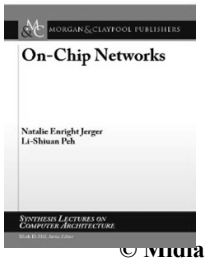
Routing algorithms

Midia Reshadi, Ph.D

CE Department of
Science and Research Branch
Islamic Azad University
(SRBIAU)
reshadi@ieee.org
http://m-reshadi.net

Outline

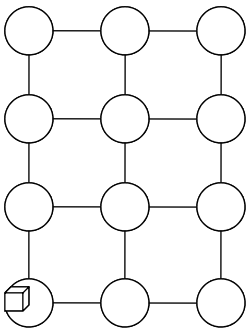
- Types of the routing
- Deadlock
- References: main reference



Basic Definition

- What is routing algorithm?
- Types of routing algorithm:
  - Deterministic
    - All messages from node A to B will always traverse the same path
  - oblivious
  - Adaptive
- Deterministic routing in on-chip networks:
  - Dimension-Ordered Routing (DOR)
    - A message traverses the network dimension by dimension, reaching the ordinate matching its destination before switching to the next dimension.
    - Simplicity

Example:: DOR

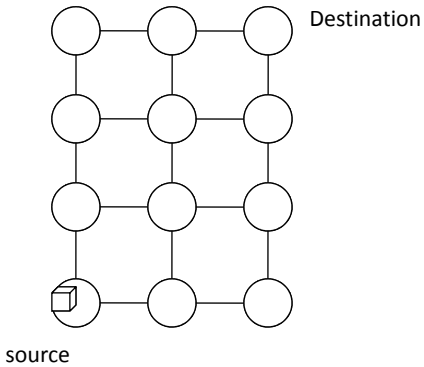


XY routing pseudo code

Algorithm: XY Routing for 2-D Meshes
Inputs: Coordinates of current node (*Xcurrent, Ycurrent)*
and destination node (*Xdest, Ydest)*
Output: Selected output *Channel*
Procedure:
Xoffset := Xdest − Xcurrent;
Yoffset :=Ydest −Ycurrent;
if *Xoffset < 0 then*
Channel := X−;
endif
if *Xoffset > 0 then*
Channel := X+;
endif
if *Xoffset = 0 and Yoffset < 0 then*
Channel := Y−;
endif
if *Xoffset = 0 and Yoffset > 0 then*
Channel := Y+;
endif
if *Xoffset = 0 and Yoffset = 0 then*
Channel := Internal;
endif

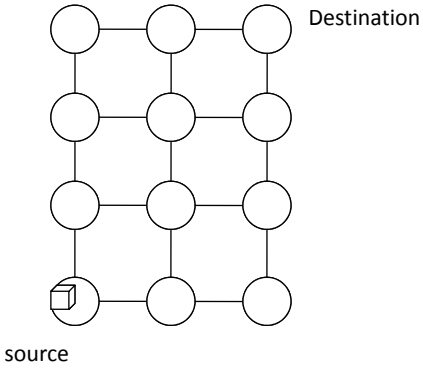
oblivious routing

- **The messages traverse different paths from A to B, but the path is selected without regard to network congestion.**



Adaptive routing

- The path a message takes adaptive routing from A to B depends on network traffic situation



minimal and non-minimal routing

- Minimal routing:
  - select only paths that require the smallest number of hops between the source and the destination
- Non-minimal routing
  - allow paths to be selected that may increase the number of hops between the source and destination.
- In the absence of congestion
  - Non-minimal routing increases latency and power > minimal routing

Deadlock avoidance

- A deadlock occurs when a cycle exists among the paths of multiple messages.

