这是中国科学技术大学计算机系统概论2020秋季学期课程

老师：安虹

office hour and Discussion:

(1) 4 TAs,fellow students

(2) Face to face help

Questions to answer:(1)What can computer do

(2)How are they done

(3)What can't computer do

什么是计算思维

实际问题——计算系统实现

bool（true&&false）：== != && || ! < > >= <=

Great Idea in Computer System

Great idea0:0 and 1

Great idea1:Turning Computering model:turning machines are universal

Universal Turing Machine 计算模型：Turning 结构模型：Von Neummann

theory——Practice：time cost power

ENIAC：with a Turing machine but without a Von Neummann Archeticture

Chapter 4

基本部件

内存

$2^{28} - by - 8 - bits$模式，寻址空间为$2^2 8$，寻址能力为8位

- 访存的流程
  - 读操作：向MAR加载地址(1 cycle),发送信号通知内存读取内存请求，内存将数据放回MDR
  - 写操作：将写入的数据放入MDR，地址放入MAR，再向内存发送写信号

处理单元

机器字长：一次处理的数据的量化大小

寄存器文件，减少访存的次数

输入和输出单元

控制单元

- 指令寄存器：保存正在执行的指令
- PC寄存器：指示下一条要处理的指令

LC-3机器模型

- 16位可寻址的
- 输入输出，采用内存映射外设控制寄存器实现
- 控制单元，发出控制信号，采用有限状态机实现

指令处理

指令与指令周期

- 指令的处理过程是在控制单元的作用下，精确地，一步一步地完成的，称这个执行的步骤和顺序称为指令周期，每一步称为一个节拍
- 一个指令周期一般包含6个节拍

取指令

- 将PC寄存器的内容装入MAR寄存器
- 地址对应的内存单元的内容被装入MDR，读内存
- 控制单元将MDR的内容装入IR寄存器
- 取指令花费的时间周期分析
    - 加载MAR,MDR各需要一个cycle
    - 读内存需要的cycle不确定

译码

检查指令的类型，输入4位操作码字段，输出116根使能线，同一时刻只有一根使能线有效

地址计算

指令存在地址计算操作，在此周期实现

取操作数

执行

存放结果

写入目的寄存器，MDR或register file中的寄存器

改变执行顺序

- 控制指令，改变指令的控制流

指令周期的控制

1 instruction cycle= 6 Phases 1 Phase= some steps

取指令节拍

- 状态1：实现PC自增和加载PC进入MAR
- 状态2：读入内存，装入MDR
- 状态3：MDR拷贝到IR

译码节拍

- 基于IR的输入，和操作码部分，有限状态机分支

执行节拍

- 修改PC的值(控制流转移指令)

停机操作

- 始终信号的产生-时钟信号发生器和run状态开关
- run=1透明输入，run=0，停止时钟

Chapter 5 LC-3结构

ISA概述

内存和寄存器组织

- 寄存器具备记忆特性和独立寻址的特点
- 最小的存储单元(可寻址)是一个word，大小为16bits

指令集

- 操作码+操作数
- 指令集还包含操作码集合，数据类型和寻址模式
- ADD指令的寻址模式是寄存器模式

操作码

- 三类指令：运算，数据传送和控制流转移

数据类型，寻址模式和条件码

- 数据类型：补码整数
- 寻址模式：立即数，寄存器，PC相对寻址，间接寻址，寄存器基址偏移
- 条件码:NZP

数据搬移指令

间接寻址：先用PC偏移寻址找到内存中某个存储单元的值，这个值是一个地址LDI,STI

基址偏移寻址和PC相对寻址：一个相对的是基址寄存器，一个相对的是PC寄存器

立即数寻址：PC与立即数偏移相加，装入目标寄存器，不需要访问内存

控制指令

条件跳转指令

立即数增量相对于增量PC而言，在执行节拍检测是否满足跳转条件

循环控制的方法

- 计数器
- 哨兵：事先不知道循环进行多少次，对数据内容判断，在处理的数据内容的结尾增加一个哨兵

JMP

- 解决下一跳指令不超过可计算的范围的问题
- 跳转到目标寄存器的存储值

Chapter 7

常用伪操作

.ORIG程序的开始位置

.BLKW开始占用一连串的地址空间

.STRINGZ初始化n+1个内存单元

汇编过程

两遍扫描

- 第一遍扫描：得到符号表
- 第二遍扫描：生成机器语言程序

可执行映像

多目标文件

- 一个可执行映像由国歌模块文件组成
- .EXTERNAL start，表示若本文件中没有发现这个标号，则可能在另一个外部模块中定义
- 链接阶段，补充生成完整的符号表

