Round-robin Arbiter Design and Generation

Eung S. Shin

Prof. Vincent J. Mooney III

Prof. George F. Riley

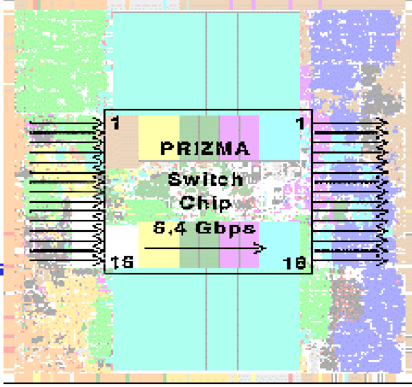
Electrical and Computer Engineering

Georgia Institute of Technology

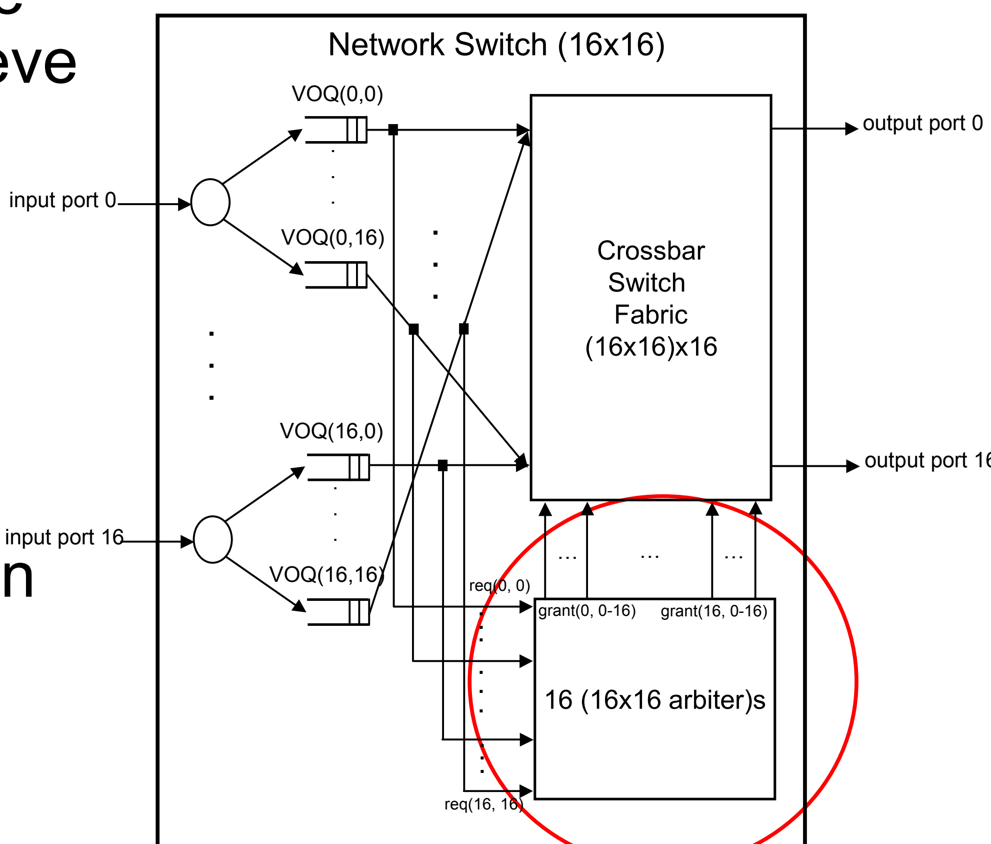
Outline

- Introduction
- Terminology
- Related Work
- Bus Arbiter (BA) Design
- Switch Arbiter (SA) Design
- Round-robin Arbiter Generator (RAG)
- Comparison with other Switch Arbiters
- Conclusion

Introduction



- As the number of bus masters increases in a single chip, the importance of fast and powerful arbiters commands more attention.
- A fast arbiter is one of the dominant factors to achieve terabit switching speeds.
- To design with high performance and fairness in arbitration is a tedious and error-prone task.
- Our goal is to provide a fast and fair arbiter design with a tool for automatic generation.

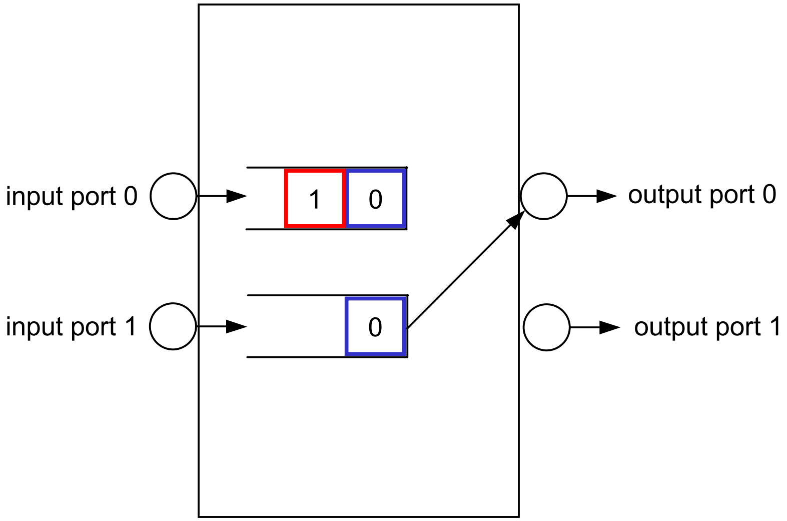


Terminology

- **MxN Switch**: M-input by N-output switch.
  - Example: A 32x32 switch is a 32-input by 32-output switch with 1024 (32^2) possible connections between input ports and output ports.
- **Virtual Output Queues (VOQs)**: there are VOQs in a switch to remove possible output port contention (Head of Line (HOL) blocking).
- **VOQ (*m*, *n*)**: *m* is the input port index and *n* is the output port index.
  - Example: VOQ (*1*, *0*) is the VOQ of input port 1and queues packets destined to output port 0.

