# 一、课程设计步骤

## 1、确定模型计算机功能及用途

完成一个较简单的计算机主机系统的设计，加深对微程序控制的计算机主机的基本构成，部件设计，部件间的连接，微程序的编制与调试等全过程的体验和认识。所设计的模型机应具备：在自行设计的模型机指令集基础上，运行简单用户程序，通过微程序实现对相应硬件的控制，实现模型机的特定功能，具体来说，应支持以下功能的用户程序实现：

1. 从内存中取两个数，相加后将结果存入第三个内存单元；
2. 在(1)的基础上，将相加过程中产生的“产生进位、溢出、结果为负、结果为0”状态存入状态寄存器（PSW）中;
3. 在以上的基础上，支持条件跳转功能，即“若状态寄存器具有某状态，则进行跳转”；
4. 实现乘法功能，采用累加方法，乘积16位；
5. 程序运行结束后停机。

为支持以上程序的运行，需要在控存中编写相应的微程序控制硬件，以支持指令集的功能。

## 2、指令系统

为实现相应功能，设计的指令系统有如下指令：

1. 取数指令LD Ri,AD，双操作数，微程序入口地址10H：

功能：将RAM中地址为AD的单元中数据存入寄存器Ri中，(AD)->Ri；

格式：双字长、双操作数。

第一个字节高四位为操作码1H，低两位为寄存器编号（00、01、10、11）。

第二个字节为AD，即数据来源地址（直接寻址）。

1. 存数指令ST Ri,AD，微程序入口地址20H：

功能：将Ri寄存器的内容存入RAM中地址为AD的内存单元，Ri->(AD)；

格式：双字长、双操作数。

第一个字节高四位为操作码2H，低两位为寄存器编号。

第二个字节为AD，即数据目标地址（直接寻址）。

1. 停机指令HALT，微程序入口地址30H：

功能：停机，此指令执行完毕后不再执行任何周期；

格式：单字长、无操作数，高四位为操作码3H。

1. 条件跳转指令JX A，微程序入口地址40H：

功能：分为JC、JV、JN、JZ，即“进位/溢出/为负/为零时跳转”，跳转意即将程序计数器PC的值加上A，即相对寻址。若条件不满足则PC+1->PC；

格式：双字长、双操作数。

第一个字节高四位为操作码4H，此后两位代表X：（00:C、01:V、10:N、11:Z）。

