# VLSI Logic Test, Validation & Verification

Lecture 2

- Introduction to Sets, Relations, Functions & Graphs

- Boolean logic as Sets and their Graph Representations

- Set = collection of objects; Cardinality of $S = |S|$

- $S = \{a, b, c\}, |S| = 3$

- Notation: $\in, \subset, \supset, \cup, \cap$

- Complement of a set X is $\overline{X}$

- Universe $U = \{1, 2, 3, 4\}, X = \{1, 3\}, \overline{X} = \{2, 4\}$

- $|A \cup B| = |A| + |B| - |A \cap B|$

- Set Difference: $A - B = A \cap \overline{B}$

Example of Set Difference Application

- Universe = all possible states: $U = \{1, 2, 3\}$

- 3 sets: $reached, to, new$; reset state = $\{1\}$

- Reachable states from 1: $new = to - reached = to \cap \overline{reached}$
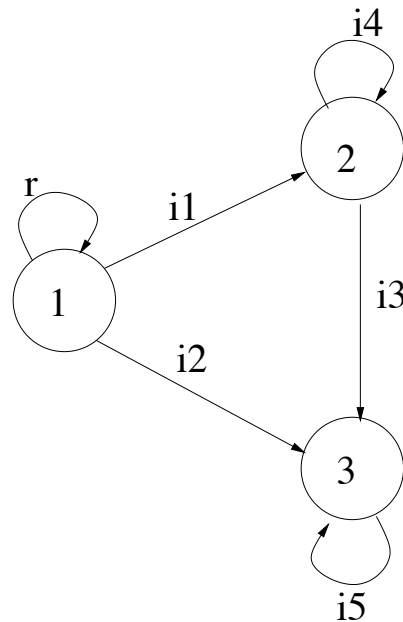
Relations

- Cartesian product $X \times Y$, set of ALL ordered pairs (x, y)

- $X = \{x_1, x_2\}, Y = \{y_1, y_2\}, X \times Y =$
  $\{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)\}$

- Relation R between sets A and B is a mapping between them

  - $R \subset A \times B$, written aRb, $a \in A, b \in B, (a, b) \in R$.

  - Reflexive: $(a, a) \in R$

  - Symmetric: $(a, b) \in R \Rightarrow (b, a) \in R$

 - Transitive: $(a, b) \in R, (b, c) \in R \Rightarrow (a, c) \in R$

R

R1

R2

(R1, R)
(R2, R)

(R1, R2)
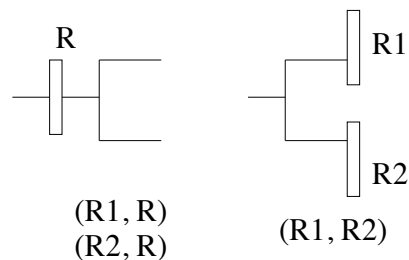
Functions

- Function f: mapping between sets A, B, $f : A \to B$

- A is the domain of f, B is the co-domain

- $y = f(x), x \in A, y \in B$ then $f(x)$ is codomain or range?

- **Image of a function**

  - $y = f(x) : A \to B, x \in A, y \in B$

  - y is the image of x under f

  - Let $C \subseteq A$, then $IMG(f, C) = \{f(x) | \forall x \in C\}$

IMAGE(F, 1)

