Buffer Block Planning for Interconnect-Driven Floorplanning

Jason Cong, Tianming Kong
& David Zhigang Pan

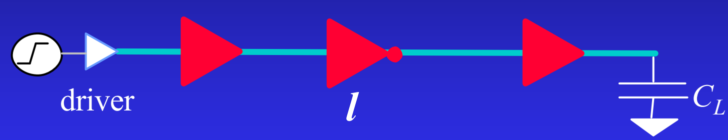
UCLA Computer Science Department
Los Angeles, CA 90095
Sponsored by SRC

Outline

- Introduction
- Feasible Region for Buffer Insertion
- Buffer Block Planning Algorithm
- Experimental Results
- Discussion

Buffer Insertion

- Interconnect dominates transistor delay for deep sub-micron designs.
- Buffer insertion is a very effective way to trade active devices for better interconnect performance, noise reduction, etc.



- Without buffer: delay µ l^2 or µ $l^2/W(l)$ [Cong-Pan'98]
- With opt. Buffers: delay µ l [Bakoglu'90; Otten-Brayton'98; Cong-Pan'98]

Demand of Buffers in DSM Design

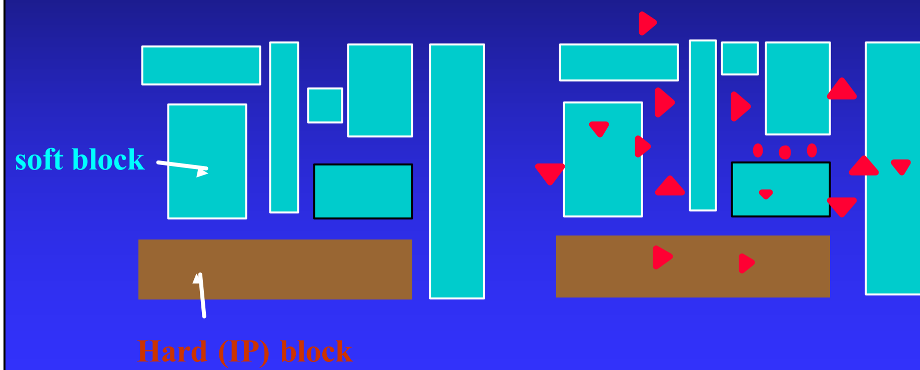
- For high-performance DSM designs, many buffers may be inserted to optimize/meet interconnect delay

Technology (um)	*0.25*	*0.18*	*0.13*	*0.10*	*0.07*
#buffer per chip	5k	25k	54k	230k	797k

Source: [Cong'97, SRC Work Paper]
http://www.src.org/research/frontier.dgw
(Data based on NTRS'97)

Need Buffer Planning

- The insertion of so many buffers will significantly change a floorplan; thus shall be planned ahead-of-time to ensure timing/design convergence.

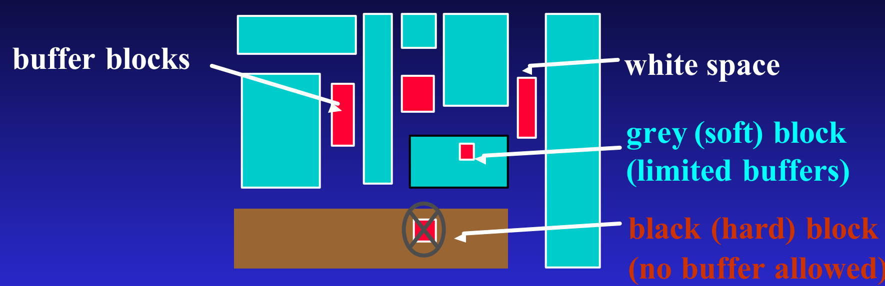


Limitation of Previous Works

- Buffer Insertion:
  - mostly done in a net by net manner after detailed placement
  - mostly no obstacles (hard IP blocks, etc) considered
  - no global buffer planning (only manual or semi-manual planning)
    - buffers are distributed in almost random manner across the entire chip
    - complicate global/detailed routing and p/g networks

Solution: Buffer Block Planning with Floorplanning

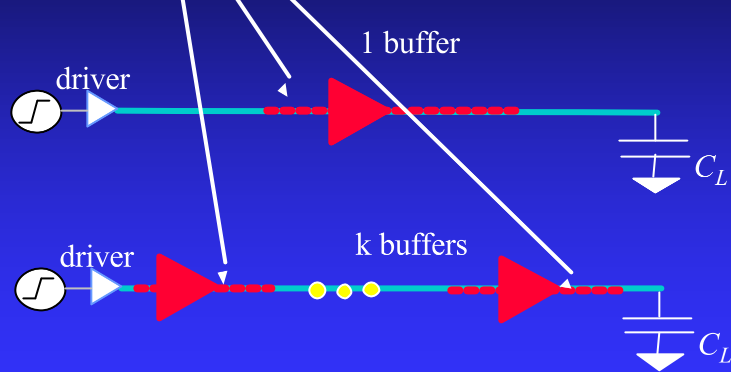
Buffer Block Planning Problem



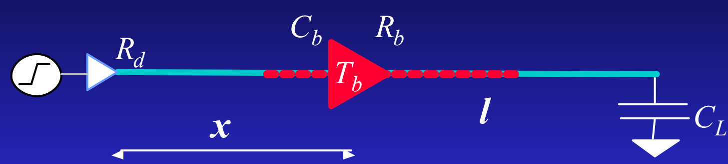
- **Given**: (1) initial floorplan, (2) buffer capacity for each soft block/white space, (3) and performance constraint for each net.
- **Output**: "optimal" location/dimension of buffer blocks such that the overall chip area and the number of buffer blocks are minimized.