第二个字节为偏移量，用补码表示（立即数）。

1. 加法指令ADD Ri,Rj，微程序入口地址50H：

功能：将寄存器Ri、Rj的和存入寄存器Rj中，并将产生的状态打入PSW，(Ri)+(Rj)->Rj；

格式：单字长、双操作数。高四位为操作码5H，接下来两位代表i、最低两位代表j；

1. 无条件跳转指令JMP A，微程序入口地址60H：

功能：将PC的值加上A，即相对寻址；

格式：双字长、单操作数。首字节高四位为操作码6H，第二个字节为偏移量（立即数）。

1. 递增指令INC Ri，微程序入口地址70H：

功能：将寄存器Ri的值加1模256（按无符号数处理）；

格式：单字长、单操作数，高四位为操作码7H，低两位代表i。

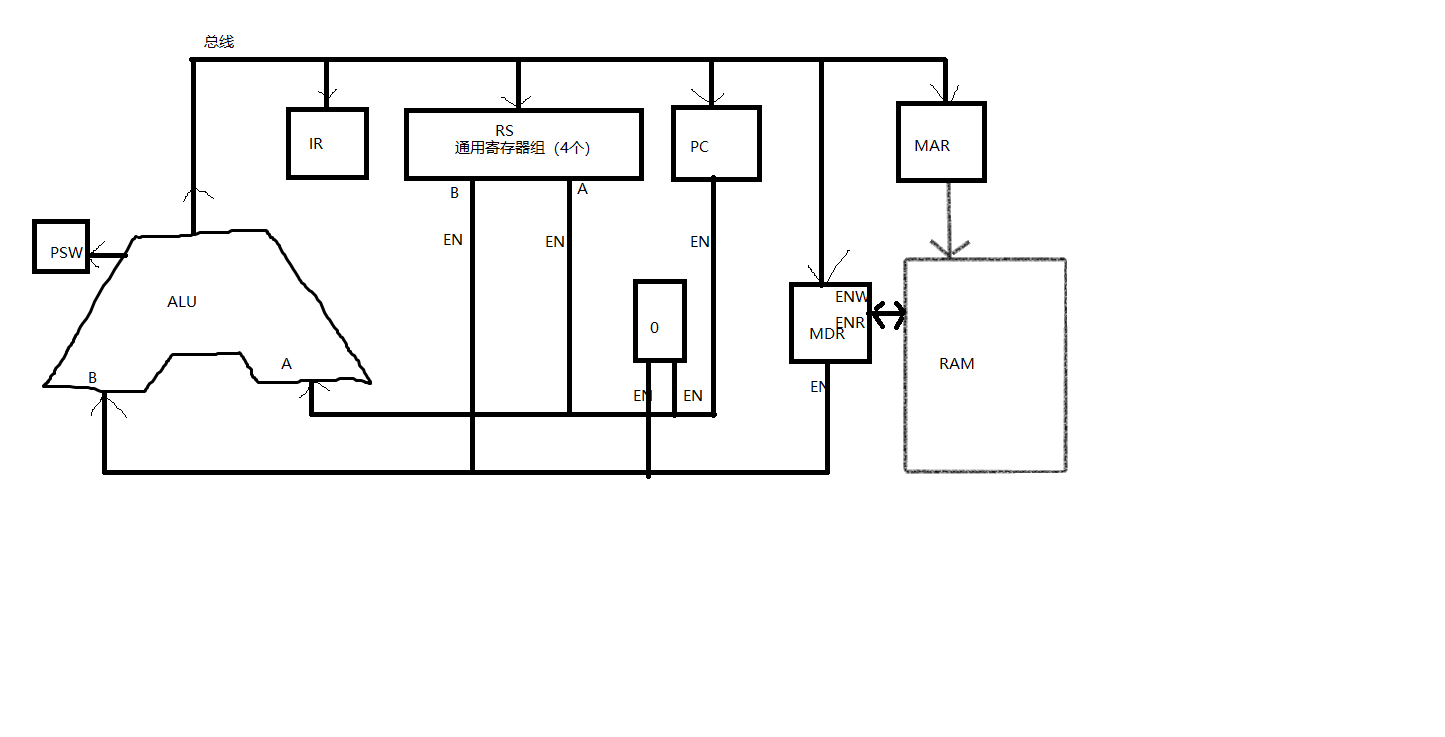
1. 带状态递增指令INC+ Ri，微程序入口地址A0H：

功能：同递增指令，但将在递增过程中产生的状态打入PSW；

格式：单字长、单操作数，高四位为操作码AH，低两位代表i。

## 3、总体结构与数据通路

总体结构图：



IR输出的与控存（ROM）中输出的控制信号、uPC、时钟信号等结构未画出。

该模型机的数据通路是以总线为基础，以ALU为核心构成的，寄存器间数据传输均需经过ALU，包括直送。每条数据传输指令执行时，数据通常由某寄存器使能输出到ALU参与运算（可能是直送）后，根据脉冲打入到对应寄存器中。

打入脉冲共6种。他们可以将总线上的数据打入到相应的寄存器。

分别为：CPPSW、CPIR、CPRS、CPPC、CPMDR、CPMAR。

其中CPPSW独立于其他五种，另外五种由单一控制信号决定，故互斥。

三态门输出使能信号共6种（扩展要求需要额外的一种，见扩展说明部分）。他们控制ALU的A、B两端的数据来源。

与A相连的有三种：ENRS(A)、EN0、ENPC，即A端数据来源；

与B相连的有三种：ENRS(B)、EN0、ENMDR，即B端数据来源。

此外还有MDR与RAM交互的双向三态门，有ENwrite与ENread两种使能信号，控制RAM的读写，实现了CPU与内存数据的隔离。