Low Power Gated Bus Synthesis using Shortest-Path Steiner Graph for System-on-Chip Communications

Renshen Wang[1]
Nan-Chi Chou[2]
Bill Salefski[2]
Chung-Kuan Cheng[1]

[1]University of California San Diego

[2] Mentor Graphics Corporation

Goal of this talk

- Power saving techniques on bus communications
  - AMBA bus architecture
  - Steiner tree → Steiner graph
- Minimal power for on-chip communications under practical constraints
  - Fixed placement of components
  - Limited routing resource
  - Compatible with existing protocols

Introduction

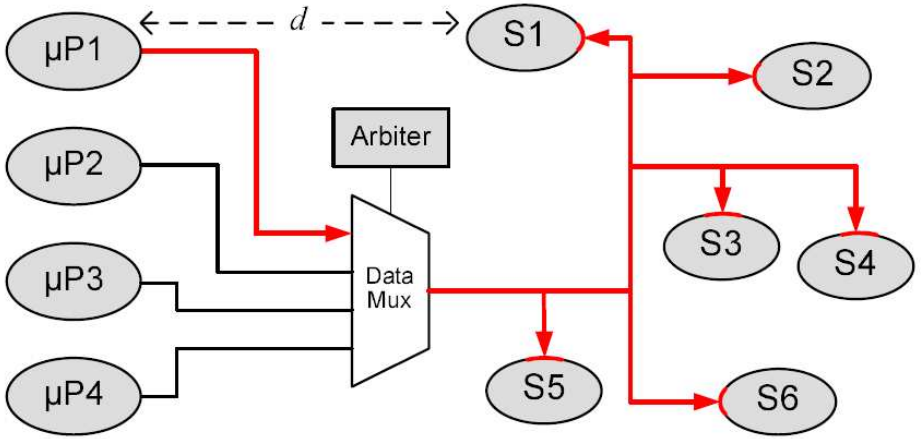
- Low power design
- For battery life of embedded systems
- Lower CO_2 emission and environmental impact

On-chip Bus

- Connection bridge of System-on-Chips
  - Global connection, inter-module
  - Scaling up (relatively, to Moore's law)

    "The future of wires" by Ho, Mai, & Horowitz. *Proceedings IEEE*, 89:490-504, 2001
- Consuming about **15%** of system power
  - "Power analysis of system-level on-chip communication architectures" by Lahiri & Raghunathan. *CODES* 2004

Bus Power

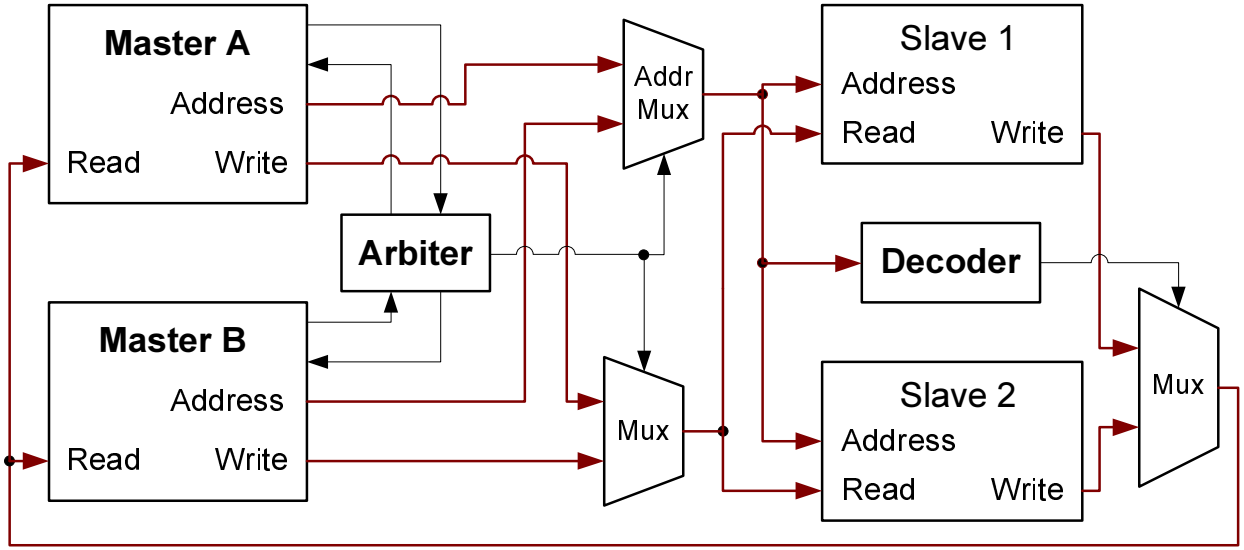


- Microprocessor 1 sending message to slave device 1
- What is necessary
- What is being taken by current bus architectures
- Low power efficiency on
  - Wires (14% of power on communication)
  - Component interfaces (35% of power)

Power Saving

- Clock gating
  - Masking off unused components
  - Dynamic power
- Power gating
  - Turning off unused components
  - Leakage power
- Bus gating
  - Dynamic power

