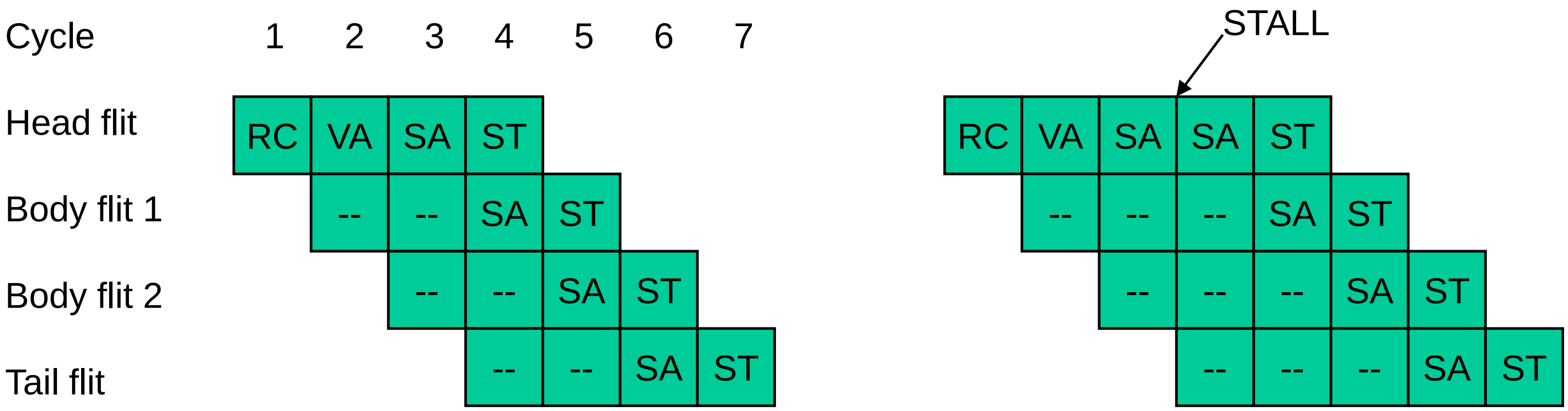
Lecture 21: Router Design

Papers:

- Power-Driven Design of Router Microarchitectures in On-Chip Networks, MICRO'03, Princeton
- A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks, ISCA'06, Penn-State
- ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers, MICRO'06, Penn-State

Router Pipeline

- Four typical stages:
  - RC routing computation: compute the output channel
  - VA virtual-channel allocation: allocate VC for the head flit
  - SA switch allocation: compete for output physical channel
  - ST switch traversal: transfer data on output physical channel



Cycle	1	2	3	4	5	6	7
Head flit	RC	VA	SA	ST			
Body flit 1		--	--	SA	ST		
Body flit 2			--	--	SA	ST	
Tail flit				--	--	SA	ST

RC	VA	SA	SA	ST			
	--	--	--	SA	ST		
		--	--	--	SA	ST	
			--	--	--	SA	ST

Flow Control

- VC allocation: when the tail flit is sent, the router knows that the downstream VC is free (or will soon be); the VC is therefore assigned to the next packet and those flits carry the VCid with them; the two routers need not exchange signals to agree on the VCid

- Head-of-Line (HoL) blocking: a flit at the head of the queue blocks flits (belonging to a different packet) behind it that could have progressed… example: if a VC holds multiple packets because the upstream node assumed the previous packet was handled (as above)

- Flow control mechanisms:
  - Store-and-Forward: buffers/channels allocated per packet
  - Cut-through: buffers/channels allocated per packet
  - Wormhole: buffers allocated per flit; channels per packet
  - Virtual channel: buffers/channels allocated per flit

Data Points

- On-chip network’s power contribution
  in RAW (tiled) processor: 36%
  in network of compute-bound elements (Intel): 20%
  in network of storage elements (Intel): 36%
  bus-based coherence (Kumar et al. ’05): ~12%

- Contributors:
  RAW: links 39%; buffers 31%; crossbar 30%
  TRIPS: links 31%; buffers 35%; crossbar 33%
  Intel: links 18%; buffers 38%; crossbar 29%; clock 13%

