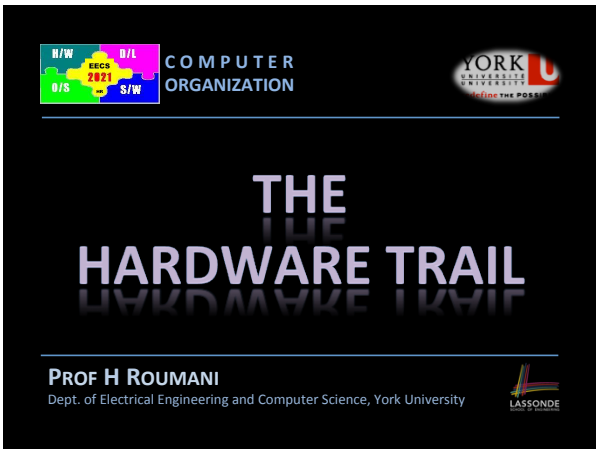
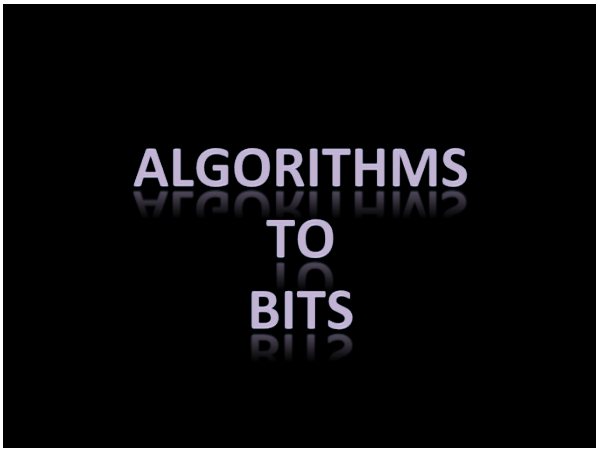
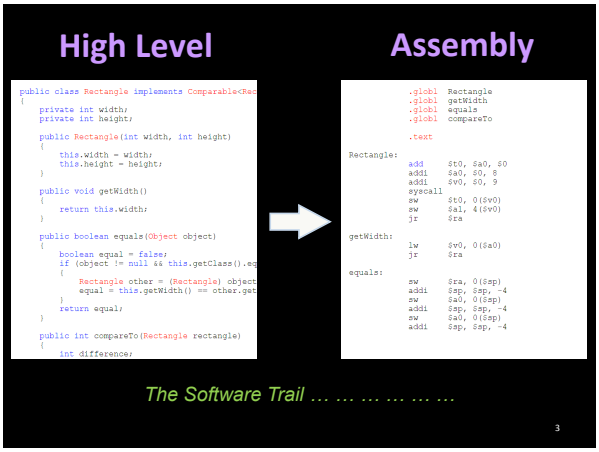
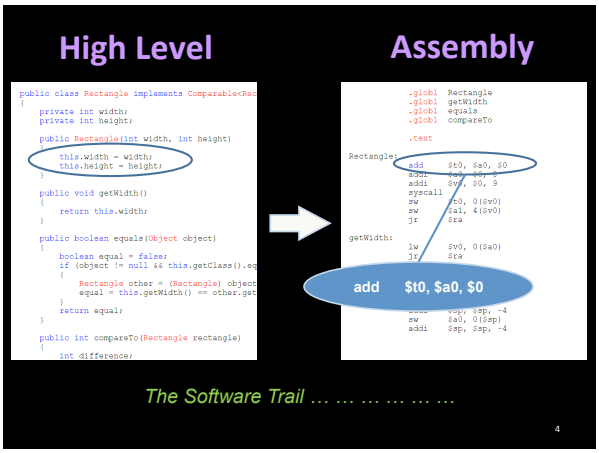
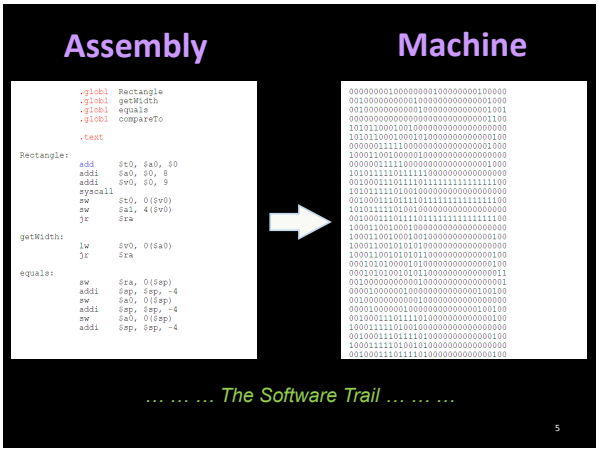
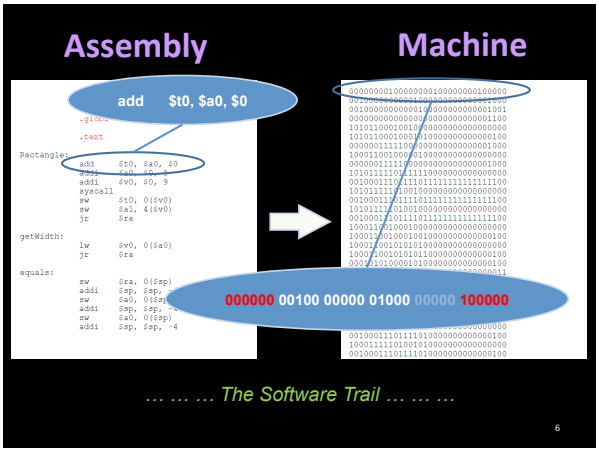

