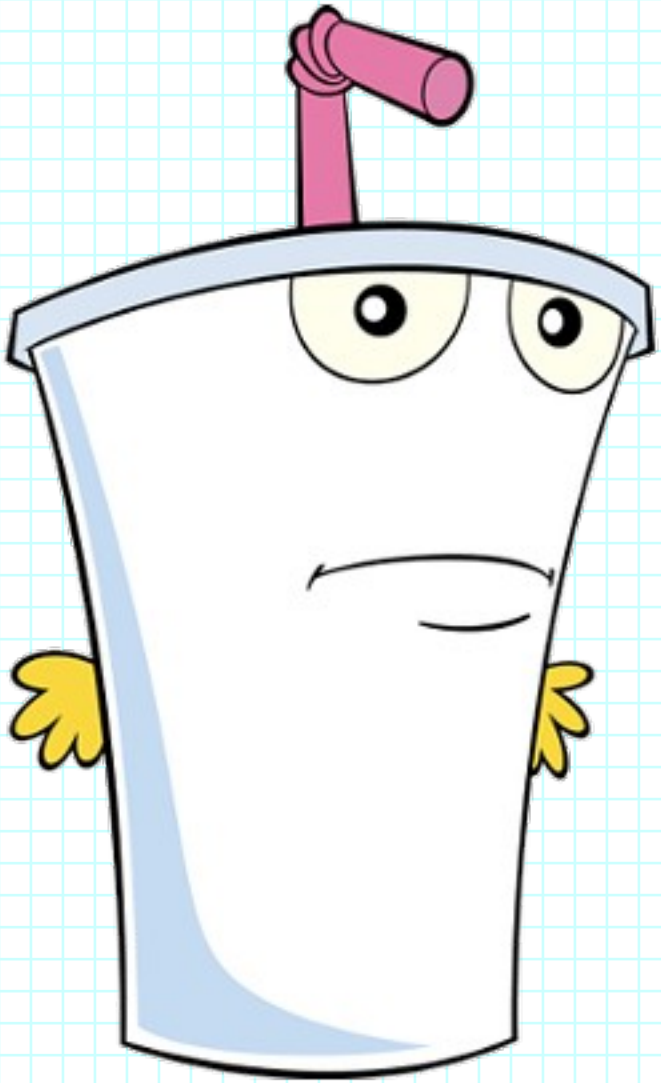
Introduction to Digital Logic Design and Computer Systems

- Course structure and expectations
- Intro to digital circuits

Doug Shook via Jon Turner and Anne Bracy

Hello, world!



Course Website

http://cse.wustl.edu/~dshook/cse260

Grading

- Homework (15%)
- Labs (30%)
- Exams (3 exams, 15% each)
- Zybook (10%)
  - See webpage for info on how to register

What's a Lecture, Anyway?

- Not really a lecture....

- You must come prepared!
  - Bring questions
  - Be ready to answer questions

Textbook

zyBooks

Academic Dishonesty

- Don't cheat!
  - Zero tolerance.
- If you are ever in doubt ASK.

