Exam 1: Study Guide

1 Famous Architects and Machines

Architect	Contribution
Eckert and Mauchly	First Working Electronic Computer (ENIAC)(1946)
Wilkes	First Store Program Computer (EDSAC I) (1949)
Amdahl	Amdahl's Law
Cray	1st Super Computer
Patterson	RISC Processor Design
Hennessy	MIPS
Rau	Very Long Instruction Word
Smith	Branch Prediction Strategies
Patt	First Complex Logic Gate On Silicon
Hwu	Compiler Optimization
Sohi	Non-blocking Microprocessor Caches

Computer	Claim To Fame
Univac I	First Commercial Machine (1951)
IBM System 360	Concept Of Family Of Machines (1964)
DEC PDP-8	Mini-computer
DEC PDP-11	Unix Developled On This Machine
Cray-1	Super Computer (1976)
Intel 4004	First Microprocessor
Intel 8086	Basic Architecture of IA32 PC
Intel 80486	Pipelined IA32
Pentium	Superscalar IA32
AMD Barcelona	L1, L2, L3 Cache; nm Technology; 1.6 GHz clock

2 Fabrication

Yield

$$\textbf{Cost Per Die} = \frac{\text{Cost Per Wafer}}{\text{Dies Per Wafer} \cdot \text{Yield}}$$

$$\textbf{Wafer Area} = \pi \cdot r^2$$

$$\textbf{Dies Per Wafer} = \frac{\text{Wafer Area}}{\text{Die Area}}$$

$$\textbf{Yield} = \frac{1}{(1 \ + \ (\text{Defects Per Area} \cdot \frac{\text{Die Area}}{2}))^2}$$

Power

$$\textbf{Dynamic Power} = \frac{1}{2} \cdot \text{Capacitive Load} \cdot \text{Voltage}^2 \cdot \text{Clock Rate}$$

3 MIPS Design Decisions

Why is MIPS limited to 32 registers?

1. A very large number of registers may increase the clock cycle time simply because it takes electronic signals longer when they must travel farther.

2. You would have to expand the instruction format to accomadate more bits.

What are the 3 design principles of MIPS

1. Simplicity favors regularity.
2. Smaller is faster.
3. Good design demands good compromises.

Why do words in MIPS have to start at addresses that are multiples of 4?

1. This alignment restriction leads to faster data transfers.
2. It also allows programs to always use lw and sw to access the stack.

Why does MIPS keep all instructions the same length?

1. MIPS follows the design principle that states 'simplicity favors regularity.'

4 Intel x86 Idiosyncrasies

1. Intel x86 only supports 8 general purpose register.
2. The x86 arithmetic and logical instructions must have one operand act as both a source and a destination.
3. One instruction operand can be in memory.
4. x86 instructions are not all 4 bytes in length.
5. Registers have dedicated uses.

5 Amdahl's Law

$$\textbf{Speed Up} = \frac{\text{Old Time}}{\text{New Time}}$$
$$\textbf{Speed Up}_{\textbf{Overall}} = \frac{1}{(1 - \text{F}) + \frac{\text{F}}{\text{S}}}$$

where F is the Percentage to be enchanced and S is the factor it is to be enhanced by

6 CPI and Performance

CPU Time

$$\begin{aligned}\textbf{CPU Time} &= \frac{\text{Instruction Count} \cdot \text{CPI}}{\text{Clock Rate}} \\ &= \frac{\text{Insructions}}{\text{Program}} \cdot \frac{\text{Clock Cycles}}{\text{Instructions}} \cdot \frac{\text{Seconds}}{\text{Clock Cycle}}\end{aligned}$$
$$\textbf{Overall Effective CPI} = \sum_{i=1}^{n} (\text{CPI}_i \cdot \text{IC}_i)$$

MIPS

$$\textbf{MIPS} = \frac{\text{Instruction Count}}{\text{CPU Time} \cdot 10^6}$$

Performance Ratio

$$\frac{\text{performance}_\text{A}}{\text{performance}_\text{B}} = \frac{\text{CPU Time}_\text{A}}{\text{CPU Time}_\text{B}} = \text{n}$$

So A is n times faster than B

SPEC

$$\textbf{SPEC Ratio} = \frac{\text{ReferenceTime}}{\text{ExecutionTime}}$$

7 Counting Gate Delays

Wallace Tree

$$\textbf{Gate Delays} = log_{\frac{3}{2}}\text{N}$$

where N is the number of bits in the multiplier

$$\textbf{Gate Delays}_{\textbf{Booth Recoding}} = log_{\frac{3}{2}}\frac{\text{N}}{2}$$

Ripple Carry Adder

$$\textbf{Gate Delays} = \text{N} \cdot 2$$

where N is the number of bits in the numbers being added

Carry Look Ahead Adder

1. Each FA is like a ripple carry adder and has 1 gate delay each.
2. The first-level and first CLA is 3 gate delays with an additional 2 after each CLA on the same level.
3. Each additional first CLA in a level is another 2 gate delays on top of the 3 for the first-level CLA.

8 Binary Arithmetic

Overflow

Unsigned

1. Overflow occurs when there is a 1 in the carry out.

Signed

1. Overflow occurs when you add 2 positive numbers and get a negative, or when you add 2 negative numbers and get a positive.

Addition

Unsigned

1. The result should have the same amount of bits are the original numbers.

Signed

1. Same as Unsigned except you have to watch out for the MSB being a 1 or 0 indicating a positive or negative number respectively.

Subtraction

Unsigned

1. Change the subtrahend to its 2's complement and add it to the minuend.

Signed

1. Same as Unsigned.

Multiplication

Unsigned

1. If multiplying an N-bit number by an M-bit number, the result should have N+M bits.

Signed

1. Same as Unsigned.
2. If the multiplier is negative take the 2's complement on both numbers and then multiply them.
3. Use Booth's recoding to reduce the number of addition needed to be done by half.
4. If using Booth's recoding and the the multiplier has a -1 in the position add the 2's complement of the multiplicand to the product.
5. If using Booth's recoding and the mulitipler has a -2 in the position find the 2's complement of the multiplicand and shift it left 1 bit and then add it to the product.
6. If using Booth's recoding and the mulitipler has a 2 in the position shift the multiplicand left 1 bit and then add it to the product.

Division

Unsigned

1. Remember when borrowing from the left that the 1 really carries over 2 1's.

Signed

1. Make both numbers positive and then figure out the sign at the end and change the bits in the result to reflect that.