《数字逻辑》综合设计指导书

湖南大学信息科学与工程学院
数字逻辑课程教学组

一、设计目的

连贯运用《数字逻辑》所学到的知识，熟练掌握 EDA 工具的使用方法，为学习好后续《计算机原理》课程做铺垫。

二、设计任务

① 按给定的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系统；

② 要求灵活运用各方面知识，使得所设计的计算机系统具有较佳的性能；

③ 对所做设计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

本设计的主要目的是希望学生巩固在《数字逻辑》课程中学到的理论知识，并加以灵活运用。因此，要设计的计算机系统非常简单。这个系统具有寄存器直接寻址和寄存器间接寻址两种寻址方式，除跳转指令为双字节指令外，其它指令均为单字节指令，字长为 8 位。

1、 数据格式

数据字采用 8 位二进制定点补码表示，其中最高位（第 7 位）为符号位，小数点可视为最左或最右，其数值表示范围分别为：$-1 \le X < 1$ 或 $-128 \le X < 127$。

2、 寻址方式

指令的高 4 位为操作码，低 4 位分别用 2 位表示目的寄存器和源寄存器的编号，或表示寻址方式。本机有 2 种寻址方式。

⑴ 寄存器直接寻址

操 作 码	R1	R2

当 R1 和 R2 均不是“11”时，R1 和 R2 分别表示两个操作数所在寄存器的地址（寄存器编号），其中 R1 为目标寄存器地址，R2 为源寄存器地址。

R1 或 R2 的值	指定的寄存器
00	A 寄存器
01	B 寄存器
10	C 寄存器

⑵ 寄存器间接寻址

操 作 码	R1/11	R2/11

当 R1 或 R2 中有一个为“11”时，表示相应操作数的地址在 C 寄存器中。

3、 指令系统

该机给定的指令系统有 16 条指令，指令格式见指令系统表。应该指出的是，各条指令的编码形式可以多种多样。为叙述方便，下面采用汇编符号对指令进行描述，其中 R1 和 R2 分别表示“目标”和“源”两个寄存器，M 表示地址在寄存器 C 中的存贮单元。

表 1　　指 令 系 统 表

指令的汇编符号	指令的功能	指令的二进制编码
MOV　R1，R2	（R2）→ R1	1111　R1 R2
MOV　M，R2	（R2）→（C）	1111　11 R2
MOV　R1，M	((C)) →R1	1111　R1 11
ADD　R1，R2	（R1）+（R2）→ R1	1001　R1 R2
SUB　R1，R2	（R1）-（R2）→ R1	0110　R1 R2
AND　R1，R2	（R1）∧（R2）→ R1	1110　R1 R2
NOT　R1	/（R1）→ R1	0101　R1 XX
SHR　R1	（R1）逻辑右移一位→ R1	1010　R1 00
SHL　R1	（R1）逻辑左移一位→ R1	1010　R1 11
JMP　add	add → PC	0001　00 00，address
JZ　add	结果为 0 时 add → PC	0001　00 01，address
JC　add	结果有进位时 add → PC	0001　00 10，address
IN　R1	（开关 7-0）→ R1	0010　R1 XX
OUT　R1	（R1）→ 发光二极管 7-0	0100　R1 XX
NOP	（PC）+1 → PC	0111　00 00
HALT	停机	1000　00 00

四、数据通路及其说明

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机器各部分之间的流动，数据流所经过的路程称作机器的数据通路。数据通路不同，指令执行所经过的操作过程就不同，机器的结构也就不一样。如何设计一个好的数据通路已经超出了本课程的范围，在此我们不予讨论。我们假设所设计的计算机的数据通路如图 1 所示。