Network Power

- Energy for a flit = $E_R . H + E_{wire} . D$

  $= (E_{buf} + E_{xbar} + E_{arb}) . H + E_{wire} . D$

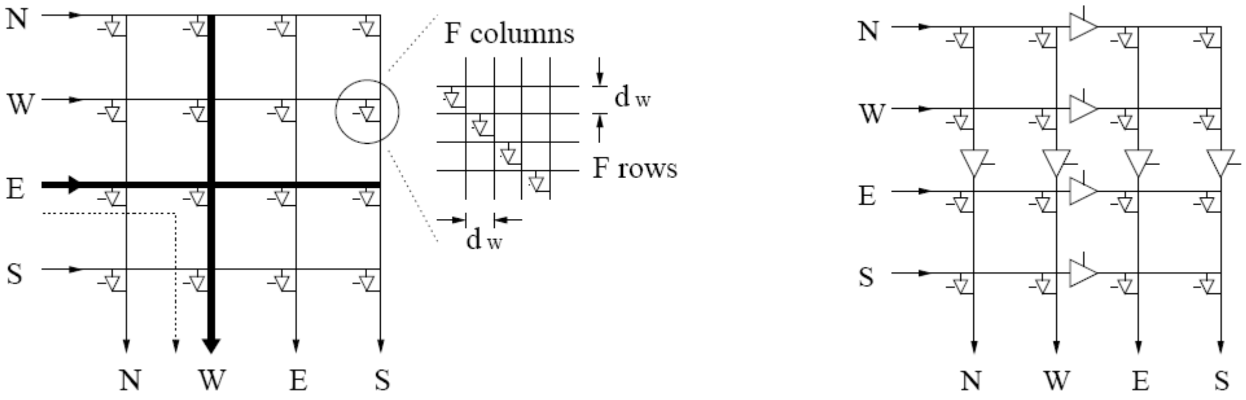
E_R = router energy
E_{wire} = wire transmission energy
E_{buf} = router buffer energy
E_{arb} = router arbiter energy

H = number of hops
D = physical Manhattan distance
E_{xbar} = router crossbar energy

- This paper assumes that $E_{wire} . D$ is ideal network energy (assuming no change to the application and how it is mapped to physical nodes)

- Optimizations are attempted to E_R and H

Segmented Crossbar



(a) A 4×4 matrix crossbar.

(b) A 4×4 segmented crossbar with 2 segments per line.

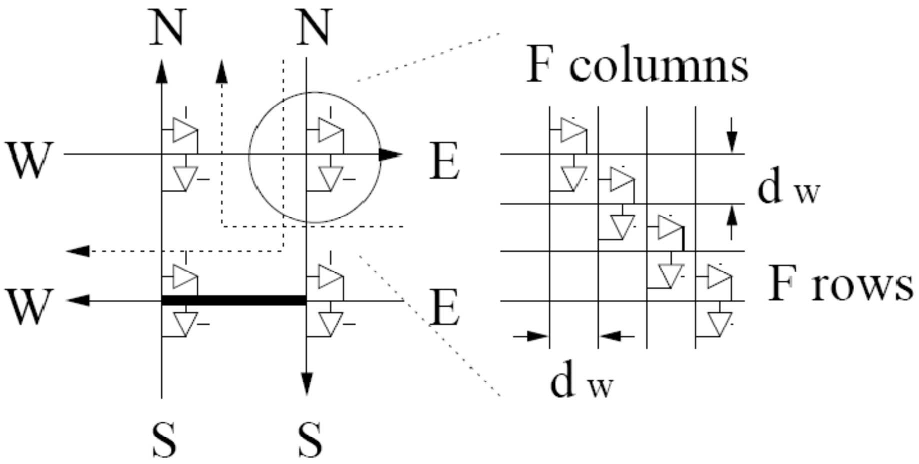
- By segmenting the row and column lines, parts of these lines need not switch → less switching capacitance (especially if your output and input ports are close to the bottom-left in the figure above)
- Need a few additional control signals to activate the tri-state buffers (~2 control signals, ~64 data signals)
- Overall crossbar power savings: ~15-30%

