

计组复习知识点

2019年1月7日 23:18

第一章 概论

冯诺依曼的核心思想：

- 1.计算机应由5部分组成：运算器、控制器、存储器、输入设备和输出设备
- 2.计算机中采用二进制形式表示信息（数据、指令）
- 3.采用存储程序的工作方式，这也是冯诺依曼体系最为核心的思想

P9

第一台计算机的相关信息：

世界上第一台现代电子计算机埃尼阿克(ENIAC：Electronic Numerical Integrator and Calculator，电子数字积分器与计算器)，诞生于1946年2月14日的美国宾夕法尼亚大学，并于次日正式对外公布，它标志着电子计算机时代的到来。

ENIAC长30.48米，宽6米，高2.4米，占地面积约170平方米，30个操作台，重达30英吨，耗电量150千瓦，造价48万美元。它包含了17,468根真空管（电子管）7,200根晶体二极管，1,500个中转，70,000个电阻器，10,000个电容器，1500个继电器，6000多个开关，计算速度是每秒5000次加法或400次乘法，是使用继电器运转的机电式计算机的1000倍、手工计算的20万倍。

P9

计算机的主要性能指标：

- 1.基本字长
- 2.外频
- 3.CPU的运算速度
- 4.数据通路宽度与数据传输率
- 5.存储容量
- 6.外围设备配置
- 7.软件配置

P24

第二章 数据的表示、运算与校验

8位二进制有符号的表示范围：

-128~127

机器数：

在计算机内部使用，连同数符一同数字化了的数称为机器数

P35

0的表示在原、反、补中的不同：

原：-0、+0

[+0]原=00000000

[-0]原 =10000000

反：

[+0]反=00000000

[-0]反 =11111111

补：

[+0]补=00000000

[-0]补 =00000000

补码中10000000表示-128

补码没有正0与负0之分。正数的反码、补码和其原码相同；负数的反码是其原码除符号位以外其他位取反，补码是其取反码后加1。

P36

补码的计算：

秒杀：从右往左数，第一个1到符号位之间的数字全部变反，其余

正数的补码和原码相同

变补：

所谓变补，就是在补码的基础上求其原码相反数的补码

计算方法：从右往左数，第一个1以左的数字全部变反（包括符号位）

P48

双符号位的作用：

答：双符号位能容易检查加、减运算中的溢出情况。当符号位相同，数值结果正确，00（结果为正，无溢出）、11（结果为负，无溢出）；当符号位为01（结果正溢）或10（结果负溢）时表示数值溢出。单符号位的信息量只能表示两种可能：数为正或负，如果发生溢出，就会使符号位的含义产生混乱。将符号位扩充为两位，就能判别是否有溢出以及正确的结果符号。（p51）

正溢、负溢时双符号位是什么：

正溢：01

负溢：10

P50

加减法运算是哪个码强（原、反、补？）：

补码

P48

乘除法运算哪个码强（原、反、补？）：

原码

P53

校验方法有几种：

奇偶校验、海明校验、循环冗余校验

奇偶校验码距、码字：

任何一种编码都有码字组成，两码字之间最小二进制变化位数称为码距

奇偶校验码的码距为2

P68

内存里采用最常见的奇偶校验方式

第三章 CPU子系统

CPU设计步骤有哪几个方面：

- 1.拟定指令系统
- 2.确定总体结构
- 3.安排时序
- 4.拟定指令流程和微命令序列

5.形成控制逻辑

P116

ALU的全称、CPU组成、大循环图：

ALU全称：

算术逻辑运算部件(Arithmetic Logic Unit)

