GRE Architecture Session

Session 2: Saturday 23, 1995

Young H. Cho
e-mail: youngc@cs.berkeley.edu
www: http://http.cs.berkeley/~youngc

Review

- *Homework*
- *Basic Gate Arithmetics*
- *Bubble Pushing*
- *Logic Design*
- *Complex Digital Circuits*

Logic Design & Complex Circuits

- *Karnaugh-Map “K-MAP”*
- *Design with 4+ Variables*
- *Debounced Flip-Flop (D-FF)*
- *Culmination of small design concept*

Computer

- *What is your idea?*
- *RISC (Reduced Instruction Set Computer)*
- *CISC (Complex Instruction Set Computer)*
- *RISC versus CISC - Robot arm Analogy*

Computer - Parts

- *Control*
- *Datapath*
- *Memory*
- *Input*
- *Output*

Interconnections

- *BUS*
  - *MBus*
  - *Mem Bus*
  - *SCSI*
  - *External Bus*
- *Network*
  - *LAN (Local Area Network)*
  - *WAN (Wide Area Network)*

Memory Hierarchy

- Registers
- Cache
- Main Memory
- Disk
- Tape

Performance

- 1 / Execution Time
- CPI = Cycles per program/Instruction Count per Program
- MIPS = Instruction Count / (Time X 10^6) = Clock rate / (CPI X 10^6)
- MFLOPS = Floating point operation / (Time X 10^6)
- Benchmarks

Benchmarks

- *SPEC - System Performance Evaluation Cooperative*
- *CPU time = (Instr / Prog) X (Cycles / Instruction) X (Sec / Cycle)*
- *Amdahl's Law*
  - *Speedup(w/Enhancement) = (Exec Time without Enh) / (Exec Time with Enh)*

Instruction Set Architecture (ISA)

- *Stack*
  - *operands on top of stack*
- *Accumulator*
  - *one operand is implicitly the accumulator*
- *General Purpose Register*
- *Register/Memory*
  - *only explicit operands - either memory or registers*
  - *acess memory as part of any instruction*
- *Load/Store*

ISA History

- *Single Accumulator*
  - *EDSAC: 1950*
- *Accumulator + Index Registers*
  - *Manchester Mark I, IBM 700 series 1953*
- *Separation of Programming Model from Implementation*
  - *High-level Language Based (B5000: 1963)*
  - *Concept of a Family (IBM 360: 1964)*
- *General Purpose Register Machines*
  - *Complex Instruction Sets (VAX, Intel 432;*

Pipeline

- *Laundry Example*
- *Evident Speedup*
- *Patch things up - compiler tricks and hardware tricks*

Basic Technology

- *Complementary Metal Oxide Silicon (CMOS) transistors*
  - *NMOS*
    - *Turns on when High (Vdd, 5V) is applied*
    - *Turns off when Low (Gnd, 0V) is applied*
    - *(Analogy: Opens the gate at the hill to let the water flow. Gate controlled by stream of water - Water applied to gate controls the gate.)*
  - *PMOS*
    - *Turns off when High (Vdd, 5V) is applied*
    - *Turns on when Low (Gnd, 0V) is applied*

Gate Comparison

- *If PMOS transistor is faster: NOR gate is preferred*
  - *NMOS in Series*
  - *PMOS in Parallel*
  - *H to L is more critical than L to H*
- *If NMOS transistor is faster: NAND gate is preferred*
  - *NMOS in Parallel*
  - *PMOS in Series*
  - *L to H is more critical than H to L*

Summary

- Performance Measurements
- Instruction Set Architecture:
  - Stack
  - Accumulator
  - General Purpose Register:
  - Register/Memory
  - Load/Store

Outline

- Performance
- ISA
- Technology
- ALU
- Computer Arithmetics
  - Binary Arithmetics
  - Floating point Arithmetics
- Single cycle and pipelines

Common Sense: 5 Basic Components

- *Datapath*
- *Control*
- *Memory*
- *Input*
- *Output*

I. Performance

- *Comments: Important to Keep this mind when evaluating benchmarks*
- *Speedup - Amdahl's Law*
- *Compiler Problem - MIPS and CPI*
- *Equations*

II. ISA

- Comments: Many different Architecture
- Stack
- Accumulator
- General Purpose Register:
- Register/Memory
- Load/Store

III. Technology

- Comments: Real issue in hardware implementations
- CMOS technology
- Internal Delay
- Cycle time

IV. Arithmetic Logic Unit (ALU)

- *Comments: Basic intelligent unit implementations*
- *Carry-Look-Ahead*
- *Carry Select*

V. Computer Arithmetics

- *Binary Standard*
- *Multiplication*
  - *Booths*
- *Floating point (IEEE Standard)*
- *Multiply, Shift, and FP Number*
- *Comments: Important to know the concept*
  - *IEEE 31 / 30-23 / 22-0*
  - *(-1)^s * (1+significand) x 2^(exponent-bias)*

VI. Single Cycle and Pipeline Datapath

- *Datapath*
- *Draw your idea of what Computer is*
- *Block diagram*