课堂记录：

COMPUTER分为 software与hardware

software：分为system software与application software

system software: C++等编程语言等等，特点是Not fun，系统本身也是system software。

application software:游戏等等

resource：内存，显示器…

读取内存：load,seek,out 这三个method:本质

write(A,B,C//loadseekout三个method用的参数){

load(A);

seek(B);

out(C);

}

通过把三个method合成一个write method，进行了第一步的整合。

然后再把这个methood保存成一个文件，进一步整合，我们想读取的时候直接激活这个文件，这些文件组合成了系统operatin system

resource sharing：

space multipled:

比如说内存，这个任务用内存的A部分，那个任务用内存的B部分，（实际上不是同时运行的）

time-multipled

resource无法切割，比如说打印机，你不可能同时打印两份文档，所以，我们分不同时间打印，

CPU的使用也是time-multipled,每次运行memory中的一部分，所以实际上memory也是time multipled,但是因为CPU的运行速度很快，相当于内存同时运行的

假设，task1 用时12， task 2用时28

那么Multiple program用时28,（最大数）， unique program用时40（总和），

multiple program存在的问题：隐私泄露，

batch processing:

批处理任务（英语：batch processing），又称批次处理任务，是指在[计算机](https://zh.wikipedia.org/wiki/%E8%AE%A1%E7%AE%97%E6%9C%BA)上无须人工干预而执行系列程序的作业。

批处理任务无须人工交互，所有的输入数据预先设置于程序或命令行参数中。这是不同于需要用户输入数据的交互程序的概念。

//早期的，因为当时没有图形界面

timesharing system

多个用户分享同一个主机，CPU被分为多个时间片，轮流为每个终端用户服务，现在没了因为PC便宜。

问题：security,多人用一个

问题2：service，每一个人必须平等，不然一个人占尽了资源别人没法用，所以跑得慢，

总共任务数量为N\*M, 用户数\*每个人任务数量

embedded system（dedicated system）:

嵌入式系统，与电脑不同，处理特定任务，如记不起，电子手表,MP3等等，被嵌入进了机械内部

personal computer: one user,one program

distrubuted system:

用网络连接多个电脑的系统

WINDOWS:

他叫做windows因为你可以开很多视窗，他是multiple programming的

假如有一个水坝，有open与close两个操作，

我们有两种方法控制：

1.applicatio与OS:创造两个software, open()与close()，可以分别激活OS中的

open(){

………

}

与CLOSE//在OS中被详细描述，application只是为了激活

2.只有application，在application中就被详细描述

第二种更方便，因为不用激活，直接使用，节省时间

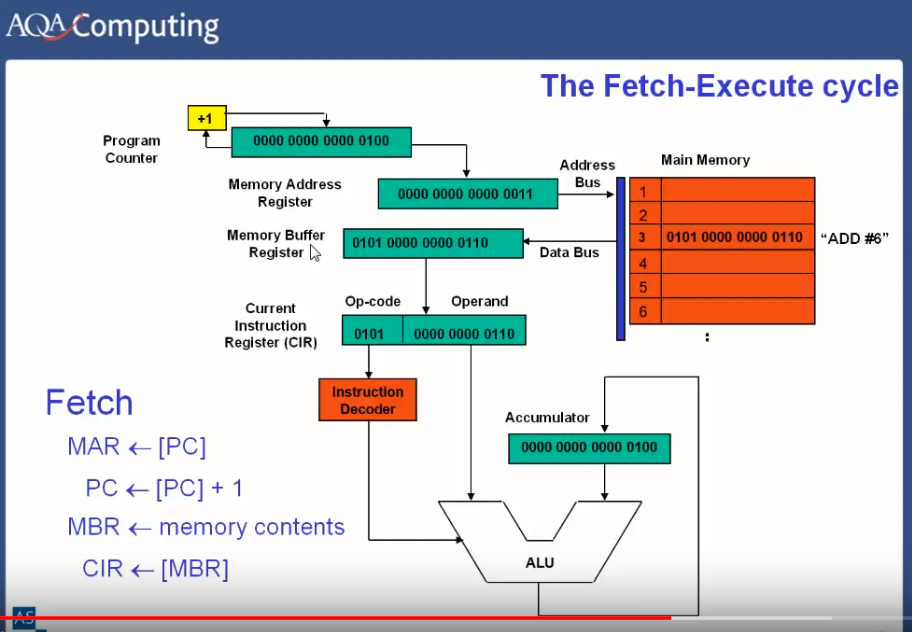
RAM ROM区别，

RAM:random access memory:与CPU直接交换数据的内部存储，速度很快，通常作为操作系统或者其他正在运行中的程序的临时数据存储介质，是随机存取的，随便进入的

例子：只要一开机，系统就会被存入RAM里，关机，RAM被清空，系统在hard disk里

ROM：read only memory,只能被写一次，然后就只能读取，开关机不会让数据清空，用来存储特定程序，比如说，ROM可以用来激活OS，ROM有一条真实存在的线，直接连到PC（program counter），这样通电时就会有一个初始PC值，可以激活系统，

fetch cycle:



先把PC里的值0000 0000 0000 0011 赋值给MAS，

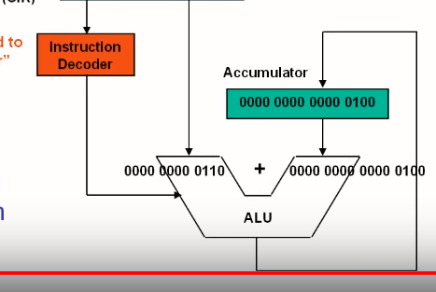
然后PC+1=0000 0000 0000 0100，

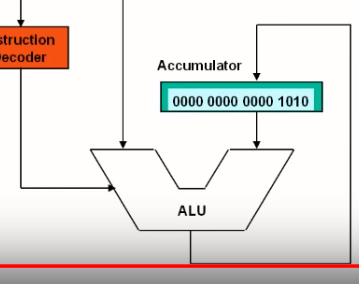
MAS也就是PC初始值通过address bus，也就是等于3，激活main memory里的数据，这个数据通过data bus传给memory buffer register，

0101 是op code，意思是加号，0000 0000 0110是operand也就是要加的数，

0101通过instruction deccoder解码器做成加号，给ALU使用，0000

0000 0110也就是6给ALU的一端使用，另一端是accumulator累加器，是之前就有的数，现在为4，

6+4=10

然后10被赋给累加器

IR:instruction register,指令暂存器，就是图里的CIR,暂时存放指令

PC:program counter，一个增长的计数器，来track 下一个要被执行的操作的memory address

MAR:图里的MAR，存储当前被执行的操作的地址，

MDR:图里的MBR，存储memory中的data

CU：解码指令

还有一个context switching 没看

multiprocessing:用了多核

multiprogramming:通过switch task in cpu，让一个单核CPU同时处理多个process的能力

multitasking:哪怕是单核CPU，在同一时间，也能开多个软件（具体体现）