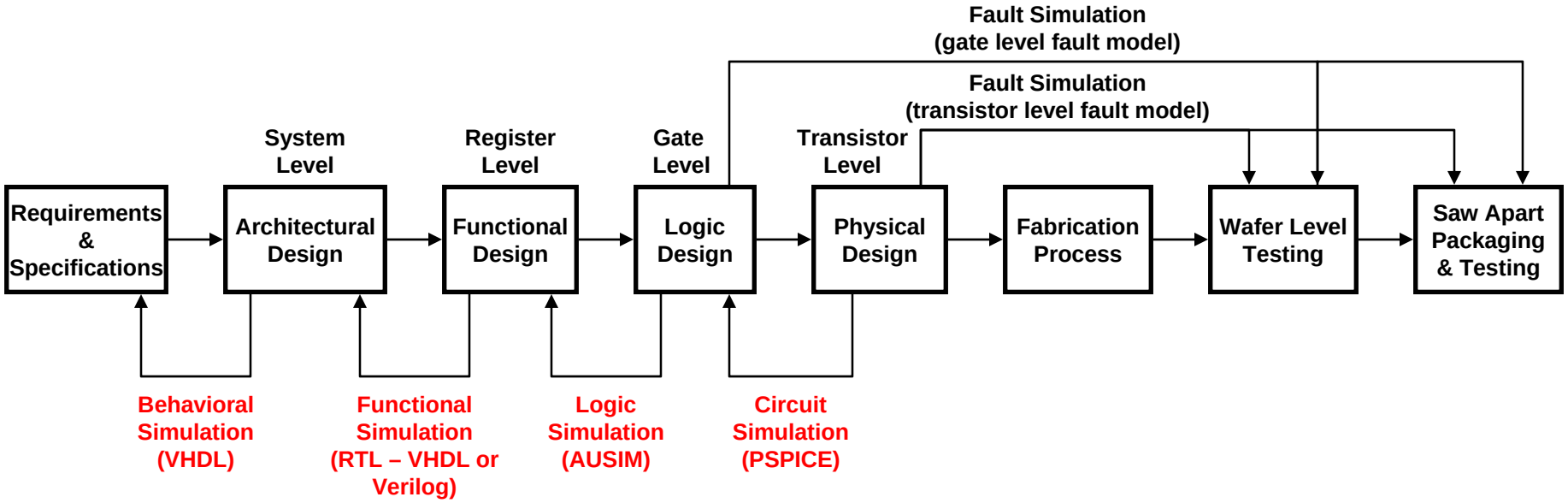
Example of Digital System Design

- The integrated circuit design process



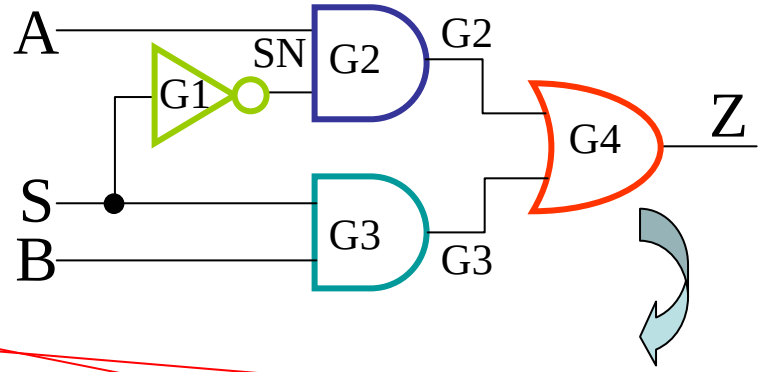
- Note all the simulation (design verification) - helps to ensure the design works and assists in debugging design errors
- In order to simulate a circuit, we must describe it in a manner that can be interpreted and understood by the simulator.

Design Capture with Hardware Description Languages

- Netlist
  - Connections via signal name (like ASL)
- Schematic
  - Connections either explicit or via signal name
  - Produces a netlist for simulation
- Higher level language (VHDL or Verilog)
  - Synthesis to gate level netlist
- All of these design descriptions can go to simulation for design verification

Key Ingredients of Hareware Description Languages (HDLs)

