

Circuit and System Representation

IC Designers must juggle several different problems

- Multiple levels of abstraction

IC designs requires refining an idea through many levels of detail, specification -> architecture -> logic design -> layout.

- Multiple conflicting cost

Designs can be judged against different criteria.

Most important criteria include speed, area and power.

If both speed and area constraints are to be satisfied simultaneously, many design decisions will improve one at the expense of the other.

Design is dominated by process of balancing conflicting constraints.

- Short design times

Chips that appear too late may make little or no money because of competitors.

Design time is especially tight for ASICs.

Circuit and System Representation

Hierarchical design

Divide and conquer, complexity is reduced by recursively breaking it down into manageable parts.

Each level of the hierarchy adds complexity by adding components.

Commonly used in programming.

Design abstraction

Complexity is reduced by successively replacing detail with simplifications at higher levels of abstraction.

Regularity

Use hierarchy and attempt to divide the module into a set of similar submodules or blocks.

Modularity

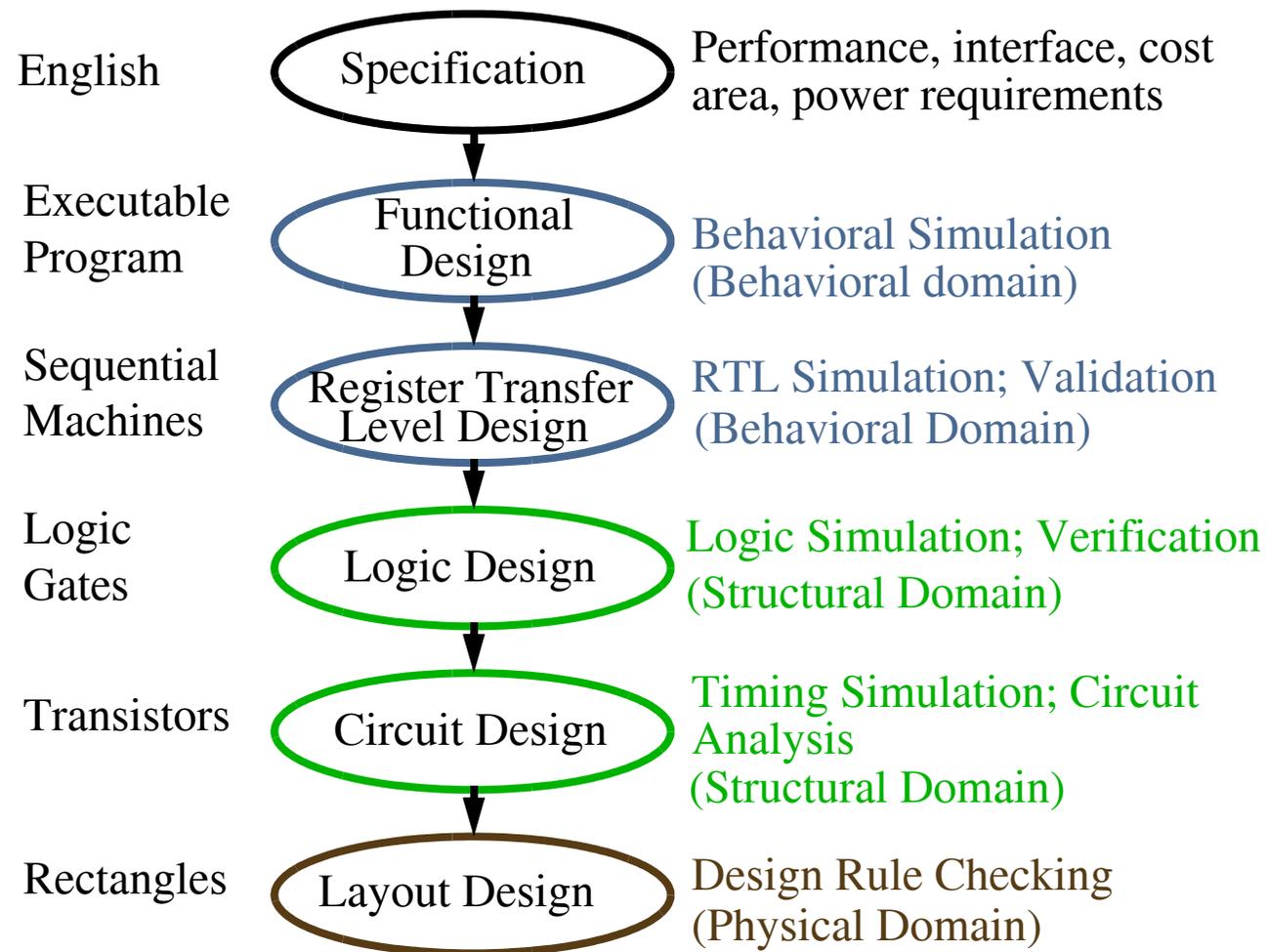
Modularity means that modules should have well-defined functions and interfaces.

Interactions with other modules easier if all modules follow this principle.

Locality

Both Physical and Temporal locality is exploited.

All the signals and edges defined with respect to a single global signal usually clock.