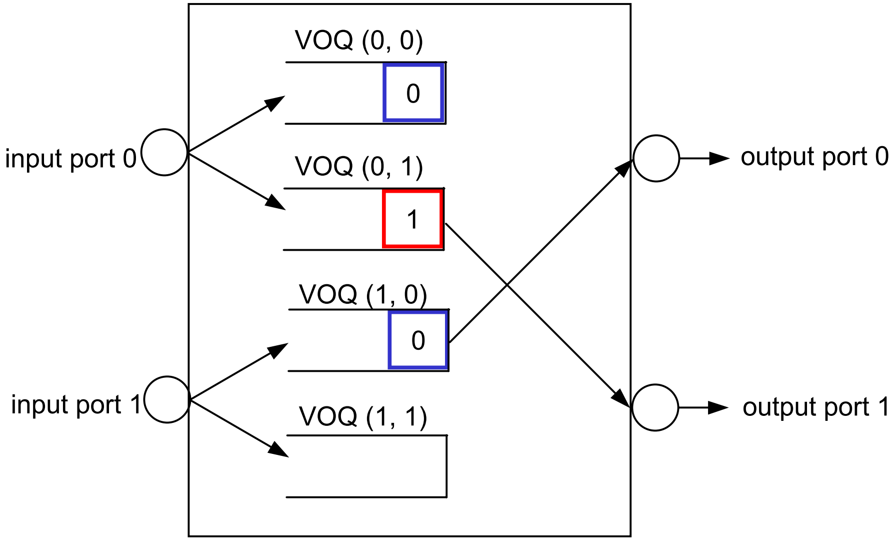
HOL Blocking Example

Without VOQs



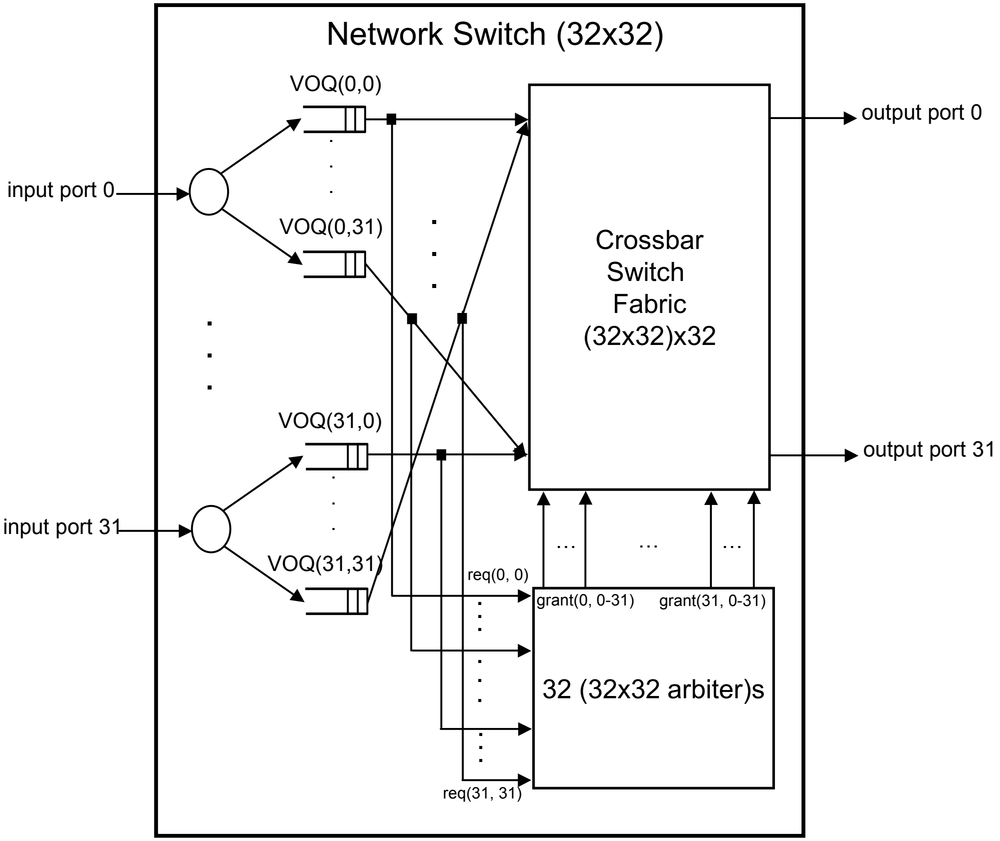
HOL Blocking Example

With VOQs



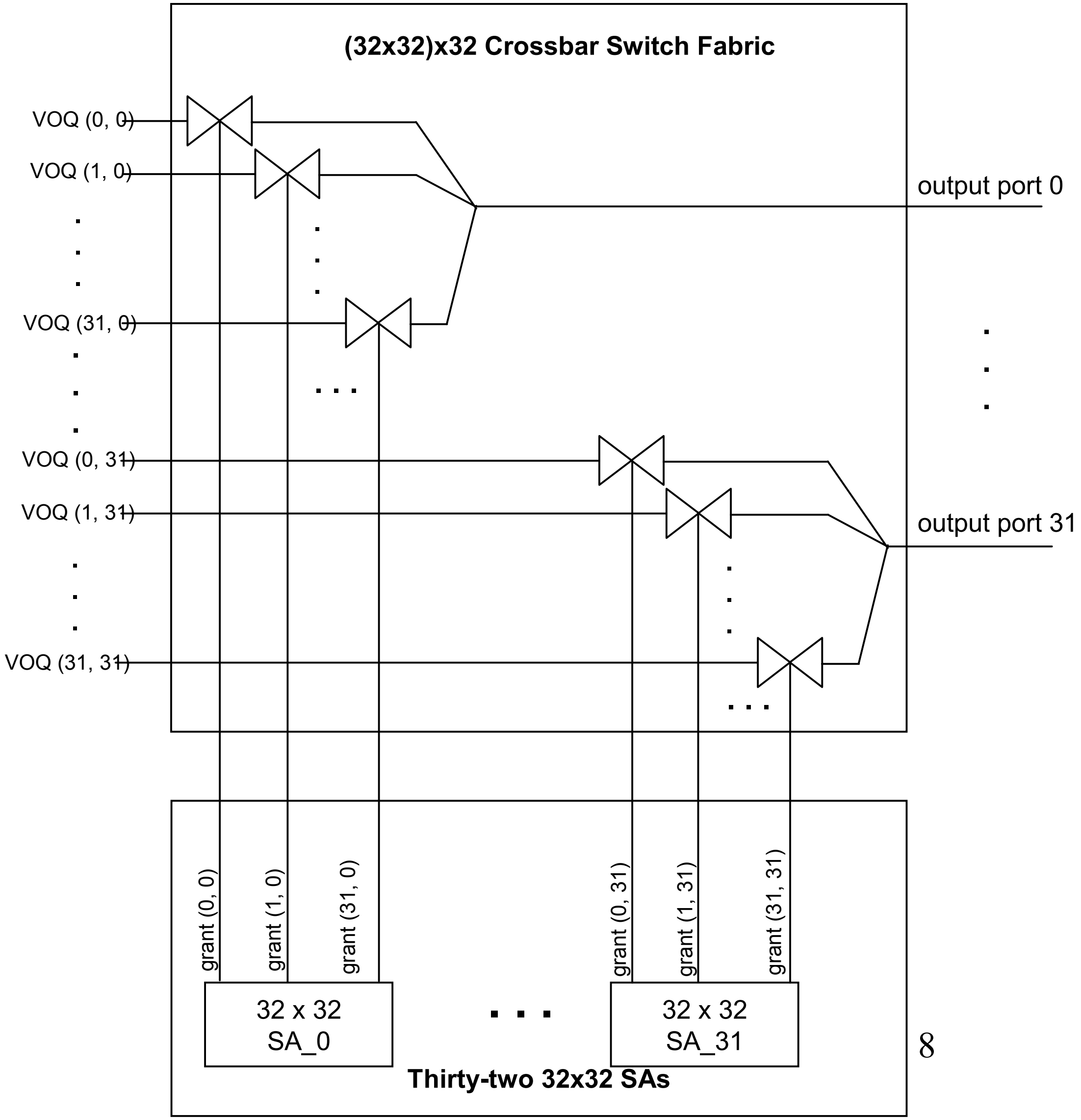
Terminology (Continued)

- **(MxV)xN Switch**:
  - M is the number of input ports of an MxN switch.
  - V is the number of VOQs per input port.
  - N is the number of output ports of an MxN switch.
  - Typically, V is equal to N.
  - The total number of VOQs in an MxN switch is M∗N.



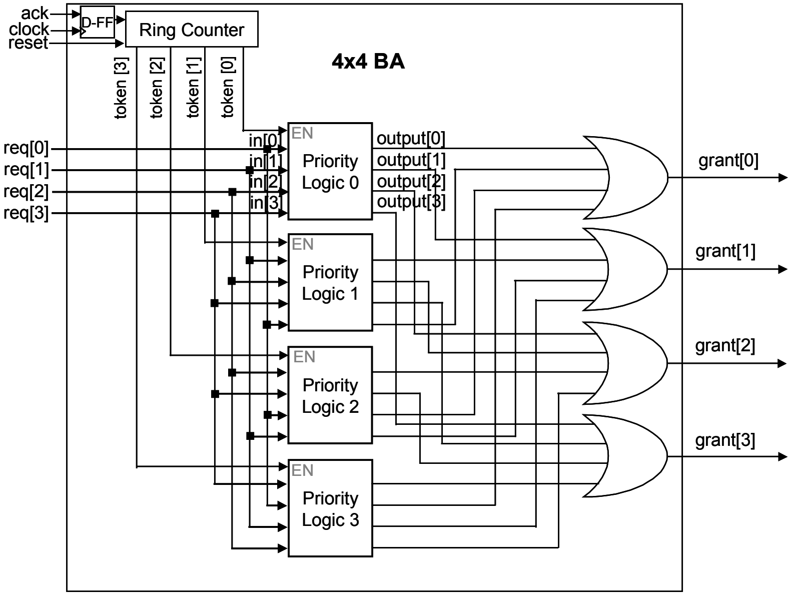
Terminology (Continued)

