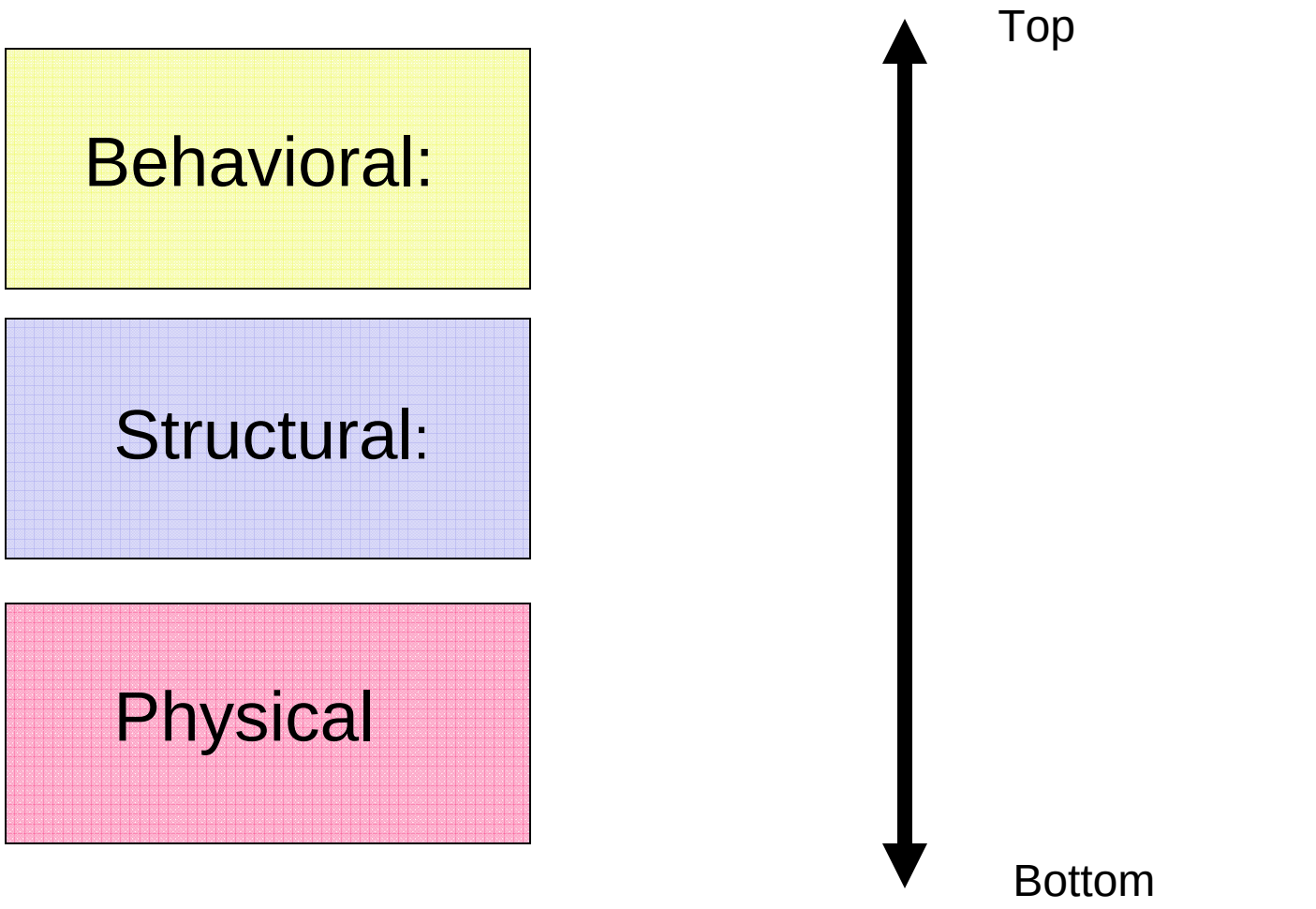
EE 434

Lecture 29

Logic Design

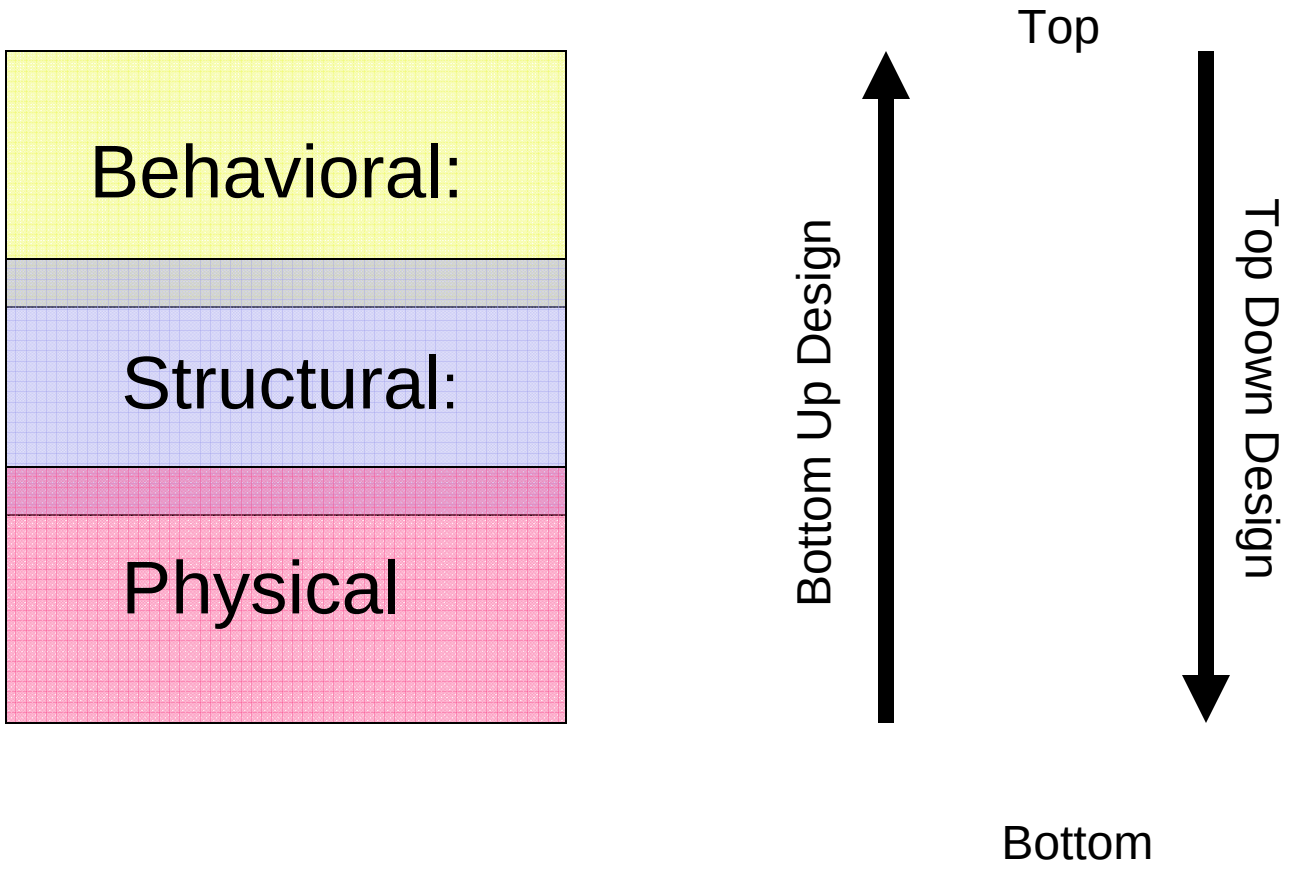
This lecture will focus on the various hierarchical levels used in the design of digital systems. Comments will be made about Hardware Description Languages (HDL) which is an integral part of essentially all digital system design. There are two widely used hardware description languages. One is VHDL and the other is Verilog. In this lecture comments will be made about VHDL. In the balance of this course we will focus on Verilog. There is still debate about which is most popular and that continues in this department. We will focus on Verilog in the balance of this course because we believe most ISU students have more familiarity with Verilog than with VHDL. Both will be used in industry and designers will usually be expected to be comfortable working in both.

