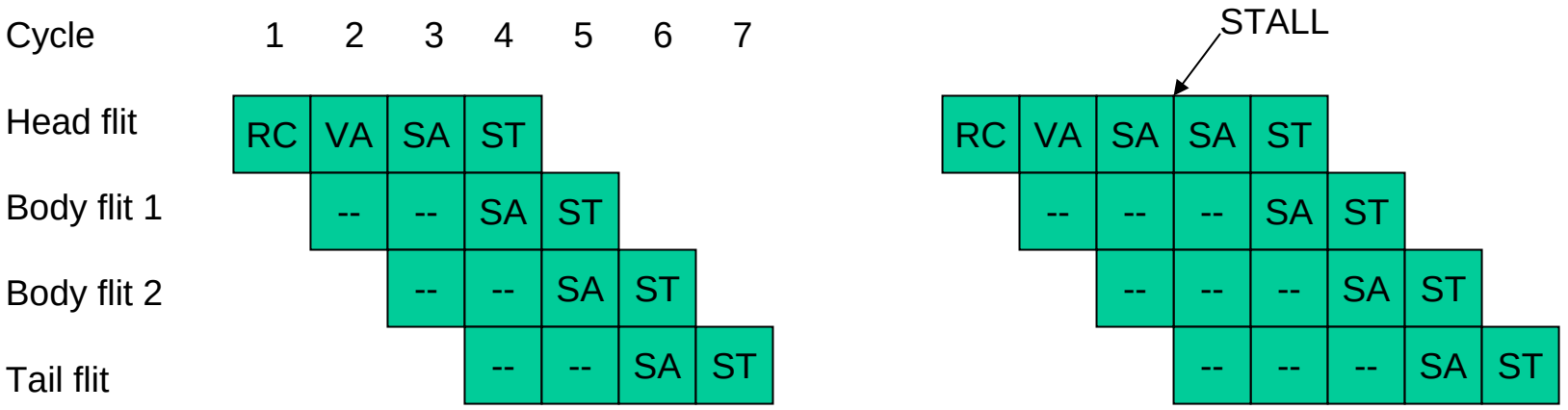
Lecture 23: Router Design

Papers:

- 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



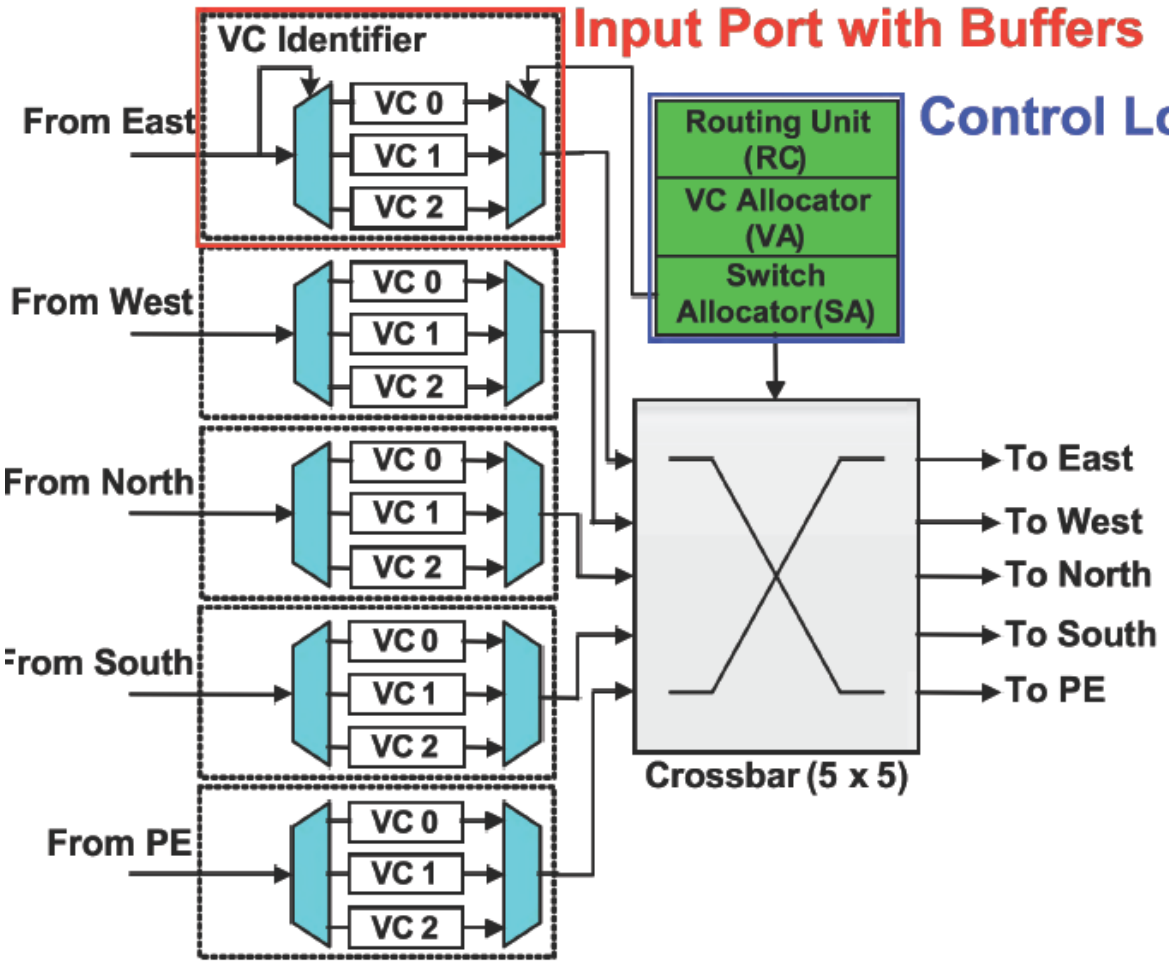
Flow Control

- 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)