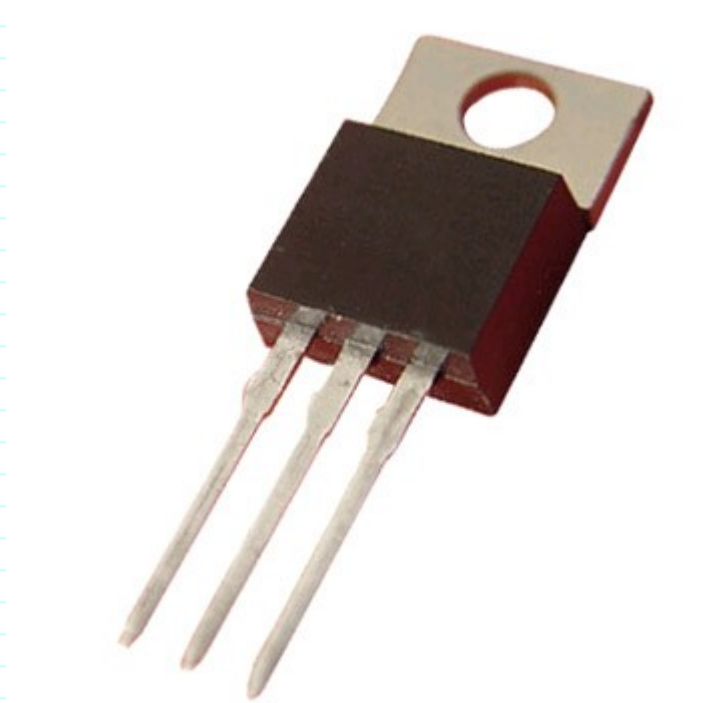
Problem

- We want to compute things
  - What are the essential pieces?
  - What is the simplest computer you can conceive of?
    - Don't worry about speed

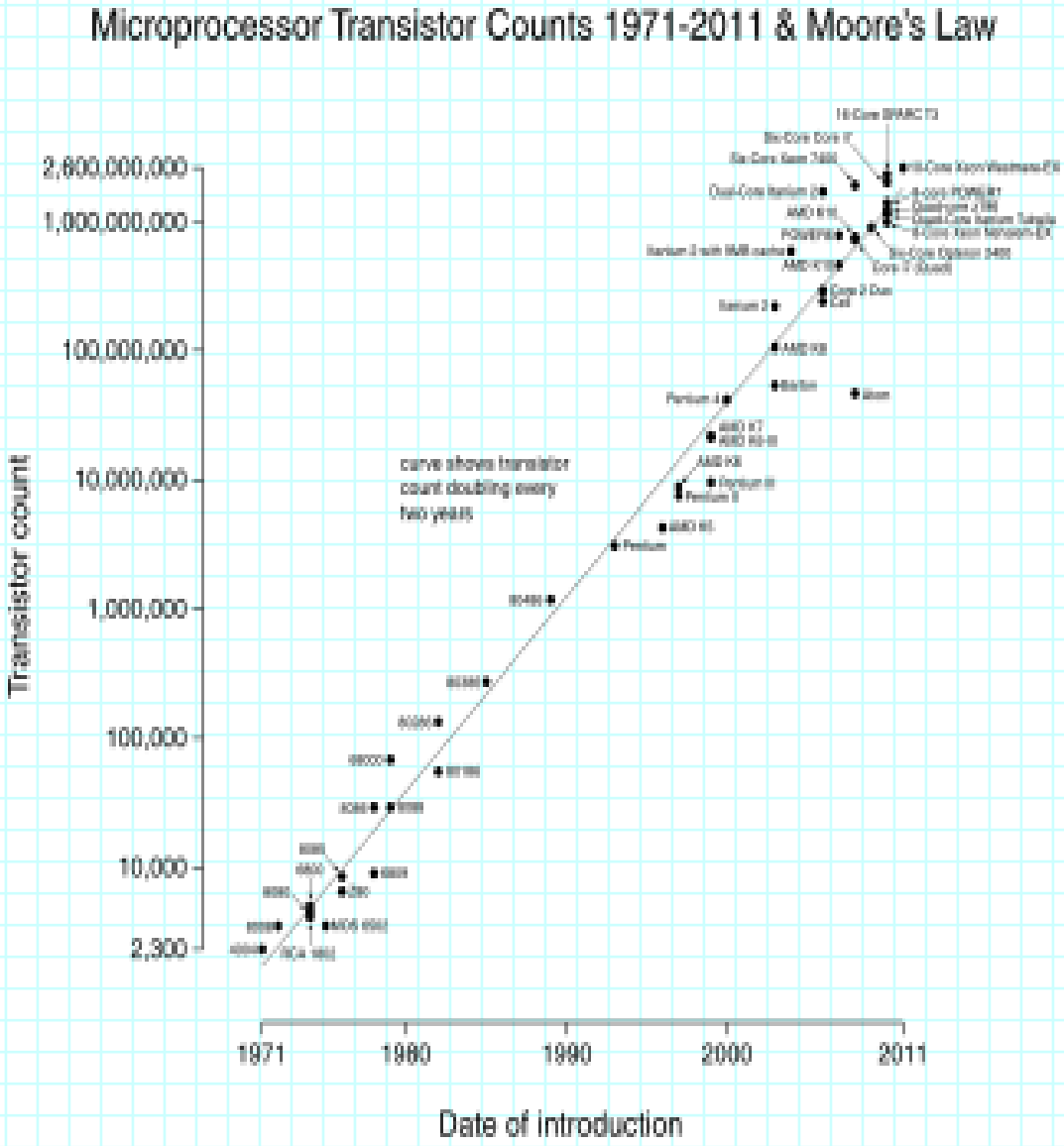
Problem

- Now consider speed: how can we make it go faster?
  - How does this change the essential pieces of your simple machine?

