读出或写入。  
③存储器读命令 RD，低电平有效。  
④存储器写命令 WR，低电平有效。

## 微指令的格式：

μIR3为 1 产生 CPR； μIR2为 1 产生 CPMAR； μIR4为 1 产生 CPPC； μIR1为 1  
产生 WR； μIR0为 1 产生 RD； μIR5为 1 产生 XMAR 选择信号； μIR6为 1 产生 XPC  
选择信号； μIR7为 1 产生寄存器 R 的三态允许信号 OE。  
因为利用微程序，还必须设计微地址增 1 计数器 µPC。  
微程序的编制，由存储器的操作确定：  
① 读出地址 ad1 放入 MAR 寄存器：  
XPC、 RD、 CPMAR 有效，即微指令： 000045H  
② M（ad1）即 x 送入寄存器 R：  
XMAR、 RD、 CPR、 CPPC（PC 增 1）即微指令： 000039H  
③ 读出 ad2 送入 MAR ，即微指令： 000045  
④ x 送入地址 ad2 中  
XMAR、 WR、 OE 有效，即微指令： 0000A2H  
调试步骤：  
① 设计存储器外围逻辑并下载至 FPGA 中。  
② 设计微程序并将微程序以十六进制代码键入 ROM3# 、 ROM2#、 ROM1#中  
③ 按一次脉冲键读出并执行一条微指令，并检查微指令执行情况，直至结  
束。  
④ 在 RAM 中，预先写入 ad1、 ad2、 x 具体数据。