Deterministic Dimension ordered routing

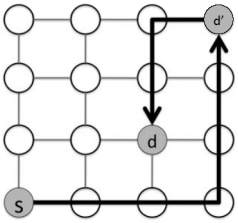
- Routing turns

(a) All turns (b) X-Y turns

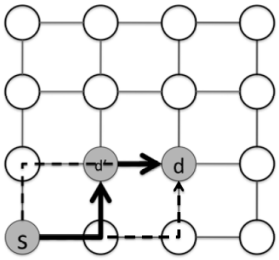
- To prevent these cyclic dependencies, turns may be disallowed.

oblivious routing

- Routing paths are chosen without regard to the state of the network.
- Example: Valiant's randomized routing algorithm
- This routing can be restricted to support only minimal routes



(a) Valiant's Routing Algorithm

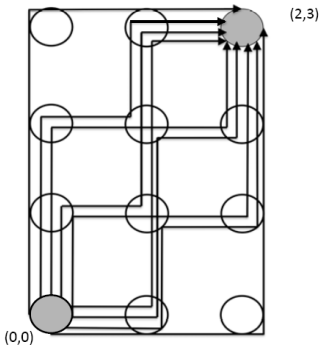


(b) Minimal Oblivious Routing Algorithm

Adaptive routing

- Routing decision may be based on:
  - Local information
    - Queue occupancy
  - Global information

All possible (minimal) routes that a message can take from Node (0,0) to Node (2,3).

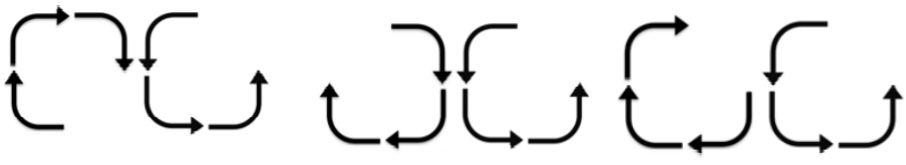


Adaptive routing

- Adaptive routing + non-minimal path (misroutes)→ livelock problem
- Fully adaptive routing→ deadlock problem
  - Solution: restricting adaptivity to only two dimensions at a time. (Planar-adaptive routing)

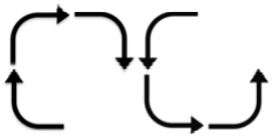
Adaptive turn model routing

- Adaptive turn model routing eliminates the minimum set of turns needed to achieve deadlock freedom
- Allowing six out of eight turns
- Only one turn from each cycle is eliminated



West First Turns | North Last Turns | Negative First Turns

Adaptive turn model routing



West First Turns

the North to West turn, the South to West turn is eliminated.

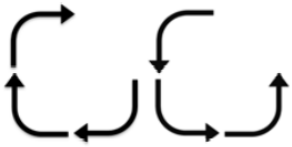
A message must first travel in the West direction before traveling in any other direction.

North Last Turns

eliminates both the North toWest and the North to East turns.

Once a message has turned North, no further turns are permitted; hence, the North turn must be made last.

Adaptive turn model routing



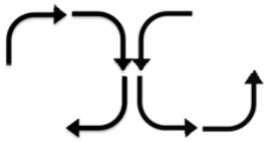
Negative First Turns

Removes turns from North to West and East to South to create the Negative-First algorithm

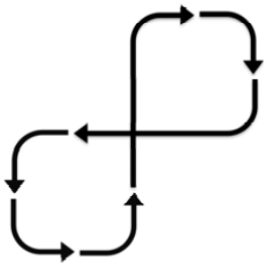
A message travels in the negative directions (west and south) first before it is permitted to travel in positive directions (east and north).

Adaptive turn model routing

- A possible turn elimination that is not deadlock-free:
- elimination of North to West + the elimination of West to North



Illegal Turn Model Routing



Resulting DeadlockCycle

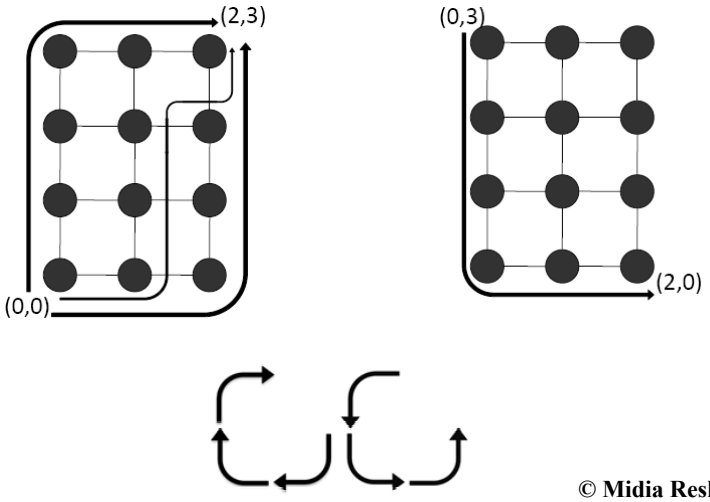
Odd-even turn model routing

- eliminating a set of two turns depending on whether the current node is in an odd or even column.
- If packet is in an even column1:
  - turns from East to North and from North to West are prohibited
- If packets is in odd column node
  - turns from East to South and from South to West are prohibited.
- 180◦ turns are disallowed.