Transistors!



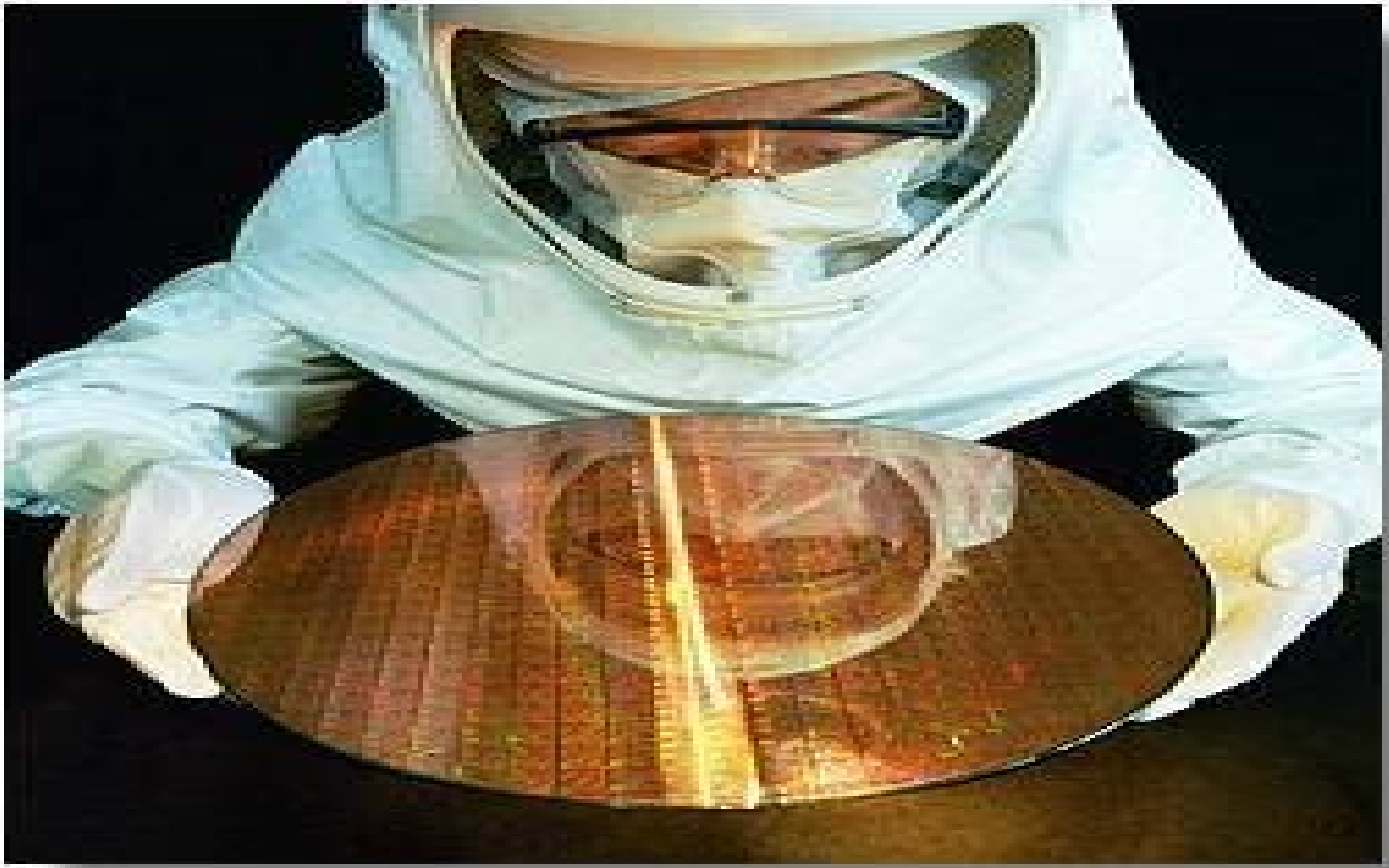
Moore's Law


Microprocessor Transistor Counts 1971-2011 & Moore's Law
Transistor count
2,600,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
2,300
curve shows transistor count doubling every two years
1971
1980
1990
2000
2011
Date of introduction

