

1.

算术逻辑部件（ALU）：用来进行算术逻辑运算的部件，能实现多组算术运算和逻辑运算的组合逻辑电路。

程序计数器（PC）：用于存放下一条指令所在单元的地址的地方。在执行当前指令过程中，自动计算出下一条指令的地址并送到PC中保存。

指令寄存器（IR）：用于临时储存从主存取来的指令。

控制器：用于自动逐条取出指令并进行译码的部件。

指令集体系结构（ISA）：软件和硬件之间接口的一个完整定义。ISA定义了一台计算机可以执行的所有指令的集合，每条指令规定了计算机执行什么操作，所处理的操作数存放的地址空间以及操作数类型。ISA规定的内容包括数据类型及格式，指令格式，寻址方式和可访问地址空间的大小，程序可访问的寄存器个数、位数和编号，控制寄存器的定义，I/O空间的编制方式，中断结构，机器工作状态的定义和切换，输入输出结构和数据传送方式，存储保护方式等。因此，可以看出，指令集体系结构是指软件能够感知到的部分，也称软件可见部分。

机器指令：是CPU能直接识别并执行的指令，它的表现形式是二进制编码。机器指令通常由操作码和操作数两部分组成，操作码指出该指令所要完成的操作，即指令的功能，操作数指出参与运算的对象，以及运算结果所存放的位置等。

2.

(1)冯诺依曼计算机：

①运算器：计算机中执行各种算术和逻辑运算操作的部件。运算器的基本操作包括加、减、乘、除四则运算，与、或、非、异或等逻辑操作，以及移位、比较和传送等操作，亦称算术逻辑部件（ALU）；

②控制器：由程序计数器、指令寄存器、指令译码器、时序产生器和操作控制器组成，它是发布命令的“决策机构”，即完成协调和指挥整个计算机系统的操作。运算器和控制器统称中央处理器，也叫做CPU。中央处理器是电脑的心脏；

③存储器：存储器分为内存和外存。内存是电脑的记忆部件，用于存放电脑运行中的原始数据、中间结果以及指示电脑工作的程序。外存就像笔记本一样，用来存放一些需要长期保存的程序或数据，断电后也不会丢失，容量比较大，但存取速度慢。当电脑要执行外存里的程序，处理外存中的数据时，需要先把外存里的数据读入内存，然后中央处理器才能进行处

理。外存储器包括硬盘、光盘和优盘；

④输入设备：输入设备是向计算机输入数据和信息的设备。是计算机与用户或其他设备通信的桥梁。输入设备是用户和计算机系统之间进行信息交换的主要装置之一。键盘，鼠标，摄像头，扫描仪，光笔等都属于输入设备。

⑤输出设备：是计算机硬件系统的终端设备，用于接收计算机数据的输出显示、打印、声音、控制外围设备操作等。也是把各种计算结果数据或信息以数字、字符、图像、声音等形式表现出来。常见的输出设备有显示器、打印机等。

(2)存储程序工作方式：就是把解决问题的指令提前存入存储器中，在机器通电之后，就能按照指定的逻辑循序取出存储器的指令进行处理自动的完成工作。

(3)

①取指令阶段

取指令（**Instruction Fetch, IF**）阶段是将一条指令从主存中取到指令寄存器的过程。

②指令译码阶段

取出指令后，计算机立即进入指令译码（**Instruction Decode, ID**）阶段。

在指令译码阶段，指令译码器按照预定的指令格式，对取回的指令进行拆分和解释，识别区分出不同的指令类别以及各种获取操作数的方法。

在组合逻辑控制的计算机中，指令译码器对不同的指令操作码产生不同的控制电位，以形成不同的微操作序列；在微程序控制的计算机中，指令译码器用指令操作码来找到执行该指令的微程序的入口，并从此入口开始执行。

③执行指令阶段

在取指令和指令译码阶段之后，接着进入执行指令(**Execute, EX**)阶段。

此阶段的任务是完成指令所规定的各种操作，具体实现指令的功能。为此，CPU的不同部分被连接起来，以执行所需的操作。

④访存取数阶段

根据指令需要，有可能要访问主存，读取操作数，这样就进入了访存取数(**Memory, MEM**)阶段。

此阶段的任务是：根据指令地址码，得到操作数在主存中的地址，并从主存中读取该操作数用于运算。

⑤结果写回阶段

作为最后一个阶段，结果写回(Writeback, WB)阶段把执行指令阶段的运行结果数据“写回”到某种存储形式：结果数据经常被写到CPU的内部寄存器中，以便被后续的指令快速地存取；在有些情况下，结果数据也可被写入相对较慢、但较廉价且容量较大的主存。许多指令还会改变程序状态字寄存器中标志位的状态，这些标志位标识着不同的操作结果，可被用来影响程序的动作。

在指令执行完毕、结果数据写回之后，若无意外事件（如结果溢出等）发生，计算机就接着从程序计数器PC中取得下一条指令地址，开始新一轮的循环，下一个指令周期将顺序取出下一条指令。

5.