该部分的详细电路图见附录。

# 四、简单模型机设计——微程序实现

## 设计目标：

本次设计要完成一个用微程序设计的简单模型机。 ALU 部分用 74181 芯片与  
74182 芯片来设计。

## 拟定指令系统

**①基本字长：**

模型机基本字长定位8位，存储容量为256\*8位，

**②指令类型：**

指令类型有三种：单操作数指令、双操作数指令和无操作数指令。操作码OP共四位，最多可以定义16条指令。

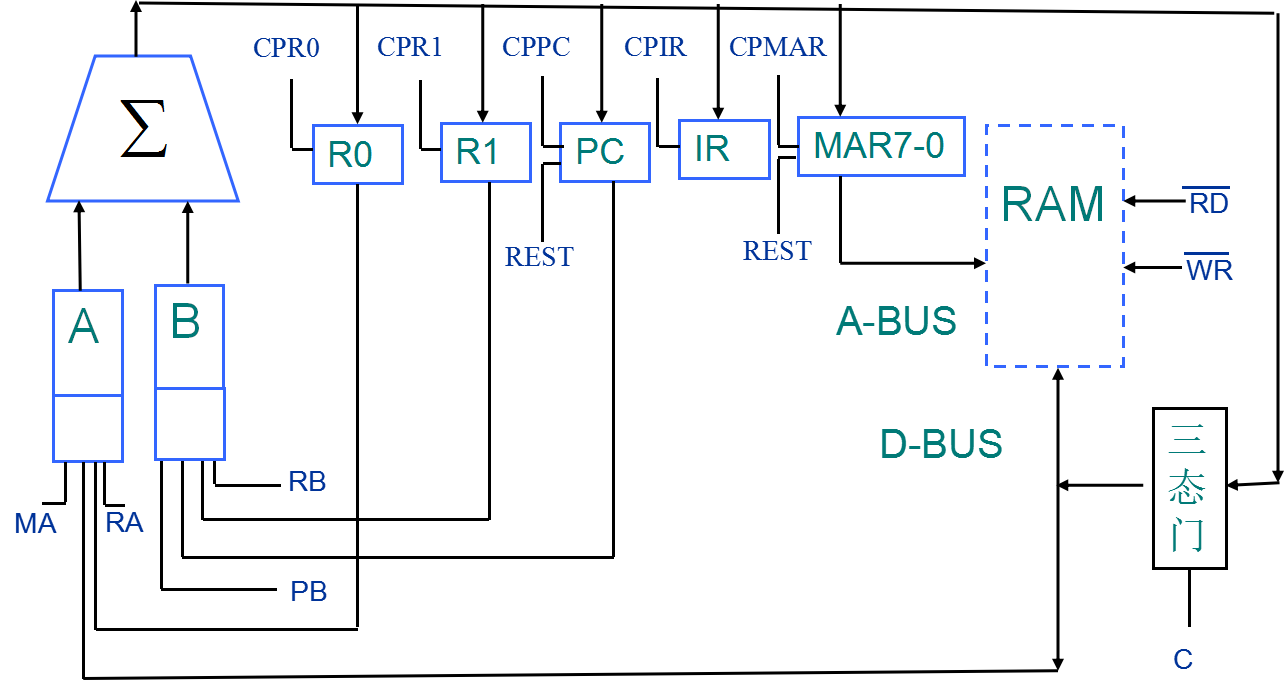
**③寻址方式：**

当寻址方式位为0，是寄存器寻址，操作数在指定的寄存器中，相应的寄存器号位为0是R0，为1是寄存器 R1；当寻址方式位为1时，寻址方式位和寄存器号位组合，

10：是立即数寻址，操作数在指令的下一个单元；

11：是直接寻址，操作数地址在指令的下一个单元。

## 确定总体结构



**①寄存器组设置：**

R0、R1为通用寄存器，8位。

IR为指令寄存器，8位。

PC程序计数器，8位。

MAR为地址寄存器，8位。

**运算器ALU的设置：**

加法、减法、逻辑运算采用两片74181芯片实现。

乘法操作采用一个74284芯片和一个74285芯片组合实现

**选择器的设置**

连入A选择器的数据来源是RAM的读出数据和R0寄存器的数据。

连入B选择器的数据来源是PC的数据和R1的数据

**数据通路**

模型机的数据通路是以总线为基础,以CPU为核心构成的。

**取指令：**

MA A直传 CPIR

RAM → 选择器A →∑ → Bus → IR

**送指令地址**

PB B直传 CPMAR

PC → 选择器B →∑ → Bus → MAR

**指令计数器＋1**

PB A加B加1 （A为0） cppc

PC → 选择器B → ∑→ Bus → PC

**R0→R1**

RA A直传 CPR1

R0 → 选择器A →∑ → Bus → R1

**R1→RAM**

RB B直传 C WR

R1 → 选择器B →∑ → Bus → RAM

## 关键部件设计：

**具体设计电路见附录。**

总体结构中，虚线框内的RAM是FPGA之外预先配置好的。

**（1）ALU的逻辑设计**

由两片74181组成。

1. **寄存器的设计**

①结构中R0、R1通用寄存器，可存放操作数或结果、中间结果，每个寄存器均由8个D触发器构成。

②在CPRi的作用下接收总线的数据送入寄存器，输出连入选择器。

③指令寄存器IR1、IR2其结构同通用寄存器。

④结构中MAR地址寄存器是一个带复位的寄存器，带复位是指当有复位信号时，MAR清零。

统一调用系统自带的74273芯片实现。

⑤程序计数器也是一个带复位信号的八位寄存器，PC加1是通过加法器实现的。复位信号RET 的作用是有复位信号时，计数器PC清零。

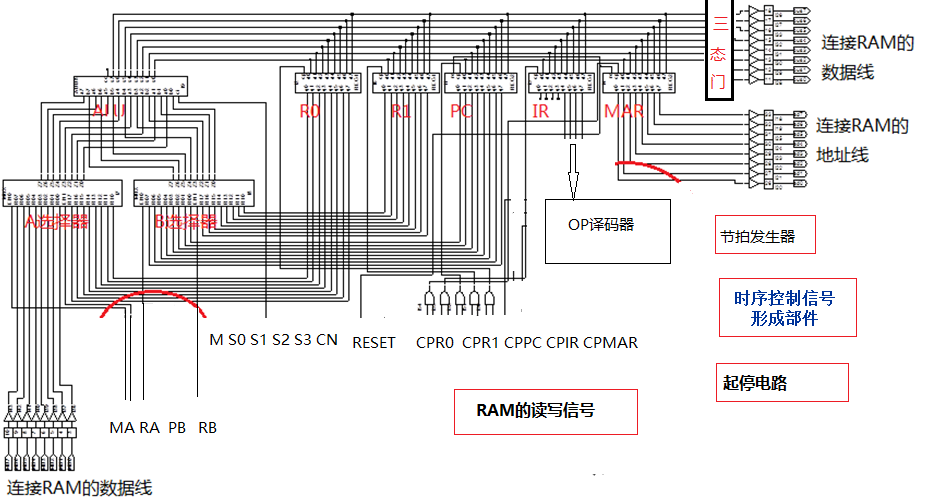
1. **三态门的设计**

采用8个tri原件实现，当控制输入端为高电平时1，低电平时为高阻态。

**（4）部件之间的连接**

由系统结构图可看出，部件之间的连接是采用以CPU为中心的总线连接方式。运算器的输出通过总线BUS连接到所有寄存器和存储器的输入端，除指令寄存器IR和地址寄存器MAR的输出端外，其它部件的输出端分别送入选择器A和选择器B。

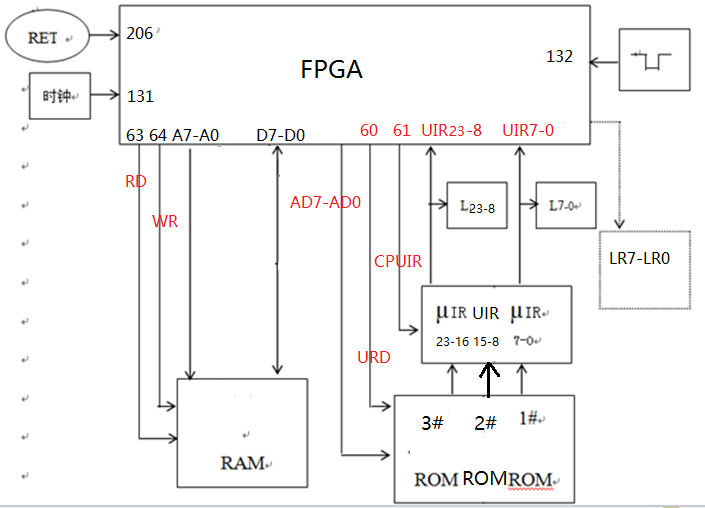
连线图如下所示。



## 控制方式

**（一）微程序实现**

控制方式有两种，硬布线逻辑电路方式和微程序方式。先考虑微程序实现方式。控制命令是确定信息的流向，不同的数据通路需要不同的控制命令。涉及到了许多控制命令例如CPR0、CPMAR、MA、RB等等，这些控制命令由以下内容， 主要由控制存储器ROM3#、 ROM2#、ROM1#、微指令寄存器 IR23-16、 IR15-8、 IR7-0构成。