Hierarchical Digital Design Domains:

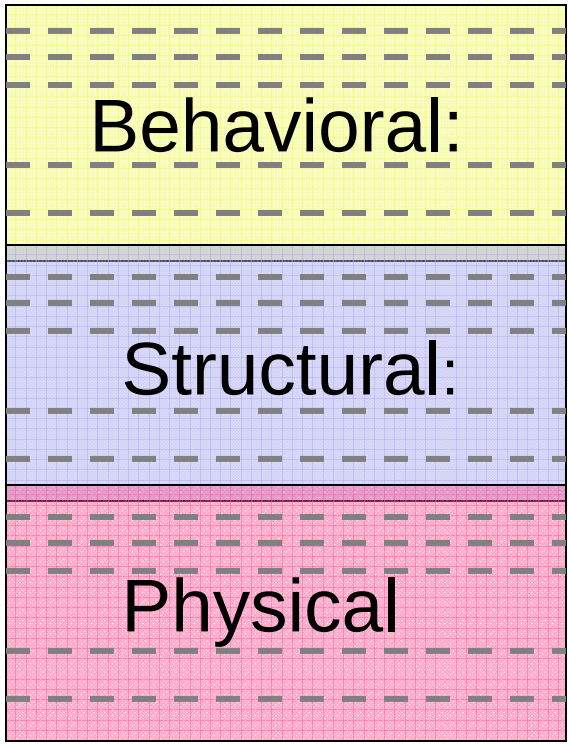


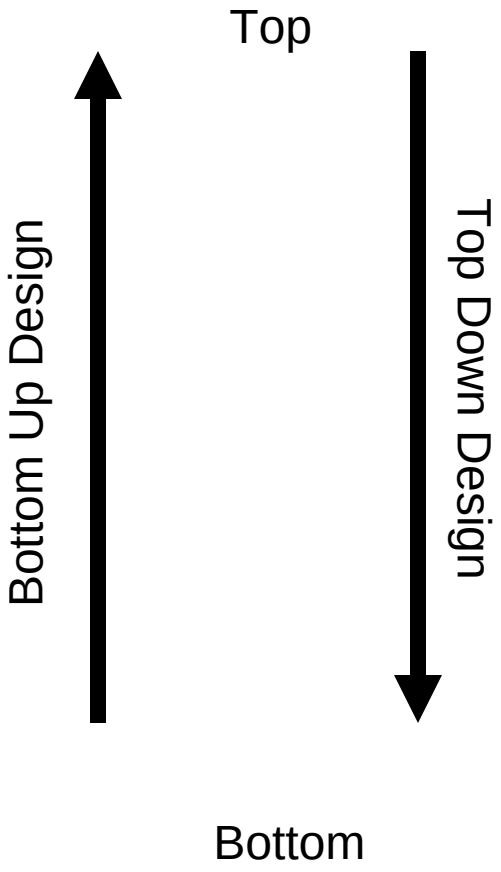
Multiple Levels of Abstraction

Hierarchical Digital Design Domains:



Hierarchical Digital Design Domains:

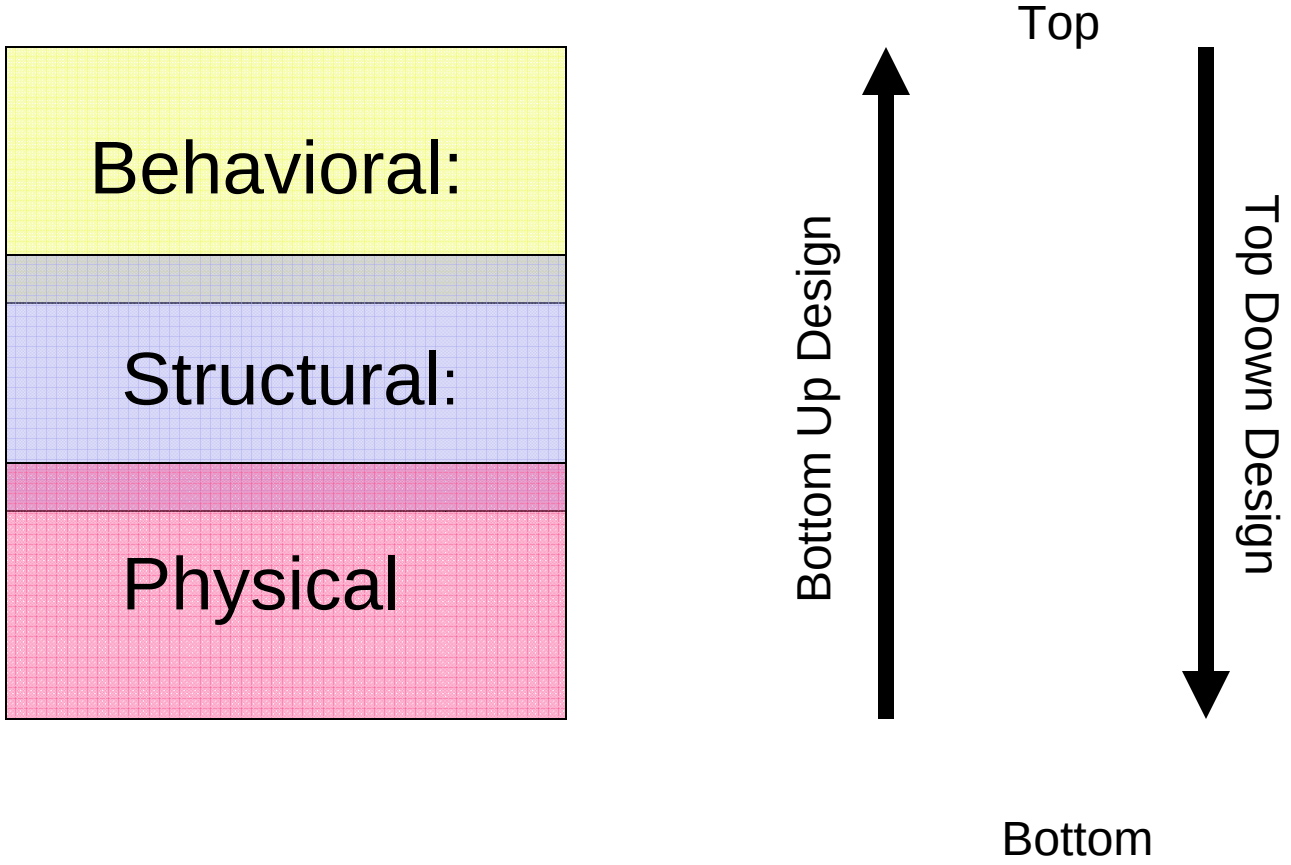




Multiple Sublevels in Each Major Level

All Design Steps may not Fit Naturally in this Description

Hierarchical Analog Design Domains:



Hierarchical Digital Design Domains:

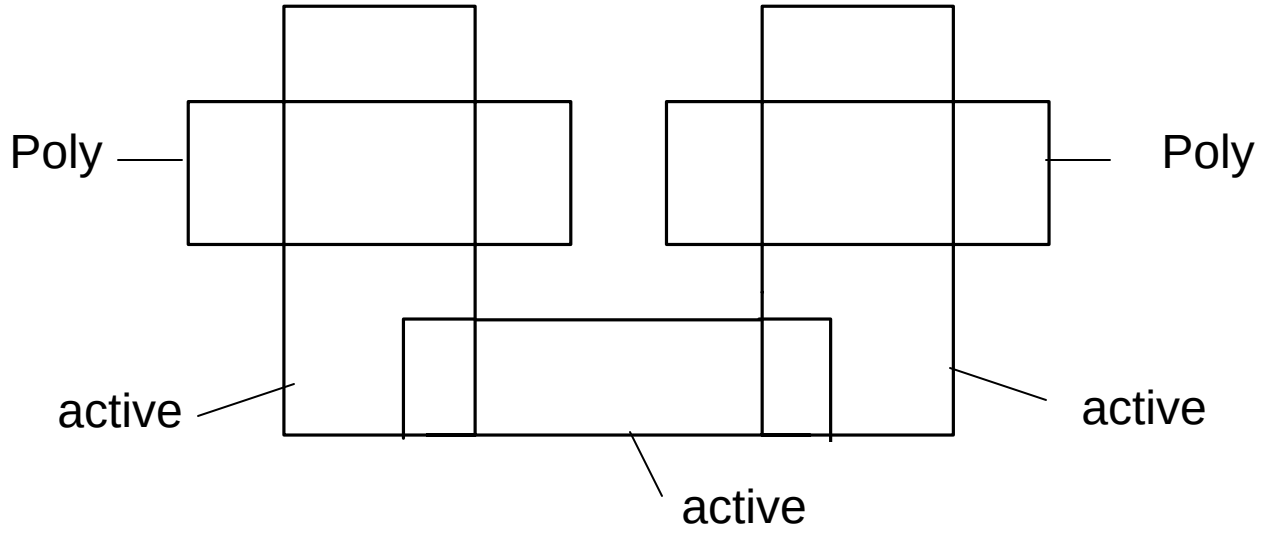
Behavioral : Describes what a system does or what it should do

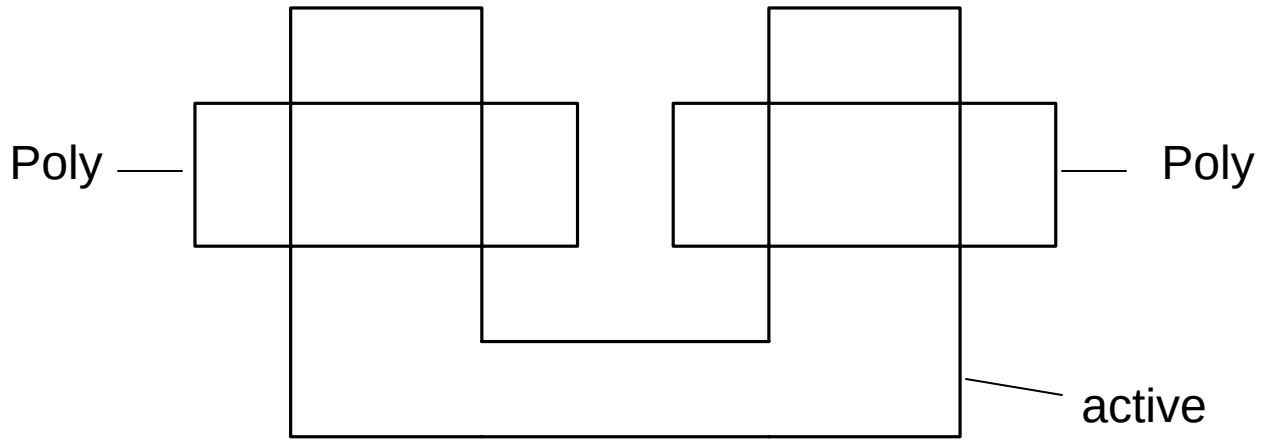
Structural : Identifies constituent blocks and describes how these blocks are interconnected and how they interact

Physical : Describes the constituent blocks to both the transistor and polygon level and their physical placement and interconnection