H/W
D/L
EECS
2021
D/S
S/W
COMPUTER
ORGANIZATION
YORK
UNIVERSITY
THE HARDWARE TRAIL
PROF H ROUMANI
Dept. of Electrical Engineering and Computer Science, York University
LASSONDE

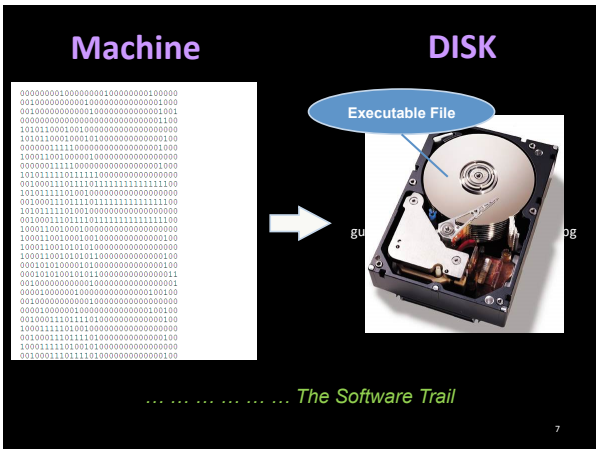

ALGORITHMS
TO
BITS


High Level
Assembly
public class Rectangle implements Comparable<Rec
{
private int width;
private int height;
public Rectangle(int width, int height)
{
this.width = width;
this.height = height;
}
public void getWidth()
{
return this.width;
}
public boolean equals(Object object)
{
boolean equal = false;
if (object != null && this.getClass().eq
{
Rectangle other = (Rectangle) object
equal = this.getWidth() == other.get
}
return equal;
}
public int compareTo(Rectangle rectangle)
{
int difference;
.globl Rectangle
.globl getWidth
.globl equals
.globl compareTo
.text
Rectangle:
add $t0, $a0, $0
addi $a0, $0, 8
addi $v0, $0, 9
syscall
sw $t0, 0($v0)
sw $a1, 4($v0)
jr $ra
getWidth:
lw $v0, 0($a0)
jr $ra
equals:
sw $ra, 0($sp)
addi $sp, $sp, -4
sw $a0, 0($sp)
addi $sp, $sp, -4
sw $a0, 0($sp)
addi $sp, $sp, -4
The Software Trail … … … … … …
3