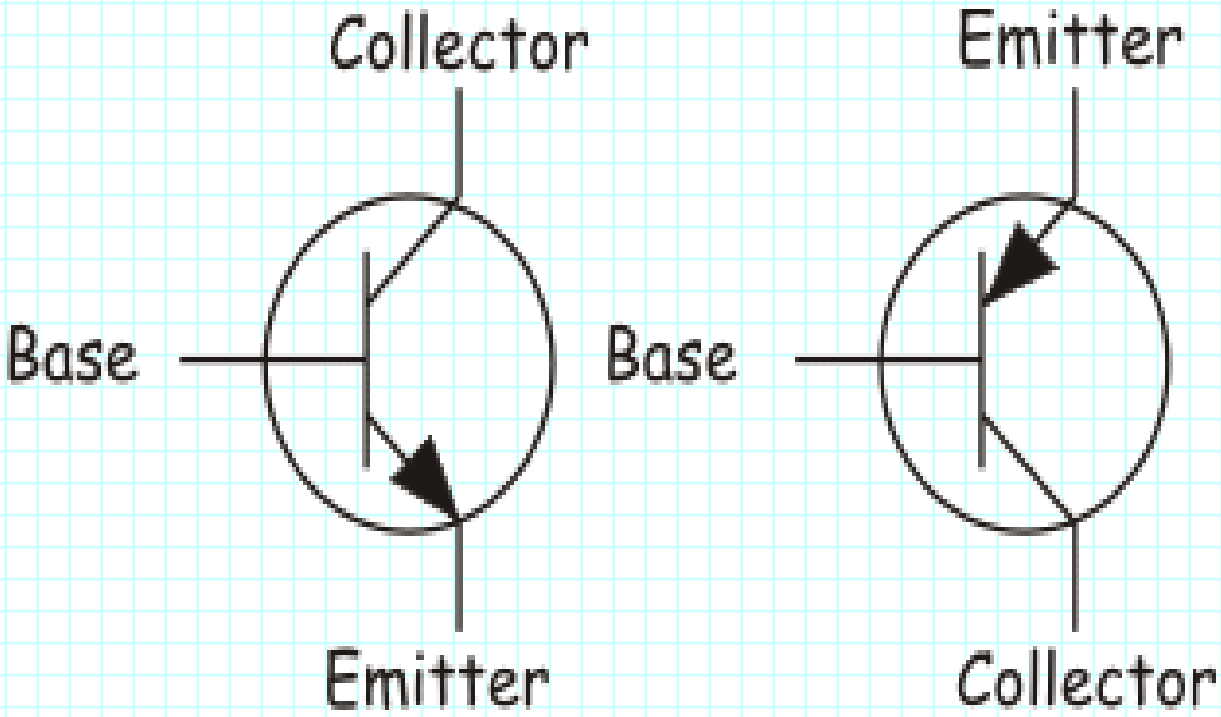
Transistor Fabrication



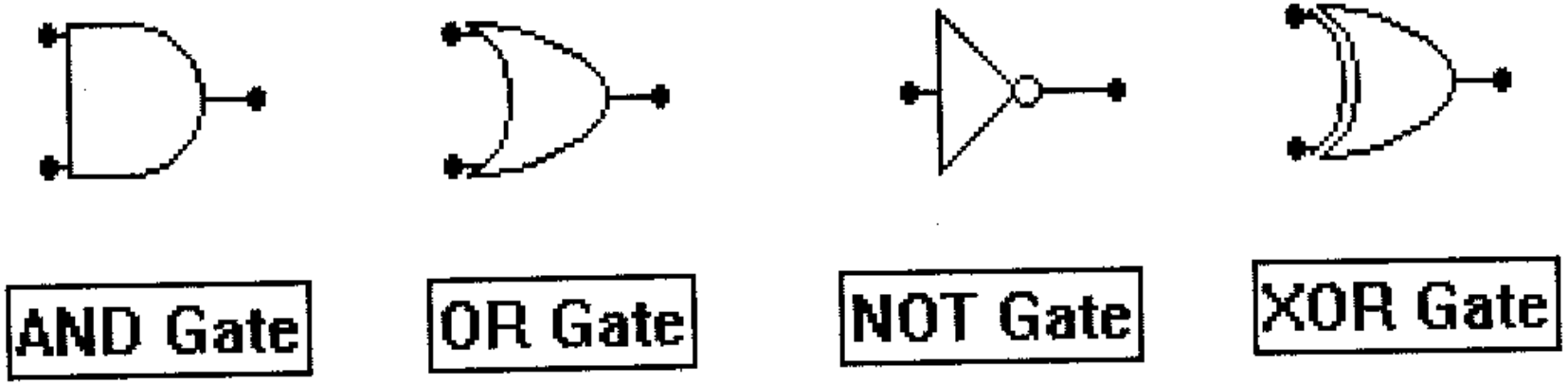