P122

CPU基本部件：

运算部件、寄存器组、微命令产生部件、时序系统、CPU内部的数据通路结构

P75

大循环图：

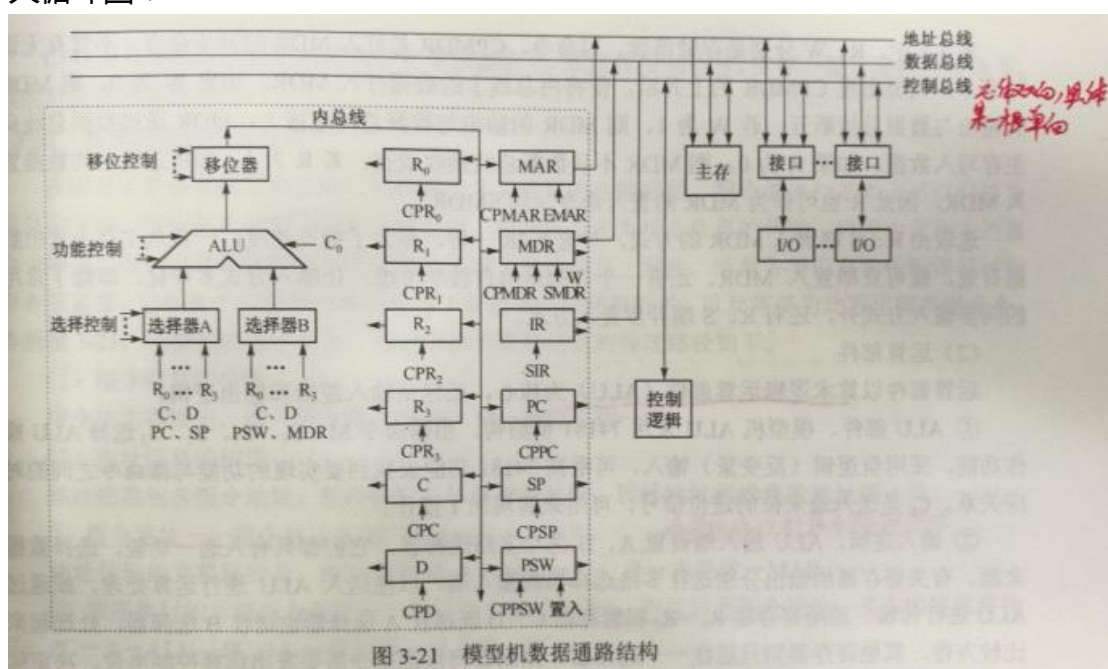


图 3-21 模型机数据通路结构

暂存器C、D：

模型机的特别安排，从主存中读取源操作数地址或源操作数时，就是用暂存器C；

从主存中读取源操作数地址或源操作数时，以及需要暂存目的地址或运算结果时，就是用暂存器D。

指令寄存器IR(Instruction Register)：

为提高读取指令的速度，将指令从主存中读出以后，经数据总线直接将其置入IR。

与主存的接口寄存器MAR(Memory Address Register)、MDR(Memory Data Register)：

CPU访问主存的地址由地址寄存器MAR提供；

数据寄存器MDR既可以与CPU内的部件交换数据，也可以与系统总线交换数据。

PC(Program Counter) :

程序计数器

SP(Stack Pointer):

堆栈指针

PSW(Program Status Words):

程序状态字

可编程访问的寄存器：R1-3、SP、PSW、PC

P121

CPU对外的3个窗口：

MAR , MDR , IR (主要是前两者)

