

Homework 10 — June 5

Lecturer: Ke Zhang

Completed by: 吉骏雄

课后习题 10.2, 10.7, 10.8, 10.15, 10.21

10.2 写出完成下列指令的微操作及节拍安排 (包括取指操作)。

(1) 指令 “ADD R₁, X” 完成将 R₁ 寄存器的内容和主存 X 单元的内容相加结果存于 R₁ 的操作。

(2) 指令 “ISZ X” 完成将主存 X 单元的内容增 1, 并根据其结果若为 0, 则跳过下一条指令执行。

解

(1) 指令 “ADD R₁, X” 的微操作及节拍安排如表 10.1所示。

表 10.1. 10.2 (1)

时钟周期	微操作
T_0	$PC \rightarrow MAR, 1 \rightarrow R$
T_1	$M(MAR) \rightarrow MDR, PC + 1 \rightarrow PC$
T_2	$MDR \rightarrow IR, OP(IR) \rightarrow ID$
T_0	$X \rightarrow MAR, 1 \rightarrow R$
T_1	$M(MAR) \rightarrow MDR$
T_2	$(R_1) + (MDR) \rightarrow R_1$

(2) 指令 “ISZ X” 的微操作及节拍安排如表 10.2所示。

表 10.2. 10.2 (2)

时钟周期	微操作
T_0	$PC \rightarrow MAR, 1 \rightarrow R$
T_1	$M(MAR) \rightarrow MDR, PC + 1 \rightarrow PC$
T_2	$MDR \rightarrow IR, OP(IR) \rightarrow ID$
T_0	$X \rightarrow MAR, 1 \rightarrow R$
T_1	$M(MAR) \rightarrow MDR$
T_2	$MDR \rightarrow C$
T_0	$C + 1 \rightarrow ALU$
T_1	$ALU \rightarrow MDR, 1 \rightarrow W$
T_2	$MDR \rightarrow M(MAR), Z \cdot (PC + 1) + \bar{Z} \cdot PC \rightarrow PC$

10.7 画出组合逻辑控制单元的组成框图, 根据指令处理过程, 结合有关部件说明其工作原理.

解 如图 10.1所示.

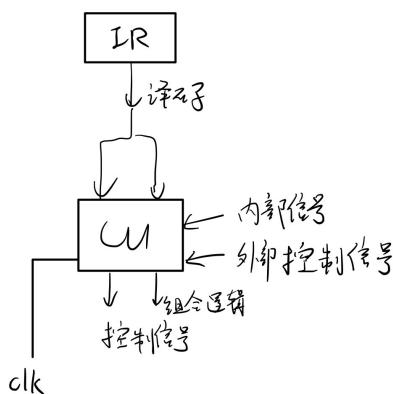


图 10.1. 10.7 题图

指令处理过程分为四个阶段, 分别为取指周期, 间址周期, 执行周期和中断周期, 并且间址周期可能不使用, 中断周期也可能并不进入. 取指阶段的工作如下:

1. 经过 CU 输出的控制信号的控制, 从主存中取出指令, 存入指令寄存器 IR 中.
2. 从 IR 中取出指令, 经过指令译码器的译码与 CU 的逻辑电路, 产生相应的控制信号.
3. 控制信号送到控制单元 (比如控制某线路/总线上的信号是否可通), 控制单元根据控制信号控制各部件的工作.

间址阶段的工作如下:

1. 从 IR 中取出指令 (中的一部分, 用来作为间址地址), 经过指令译码器的译码与 CU 的逻辑电路, 产生相应的控制信号.
2. 控制信号送到控制单元 (比如控制某线路/总线上的信号是否可通), 控制单元根据控制信号控制各部件的工作, 在这里是访存获得数据, 然后存入相应的寄存器 (比如覆写 IR 的地址部分).

执行阶段的工作为: 根据指令在 CU 中产生的信号, 控制线路进行数据传输, 比如从寄存器中取出数据, 存入寄存器中, 或者从寄存器中取出数据, 送到 ALU 中进行运算, 然后将结果存入寄存器中, 也可能是和内存进行数据传输.

中断阶段的工作为: 根据中断信号决定是否进入中断周期, 若进入, 则从 PC 中取出下一条指令的地址, 存入相应的寄存器中, 然后跳转到中断处理程序中.

10.8 画出微程序控制单元的组成框图, 根据指令处理过程, 结合有关部件说明其工作原理.