Transistor Fabrication



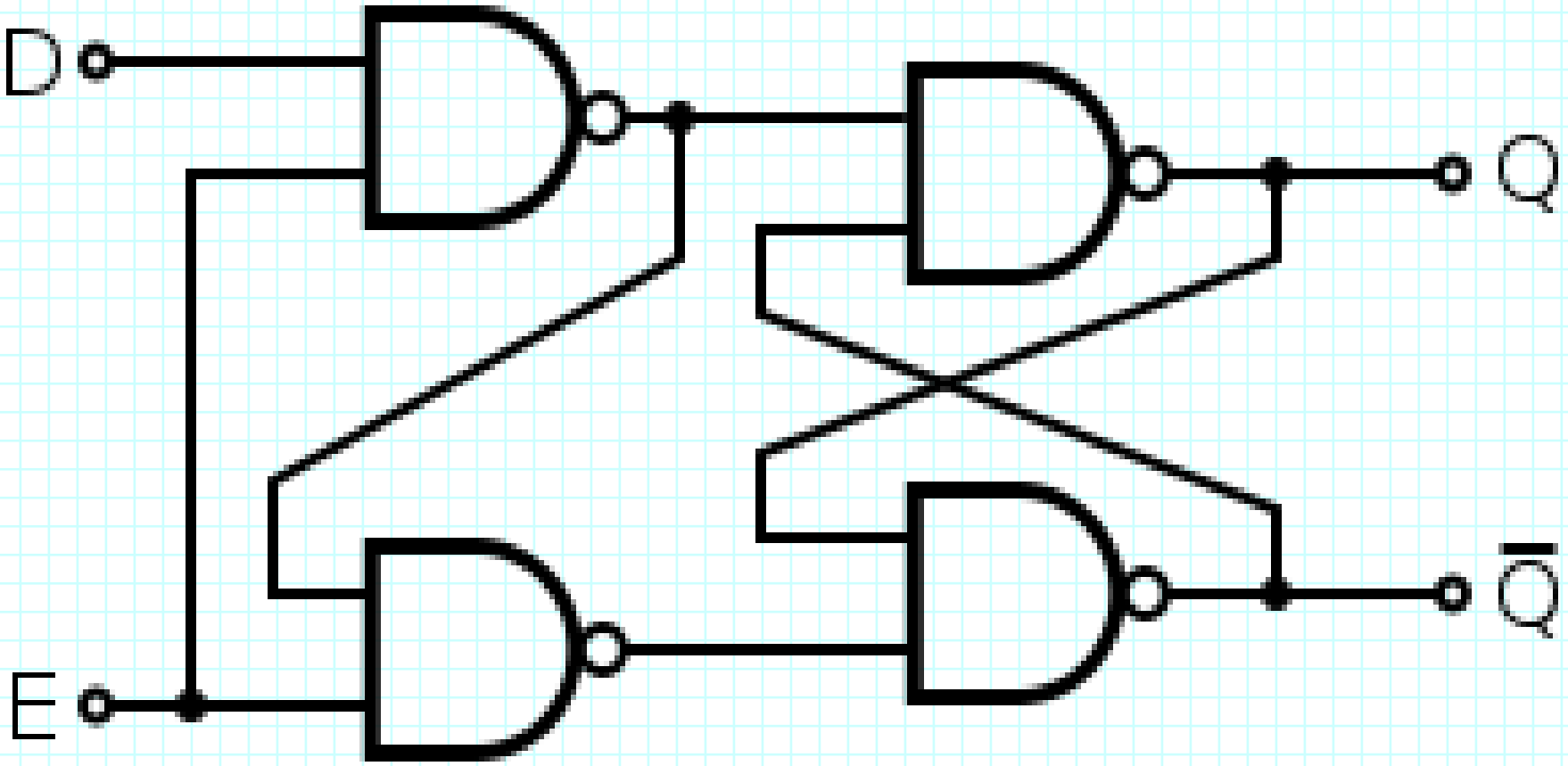
Why are transistors so important?