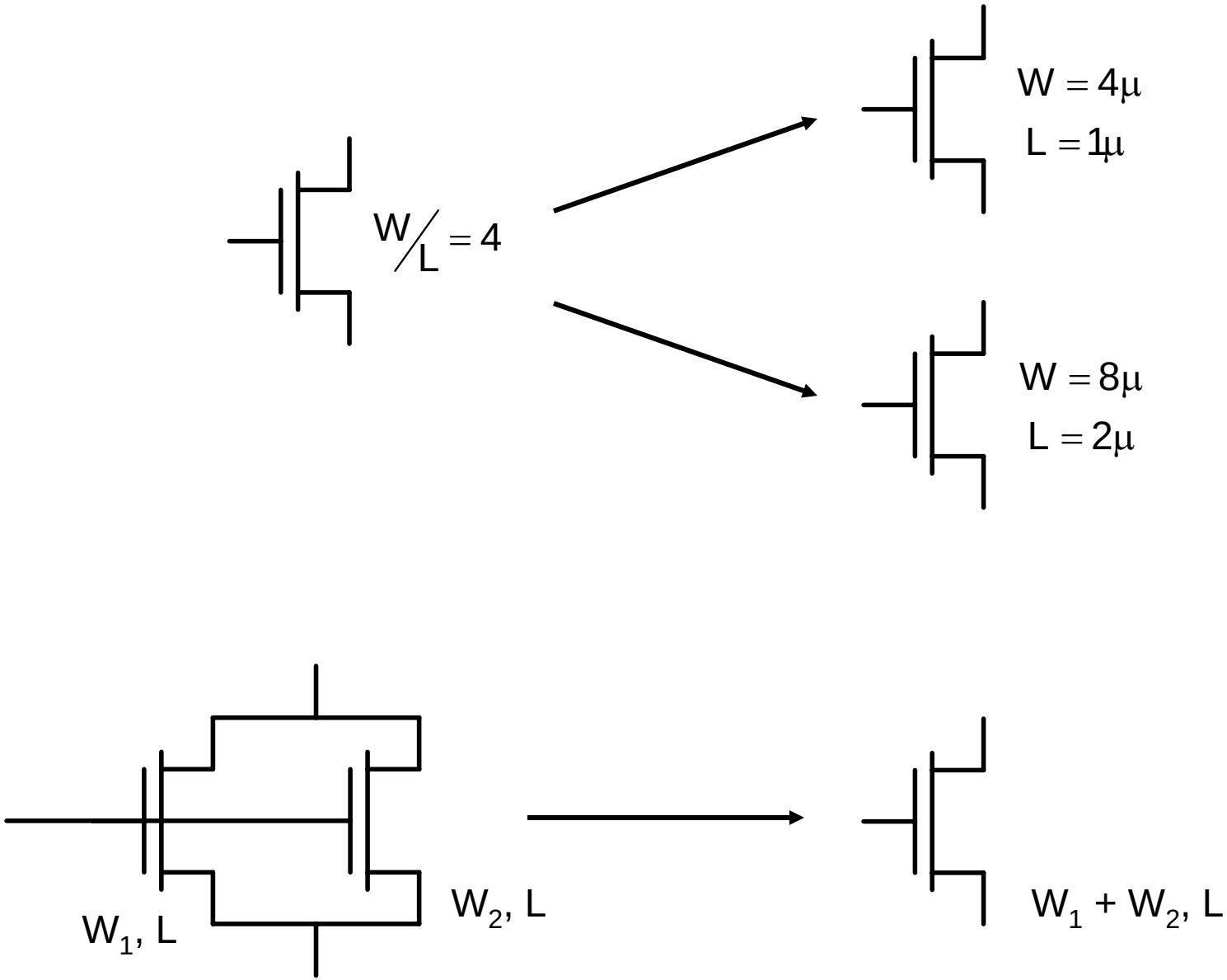
Multiple representations often exist at any level or sublevel

