Kungl Tekniska Högskolan

International Master Program in System-on-Chip Design

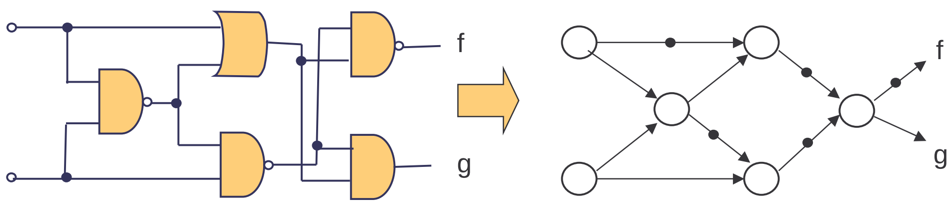
L5: Boolean Circuits

Boolean circuits

- Used for two main purposes
  - as representation for Boolean functions
  - as target structure for logic implementation which gets restructured in a series of logic synthesis steps until result is acceptable
- Efficient representation for most Boolean problems we have in CAD
  - memory complexity is same as the size of circuits we are actually building
- Close to input representation and output representation in logic synthesis

AND-INVERTER circuits

- Base data structure uses two-input AND function for vertices and INVERTER attributes at the edges (individual bit)
  - use De'Morgan's law to convert OR operation etc.
- Hash table to identify and reuse structurally isomorphic circuits (similar to BDD hash tables)



Notation

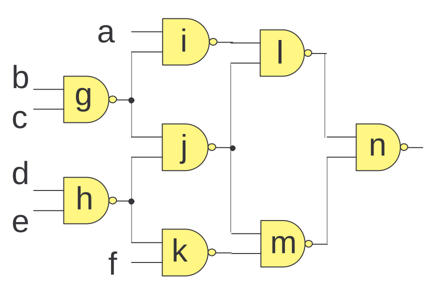
- C = (V, E, root) is a single-output directed acyclic circuit graph
  - V is a set of gates and primary inputs
  - E $\subseteq V \times V$ is the set of edges, describing the nets connecting the gates
  - root $\in$ V is the circuit output

Dominators in circuits

- Dominators provide a general mechanism to deal with re-converging paths in circuits
  - They give the precise starting and ending points of the path
- This can be used for scheduling the creation and elimination of auxiliary variables in many applications
  - Signal probability computation
  - Average switching activity computation
  - Equivalence and property checking (cut-points)
- Multiple-vertex dominators allow us to utilize this concept in a more practical way because they occur more frequently

Single-vertex dominators

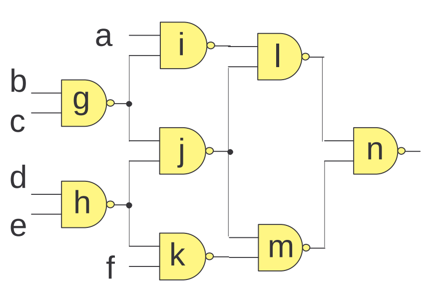
- A vertex v dominates another vertex u if every path from u to the root contains v

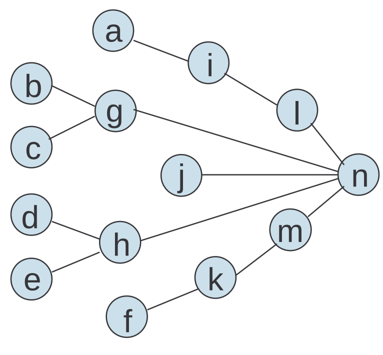


Immediate dominators

- Vertex v is the immediate dominator of u, denoted by v = idom(u), if v dominates u and every other dominator of u dominates v
- Every vertex v ∈ V except root has a unique immediate dominator
- The edges between idom(v) and v, for all v ∈ V, form a directed tree rooted at root, called the dominator tree T of C

Dominator tree example



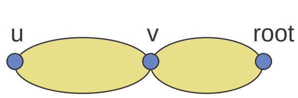


Computing single-vertex dominators

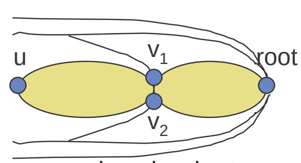
- Efficient O(|V| log|V|) algorithm for constructing the dominator tree exists (Lengauer and Tarjan,1979)
- Single-vertex dominators are generally too rare in circuits
- Usually, there are significantly more double- and triple-vertex dominators

Multiple-vertex dominators

- A set of vertices D = $\{v_1,\ldots,v_k\}$ is a multiple-vertex dominator for a vertex u, if:
  1. every path from u to the root contains some $v_i \in D$
  2. for every $v_i \in D$ there exist at least one path from u to the root which contains $v_i \in D$ and does not contain any other $v_j \in D$



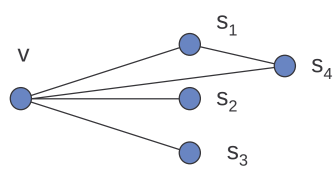
v dominates u



v_1 and v_2 dominate u

Properties of dominators

Property: If S is the set of successors of v, then there exist $S' \subseteq S$ which dominates v