Cut-Through Crossbar

- Attempts to optimize the common case: in dimension-order routing, flits make up to one turn and usually travel straight

- 2/3rd the number of tristate buffers and 1/2 the number of data wires

- "Straight" traffic does not go thru tristate buffers



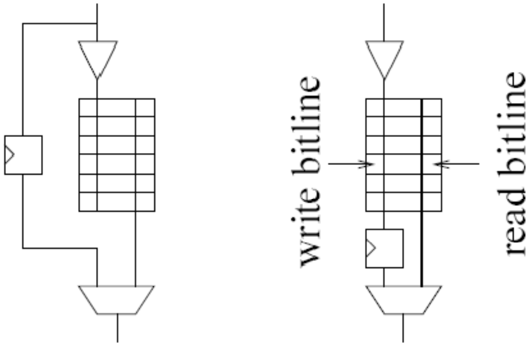
(a) A 4×4 cut-through crossbar.

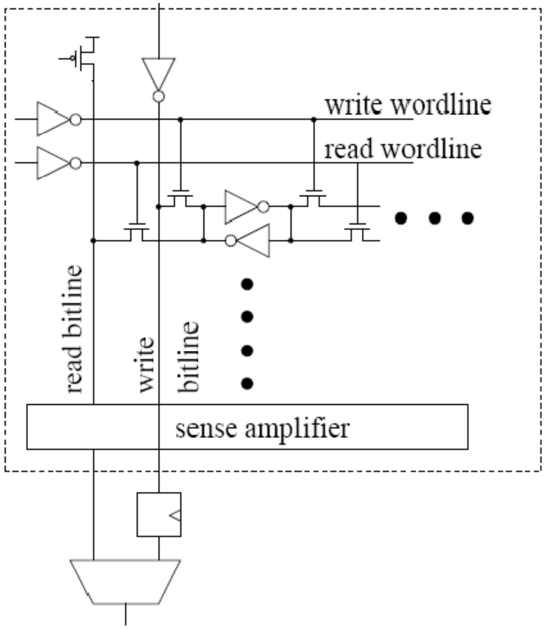
- Some combinations of turns are not allowed: such as E → N and N → W (note that such a combination cannot happen with dimension-order routing)

- Crossbar energy savings of 39-52%; at full-load, with a worst-case routing algorithm, the probability of a conflict is ~50%

Write-Through Input Buffer

- Input flits must be buffered in case there is a conflict in a later pipeline stage
- If the queue is empty, the input flit can move straight to the next stage: helps avoid the buffer read
- To reduce the datapaths, the write bitlines can serve as the bypass path
- Power savings are a function of rd/wr energy ratios and probability of finding an empty queue



