模型机设计指导书

湖南大学信息科学与工程学院
电路与电子课程教学组

一、设计目的

本课程力图以“培养学生现代数字系统设计能力”为目标，贯彻以 CPU 设计为核心，以层次化、模块化设计方法为抓手的组织思路，培养学生设计与实现数字系统的能力。

完整、连贯地运用课程所学知识，熟练掌握现代 EDA 工具基本使用方法，为后续课程学习和今后从事相关工作打下良好的基础或做下一些铺垫。

二、基本任务

1、按照给定的结构框架、数据格式和指令系统，使用 EDA 工具设计一台用硬连线逻辑控制的简易计算机；

2、要求灵活运用各方面知识，使得所设计的计算机具有较佳的性能；

3、对所设计计算机的性能指标进行分析，整理出设计报告。

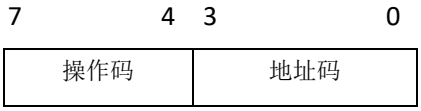
三、数据格式与指令系统

1、 数据格式

数据采用 8 位二进制补码表示，其中最高位（第 7 位）为符号位，其数值表示范围为：$-128 \leq X \leq +127$。

2、 指令格式

除立即数指令是双字节外，其它指令均为单字节，指令高 4 位为操作码，表示指令执行的操作，用以区分指令；低 4 位为地址码，指明操作数的来源。



3、 寻址方式

寻址方式是确定下一条要执行的指令地址和数据地址的方式，包含指令寻址和数据寻址。

1）指令寻址

指令寻址分为顺序寻址和跳转寻址。

顺序寻址是通过程序计数器 PC 加 1 自动形成下一条指令的地址。

跳转寻址是转移类指令执行时，将转移地址从寄存器 R3 取出，写入程序计数器 PC。

2）数据寻址

指令系统提供灵活的数据寻址方式，用尽量短的地址码提供操作数地址。本模型机共有 3 种数据寻址方式。

⑴ 立即寻址

操作数包含在指令中，跟在操作码后面，作为指令的一部分，因此也叫立即数。

⑵ 寄存器直接寻址

操作数存放在通用寄存器中。，指令地址码 2 位一组分别表示两个寄存器编号。此时指令格式如下图所示，Rs 和 Rd 分别表示两个操作数所在的寄存器编号，其中 Rs 是源寄存器编号，Rd 是目的寄存器编号。

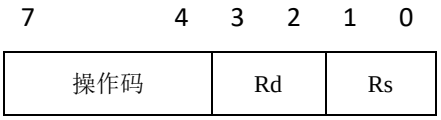
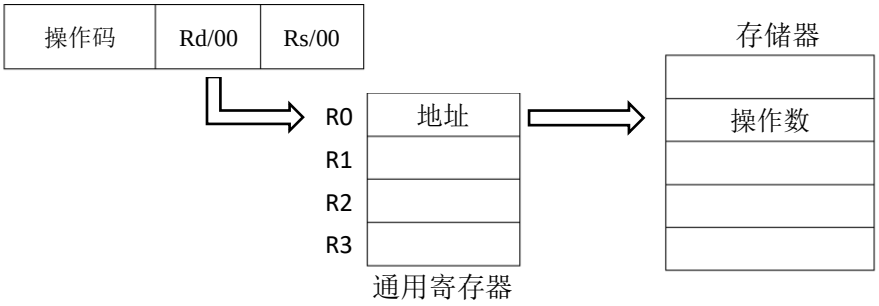


表 1 寄存器编号

Rs 或 Rd 的值	指定的寄存器
00	寄存器 R0
01	寄存器 R1
10	寄存器 R2
11	寄存器 R3

⑶ 寄存器间接寻址

操作数存放在存储器中，地址码中有一个寄存器编号为“00”，此时存放操作数的存储单元地址在寄存器 R0 中，通过访问寄存器 R0 获取存储单元地址，再从存储器对应单元读取的才是操作数，具体过程为：



4、 指令系统

指令系统有 12 条指令，具体格式见指令系统表。应该指出的是，各条指令的编码形式可以多种多样。为了叙述方便，下面采用汇编符号对指令进行描述，其中 Rs 和 Rd 分别表示“源”和“目的”寄存器，源和目的寄存器可以是通用寄存器中的任意一个寄存器，M 表示地址在寄存器 R0 中的存储单元。

表 2 指令系统表

汇编符号	功能	机器码	备注
MOVA Rd, Rs	(Rs) ⟶ Rd	0100 Rd Rs	
MOVB M, Rs	(Rs) ⟶ (R0)	0101 00 Rs	
MOVC Rd, M	((R0)) ⟶ Rd	0110 Rd 00	
MOVD R3, PC	(PC) ⟶ R3	0111 11 XX	

