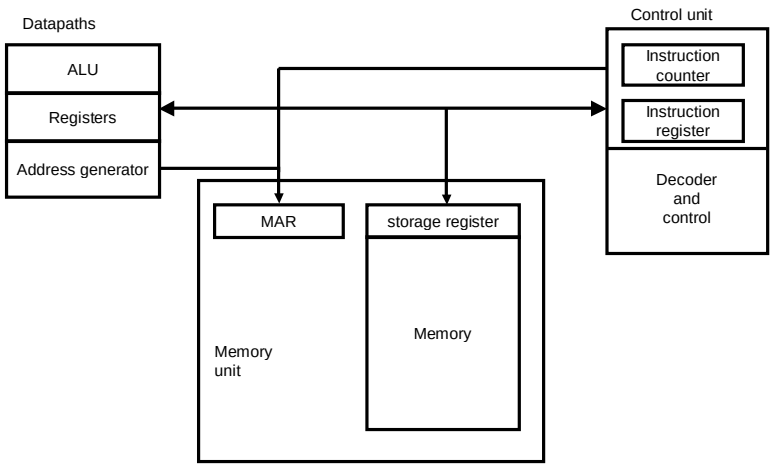
Processor Design
–
Introduction, part II

Professor Jari Nurmi
Institute of Digital and Computer Systems
Tampere University of Technology, Finland
email jari.nurmi@tut.fi

Datapath – Control – Memory

- The basic units of a processor



Explicit vs. Implicit Registers

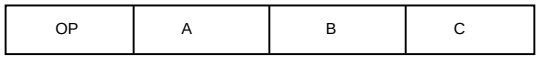
- Explicit registers
  - general purpose registers
  - accumulators
  - address registers (index, base, etc.)
- Implicit registers
  - PC
  - Instruction Register
  - MAR
  - memory buffer register
  - special use registers – usage depending on instruction

Instruction Decode

- Converts instructions (in IR) to control signals
- Different types
  - Hardwired
  - Microprogrammed

Instruction – Microinstruction

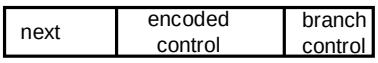
- Instruction is a bit vector partitioned into fields, identifying properties or actions, e.g.
  - format
  - object address
  - operation (”OP”)
  - sequence control

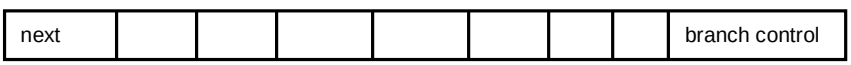


- Instructions are stored in memory, fetched and executed in a sequential manner (”stored program concept”)
- Microinstructions are a part of the implementation, describing similarly the steps of an instruction

Horizontal vs. Vertical

- Especially in the context of microcode
- Also referred to as minimal vs. maximal encoding

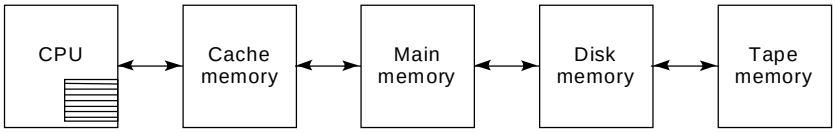




expanded control signal values (fields)

The Memory Hierarchy

- Modern computers have a hierarchy of memories
  - Allows tradeoffs of speed/cost/volatility/size, etc.
  - Illusion of large AND fast memory
- CPU sees common view of levels of the hierarchy.



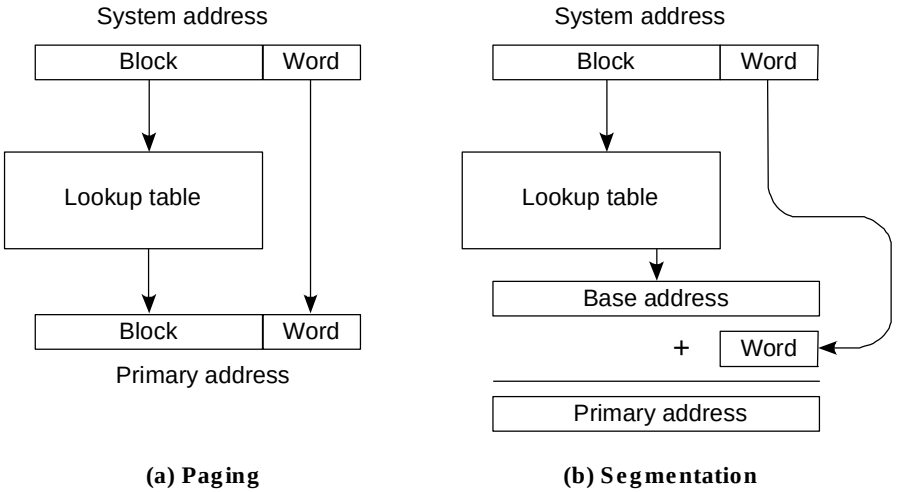
Memory Hierarchy (cont'd)

- Cache types
  - By mapping
    - direct mapped
    - set-associative
    - fully associative
  - By write functioning
    - write-back
    - write-through
  - By replacement algorithms
  - By allocate-on-write
  - By use
    - unified
    - data only
    - instructions only

