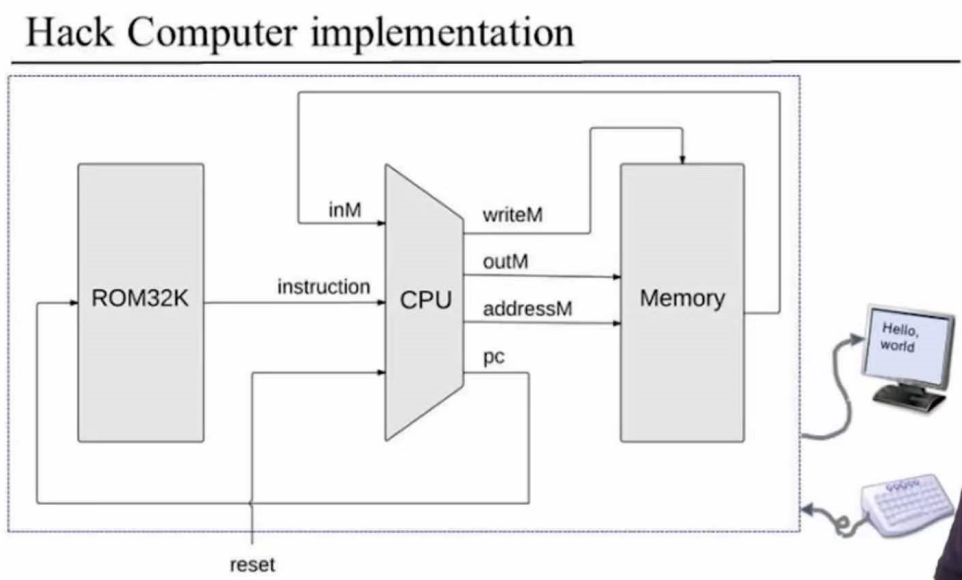


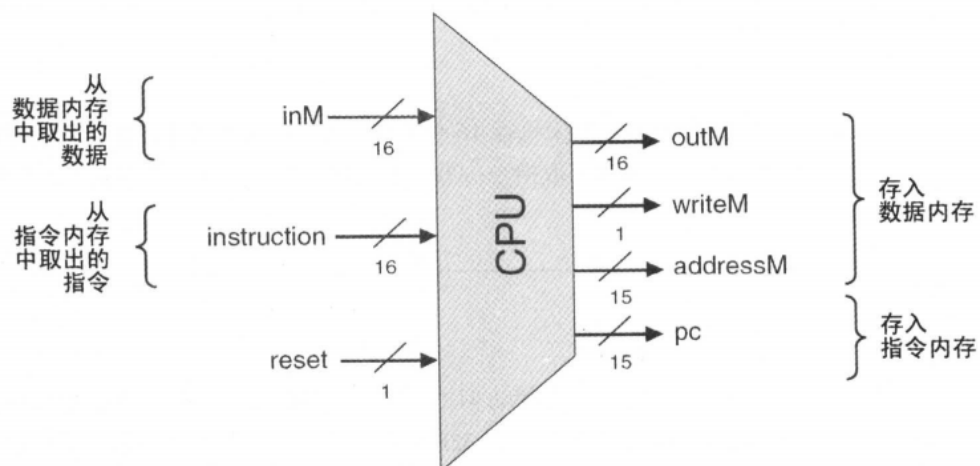
Chapter 5 Computer Architecture

- *Form ever follows function
- 形式永远跟随功能。
- Introduction
 - 通用图灵机
 - 多用途单机器（计算机器）模型可以“运行”任何任意（但well-formed）指令序列（称为 "quintuples"）
 - 实现的架构：冯诺依曼架构
 - 实践：通用计算机Hack
- Background
 - 存储程序：The stored program
 - 机械计算机的程序逻辑被嵌入到硬件中
 - 数字计算机：功能多样性，程序逻辑在存储设备中，形成了软件
 - 基于固定的硬件平台，执行指令集
 - 冯诺依曼架构
 - 通用图灵机（1936）：逻辑基础
 - 冯诺依曼机（1945）：实际应用型的体系结构，是所有pc的基础
 - 中央处理单元：CPU
 - 记忆设备：内存（存储数据与指令
 - 输入输出：IO
 - 内存
 - 以二进制数存储在随机存储器中，一个独立的word通过地址指定
 - 数据内存：变量，数组，对象
 - 通过寻址可以进行读写
 - 指令内存
 - 高级命令——机器语言：二进制
 - 在计算机操作的每一步，计算机会从指令中取出一个word，解码执行，然后计算下一条
 - CPU
 - 执行已被加载到指令内存的指令，进行计算，读写内存，跳转到其他指令
 - 算术逻辑单元：ALU进行算术操作，逻辑操作
 - 寄存器：把与运算有关的数据暂存到CPU内
 - 控制单元：
 - 解码指令