Comparison

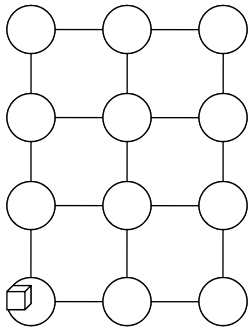
- The odd-even turn model provides better adaptivity than other turn model algorithms such as West-First.
- Why?

Example: The Negative-First turn model routing



implementation

- Source routing
  - The route can be embedded in the packet header at the source
- X-Y routing:
  - *< E, E, N, N, N, Eject >*
- Y-X route:
  - *< N, N, N, E, E, Eject >*



Benefits

- Saving latency:
  - The route does not need to be computed or looked up at each hop
- The per-router routing hardware is also saved
  - no combinational routing logic or routing tables are needed
- Reconfiguration of source routing tables to deal with faults and can support irregular topologies

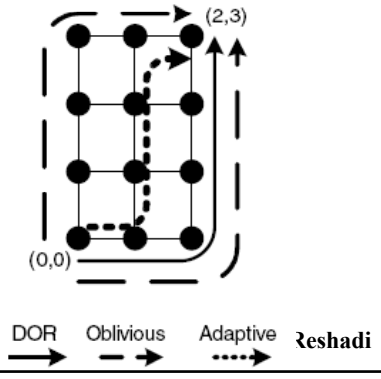
Benefits

- Multiple routes per source-destination pair can be stored in the table
- Random selection for each packet to improve load balancing.

The disadvantages of source routing

- The bit overheads:
  - To store the routing table at the network interface
  - To store the entire routing path in each packet

Destination	Route 0	Route 1
00	X	X
10	EX	EX
20	EEX	EEX
01	NX	NX
11	NEX	ENX
21	NEEX	ENEX
02	NNX	NNX
12	ENNX	NENX
22	EENNX	NNEEX
03	NNNX	NNNX
13	NENNX	ENNNX
23	EENNNX	NNNEEX



Node table-based routing

- Using routing tables at each hop:
  - It store the outgoing channel

West-first turn model algorithm for 3-ary2-mesh

	To								
From	00	01	02	10	11	12	20	21	22
00	X -	N -	N -	E -	E N	E N	E -	N E	N E
01	S -	X -	N -	E S	E -	E N	E S	E -	E N
02	S -	S -	X -	E S	E S	E -	E S	E S	E -
10	W -	W -	W -	X -	N -	N -	E -	E N	E N
11	W -	W -	W -	S -	X -	N -	E S	E -	N E
12	W -	W -	W -	S -	S -	X -	E S	E S	E -
20	W -	W -	W -	W -	W -	W -	X -	N -	N -
21	W -	W -	W -	W -	W -	W -	S -	X -	N -
22	W -	W -	W -	W -	W -	W -	S -	S -	X -

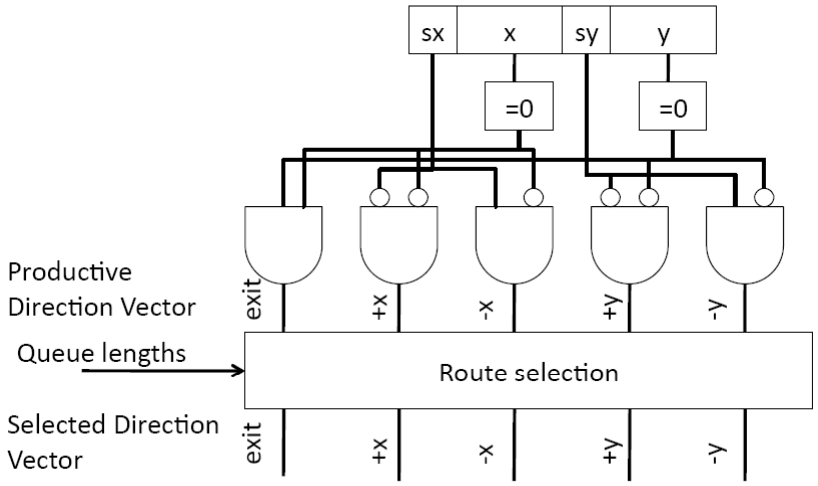
Advantages & Disadvantages

- Advantages
  - Smaller routing table
  - Supporting adaptively and fault tolerance
  - Programmability
- Disadvantages
  - Increasing the packet delay
    - a look-up is expended at each hop