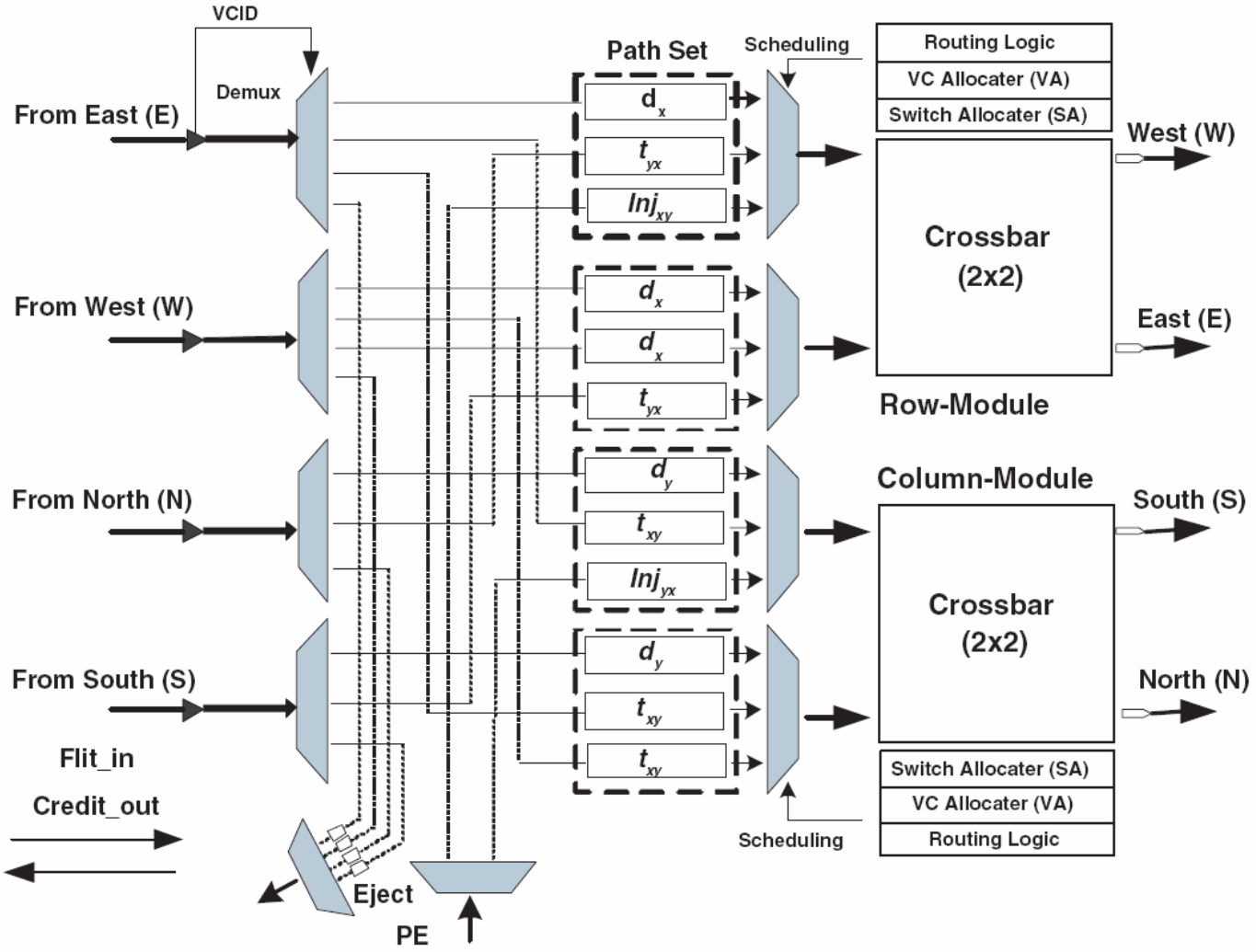
- Flow control mechanisms:
  - Store-and-Forward: buffers/channels allocated per packet
  - Cut-through: buffers/channels allocated per packet
  - Wormhole: buffers allocated per flit; channels per packet
  - Virtual channel: buffers/channels allocated per flit