$S = \{s_1, s_2, s_3, s_4\}$

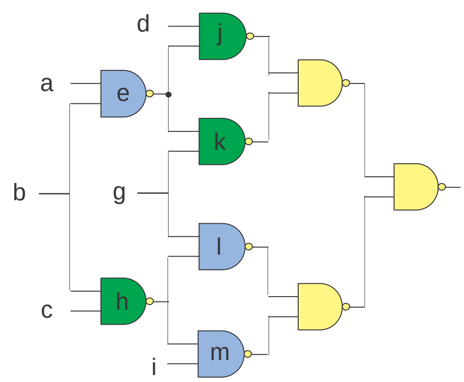
$S' = \{s_2, s_3, s_4\}$

S' dominates v

Immediate k-vertex dominators

- The set D_I is the immediate k-vertex dominator of u, if D_I dominates u and there is no other k-vertex dominator D of u such that every vertex of D is dominated by D_I
- For k > 2, immediate k-vertex dominators may not be unique
- No unique dominator tree exists

Example



Vertex b has two immediate 3-vertex dominators: **{e,l,m}** and **{h,j,k}**

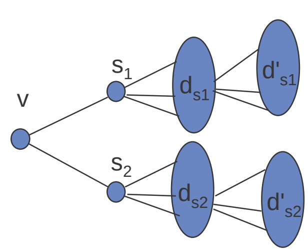
For k = 2, immediate dominators are unique

Algorithm 1 of computing all multiple-vertex dominators

- Candidate set of the multiple-vertex dominators of v is contained in the Cartesian product

$D_v \subseteq \{\{s_1\}, \{D_{s1}\}\} \times \{\{s_2\}, \{D_{s2}\}\} \times \ldots \times \{\{s_r\}, \{D_{sr}\}\}$

where $\{s_1\ s_2\ ,\ldots, s_r\}$ is the set of successors of v



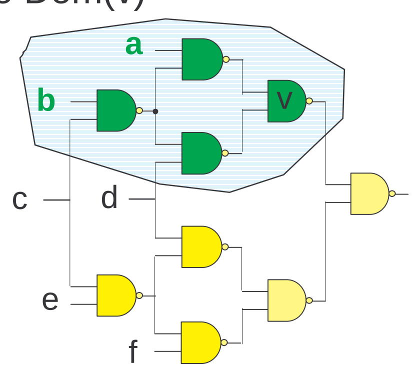
$\{s_1,s_2\}$, $\{s_1,d_{s2}\}$, $\{s_1,d'_{s2}\}$, $\{s_2,d_{s1}\}$, $\{s_2,d'_{s1}\}$,$\{d_{s1},d_{s2}\}$, …

Alternative approach

- Large size dominators are not useful generally
  - 2^k choices need to be examined for a k-vertex dominators
  - k = 2 and k = 3 are mostly interesting
- It is possible to efficiently compute dominators of a fixed small size k
  - Basic idea: Reduce the problem of computing the dominators of size k to the problem of computing the dominators of size k-1

Dominated vertices

- Let Dom(v) be a set of vertices dominated by v
- v belongs to Dom(v)



Algorithm II for computing all dominators of size k

For all vertices $v \in V$:

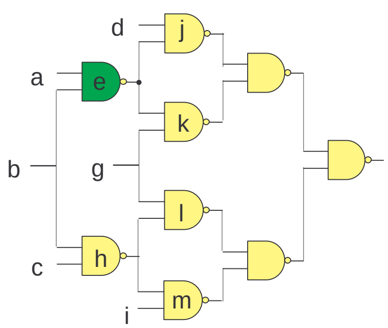
1. Reduce the circuit graph C = (V,E,root) with respect to v by removing from V all vertices dominated by v:

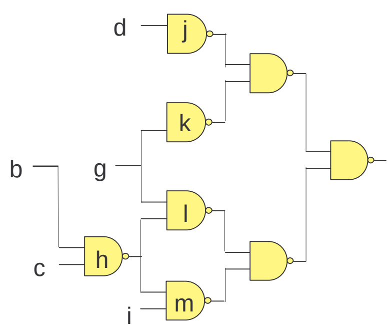
   V' = V - Dom(v)

   $E' = E - \{(u,w) \mid u \in Dom(v) \vee w \in Dom(v)\}$

2. Compute dominators of size k-1 for the resulting restricted graph C' = (V',E',root)
3. If D is a k-1-vertex dominator for $u \in C'$, then {D,v} is a k-vertex dominator for $u \in C$, provided D is not a k-1-vertex dominator for $u \in C$

Example





e dominates a:
- remove vertices a, e
- remove edges (a,e),(b,e),(e,g),(e,k)

h dominates b in the resulting circuit ⇒ {h,e} dominates b in the original circuit

Summary

- A polynomial-size algorithm for computing multiple-vertex dominators of a fixed size exists
- Useful for circuit synthesis and analysis

Open problems

- For multiple-vertex dominators, the edges $\{(idom(v),v) \mid v \in V - \{root\}\}$ generally form an acyclic graph, rather than a tree
- How to represent the set of all possible dominators efficiently is an open problem