Mid-Review

LC3 basics,

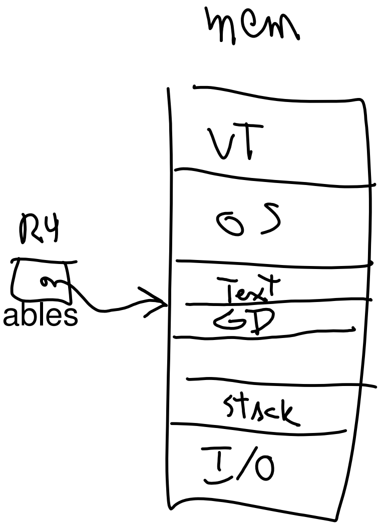
- architecture, ISA

- I/O programming
    - polling
    - interrupt/exception basics
      state+restart, OS entry, context switch

- C+Assembly
    - basic translation (if-then, while, variable access)
    - call frames, local variables, arguments, return values, return addresses
    - Calling .asm from C
        - passing args
        - returning values
        - C-callable wrappers for TRAP routines
    - OS service structure
        - low-level services, higher-level services
        - interrupt/request service pairs

- Linking
    - object files, headers, symbol tables
    - relocation, libraries
    - static versus dynamic linking
    - memory maps, global data, function pointers, pointer variables

--



Performance

- -- Measures
  - -- avg, best, worst, actual cases
  - -- latency, throughput, response
  - -- Time, wall clock, OS+user, user cpu
  - -- Energy/power

$$f_{max} = \propto V$$

$$E = \frac{1}{2} C V^2 \qquad P = \frac{1}{2} C V^2 CR$$

- -- Basic performance equation

$$T_{cpu} = \frac{\#\text{cycles}}{\text{Prog}} \left( \frac{\text{secs}}{\text{cylce}} \right)$$

$$= IC\ \overline{CPI}\ \left( 1/CR \right)$$

$$= \left( \sum IC_i\ \overline{CPI_i} \right) \left( 1/CR \right)$$

- -- Speedup

$$S = V_{new} / V_{old} = \frac{W_{new} / T_{new}}{W_{old} / T_{old}}$$

- -- Amdahl's law

$$S = \frac{1}{(1-f) + f/S_p}$$

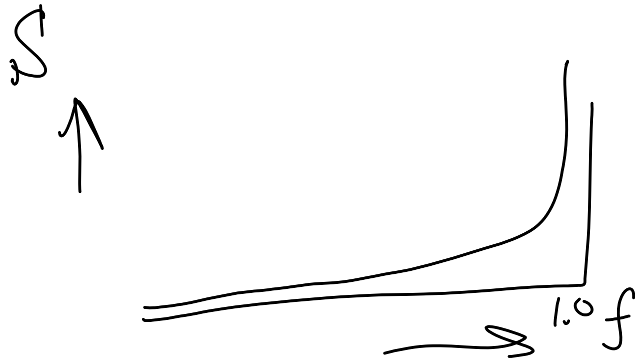
given $T = T_s + T_p = W_s/V_s + W_p/V_p$

$$W = W_s + W_p$$

$$= (1-f)W + fW$$

$$V_s^{old} = V_p^{old} = V_s^{new}$$

$$V_p^{new} = S_p V_p^{old}$$



Performance (continued)

-- Benchmarks
    -- averaging performance (speedup comparisons, GM)
    -- absolute performance comparisons (speedup)
    -- job mix/size dependency

-- Instruction counts
    -- tracing
    -- averaging CPI by classes

Parallelism Principles

-- Pipelining
    -- cut set principle
    -- register setup time, clock skew
    -- CR speedup

-- Interleaving
    -- hiding latencies
    -- banking, duplication

-- Redundancy
    -- Common case
    -- Duplication
    -- heterogenous, multiple different units

-- Fault Tolerance
    -- Error correction (codes, duplicated functional units)
    -- Duplication for faulty unit replacement

## Costs

- -- Chip cost curves w/ time

- -- Silicon
  - -- wafers, dice, testing, yield
  - -- fixed cost overhead, mask sets
  - -- customization versus reconfigurability