ADD Rd, Rs	(Rd)+(Rs) ⟶ Rd	1000 Rd Rs	
SUB Rd, Rs	(Rd)-(Rs) ⟶Rd IF(Rd>Rs), THEN G=1, ELSE G=0	1001 Rd Rs	
JMP	(R3) ⟶PC	1010 XX 11	
JG	IF G=1, THEN (R3) ⟶PC	1011 XX 11	
IN Rd	外设输入 ⟶ Rd	1100 Rd XX	
OUT Rs	(Rs) ⟶ 外设	1101 XX Rs	
MOVI IMM	立即数 IMM⟶ R0	1110 00 XX IMM	双字节
HALT	停机	1111 00 00	

四、模型机结构框架

计算机的工作过程可以看作是许多不同的数据流和控制流在各模块之间流动，数据在控制信号的控制下进行流动，控制信号决定了数据流动的时间和方向。数据流所经过的路径称作数据通路。数据通路不同，指令执行的操作过程就不同，模型机的结构也就不一样。本模型机的结构框架如图 1 所示：

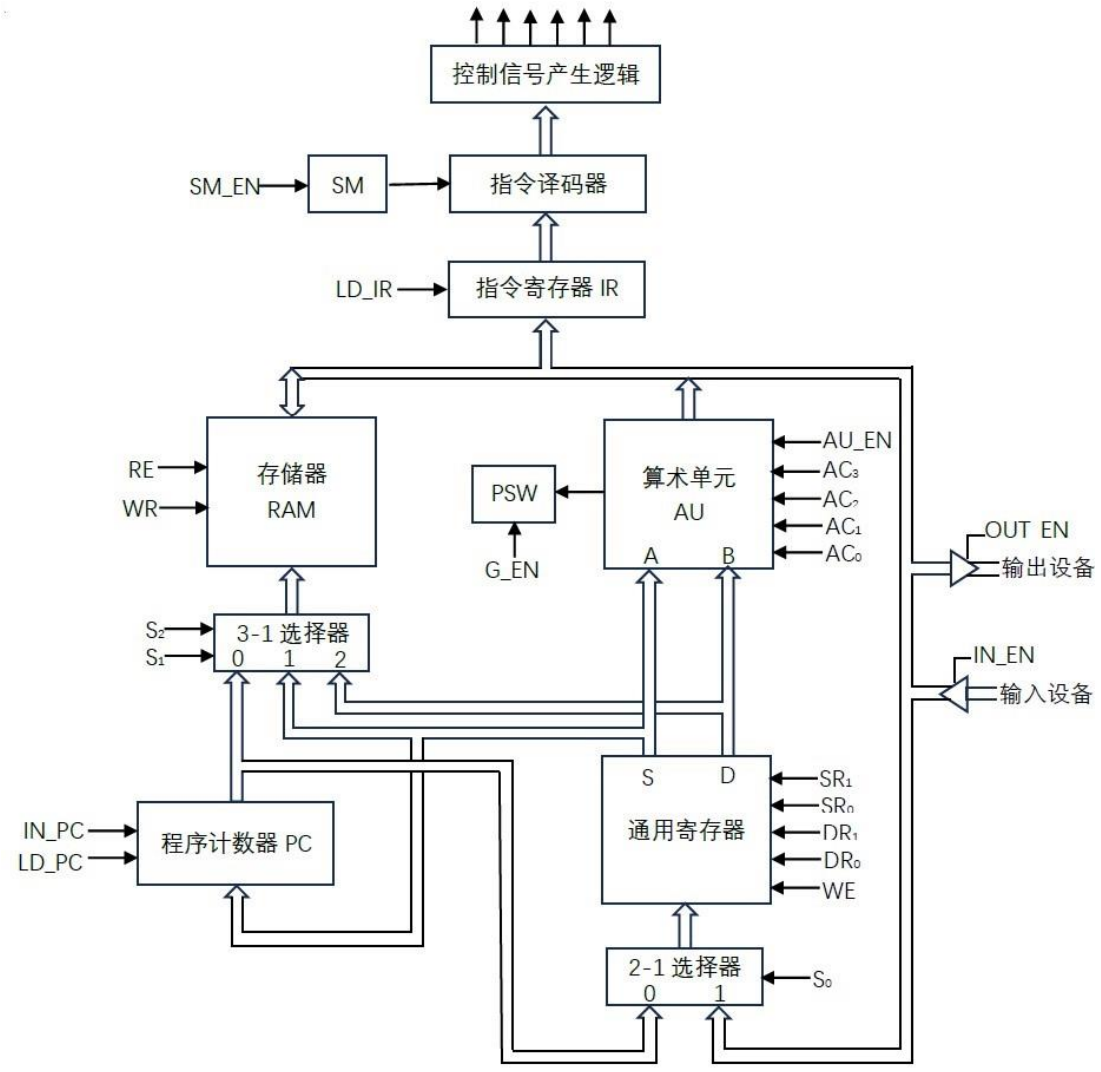


图 1　模型机结构框架

1、模块功能

存储器、指令寄存器、算术单元和通用寄存器之间互连的通道称为总线 BUS，它由 8 根导线组成，同时传输 8 位数据，实现模块间的数据交互。

程序计数器 PC：存放当前欲执行指令在 RAM 中的存储地址。LD_PC=1 时，PC 装载跳转地址，IN_PC=1 时，PC 进行自加 1 操作。

3-1 选择器：选择 RAM 的地址来源，其有 3 个输入，每个输入 8 位，分别接程序计数器 PC 和通用寄存器的 S 口和 D 口，由控制信号 S2、S1 选择其中一个传送至 RAM 的地址输入端。

存储器 RAM：存放指令和数据，其大小为 256×8 位，即有 256 个存储单元，每个单元存放 8 位数据，所以该 RAM 有一个 8 位的地址输入端以访问每个存储单元，一个双向的 8 位数据输入输出端。RAM 每个存储单元存放着一条指令或一字节数据。指令在 RAM 中是顺序存放的，如 0 号存储单元存放第一条指令，1 号存储单元存放第二条指令，依次存放。