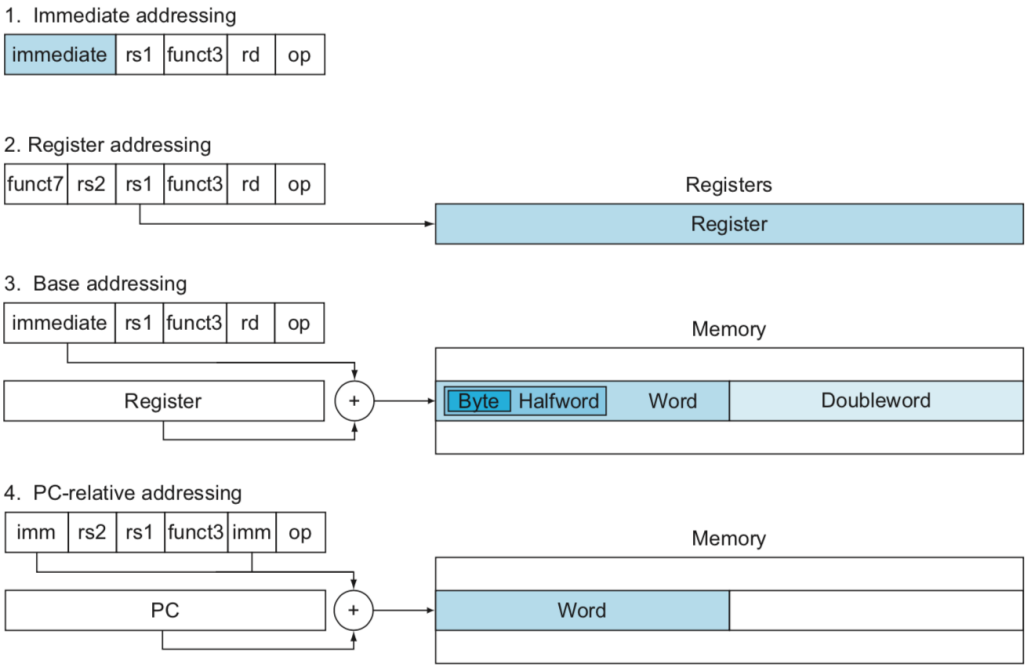
Computer Organization and Design A4 Cheatsheet for Final Exam - by Oscillated

- DMA
  - transfer between mem and I/O, no CPU intervention
  - steal a part of CPU period
  - notify CPU by interrupt
  - better for large transfers than interrupts
- Interrupt
  - 异步
  - When the process is returned after an interrupt service, **Register contents, Condition codes** should be loaded again.
- Polling
  - 开销大，inefficient，不常用
- Von Neumann architecture has data and instructions in the same memory space.
  - 哈佛结构是一种将程序指令存储和数据存储分开的存储器结构
- 在小端存储格式中，低地址中存放的是字数据的低字节，高地址存放的是字数据的高字节。
  - 大端：字数据的高字节存储在低地址中，而字数据的低字节则存放在高地址中
  - 32bit宽的数0x12345678在Little-endian模式CPU内存中的存放方式（假设从地址0x4000开始存放）为：

内存地址	**0x4000**	**0x4001**	**0x4002**	**0x4003**
存放内容	0x78	0x56	0x34	0x12

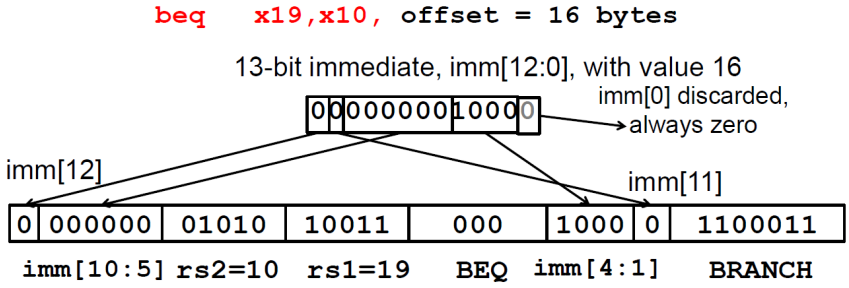
- multithreading can not increase the performance of a specific computation
- strlen不算字符串末尾的'\0'，sizeof算
- Program Space
  - The Stack
    - local variable storage, parameters (declared inside procedure), return address, grows downward
  - The Heap (dynamic malloc storage): data lives until deallocated by programmer, grows upward
  - Static storage : global variable storage (declared outside procedure) , basically permanent, entire program run
  - code
  - 常量constant可能位于：code、static、stack
  - string literal可能位于：static、stack
- Addressing