- -- Energy
  - -- dynamic power
  - -- static power

## Caches

- -- Locality
  - -- spatial
  - -- temporal

- -- Latency
  - -- hiding
  - -- interleaving
  - -- reordering
  - -- dataflow

- -- Performance tradeoffs
  - -- total size
  - -- block size
  - -- associativity
  - -- levels
  - -- complexity
  - -- splitting, banking, pipeling

- -- Types
  - -- DM
  - -- FA
  - -- SA
  - -- addressing
  - -- tag/index/offset bits
  - -- replacement methods (LRU, random)

$$T = T_{hit} + MR \cdot T_{penalty}$$

added time

$$\downarrow$$

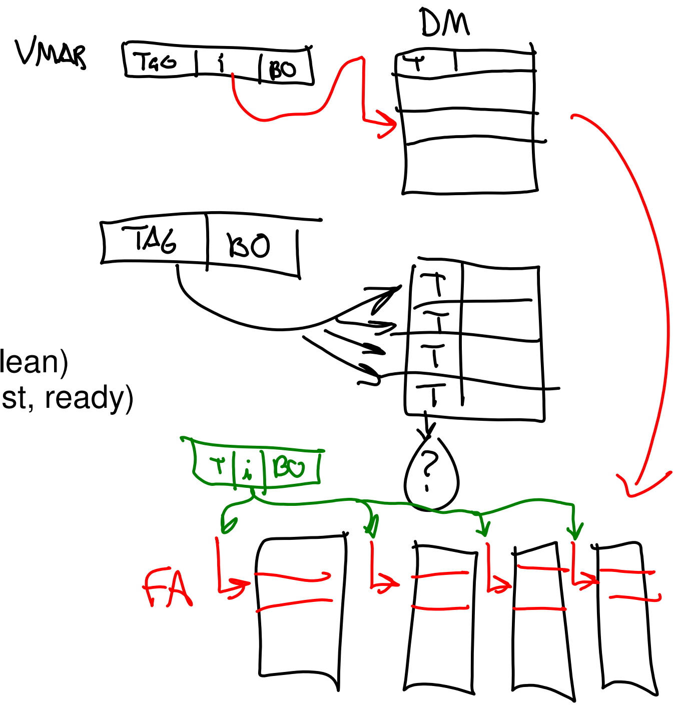
$$\boxed{(T_{hit} + MR \cdot T_p)}$$

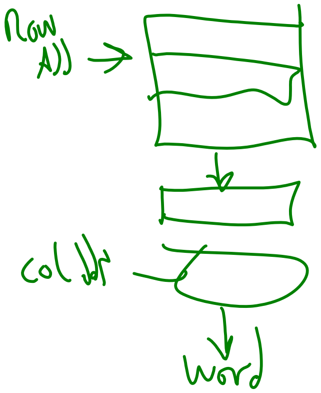
Caches (continued)

- -- Write misses
  - -- write-through/write-back
  - -- write buffers, victim caches, stalling
  - -- allocation, fetch/no-fetch

- -- Controllers
  - -- FSM states (idle, hit, WB/miss-dirty, R/miss-clean)
  - -- control signals (address, data, hit, miss request, ready)

- -- Memory technology
  - -- DRAM, refresh
  - -- SRAM, speed + power
  - -- Row access, row address
  - -- Col access, col address
  - -- bit planes
  - -- word size, block size
  - -- modes: overlapped, pipelined, page mode, burst
  - -- latency versus bandwidth
    - -- CPU gap



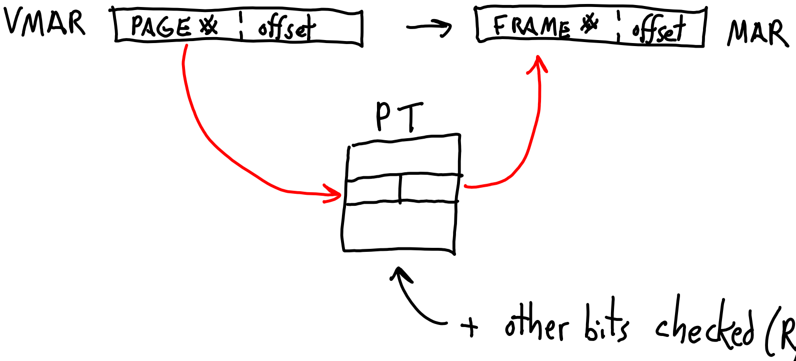


Virtual Memory

- -- Motivation
  - -- large address space
  - -- protection
  - -- interleaving (multiple programs, time sharing)

