0x19 - Wrap Up

ENGR 3410: Computer Architecture

Jon Tse

Fall 2020

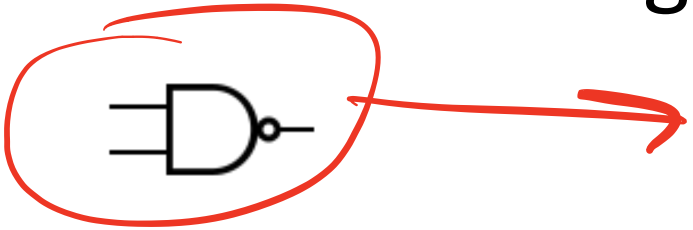
Housekeeping

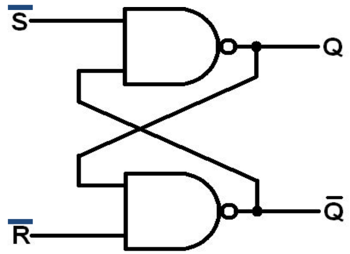
- Write Up Draft due Dec 14, Midnight Eastern
- Presentation Dec 15/17 during “class time”
  - Sign up in the Google Spreadsheet
  - Conflicts/special casing, please email me.
- Final Write Up (if necessary) due Dec 18.

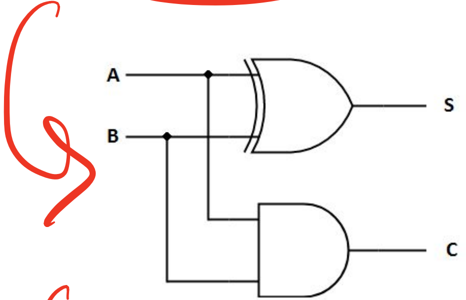
Retrospective

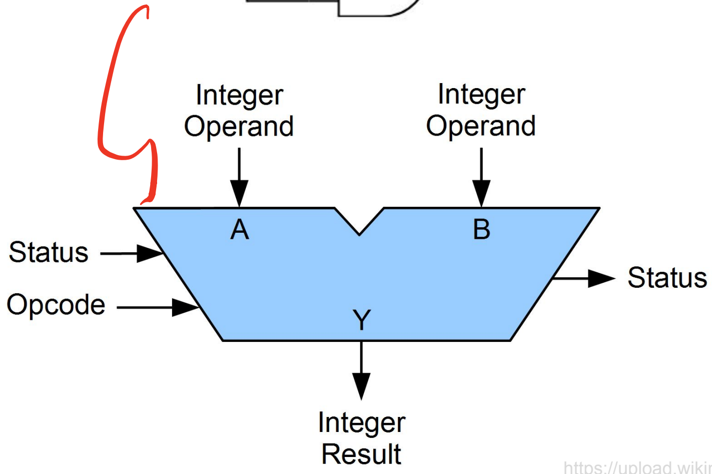
- 3 Classes in 1
  - Digital Logic
  - Computer Organization
  - Computer Architecture
- Bonus
  - Pareto Optimality
  - Heuristic Optimization Methods