- 决定下一步的指令
- CPU的工作类似于loop，取指令，执行
- 寄存器：
 - 数据寄存器
 - 局部计算
 - 寻址寄存器
 - 间接寻址
 - 程序计数寄存器
 - 下一条指令在指令内存中的地址
- 输入输出
 - I/O映像
- Specification
 - Hack Computer 架构：运行任何以hack语言编写的程序（通用图灵机）

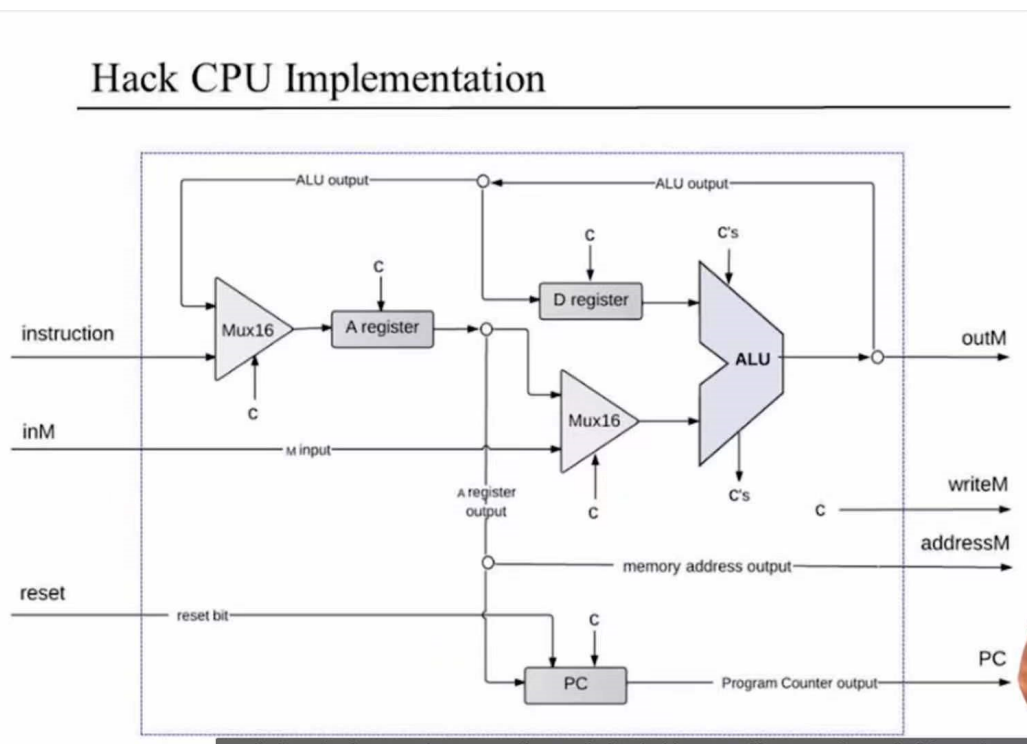


- CPU
 - CPU是计算机体系的核心

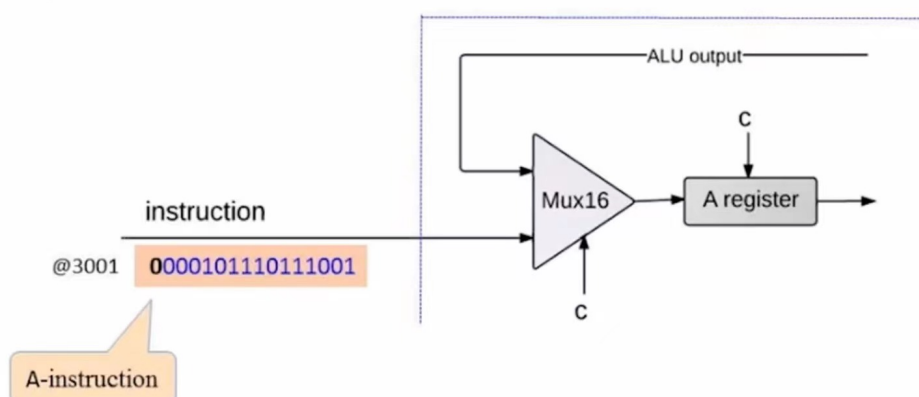


- 输入
 - 数据值
 - 指令
 - 重置位
- 输出
 - 写入的数值outM
 - 写入地址addM
 - 加载位, 写入启动
 - pc: 下一个指令的地址