- -- Disk as memory, memory as cache
  - -- T_penalty large
  - -- reducing misses
    - -- page size
    - -- FA

- -- Address mapping
  - -- basics

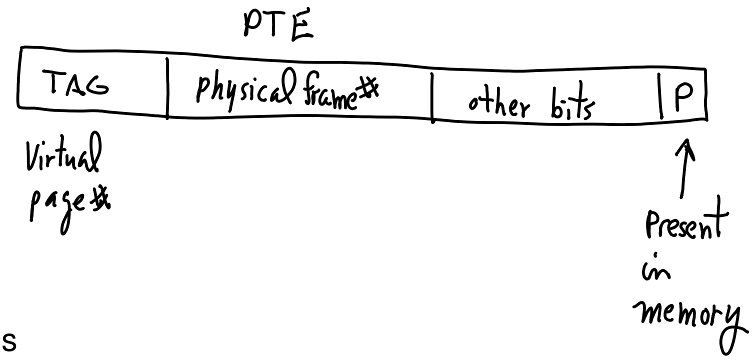


+ other bits checked (R/W/X, user,...)

Virtual Memory (continued)

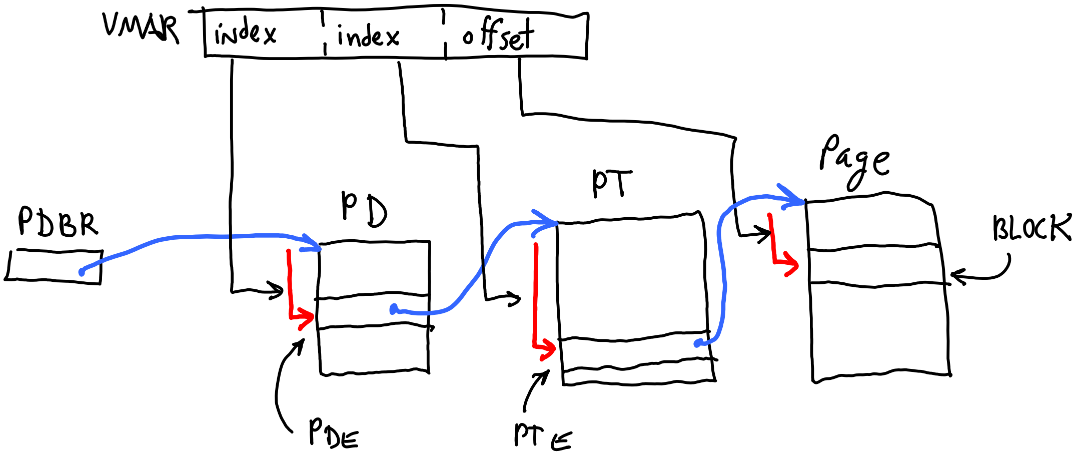
- Page faults
  - disk mapping
  - exception handling
  - replacement policy
  - aligned/non-aligned accesses
  - DMA
    - operation cycle, start, stop, data path, bus arbitration

- Caching translations
  - TLB speed versus size
  - TLB data and PTEs
  - TLB tags (number of bits)
  - TLB misses
    - PT location
    - exception handling, restart



- TLBs and caches
  - critical path and CR
  - physical versus virtual cache tags
  - context switches: flushing TLBs, caches
  - PIDs and slow flushing

- Page sizes, page table sizes, and address space size
  - Page table size versus page size
  - Page size tradeoffs
    - fragmentation
    - page fault overhead (loading versus disk addressing)
    - multi-level paging


VMAR
index
index
offset
PDBR
PD
PT
Page
BLOCK
PDE
PTE