NASA X43A ➜ Reached just over seven times the speed of sound! (shown here being launched from a B52 bomber)

Wow… NY ➜ London < 5 hrs!
Things to Remember

• Virtual memory to Physical Memory Translation too slow?
  • Add a cache of Virtual to Physical Address Translations, called a **TLB**
  • Need more compact representation to reduce memory size cost of simple 1-level page table (especially 32- ⇒ 64-bit address)

• Spatial Locality means Working Set of Pages is all that must be in memory for process to run fairly well

• Virtual Memory allows protected sharing of memory between processes with less swapping to disk
What are “Machine Structures”?

Cooperation of many *levels of abstraction*

We’ll investigate lower abstraction layers! (contract between HW & SW)
Below the Program

• High-level language program (in C)

```c
swap int v[], int k){
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

• Assembly language program (for MIPS)

```assembly
swap: sll $2, $5, 2
add $2, $4,$2
lw  $15, 0($2)
lw  $16, 4($2)
sw  $16, 0($2)
sw  $15, 4($2)
jr  $31
```

• Machine (object) code (for MIPS)

```
000000 00000 00101 0001000010000000
000000 00100 00010 0001000000100000 . . .
```
Logic Design

• Next 2 weeks: we’ll study how a modern processor is built starting with basic logic elements as building blocks.

• Why study logic design?
  • Understand what processors can do fast and what they can’t do fast (avoid slow things if you want your code to run fast!)
  • Background for more detailed hardware courses (CS 150, CS 152)
Logic Gates

• Basic building blocks are logic gates.
  • In the beginning, did ad hoc designs, and then saw patterns repeated, gave names
  • Can build gates with transistors and resistors

• Then found theoretical basis for design
  • Can represent and reason about gates with truth tables and Boolean algebra
  • Assume know truth tables and Boolean algebra from a math or circuits course.
  • Section B.2 in the textbook has a review
Physical Hardware

Let’s look closer…
Gate-level view vs. Block diagram

A | B | C
---|---|---
0  | 0 | 1
0  | 1 | 1
1  | 0 | 1
1  | 1 | 0
Signals and Waveforms: Clocks

\[ T' = \frac{1}{\text{freq.}} \approx 1 \text{ ns} \]
Signals and Waveforms: Adders
Signals and Waveforms: Grouping

\[ X_3 \ X_2 \ X_1 \ X_0 \]

\[ X_0 \]
\[ X_1 \]
\[ X_2 \]
\[ X_3 \]

\[ X \]
\[ 5 \ 6 \ 8 \ 1 \ 7 \]
Signals and Waveforms: Circuit Delay

A = [a₃, a₂, a₁, a₀]
B = [b₃, b₂, b₁, b₀]

A → 4 → C

A → 4 → .photos

A → adder propagation delay
Combinational Logic

• Complex logic blocks are built from basic AND, OR, NOT building blocks we’ll see shortly.

• A *combinational* logic block is one in which the output is a function only of its current input.

• Combinational logic *cannot have memory* (e.g., a register is not a combinational unit).
Circuits with STATE (e.g., register)
Administrivia

• How was your spring break?

• POLICY: All official project spec clarifications will be posted on the project page and the News section of the website. If it’s just mentioned in the newsgroup, it won’t be in our tests
A. SW can peek at HW (past ISA abstraction boundary) for optimizations

B. SW can depend on particular HW implementation of ISA

C. Timing diagrams serve as a critical debugging tool in the EE toolkit
Peer Instruction – Predict NCAAs

1: Duke over GT
2: Duke over OSU
3: UConn over GT
4: Uconn over OSU
5: GT over Duke
6: GT over UConn
7: OSU over Duke
8: OSU over UConn
9: Don’t know/care...
And in conclusion…

- ISA is very important abstraction layer
  - Contract between HW and SW
- Basic building blocks are logic gates
- Clocks control pulse of our circuits
- Voltages are analog, quantized to 0/1
- Circuit delays are fact of life
- Two types
  - Stateless Combinational Logic (&,|,~)
  - State circuits (e.g., registers)