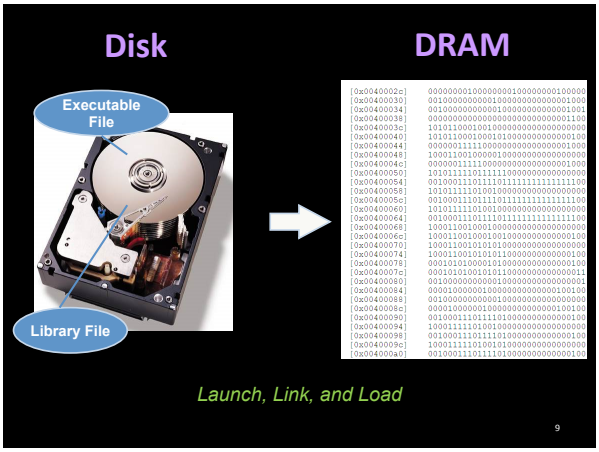

High Level
Assembly
add $t0, $a0, $0
The Software Trail … … … … … …
4

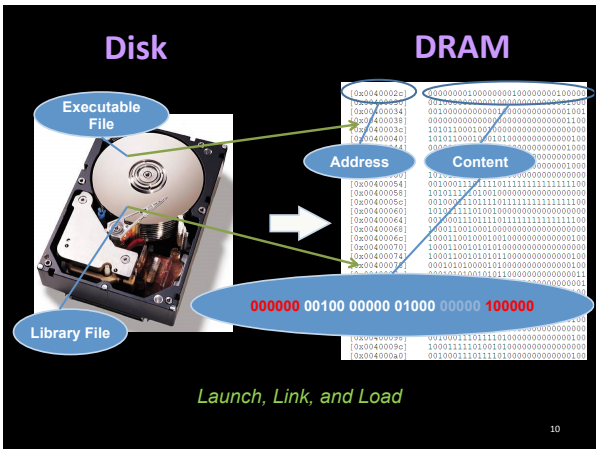

Assembly
Machine
… … … The Software Trail … … …
5

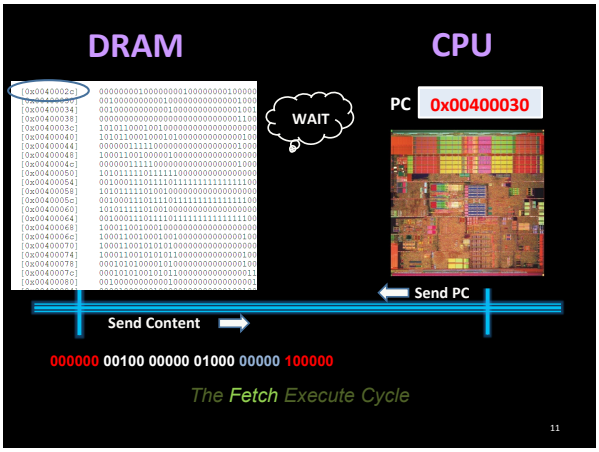

Assembly
Machine
add $t0, $a0, $0
000000 00100 00000 01000 00000 100000
… … … The Software Trail … … …
6