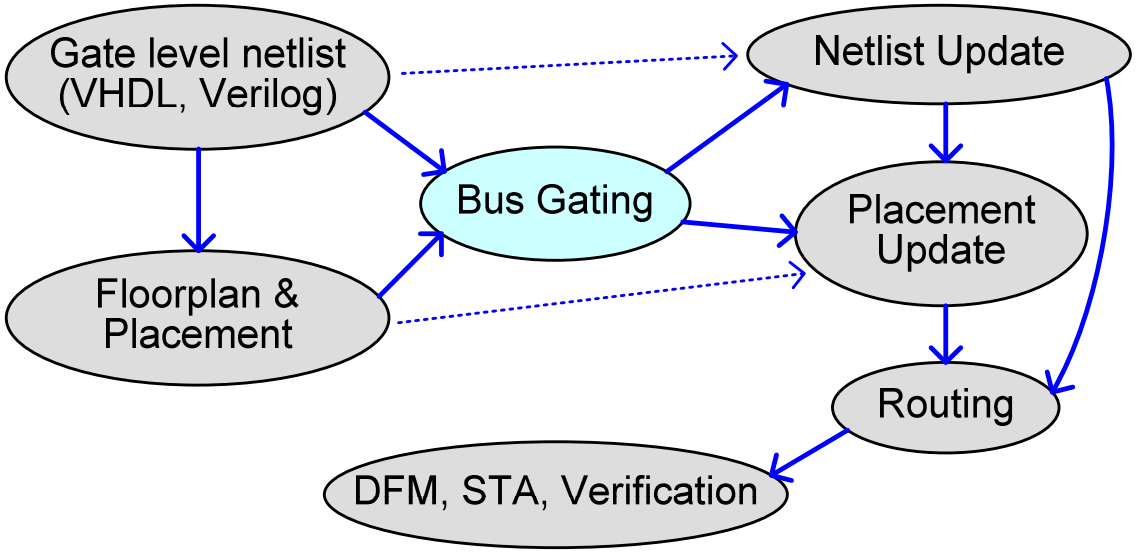
On-chip Bus Architectures



- AMBA Ahb protocol
  - Also in IBM CoreConnect, Altera Avalon, *etc*
- Pros: Simplicity
- Cons: Low efficiency on power and wires

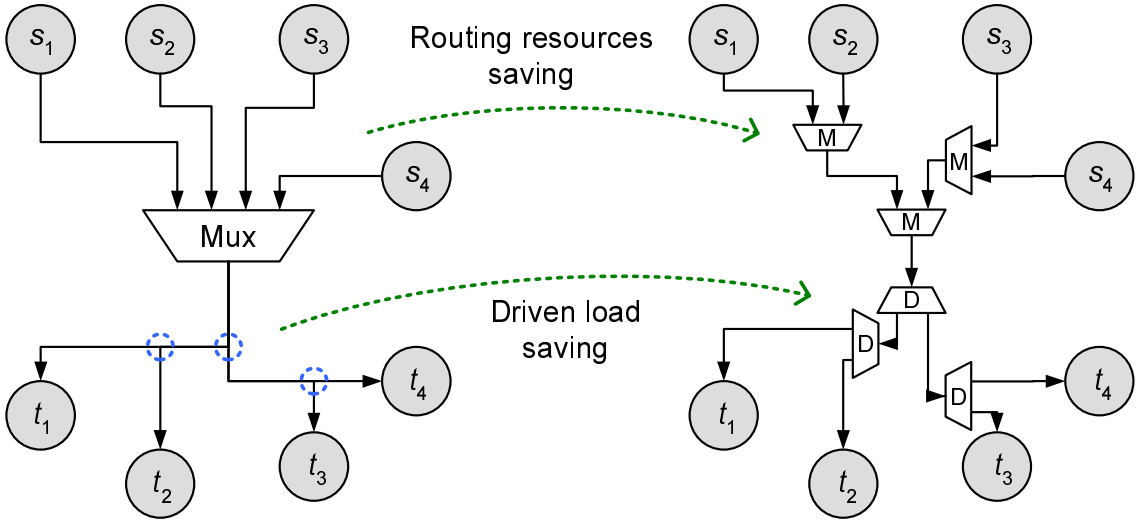
Objective of Improvement

- Greatly reduce bus power consumption
- Not greatly consume other resources
- Bus gating: may result in increased complexity in design flow



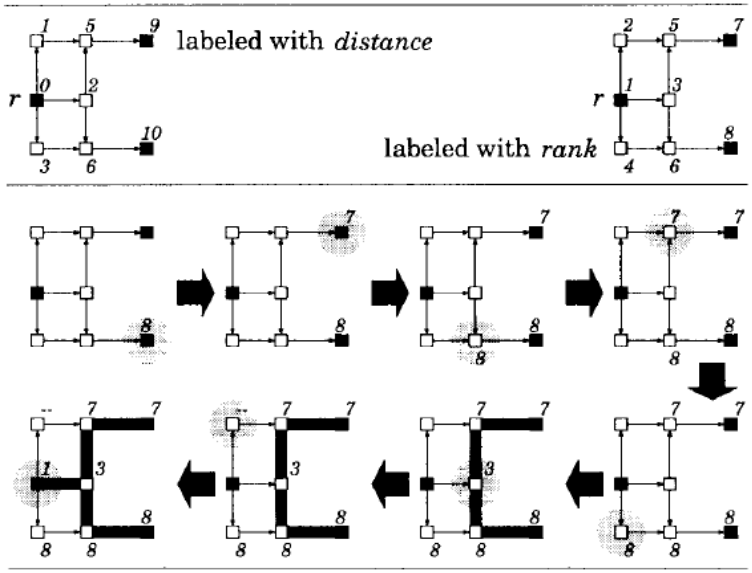
Step 1: Steiner Tree

- Use distributed multiplexers
- Add distributed de-multiplexers
  - Minimum rectilinear Steiner arborescence (MRSA)
  - Arborescence = Shortest-path Steiner tree