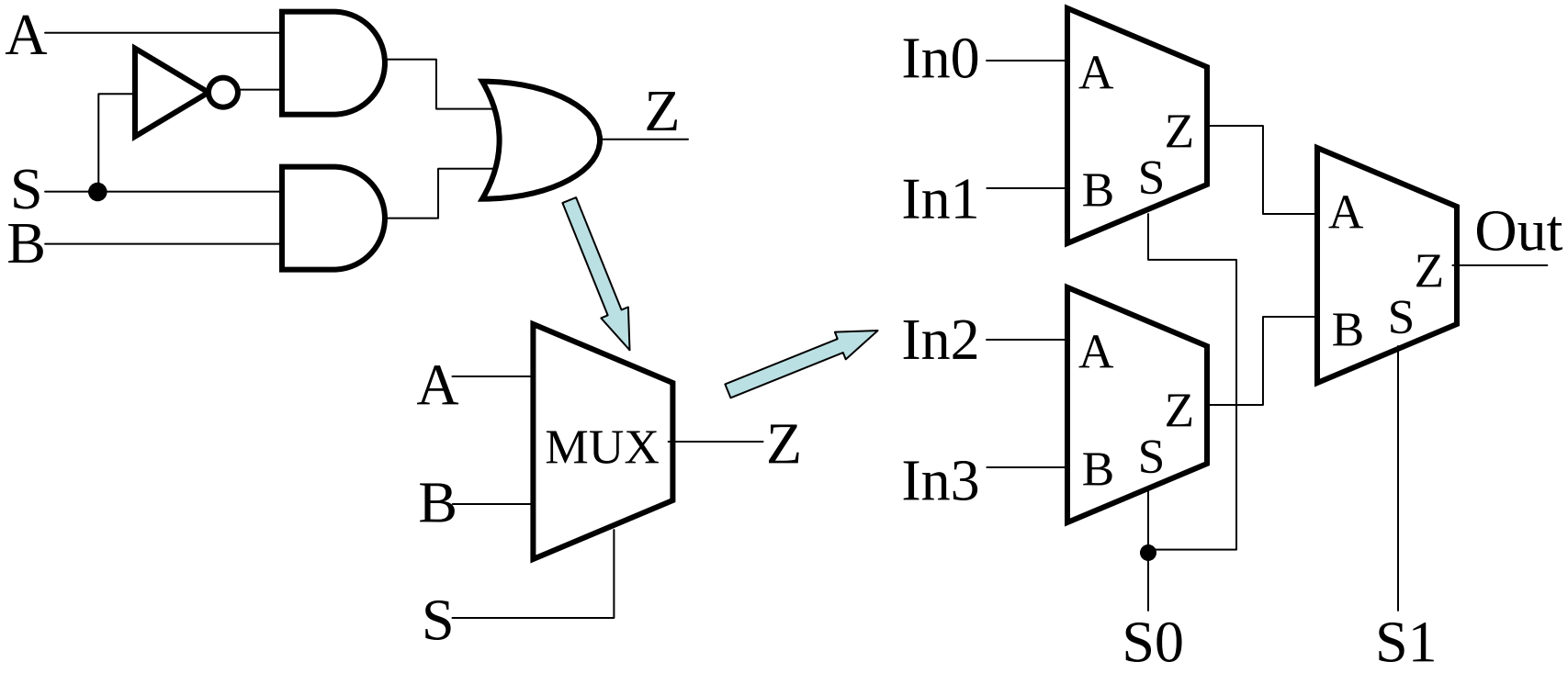
- Circuit statement:
  - Circuit name
  - Inputs
  - Outputs
- Component statements:
  - Component type
  - Component instantiation
  - Inputs signals
  - Output signals




```
ckt: MUX in: A B S out: Z ;
not: G1 in: S out: SN ;
and: G2 in: A SN out: G2 ;
and: G3 in: S B out: G3 ;
or: G4 in: G2 G3 out: Z ;
```

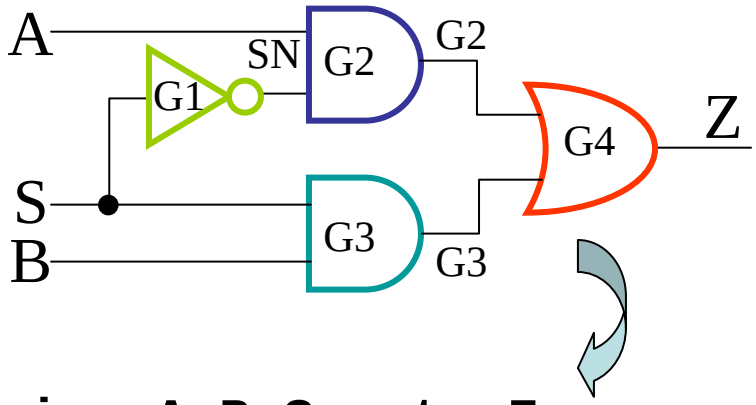
Hierarchical Design

- Hierarchical design saves time & design errors
  - Once a subcircuit has been simulated & is known to work
  - Most HDLs support hierarchical design



Key Ingredients of HDLs

- Hierarchy (subcircuits)
- Connection via signal names (or netnames)
  - Keyword notation
  - Positional notation
  - Bus notation