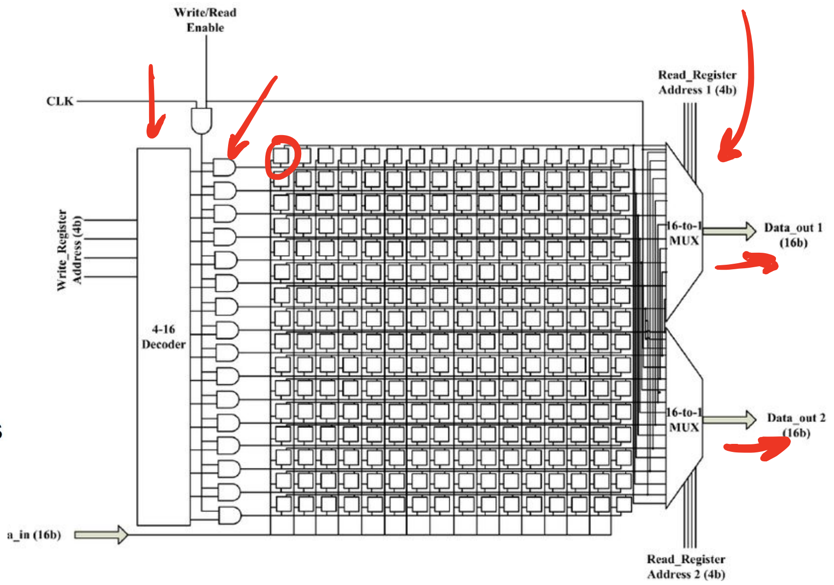
Class 1 - Digital Logic





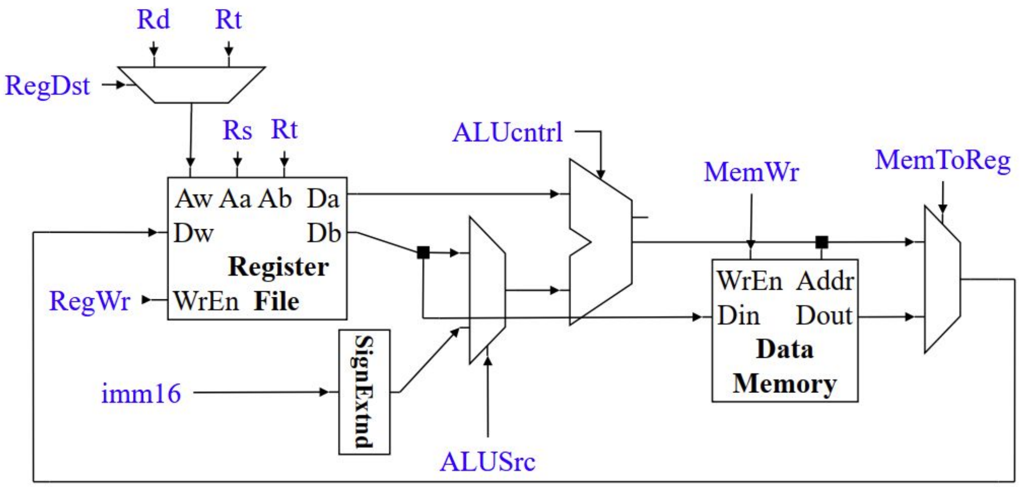


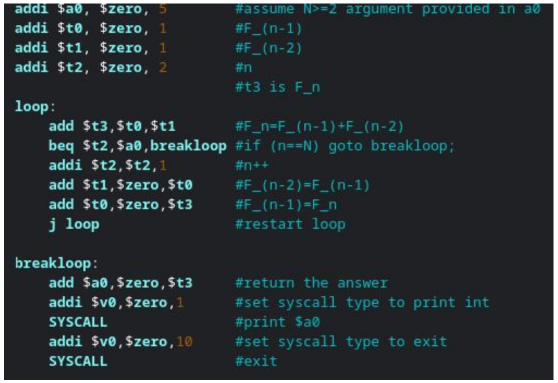




https://upload.wikimedia.org/wikipedia/commons/0/0f/ALU_block.gif
https://www.researchgate.net/figure/Structure-of-16-bit-Register-File_fig22_269818702