MRSA Construction

- RMST, NP-complete
- MRSA, NP-complete (correction to the paper)
- Basic RSA/G heuristic for MRSA
  - Merge subtrees such that the merging point is as far from source as possible



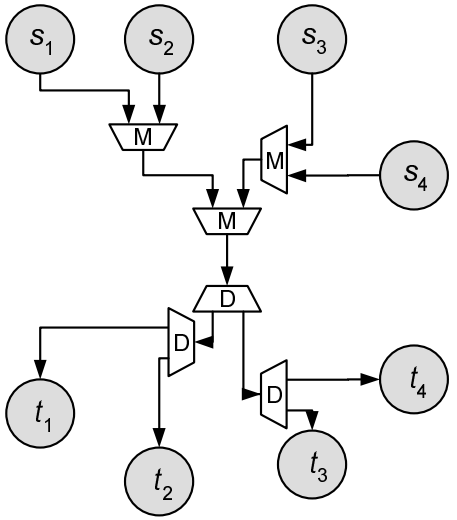
Given a source s and n terminals $t_1, \cdots, t_n$, $v_1, \cdots, v_N$ are the Hanan grid nodes sorted by $\Delta_s(v_1) > \cdots > \Delta_s(v_N)$;

$P \leftarrow \phi$;
for $i = 1$ to N do
  if there is t_j at v_i, then (**TMO**)
    $P \leftarrow P \bigcup \{v_i\}$;
  $X \leftarrow P \bigcap \{v_j | \Delta_s(v_j) = \Delta_s(v_i) + \Delta(v_i, v_j)\}$;
  if ($|X| \geq 2$) then (**SMO**)
    merge the nodes in X rooted at v_i
    $P \leftarrow (P \bigcap \overline{X}) \bigcup \{v_i\}$;
return the arborescence rooted at s;

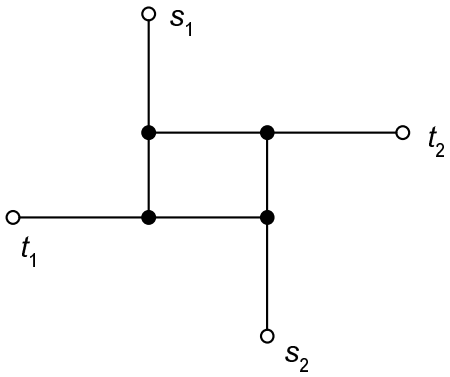
MRSA Construction (cont.)

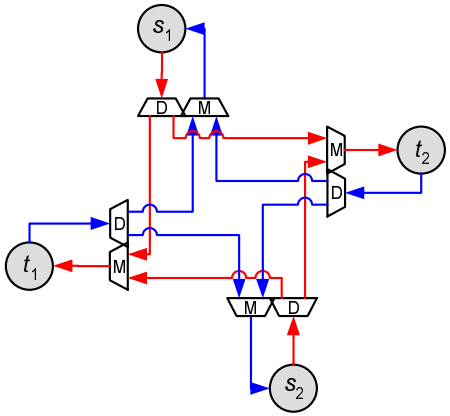
- Heuristics are explored in
  - "Efficient algorithms for the minimum shortest path Steiner arborescence problem…" by Cong, Khang, & Leung. IEEE *TCAD* 1998
- k-IDeA (iterated *k*-deletion for aborescence)
  - Remove up to k nodes when running RSA/G
  - The best set of skipped nodes are marked as permanently deleted
  - Repeat iterations until no further improvement
  - 2-IDeA has the best overall performance

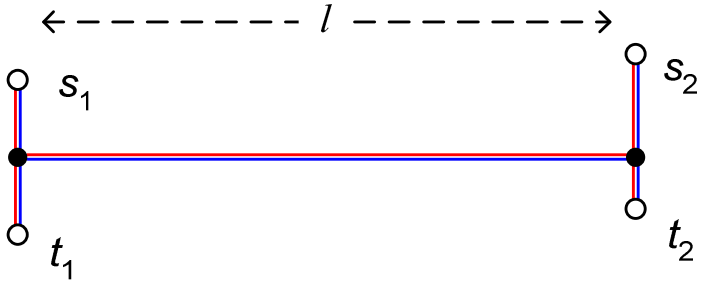
Step 2: Steiner Graph



- Single arborescence
  - Not good enough for all
- Shortest-path Steiner graph
  - Shortest path between each master-slave pair
  - Wire length may increase (but not necessarily)