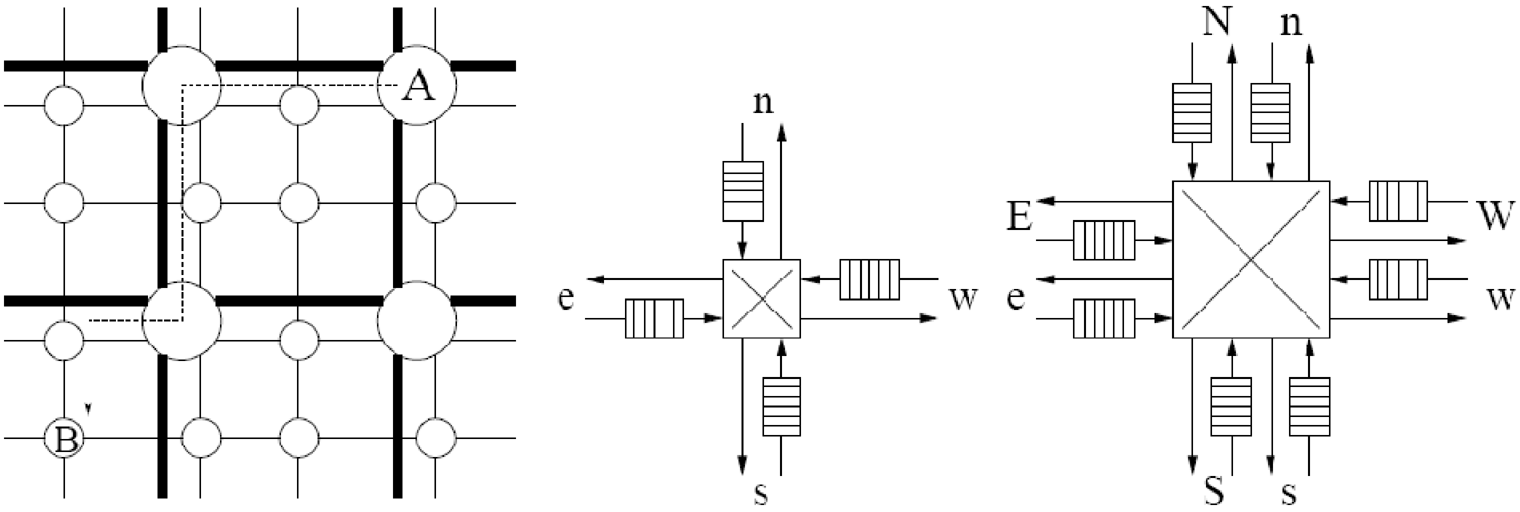


Express Channels

- Express channels connect non-adjacent nodes – flits traveling a long distance can use express channels for most of the way and navigate on local channels near the source/destination (like taking the freeway)