Combinational circuits

- Higher:
  - Speed
  - Simplicity
- Lower overhead
- Specific to:
  - One topology
  - One routing algorithm
- No generality and configurability

Example



Adaptive routing

- Need mechanisms to track network congestion levels, and update the route
- Advantages:
  - Achieving higher bandwidth and reducing the congestion latency
- The disadvantage
  - Additional circuitry
  - increasing the latency of a routing decision
  - increasing area of the router

Routing irregular toplogies

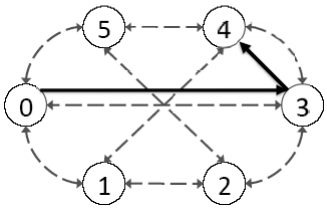
- Based on source table routing or node-table routing

Bibliographic notes

- **IBMCell**
  - Bus based communication
- **IntelTeraFLOPS**
  - Source-based routing
  - 3 bits for naming 5 ports
- **Tilera TILE64**
  - Dimension order routing
  - X-Y style addressing

Bibliographic notes

- Spirdergon
  - Across-first routing



Example

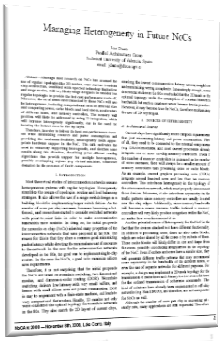
NoCArc

First International Workshop on

Network on Chip Architectures

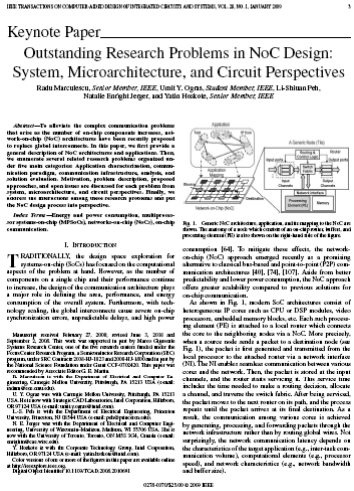
(NoCArc 2008)

In conjunction with the
41st Annual IEEE/ACM International Symposium on Microarchitecture
(MICRO-41)
November 8th, 2008
Lake Como, Italy



Managing Heterogeneity in Future NoCs
Jose Duato
Parallel Architectures Group
Technical University of Valencia, Spain
email: jduato@disca.upv.es

The Future



Keynote Paper

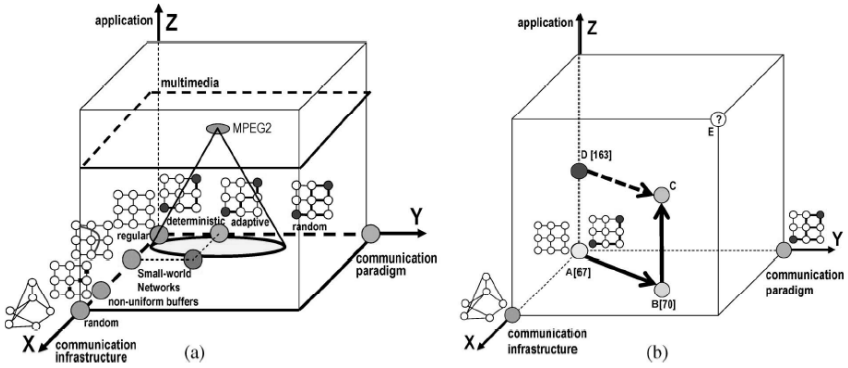
Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives

Radu Marculescu, *Senior Member, IEEE*, Umit Y. Ogras, *Student Member, IEEE*, Li-Shiuan Peh, Natalie Enright Jerger, and Yatin Hoskote, *Senior Member, IEEE*