解 如图 10.2 所示. 这里是简化之后的版本, 省略了 CMAR 这个寄存器.

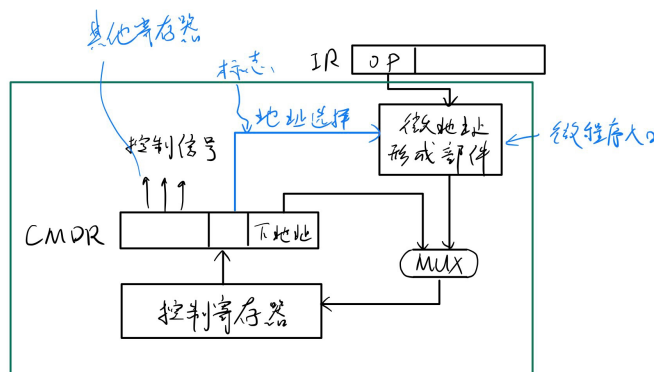


图 10.2. 10.8 题图

指令处理过程分为四个阶段, 分别为取指周期, 间址周期, 执行周期和中断周期, 并且间址周期可能不使用, 中断周期也可能并不进入. 取指阶段的工作如下:

1. 读取微指令存储器中的微指令, 存入控制寄存器中, 然后微指令寄存器中取出微指令发送到 CMDR.
2. 微指令解码后, 发送控制信号给逻辑电路, 产生相应的控制信号. 控制信号送到控制单元外 (比如控制某线路/总线上的信号是否可通), 各部件根据控制信号工作, 实现取出 PC 送到 IR 的操作.
3. 以上和以下操作, 都是依靠重复地选择进入下一条微指令的地址, 存入微指令地址寄存器中, 然后执行微指令来实现的.

间址阶段的工作如下: 取微指令, 根据微程序中得到的控制信号, 从 IR 中取出指令 (中的一部分, 用来作为间址地址), 经过访存获得数据, 然后覆写 IR 的地址部分.

执行阶段和中断阶段的工作和上面的基本相同.

10.15 设控制存储器的容量为 512×48 位, 微程序可在整个控存空间实现转移, 而控制微程序转移的条件共有 4 个 (采用直接控制), 微指令格式如图 10.3. 试问微指令中的 3 个字段分别为多少位?

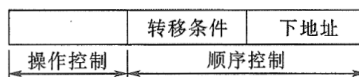


图 10.3. 10.15 题图

解 按照水平型 (水平微程序) 的微指令格式, 由于控制存储器的容量为 512×48 位, 下地址需要 9 位来存储 ($2^9 = 512$); 由于转移的条件共有 4 个, 因此转移条件需要 4 位 (每一位代表一个转移条件); 剩余的 35 位均可留给操作控制段.

10.21 表 10.3 给出 8 条微指令 $I_1 I_8$ 及所包含的微命令控制信号, 设计微指令操作控制字段格式, 要求所使用的控制位最少, 而且保持微指令本身内在的并行性.

表 10.3. 题 10.21 表

微指令	所含的微指令
I_1	a b c d e
I_2	a d f g
I_3	b h
I_4	c
I_5	c e g i
I_6	a h j
I_7	c d h
I_8	a b h

解 先将微指令与对应的微命令绘制成表格, 并挖掘微命令的并行性, 将之重叠, 如表 10.4 所示.

表 10.4. 微操作与微命令并行对应

微操作码	所含的微指令									
	a	bgj	c	d	e	fhi	g	h	i	j
I_1	1	1	1	1	1					
I_2	1	1		1		1	1			
I_3		1				1		1		
I_4			1							
I_5		1	1		1	1	1		1	
I_6	1	1				1		1		1
I_7			1	1		1		1		
I_8	1	1				1		1		

这样, 我们可以用 $1 \cdot 4 + 2 \cdot 2 = 8$ 位作为控制位, 分别对应 a bgj c d e fhi. 这六段的表示如表 10.5所示.

表 10.5. 微操作控制位与微命令的对应

编码	0	1	2	3
a	无操作	微命令 a		
bgj	无操作	微命令 b	微命令 g	微命令 j
c	无操作	微命令 c		
d	无操作	微命令 d		
e	无操作	微命令 e		
fhi	无操作	微命令 f	微命令 h	微命令 i

如果用直接控制阀需要 10 位, 而这里只需要 8 位, 因此可以减少 2 位.