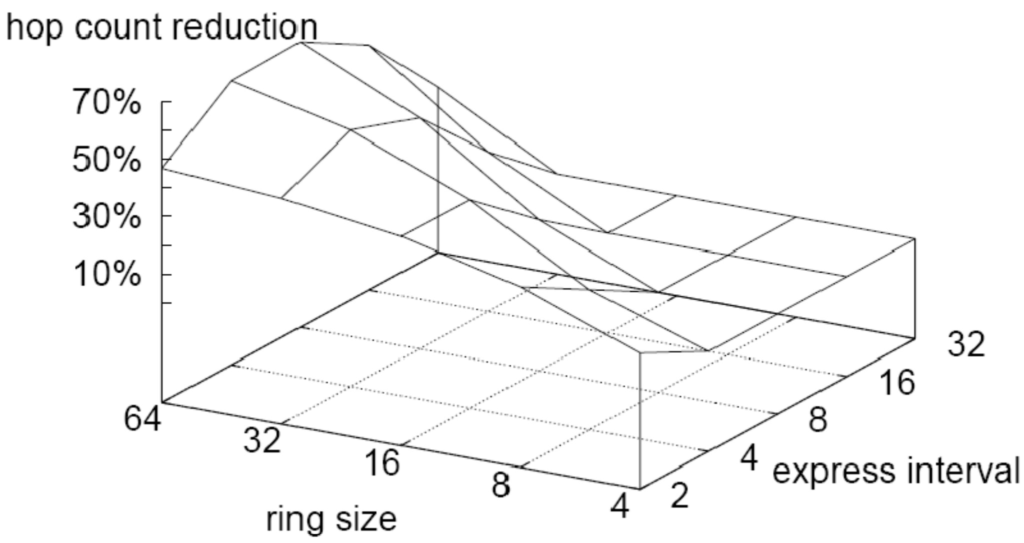
- Helps reduce the number of hops

- The router in each express node is much bigger now



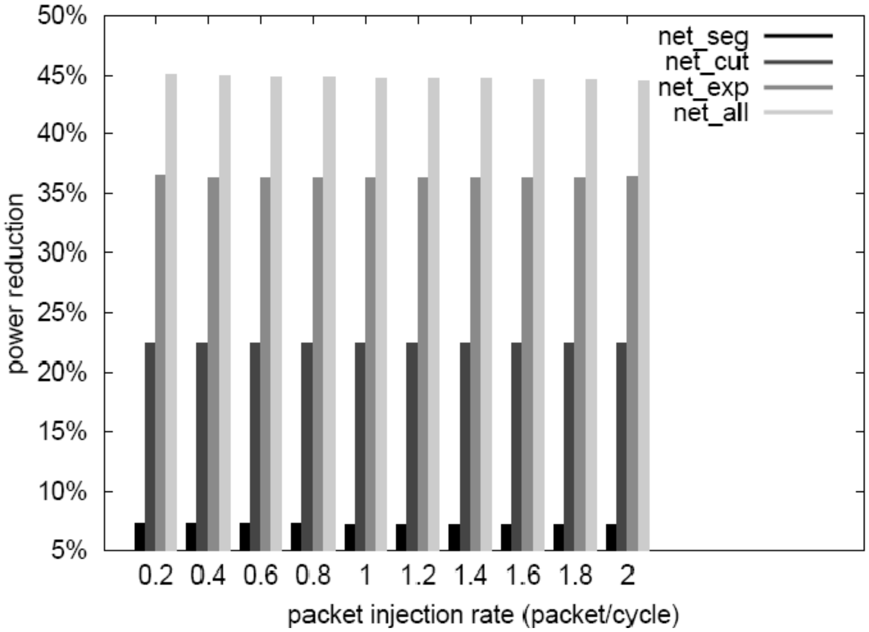
Express Channels

- Routing: in a ring, there are 5 possible routes and the best is chosen; in a torus, there are 17 possible routes
- A large express interval results in fewer savings because fewer messages exercise the express channels



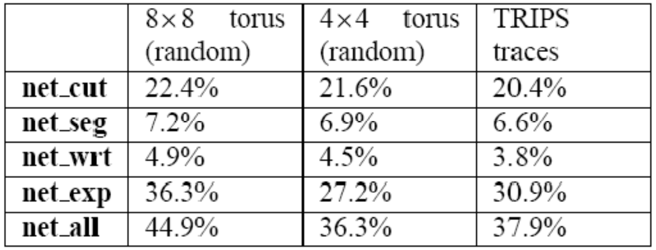
Results

- Uniform random traffic (synthetic)
- Write-thru savings are small
- Exp-channel network has half the flit size to maintain the same bisection-bandwidth as other models (express interval of 2)
- Baseline model power breakdown: link 44%, crossbar 33%, buffers 23%
- Express cubes also improve 0-load latency by 23% -- the others have a negligible impact on performance