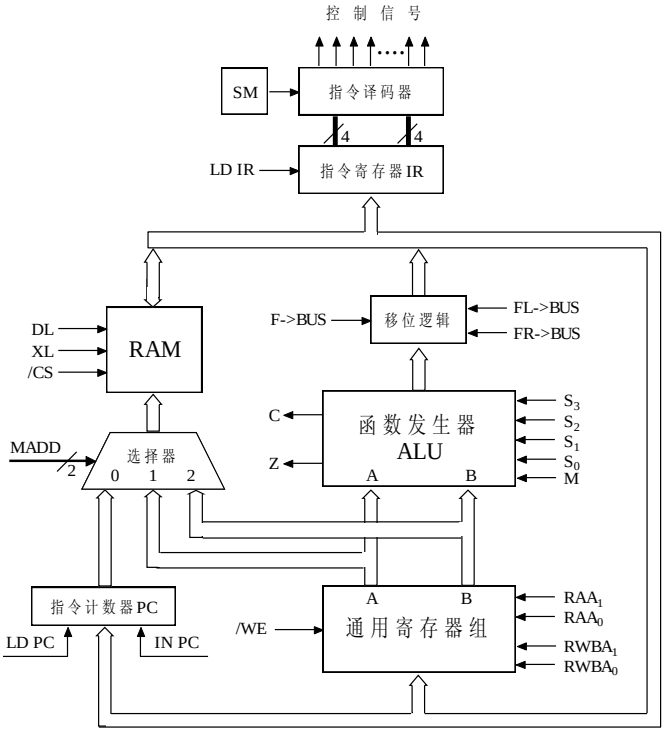


图 1　模型机数据通路

1、 数据传送类指令的执行过程

寄存器之间的传送

MOV　　R1，R2

要求完成的操作为（R2）→R1，执行过程为：

由 R2 的编码通过 RAA1、RAA0 从通用寄存器组 A 口读出 R2 的内容，在 S3～S0 和 M 的控制下，经 ALU 送入总线 BUS；由/WE 控制和 R1 的编码选择 RWBA1、RWBA0，将 BUS 上的数据写入通用寄存器 R1。

寄存器到内存的传送

MOV　　M，R2

要求完成的操作为（R2）→（C），执行过程为：

由 M 的编码 11 通过 RWBA1、RWBA0 从通用寄存器 B 口读出 C 寄存器中的地址，在 MADD=2 的控制下，地址通过选择器到达存储器 RAM 的地址输入端；由 R2 的编码通过 RAA1、RAA0 从通用寄存器组 A 口读出 R2 的内容，在 S3～S0 和 M 的控制下，经 ALU 送入总线 BUS，并在/CS 和 XL 控制下将 BUS 上的数据写入存储器 RAM。

内存到寄存器的传送

MOV　　R1，M

要求完成的操作为（(C)）→R1，执行过程为：

由 M 的编码 11 通过 RAA1、RAA0 从通用寄存器 A 口读出 C 寄存器中的地址，在 MADD=1 的控制下，地址通过选择器到达存储器 RAM 的地址输入端，/CS 和 DL 使数据出现在 BUS 上；由/WE 控制和 R1 的编码选择 RWBA1、RWBA0，将 BUS 上的数据写入通用寄存器 R1。

2、 算术逻辑运算类指令的执行过程

ADD　　R1，R2

SUB　　R1，R2

AND　　R1，R2

这类指令的执行过程为：

由 R2 的编码通过 RAA1、RAA0 从通用寄存器组 A 口读出 R2 的内容，由 R1 的编码通过 RWBA1、RWBA0 从通用寄存器组 B 口读出 R1 的内容，在 S3～S0 和 M 的控制下，经 ALU 送入总线 BUS；由/WE 控制和 R1 的编码选择 RWBA1、RWBA0，将 BUS 上的数据写入通用寄存器 R1。其中 ADD 和 SUB 指令影响状态位 C_f 和 Z_f。

3、 移位指令的执行过程

SHR　　R1

SHL　　R1

这类指令的执行过程为：

由 R1 的编码通过 RWBA1、RWBA0 从通用寄存器组 B 口读出 R1 的内容，在 S3～S0 和 M 的控制下通过 ALU，经移位逻辑右移或左移后送入总线 BUS；再由/WE 控制和 R1 的编码选择 RWBA1、RWBA0，将 BUS 上的数据写入通用寄存器 R1。

但是，标准的 ALU 模块没有移位功能，需要在该模块出口与总线接口处增加一部分电路以实现相应的移位功能（此问题请同学们自行解决）。

4、 转移类指令的执行过程

JMP　　add

JZ　　add

JC　　add

这类指令为双字节指令，第一字节为指令码，第二字节为转移目标地址。这类指令的执行过程为：

在 MADD=0 的控制下，程序计数器 PC 中的地址通过选择器到达存储器 RAM 的地址输入端，在/CS 和 DL 控制下转移地址从 RAM 中读出并送入 BUS；如果条件满足（IN PC=0）则在 LD PC 允许下将 BUS 上的地址打 PC，否则 PC 加 1 计数。

