

# 2.5D Network on Chip Router

# Προτάσεις Υλοποίησης

## Τοπολογία :

- *Πλέγμα* (2D-4x4 mesh)
- *5 Θύρες* (N,S,E,W,Local ή Core)

## Δρομολόγηση (Routing) :

- *Minimal Adaptive Routing* με *XY Escape VCs*
- *Πρωτόκολλο Duato*

## Flow Control :

- *Virtual-Channel* Flow Control (Ίσως 2 VCs των 4 flits ~ Παραμετρ.)
- *Credit-Based* Backpressure Mechanism

# Προτάσεις Υλοποίησης

## Επαναδέσμευση Καναλιού :

- **Συντηρητική** επαναδέσμευση καναλιού (αποφυγή **εξάρτησης**)

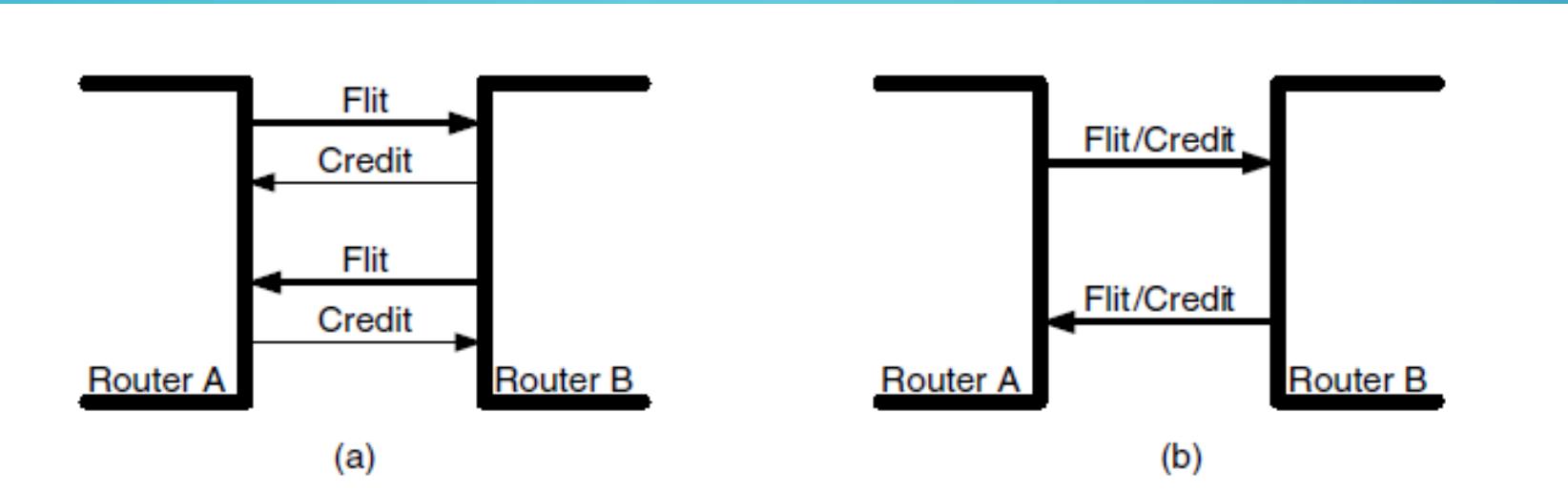
## Speculation – Next Routing Computation (NRC) :

- **Μόνο αν** το επιτρέπει η υλοποίηση, εξαρτάται από **άλλες παραμέτρους**

## Οργάνωση Buffers και Switch :

- **Κυκλικοί** buffers ανά VC, αν χρειαστεί αναβάθμιση, **αργότερα**
- 5x5 **Crossbar Switch** με κάποιο **speedup** (μάλλον input speedup = 2)

# Κωδικοποίηση Flit και Credit



(b) Διαμοιρασμός ζεύξης ανάμεσα σε Flits και Credits

# Κωδικοποίηση Flit και Credit

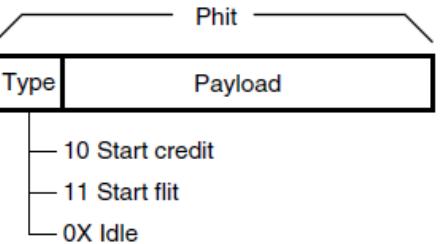
Με την χρήση διαφορετικών ζεύξεων για τις δύο οντότητες :

- Ανεκμετάλλευτο **bandwidth** όταν δεν αποστέλλονται **Credit**
- Υψηλότερη **καθυστέρηση** λόγω **στενότητας** του **Credit** καναλιού

Μέθοδοι μείωσης ζεύξεων και καλωδίων (με κωδικοποίηση) :