Outstanding Research Problems in NoC Design: System, Microarchitecture, and Circuit Perspectives
Radu Marculescu,, *Umit Y. Ogras,, Li-Shiuan Peh,* Natalie Enright Jerger, and Yatin Hoskote,

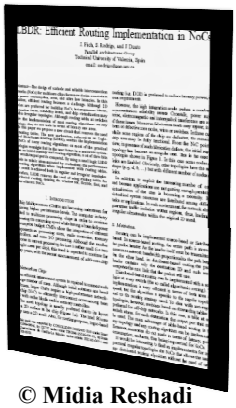
© Midia Reshadi

Big Picture



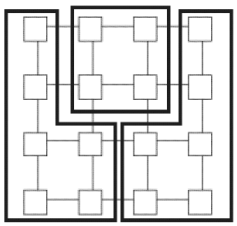
State of the art in routing

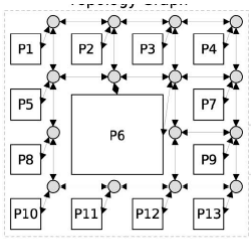
- **LBDR: Efficient Routing Implementation in NoCs**
  - **J. Flich, S. Rodrigo, and J. Duato**



LBDR

- Motivation
  - Irregular topologies
  - virtualization



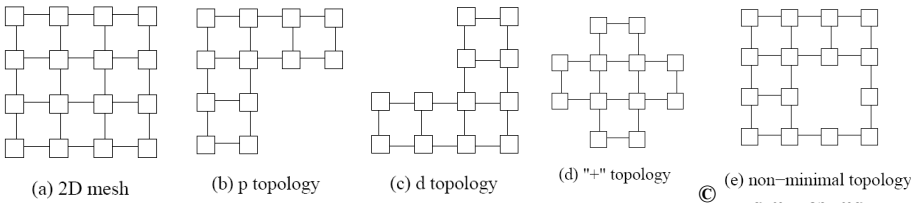


LBDR

- No virtual channel
- Switching strategy: wormhole or virtual cut-through
- Address: X & Y
  - X_{dest}, Y_{dest}, X_{curr}, Y_{curr}

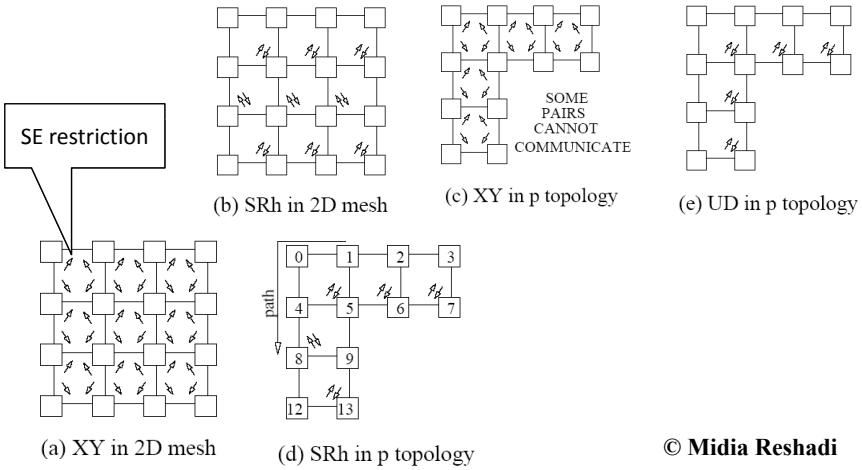
Conditions

- At least one end-node (IP core) attached to each switch
- All node pairs can communicate with the rest of nodes through a minimal path defined in the original mesh topology
- Example:



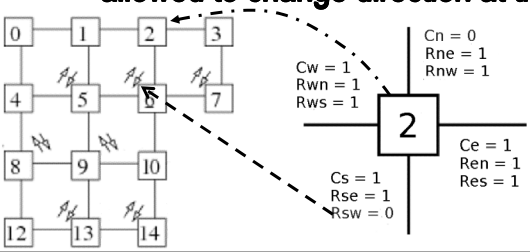
Routing restrictions

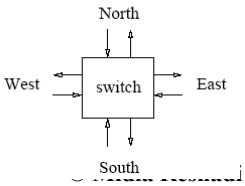
- Routing algorithms: SRh, XY, UD



Port naming

- **Naming: by 3bits per each output port**
  - **Connectivity bits indicate whether a switch is connected with its neighbors**
  - **Routing bits indicate which routing options can be taken**
  - **In other words, these bits indicate whether packets are allowed to change direction at the next switch**