当数据通路设计好之后，就要进行详细电路设计，这时需要考虑其它各种因素，比如进行触发器 C_f 和 Z_f 的设置。

五、控制器设计

有了指令系统和数据通路之后，就可以进入控制器设计阶段。控制器设计有两种方法，一种是组合逻辑实现方法，另一种是微程序实现方法。我们采用第一种方法。

1、微控制信号

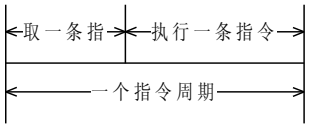
指令寄存器 IR 接收到一条机器指令后，这条指令就被译码执行。指令通过译码产生出的各种控制信号在时钟信号的配合下控制着指令执行的全过程。为此，需要将执行每条指令所需的全部基本微操作的控制信号罗列出来，进行综合分析、化简，并落实到不同的周期、节拍之中，然后用各种逻辑门电路实现。以下是所用基本控制信号列表。

表 2　基本控制信号及功能表

序号	信号	功能
1	IN PC	与 LD PC 配合使用，为 1 时 PC 加 1 计数，为 0 时加载 BUS 上的数据。
2	LD PC	当 IN PC=1 允许对 PC 加 1 计数，否则允许把 BUS 上的数据打入 PC。
3	LD IR	允许把 BUS 上的数据打入指令寄存器 IR。
4	/WE	允许把 BUS 上的数据打入通用寄存器组，低电平有效。
5	F→BUS	ALU 的运算结果通过移位门直接送到总线 BUS 的对应位。
6	FL→BUS	ALU 的运算结果通过移位门左移一位送到总线 BUS，且 F7 送 C_f。
7	FR→BUS	ALU 的运算结果通过移位门右移一位送到总线 BUS，且 F0 送 C_f。
8	/CS	允许访问存储器，低电平有效。
9～10	MADD	存储器 RAM 地址来源。0：指令计数器，1：通用寄存器 A 口，2：B 口。
11	DL	读存储器 RAM。
12	XL	写存储器 RAM。
13	M	M=1，表示 ALU 进行逻辑运算操作。
14～17	S_3～S_0	使 ALU 执行各种运算的控制位。
18	HALT	此位为“1”时停机，下次输入人工操作。

2、指令周期与工作脉冲设置

指令同期与数据通路结构、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—执行—再读取—再执行……”。串行工作方式虽然工作速度和主机效率都要差一些，但它的控制简单。因此，本机指令周期可以确定为：



读取指令的时间随所使用的 RAM 的性能而异。执行一条指令所需工作脉冲的个数与宽度要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。例如，本机中写入 RAM 和寄存器组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”，这就需要工作脉冲来控制时序。

六、控制台与控制指令设计

有了数据通路和控制器，还要有输入/输出部分，我们才能将编好的程序送入系统，看到运算结果。为简单起见，我们仅考虑最简单的 I/O 部分。虽然是最简单的，但仍必须有以下几个功能：

① 能够启动系统运行；
② 能够输入地址和程序；
③ 能够检查内存的内容；
④ 能够终止系统的运行；
⑤ 能够对系统进行初始化；
⑥ 能够提供某些出错信息，如溢出等。

1、 命令键设置及实现

为了实现控制台的功能，我们可以设置以下几个命令键：

⑴ START 键：启动系统运行，起始地址由 PC 当前内容确定；
⑵ STOP 键：终止系统运行；
⑶ INA 键：输入地址，将开关上设置的数作为地址送入 PC 中；
⑷ INP 键：输入程序，将开关上设置的数作为机器指令代码送入当前 PC 所指定的内存单元中；
⑸ CHP 键：检查程序，将当前 PC 所指定的内存单元的内容送显示器显示；
⑹ RESTART 键：系统总复位命令。

读者还可以设置一些其它命令键。在上面 6 个键中，INA、INP、CHP 不仅要启动系统运行，还需要在命令执行完后停机。要注意，这样的命令没有自己的指令格式，需要另外设计编码电路对这些命令进行编码，并将其送入指令寄存器 IR。

2、 输入输出部件

输入输出可以采用二进制、十进制或 ASCII 码。设备有多种多样，如 LCD 显示器及键盘、打印机等。为简单起见，本机采用最简单的二进制开关作为输入部件，发光二极管作为输出部件。发光二极管所显示的内容要由寄存器保存。