P脉冲的低电平用做控制存储器读命令μRD

P脉冲的上升边沿将读出的微指令送 μIR

**（1）微指令格式**

微指令字长24位即 μIR23～ μIR0。

**（2）微指令字段定义**

ALU控制： μIR21·μIR20μIR19·μIR18μIR17·μIR16

M S3 S2 S1 S0 C0

三态门控制： μIR6

0 高阻态 使C＝1

1 三态门使能 使C＝0

停机控制： μIR3

0 G=0，运行

1 G=1，停机

A选择器控制： μIR15·μ IR14

0 0 备用

0 1 RA

1 0 MA

1 1 备用

B选择器控制： μIR13· μIR12

0 0 备用

0 1 PB

1 0 RB

1 1 备用

输出分配： μIR11· μIR10· μIR9

0 0 0 备用

0 0 1 CPR0

0 1 0 CPR1

0 1 1 CPPC

1 0 0 CPIR

1 0 1 CPMAR

1 1 0 备用

1 1 1 备用

存储器读写控制： μIR5· μIR4

1 0 RD

0 1 WR

后继微地址形成方式：

μIR2· μIR1· μIR0

0 0 0 备用

0 0 1 μ PC＋1 顺序执行

0 1 0 JP无条件转移，地址由 IR15-8提供。

0 1 1 QJP高四位按操码转移，低4位为0。

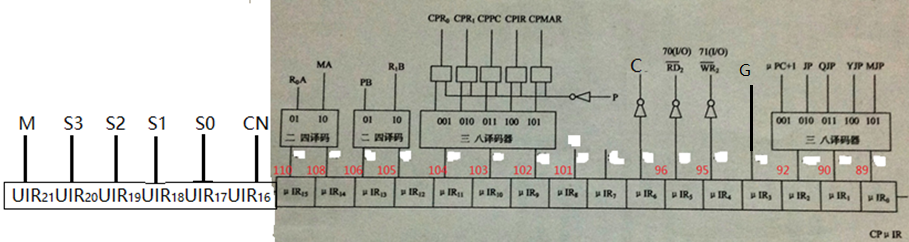
1 0 0 YJP给定高4位低4位按源寻址方式转移。

1 0 1 MJP给定高4位低4位按目寻址方式转移。

1 1 0 备用

1 1 1 备用

**（3）微命令形成逻辑**

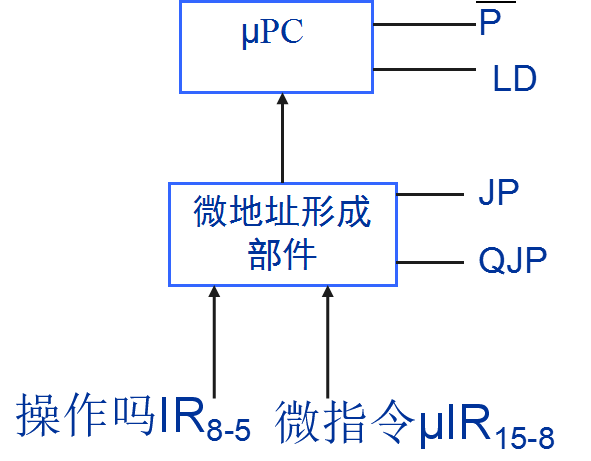


**（4）后继微地址产生逻辑**

为简单起见只选三种后继微地址生成方式

即增量方式、无条件转移方式、按操作码转移方式。

其结构框图如图9所示。



当LD＝1时，微程序计数 μPC执行加1操作。

当LD＝0时且JP＝1时，无条件转移，有微指令的中八位提供转移地址。

当LD＝0时且QJP＝1时，按操作码转移。

功能：多路选择器

当JP=1，QJP=0时，Y7Y6Y5Y4Y3Y2Y1Y0 = μIR15 μIR14 μIR13 μIR12 μIR11 μIR10 μIR9 μIR8

当JP=0，QJP=1时，Y7Y6Y5Y4Y3Y2Y1Y0 = IR7 IR6 IR5 IR4 0000

链接时， Y7Y6Y5Y4Y3Y2Y1Y0连接µPC的D7~D0, µPC的RE接高电平vcc。

## 编制指令流程及书写微程序

编写程序 代码

MOV1 05#，R0 0001 1000

MOV2 01#，R1 0000 0101

SUB R0， R1 0011 0001

MOV3 R1，(FA#) 0100 0111

1111 0000 ；F0

ADD R0,R1 0101 0001

MOV4 R1,(FA#) 0110 0111

1111 0001 ;F1

AND R0,R1 0101 0001

MOV5 R1,(FA#) 0110 0111

1111 0010 ;F2

XOR R0,R1 1001 0001

MOV6 R1,(FA#) 1000 0111

1111 0011 ;F3

OR R0,R1 1011 0001

MOV7 R1,(FA#) 1100 0111

1111 0100 ;F4

MUL R0,R1 1101 0001

MOV8 R1,(FA#) 1110 0111

1111 0101 ;F5

HALT 1111 0000

操作码二进制代码

MOV1：0001

MOV2：0010

SUB: 0011

MOV3: 0100

ADD: 0101

MOV4: 0110

AND: 0111

MOV5: 1000

XOR: 1001

MOV6: 1010

OR: 1011

MOV7: 1100

MUL: 1101

MOV8: 1110

HALT: 1111

ADD ： 0011

微程序入口（十六进制代码）

取指周期微指令 入口：00H

MOV1执行周期微指令 入口：10H

MOV2执行周期微指令 入口： 20H

SUB执行周期微指令 入口：30H

MOV3执行周期微指令 入口：40H

ADD执行周期微指令 入口：50H

MOV4执行周期微指令 入口：60H

AND执行周期微指令 入口：70H

MOV5执行周期微指令 入口：80H

XOR执行周期微指令 入口：90H

MOV6执行周期微指令 入口：A0H

OR执行周期微指令 入口：B0H

MOV7执行周期微指令 入口：C0H

MUL执行周期微指令 入口：D0H

MOV8执行周期微指令 入口：E0H

HALT执行周期微指令 入口：F0H

## 调试

①微程序经过检查无误后通过“计算机组成原理与系统结构安装软件”以十六进制写入3#ROM2#ROM和1#ROM的相应单元中去。

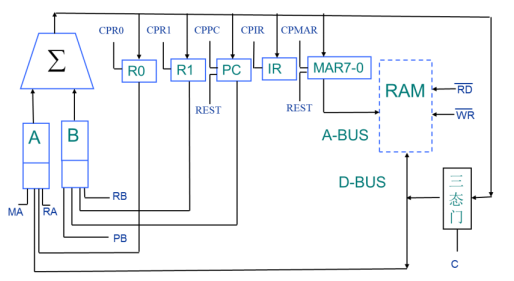
②按复位键RET使MAR清0、指令计数器PC清0，保证从存储器0号单元取指令。使微程序计数器UPC清0，保证从3#RAM、2#ROM、1# ROM的0#单元取出取指令微程序的第一条微指令。

2、 执行微程序  
按复位键后, µPC,PC,MAR 为 0。  
按一次脉冲键产生一负脉冲(作为 μRD )，将 ROM1# 、 ROM2# 0 号单元的  
16 位微指令代码读出，用 的上升沿将微指令送入 µIR15-0，看是否正确。第一条微指令产生的命令是： MA、RD2 、CPIR，后继微地址产生方式为 µPC+1，其操作是： RD2 读 RAM，单元地址为 0，即读 0 号单元的内容。 0 号单元的内容是一条指令，指令代码读出后，在 MA 的作用下，进入加法器至总线。按一次脉冲键又产生一负脉冲。该负脉冲反相后的上升沿产生 CPIR,将上条微指令读出的指令代码送 IR，同时上升沿还将 µPC+1。该负脉冲的低电平用以读出 µPC 指示的第二条微指令。

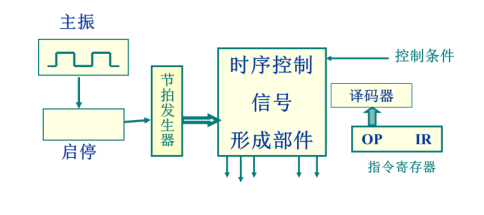
# 五、模型及设计——硬布线实现

设计目标：  
本次设计要实现一个硬布线控制的模型机， ALU 部分用 74181 芯片与 74182芯片来设计。本模型机所要实现的功能包括：加法、减法、增 1、减 1、两数之和增一、逻辑与、或、异或、直传、非和停机。 并对该模型机进行一定的优化，使其节省节拍信号数，以获得更快的执行速度。

字长采用八位字长， 指令格式与微程序设计中相同。

总体结构：  


硬布线逻辑电路控制器的结构如下图所示：



关键部件设计：

具体设计电路图见附录。

ALU，二选一选择器，三态门，寄存器组的设计与微程序实现的模型机一致。

部件之间的连接是采用以 CPU 为中心的总线连接方式。 ALU 的输出通过总线BUS 连接到所有寄存器和存储器的输入端，除指令寄存器 IR 和地址寄存器 MAR的输出端外，其它部件的输出端分别送入选择器 A 和选择器 B。

## 各部件的设计

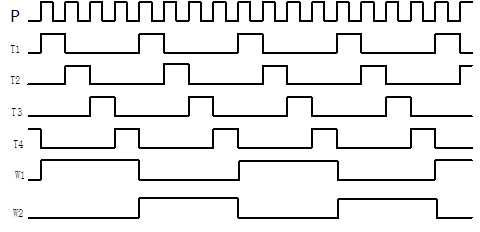
**主振和启停电路的设计**

在微程序控制的模型机中我们已经实现了可以控制的连续脉冲的启停电路。微程序控制的启停电路在硬布线实现的模型机中也可以使用，但发现运算指令的执行周期所需的时钟周期数最少，可优化，因而选择对该种指令进行优化。只需要当期执行周期的第一个时钟周期执行完成后，对节拍发生器中的模四增一计数器进行清零即可。

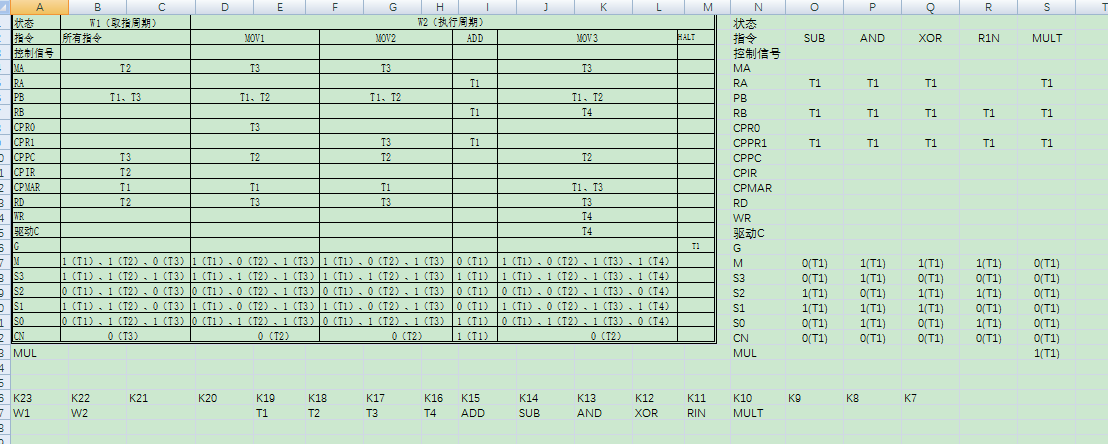
**节拍发生器（TIMING）设计与实现**

根据指令执行流程，设计时序为：分取指周期和执行周期，每个周期为4节拍，波形图如下：

W1高电平时是取指周期的4节拍，W2高电平时是执行周期的4节拍。



**时序控制信号产生部件的设计;**



MA = W1\*T2 + W2\*T3\* (MOV1 +MOV2+MOV3)