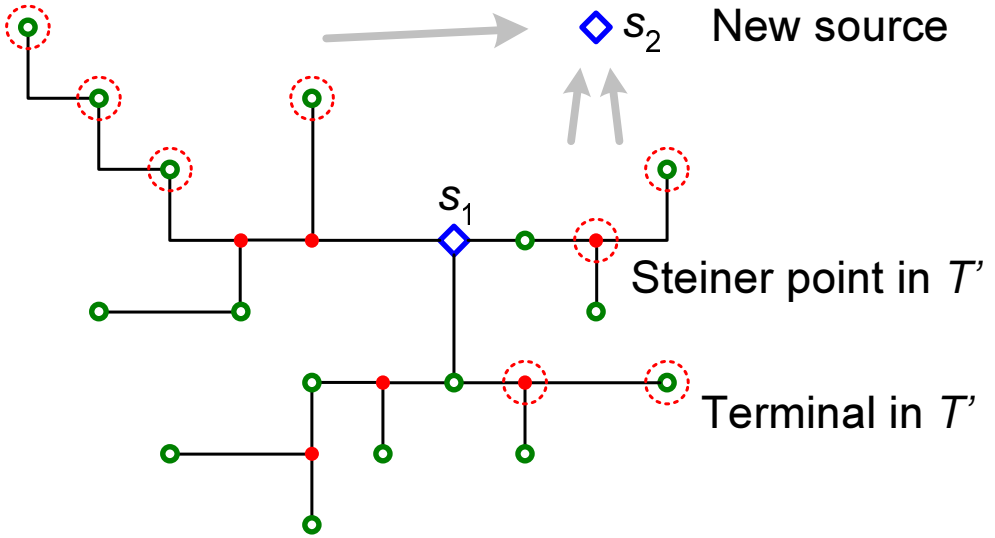






Shortest-path Steiner Graph

- Multiple arborescences
  - By multiple MRSA constructions



- Wires can be shared
  - Starting from the second arborescence

- Example with 2 sources
  - 8 terminals need to be connected for s_2

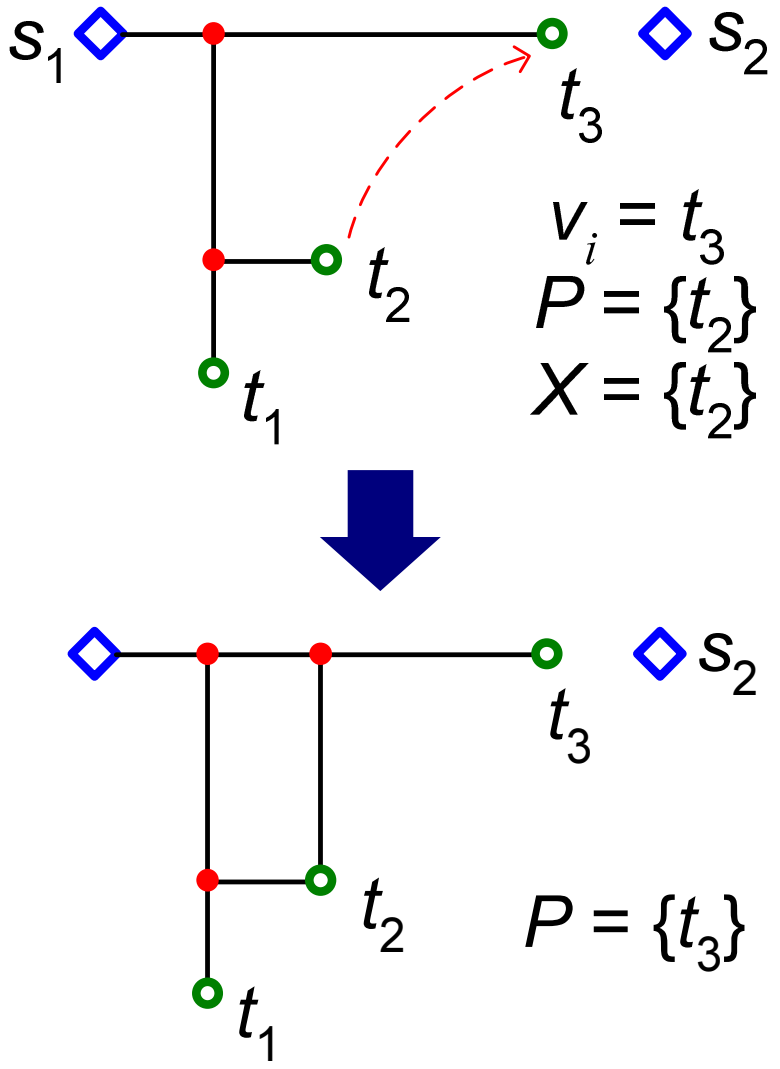
SPSG Construction

- Revised RSA/G for SPSG

Given existing Steiner graph G, source s_k, terminals $t_1, \cdots, t_n$, and $v_1, \cdots, v_N$ are same as in RSA/G;