P121

算术运算器核心是什么：

以算术逻辑运算部件 (ALU) 为核心，还包括输入逻辑和输出逻辑

P124

逻辑组合、微程序控制器定义、优缺点：

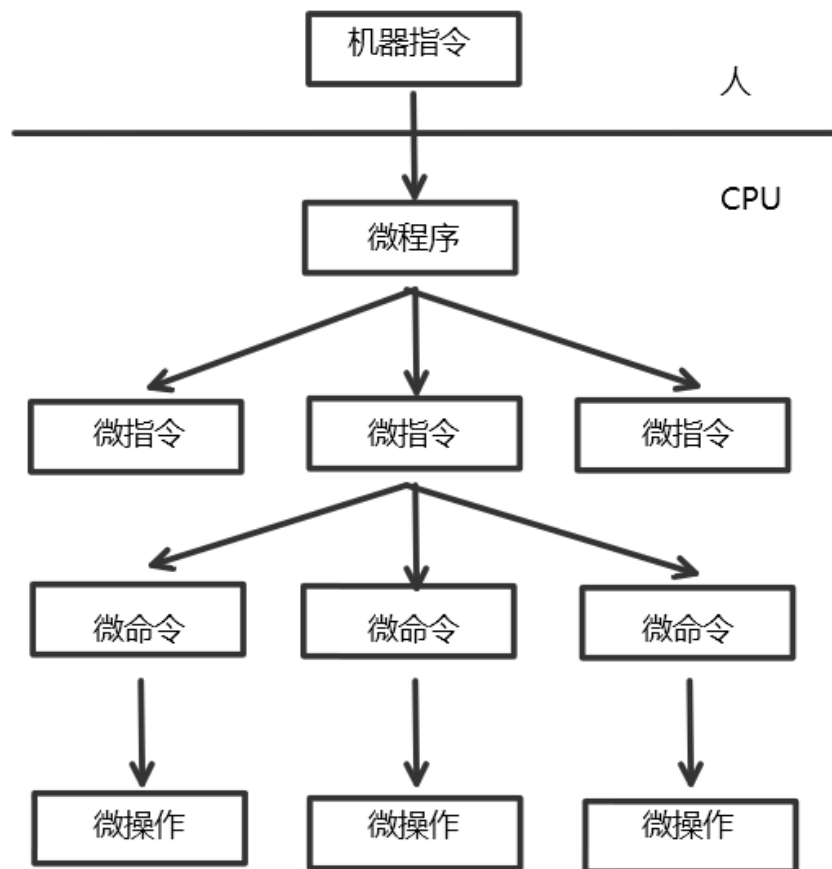
控制信号：

微程序控制器：

是将全部控制信号存储在控制存储中。控制信号的逻辑实现、设计、改动都比较容易；产生控制信号的时间较长。

组合逻辑控制器：

硬布线方案控制器，用组合逻辑的门电路实现控制信号。产生控制信号所需的时间较少，对提高系统的运行速度有利；
控制信号的设计复杂，用门电路设计实现也比较困难，尤其是要变动一些设计极其不方便



机器指令：

提供给使用者编制程序的基本单位，它表明CPU所能完成的基本功能

微程序：

一系列微指令的有序集合称为微程序，用来解释执行机器指令

微指令：

为实现机器指令中一步操作的微命令组合，它作为CPU内部的控制信息，通常不提供给使用者，对程序员透明

微命令：

构成控制信号的最小单位称为微命令

微操作：

由微命令控制实现的最基本的操作称为微操作，如开门、关门

微周期：

从控制存储器中读取一条微指令，并执行相应操作所需要的时间

P78

指令周期：

将一条机器指令从读取到执行完所用的时间

机器周期：

CPU完成一次基本操作所花费的时间，通常用从内存中读取一个指令字并执行完所花的时间

时钟周期：

也叫节拍、频率。一次从M读出，并经数据通路传送的操作或一次数据通路传送的操作或一次向M写入的操作

P135

模型机的三级时序：

工作周期，时钟周期，工作脉冲三级时序

P135

操作数的三种位置：

指令中、内存中、寄存器中

P93

寻址方式（写一道指令，判断是哪一种寻址方式）：

1.立即寻址：

指令中就含有操作数，可立即使用

2.直接寻址：

操作数在内存中，指令给出操作数的有效地址，根据该地址可从主存中读取或写入操作数

3.寄存器直接寻址：

操作数放在寄存器中，指令直接给出寄存器号，操作数实际存储在指定编号的寄存器中

4.主存间接寻址：

指令中给出主存间址单元地址，从间址单元中取出操作数地址，再用该地址从主存中读取或写入操作数

5.寄存器间接寻址：

指令中给出寄存器号，该寄存器中存放操作数的有效地址，通过该地址从主存中读取或写入操作数

6.变址寻址：

指令提供基准量（不变），R提供修改量（可变）；适于处理一维数组

7.基址寻址：

指令提供位移量（不变），R提供基准量（可变）

变址寻址方式立足于面向用户，可用于访问字符串、数组、表格等成批数据（或其中的某些元素）；基址寻址方式立足于面向系统，可用来解决程序在实际主存中的重定位问题，以及在有限字长指令中扩大寻址空间等。