若在该模拟机上实现 $z=(x-y) \times y$ ；x和y分别储存在主存7,8号单元，结果z储存在9号单元

主存地址	主存单元内容	内容说明（II表示第I条指令）	指令的符号表示
0	1110 1000	I1:R[0]←M[8],op=1110:取数操作	load r0,8#
1	0000 0100	I2:R[1]←R[0],op=0000:传送操作	mov r1,r0
2	1110 0111	I3:R[0]←M[7],op=1110:取数操作	load r0,7#
3	0010 0001	I4:R[0]←R[0]-R[1],op=0010:减操作	sub r0,r1
4	0011 0001	I5:R[0]←R[0]*R[1],op=0011:乘操作	mul r0,r1
5	1111 1001	I6:M[9]←R[0],op=1111:存数操作	store 9#,r0
6	(x的二进制)	操作数x	
7	(y的二进制)	操作数y	
8	00000000	z，初始值为0	

实现 $z=(x-y) \times y$ 功能的每条指令执行过程

	I1:1110 1000	I2:0000 0100	I3:1110 0111	I4:0010 0001	I5:0011 0001	I6:1111 1001
取指令	IR←M[0000]	IR←M[0001]	IR←M[0010]	IR←M[0011]	IR←M[00100]	IR←M[0101]

	I1:1110 1000	I2:0000 0100	I3:1110 0111	I4:0010 0001	I5:0011 0001	I6:1111 1001
指令译码	op=1110,取数	op=0000:传送	op=1110:取数	op=0010:减	op=0011:乘	op=1111:存数
PC增量	PC←0000 +1	PC←0001 +1	PC←0010 +1	PC←0011 +1	PC←0100 +1	PC←0101 +1
取数并执行	MDR←M[1000]	A←R[0]、 mov	MDR←M[1001]	A←R[0]、 B←R[1]、 sub	A←R[0]、 ←R[1]、mul	MDR←R[0]
送结果	R[0]←MDR	R[1]←F	R[0]←MDR	R[0]←F	R[0]←F	M[1001]←MDR

指令执行各阶段都包含若干个微操作，微操作需要相应的控制信号(control signal) 进行控制。

取指令阶段 $IR \leftarrow M[PC]$ 微操作有: $MAR \leftarrow PC$;控制线 $\leftarrow Read$; $IR \leftarrow MDR$ 。

取数阶段 $R[0] \leftarrow M[addr]$ 微操作有: $MAR \leftarrow addr$; 控制线 $\leftarrow Read$; $R[0] \leftarrow MDR$ 。存数阶段 $M[addr] \leftarrow R[0]$ 微操作有: $MAR \leftarrow addr$; $MDR \leftarrow R[0]$; 控制线 $\leftarrow Write$ 。

ALU运算 $R[0] \leftarrow R[0] - R[1]$ 微操作有: $A \leftarrow R[0]$; $B \leftarrow R[1]$; $ALUOp \leftarrow sub$; $R[0] \leftarrow F$ 。ALU操作有加(add)、减(sub)、与(and)、或(or)、传送(mov) 等类型，ALU操作控制信号ALUOp可以控制ALU进行不同的运算。例如， $ALUOp \leftarrow mov$ 时，ALU 的输出 $F=A$; $ALUOp \leftarrow sub$ 时，ALU 的输出 $F=A - B$ 。

3.

计算机在运行时，先从内存中取出第一条指令，通过控制器的译码，按指令的要求，从存储器中取出数据进行指定的运算和逻辑操作等加工，然后再按地址把结果送到内存中去。接下来，再取出第二条指令，在控制器的指挥下完成规定操作。依此进行下去。直至遇到停止指令。程序与数据一样存取，按程序编排的顺序，一步一步地取出指令，自动地完成指令规定的操作。