Conventional Router



Slide taken from presentation at OCIN'06

The RoCo Router

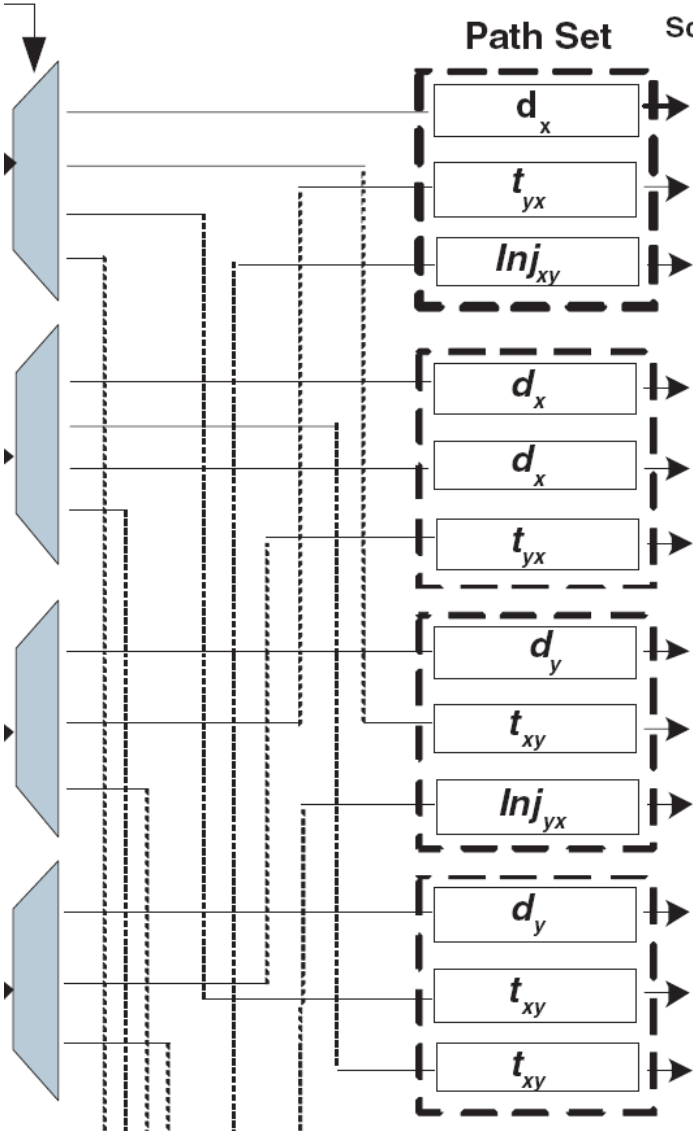


VC Allocation

- XY routing is deadlock-free; need a minimum of 8 VCs to allow every possible flit traversal: 2 d_x, 2 d_y, 2 t_{xy}, 1 Inj_{xy}, 1 Inj_{yx}
- XY-YX routing needs 2 more VCs to enable deadlock freedom
- Adaptive routing needs 12 VCs
- Additional constraints on VCs may lower performance

Input Port	Row-Module		Column-Module	
	Port 1	Port 2	Port 1	Port 2
Adaptive	$d_x\ t_{yx}\ Inj_{xy}$	$d_x\ d_x\ t_{yx}$	$d_y\ t_{xy}\ Inj_{yx}$	$d_y\ t_{xy}\ t_{xy}$
XY-YX	$d_x\ t_{yx}\ Inj_{xy}$	$d_x\ d_x\ t_{yx}$	$d_y\ t_{xy}\ Inj_{yx}$	$d_y\ d_y\ t_{xy}$
XY	$d_x\ d_x\ Inj_{xy}$	$d_x\ d_x\ Inj_{xy}$	$d_y\ t_{xy}\ Inj_{yx}$	$d_y\ d_y\ t_{xy}$

Table 1. VC Buffer Configuration for the Three Routing Algorithms

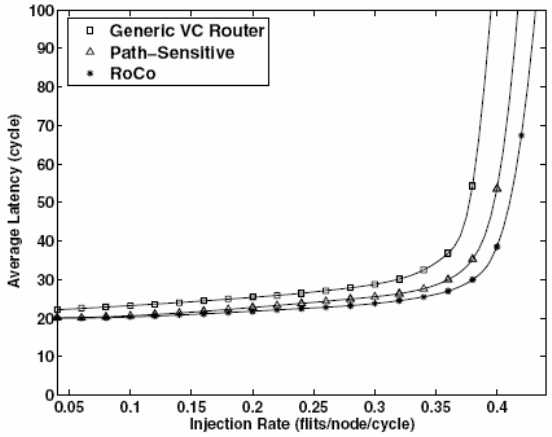


RoCo Router

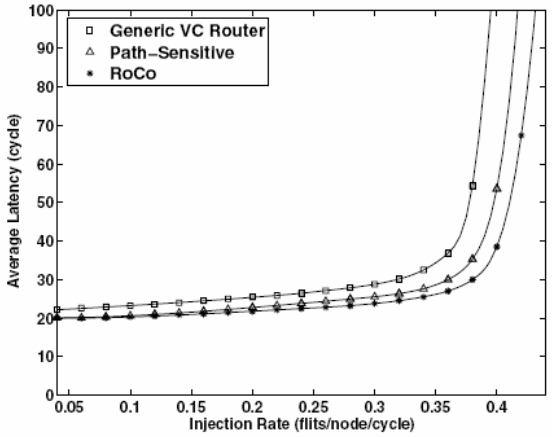
Key features:

- Early ejection mechanism for flits destined for the PE (saves 2 cycles since they don't have to go through SA and xbar stages)
- Flits are steered to the appropriate crossbar thanks to routing info computed in previous stage – enables use of 2 2x2 crossbars instead of 1 5x5 crossbar
- Results show much lower contention probability for RoCo (?!)
- Need fewer and smaller arbiters: 2x2 xbar arbiter algorithm: (mirroring)
  Generic case: for each input port, one arbiter selects the winner
  for each output port, an arbiter selects the winner
  RoCo: for each input port, two arbiters select two winners
  for each 2x2 xbar, one arbiter selects the winner for one port
  and the outcome is mirrored on the 2nd port

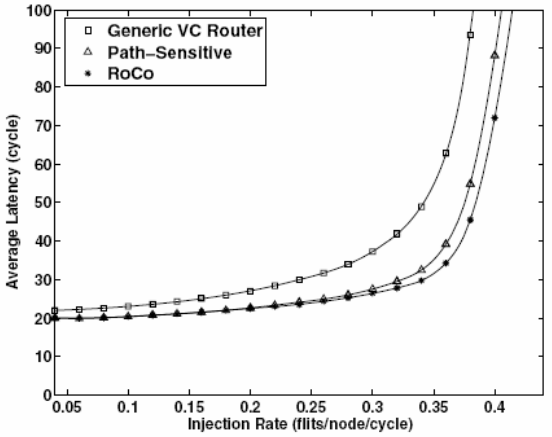
Results



(a) Deterministic Routing



(b) XY-YX Routing



(c) Adaptive Routing

Figure 8. Uniform Random Traffic

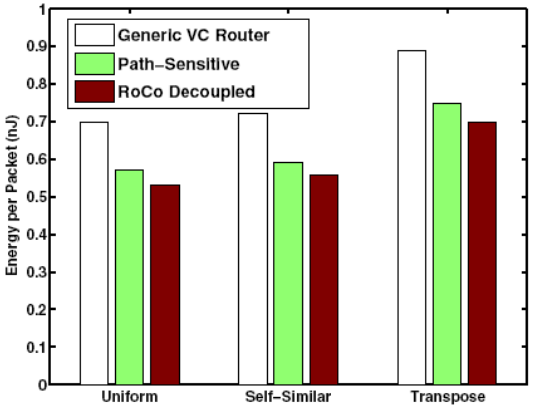
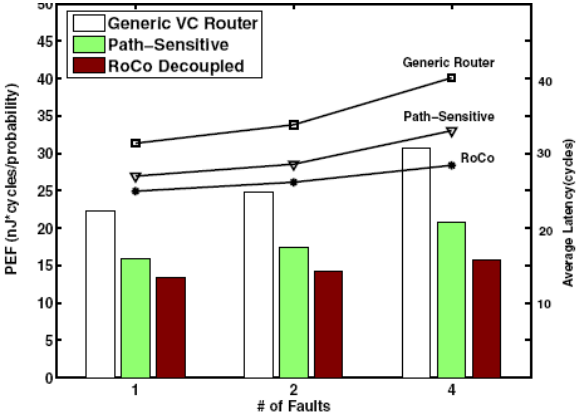
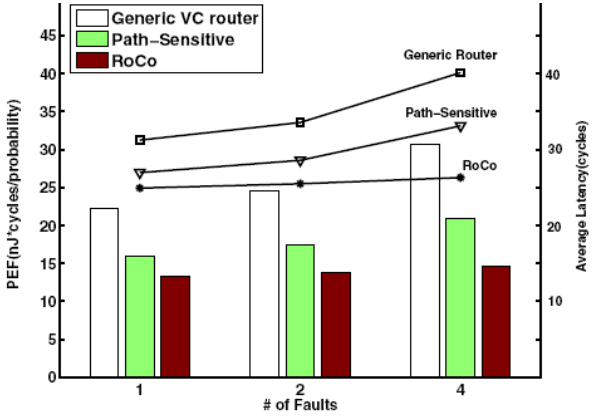


Figure 13. Energy per Packet



(a) Critical Region Fault



(b) Non-Critical Region Fault

Figure 14. Performance-Energy-Fault (PEF) Product

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)