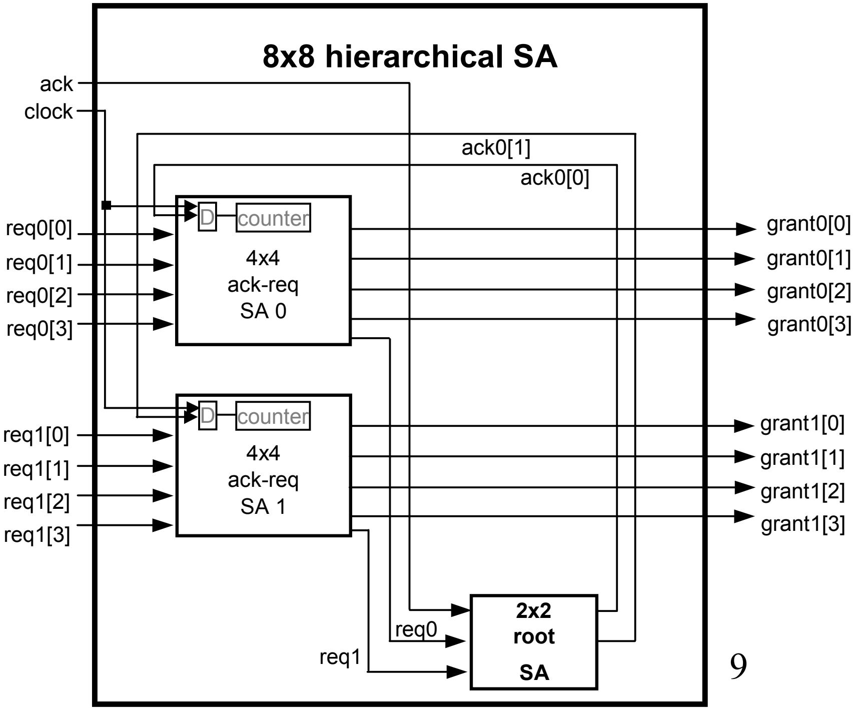
- **(MxV)xN crossbar switch fabric**:
  - There are connections between (MxV) inputs (from VOQ (0, 0) to VOQ (M-1, V-1)) and N outputs, the number of output ports in the switch fabric.
- **MxM Switch Arbiter (SA)**:
  - An MxM SA controls M specific transmission gates between M VOQs and a particular output port.
  - There are N MxM SAs in an MxN switch.



Terminology (Continued)

- **MxM distributed SA (MxM hierarchical SA)**: plays the same role as an MxM SA.
  - Consists of smaller switch arbiter in the form of a hierarchical tree structure.
- **Bus Arbiter (BA)**: resolves bus conflicts when multiple bus masters request a bus in the same cycle.





Related Work

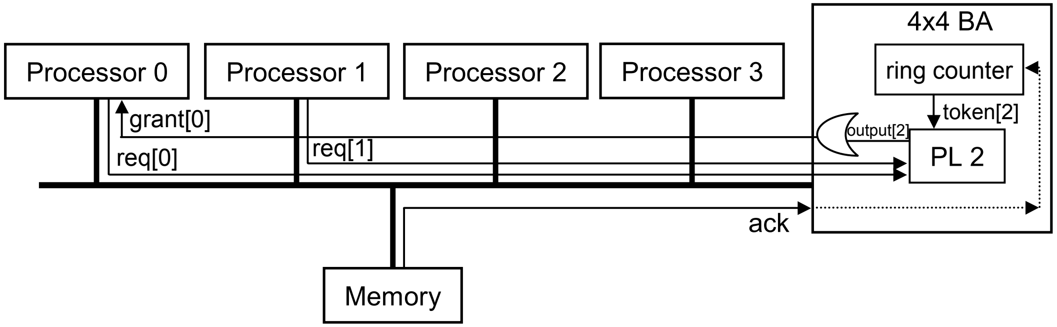
- Centralized Switch Arbiters:
  - Dual Round-Robin Matching algorithm (DRRM)
    - H. J. Chao and J. S. Park, “Centralized Contention Resolution Schemes for a Larger-capacity Optical ATM Switch,” *Proceedings of IEEE ATM Workshop*, 1998, pp. 11-16.
  - Programmable Priority Encoder (PPE) implementing iterative round-robin algorithm (iSLIP)
    - P. Gupta and N. Mckeown, “Designing and Implementing a Fast Crossbar Scheduler,” *IEEE Micro*, 1999, pp. 20-28.
    - N. Mckeown, P. Varaiya, and J. Warland, “The iSLIP Scheduling Algorithm for Input-Queued Switch,” *IEEE Transaction on Networks*, 1999, pp. 188-201.
- Distributed Switch Arbiter:
  - Ping Pong Arbiter (PPA)
    - H. J. Chao, C. H. Lam, and X. Guo, “A Fast Arbitration Scheme for Terabit Packet Switches,” *Proceedings of IEEE Global Telecommunications Conference*, 1999, pp. 1236-1243.
- We will show how our generated SA achieves throughput 2.4X higher than PPE and 1.9X higher than PPA (and thus, at least 1.9X higher than DRRM since PPA outperforms DRRM).

Bus Arbiter Design

- Implemented based on ring counter for a token and “priority logic”.
- Priority Logic for 4 inputs:
  - output[0] = EN•in[0]
  - output[1] = EN•in[0]'•in[1]
  - output[2] = EN•in[0]'•in[1]'•in[2]
  - output[3] = EN•in[0]'•in[1]'•in[2]'•in[3]

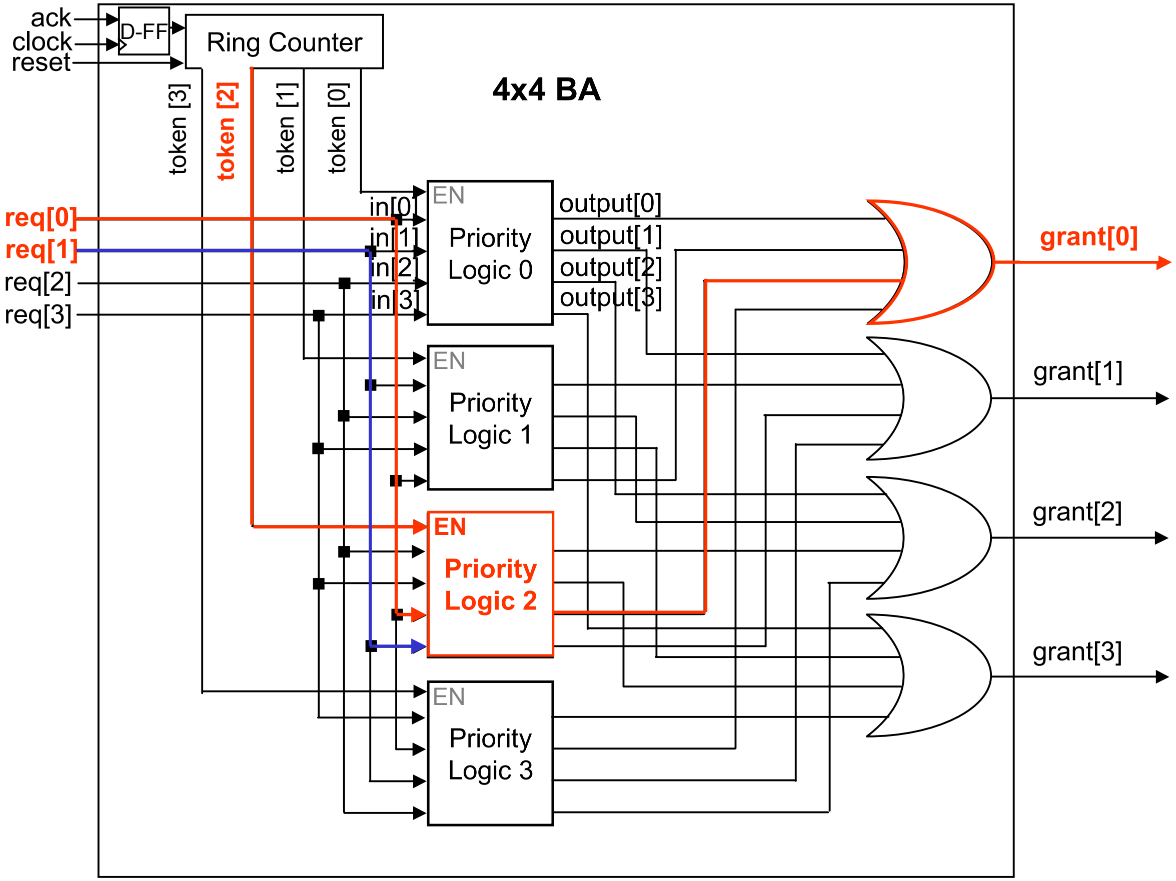
EN	in [0]	in [1]	in [2]	in [3]	output [0]	output [1]	output [2]	output [3]
0	X	X	X	X	0	0	0	0
1	0	0	0	0	0	0	0	0
1	1	X	X	X	1	0	0	0
1	0	1	X	X	0	1	0	0
1	0	0	1	X	0	0	1	0
1	0	0	0	1	0	0	0	1

Example: Bus Arbiter



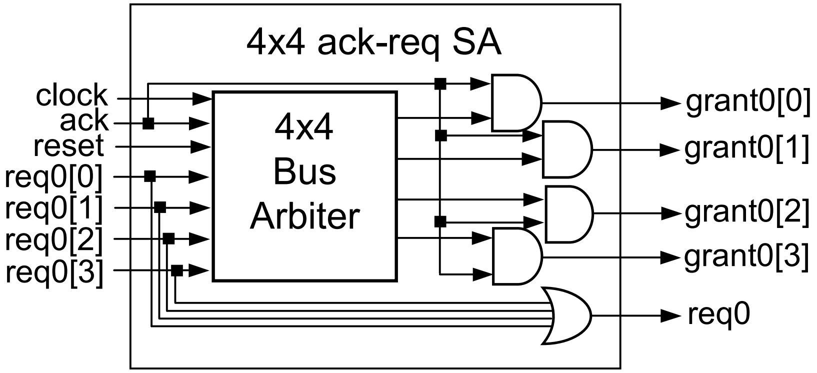
- Condition:
  - Token=4'b0100 → Processor 2 has the highest priority.
  - Processor 0 and processor 1 request a bus.
- Result:
  - Only Priority Logic 2 is enabled.
  - Processor 0 is granted because the higher priority parties (processor 2 and processor 3) do not request a bus.
  - Token is rotated to 4'b1000 after the ring counter receives ack signal.