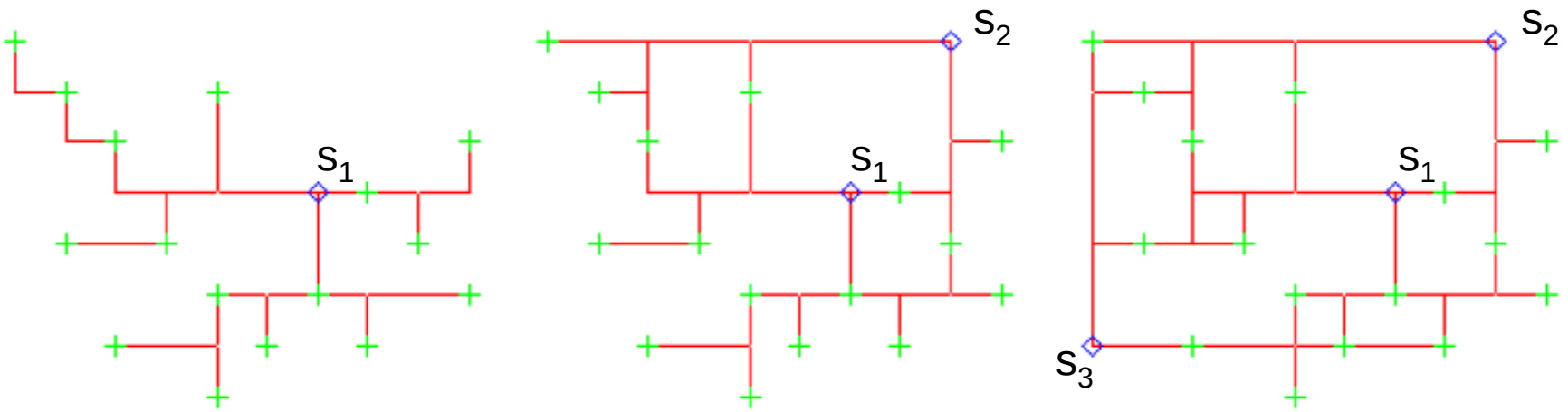
Routine Necessitate(vertex v);
  $U \leftarrow \{u \in G$ and exists a wire path from v to u of length $\Delta_{s_k}(v) - \Delta_{s_k}(u)\}$;
  $T' \leftarrow T' \bigcup \{u_m \in U$ with minimum $\Delta_{s_k}(u)\}$;

$T' \leftarrow \phi$;
for $i = 1$ to n do Necessitate(t_i);
$P \leftarrow \phi$;
for $i = 1$ to N do
  if $v_i \in T'$ then $P \leftarrow P \bigcup \{v_i\}$; **(TMO)**
  $X \leftarrow P \bigcap \{v_j | \Delta_{s_k}(v_j) = \Delta_{s_k}(v_i) + \Delta(v_i, v_j)\}$;
  if ($|X| \geq 1$ and $v_i \in G$) then **(SMO)**
    for each ($u \in X$) connect(v_i, u);
    $P \leftarrow P \bigcap \overline{X}$;
    Necessitate(v_i);
  else if ($|X| \geq 2$) then **(SMO)**
    merge the nodes in X rooted at v_i
    $P \leftarrow (P \bigcap \overline{X}) \bigcup \{v_i\}$;
return; (the MRSA rooted at s_k is added to G)



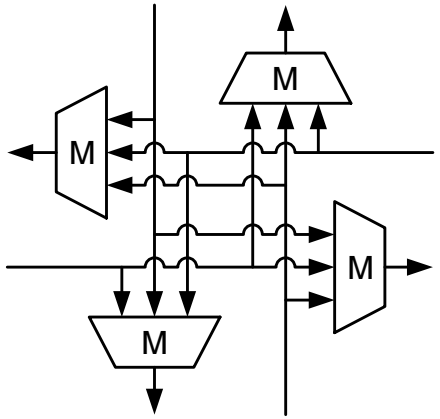
SPSG Construction (cont.)

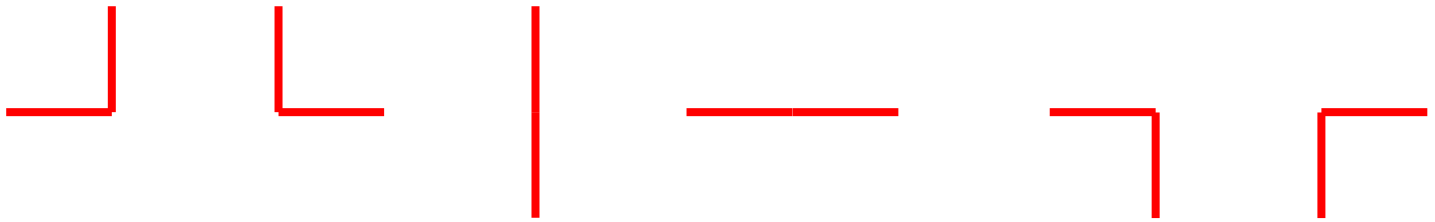
- k-IDeA (iterated *k*-deletion for aborescence)
  - Same iterations as in MRSA construction
  - To construct each aborescence
- Run k-IDeA iterations on each source s_i



Switch Control on Graph Nodes

- Each node has degree 3 or 4
  - Needs a switch to guide signals
- A switch of node degree 4 →
- A path through a 4-way switch
  - C(4,2) = 6 possible cases

