Simultaneous Gate Sizing and Fanout Optimization

Wei Chen*, Cheng-Ta Hsieh+, Massoud Pedram*
***Department of EE-Systems**
University of Southern California
+Verplex Systems Inc.

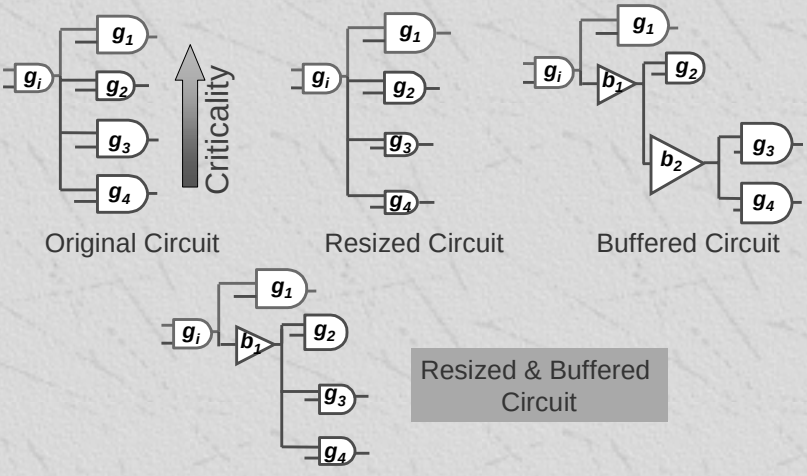
Outline

- Introduction
- Delay Model
- Algorithm
- Experimental Results
- Conclusions

Previous Work

- Gate Sizing Only
  - Discrete sizing: P. Chan ’90, O. Coudert ’96
  - Continuous sizing: A. Dunlop ’85, M. Berkelaar ’90
- Fanout Optimization Only
  - Discrete buffer sizes: H. Touati ’90, P. Cocchini ’98
  - Continuous buffer sizes: D. Kung ’98, P. Rezvani ’99

Motivation



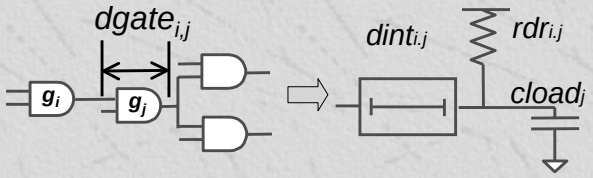
Original Circuit | Resized Circuit | Buffered Circuit

Resized & Buffered Circuit

Motivation (Cont'd)

- Interleaved Gate Sizing and Fanout Optimization, Y.Jiang '98
  - For each multi-pin net in the circuit, try out both gate sizing and buffer insertion, and implement the one that yields a better solution
- Integrated Gate Sizing and Fanout Optimization
  - This is the focus of our presentation

Gate Sizing Delay Model

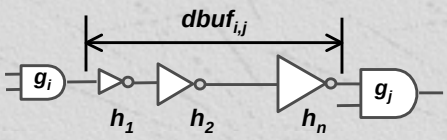


$$dgate_{i,j} = dint_{i,j} + rdr_{i,j}(z_j) \cdot cload_j$$

$$\text{where } cload_j = \sum_{g_k \in fanout(g_j)} cin_{j,k}(z_k)$$

The interconnect capacitance is ignored

Buffer Insertion Delay Model

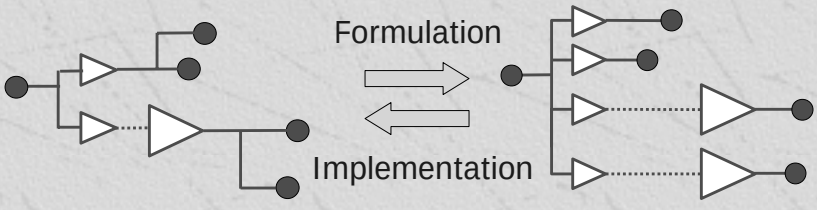


- Delay of buffer $d=\tau(p+g\cdot h)$ where p, g and h denote the intrinsic delay, logical effort and electrical effort, respectively
- Under a required time constraint on g_i, the load of g_i *is* minimized when $h_1=h_2=\ldots=h_n$
- The path delay of the optimal buffer chain is calculated as