Example: Two distinct representations at the physical level (polygon sublevel)

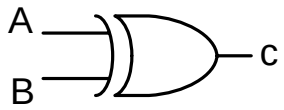




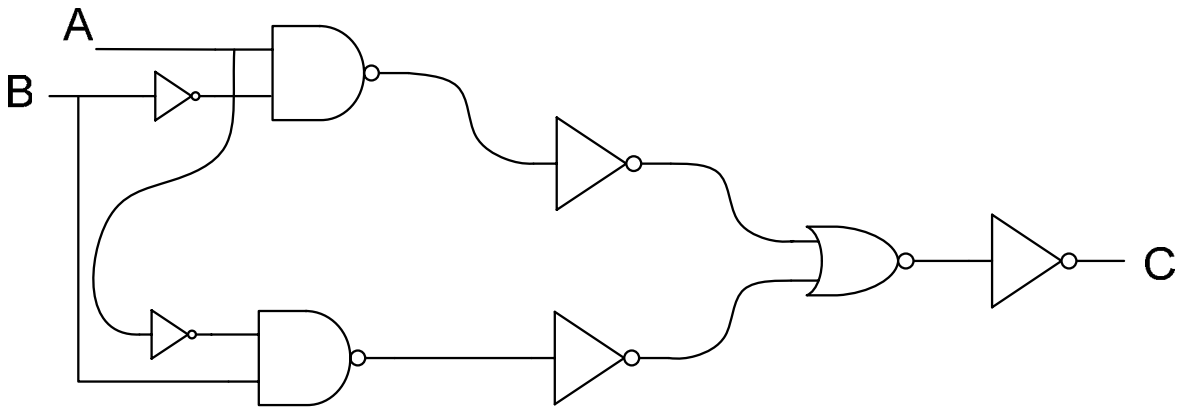
Example: Two distinct representations at physical level (schematic sublevel)

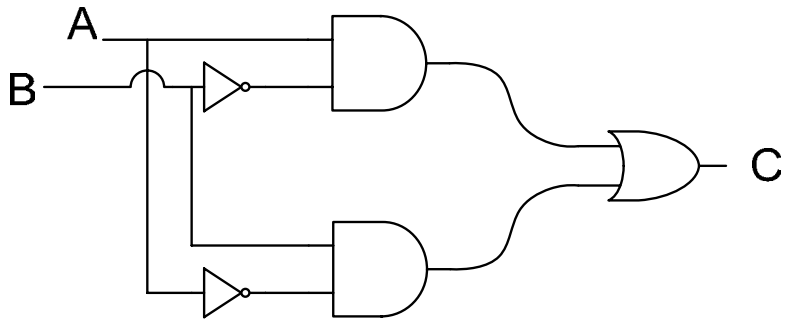


Example: Two distinct representations at the structural/behavioral level (gate sublevel)



$$C = A \oplus B$$





In each domain, multiple levels of abstraction are generally used.

Consider Physical Domain

- Consider lowest level to highest
  - 0 - placement of diffusions, thin oxide regions, field oxide, ect. on a substrate.
  - 1 - polygons identify all mask information (not unique)
  - 2 - transistors (not unique)
  - 3 - gate level (not unique)
  - 4 - cell level Adders, Flip Flop, MUTs,…

Information Type

PG data
G.D.F
Netlist
HDL Description

Structural Level:

- DSP
- Blocks (Adders, Memory, Registers, etc.
- Gates
- Transistor

Information Type

HDL

Netlists

Behavior Level (top down):

- Application
- Programs
- Subroutines
- Boolean Expressions

Information Type

High-Level Language
HDL

Example:

Behavioral Level: System which will give a command to dial an emergency number if a temperature indicator identifies a problem and a humidity indicator indicates no problem **or** if the temperature indicator indicates no problem but the humidity indicator indicates a problem

hum: humidity

tem: temp

call : time to dial emergency number

VHDL Code:

```
entity alarm is

port (hum,tem : in bit ;
call : out hit) ;

end alarm
```