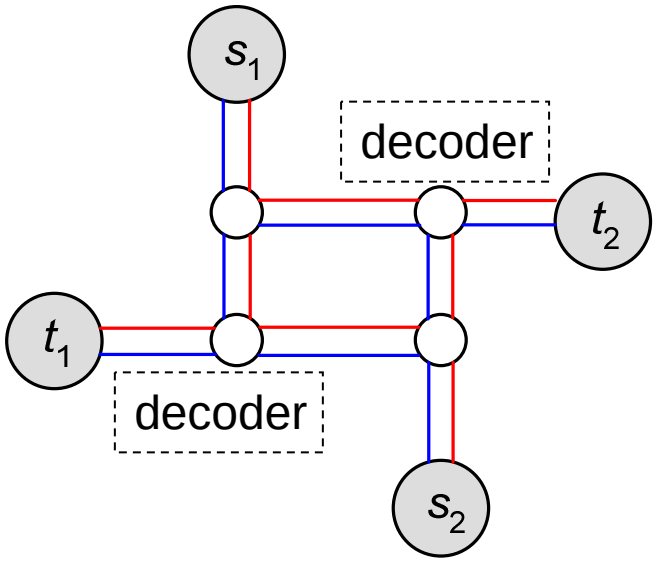




  - May need 3 control signals

Reducing Control Signals

- Path guidance requirement
  - Paths only exist between master-slave pairs
  - Actual cases can be less than C(4,2) or C(3,2)
- Decoder duplication
  - Reduce wire at the cost of silicon resource
  - Not adopted in our experiments
- More techniques…



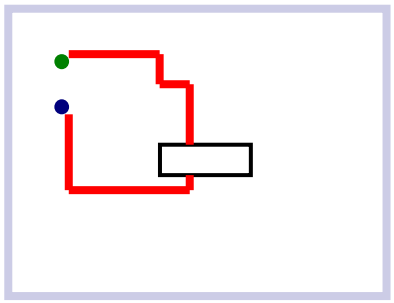
Experimental Results

- Average wire capacitance in transmission

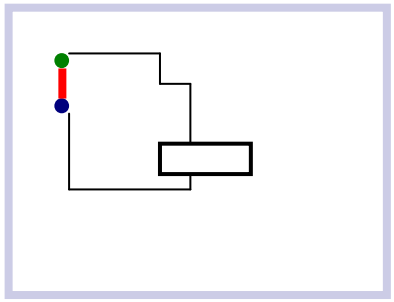
Table 3: Average data transaction wire capacitance

Case(m,n)	AHB	MRSA	SPSG	P_{save}
T0 (1,16)	5934	1638	469	72.4%~92.1%
T0 (2,16)	7584	2138	613	71.8%~91.9%
T0 (3,16)	8418	2038	635	75.8%~92.5%
T1 (3,16)	14550	4067	1754	72.0%~87.9%
T2 (2,30)	12305	2074	669	83.1%~94.6%
T3 (3,16)	8230	2019	691	75.5%~91.6%
T4 (5,15)	9674	1862	689	80.8%~92.9%
T5 (6,16)	11210	2012	694	82.1%~93.8%
T6 (8,8)	8952	1945	689	78.3%~92.3%
T7 (12,6)	12107	1933	657	84.0%~94.6%
T8 (16,10)	14377	1905	683	86.7%~95.2%
T9 (8,16)	11652	1883	618	83.8%~94.7%
T10 (8,16)	12899	2103	749	83.7%~94.2%
T11 (6,12)	8970	1897	668	78.9%~92.6%
T12 (12,12)	13242	1936	669	85.4%~94.9%

$$\frac{C_{SPSG} - C_{AHB}}{C_{AHB}}$$



MRSA



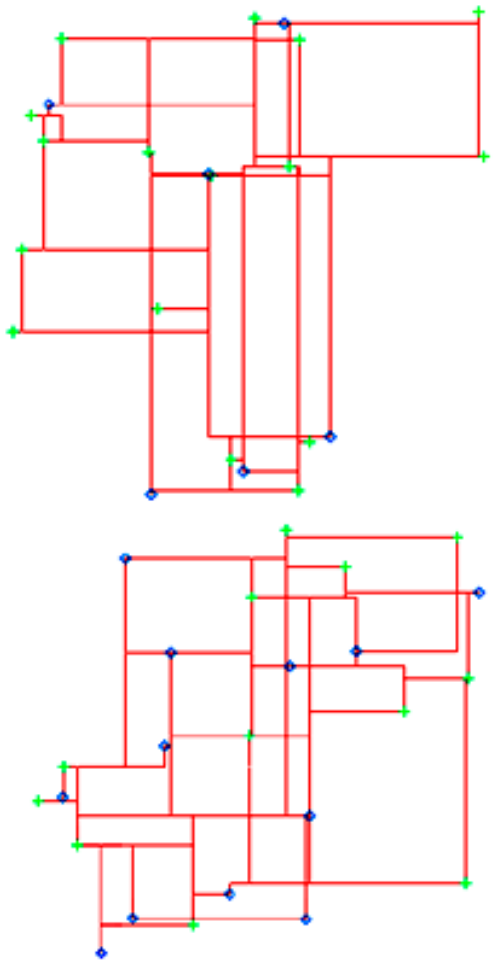
SPSG

Experimental Results (cont.)