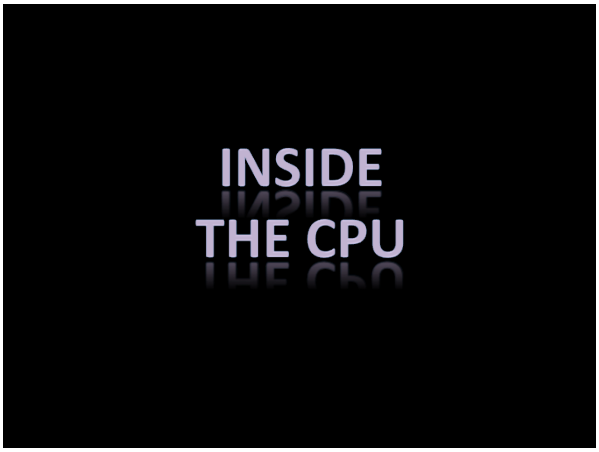

Machine
DISK
Executable File
… … … … … … The Software Trail
7

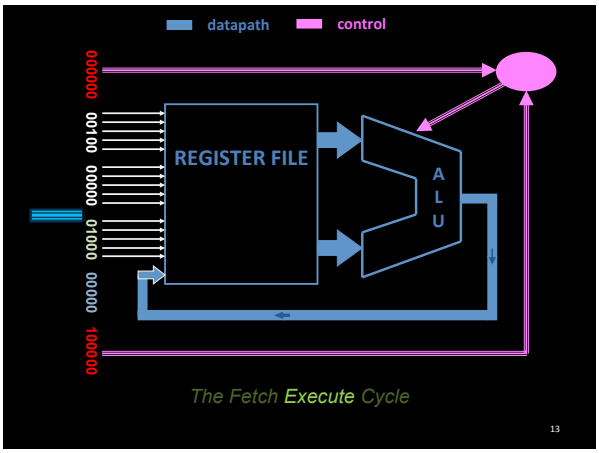

THE
S/W – H/W
INTERFACE

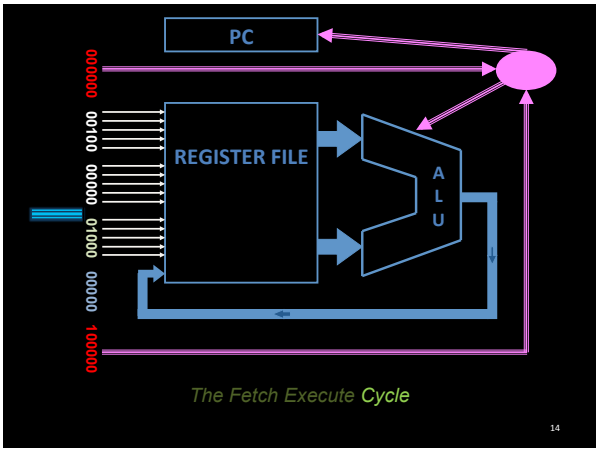

Disk
DRAM
Executable File
Library File
Launch, Link, and Load
9

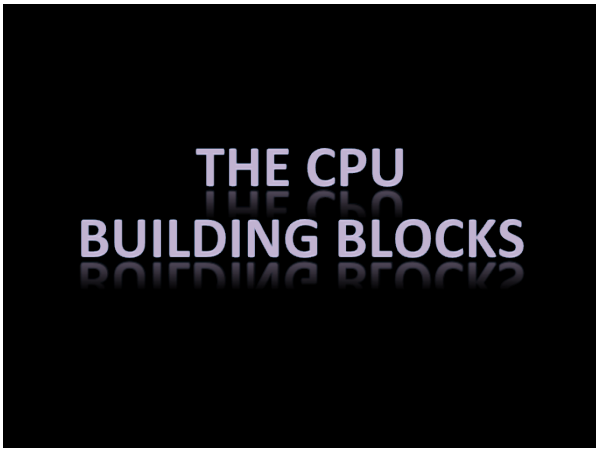

Disk
DRAM
Executable File
Address
Content
Library File
000000 00100 00000 01000 00000 100000
Launch, Link, and Load
10

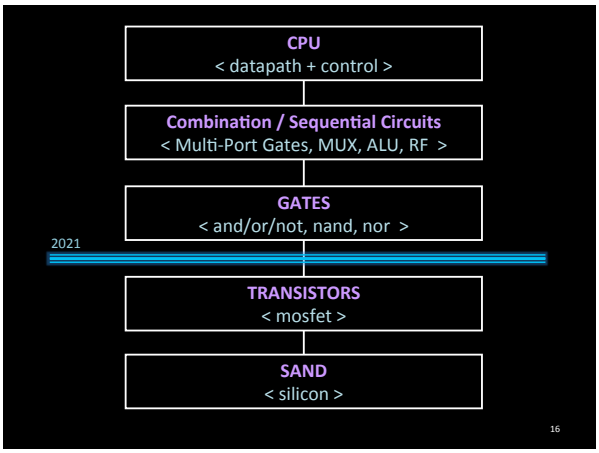

DRAM
CPU
WAIT
PC
0x00400030
Send PC
Send Content
000000 00100 00000 01000 00000 100000
The Fetch Execute Cycle
11