Circuit and System Representation

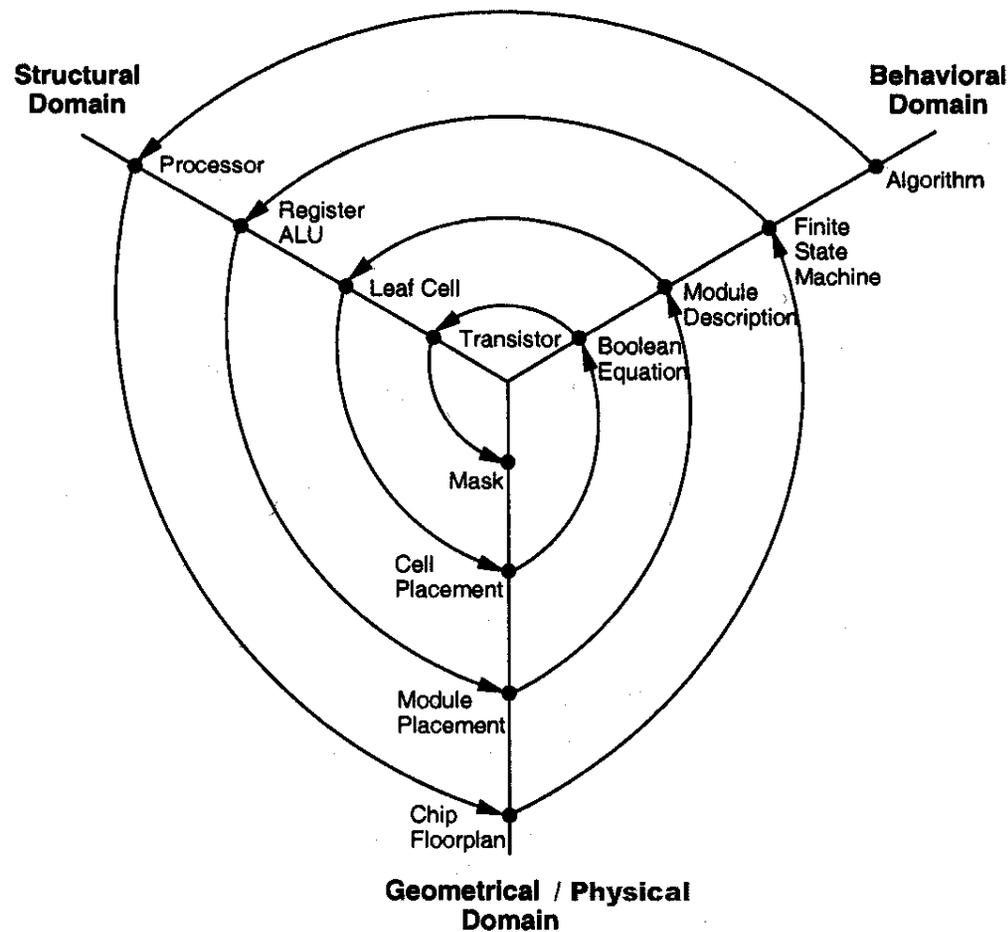
Circuit and System Representation: Gajski Y-Chart

Three design domains:

Behavioral: specifies what a particular system does.

Structural: specifies how entities are connected together.

Physical: specifies how to actually build a structure.



Behavioral Domain

Algorithm written in C, behavioral VHDL or behavioral Verilog, e.g.,

```
module triangle (wave);  
  output [0:3]wave;  
  ...  
  always @(posedge, clock)  
    begin  
      if (wave == 15)  
        begin  
          inc = -1  
        end  
      ...  
    end
```

Functional simulations would be run to verify the behavior and compliance with the specification.

Levels of abstraction include

- Algorithmic (HDLs).
- Register-level transfer: description of specific hardware registers and the communication between them.
- Boolean equations.

Structural Domain

Structural Verilog description

```
module triangle_gen (output, clk, rst);  
  input clk, rst;  
  output [3:0] wave;  
  ...  
  and a1 (s1, output[0], output[1], output[2], output[3]);  
  nor a1 (s2, output[0], output[1], output[2], output[3]);  
  or o1 (s3, s1, s2);  
  ...  
endmodule;
```

Conversion from behavioral to structural domain may be automatic or manual.

Usually RTL description is required by most automated synthesis tools.

Simulations would be run to verify compliance with the behavioral specification.

As gate level netlist is available timing specifications can also be verified.

Structural Domain

Levels of abstraction include

- Module level: e.g., cascading of 1-bit adders to form a 4-bit adder.
- Gate level: (See above).
- Switch level: technology dependent since transistor structure is specified.
- Circuit level: SPICE language allows timing behavior to be assessed, e.g.

```
M1 105 107 108 1 pfet L=2.0U W=4.0U
R5 102 109 139.0
R6 104 110 195.5
M2 0 109 110 0 nfet L=2.0U W=4.0U
R7 104 111 195.5
R8 106 112 139.0
M3 111 112 0 0 nfet L=2.0U W=4.0U
C0 104 0 .01P
C1 100 0 11F
```

Physical Domain

Conversion from structural domain to the physical domain may be automatic or manual

- Gates are usually mapped to standard cells provided in a vendor library.
- Automated Place-and-Route algorithms used to generate layouts.
- Placement involves optimal arrangement of cells and routing solves non-planar interconnection problem.

Detailed parasitic information (capacitance and resistance) is available and can be used for detailed time analysis and estimating power dissipation using various simulation and CAD tools.

Levels of abstraction include

- Module level: Rectangle or polygon that specifies outer boundary of all the geometry plus a collection of ports specifying the position, layer and width.
- Layout level: transistors, wires and contacts.
- Photo-mask information.