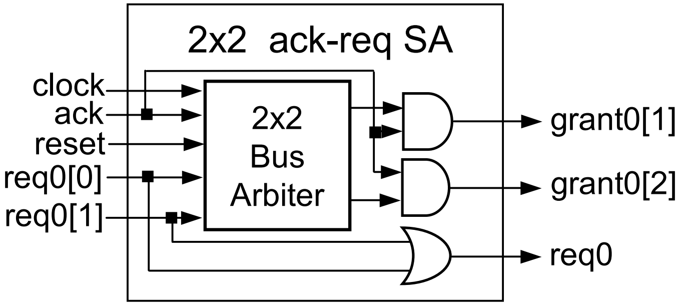
Example: Bus Arbiter (Continued)

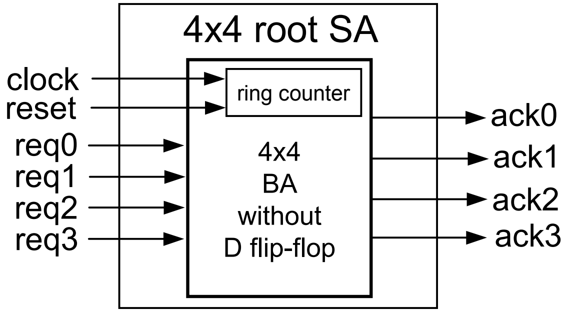


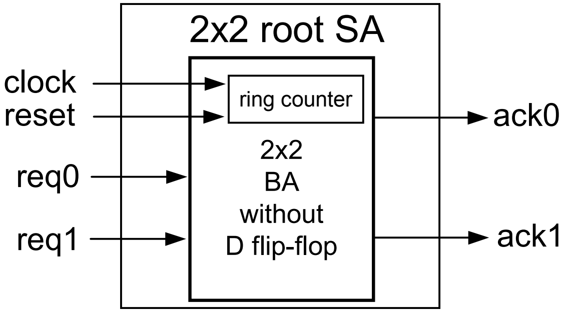
Switch Arbiter Design

- A hierarchical SA consists of small switch arbiter blocks.
- There are four types of switch arbiter blocks.
  - 2x2 ack-req SA.
  - 4x4 ack-req SA.
  - 2x2 root SA.
  - 4x4 root SA.
- A root SA placed on the top of a hierarchy.

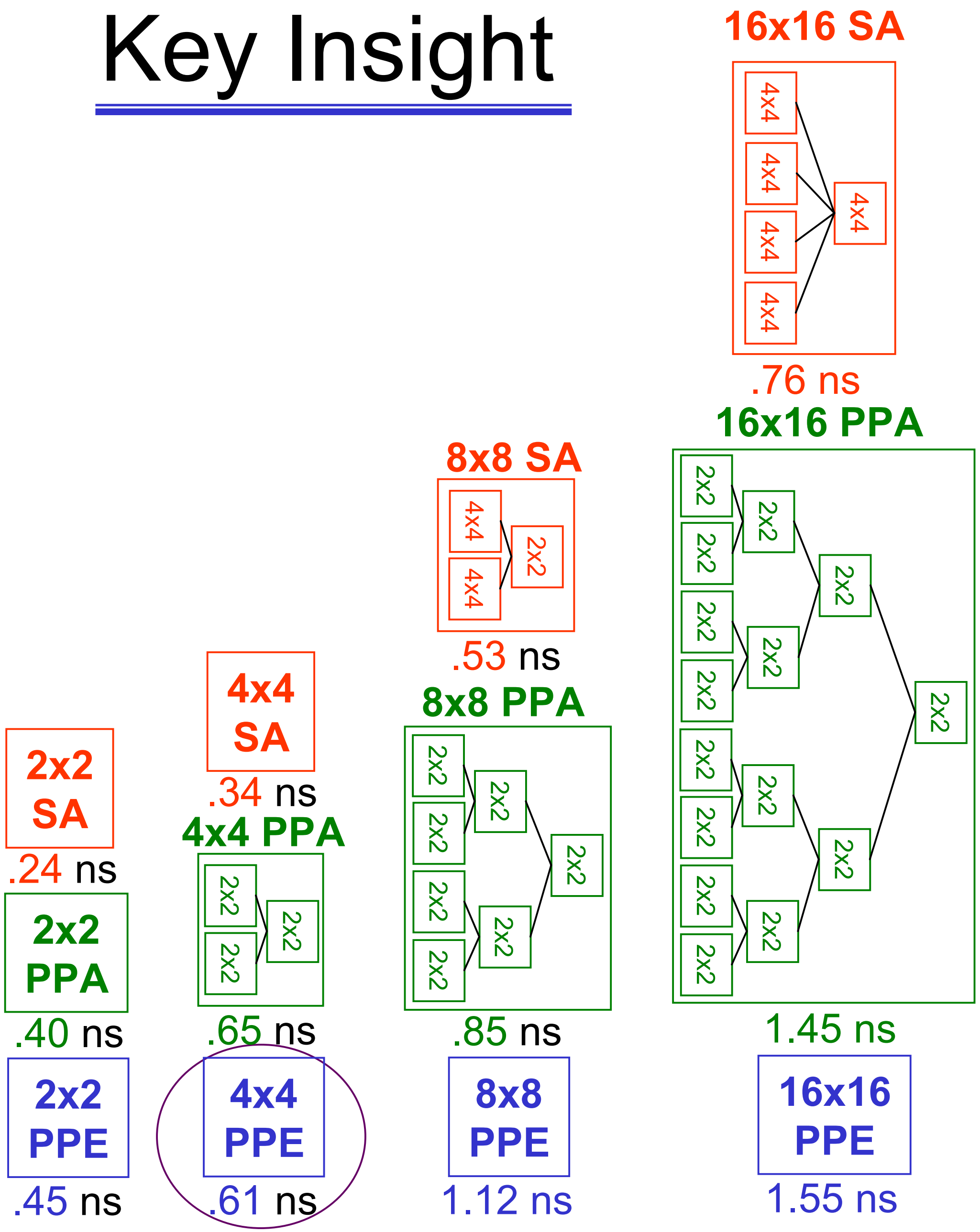








Key Insight

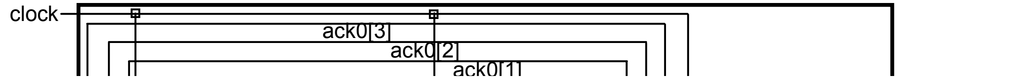


- With TSMC .25µ std. cell library from LEDA Systems, 4x4 is the “sweet spot” of high performance → analogous to std. cell design where using 4-input gates in design speeds up over, say only 2-input gates or 8-input gates.
  - Use as many 4x4 as possible.
  - Use 2x2 if needed.