RA = W2\*T1\*(ADD + SUB+ AND + XOR +MULT)

PB = W1\*(T1+T3) + W2\*(T1+T2)\*(MOV1 + MOV2 + MOV3)

RB = W2\*T1\*(ADD+SUB+AND+XOR+R1N+MULT) + W2\*T4\*MOV3

CPR0 = W2\*T3\*MOV1

CPR1 = W2\*T1\*(ADD+SUB+AND+XOR+R1N+MULT) + W2\*T3\*MOV2

CPPC = W1\*T3 + W2\*T2\*(MOV1 + MOV2 + MOV3)

CPIR = W1\*T2

CPMAR = W1\*T1 + W2\*T1\*(MOV1 + MOV2 + MOV3) + W2\*T3\*MOV3

RD = W1\*T2 + W2\*T3\*(MOV1+MOV2+MOV3)

WR = W2\*T4\*MOV3

驱动C = W2\*T4\*MOV3

G = W2\*T1\*HALT

M = W1\*(T1 + T2 + ~T3) + W2\*(MOV1\*(T1 + ~T2 + T3) + MOV2\*(T1 + ~T2 + T3) + MOV3\*(T1 + ~T2 + T3 + T4)) + W2\*T1\*(AND+ XOR + R1N) + W2\*~T1\*(ADD + SUB + MULT)

= W1\*(T1 + T2 + ~T3) + W2\*(MOV1+MOV2+MOV3)\*(T1 + ~T2 + T3) + W2\*MOV3\* T4 + W2\*T1\*(AND+ XOR + R1N) + W2\*~T1\*(ADD + SUB + MULT)

S3 = W1\*(T1 + T2 + T3) + W2\*(T1+T2+T3)\*(MOV1+MOV2+MOV3) + W2\*T4\*MOV3 + W2\*T1\*(ADD+AND) + W2\*~T1\*(SUB + XOR + R1N + MULT)

S2= W1\*(~(T1+T3)+T2)+W2\*(~(T1+T2)+T3)\*(MOV1+MOV2)+W2\*(~(T1+T2+T4)+T3)\*MOV3+

W2\*T1\*(SUB+XOR+R1N) + W2\*~T1\*(ADD+AND + MULT)

S1 = W1\*(T1 + T2 + ~T3) + W2\*(T1 + ~T2 + T3)\*(MOV1+MOV2+MOV3) + W2\*T4\*MOV3 + W2\*T1\*(SUB+AND+XOR) + W2\*~T1\*(ADD+ R1N + MULT)

S0 = W1\*(~T1+T2+T3)+W2\*(MOV1+MOV2)\*(~T1+T2+T3)+W2\*MOV3\*(~(T1+T4)+T2+T3)+ W2\*T1\*(ADD + AND + R1N) + W2\*~T1\*( SUB + XOR + MULT)

CN = W1\* ~T3 + W2\*~T2\*(MOV1+MOV2+MOV3) + W2\*(T1\*ADD + ~T1\*(SUB + AND + XOR + R1N + MUTL))

MUL = W2\*T1\*MULT

**RAM编写代码**

地址:

00

数据:

18,05,29,02,31,47,F0,18,05,29,02,51,47,F1,18,05,61,47,

F2,18,05,71,47,F3,18,05,29,02,81,47,F4,18,05,29,02,91,

47,F5,18,05,29,02,A1,47,F6,18,05,B1,47,F7,18,05,C1,47,

F8,18,05,29,02,D1,47,F9,E0;

## 调试：

1、按复位键RET

使MAR清0、指令计数器PC清0，保证从存储器0号单元取指令。

2、按单脉冲键，启动程序执行。

# 六、问题分析

1.做硬布线的模型机的时候，出现了一系列的问题。经过排查原因主要有两个，一个是由于脉冲波形的问题，需要在 CPR0、 CPR1、CPPC、 CPIR、 CPMAR 信号的 P 输入端加上一个非门。

2.uIR14（pin108无法使用）

需要进行一定的设置。首先打开选择208C8的页面，点击页面中部右侧的Device and Pin Options按钮。随后在上方的卡片中选择Dual-Purpose Pins，最后将nCEO改为如图所示的Use as regular I/O即可。

3.乘法器

因为这次实验我们组采用的是用两片74285实现乘法器，输入是两个四位，输出是两个八位，而且乘法器的操作并不涉及到M、S0-S3，所以要把乘法器和其他ALU部件放在一个层次上，用一个选择信号控制，即UIR7。

4.微程序控制的存储器读写系统的设计的时候，无法正确读写 RAM。经检查， 是因为没有添加CPuIR。

# 七、实验感悟

微指令实现模型机的操作的关键在于整个模型机的运行是按步骤实现的，一步步执行整个程序，对于下地址形成部件，选择器用来提取IR中高四位来确定微程序的入口，uPC是用来执行确定执行下一步微指令。逻辑设计其实是将具体操作划分为指令，指令又划分为微指令，微指令中含有uIR的各种端口来响应操作。

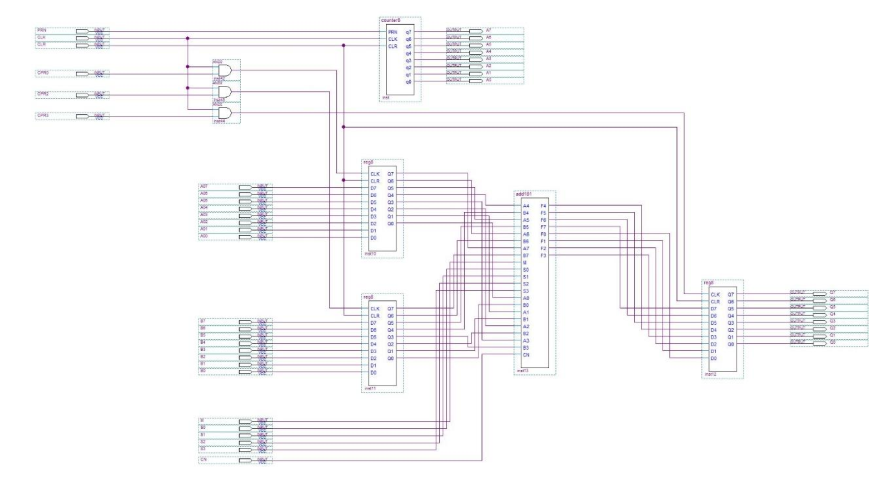
硬布线电路不用ROM存储器，直接通过相应的硬布线电路来设计这些控制信号，不同于微程序模型机通过微指令信号中的uIR23—uIR0控制。

无论是底层基础元件的设计，还是整体系统的连接都是不小的挑战，工作极为繁琐需要个人的细致和耐心，对于脉冲，节拍和指令有了进一步的理解，继修读了《计算机组成原理》后算是真正明白了一台基本计算机的执行过程，对计算机整体有了进一步的理解。