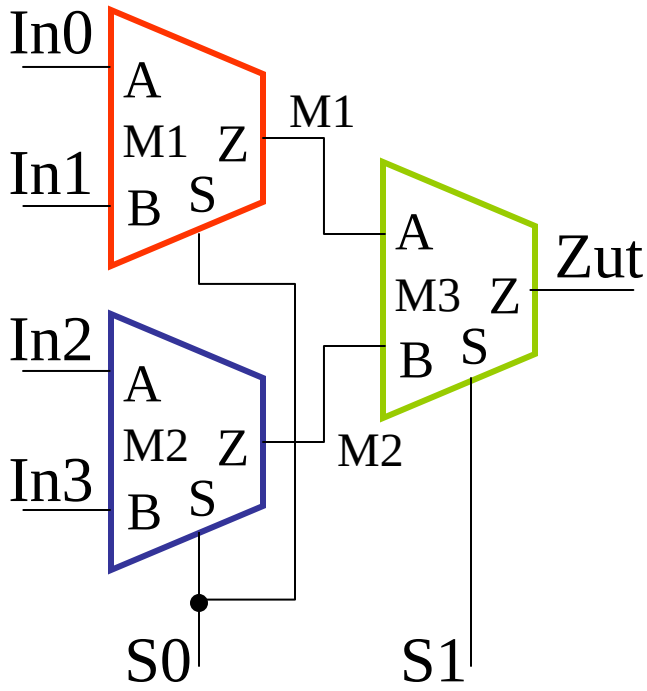

```
subckt: MUX in: A B S out: Z ;
not: G1 in: S out: SN ;
and: G2 in: A SN out: G2 ;
and: G3 in: S B out: G3 ;
or: G4 in: G2 G3 out: Z ;
ckt: MUX4 in: In[0:3] S[0:1]out: Out ;
MUX: M1 in: In0 In1 S0 out: M1 ;
MUX: M2 in: In2 In3 S0 out: M2 ;
MUX: M3 in: M1 M2 S1 out: Out ;
```

Hierarchical Design

- Hierarchical design supported in most HDLs
  - VHDL, Verilog, ASL

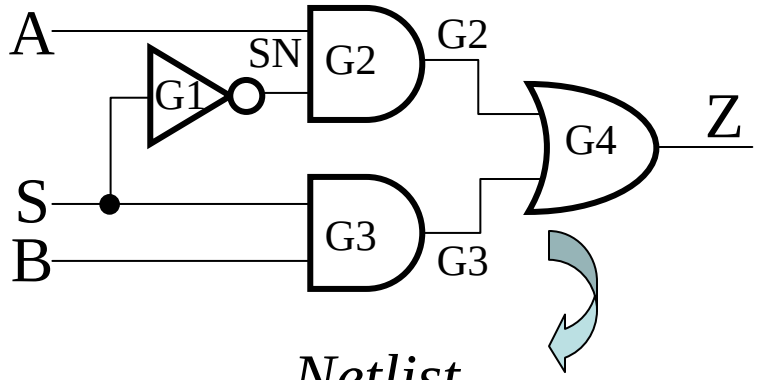
Netlist

```
subckt: MUX in: A B S out: Z ;
not: G1 in: S out: SN ;
and: G2 in: A SN out: G2 ;
and: G3 in: S B out: G3 ;
or: G4 in: G2 G3 out: Z ;
ckt: MUX4 in: In[0:3] S[0:1]
       out: Zut ;
MUX: M1 in: In0 In1 S0 out: M1 ;
MUX: M2 in: In2 In3 S0 out: M2 ;
MUX: M3 in: M1 M2 S1 out: Out ;
```



Design Verification

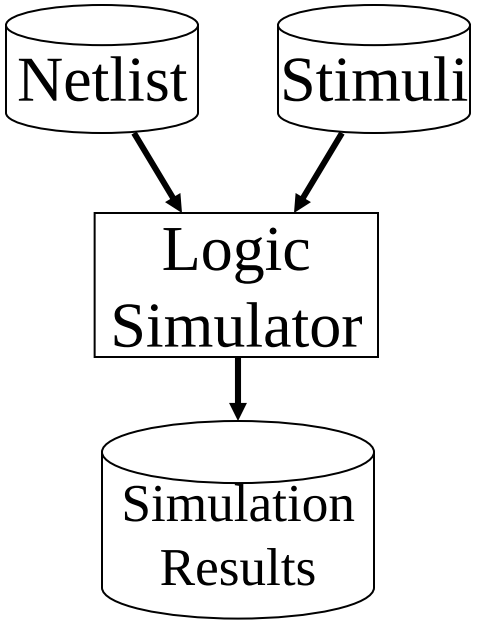
Schematic (or logic diagram)