B^n
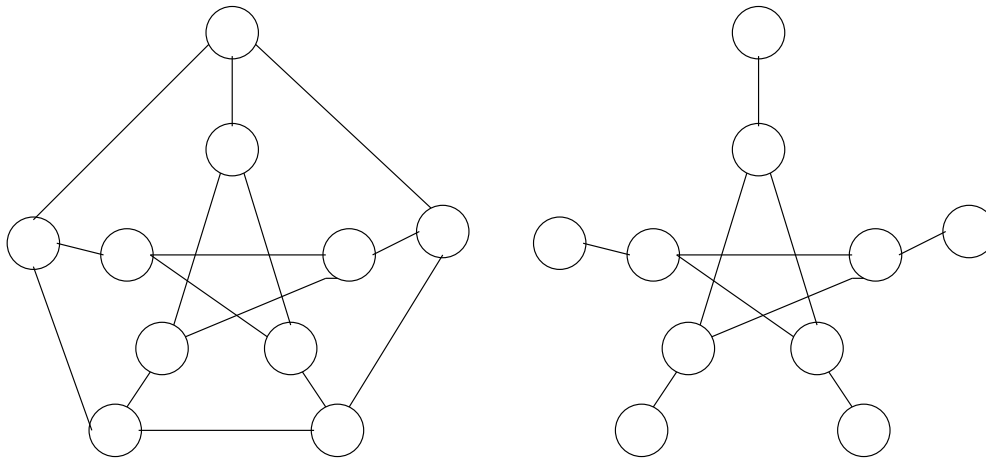
B^m

C

N

N = IMAGE(F, C)
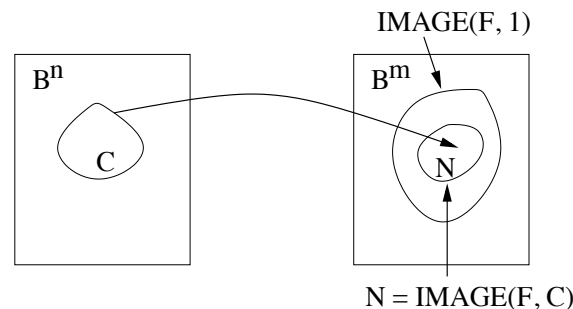
Graph Theory

- Graph $G(V, E)$ is a pair (V, E) where V, E are sets of vertices and edges.

- Edge $e_{ij} = (v_i, v_j) \in R; v_i, v_j \in V; e_{ij} \in E$

- When Edge e meets Vertex v: e is **incident** on v

- **Degree** of v = number of incident edges.
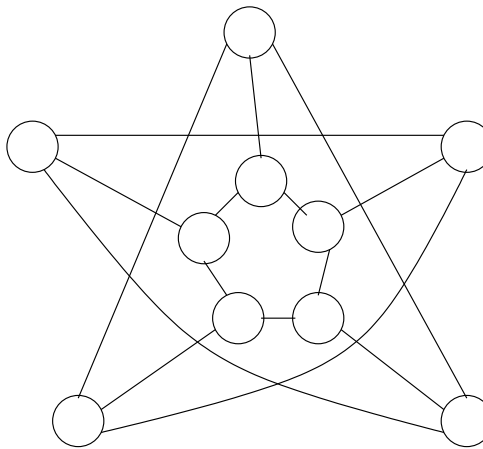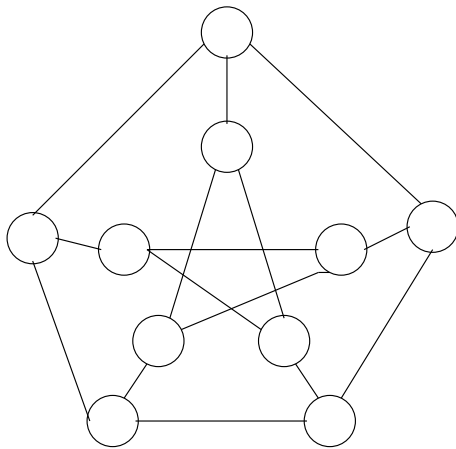
Traversing a Graph

- **Walk**: Sequence of v and e

- **Path**: Walk with distinct vertices (no repetition of v)

- **Connected Graph**: From each vertex, there is a path to all vertices

- Directions on edges == directed graph or **digraph**

Circuits as Graphs

- Represent gates as vertices (nodes), wires as **directed** edges

- Inputs and outputs are also represented as vertices

- A path from input-to-output has no duplicated vertices (gates)

- Delay of a circuit == **longest path** from input to output

- Directed graph with no cycles = Directed Acyclic Graph!

- What if there are directed cycles in a graph corresponding to a circuit?