Port naming

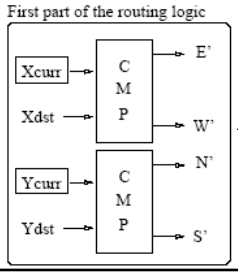
Routing and connectivity bits required per switch (12 bits, 3 per output port)

bits at the North port	Rne, Rnw	bits at the West port	Rwn, Rws	Cn, Ce
bits at the East port	Ren, Res	bits at the South port	Rse, Rsw	Cw, Cs

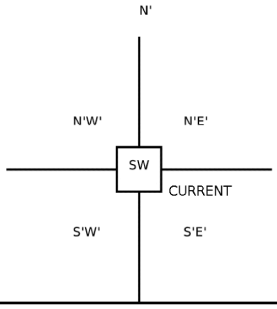
Whether packets routed through the E output port may later at the next switch take the N port or S port

The routing logic

- is divided in two parts
  - The first part
    - Computes the relative position of the packet's destination.
  - Two comparators are used and X_{curr} and Y_{curr} are compared with X_{dst} and Y_{dst}

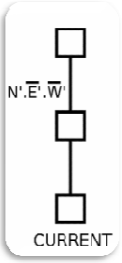


Outputs: If the packet is in the NW quadrant then N′ and W′ signals are set



The routing logic

- Second part:
- Example: N output port is considered for routing the incoming packet
  - Following –three- conditions is met:
    - The packet's destination is on the same column



Second part of the routing logic

$N'' = N' \cdot \overline{E'} \cdot \overline{W'} + N' \cdot E' \cdot Rne + N' \cdot W' \cdot Rnw$

$E'' = E' \cdot \overline{N'} \cdot \overline{S'} + E' \cdot N' \cdot Ren + E' \cdot S' \cdot Res$

$W'' = W' \cdot \overline{N'} \cdot \overline{S'} + W' \cdot N' \cdot Rwn + W' \cdot S' \cdot Rws$

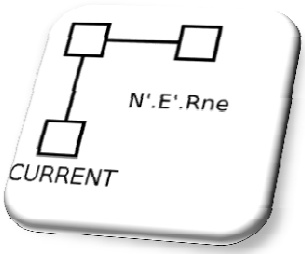
$S'' = S' \cdot \overline{E'} \cdot \overline{W'} + S' \cdot E' \cdot Rse + S' \cdot W' \cdot Rsw$

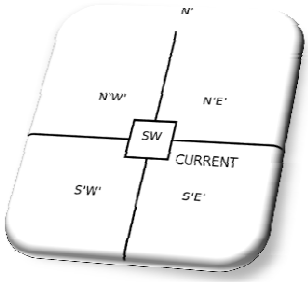
$N = N'' \cdot Cn$ $\quad W = W'' \cdot Cw$

$E = E'' \cdot Ce$ $\quad S = S'' \cdot Cs$

The routing logic

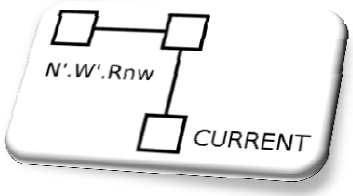
– The packet's destination is on the NE quadrant and the packet can take the E port at the next switch through the N port