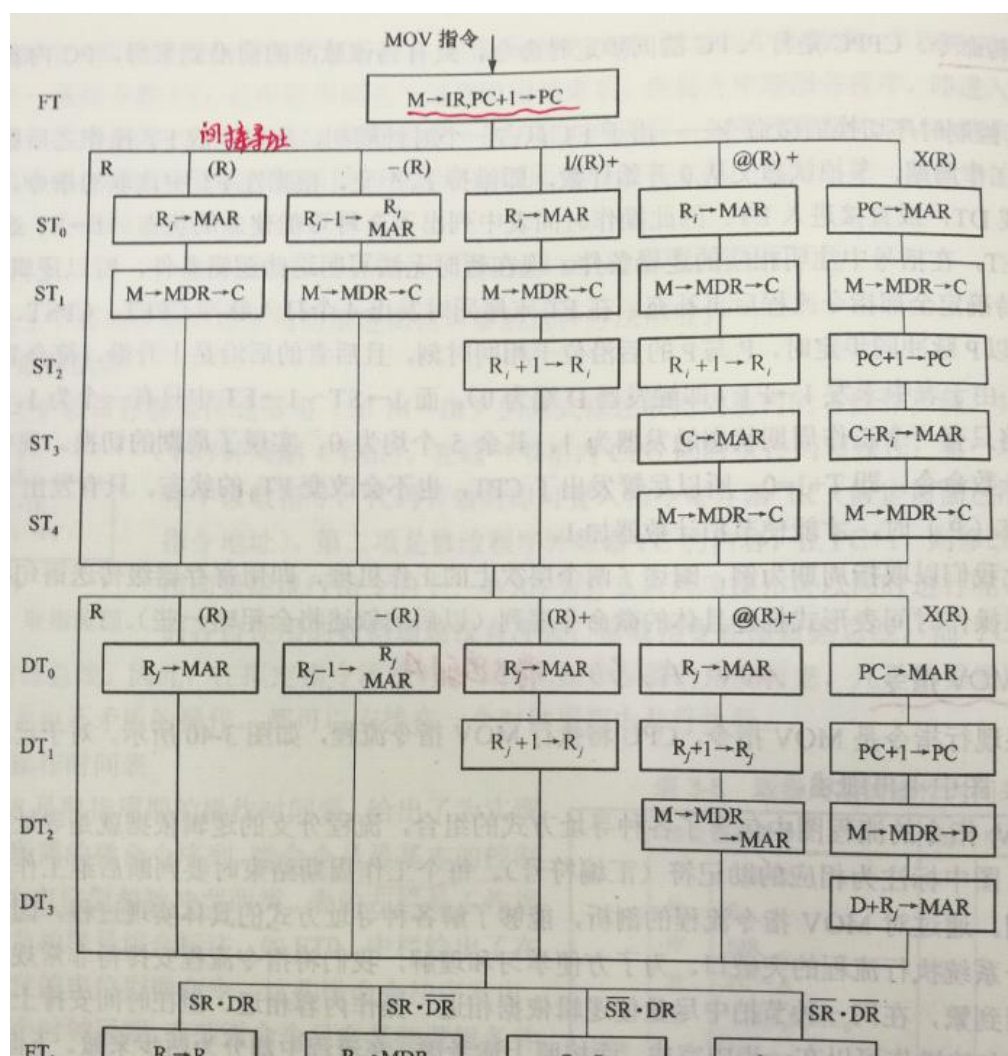
8.基址加变址寻址

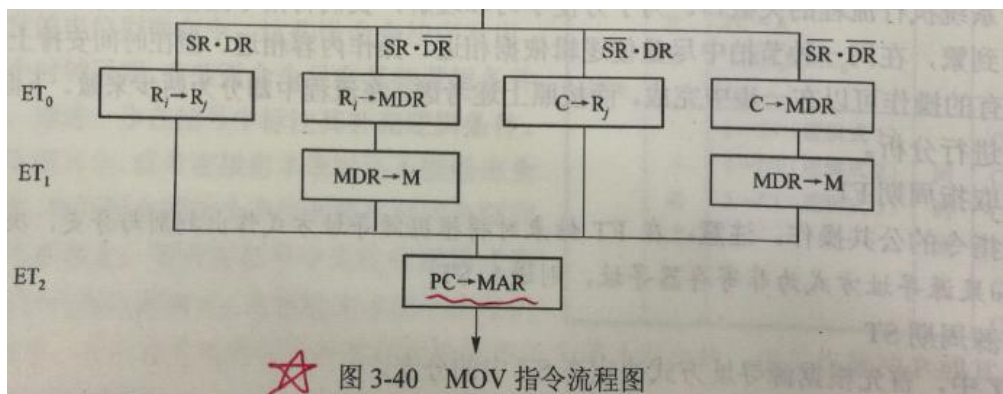
9.相对寻址

10.页面寻址

11.堆栈寻址

P93





直接寻址不带帽子，间接寻址带

X 是变址方式标志，即寻址方式字段代码为 101； \bar{X} 表示寻址方式为非变址型； SR 表示源操作数采用寄存器寻址方式，若不是寄存器寻址方式，则以 \bar{SR} 表示； DR 表示目的地址采用寄存器寻址方式，否则以 \bar{DR} 表示。