# 附录

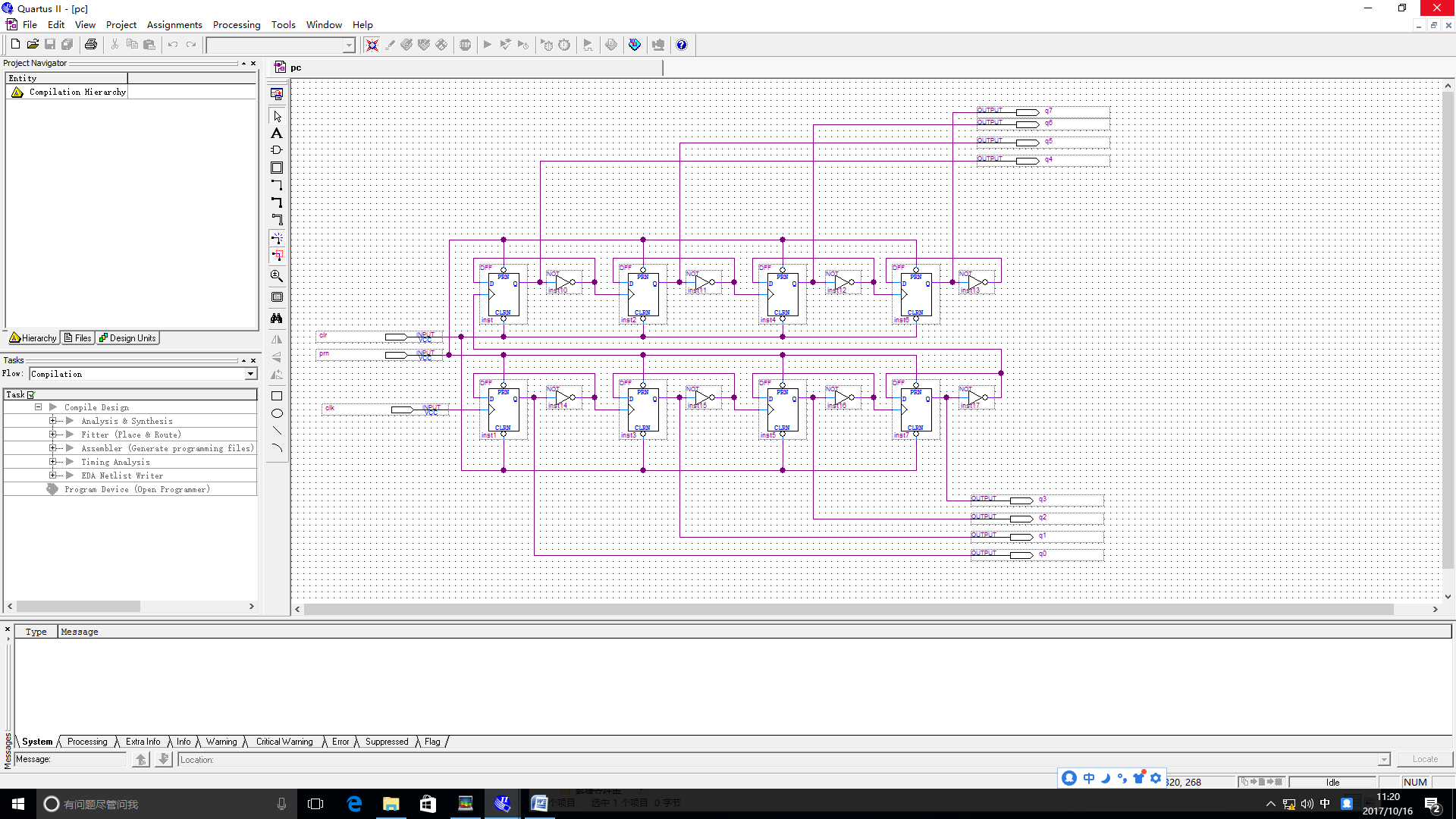
## 1.微程序控制的运算器：

主电路图：

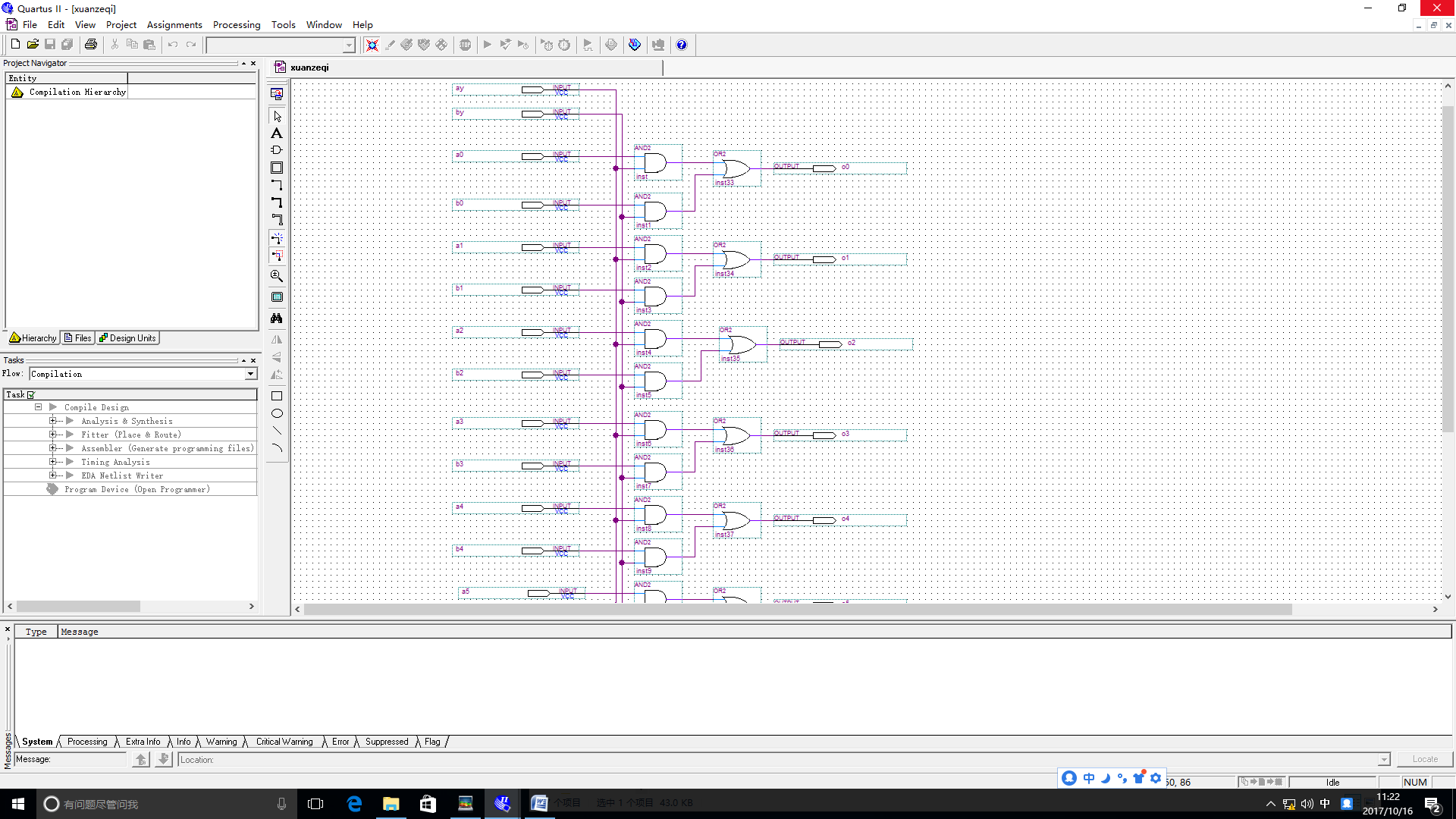


## 2.微程序控制的存储器读写系统设计：

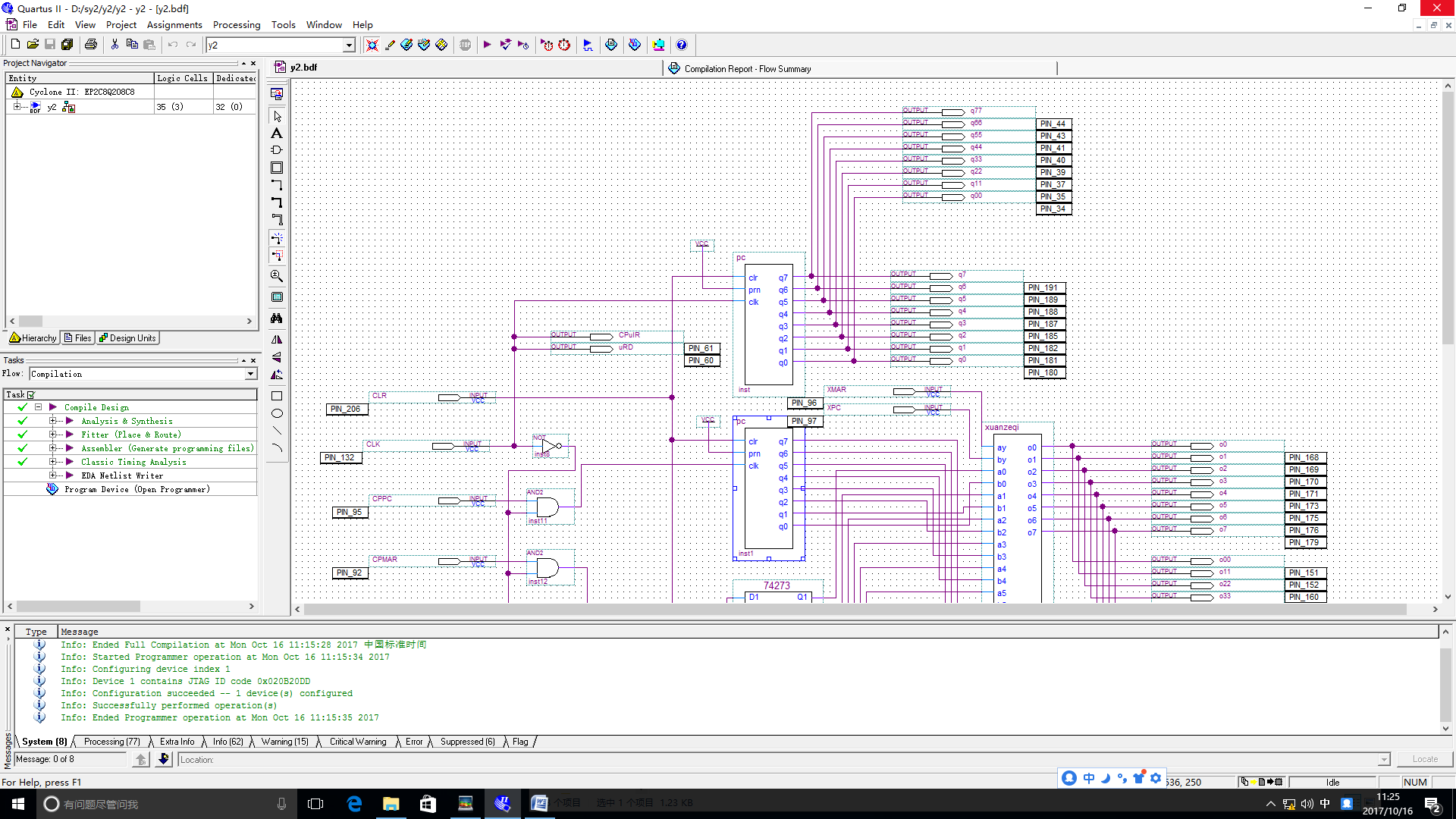
Pc的设计：

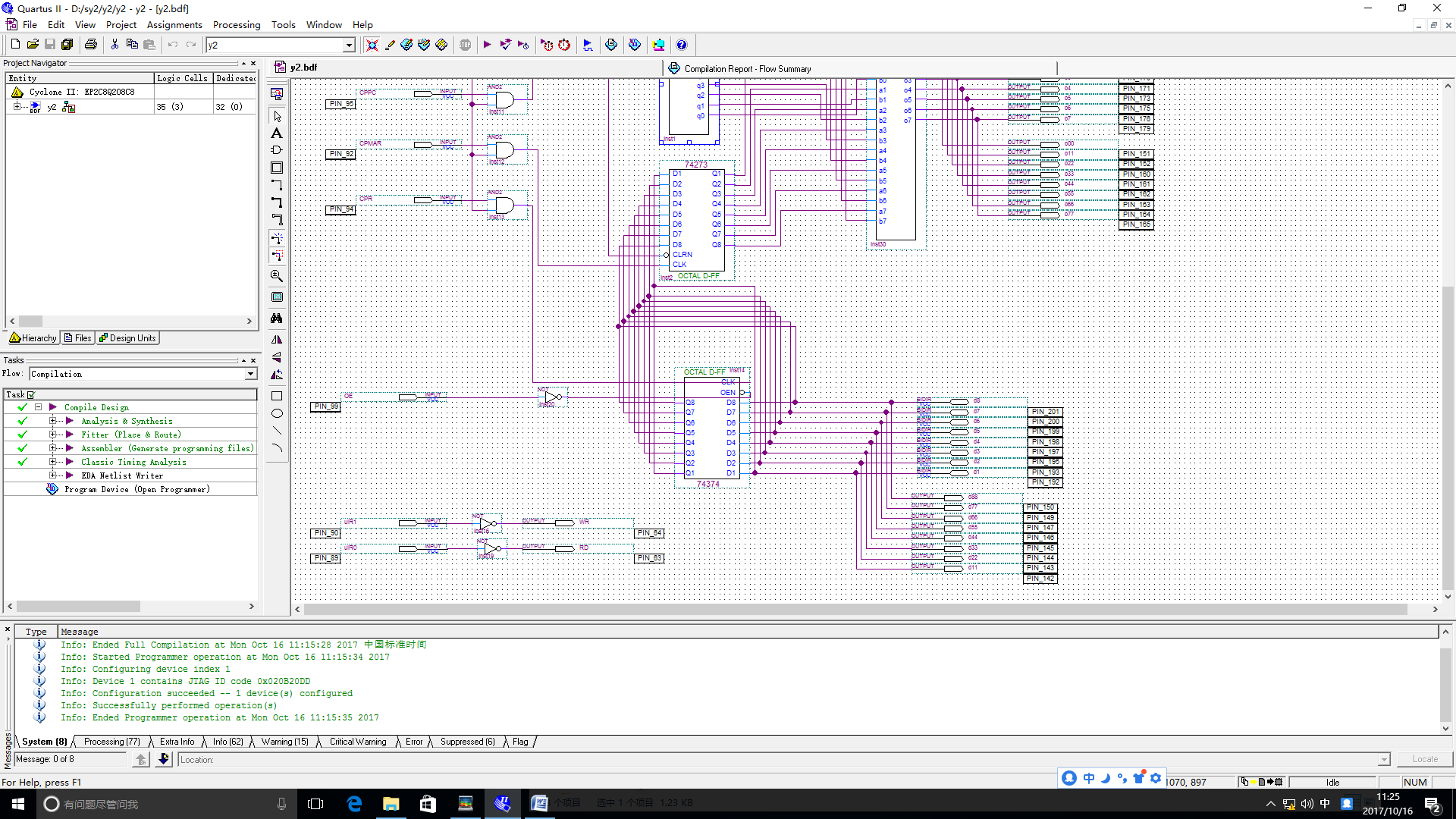


二选一寄存器：



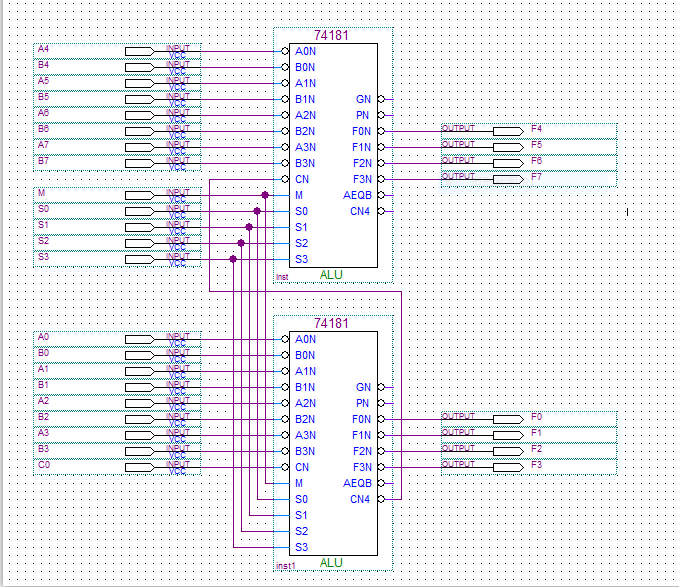
总电路：



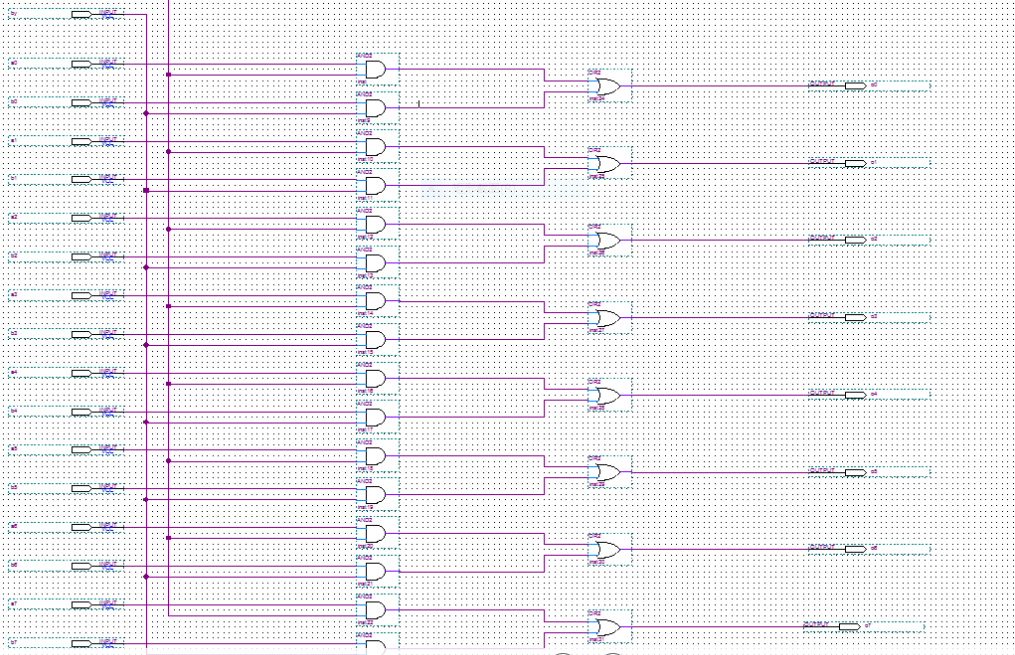


## 3、微程序实现的模型机

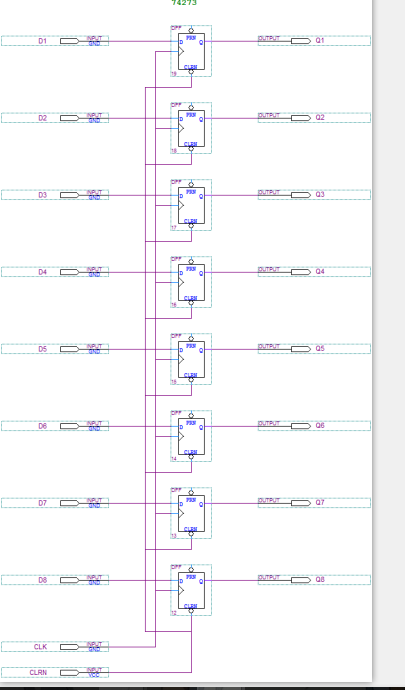
ALU



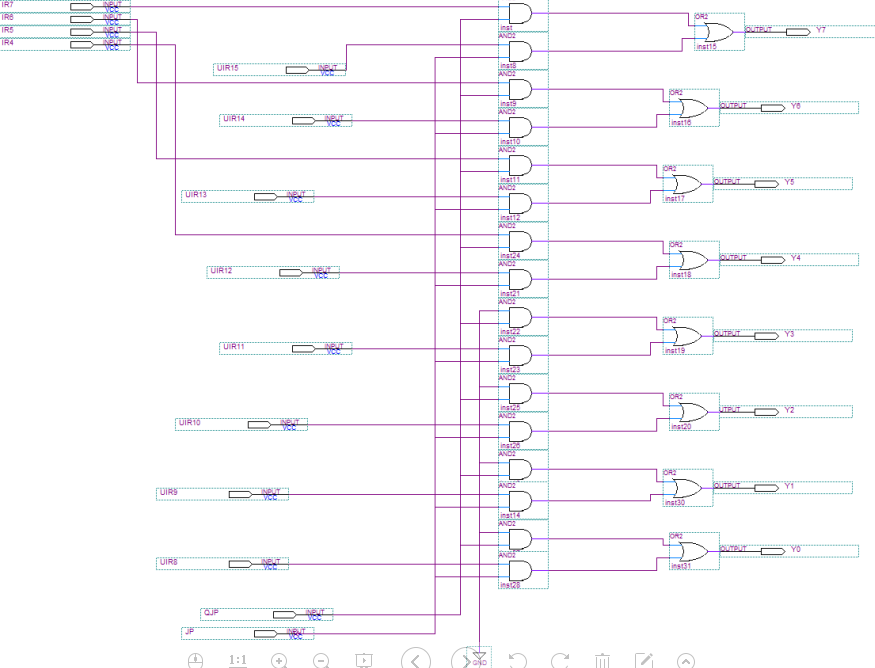
二选一选择器：



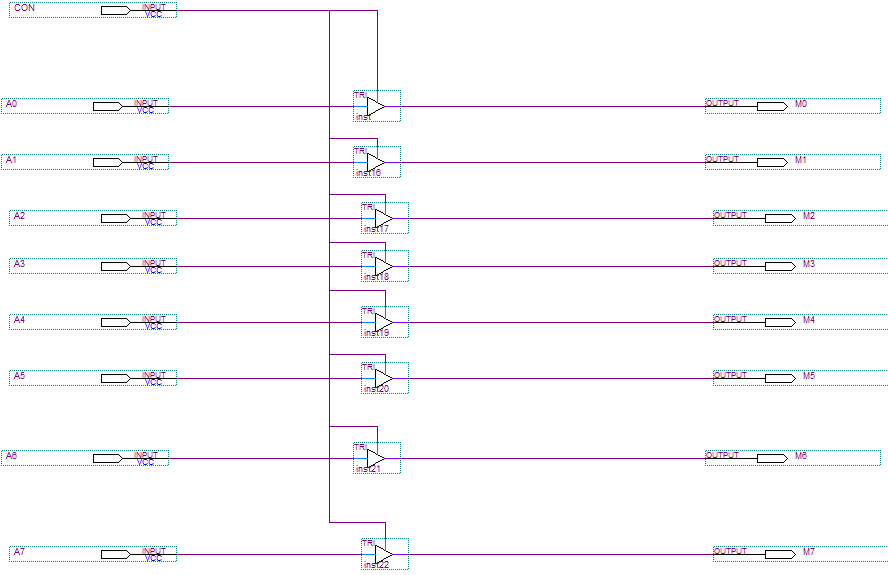
八位寄存器：

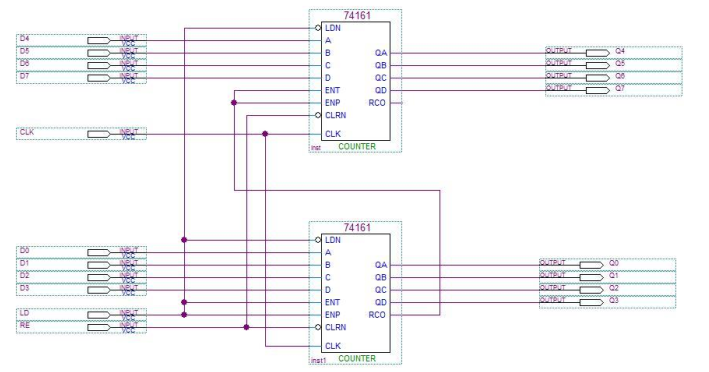


后续微地址形成部件：



三态门：

uPC：



总体电路：

PIN\_117

VCC

M

INPUT

PIN\_113

VCC

S0

INPUT

PIN\_114

VCC

S1

INPUT

PIN\_115

VCC

S2

INPUT

PIN\_116

VCC

S3

INPUT

PIN\_112

VCC

C0

INPUT

PIN\_92

VCC

UIR2

INPUT

PIN\_90

VCC

UIR1

INPUT

PIN\_89

VCC

UIR0

INPUT

PIN\_95

VCC

UIR4

INPUT

PIN\_96

VCC

UIR5

INPUT

PIN\_94

VCC

G

INPUT

PIN\_97

VCC

C

INPUT

PIN\_99

PIN\_132

VCC

DAN\_CLK