Examples of Buffer Inefficiencies

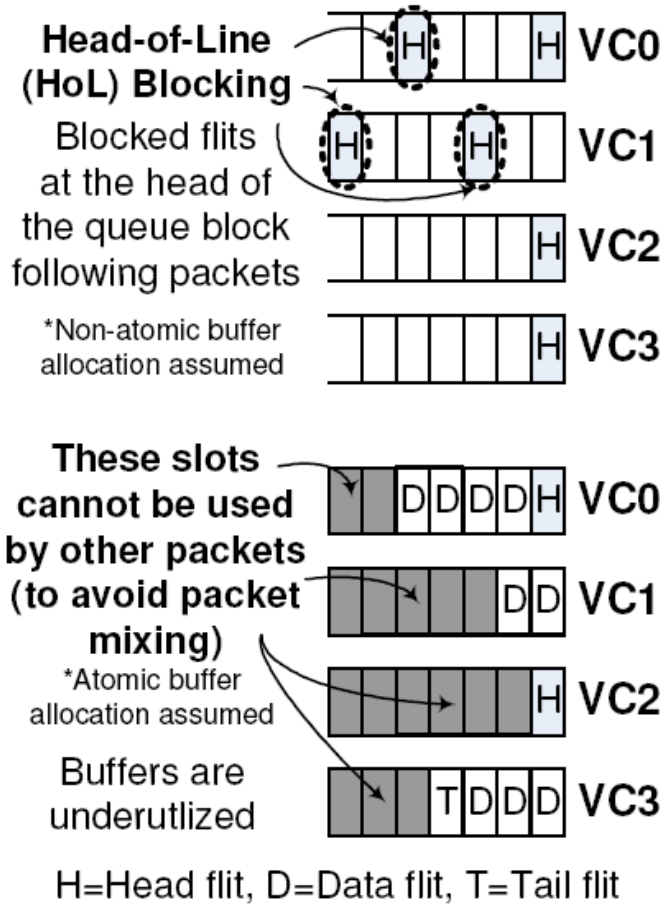


Figure 3. Limitations of Existing FIFO Buffers

Proposed Architecture

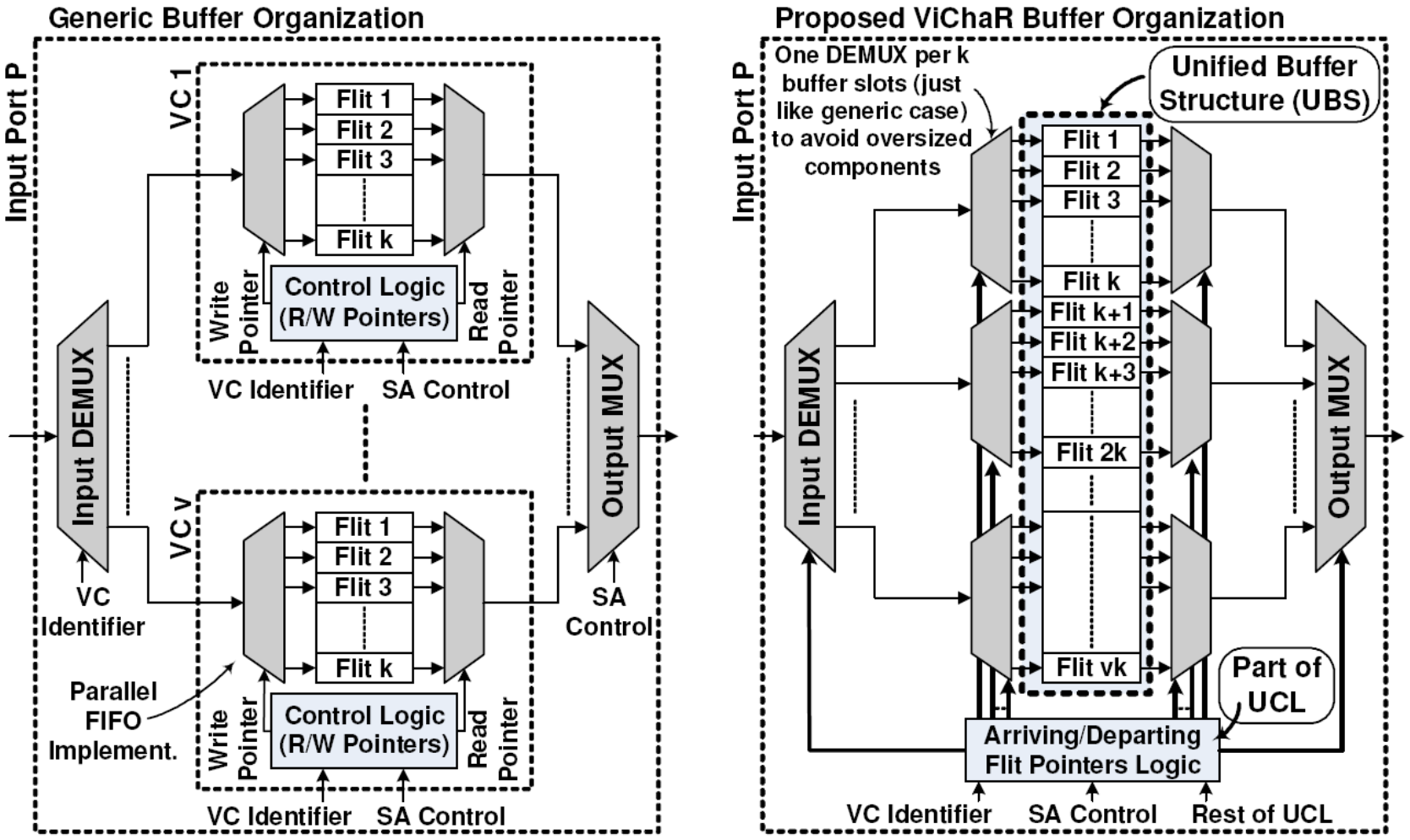


Figure 4. The Proposed ViChaR Architecture