Class 2 - Computer Organization

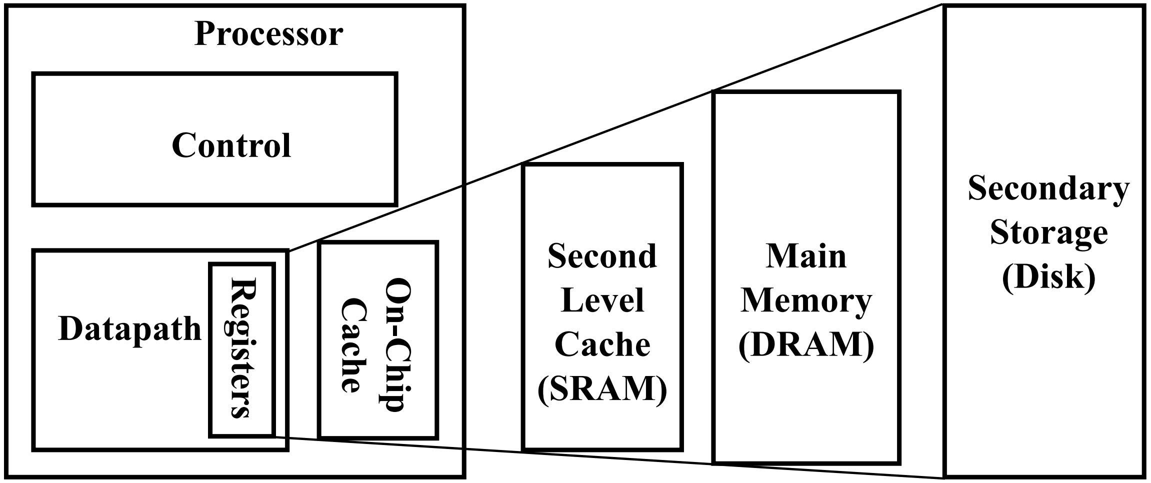


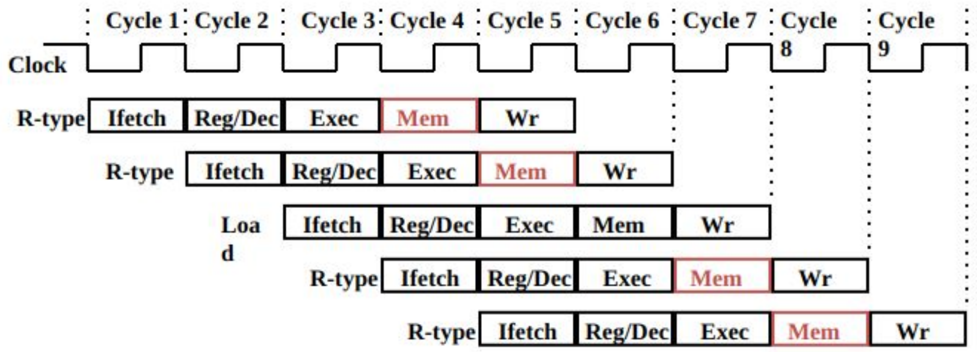


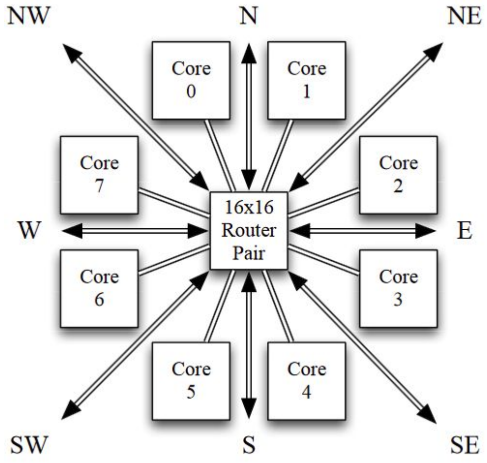
```
addi $a0, $zero, 5          #assume N>=2 argument provided in a0
addi $t0, $zero, 1          #F_(n-1)
addi $t1, $zero, 1          #F_(n-2)
addi $t2, $zero, 2          #n
                            #t3 is F_n
loop:
    add $t3,$t0,$t1         #F_n=F_(n-1)+F_(n-2)
    beq $t2,$a0,breakloop   #if (n==N) goto breakloop;
    addi $t2,$t2,1          #n++
    add $t1,$zero,$t0       #F_(n-2)=F_(n-1)
    add $t0,$zero,$t3       #F_(n-1)=F_n
    j loop                  #restart loop

breakloop:
    add $a0,$zero,$t3       #return the answer
    addi $v0,$zero,1        #set syscall type to print int
    SYSCALL                 #print $a0
    addi $v0,$zero,10       #set syscall type to exit
    SYSCALL                 #exit
```