$$dbuf_{i,j} = x_{i,j} \cdot (p + g \cdot h_{i,j})$$

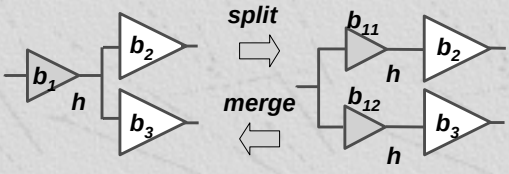
Buffer Tree Formulation

- Difficulty
  - Topology of buffer tree is unknown
- Solution
  - Recursively split the buffer tree into separate buffer chains

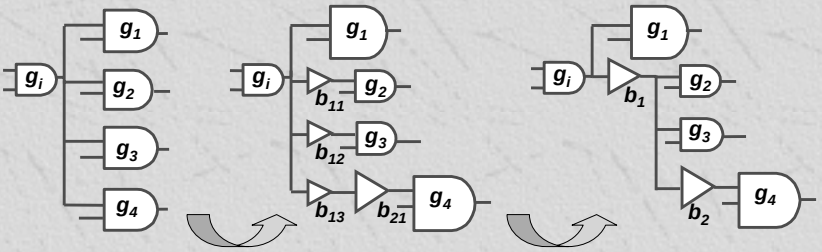


Merge and Split Transformations

- When gains of b_1, b_{11}, b_{12} are the same, the timing and input capacitance properties are preserved by the merge/split transformations



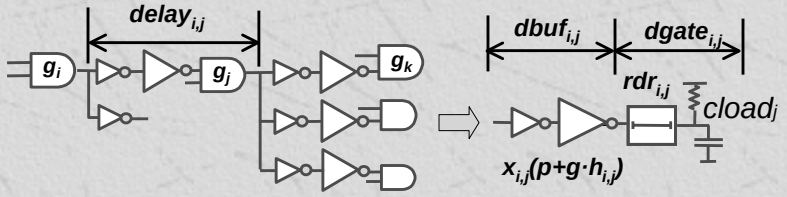
Buffer Tree Construction Example



Size the gates and build the buffer chains

Merge the individual buffer chains

Complete Delay Model for Simult. Gate Sizing and Fanout Optimization



$$delay_{i,j} = dbuf_{i,j} + dgate_{i,j}$$

$$dbuf_{i,j} = x_{i,j} \cdot (p + g \cdot h_{i,j})$$

$$dgate_{i,j} = dint_{i,j} + rdr_{i,j}(z_j) \cdot \sum_k \frac{cin_k(z_k)}{(h_{j,k})^{x_{j,k}}}$$

Global Problem Formulation

- $G(V,A)$: the circuit netlist
  - Vertex set V : gates in circuit
  - Edge set A : source to sink connections

minimize $cycleTime$

s.t. $a_i \geq T_{start}$ $\forall v_i \in PI$

$a_i \leq cycleTime$ $\forall v_i \in PO$

$a_j \geq a_i + dbuf_{i,j} + dgate_{i,j}$ $\forall (v_i, v_j) \in A$

- This problem formulation is solved in one step. No iterations are needed.

Global Problem Formulation (Cont'd)

- Exact formulation
- *n* gates and *e edges*
- *2(n+e)* variables
  - *2n*: 2 variables for each gate for the gate arrival time and the gate size
  - *2e*: 2 variables for each edge for the gain and the stage count in the buffer chain
- edge constraints
  - 1 timing constraint for each edge

Definition of Critical Section

- Refer to the set of *k* most critical paths in the circuit as the critical path set and denote it by *C(k)*
- Refer to all immediate non-critical fanout gates of the critical path set as the neighbor set and denote it by *Ne(k)*
- The critical section refers to the union of *C(k)* and *Ne(k)*

Critical Section Problem Formulation

minimize $cycleTime$

S.T. $a_i \geq T_{start}$ $\forall v_i \in PI, v_i \in C(k)$

$a_j \geq a_i + dbuf_{i,j} + dgate_{i,j}$ $\forall (v_i, v_j) \in A, v_i \in C(k)$