Process vs. System Addresses

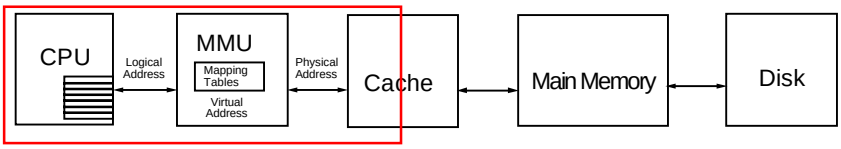
- Process addresses
  - User writes program assuming user is the only occupier of memory
  - System relocates and protects multiple user programs in a common memory space
  - Memory management loads only those ”localities” or pages to the main memory that are expected to be needed shortly
  - The most active localities are loaded into a cache by hardware
- System addresses
  - Some sort of segmentation or paging is needed for the memory to allow allocation of memory to several user (and system) processes

Paging and Segmentation



Virtual Memory

CPU Chip



Some addressing terminology:

- **Logical address — an address issued by by the processor while executing a user program (process).**
- **Virtual address — the address generated from the logical address by the memory management unit, MMU.**
- **Physical address — the address presented to the memory unit.**

(Note: *Every* address reference must be translated in the VM system.)

Page Table and TLB

- Page table is used for translating virtual address to physical address
  - virtual page number (part of the virtual address) is the address
  - the page table entry contains the corresponding physical address and other information (e.g. on the page access rights and presence)
  - slow operation!
- TLB (Translation Lookaside Buffer) is used for speeding up the translation process
  - a cache of previous translations
  - contains the virtual page number (as a tag), the translation, etc.
- Usually the lower end of virtual address = offset on the physical page as well

Instruction Timing

- Only a small portion of instruction processing time is consumed in execution
- All these times have to be taken into account
  - Instruction fetch (with possible translations and misses)
  - Decode
  - Operand fetch (if not load-store)
  - Execution
  - Calculation of next address in branches
  - Writeback phase
- Here we come to clocking and pipelining issues

Pipelining

- Dividing the operation of the instruction to separated stages, enabling a new instruction to start before the previous is completed
- The design task is to minimize program execution time (not cycle time or CPI)
- The execution time depends on the others
  - remember: program execution time $T_{ex} = N \times CPI \times t_{clk}$

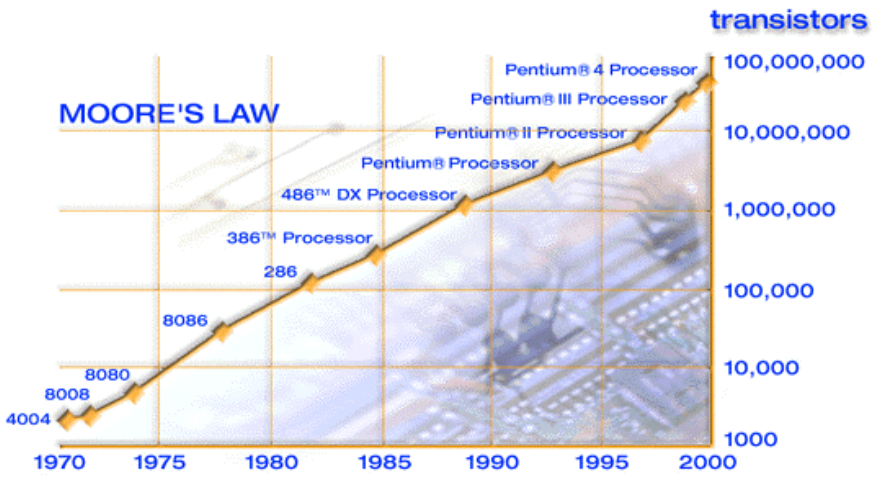
Clocking

- Cycle time is the sum of
  - operation time (slowest stage)
  - clocking overhead (setup and hold times)
  - clock skew

Cost – Area

- Cost is very important in all type of processor systems
  - computer processors (least sensitive to cost)
  - embedded microprocessors
  - DSPs
  - cores integrated in System-on-Chip
- In integrated systems the cost has two major factors
  - non-recurring (fixed) costs
  - incremental cost from fabrication, assembly and testing of chips
- The incremental circuit cost depends mainly on two things
  - silicon area
  - number of I/O
- The cost is not a linear function of silicon area due to yield

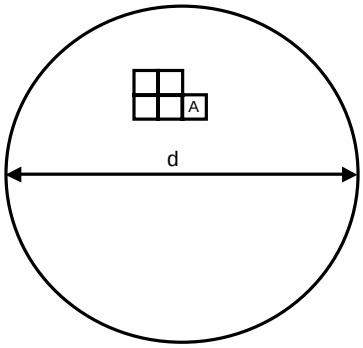
Moore's Law



Source: Intel

Wafers and Chips

- Consider a wafer of diameter ***d*** and chips on that each with area ***A***
- The number of chips is roughly calculated as $N = (\pi d^2 / 4) / A$
  (the ratio of wafer area and A)
- Yield is the proportion of good dies of the total number of dies
  $Y = N_g / N = e^{-\rho_D A}$
- ρ_D is the defect density, typically in the range of 1 / cm^2
- Yield decreases exponentially when increasing area
- Moreover, more chips lost at the edges when chip size increases (consider A → wafer area !)



Economics of Integration

- What is being integrated is based BOTH on technical and economic issues
- Comparison of 1, 2, ... chip solutions taking into account the yield (area allocation and optimization)
  - e.g. how much of L1 cache is integrated with the CPU
- Moore's law allows more and more functionality to be integrated
- Microprocessor generations take advantage of the developing technologies by shrinking the same functionality chip into smaller dimensions
  - lower fabrication costs
  - higher speed
  - less power consumption

Thank you!