It is all quite logical.



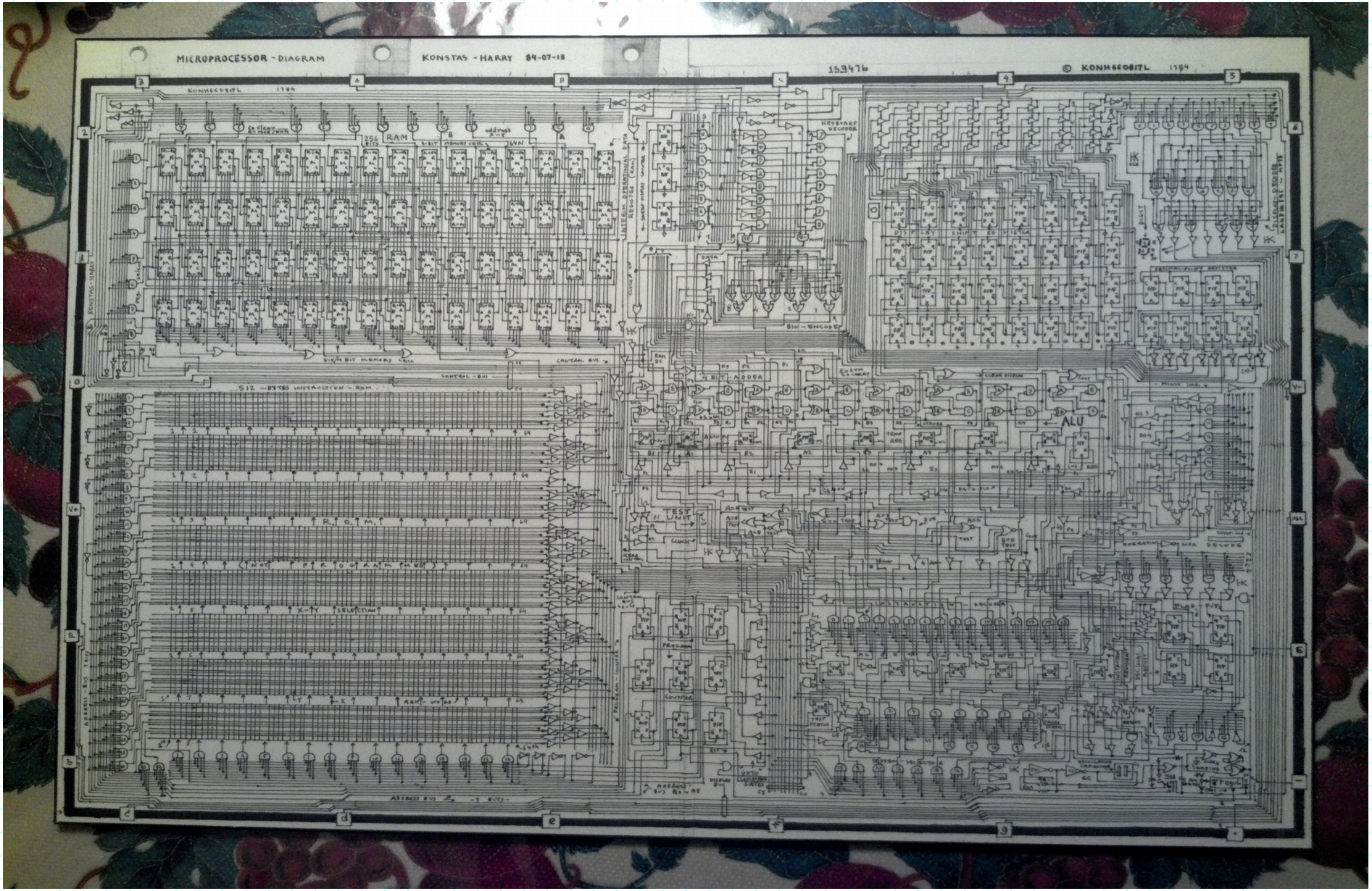
Binary Storage



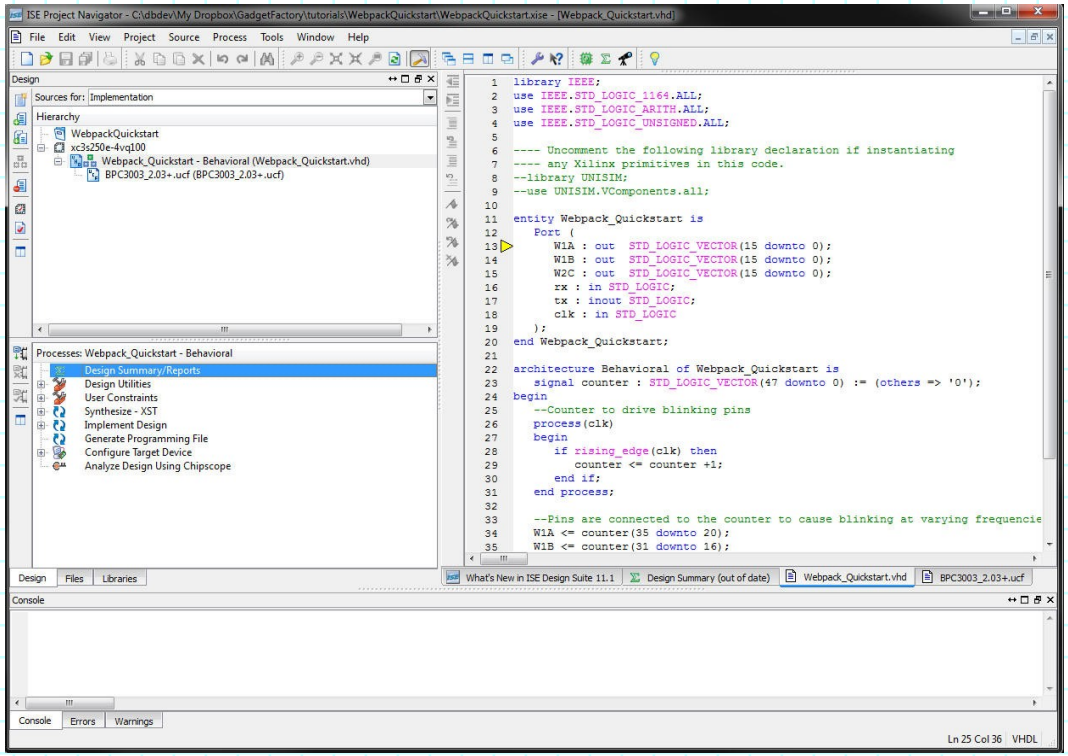
The Bit Three

- Transistors – The workhorse
- Logic – The law of the land
- Binary – The language of our future overlords