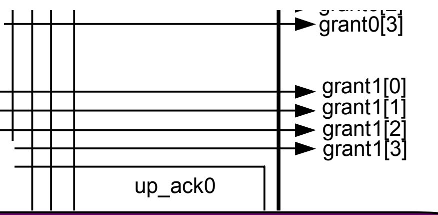
🟥 Our SA from RAG

🟩 PPA

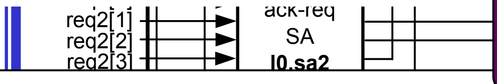
🟦 PPE

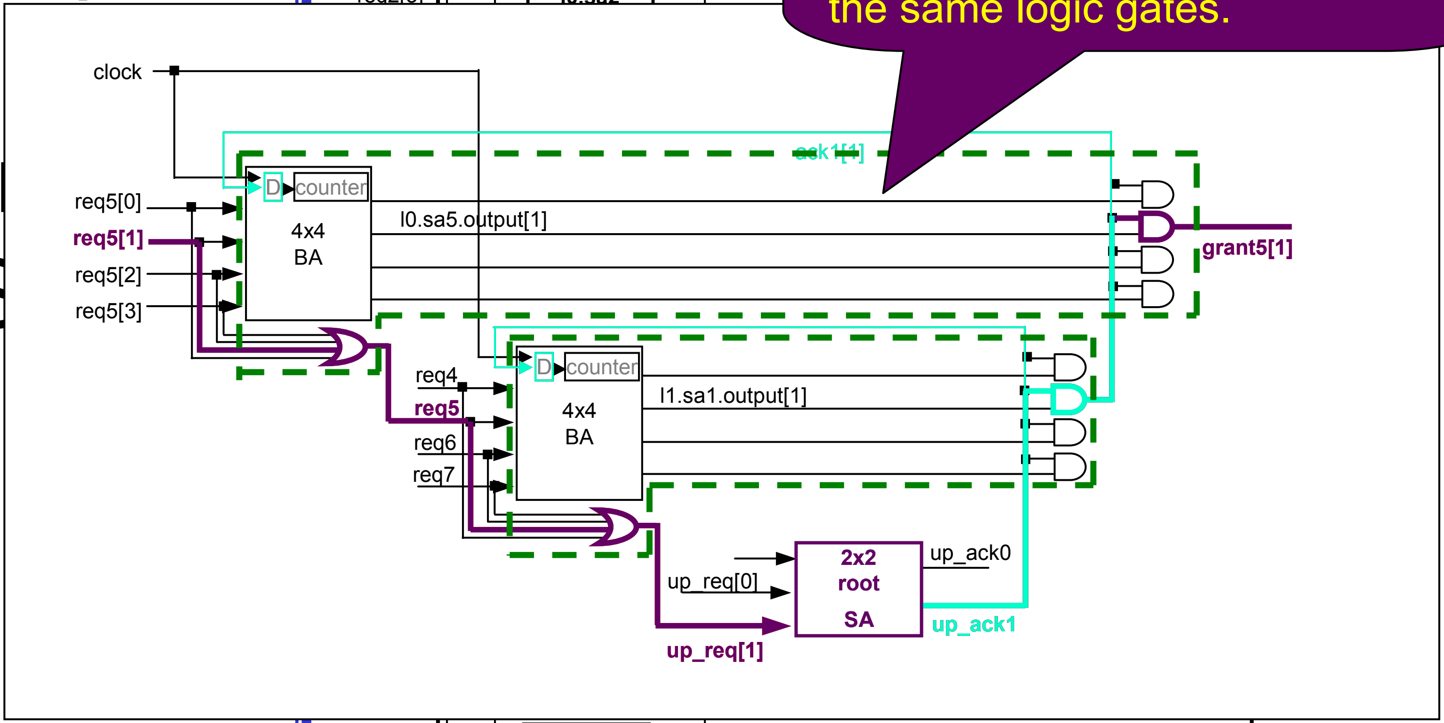


- 32 x 32 SA Critical Path:
  - •Travels through two 4-input OR gates in series
  - •Then through a 2x2 root SA
  - •Finally through two 2-input AND gates in series
  - •Results in 0.94ns delay using a TSMC
    from LEDA Systems.

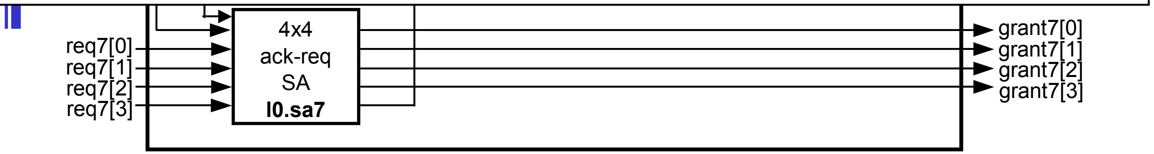


Example:
32
hierar
S



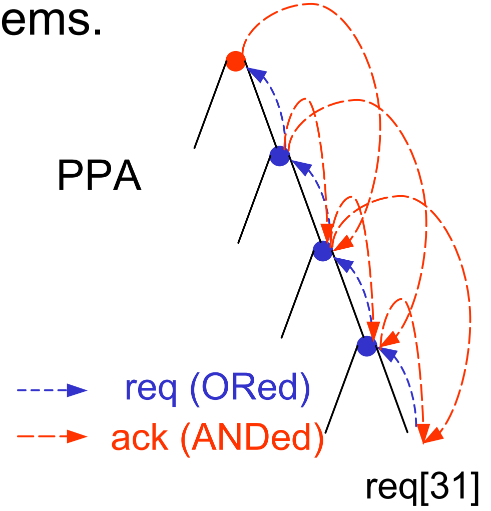


ack signals look like feedback path through the same logic block. In fact there is no input to the same logic gates.



Comparison w/32x32 PPE and PPA

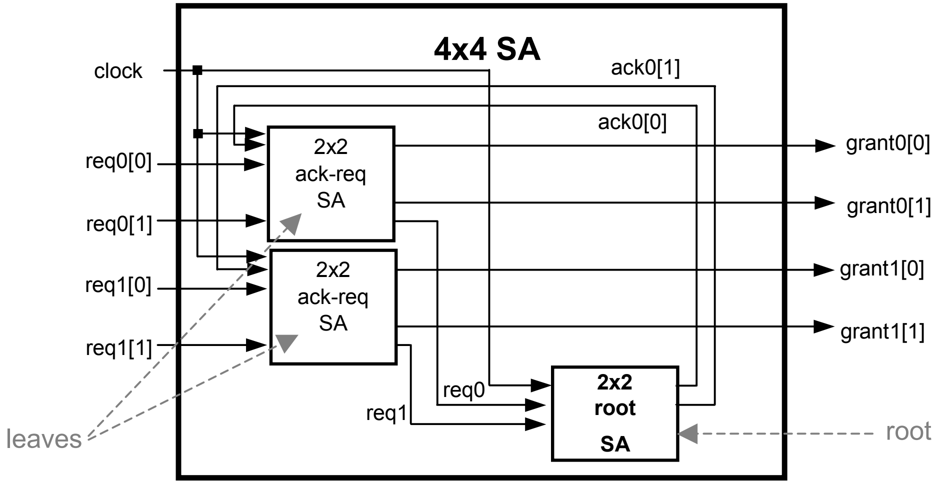
- PPE Critical Path:
  - output[31] = in[0]’•in[1]’•…•in[30]’•in[31] plus output encoding.
  - Associates with 8 4-input AND gates and 31 inverters.
  - Results in 2.17ns delay using a TSMC 0.25μ std. cell library from LEDA Systems.



- PPA Critical Path:
  - Only use 2x2 arbiters.
  - 2x2 PPA: 0.4ns while our 2x2 SA:0.24ns
  - 5 levels in a binary tree structure.
  - Associates with 4 serially connected 2-input OR gates for ORed request.
  - Associates with 2 acknowledgements from two higher levels →3 3-input AND gates.
  - Results in 1.7ns delay using a TSMC 0.25μ std. cell library from LEDA Systems.