P140

第四章 存储子系统

虚拟存储器概念：

虚拟存储器只是一个容量非常大的存储器的逻辑模型，不是任何实际的物理存储器。虚拟存储器的主要思想是把可访问的逻辑地址空间和实际的物理内存空间分开，借助于磁盘等辅助存储器来扩大主存容量，可以为更大或更多的程序所使用。

P190

存储器的技术指标：

首要特征是存储容量，主存储器通常用字节或字表示，外存储器的容量通常用字节表示。与容量相关的一个概念是传输单位，对于主存，传输单位是指每次读出或写入存储器的位数，通常等于字长；对于外存储器，传输单位一般为数据块。

存储器的其它重要特性是与存取速率相关的三个参数：存取时间、存取周期、数据传输率

P193

动态RAM的特点，有哪几种刷新方式，与静态RAM的区别：

SRAM的制造工艺比DRAM稍复杂，相同体积情况下，每片SRAM的容量约为DRAM的1/16。但SRAM的速率较快，每片容量相同时，SRAM的访问时间约为DRAM的1/3 ~ 1/2。

P197

DRAM基本储存原理：

将信息以电荷形式存于电容之间，通常将其定义为电容充电至高电平，对应的信息为1；放电至低电平，则对应的信息为0

特点：

用电容存储电荷的方式来存储信息，不需要双稳态电路，因而可以简化结构。充电后MOS管断开，既可使电容电荷泄放极少，还能大大降低芯片的功耗。

P201

刷新方式：

- 1、集中刷新方式
- 2、分散刷新方式
- 3、异步刷新方式

P221

cache的作用、位置、几种替换算法：

作用：

cache用来存放当前最活跃的程序和数据,作为主存某些局部区域数据的副本

位置：

位于主存储器与CPU的通道寄存器组之间

P255

常用的替换算法：

先进先出算法（FIFO）、近期最少用算法（LRU）

P260

x86是统一编址还是独立编址：

x86是独立编址的体系结构

百度

统一编址和独立编址各自的优缺点：

独立编址(专用的I/O端口编址)----存储器和I/O端口在两个独立的地址空间中

优点：

I/O端口的地址码较短，译码电路简单，存储器同I/O端口的操作指令不同，程序比较清晰；存储器和I/O端口的控制结构相互独立，可以分别设计

缺点：

需要有专用的I/O指令，程序设计的灵活性较差

统一编址(存储器映像编址)----存储器和I/O端口共用统一的地址空间，当一个地址空间分配给I/O端口以后，存储器就不能再占有这一部分的地址空间

优点：

不需要专用的I/O指令，任何对存储器数据进行操作的指令都可用于I/O端口的数据操作，程序设计比较灵活；由于I/O端口的地址空间是内存空间的一部分，这样，I/O端口的地址空间可大可小，从而使外设的数量几乎不受限制

缺点：

I/O端口占用了内存空间的一部分，影响了系统的内存容量；访问I/O端口也要同访问内存一样，由于内存地址较长，导致执行时间增加

百度

堆栈的特点：

在计算机领域，堆栈是一个不容忽视的概念，堆栈是两种数据结构。堆栈都是一种数据项按序排列的数据结构，只能在一端(称为栈顶(top))对数据项进行插入和删除。在单片机应用中，堆栈是个特殊的存储区，主要功能是暂时存放数据和地址，通常用来保护断点和现场。要点：堆，队列优先,先进先出（FIFO—first in first out）；栈，先进后出(FILO—First-In/Last-Out)。

堆栈空间分配：

栈（操作系统）：

由操作系统自动分配释放，存放函数的参数值，局部变量的值等。其操作方式类似于数据结构中的栈。

堆（操作系统）：

一般由程序员分配释放，若程序员不释放，程序结束时可能由OS回收，分配方式倒是类似于链表。

堆栈缓存方式：

栈使用的是一级缓存，他们通常都是被调用时处于存储空间中，调用完毕立即释放。

堆则是存放在二级缓存中，生命周期由虚拟机的垃圾回收算法来决定（并不是一旦成为孤儿对象就能被回收）。所以调用这些对象的速度要相对来得低一些。

堆栈数据结构区别：

堆（数据结构）：

堆可以被看成是一棵树，如：堆排序。

栈（数据结构）：

一种先进后出的数据结构。

[百度百科](#)