Unified Buffer Design

- What does it take to design such a router?

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

Arbitration

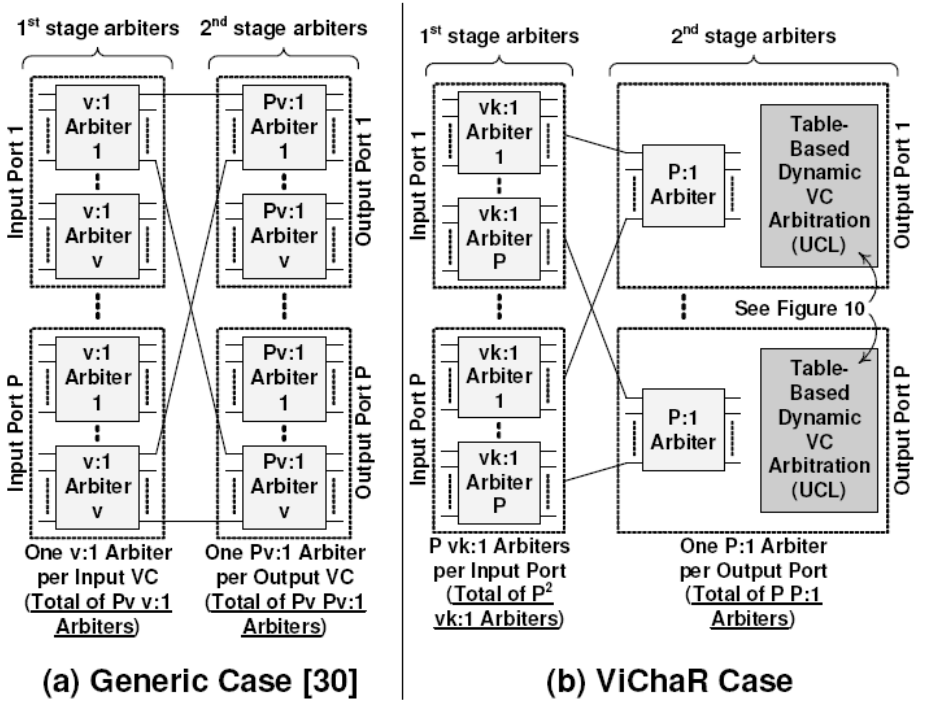


Figure 7. Virtual Channel Arbitration (VA)