Computer Design in the Past



How design is done today



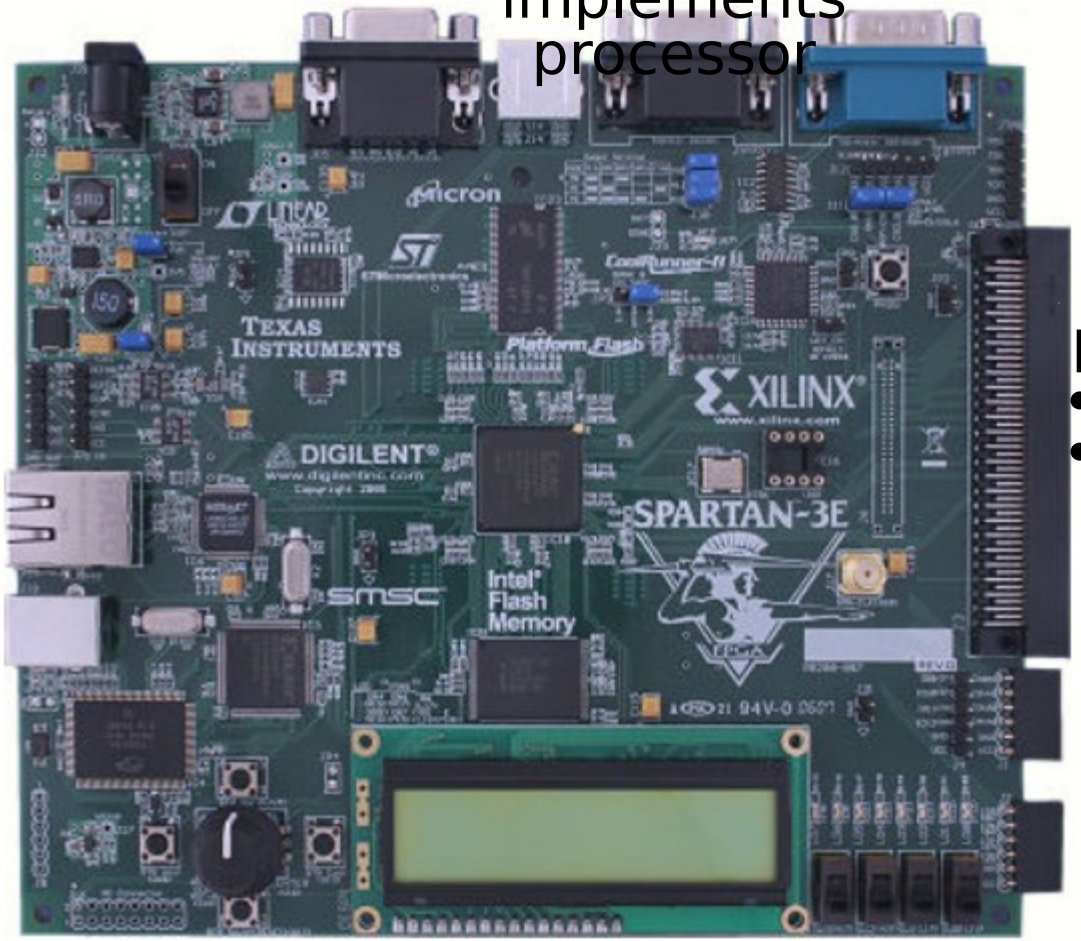


What's an FPGA?

VGA display connector

Field Programmable Gate Array (FPGA) implements processor

Program button configures FPGA



Push buttons
- reset
- single step
- load data into memory

Display showing
- internal registers
- memory locations

Knob for entering data

Slide switches for selecting what to display

So what's the difference?

- CPUs
  - »Unchangable
  - »General Purpose
  - »High cost of development
  - »Millions of transistors
  - »High Speed
- FPGAs
  - »Reconfigurable
  - »As specific as you want
  - »Cheap (relatively)
  - »1000s of transistors
  - »Slower (but still adequate for design purposes)

Still one piece missing

- How do we tell a computer what to do?

Machine language

```
A[300] = h + A[300];      /* $t1 <= base of array A; $s2 <= h */
```



Compiler

```
lw   $t0, 1200($t1)     # temporary register $t0 gets A[300]
add  $t0, $s2, $t0      # temporary register $t0 gets h +A[300]
sw   $t0, 1200($t1)     # stores h + A[300] back into A[300]
```



Assembler

35	9	8	1200		
0	18	8	8	0	32
43	9	8	1200		

100011	01001	01000	0000 0100 1011 0000		
000000	10010	01000	01000	00000	100000
101011	01001	01000	0000 0100 1011 0000		

Talking to Machines

- What kinds of instructions would you want to give a computer?
  - How many instructions would be enough?

So what next?

- Bottom up approach:
  - Part 1: Transistors and combinational logic
  - Part 2: Sequential logic
  - Part 3: Constructing a CPU

For Next Time

- Get the Zybook
  - This is required!
- Do the Zybook sections before next class
  - Listed on the course website
- Come with questions!
  - And come prepared to answer questions
- Find a partner for homework assignments
  - First one is coming very soon