INPUT

PIN\_131

VCC

LIANXU\_CLK

INPUT

PIN\_77

VCC

K0

INPUT

PIN\_206

VCC

CLR

INPUT

PIN\_187

ROM4

OUTPUT

PIN\_188

ROM5

OUTPUT

PIN\_189

ROM6

OUTPUT

PIN\_191

ROM7

OUTPUT

PIN\_180

ROM0

OUTPUT

PIN\_181

ROM1

OUTPUT

PIN\_185

ROM3

OUTPUT

PIN\_182

ROM2

OUTPUT

PIN\_64

WR

OUTPUT

PIN\_63

RD

OUTPUT

PIN\_60

URD

OUTPUT

PIN\_61

CPUIR

OUTPUT

PIN\_168

ADDRRAM1

OUTPUT

PIN\_169

ADDRRAM2

OUTPUT

PIN\_170

ADDRRAM3

OUTPUT

PIN\_171

ADDRRAM4

OUTPUT

PIN\_173

ADDRRAM5

OUTPUT

PIN\_175

ADDRRAM6

OUTPUT

PIN\_176

ADDRRAM7

OUTPUT

PIN\_179

ADDRRAM8

OUTPUT

PIN\_201

VCC

RAMDATA7

BIDIR

PIN\_200

VCC

RAMDATA6

BIDIR

PIN\_199

VCC

RAMDATA5

BIDIR

PIN\_198

VCC

RAMDATA4

BIDIR

PIN\_197

VCC

RAMDATA3

BIDIR

PIN\_195

VCC

RAMDATA2

BIDIR

PIN\_193

VCC

RAMDATA1

BIDIR

PIN\_192

VCC

RAMDATA0

BIDIR

OCTAL D-FF

D2

D3

D1

D4

D7

D6

D5

D8

CLK

CLRN

Q4

Q1

Q2

Q3

Q6

Q5

Q7

Q8