    - base addressing - load store jalr
    - PC-relative - beq, jal, j(or Pseudodirect addressing)
    - Add can use both immediate(addi) and register(add) addressing
- The unit which acts as **an intermediate agent** between memory and backing store to reduce process time is **Cache**
- 补码的相反数：用周期溢出定义相反数
    - 补码表示下，用周期来减一个数字，就可以得到它的负数 8位表示下
        - 83的负数是-45
- dirty bit is set whenever a page / block is updated (not accessed)
- 数字表示方法
    - one's complement 反码 (+25)10 -> 0011001 and (-25)10 -> (1110 0110)
    - two's complement 补码 (+25)10 -> 0011001 and (-25)10 -> (1110 0111)
    - signed magnitude 原码 (+25)10 -> 0011001 and (-25)10 -> (10011001)
- 流水线stall问题（先执行的是冲突源指令，后执行的是被冲突指令
    - 不允许使用forwarding
        - 使冲突源指令的WB与被冲突指令的ID对齐（默认有RAW特性）
    - 允许使用forwarding：原则是对应流水寄存器级输出->被冲突指令EX输入
        - 冲突源是lw sw
            - 必须插一个气泡(load slot)
    - 如果是control hazard,BEQ作为冲突源指令
        - 使被冲突的指令IF与BEQ的WB对齐（BEQ的MEM过后才会detect the branch）
- L1 Hit time important, L2 miss rate important
- 结构的位置
    - 页表 - 主存Memory(DRAM)中
    - TLB, Cache - 单独的寄存器结构
    - Cache可用SRAM做
    - DRAM = Memory = physical memory space
- Each stage in pipelining should be completed within 1 cycles.
- 超线程处理器 逻辑CPU数>物理CPU
- 死锁：a system state in which no progress is possible

- #entry = #set * associativity
- MISD
  - no use today
- SIMD
  - Data-level parallelism
- MIMD
  - Thread-level parallelism
- drawback of Large capacity
  - DRAM Low speed of operation
  - SRAM High cost
- 提高CPU performance
  - 提高 clock rate
  - 降低 CPI
  - 同时处理多个任务
- 线程级并行
  - 软件线程（任务）在硬件线程（处理器）上分配执行
  - 逻辑线程
- 一个程序的指令数在同一指令集下是一样的
- **指令格式**
  - lw rd, imm(rs1)
  - add rd, rs1, rs2
  - addi rd, rs1, imm
  - sw rs2, imm(rs1)
  - slli rd, rs1, imm
  - beq rs1, rs2, Label
- block = cacheline = entry
- Increase Associativity
  - 降低了miss rate
  - 但增加了miss penalty、访问时间
- Increase cache block size
  - block 数量减少
  - miss rate和access time先减后增
  - 增加了miss penalty（持续升高）
- Increase Cache size
  - 访问时间 access time 增加
- AMAT = hit time + miss penalty * miss rate
- j label (J-Format)
  - rd = x0
  - PC = PC + offset
- jal rd, label (J-Format)
  - rd = PC + 4
  - PC = PC + Offset
  - 存入指令时，要discard掉offset的最后一位（必是0）（riscv是半字寻址）

-
```
Loop:
    beq x19,x10,End
    add x18,x18,x10
    addi x19,x19, 1
    j Loop
End: # target instruction
```

  - Branch offset = 4 × 32 bit instructions = 16 bytes = 8 × 2 bytes
  - 

- jalr rd, rs, imm (I-Format)
  - rd = PC + 4
  - PC = rs + imm
- lui x1, <hi20bits> (U-Format)
  - load upper imm (high 20 bits)
  - LUI x10, 0xDEADB ; x10 = 0xDEADB000
  - 注意用addi填低位时由于符号扩展产生的误差
- auipc x1, <hi20bits> (U-Format)
  - x1 = PC + <hi20bits>
  - add upper imm to PC (high 20 bits)
- range of 32-bit instructions that can be reached from the current PC using a B-Format branch instruction $[-2^{10}, 2^{10} - 1]$
- range of 32-bit instructions that can be reached from the current PC using a J-Format branch instruction $[-2^{18}, 2^{18} - 1]$
- Numbers hardware understands

Human-friendly symbolic names in assembly code

Register	ABI Name	Description	Saver
x0	zero	Hard-wired zero	—
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5	t0	Temporary/alternate link register	Caller
x6–7	t1–2	Temporaries	Caller
x8	s0/fp	Saved register/frame pointer	Callee
x9	s1	Saved register	Callee
x10–11	a0–1	Function arguments/return values	Caller
x12–17	a2–7	Function arguments	Caller
x18–27	s2–11	Saved registers	Callee
x28–31	t3–6	Temporaries	Caller

-

$$(-1)^S \times (1+\text{Fraction}) \times 2^{E-127}$$

指数E	尾数F	表示的值
0	0	0
0	Nonzero	±denormalized number
1~254	Anything	±normalized number
255	0	±infinity
255	Nonzero	NaN （Not a number）

IEEE754 encoding of floating point numbers

- 原子操作
- SMP : symmetric multiprocessor
  - all computers are SMP
- Hazard
  - compulsory
    - increase block size
  - capacity
    - increase cache size
  - conflict
    - increase cache size, increase associativity, 改进替换策略
  - coherence (communication) （多核处理器可能出现）
    - 保持cache的一致性
- 多核处理器内cache的状态
  - Shared : up to date data, other caches may have a copy
  - Modified : up to date data, changed (dirty), no other cache has a copy, OK to write, memory out of date (i.e., write back)
  - Exclusive : up to date data, no other cache has a copy, OK to write, memory up to date
  - Owned : up to date data, other caches may have a copy (they must be in Shared state)
  - Invalid
	M	O	E	S	I
M	✗	✗	✗	✗	✓
O	✗	✗	✗	✓	✓
E	✗	✗	✗	✗	✓
S	✗	✓	✗	✓	✓
I	✓	✓	✓	✓	✓
- 写策略常见搭配：
  - write through + no write allocate
  - write back + write allocate
  - 四种写策略的含义