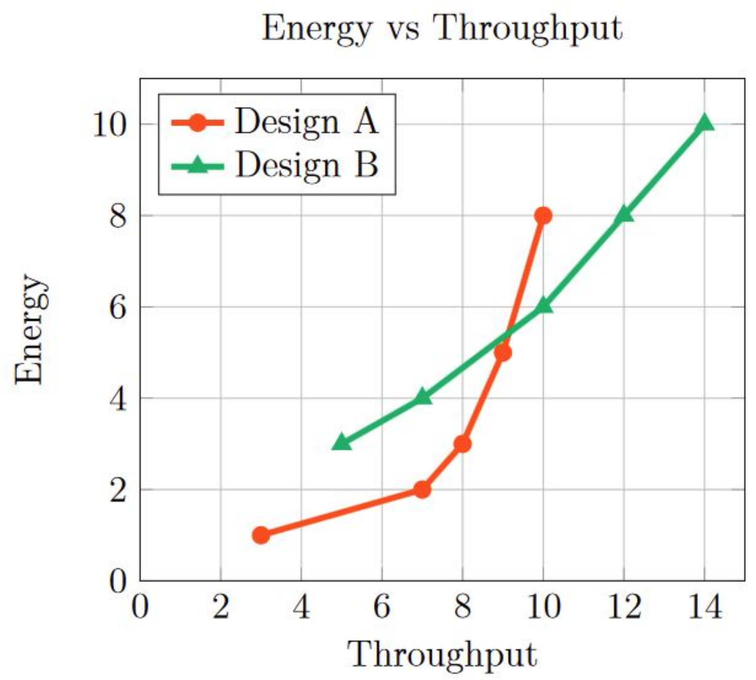
Class 3 - Computer Architecture

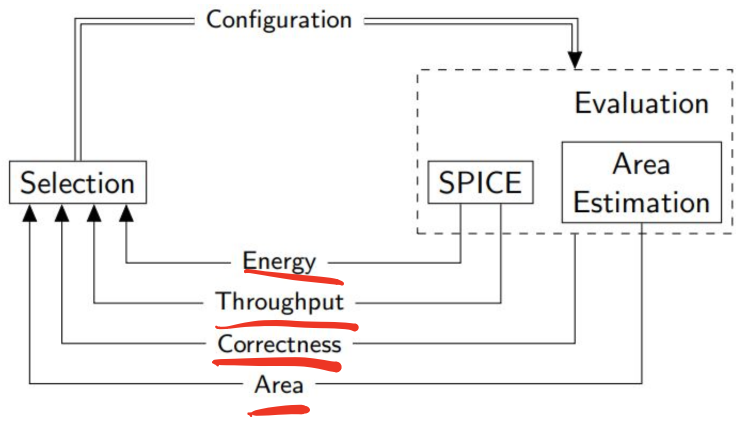



Cycle 1
Cycle 2
Cycle 3
Cycle 4
Cycle 5
Cycle 6
Cycle 7
Cycle 8
Cycle 9
Clock
R-type Ifetch Reg/Dec Exec Mem Wr
R-type Ifetch Reg/Dec Exec Mem Wr
Load Ifetch Reg/Dec Exec Mem Wr
R-type Ifetch Reg/Dec Exec Mem Wr
R-type Ifetch Reg/Dec Exec Mem Wr