74273

inst

OCTAL D-FF

D2

D3

D1

D4

D7

D6

D5

D8

CLK

CLRN

Q4

Q1

Q2

Q3

Q6

Q5

Q7

Q8

74273

inst1

VCC

OCTAL D-FF

D2

D3

D1

D4

D7

D6

D5

D8

CLK

CLRN

Q4

Q1

Q2

Q3

Q6

Q5

Q7

Q8

74273

inst7

A2

A1

A0

B0

B1

B2

B3

B4

B5

B6

B7

3-8

inst38

A2

A1

A0

B0

B1

B2

B3

B4

B5

B6

B7

3-8

inst40

AND2

inst8

AND2

inst9

AND2

inst10

AND2

inst23

AND2

inst24

A1

A0

B0

B1

B2

B3

2-4

inst26

A1

A0

B0

B1

B2

B3

2-4

inst29

NOT

inst32

NOT

inst33

A4

B4

A5

B5

A6

B6

A7

B7

M

S0

S1

S2

S3

A0

B0

A1

B1

A2

B2

A3

B3

C0

F4

F5

F6

F7

F0

F1

F2

F3

ALU

inst3

IR7

UIR15

IR6

UIR14

IR5

UIR13

IR4

UIR12

UIR11

UIR10

UIR9

UIR8

JP

QJP

Y7

Y6

Y5

Y4

Y3

Y2

Y1

Y0

ADDR

inst4

VCC

AND2

inst43

AND2

inst44

NOT

inst48

AND2

inst49

NOT

inst34

NOT

inst37

TRI

inst14

TRI

inst15

TRI

inst22

TRI

inst17

TRI

inst18

TRI

inst19

TRI

inst20

TRI