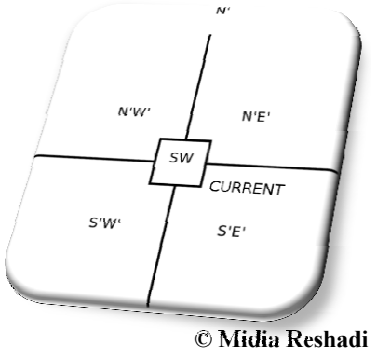




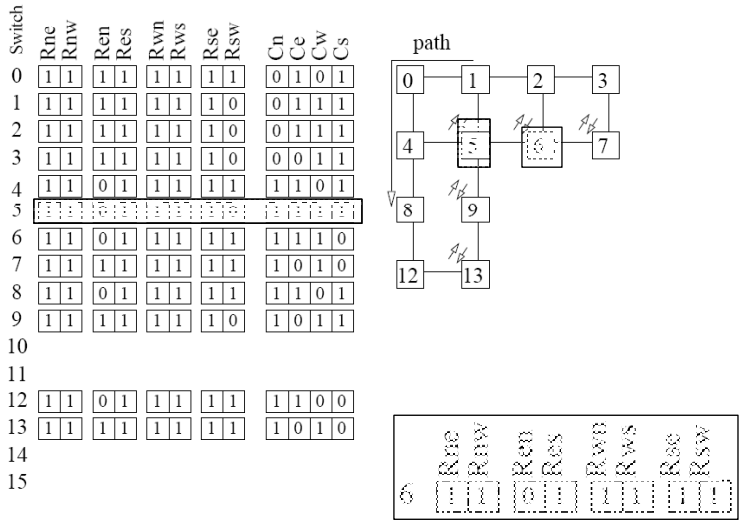
The routing logic

– The packet's destination is on the NW quadrant and the packet can take the W port at the next switch through the N port

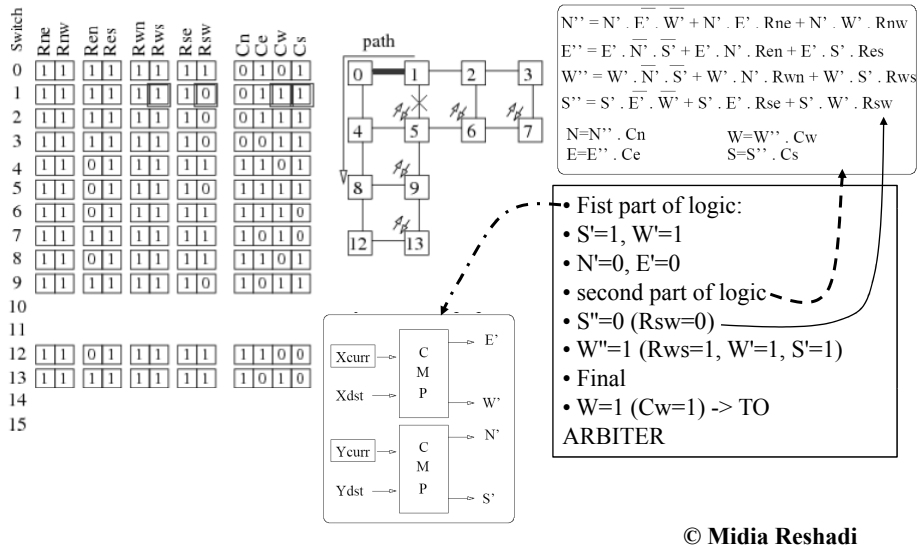




Functionality



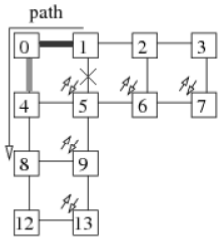
Functionality(Cont,…)



- Fist part of logic:
- S'=1, W'=1
- N'=0, E'=0
- second part of logic
- S''=0 (Rsw=0)
- W''=1 (Rws=1, W'=1, S'=1)
- Final
- W=1 (Cw=1) -> TO ARBITER

Functionality(Cont,…)

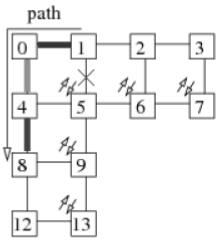
Switch	Rne	Rnw	Ren	Res	Rwn	Rws	Rse	Rsw	Cn	Ce	Cw	Cs
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	1	1	1	1	0	1
5	1	1	0	1	1	1	1	0	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	0	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	1	1	0	1	0	1	1
10												
11												
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0
14												
15												



- 1st part of logic:
- S'=1
- W'=0, N'=0, E'=0
- 2nd part of logic
- S''=1 (S'=1, E'=0, W'=0)
- Final
- S=1 (Cs=1) -> TO ARBITER

Functionality(Cont,…)