Bonus - Pareto and Optimization

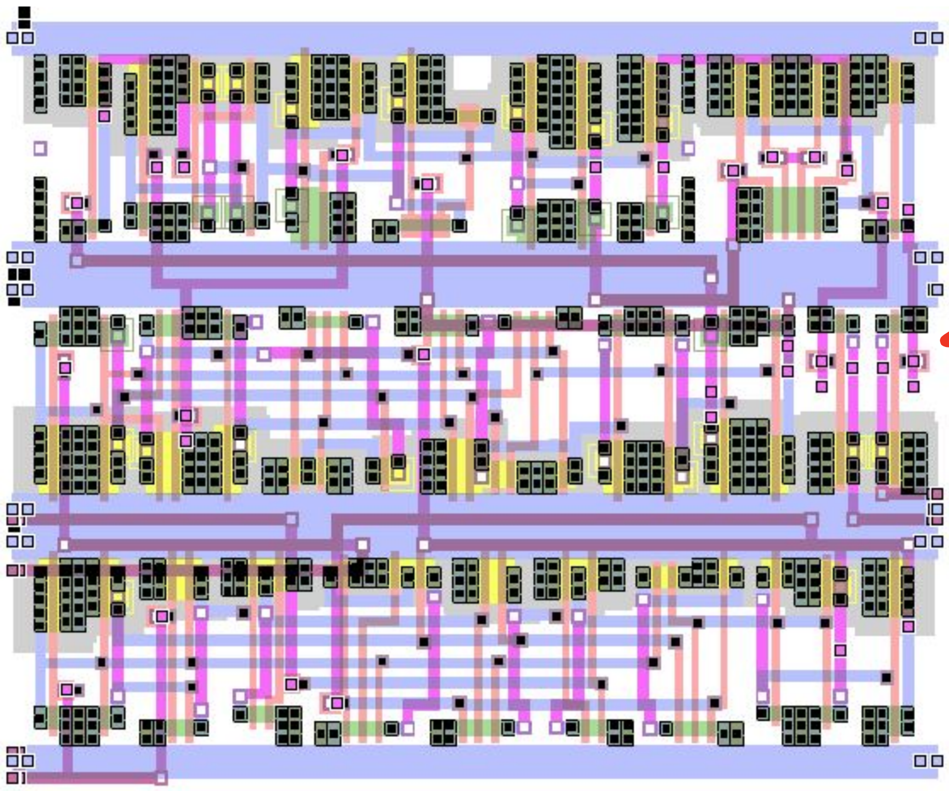




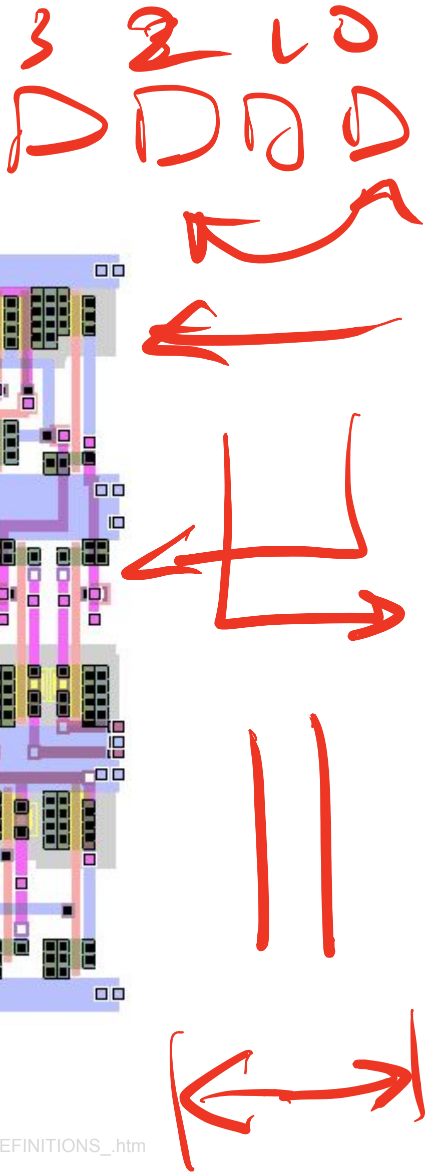
Digital System Abstraction

Transistors	Billions
Gates	100s of Millions
Microarchitectural Blocks	100s
Chips	10s
Boards	~10
System	1