chapter 7

基本概念

设备寄存器

- 保存需要和计算机传输的数据
- 设备当前状态：是否空闲，最近处理的I/O任务

内存映射I/O和专用I/O指令

- 专用I/O指令：直接对目标寄存器操作
- 内存指令I/O

异步I/O和同步I/O

- 处理I/O设备与处理器频率不一致
- I/O设备与处理器处理节奏不一致，则I/O和CPU是异步的

中断驱动与轮询

- 中断驱动-外设控制
- 轮询-处理器反复查询数据是否准备好

键盘输入

输入寄存器

- 数据寄存器-仅仅低8个bits有效
- 状态寄存器-仅仅最高位bit有效

基本输入服务程序

1. 程序询问KBSR的最高位是否为1
2. 若为1，则读入KBDR中的数据到输入缓冲区
3. 若不为1，则回到1

内存映射输入的实现

将控制寄存器映射到某几个内存单元

显示器输出

基本输出寄存器

DDR:存放数据

DSR:存状态，Ready位

唯一不同的是DSR为1的时候代表可以向DDR中写入

基本服务程序

内存映射输出

一个更复杂的输入程序

中断驱动I/O

介绍

基本功能

- 强行终止当前程序的运行
- 使得处理器执行I/O设备请求
- 最后恢复被中断程序的执行

为什么引入中断驱动I/O

- 避免反复访存判断，占据CPU资源

中断信号的产生

- 中断使能机制——产生中断信号，终止当前执行程序
- 数据传输机制——处理中断信号

几个条件

- 设备自身需要服务——Ready bit
- 设备有请求服务的权限——Interrupt Enable bit
- 设备中断请求的优先级高于当前处理器运行程序的优先级——多个外设请求中断，取优先级最高的设备(排队电路)

来自设备中断信号

- 设备有请求服务的需求
  - 设备状态寄存器中的Ready位
- 请求权限
  - 中断使能标志，状态寄存器中IE位

中断优先级

- 设置多个程序优先级

中断检测

- 保存现场
- 检测中断信号，决定是否正常跳转
- 决定跳转
  - 加载更高优先级的中断状态寄存器
  - 保存打断程序的状态

中断返回

RTI instruction：POP PC and PSR from stack

chapter 9 TRAP程序和子程序

LC-3 TRAP程序

概述

- 了解基层I/O的原理是很困难的
- 硬件寄存器有特权的，不能赋予用户程序员直接读写状态寄存器的权限
- 操作系统，是由特权的程序
- 操作系统通过trap，获取控制权

TRAP机制

- 服务程序集合：由OS提供，但以用户身份执行，最多支持256个服务程序
- 起始地址表：256个服务程序的起始地址，表位于0000-00FF，陷入矢量表
- TRAP指令：操作系统以用户程序身份执行某个特定的服务程序
- 链接：通过链接回到用户程序，操作系统提供的返回用户程序的机制

TRAP指令

- 根据矢量表项的内容，设置PC为相应服务程序的起始地址
- 提供机制，使得返回到调用TRAP的程序——链接

任务

- 陷入矢量0扩展为16bit地址，装入MAR
- 读入相应的嵌入矢量表表项，放到MDR中
- 保存当前PC寄存器到R7
- MDR装入PC

完整机制

RET语句，JMP R7

I/O中断处理程序

HALT中断程序

改用TRAP以节省一个指令操作符字段的命名空间

保存现场

- 调用者保存
- 被调用者保存

子程序

调用-返回机制

- 保存调用返回地址
- 将返回地址装入PC

JSR/JSRR指令

- 将返回地址装入R7,并跳转到对应PC处
- 区别在于寄存器偏移和PC偏移

库程序

- EXTERNAL表示出现某个符号由外部文件提供
- 在可执行映像的生成阶段(链接)，编译时负责将本程序和模块连接在一起

特权，优先级和地址空间

特权与优先级

- 特权：赋予某些程序执行某些特殊类型的权力
- 优先级：程序被执行的紧急性，决定程序是否有能力中断另一个程序执行
- 用户程序&&超级用户优先级，优先级和特权级不是相同的概念

Processor Status Register

- 程序状态寄存器
- BIT[15]：标志特权级，用户级或者系统级
- BIT[10:8]：8个优先级，最先优先级PL7,最低PL0
- 条件码

Organization of Memory

- X0000-X3000:Privileged Memory
  - x0000-SSP:System Space
  - ssp-x3000:Surpervisor Stack
- x3000-USP:User Space
- USP-XFE00:User Stack
- XFE00-XFFFF:I/O Page(外设控制寄存器的内存映射)