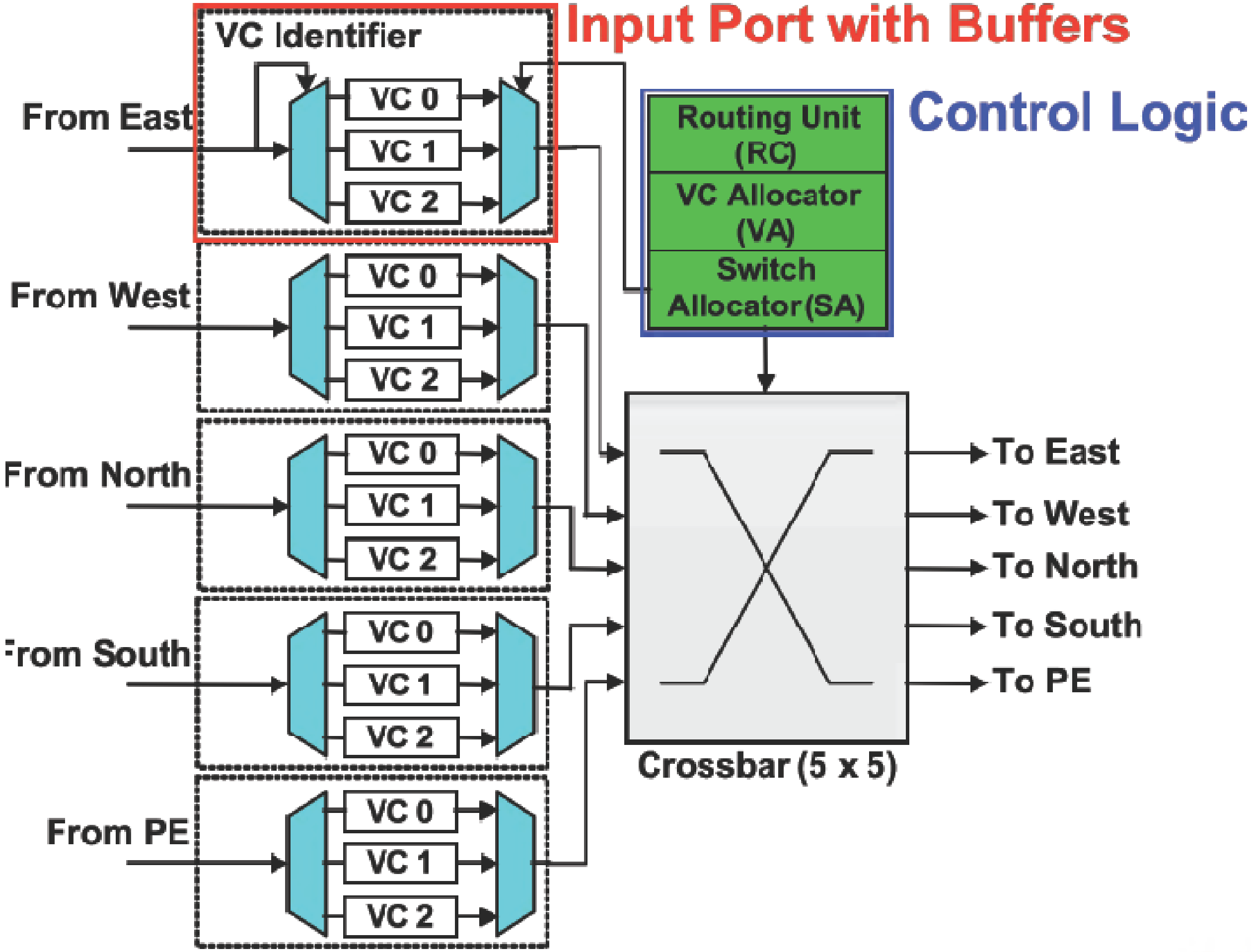
(c) 8×8 network power savings of 4 configurations.

Table 3. Average total network power savings (relative to net_base configuration).