- Impact on Total Wire Length

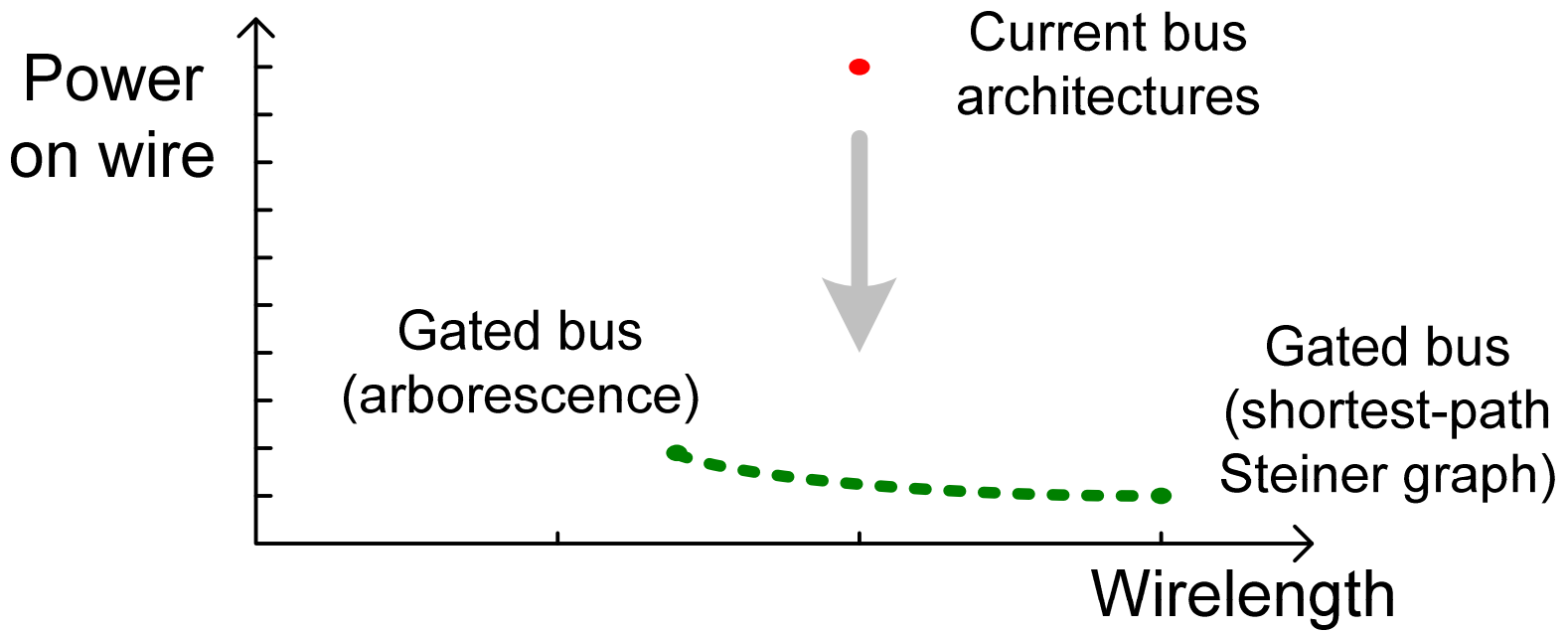
Table 4: Total wirelength

Case(m,n)	AHB	MRSA	SPSG	$L_{increment}$
T0 (1,16)	11200	10316	9656	-7.9%∼-13.8%
T0 (2,16)	14000	11850	14078	−15.4%∼0.6%
T0 (3,16)	16000	13350	21697	−16.6%∼35.6%
T1 (3,16)	27000	15131	18238	-44.0%∼-32.5%
T2 (2,30)	23257	16386	24535	−29.5%∼5.5%
T3 (3,16)	15248	11554	14274	-24.2%∼−6.4%
T4 (5,15)	17840	13729	22792	−23.0%∼27.8%
T5 (6,16)	20988	15778	35287	−24.8%∼68.1%
T6 (8,8)	16623	15800	25297	−5.0%∼52.2%
T7 (12,6)	22640	13227	27113	−41.6%∼19.8%
T8 (16,10)	27316	17018	40844	−37.7%∼49.5%
T9 (8,16)	22183	16055	35267	−27.6%∼59.0%
T10 (8,16)	24173	15906	34337	−34.2%∼42.0%
T11 (6,12)	16920	12566	22070	−25.7%∼30.4%
T12(12,12)	25369	17256	38305	−32.0%∼51.0%