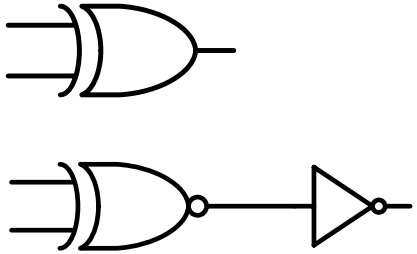
entity declaration

```
architecture  acontroller of alarm is

begin

    call <= ( hum XOR tem );

end   acontroller
```

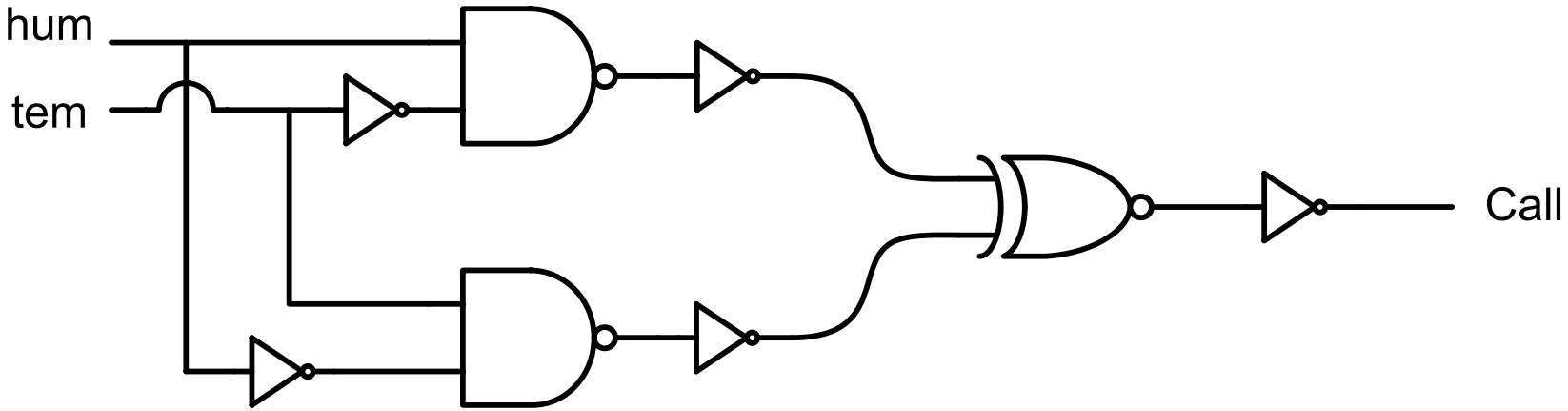
architecture construct

Structural Level:

$$call = hum \oplus tem$$



Physical Level:



Representation of Digital Systems

Standard Approach to Digital Circuit Design

1. Behavioral Description
   - Technology independent
2. RTL Description

   (must verify (1) ⇔ (2))
3. RTL Compiler

   Registers and Combinational Logic Functions
4. Logic Optimizer
5. Logic Synthesis

   Generally use a standard call library for synthesis

6. Place and Route

   (physically locates all gates and registers and interconnects them)

7. Layout Extraction
   - DRC
   - Back Annotation

8. Post Layout simulation

   May necessitate a return to a higher level in the design flow

Logic Optimization

What is optimized (or minimized) ?

- Number of Gates
- Number or Levels of Logic
- Speed
- Delay
- Power Dissipation
- Area
- Cost
- Peak Current

Depends Upon What User Is Interested In

Standard Cell Library

- Set of primitive building blocks that have been pre-characterized for dc and high frequency performance
- Generally includes basic multiple-input gates and flip flops
- P-cells often included
- Can include higher-level blocks
  - Adders, multipliers, shift registers,counters,…
- Cell library often augmented by specific needs of a group or customer