Netlist

```
ckt: MUX in: A B S out: Z ;
not: G1 in: S out: SN ;
and: G2 in: A SN out: G2 ;
and: G3 in: S B out: G3 ;
or: G4 in: G2 G3 out: Z ;
```

Logic Simulation



Design Verification

- Logic simulation results used
  - To verify proper operation of design
  - To find and fix problems (errors) in design
    - aka *debugging*

Input Stimuli (or vectors)

```
# ABS ;
  000
  001
  010
  011
  100
  101
  110
  111
```

Simulation Results

```
# ABS Z ;
  000 0
  001 0
  010 0
  011 1
  100 1
  101 0
  110 1
  111 1
```

compare simulation results to truth table

Key Ingredients of Logic Design

- Logic simulation results used
  - To verify proper operation of design (design verification)
  - To find and fix problems (errors) in design (aka *debugging*)

Input Stimuli (or vectors)

```
# ABS ;
  000
  001
  010
  011
  100
  101
  110
  111
```

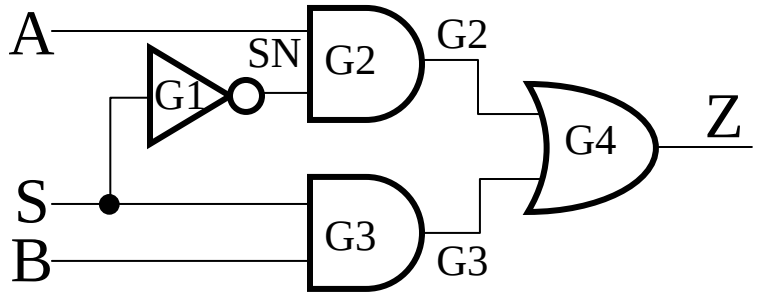
Simulation Results

```
# ABS Z ;
  000 0
  001 0
  010 0
  011 1
  100 1
  101 0
  110 1
  111 1
```

Simulation

Design Capture Input

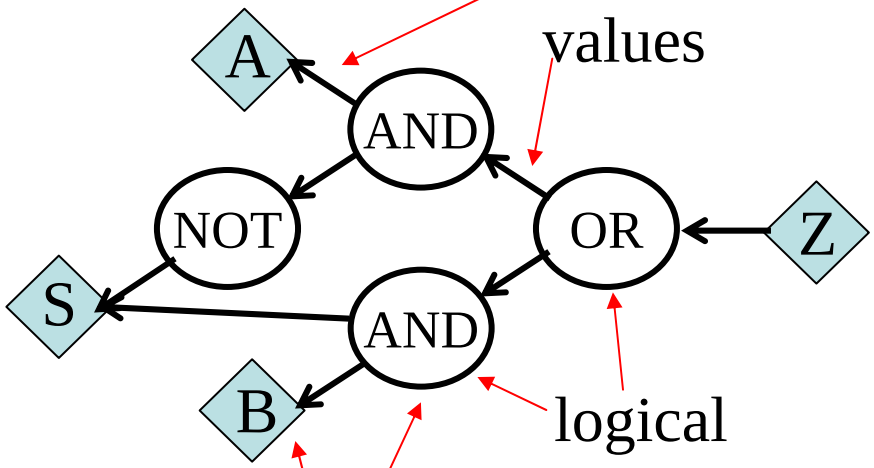
Schematic diagram



Netlist

```
ckt: MUX in: A B S out: Z ;
not: G1 in: S out: SN ;
and: G2 in: A SN out: G2 ;
and: G3 in: S B out: G3 ;
or: G4 in: G2 G3 out: Z ;
```

Computer Emulation



Pointers to source logic values

Data structures to hold logic values

logical operations performed by computer

Types of Simulators

- Compiled Simulator (AUSIM)
  - Simulation continues until circuit is stable
    - No changing logic values within circuit
  - Aka: unit delay or logic simulator
    - All gates in circuit have a finite unit delay
  - Good for initial design verification
    - Short simulation times
- Event-Driven Simulator
  - Simulation events scheduled in time
    - Circuit may not be stable when input changes
  - Aka: timing simulator
    - Gates have real delays base on intrinsic & extrinsic factors
  - More accurate for real circuits
    - longer simulation times and more computer intensive

Other CAD Tools in Logic Design

- Audits for potential design problems
  - Such as “no-connects” or multiple gates driving same net
    - Usually part of another tool (schematic capture or simulator)
- Logic minimization tools
  - Handles combinational logic circuits too big for K-maps
- Timing analysis
  - Finds and reports worst case timing delay path
    - Like P_{del} but uses actual timing parameters per gate