Trees as Graphs

- Tree = Connected acyclic graph

- Binary Tree = Tree with only one vertex of degree 2, and all other vertices of degree 3 or 1

- Truth Table versus Binary Tree

```
a  b  c  | f
0  0  0  | 0
0  0  1  | 0
0  1  0  | 0
0  1  1  | 1
1  0  0  | 0
1  0  1  | 1
1  1  0  | 1
1  1  1  | 1
```
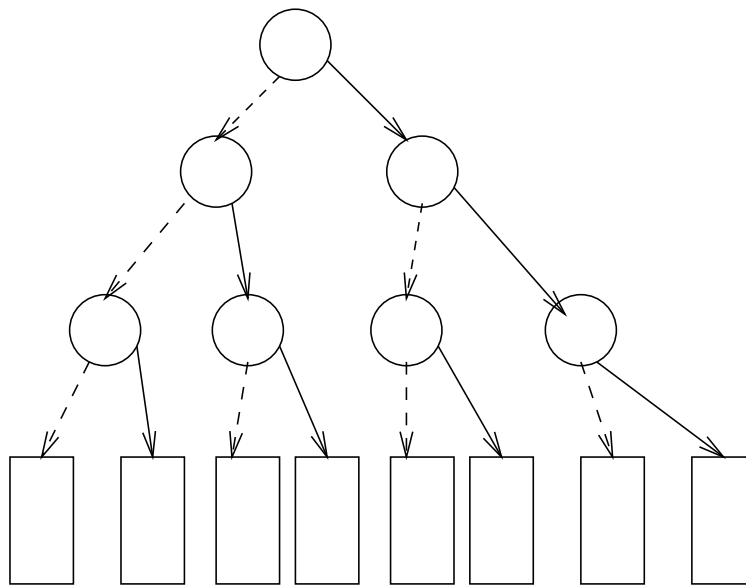
Boolean Algebra - "Basics"

- Algebra is defined over a **set** along with operations on it

- Boolean Algebra: set $= \mathbf{B} \equiv \{0, 1\}$, OPS: $+$ and $\cdot$

- Complement: $a \subset B$, denoted by a' or $\overline{a}$

- Boolean Function is a mapping between domain and co-domain

  - Completely specified Boolean function $f : B^n \to B$

  - $n =$ number of variables of the system

  - $B^n =$ Boolean space spanned by n variables: n-D **cube**

- Multi-output Boolean $f : B^n \to B^m$

- Incompletely specified Boolean function: input space is $\subset B^n$

n-Dimensional Cube

$f = a'bc + ab'c + abc' + abc$

$f = ab + bc + ac$
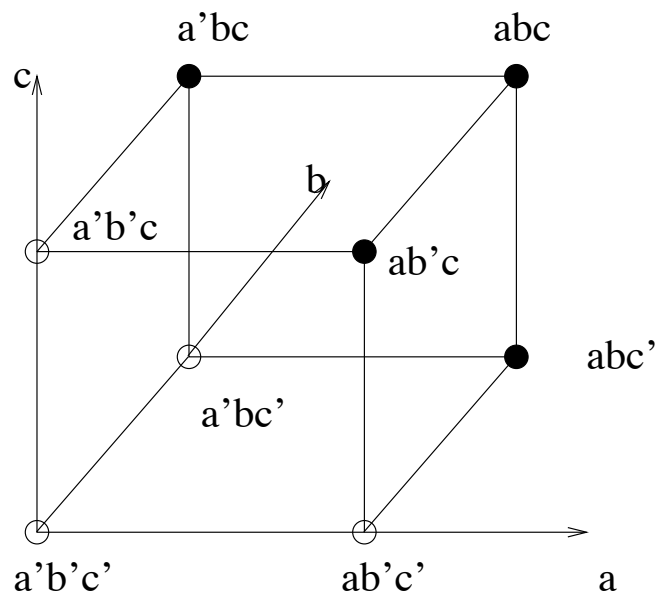
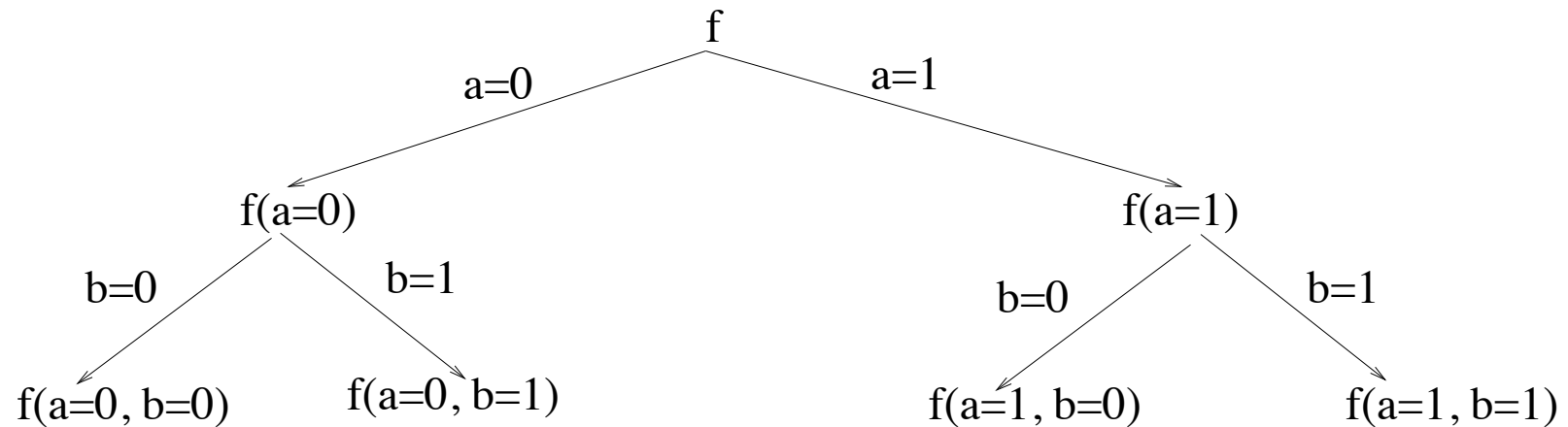| a \ bc | | | | |
|---|---|---|---|
| | | 1 | |
| | 1 | 1 | 1 |