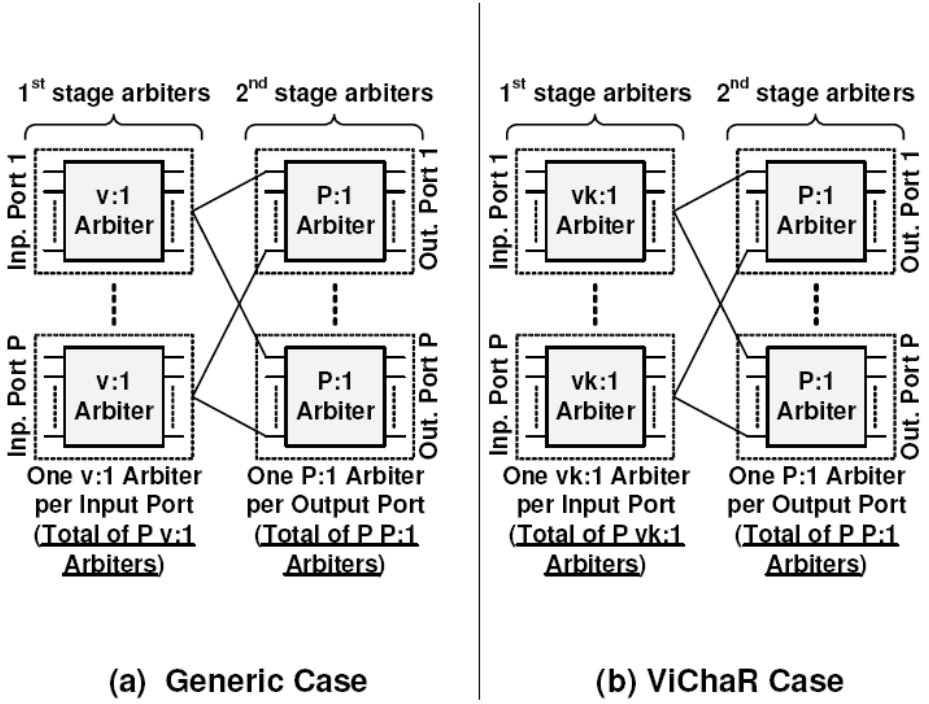


Figure 8. Switch Allocation (SA)

Area/Power

Table 1. Area and Power Overhead of the ViChaR Architecture. The results in this table assume *equal-size buffers* for both router designs. However, ViChaR's efficient buffer management scheme allows for a 50% decrease in buffer size with no performance degradation (see Section 4). In such a case, *area and power is reduced by 30% and 34%, respectively, over the whole router.*

Component (one input port)	Area (in μm^2)	Power (in mW)
ViChaR Table-Based Contr. Logic	12,961.16	5.36
ViChaR Buffer Slots (16 slots)	54,809.44	15.36
ViChaR VA Logic	27,613.54	8.82
ViChaR SA Logic	6,514.90	2.06
TOTAL for ViChaR Architecture	**101,899.04**	**31.60**
Generic Control Logic	10,379.92	5.12
Generic Buffer Slots (16 slots)	54,809.44	15.36
Generic VA Logic	38,958.80	9.94
Generic SA Logic	2,032.93	0.64
TOTAL for Gen. Architecture	**106,181.09**	**31.06**
ViChaR Overhead / Savings	**- 4,282.05 4.03% SAVINGS**	**+ 0.54 1.74% OVERHEAD**

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