从存储器的哪一个单元读取指令，由程序计数器 PC 提供单元地址。指令顺序执行时，即指令从存储器中按顺序取出来，先取出第一条，再取第二条...，这时程序计数器 PC 自加 1 计数，指向顺序执行指令在存储器中的单元地址。当执行跳转指令时，程序计数器 PC 中原有的计数值失效，跳转指令的转移地址从寄存器 R3 送到程序计数器 PC 中，PC 从新的值开始继续计数。当从存储器中取出数据，注意是取数据，不是取指令，或者向存储器存放运算结果时，存储单元的地址由通用寄存器提供。

指令寄存器 IR：存放当前执行的指令。

SM：指示当前周期是取指周期还是执行周期。SM=0，表示当前是取指周期，SM=1，表示当前是执行周期。

指令译码器：解析指令。根据指令寄存器 IR 提供的 8 位指令编码，解析是哪条指令。

控制信号产生逻辑：根据指令译码器解析的指令，产生该指令执行所需的控制信号。

2-1 选择器：选择通用寄存器的数据来源。

通用寄存器：保存操作数和中间计算结果，其有 4 个 8 位寄存器 R0、R1、R2、R3，控制信号 SR1、SR0 选择 4 个寄存器中的一个作为源寄存器，源寄存器的数据从 S 口输出；DR1、DR0 选择 4 个寄存器中的一个作为目的寄存器，目的寄存器的数据从 D 口输出，WE 是写控制信号，在 DR1、DR0 的配合下，将其输入端的数据写入目的寄存器。

算术单元 AU：实现算术运算，并在执行 MOVA、MOVB、OUT 指令时负责将数据送至总线 BUS。

状态寄存器 PSW：存放 SUB 指令产生的状态位 G。

2、控制信号

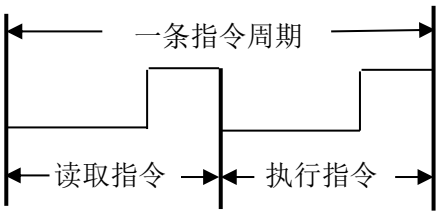
表 2　基本控制信号及功能表

序号	信号	功能
1	LD_PC	当 LD PC=1 时，将寄存器 R3 的内容写入 PC
2	IN_PC	当 IN PC=1 时，PC 进行自加 1 操作
3	S2、S1	选择 RAM 地址来源。00：程序计数器 PC，01：S 口，10：D 口
4	WE	当 WE=1 时，对存储器 RAM 进行写操作