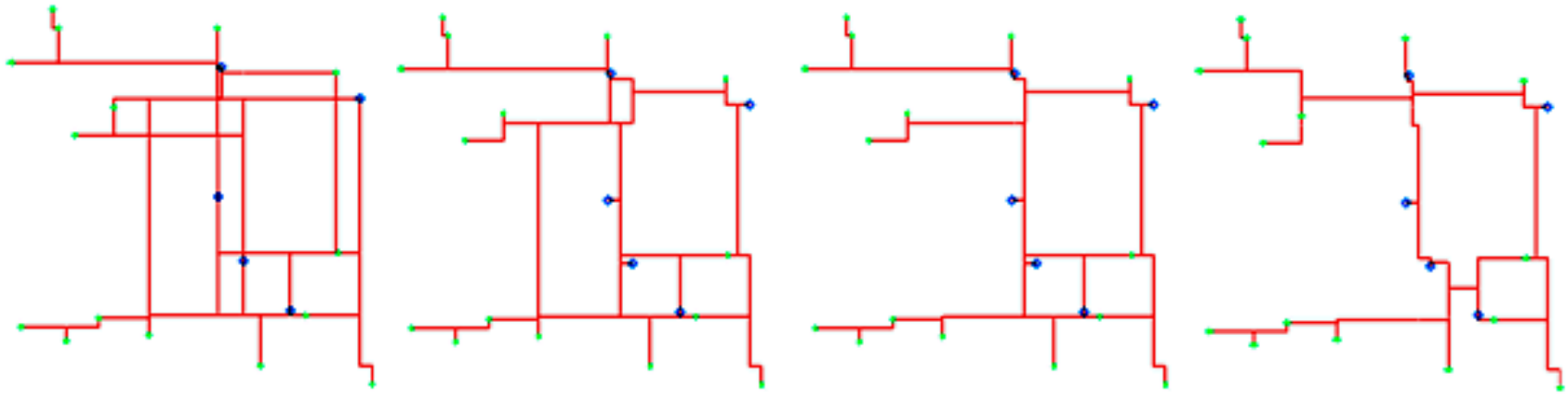
Tradeoff between power & wire

- Power reduction curve
  - Depending on routing resource
  - Always large improvement



Future Works

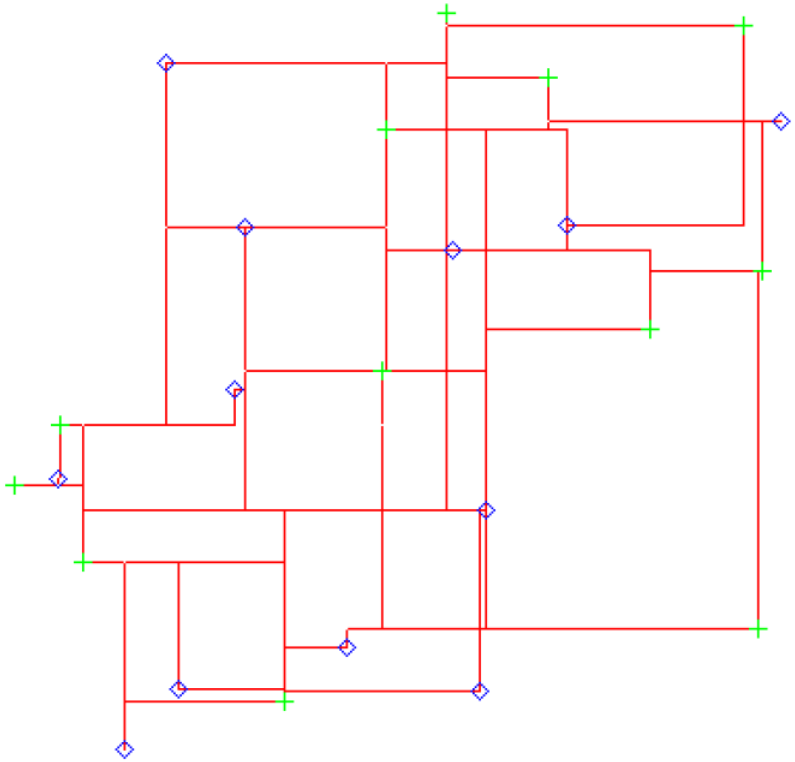
- Wirelength reduction
  - Compress narrow rectangles into lines

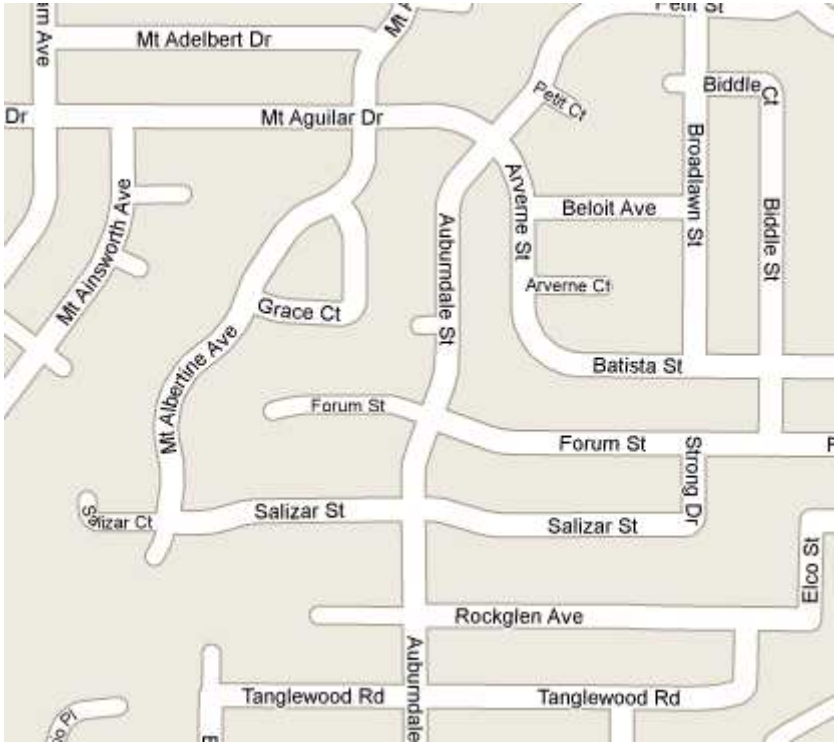


- Bandwidth extension (bus matrix)
  - Allow multiple access
  - Power-bandwidth co-optimization

Some Insights

- What does this gated bus look like?
  - A map
  - Streets created for efficient commuting





Questions?

- Thank you for your attention!

  - This research is under the support of NSF CCF-0811794 and California MICRO program