$a_i \leq a_i' + \delta \cdot slack_i'$ $\forall v_i \in Ne(k)$

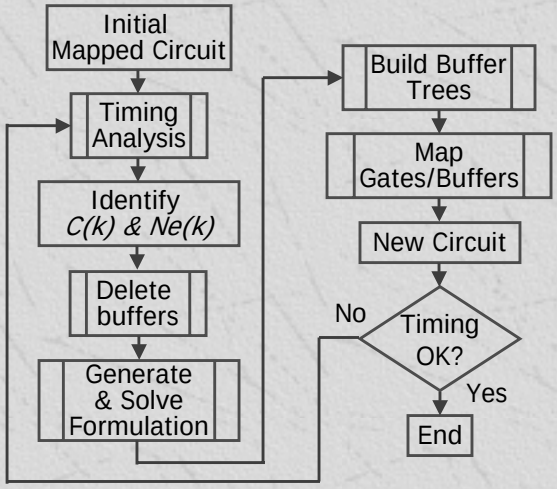
$a_i \leq cycleTime$ $\forall v_i \in PO, v_i \in C(k)$

- This problem formulation needs to be solved in a number of iterations. After each iteration, the circuit timing is recalculated.
- a_i', $slack_i'$: arrival time and slack time of g_i from previous iteration

Critical Section Problem Formulation (Cont'd)

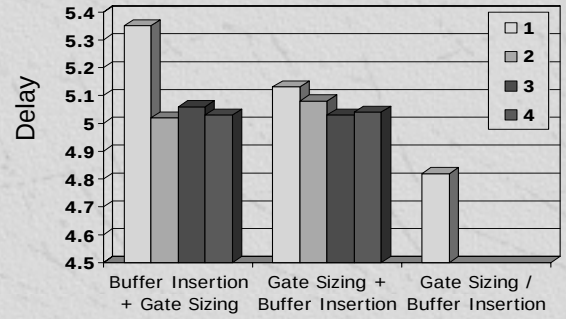
- Inexact formulation
- *n'* gates in *C(k)*, *m'* gates in *Ne(k)*, *e'* critical *edges*
- *2(n'+m'+e')* variables
  - *2(n' +m')*: 2 variables for each critical gate or neighboring gate for the gate arrival time and the gate size
  - *2e'* : 2 variables for each critical edge for the gain and the stage count in the buffer chain
- It is easy to control the problem size by changing *k*

Algorithm Flow



Experimental Results

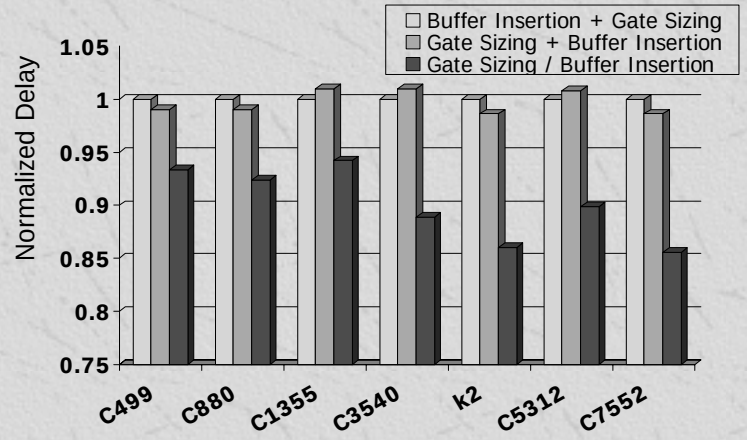
- 4 iterations sizing and buffering for the sequential flow
- One-shot solution of the simult. Sizing and buffering



C499 circuit

Results (Cont'd)

- Critical section formulation results



Conclusions

- Iteratively identify the timing-critical section in the circuit
- Perform simultaneous gate sizing and buffer chain insertion in the critical section
- Merge the buffer chains with similar gains to build the final buffer trees
- Experimental results show that the concurrent flow results in 9% improvement in the circuit delay compared to the "best" sequential flow