Terminology

- **Binary Variable**= symbol. Represents a co-ordinate of B^n

- **Literal**: Boolean variable OR its complement

- $f = a + a'b$ has how many literals?

- Objective of Logic Optimization: Minimization of literals

- **Cube**: a point, or a set of points in B^n

- Terminology comes from n-D cube

- $a'bc + abc$: 2 cubes. bc is a smaller or larger cube?

- **Support** variables of a function....

Co-Factors of a Boolean Function

- $f(x_1, \ldots x_i, \ldots x_n)$ be a function

- $f_{x_i} = f(x_1, \ldots x_i = 1, \ldots, x_n)$ is positive cofactor of f

- $f_{\overline{x_i}} = f(x_1, \ldots x_i = 0, \ldots, x_n)$ is negative cofactor of f

- $f = ac + bc + ab$

- $f_a = f_{a=1} =$

- $f_{\overline{a}} = f_{a=0} =$

- Co-factor of f has more or few or equal number of literals?

- Cofactors of f w.r.t. x DO NOT contain x

- Shannon's (Boole's) Theorem:

  $- f = x \cdot f_x + x' \cdot f_{x'}$

Further Co-factoring...

- $f_a = f_{a=1} = b + c$

- $f_{\overline{a}} = f_{a=0} = bc$

- $f = af_a + a'f_{a'} = a(b+c) + a'(bc)$

f

a=0   a=1

f(a=0)   f(a=1)

b=0   b=1   b=0   b=1

f(a=0, b=0)   f(a=0, b=1)   f(a=1, b=0)   f(a=1, b=1)

Monotonicity of Boolean Functions

- $f(x_1, \ldots, x_i, \ldots, x_n)$ is positive **unate in variable** x_i if $f_{x_i} \supseteq f_{\overline{x}_i}$

- $f(x_1, \ldots, x_i, \ldots, x_n)$ is negative **unate in variable** x_i if $f_{x_i} \subseteq f_{\overline{x}_i}$

- $f = a + b + c'$

- $f_a = 1 + b + c = 1 \equiv$

- $f_{a'} = b + c' = \equiv$

- The **minterms** of f_a **contain** the minterms of $f_{a'}$

- If $f_a \supseteq f_{a'}$ then

    - $f = a f_a + f_{a'}$
    - What if $f_a \subseteq f_{a'}$?