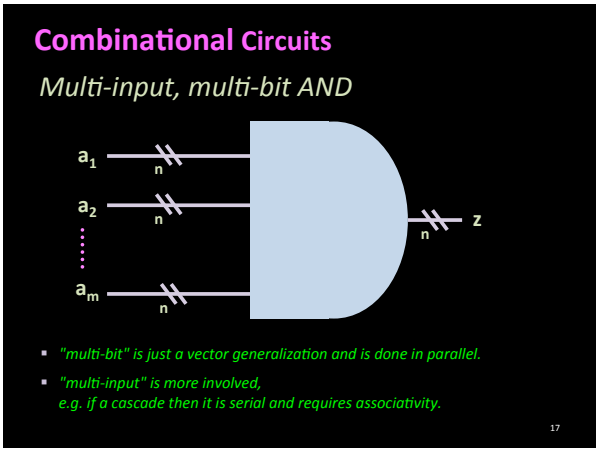

INSIDE
THE CPU

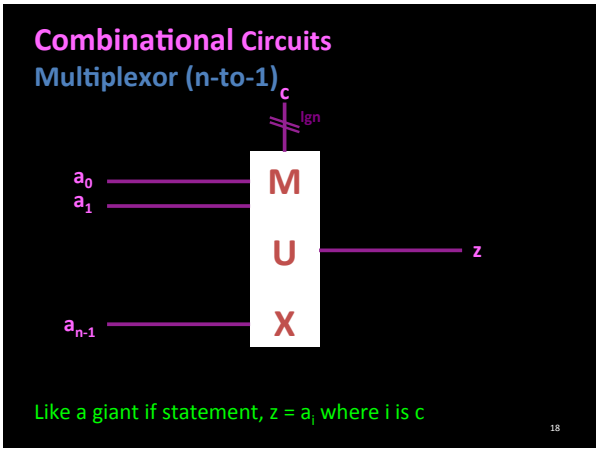

datapath
control
000000
010100
000000
010000
000000
100000
REGISTER FILE
A
L
U
The Fetch Execute Cycle
13


PC
000000
010100
000000
010000
000000
100000
REGISTER FILE
A
L
U
The Fetch Execute Cycle
14


THE CPU
BUILDING BLOCKS

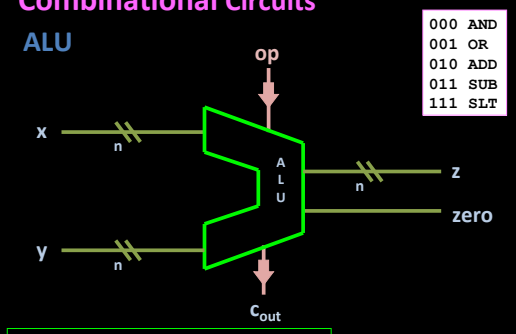

CPU
< datapath + control >
Combination / Sequential Circuits
< Multi-Port Gates, MUX, ALU, RF >
GATES
< and/or/not, nand, nor >
2021
TRANSISTORS
< mosfet >
SAND
< silicon >
16


Combinational Circuits
Multi-input, multi-bit AND
a1
a2
am
n
z
"multi-bit" is just a vector generalization and is done in parallel.
"multi-input" is more involved,
e.g. if a cascade then it is serial and requires associativity.
17


Combinational Circuits
Multiplexor (n-to-1)
c
lgn
a0
a1
an-1
M
U
X
z
Like a giant if statement, z = ai where i is c
18

Combinational Circuits

ALU



zero = (z == 0) ? 1:0
C_{out} = carry out of MSb

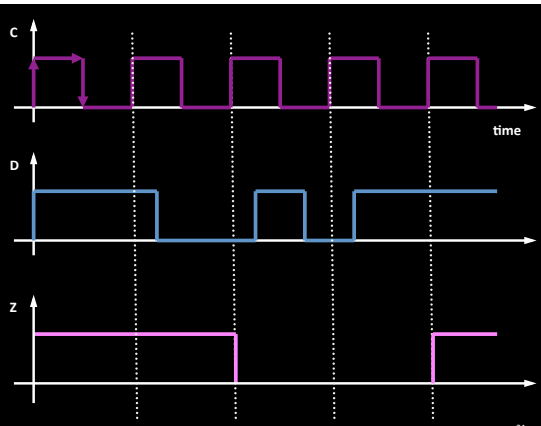
19

Sequential Circuits

Register (positive edge-triggered)