Feasible Region for BI

- **Feasible region** is the maximal region that a buffer can be placed to meet given delay constraint.



Feasible Region for One Buffer

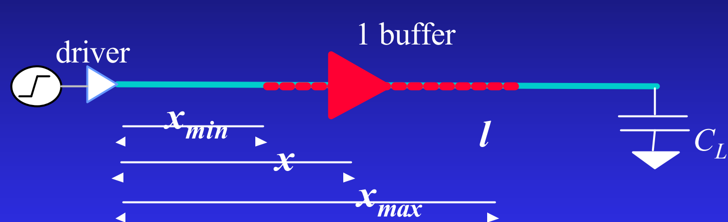


- **Feasible region condition for *x***

$$T_0(R_d, x, C_b) + T_b + T_0(R_b, l - x, C_L) \le T_{req}$$

Feasible Region for One Buffer

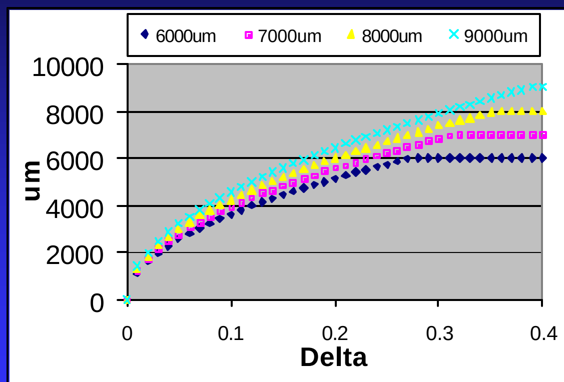
- **We obtain closed-form formula of FR for inserting one buffer to meet delay constraint**



$$x \in [x_{min}, x_{max}]$$

Crucial Observation for FR

- **Even under tight delay constraint, FR for BI can still be pretty large!**

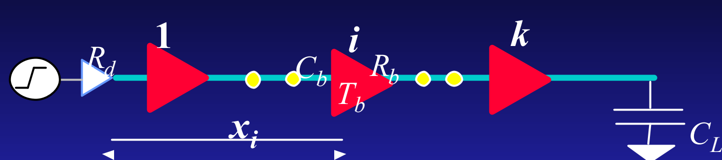


❖ **Delay budget is (1+Delta) T_{opt} (the best delay by optimal buffer insertion)**

Delta	FR
1%	19%
5%	43%
10%	60%
20%	86%

=> FR provides a lot flexibility to plan buffer location

Extension I: FR for Multiple Buffers



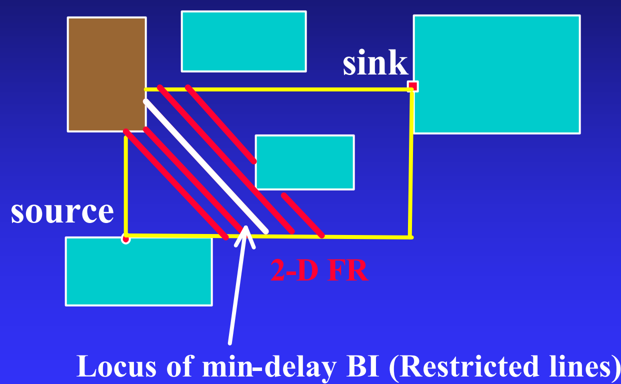
- **Feasible region condition for the *i*-th buffer x_i**

$$T_{i-1}(R_d, x_i, C_b) + T_b + T_{k-i}(R_b, l - x_i, C_L) \le T_{req}$$

- **More complicated, but still closed-form solution for FR can be obtained.**
- **We also obtain the minimum number of buffers k_{min} needed to meet delay constraint**