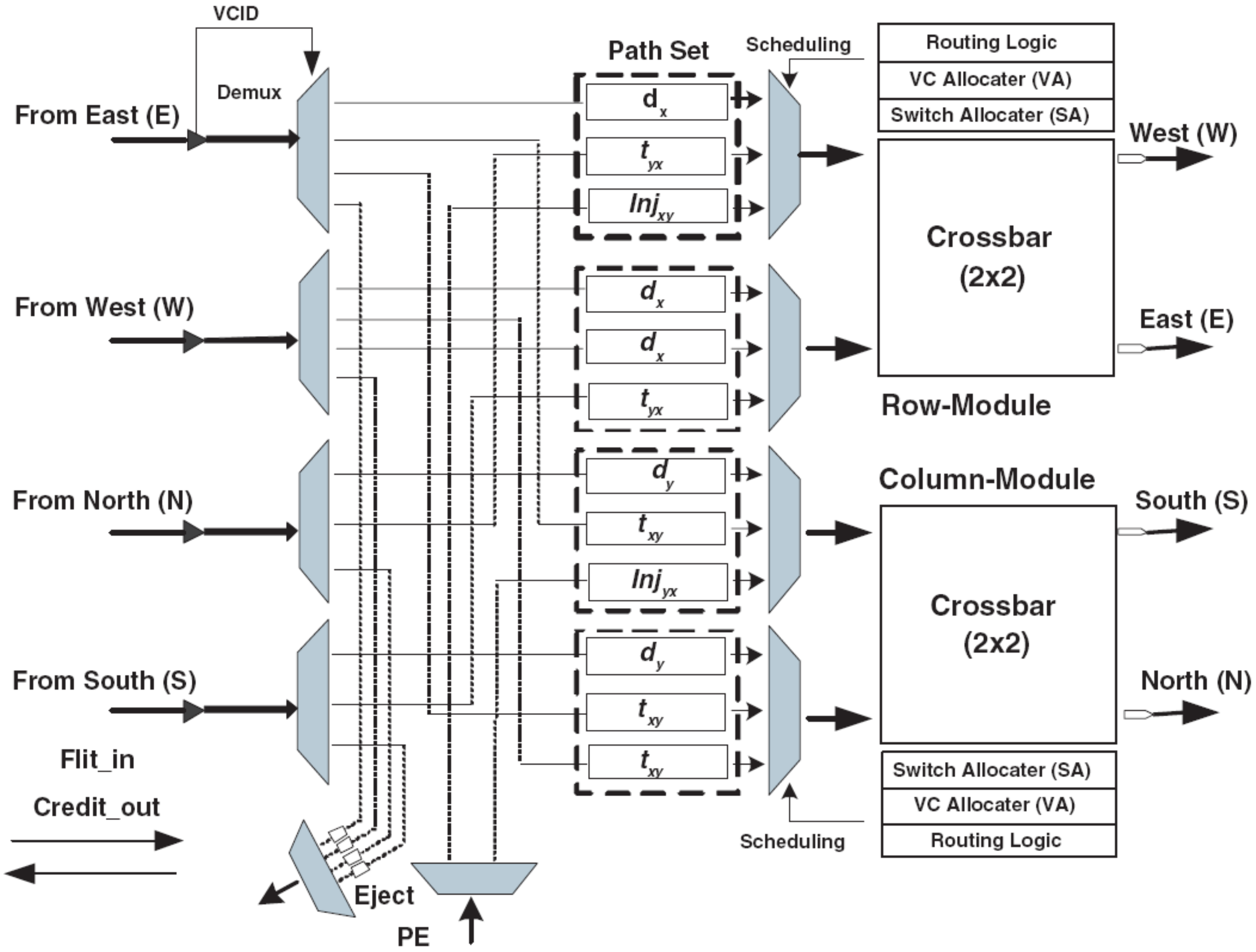
	8×8 torus (random)	4×4 torus (random)	TRIPS traces
net_cut	22.4%	21.6%	20.4%
net_seg	7.2%	6.9%	6.6%
net_wrt	4.9%	4.5%	3.8%
net_exp	36.3%	27.2%	30.9%
net_all	44.9%	36.3%	37.9%