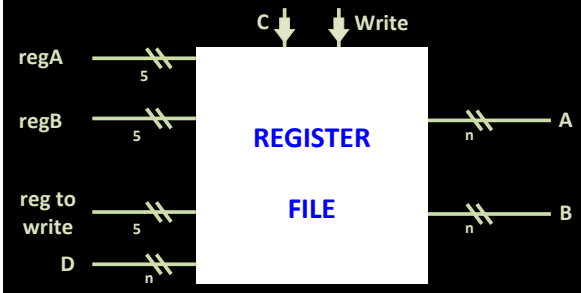
- Z = stored state, always
- Stored state = D on the 0-1 edge of C
- In real life, the rise needs time and the delay is thus >0 but it is comparatively small.

20



21

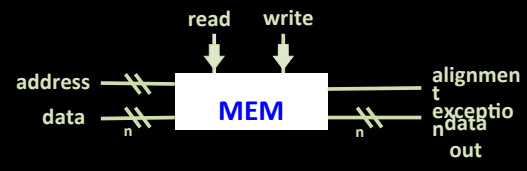
Sequential Circuits
Register File (32 n-bit registers)



Can, in parallel, lookup two reg and write to a third.

22

Sequential Circuits
Memory



Two separate controls needed to enable do-nothing.

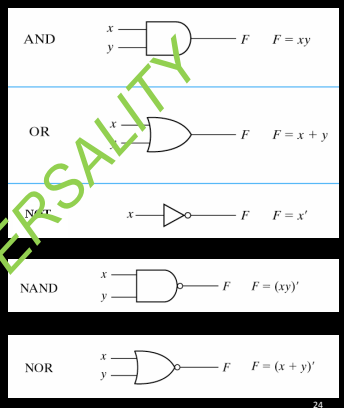
23

GATES

AND/OR/ NOT

NAND

NOR

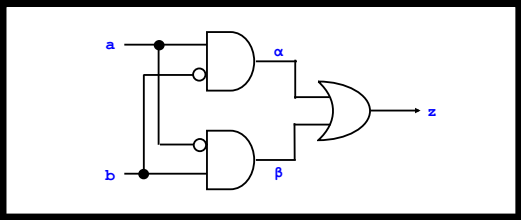


UNIVERSALITY

24

XOR Circuit

Combinational



- *Looks like xor. How can we prove it?*
- *Create it in Verilog.*
- *Create a client to test it using hard-coded, command-line, sampled, or exhaustive inputs.*

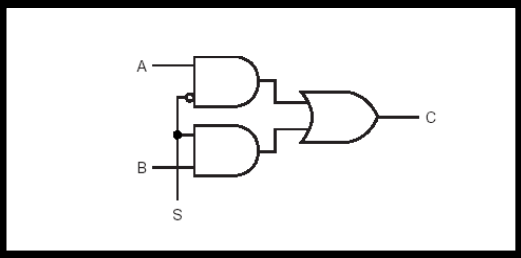
25

If Circuit

Combinational

Verify that this circuit functions as:

