

程序执行

程序执行

- 计算机总是按照需要需要把储存器里的指令复制到CPU中来执行存储器中的程序。每个指令到达CPU，就会被解码执行。从存储器中取指令的顺序与与这些指令储存在存储器中的顺序相对应，除非JUMP指令更改。
- CPU两个专用的寄存器
 - **指令寄存器(instruction register)**:用于储存正在执行的程序。
 - **程序计数器(program counter)**:包含的是下一个待执行的指令地址，用于以机器的方式跟踪程序执行到了什么地方。
- **机器周期(machine cycle)**包括三步处理：
 - 取指：因为我们的机器的每一条指令长度是2个字节,所以取指过程中需要从主储存器中读取2个储存单元内容。CPU将读取的指令存储在它的指令寄存器，然后将程序计数器的值加2，使得程序计数器包含存储器中的存储的下一条指令的地址。
 - 译码：这时指令已经存入了指令寄存器，CPU对指令译码，其中包括根据该指令的操作码将操作数字段分解为适当的部分。
 - 执行：CPU激活相应电路以执行命令，完成所请求的任务。
- 一旦指令寄存器中的指令完成，CPU又将从取指步骤开始下一个周期。
- JUMP的指令比较特殊，一般会执行两个"**machine cycle**".

程序与数据

- 许多程序可以同时储存在一个计算机的主存储器里，只要它们地址不同。机器开启时运行哪个程序，可以通过适当的设置程序计数器来决定。
- 数据也是储存在存储器中，并且和程序一样由0和1编码。所以机器不知道哪些是数据，哪些是程序。如果程序计数器被赋予了数据的地址而非所希望程序的地址，那么在没有更好的选择下，机器会像处理程序一样处理数据。
- 这个可以显示一种好的特性(虽然将数据和程序放在一起很愚蠢，有时会出错)，但它可以使得某一个程序操控其他程序（甚至是自己）。