inst21

**CLRN**

**D**

**PRN**

**Q**

DFF

inst2

A0

A1

A2

A3

B0

B1

B2

B3

Y0

Y1

Y2

Y3

Y4

Y5

Y6

Y7

MUL

inst25

EN0

R07

R06

R05

R04

R03

R02

R01

R00

EN1

R17

R16

R15

R14

R13

R12

R11

R10

Z7

Z6

Z5

Z4

Z3

Z2

Z1

Z0

CHOOSE

inst27

NOT

inst28

LOAD

D4

D5

D6

D7

CLK

D0

D1

D2

D3

ET

EP

CLR

Q4

Q5

Q6

Q7

Q0

Q1

Q2

Q3

test1

uPC

OCTAL D-FF

D2

D3

D1

D4

D7

D6

D5

D8

CLK

CLRN

Q4

Q1

Q2

Q3

Q6

Q5

Q7

Q8

74273

MAR\_0\_7

OCTAL D-FF

D2

D3

D1

D4

D7

D6

D5

D8

CLK

CLRN

Q4

Q1

Q2

Q3

Q6

Q5

Q7

Q8

74273

IR

EN0

R07

R06

R05

R04

R03

R02

R01

R00

EN1

R17

R16

R15

R14

R13

R12

R11

R10

Z7

Z6

Z5

Z4

Z3

Z2

Z1

Z0

CHOOSE

choose\_A

EN0

R07

R06

R05

R04

R03

R02

R01

R00

EN1

R17

R16

R15

R14

R13

R12

R11

R10

Z7

Z6

Z5

Z4

Z3

Z2

Z1

Z0

CHOOSE

choose\_B

## 4、硬布线实现的模型机

ALU等部件与微程序相同，在此不再重复。

总体布局：



时序控制：



乘法运算器：



指令译码器：



启停电路：