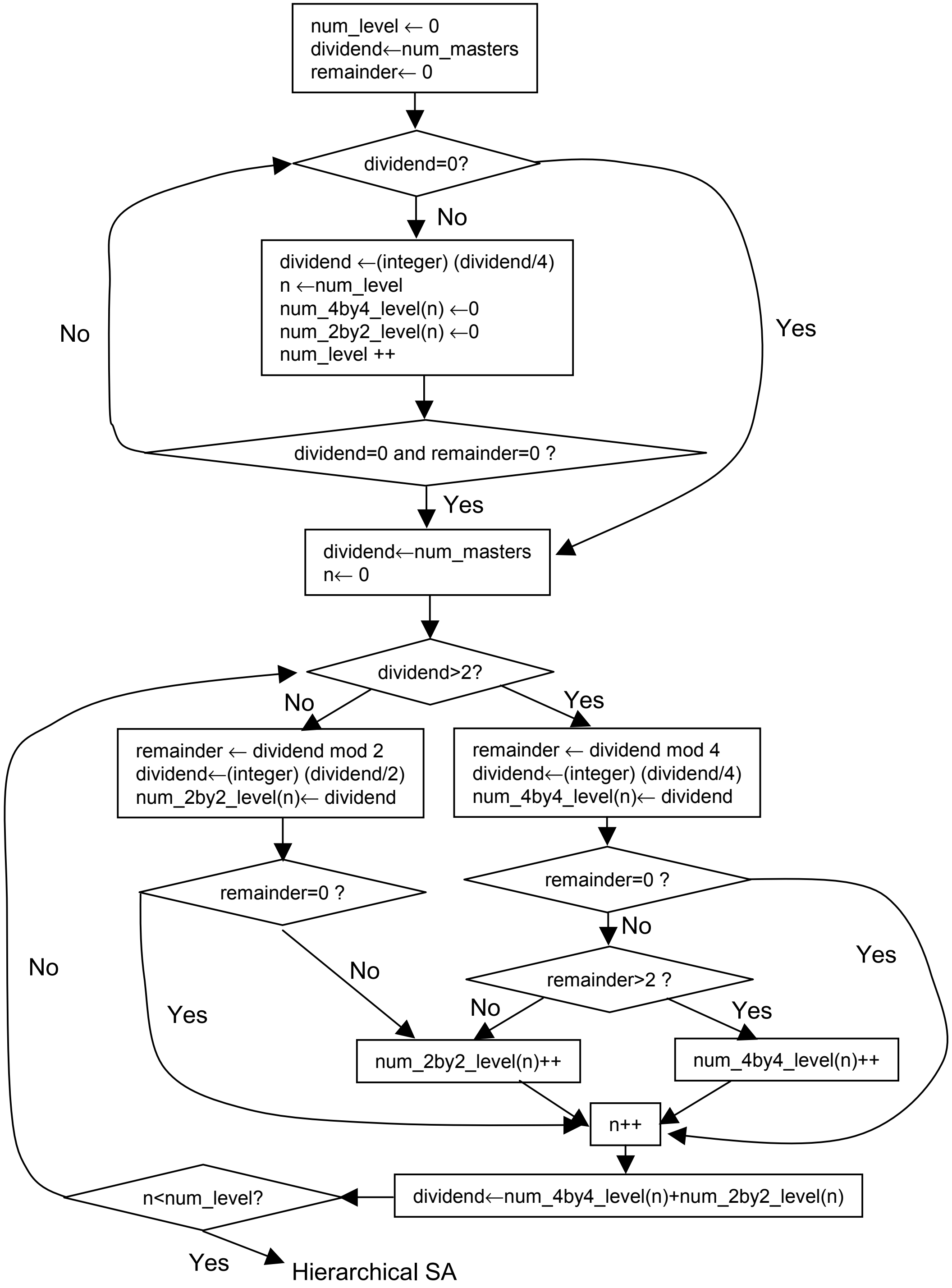
Round-robin Arbiter Generator (RAG)

- RAG is preferable to employ as many 4x4 SAs as possible to reduce the number of levels in a hierarchy.
- A hierarchical 4x4 SA has longer delay (0.46ns) than a 4x4 ack-req SA (0.34ns) in .25μ std. cell library from LEDA Systems.



RAG (Continued)

- A user specify an arbiter type either a Bus Arbiter or a Switch Arbiter.
- A user specify the number of masters (M) to be arbitrated.
- RAG generates synthesizable Verilog code for a Bus Arbiter or a Switch Arbiter at the RTL level.
- RAG is most efficient when M is a power of two.



num_level ← 0
dividend←num_masters
remainder← 0

dividend=0?

No

dividend ←(integer) (dividend/4)
n ←num_level
num_4by4_level(n) ←0
num_2by2_level(n) ←0
num_level ++

No

Yes

dividend=0 and remainder=0 ?

Yes

dividend←num_masters
n← 0

dividend>2?

No

Yes

remainder ← dividend mod 2
dividend←(integer) (dividend/2)
num_2by2_level(n)← dividend

remainder ← dividend mod 4
dividend←(integer) (dividend/4)
num_4by4_level(n)← dividend

remainder=0 ?

remainder=0 ?

No

No

Yes

No

remainder>2 ?

No

Yes

Yes

num_2by2_level(n)++

num_4by4_level(n)++

n++

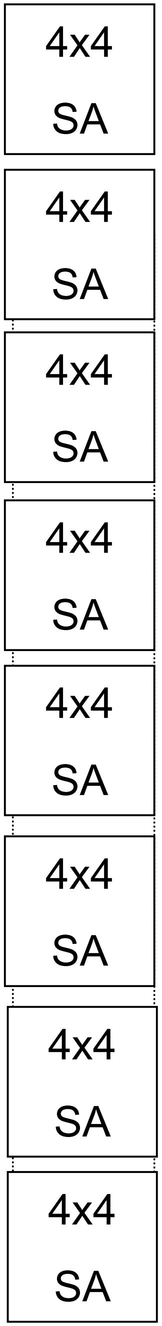
dividend←num_4by4_level(n)+num_2by2_level(n)

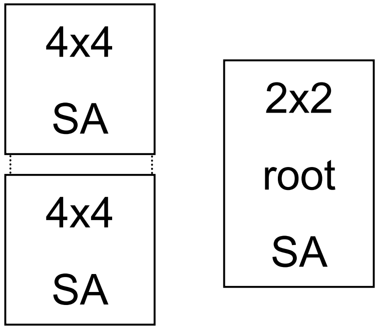
n<num_level?