Conventional Router



Slide taken from presentation at OCIN'06

The RoCo Router

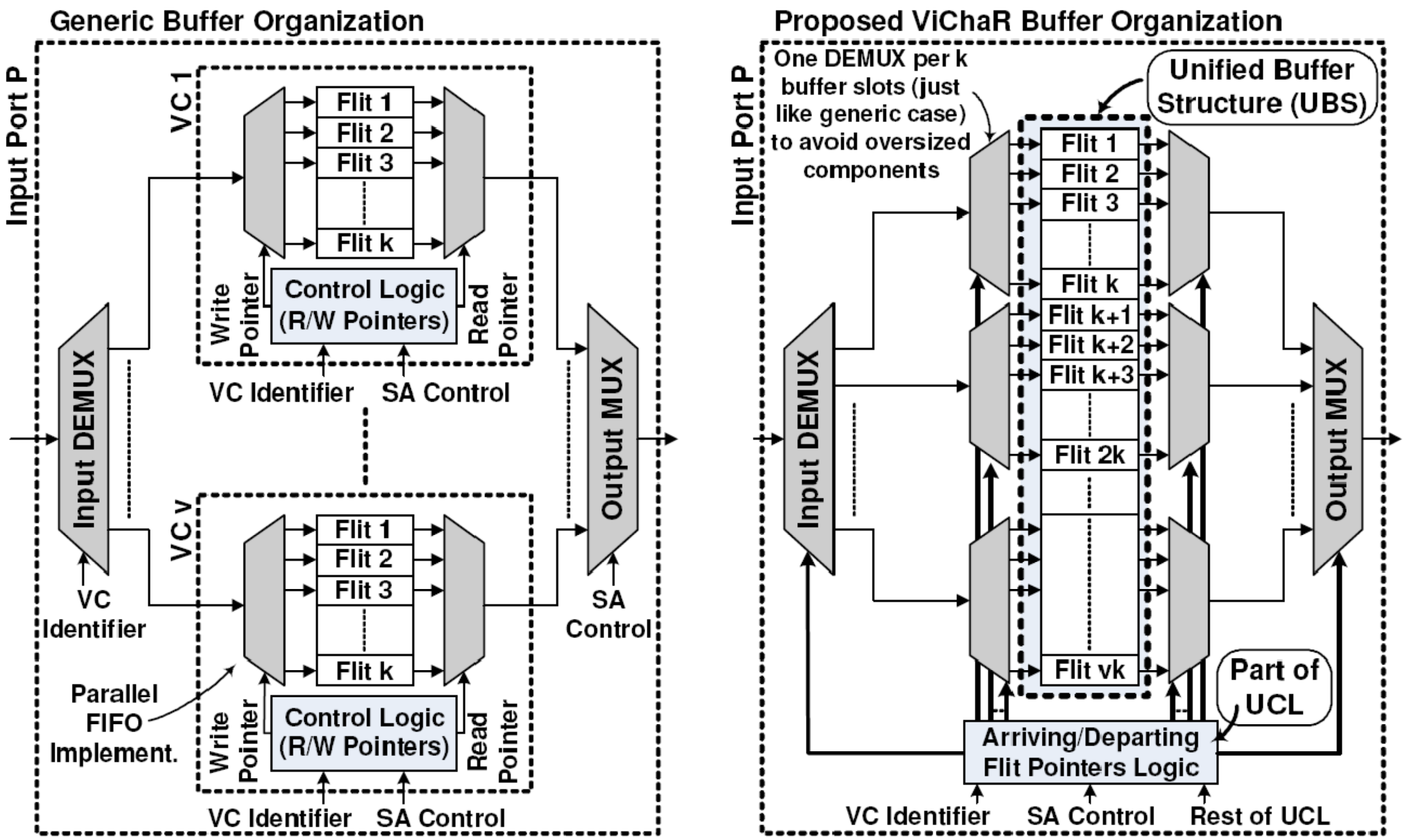


ViChaR

- Router buffers are a bottleneck:
  - consume 64% of router leakage power
  - consume up to 46% (54%) of total network power (area)
  - high buffer depth (buffers per VC) prevents a packet from holding resources at multiple routers
  - large number of VCs helps reduce contention under high load

- Primary contribution: instead of maintaining k buffers for each of the v virtual channels, maintain a unified storage of vk buffers and allow the number of VCs to dynamically vary between v and vk (buffer depth of k to 1)

Proposed Architecture



Generic Buffer Organization
Input Port P
VC 1
Flit 1
Flit 2
Flit 3
Flit k
Control Logic (R/W Pointers)
Write Pointer
Read Pointer
VC Identifier
SA Control
Input DEMUX
Output MUX
VC v
VC Identifier
SA Control
Parallel FIFO Implement.
Proposed ViChaR Buffer Organization
One DEMUX per k buffer slots (just like generic case) to avoid oversized components
Unified Buffer Structure (UBS)
Flit k+1
Flit k+2
Flit k+3
Flit 2k
Flit vk
Part of UCL
Arriving/Departing Flit Pointers Logic
Rest of UCL

Figure 4. The Proposed ViChaR Architecture

Unified Buffer Design

- A table to maintain the buffer entries for each VC

- Pointers to the head and tail of each VC

- A list of free buffer entries; a list of free VCs (some VCs are used as escape routes to avoid deadlock)

- The VCs are allocated in the upstream router – hence, when a VC is freed at a router, the upstream router is informed (this is not done in a conventional router) (process similar to credit flow to estimate buffer occupancy)

- Arbitration mechanism so packets can compete for the next channel

Results

Salient results:

- With 16 buffers per input port, ViChaR out-performs the generic router by ~25%, with a 2% power increase

- With 8 buffers, ViChaR matches the performance of a 16-buffer generic router, yielding area/power savings of 30%/34%

Title

- Bullet