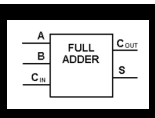
C = (S == 0) ? A : B

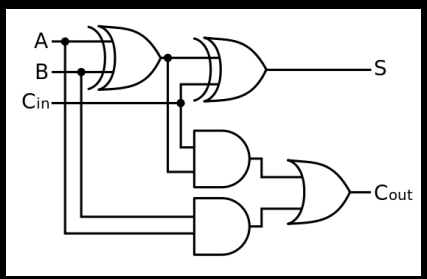


26

Adder

Combinational





27

Exercises

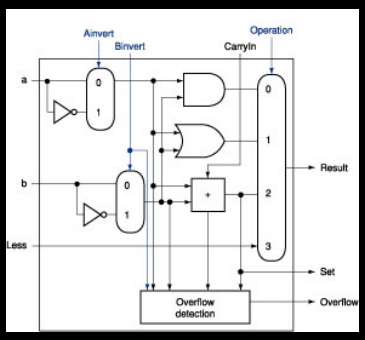
Combinational

- Build an adder / subtractor
- Scale adder/sub to 32 bits
- Build a 32-bit And'er
- Build a 32-bit Or'er
- Build a 32-bit slt circuit

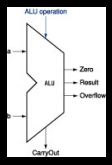
28

ALU

Combinational



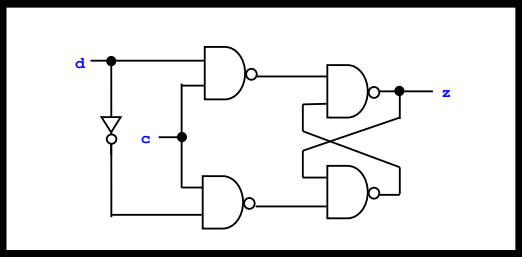

```
000 AND
001 OR
010 ADD
011 SUB
111 SLT
```



29

Flip-Flop (memory)

Sequential

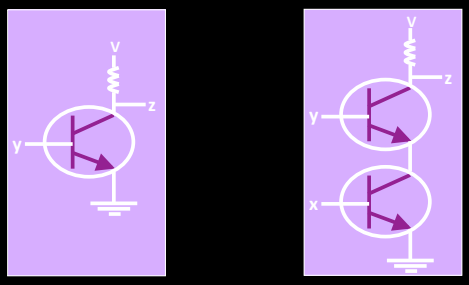


30

Peek below Gates

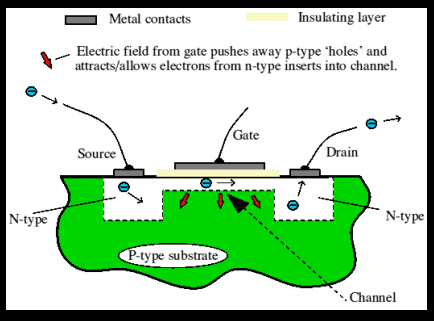
31

Transistor



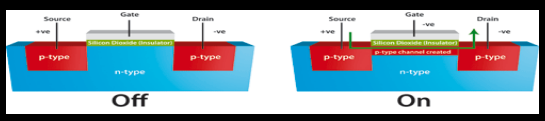
32

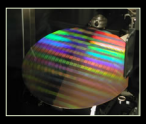
Semiconductors

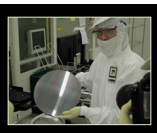


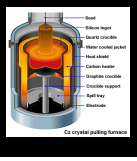
33

FAB









34

Verilog

35

Program Structure

- **Declare**
  Registers, wires, parameters, and ports
- **Instantiate building block**
  Built-in and modules
- **Connect the circuit**
  Use assign or identical port names
- **Initial and always blocks**
  Use initial in test modules only

36

Values

- **Bit**
  0/1, x, z
- **Logical**
  0/x/z are false. Non-zero is true
- **Literal**
  [size] ['radix] value
- **Real**
  At least 1 digit on both sides of . and can use E

37

Sys Tasks

- $display, $monitor
- $time, $stime
- $finish, $stop
- $random

Control Structures

- If-else, case, and the ternary ?
- For and while loops
- repeat (count) loop
- forever loop (must have a delay)

38

Operators

(unary) + - ! ~	Arith, Logical, Bitwise
* / %	Arithmetic
(binary) + -	Arithmetic
<< >> >>>	Shifts
< <= > >=	Relational
== != === !==	Relational
& ~&	Bitwise, Reduction
^ ^~	Bitwise, Reduction
| ~|	Bitwise, Reduction
&&	Logical
||	Logical
?:	Ternary

Plus: part-select [:], concatenate { , }, replication {n{ }}

39