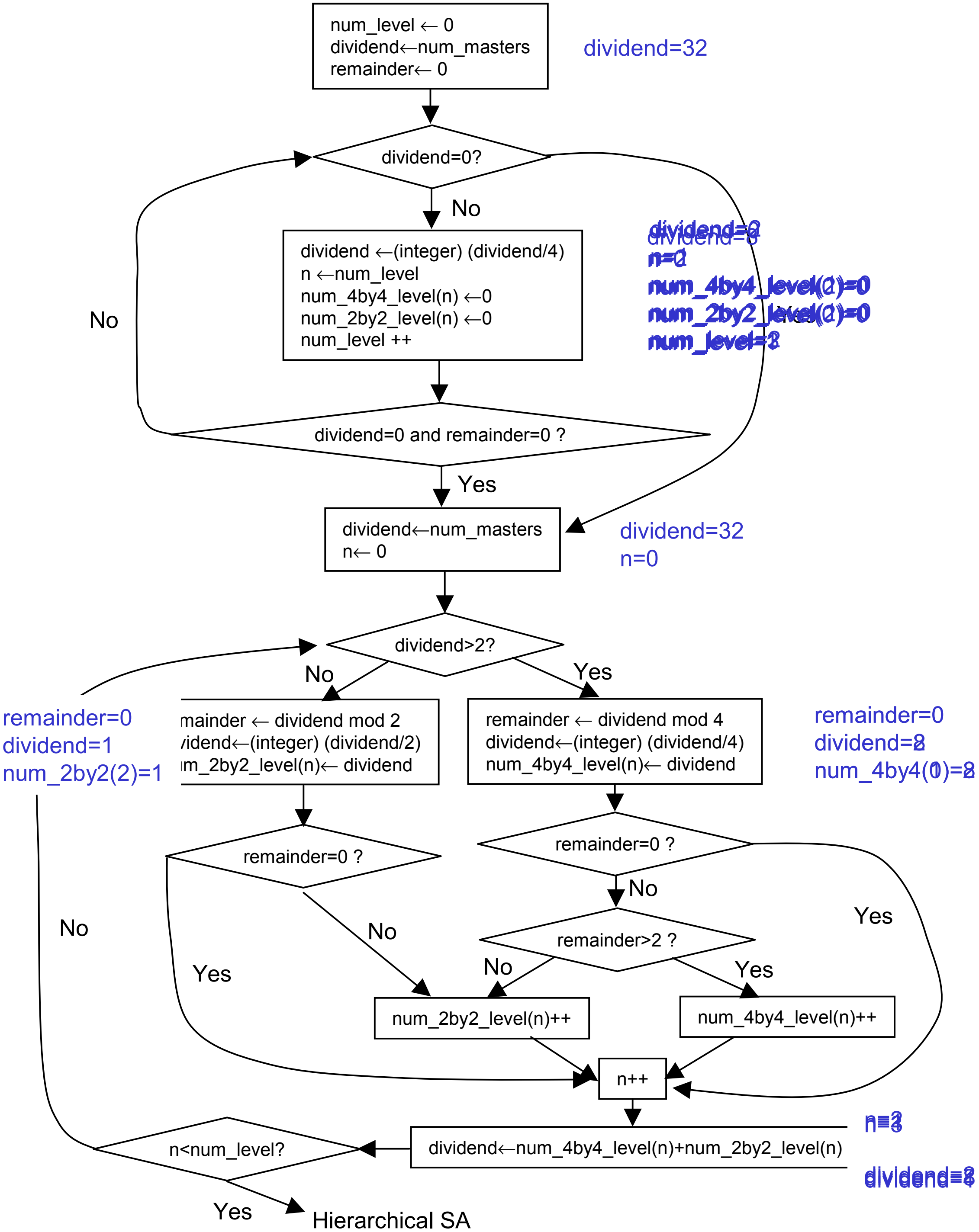
Yes

Hierarchical SA

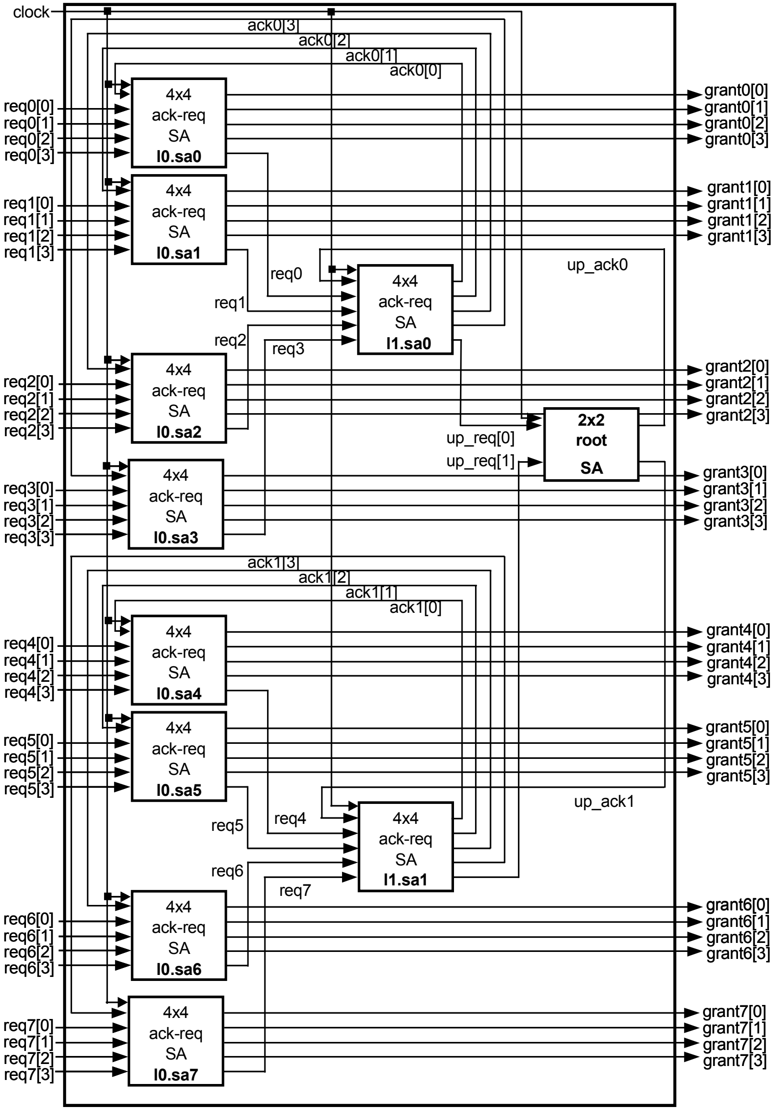
RAG (Conti

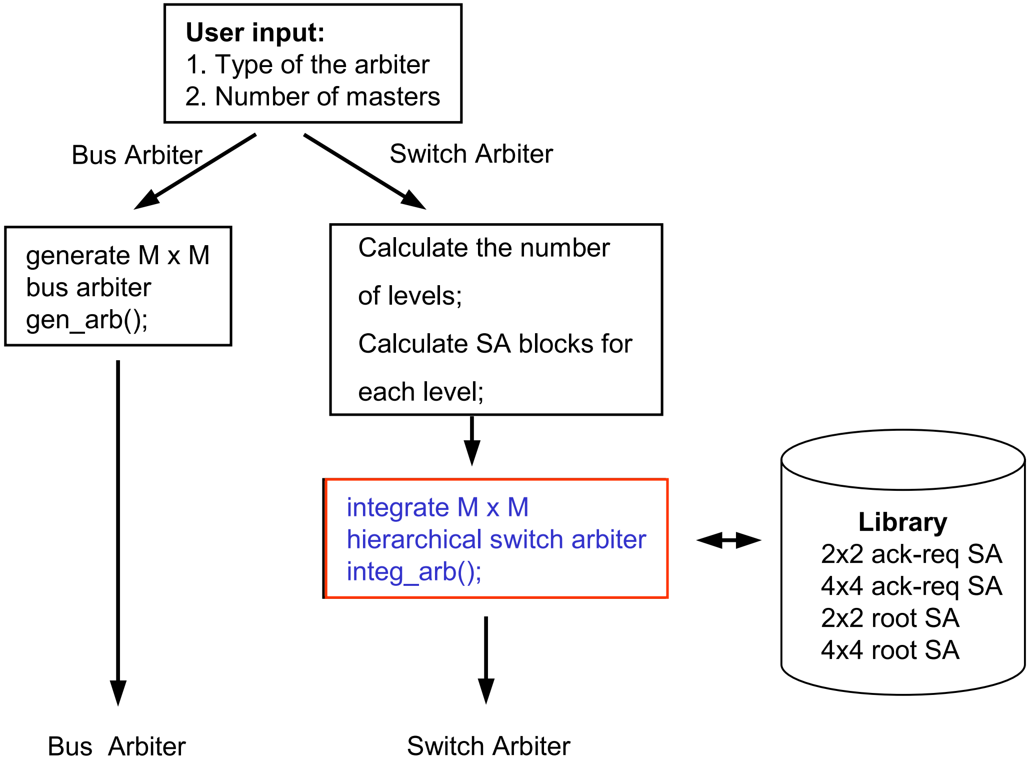

4x4 SA
4x4 SA
4x4 SA
4x4 SA
4x4 SA
4x4 SA
4x4 SA
4x4 SA


4x4 SA
4x4 SA
2x2 root SA


num_level ← 0
dividend←num_masters
remainder← 0
dividend=32
dividend=0?
No
dividend ←(integer) (dividend/4)
n ←num_level
num_4by4_level(n) ←0
num_2by2_level(n) ←0
num_level ++
dividend=2
dividend=8
n=2
n=1
num_4by4_level(2)=0
num_4by4_level(1)=0
num_2by2_level(2)=0
num_2by2_level(1)=0
num_level=2
num_level=3
No
dividend=0 and remainder=0 ?
Yes
dividend←num_masters
n← 0
dividend=32
n=0
dividend>2?
No
Yes
remainder=0
dividend=1
num_2by2(2)=1
mainder ← dividend mod 2
vidend←(integer) (dividend/2)
um_2by2_level(n)← dividend
remainder ← dividend mod 4
dividend←(integer) (dividend/4)
num_4by4_level(n)← dividend
remainder=0
dividend=2
dividend=8
num_4by4(0)=8
num_4by4(1)=2
remainder=0 ?
remainder=0 ?
No
No
Yes
Yes
No
remainder>2 ?
No
Yes
num_2by2_level(n)++
num_4by4_level(n)++
n++
n=2
n=3
dividend←num_4by4_level(n)+num_2by2_level(n)
dividend=8
dividend=2
n<num_level?
Yes
Hierarchical SA

RAG (Continued)





User input:
1. Type of the arbiter
2. Number of masters

Bus Arbiter

Switch Arbiter

generate M x M bus arbiter gen_arb();

Calculate the number of levels;

Calculate SA blocks for each level;