Extension II: 2D Feasible Region

- FR extended to 2-dimension with obstacles



Outline

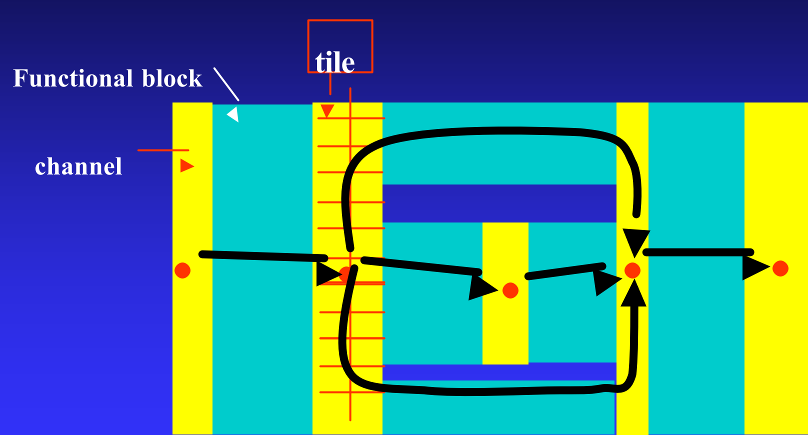
- Introduction
- Feasible Region for Buffer Insertion
- Buffer Block Planning (BBP) Algorithm
- Experimental Results
- Discussion

Overall Picture: BBP for Interconnect-Driven Floorplanning

- For each floorplan (*FL*) configuration
  - Apply BBP on the given *FL*
  - Evaluate *FL* (timing, area, #BB tradeoff, etc.)
- Return the best *FL* solution

Graph Representation

- Polar graph and tile structure



Overview of BBP Algorithm

1. Build polar graph/tile for given floorplan *FL*;
2. For each tile, compute its area slack;
3. Compute FR for each buffer of each net;
4. While (there exists some buffer to be inserted) {

    Pick_A_Tile t;

    Insert_Buffers into t;

    Update chip dimension, FR, area slack, etc.

  }

Two Key Steps in BBP

Pick_A_Tile t : two modes

1. If there exists dead area: pick the t that can insert maximum # of buffers without chip area increase;
2. Otherwise: pick the t that has maximum BI demand

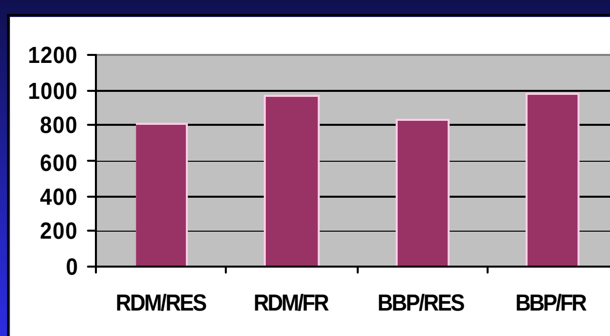
Insert_Buffers into t : also two modes

1. If t has dead area: insert buffers whose FR's intersect with t (most constrained ones first) => Buffer Blocks;
2. Otherwise: insert only one buffer into t => minimize the area increase

Experimental Setting

- Two Algorithms:
  - RDM: a buffer is randomly assigned to any feasible location (no planning)
  - BBP: buffer block planning
- Two Scenarios:
  - RES: restricted buffer insertion position(s) as to minimize delay
  - FR: feasible buffer region as to meet delay constraint
- 6 MCNC + 5 randomly generated circuits (0.18um tech)
  - Delay budget randomly assigned to be 1 to 1.2 x T_{opt}

#nets that meet delay constraints

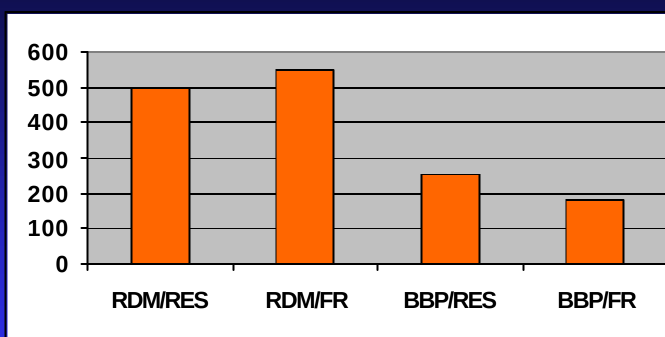


FR provides a lot more flexibility than RES to avoid obstacles during BI => can better meet delay constraints

Normalized Chip Area after BI

Circuit	RDM/RES	RDM/BBP	BBP/RES	BBP/FR
apte	1.000	1.003	0.993	0.992
xerox	1.000	1.003	0.992	0.995
hp	1.000	1.003	0.990	0.988
ami33	1.000	1.005	0.986	0.985
ami49	1.000	1.002	0.981	0.977
playout	1.000	1.003	0.975	0.972
ac3	1.000	1.003	0.988	0.986
xc5	1.000	1.000	0.984	0.977
Hc7	1.000	1.001	0.966	0.957
a9c3	1.000	1.002	0.975	0.966
Pc2	1.000	1.000	0.947	0.936

Comparison of #BB



BBP reduces #BB from RDM by a factor of up to 3x;
BBP/FR further reduces #BB from BBP/RES by up to 34%

Conclusion

- First introduce the concept of FR
- Obtain closed-form solution for FR for two-pin nets
- FR provides a lot more flexibility for buffer planning
- BBP algorithm for both area and number of BB minimization
- BBP/FR provides the best solution