5	RE	当 RE=1 时，对存储器 RAM 进行读操作
6	LD_IR	当 LD_IR=1 时，将 BUS 上的指令编码写入指令寄存器 IR
7	SM_EN	当 SM_EN=1 时，允许 SM 翻转，即 SM 由 0 变为 1 或由 1 变为 0
8	S0	选择通用寄存器的数据来源。0：程序计数器 PC，1：总线 BUS
9	SR1、SR0	选择寄存器 R0、R1、R2、R3 中的一个作为源寄存器
10	DR1、DR0	选择寄存器 R0、R1、R2、R3 中的一个作为目的寄存器
11	WE	当 WE=1 时，将输入端的数据写入通用寄存器
12	AU_EN	AU_EN=1，AU 进行算术运算或数据传输；AU_EN=0，输出高阻态
13	AC3～AC0	控制算术单元 AU 执行不同的操作
14	G_EN	当 G_EN=1 时，将 AU 产生的 G 写入状态寄存器 PSW
15	IN_EN	当 IN_EN=1 时，允许外部设备输入数据至总线 BUS
16	OUT_EN	当 OUT_EN=1 时，允许总线 BUS 上的数据输出至外部设备

五、指令的执行

指令执行与模型机结构、指令执行方式有关。指令可以串行执行，也可以并行执行。本设计采用串行工作方式，即“读取指令—执行指令—再读取指令—再执行指令……”。串行工作方式虽然工作速度和效率都要差一些，但它的控制简单。本机一条指令需要两个时钟周期完成，一个时钟周期读取指令，一个时钟周期执行指令。



读取指令的时间随所使用的 RAM 的性能而异。执行指令的时间依据控制流和数据流所经过的路径与各级门的最大延迟而定。本机中写入 RAM 和通用寄存器的操作显然不能发生在“执行阶段”的任意时刻，它必须是在运算结果已经产生，并被传送到总线的适当时刻才能“写”，这就需要时钟脉冲来控制时序。

1、读取指令的过程

要求完成的操作：从 RAM 中取出指令写入指令寄存器 IR，PC 自加 1，SM 变为 1。

具体过程为：程序计数器 PC 中的地址经 3-1 选择器送至 RAM 的地址输入端；在 RE 和地址的共同作用下，指令在时钟上升沿从 RAM 中读出送至总线 BUS；在 LD_IR 的控制下，BUS 上的指令在时钟下降沿写入指令寄存器 IR；同时程序计数器 PC 自加 1，指向下一条指令在 RAM 中的存放地址；SM 由 0 变为 1，指示下一周期为指令的执行周期。

2、指令的执行过程

每条指令的取指过程都相同，不同的是执行过程。接下将指令划分为数据传送类指令、算术运算类指令、转移类指令、输入输出类指令和停机指令，分别介绍各类指令的执行过程。每条指令执行完，SM 由 1 变为 0，指示下一周期为读取指令周期。

1）数据传送类指令的执行过程

MOVA　Rd，Rs

要求完成的操作：源寄存器 Rs 中的数据写入目的寄存器 Rd，即（Rs）→Rd。

执行过程：根据控制信号 SR1、SR0 选择源寄存器 Rs 的数据从通用寄存器 S 口输出，在 AC3～AC0 和 AU_EN 的控制下，经 AU 送入总线 BUS；S0 为 1，BUS 上的数据传送至通用寄存器的输入端；在 WE 和 DR1、DR0 的控制下，时钟下降沿将输入端的数据写入目的寄存器 Rd。

MOVB　M，RS

要求完成的操作：源寄存器 Rs 中的数据写入 RAM 的某个存储单元，该单元的地址存放在寄存器 R0 中，即（RS）→（R0）。

执行过程：控制信号 DR1、DR0 为 00（寄存器 R0 的编号），从通用寄存器 D 口输出 R0 中的内容，控制信号 S2、S1 为 10，R0 的内容通过 3-1 选择器到达存储器 RAM 的地址输入端；控制信号 SR1、SR0 选择源寄存器 Rs 的数据从通用寄存器 S 口输出，在 AC3～AC0 和 AU_EN 的控制下，经 AU 送入总线 BUS，在 WR 的控制下，时钟上升沿将 BUS 上的数据写入存储器 RAM。

MOVC　Rd，M

要求完成的操作：寄存器 R0 给出 RAM 的单元地址，读取该单元的数据写入目的寄存器 Rd，即（(R0)）→Rd。

执行过程：控制信号 SR1、SR0 为 00（寄存器 R0 的编号），从通用寄存器 S 口输出 R0 中的内容，控制信号 S2、S1 为 01，R0 的内容通过 3-1 选择器到达存储器 RAM 的地址输入端；在 RE 控制下，时钟上升沿从 RAM 中读取数据，送入总线 BUS；S0 为 1，BUS 上的数据传送至通用寄存器的输入端；在 WE 和 DR1、DR0 的控制下，时钟下降沿将输入端的数据写入目的寄存器 Rd。