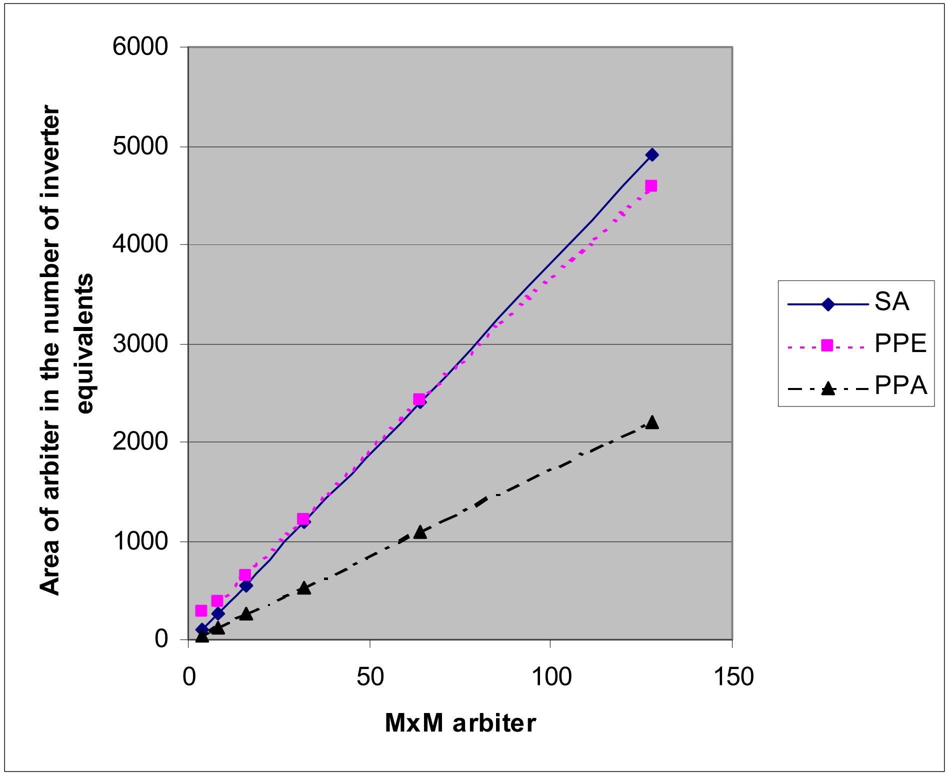
integrate M x M hierarchical switch arbiter integ_arb();

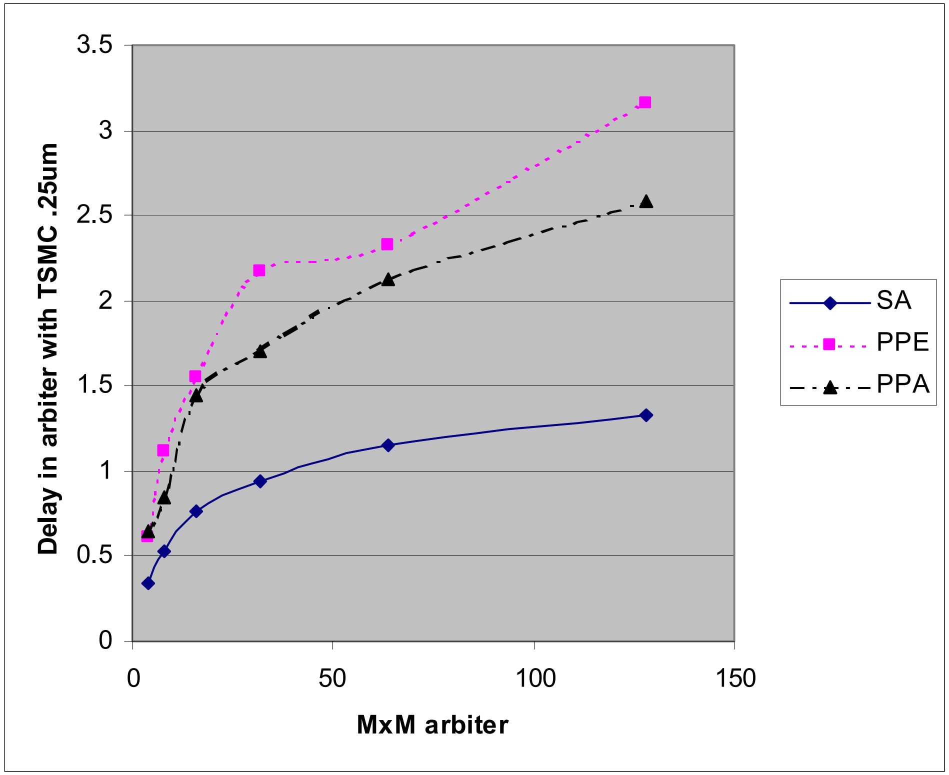
Library
2x2 ack-req SA
4x4 ack-req SA
2x2 root SA
4x4 root SA

Bus Arbiter

Switch Arbiter

Comparisons with PPE and PPA





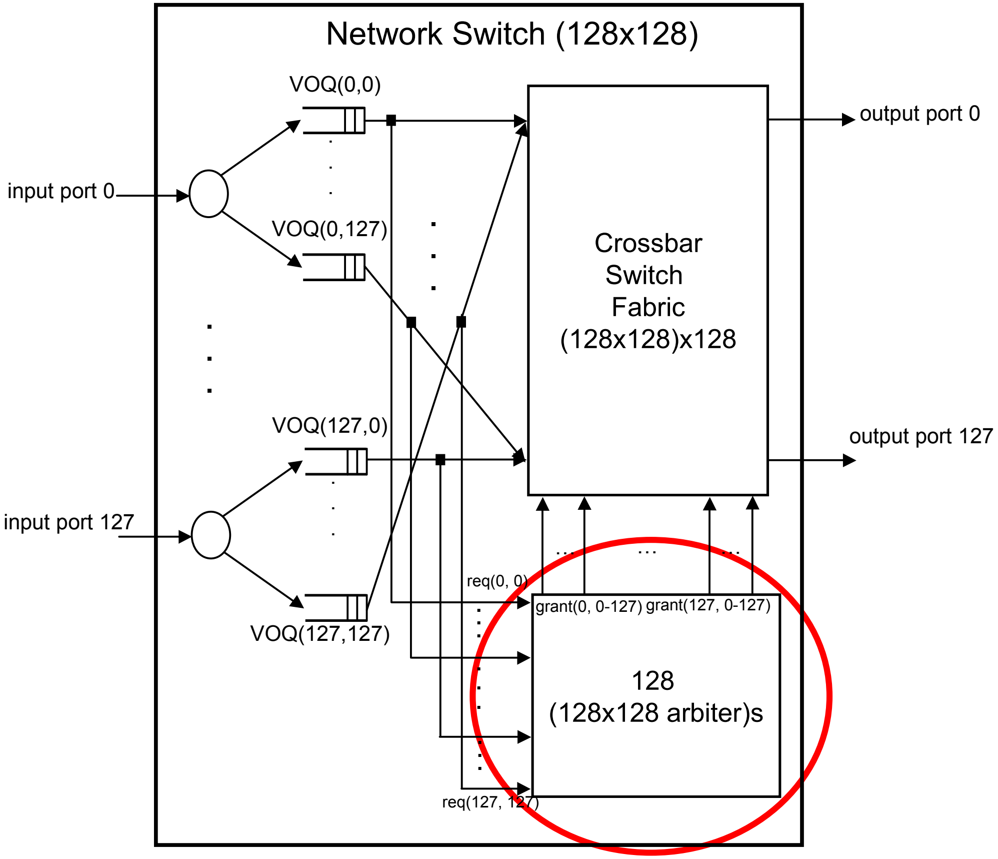
- Using TSMC 0.25µ std. cell library from LEDA Systems

Comparisons (Continued)

- The shortest delay results from
  - Limiting the size of switch arbiter blocks to 2x2 and 4x4 to reduce the critical path delay due to the expansion of priority logic blocks compared with Programmable Priority Logic Encoder (PPE), a centralized arbiter.
  - Reducing the number of levels in a hierarchy by preferring to use more 4x4 switch arbiter blocks compared with Ping-Pong Arbiter (PPA).

Speedup for a Terabit Switch

- Assumptions for comparison
  - The speed of switching is wholly determined by the arbitration cycles.
- Speedup
  - Our hierarchical 128x128 SA: 6.16Tbps.
  - 128x128 PPA: 3.18Tbps.
  - 128x128 PPE: 2.59Tbps.
  - Our SA achieves throughput 1.9X higher than PPA and 2.4X higher than PPE.
- Commercial Switches
  - Mindspeed claims up to .45Tbps for 144x144 switch using multiple chips.
  - PetaSwitch claims up to 10.24Tbps for 256x256 switch using multiple chips.
  - No details about logic design nor process technology used.



Conclusion

- BA logic
- We showed how 2x2 and 4x4 BAs are applied to 2x2 and 4x4 switch arbiter blocks.
- We demonstrated how RAG generate synthesizable Verilog codes for a BA and a SA with the example of 32x32 hierarchical SA.
- We compared areas and delays with other SAs.
- We demonstrated how our generated 128x128 hierarchical SA could achieve throughput 1.9X higher than PPA and 2.4X higher than PPE.