Switch	Rne	Rnw	Ren	Res	Rwn	Rws	Rse	Rsw	Cn	Ce	Cw	Cs
0	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	0	0	1	1	1
2	1	1	1	1	1	1	1	0	0	1	1	1
3	1	1	1	1	1	1	1	0	0	0	1	1
4	1	1	0	1	1	1	1	1	1	1	0	1
5	1	1	0	1	1	1	1	0	1	1	1	1
6	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	1	1	1	0	1	0
8	1	1	0	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	1	1	0	1	0	1	1
10												
11												
12	1	1	0	1	1	1	1	1	1	1	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0
14												
15												



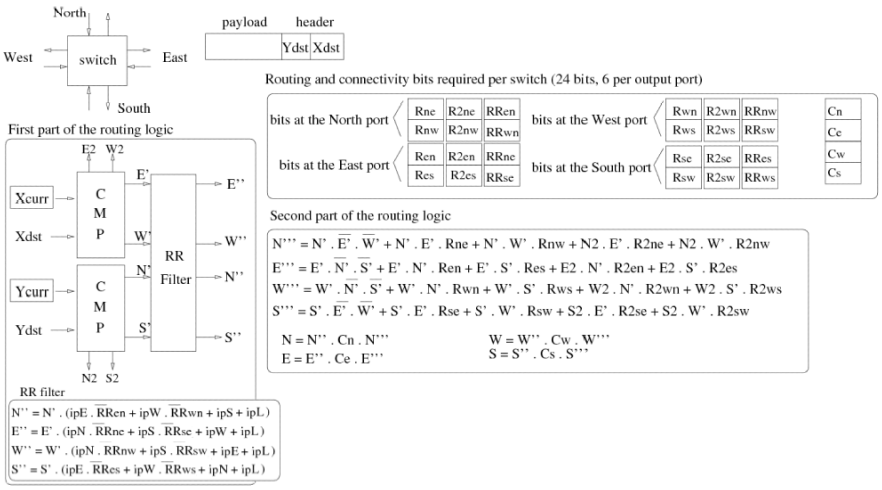
- 1st part of logic:
- S'=1
- W'=0, N'=0, E'=0
- 2nd part of logic
- S''=1 (S'=1, E'=0, W'=0)
- Final
- S=1 (Cs=1) -> TO ARBITER

© Midia Reshadi

LBDRe

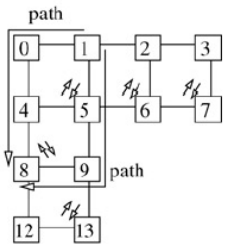
- LBDR has visibility of one hop away -> LBDRe expands visibility to two hops away
- LBDRe adds four more bits per ouput port. It is a second set of routing bits (R2xy), meaning that y direction can be taken two hops away through the x direction

LBDRe::Logic



LBDRe::naming

Switch	Rne	Rnw	R2ne	R2nw	Ren	Res	R2en	R2es	Rwn	Rws	R2wn	R2ws	Rse	Rsw	R2sc	R2sw	RRne	RRnw	RRen	RRes	RRwn	RRws	RRse	RRsw	Cn	Ce	Cw	Cs
0	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
1	1	1	0	0	1	1	0	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1
2	1	1	0	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
3	1	1	0	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	1	1	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	1
5	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	1	1	1
6	1	1	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	1	1	1	0
7	1	1	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0
8	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0	0	1	1	0	1
9	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
10																												
11																												
12	1	1	1	0	0	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0
13	1	1	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0
14																												
15																												



Why LDBRe?

Switch	Rne	Rnw	R2ne	R2nw	Ren	Res	R2en	R2es	Rwn	Rws	R2wn	R2ws	Rse	Rsw	R2sc	R2sw	RRne	RRnw	RRen	RRes	RRwn	RRws	RRse	RRsw	Cn	Ce	Cw	Cs
0	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
1	1	1	0	0	1	1	0	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1
2	1	1	0	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
3	1	1	0	0	1	1	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	1	1	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	1
5	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	1	1	1
6	1	1	0	0	0	1	0	0	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	1	1	1	0
7	1	1	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0
8	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0	0	1	1	0	1
9	1	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
10																												
11																												
12	1	1	1	0	0	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0
13	1	1	1	1	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0	1	0
14																												
15