MOVD　R3, PC

要求完成的操作：程序计数器 PC 中的内容写入寄存器 R3，即（PC）→R3。

执行过程：控制信号 S0 为 0，PC 中的内容传送至通用寄存器的输入端；WE 为 1，DR1、DR0 为 11（寄存器 R3 的编号），时钟下降沿将 PC 的内容写入寄存器 R3。

MOVI　IMM

这条指令是双字节指令，第一字节为指令码，第二字节为立即数。

要求完成的操作：将指令中的立即数写入寄存器 R0，即 IMM→R0。

执行过程：控制信号 S2、S1 为 00，程序计数器 PC 中的地址通过 3-1 选择器传至存储器 RAM 的地址输入端，在 RE 控制下，时钟上升沿将立即数从 RAM 中读出并送入总线 BUS；S0 为 1，BUS 上的数据传送至通用寄存器的输入端；WE 为 1，DR1、DR0 为 00（寄存器 R0 的编号），时钟下降沿将数据写入寄存器 R0。在 IN_PC 控制下，时钟下降沿 PC 加 1 计数，指向下一条指令在 RAM 中的存放地址。

2）算术运算类指令的执行过程

ADD　Rd，Rs

要求完成的操作：源寄存器 Rs 和目的寄存器 Rd 的数据相加，和写入目的寄存器 Rd，

即（Rs）+（Rd）→Rd。

执行过程：控制信号 SR1、SR0 选择源寄存器 Rs 的数据从 S 口输出，控制信号 DR1、DR0 选择目的寄存器 Rd 的数据从 D 口输出；在 AC3～AC0 和 AU_EN 的控制下，在 AU 中进行加法运算后将相加的和送入总线 BUS；S0 为 1，BUS 上的数据传送至通用寄存器的输入端；在 WE 和 DR1、DR0 的控制下，时钟下降沿将输入端的数据写入目的寄存器 Rd。

SUB　Rd，Rs

要求完成的操作：寄存器 Rd 的数据减去 Rs 的数据，差写入寄存器 Rd，即（Rd）-（Rs）→Rd。

执行过程：控制信号 SR1、SR0 选择源寄存器 Rs 的数据从 S 口输出，控制信号 DR1、DR0 选择目的寄存器 Rd 的数据从 D 口输出；在 AC3～AC0 和 AU_EN 的控制下，在 AU 中进行减法运算后将相减的差送入总线 BUS；S0 为 1，BUS 上的数据传送至通用寄存器的输入端；在 WE 和 DR1、DR0 的控制下，时钟下降沿将输入端的数据写入目的寄存器 Rd。SUB 指令影响状态位 G，如果 Rd>Rs，则 G=1，否则 G=0。

3）转移类指令的执行过程

JMP

要求完成的操作：寄存器 R3 的内容写入程序计数器 PC，即（R3）→PC。

执行过程：控制信号 SR1、SR0 为 11（寄存器 R3 的编号），从通用寄存器 S 口输出 R3 中的内容，在 LD_PC 的控制下，时钟下降沿将 R3 的内容写入程序计数器 PC。

JG

要求完成的操作：仅当 G=1，将寄存器 R3 的内容写入程序计数器 PC，否则 PC 的内容保持不变，即 IF(Rd>Rs), THEN G=1, ELSE G=0。

执行过程：控制信号 SR1、SR0 为 11（寄存器 R3 的编号），从通用寄存器 S 口输出 R3 中的内容，如果条件满足（即 G=1），在 LD_PC 控制下，时钟下降沿将 R3 的内容写入程序计数器 PC，否则 PC 的内容保持不变。

4）输入输出指令的执行过程

IN　Rd

要求完成的操作：将外设输入的数据写入寄存器 Rd。

执行过程：在 IN_EN 控制下，外设输入的数据送至总线 BUS；控制信号 S0 为 1，BUS 上的数据传送至通用寄存器的输入端；在 WE 和 DR1、DR0 的控制下，时钟下降沿将输入端的数据写入目的寄存器 Rd。

OUT　Rs

要求完成的操作：寄存器 Rs 中的数据输出至外部设备。

执行过程：控制信号 SR1、SR0 选择源寄存器 Rs 的数据从 S 口输出，在 AC3～AC0 和 AU_EN 的控制下，经 AU 送入总线 BUS，在 OUT_EN 控制下将 BUS 上的数据输出至外部设备。

5）停机指令的执行过程

HALT

停机指令，执行完这条指令，模型机进入停机状态，指令 HALT 后面即使还有其他指令，模型机也不再执行。