P103

存储器和CPU连接时要考虑哪几个问题：

数据锁存：指令周期译码和数据周期数据分别处理

片选：如何避免与其他外设冲突

读写信号：如何译码才能在正确的时刻给正确的地址送数

数据线和地址线：实际需要连多少根线

以上信号还要规划好时序，配合好速率，看需不需要做信号间搭电平转换

[百度](#)

第五章 总线与输入输出子系统

双端口存储器：

双端口存储器具有两个彼此独立的读/写口，每个读/写口都有一套独立的地址寄存器和译码电路，可以并行的独立工作。两个读/写口可以按各自的接收的地址同时读出或写入，或一个写入而另一个读出数据。与两个独立的存储器不同，双端口存储器两个读写端口的访问空间相同，可以访问同一区域甚至同一单元。

P266

总线、中断的定义：

总线是一种用来连接计算机各功能部件并承担部件之间信息传输任务的公共信息通道，能在各部件之间传输数据和控制命令。总线被多个部件分时共享，每一时刻只能有一个设备掌握总线进行数据收发，但多个设备可以同时从总线接收数据

P275

程序中断方式简称为中断，指在计算机的运行过程中，如果发生某种随机事态，CPU将暂停执行现行程序，转去执行中断处理程序，为该随机事态服务，并在服务完毕后自动恢复原程序的执行。由此可见，中断的这一定义包含了程序切换和

随机性两个重要特征

P296

串口、并口哪个快（频率）：

在频率相同的情况下并口更快，通常情况下串口频率远高并口，高速度更快

USB是串口

2.0、3.0传输上限：

USB 2.0的速率为480 Mbps，USB 3.0则可达到4.8 Gbps。（理论值）

[百度百科](#)

PCI外部组件互联总线：

传输上限：

到目前为止，PCI已发展到了PCI-E3.0阶段，能支持高达2.5/5/8GHz 的总线频率，采用128/130编码方式，最高可支持32个双工通道X32，其最大带宽可达64GBps。

P292

中断管理芯片：

中断控制芯片8259

P302

终端、查询、DMA对应（低、中、高速）：

DMA：高速

中断：中速

查询：低速

P318

DMA全称、背景、工作流程、与CPU的关系：

全称：

直接存储器访问（Direct Memory Access）8527

定义：

依靠硬件直接在主存与外围设备之间进行数据传输，在传输过程中不需要CPU的干预。

与CPU的关系：

DMA传输周期的插入不影响CPU程序的执行状态，除非CPU和DMA访问主

存引起冲突，否则CPU可以继续执行自己的程序，因而显著提高了CPU的利用率；但是正因无CPU的参与，DMA方式也只能处理简单的数据传输。

P318

工作流程：

P329

第六章 输入输出设备及接口

键盘的类型有哪些：

- 1:有触点式
- 2:无触点式
- 3:虚拟式

P346

补充

计算题：

补码一位乘法

表 2-6 补码一位乘的规则

Y_n (高位)	Y_{n+1} (低位)	操 作
0	0	$1/2 A_n$
0	1	$1/2 (A_n + X_n)$
1	0	$1/2 (A_n - X_n)$
1	1	$1/2 A_n$

原部为0；再右移一位
原部为1；...
原部为1；EX111；...
同一

(2) 运算实例

【例 2-49】 $X = -0.1101$, $Y = -0.1011$, 求 $[XY]_*$ = ?

设 $A = 00.0000$, $B = X_* = 11.0011$, $-B = -X_* = 00.1101$, $C = Y_* = 1.0101$ 。

步数	条件	操作	A	C	C_{n+1}
			00.0000	1.0101	0
第一步	10	-B	+ 00.1101		
			00.1101		
		→	00.0110	11.010	1
第二步	01	+B	+ 11.0011		
			11.1001		
		→	11.1100	111.01	0
第三步	10	-B	+ 00.1101		
			00.1001		
		→	00.0100	1111.0	1
第四步	01	+B	+ 11.0011		
			11.0111		
		→	11.1011	1111L	0
第五步	10	-B	+ 00.1101		
			00.1000	1111	

则 $[XY]_* = 0.10001111$

本例以分步算式的形式表明了补码一位乘法的运算过程，下面再对有关要点进行说明。

←最后一步不移位

P57

在乘除运算中广泛使用多符号位，但在主存中仍保持单符号位，运算时在扩充为多符号位，运算结束后又紧缩为单符号位，保存到主存之中。

P51

最后两道大题：

存储器的搭建 (P212)

MOV指令流程 (P140)：

MOV A,B; 将B送到A

实验：74181特点：