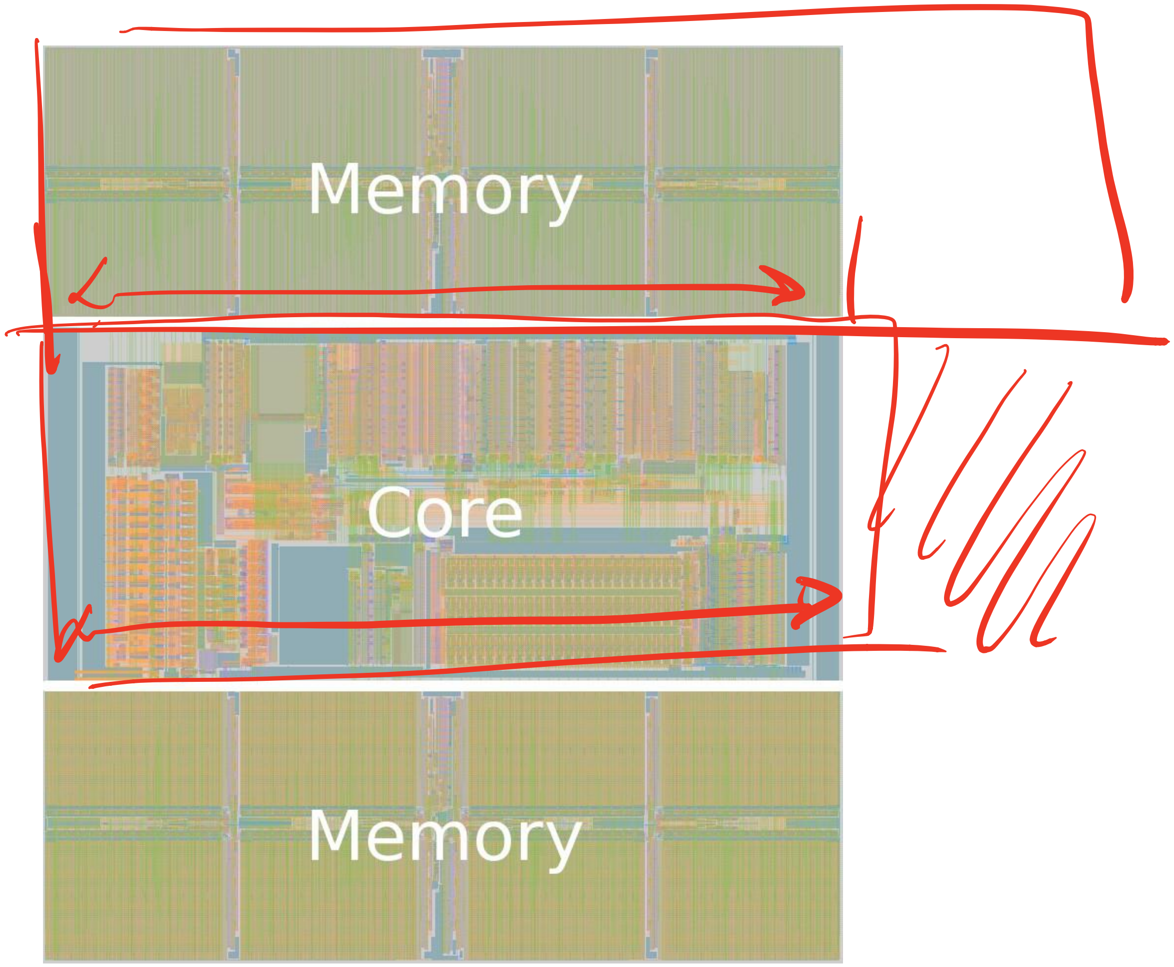
Place and Route

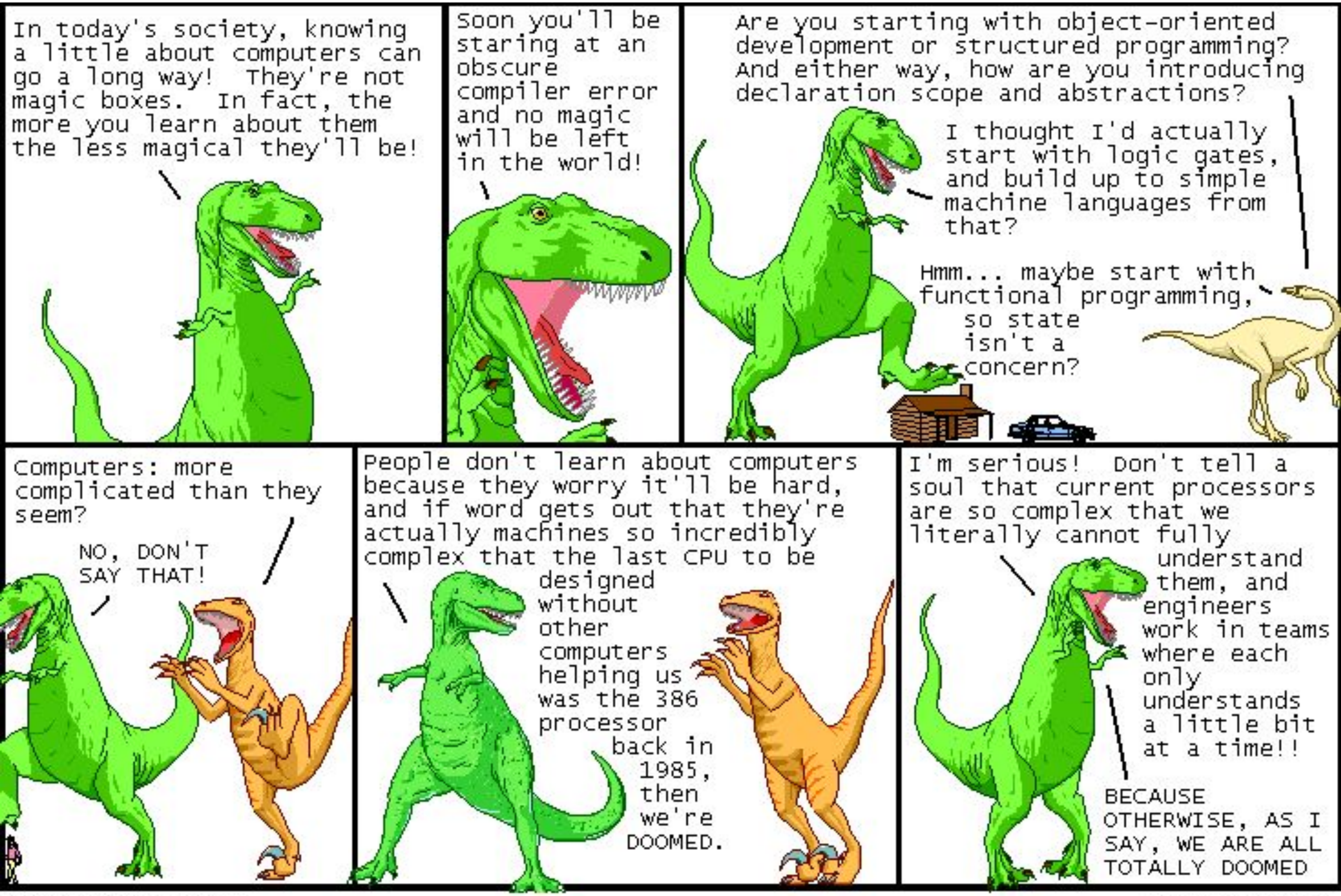


http://bwrcs.eecs.berkeley.edu/Classes/icdesign/ee141_s00/Project/STANDARD%20CELL%20TEMPLATE%20DEFINITIONS_.htm



Close Design




In today's society, knowing a little about computers can go a long way! They're not magic boxes. In fact, the more you learn about them the less magical they'll be!
Soon you'll be staring at an obscure compiler error and no magic will be left in the world!
Are you starting with object-oriented development or structured programming? And either way, how are you introducing declaration scope and abstractions?
I thought I'd actually start with logic gates, and build up to simple machine languages from that?
Hmm... maybe start with functional programming, so state isn't a concern?
Computers: more complicated than they seem?
NO, DON'T SAY THAT!
People don't learn about computers because they worry it'll be hard, and if word gets out that they're actually machines so incredibly complex that the last CPU to be designed without other computers helping us was the 386 processor back in 1985, then we're DOOMED.
I'm serious! Don't tell a soul that current processors are so complex that we literally cannot fully understand them, and engineers work in teams where each only understands a little bit at a time!!
BECAUSE OTHERWISE, AS I SAY, WE ARE ALL TOTALLY DOOMED