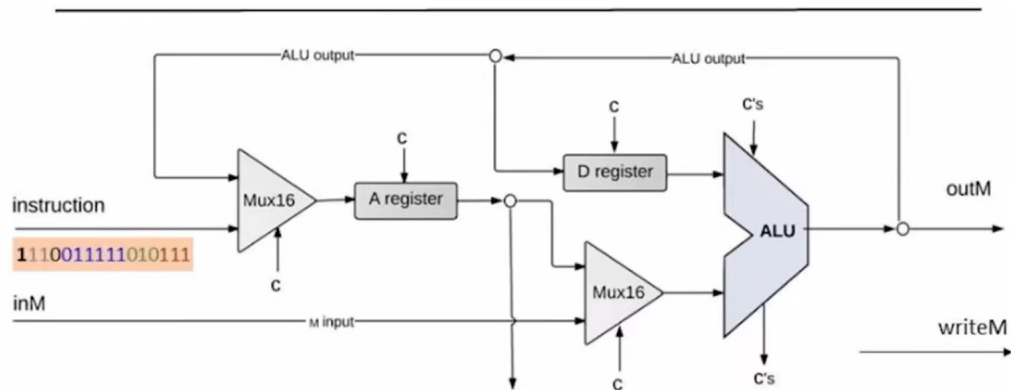
- 构造:



- 3个寄存器: A,D,PC (计数器)
- 2个MUX16
- 一个ALU
- 模块化拆分
 - 指令区

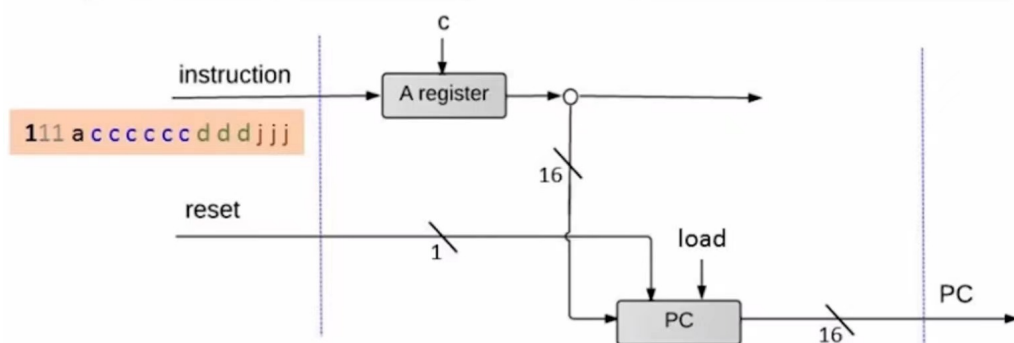


- 指令下达，解码分析A指令&C指令
- 决定MUX是接受指令的值&ALU的输出值
- 计算区



- ALU为组合芯片，任何时间均有输入输出
- ALU获取两个输出（可以看看chap4中的函数表）
 - D寄存器的数据值
 - 由Mux16负责决定，A寄存器的值&M寄存器的值
 - 实际上是一个判断直接寻址&间接寻址的过程
 - 取A直接处理A中数据
 - 取M，A其实为地址
- ALU获得六个控制位决定计算种类（6c从C指令获取）
- ALU输出：由目标位控制（3个容器独立接受）
 - D寄存器
 - 经Mux决定是否输入进A寄存器
 - 输出CPU，进入内存
- 同时ALU也会输出控制位ZR,NU，判断输出是否为0&负数
- 控制区

Control (abstraction)



- 在C指令中，可能会有jump位

- 程序计数器PC：决定从程序的哪一行执行
 - reset时重启PC，使程序从第一位开始
 - 常规无jump, PC++
 - 无条件jump, PC=A
 - 有条件jump, PC=++&A
- 实现
 - reset位=1, 则PC=0
 - load位：一个函数，判断是否跳转
 - 确定跳转, PC=A
 - 关于这里，包含跳转的C指令不会操作M，也就保证了A没有冲突。
 - @ 100 ; D=D-1 JEQ
 - 否则, PC++

• Data memory

• Instruction memory

• Implementation

- CPU:
 - 指令解码：16位指令解码
 - 指令执行：指令的各个域被同时发送到CPU的各个组件，协同执行
 - 读取下一条指令：由程序计数器输出地址
- 内存
 - RAM16K, SCR,KBD

• Perspective

- 有限状态机的硬件实现（事实上在工科创的单片机编程时处理过）
- Hack计算机：数据内存与程序内存独立（Harvard架构，而非标准冯诺依曼架构）
 - 可以在一个周期完成获取，解码，执行
 - 适合嵌入式计算机：程序已经写入ROM
 - 也被称为专用计算机
- 通用计算机：标准冯诺依曼架构
 - 双周期控制逻辑
- 发展历程
 - CISC：复杂指令集计算机，为获得更好的性能，需要更多指令集
 - RISC：精简指令集计算机：简单的指令集更优越