- Αποστολή Credits **ενσωματωμένων** στα Flits που ταξιδεύουν **αντίθετα (piggybacking)**
- **Χρονική Πολυπλεξία** με κωδικοποίηση στο επίπεδο των **Phits**

# Κωδικοποίηση Flit και Credit

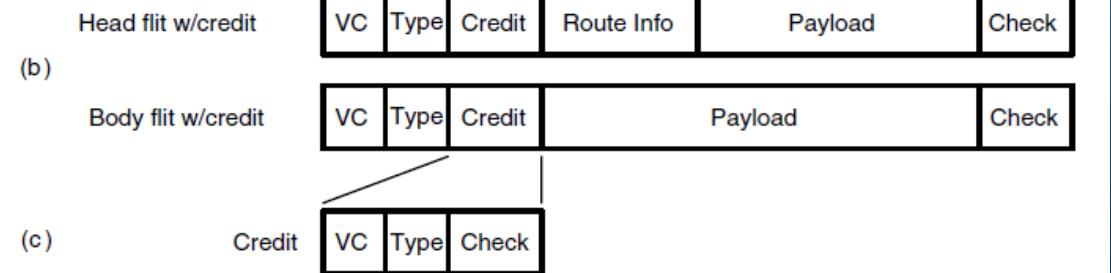


(a)

11	Flit payload
	Flit payload
	Flit payload
00	
10	Credit payload
11	Flit payload
	Flit payload
	Flit payload

(b)

Κωδικοποίηση σε επίπεδο Phit



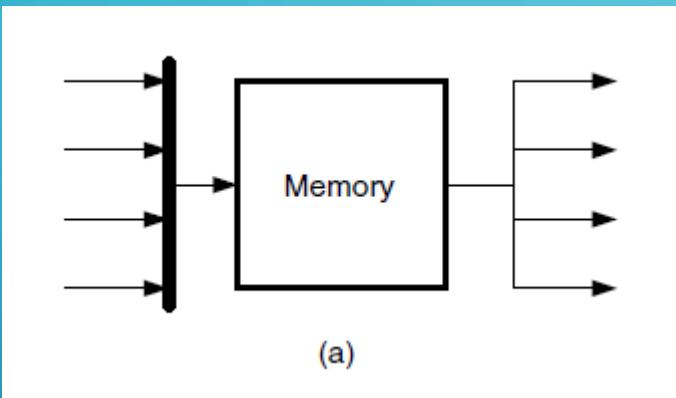
Κωδικοποίηση για piggybacked credits

# Router Datapath Components

Τα στοιχεία διαδρομής των δεδομένων αποτελούνται από :

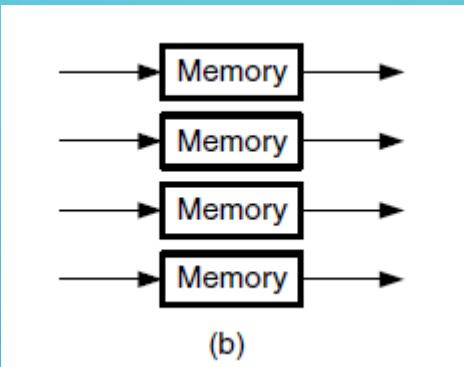
1. *Buffers εισόδου (Input Buffers)*
2. *Μεταγωγέας (Switch)*
3. *Movάδα Εξόδου (Output Unit)*

# Διαμέριση των Buffers



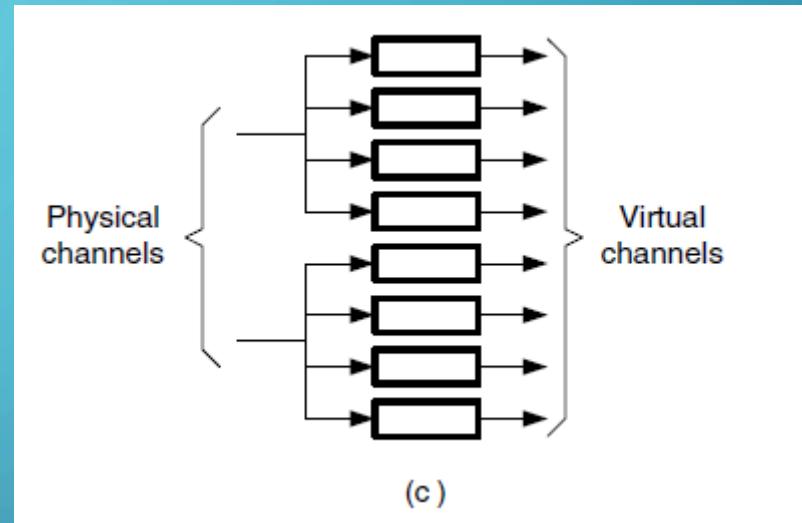
(a) Κοινή μνήμη για όλα τις θύρες εισόδου

- Οι θύρες μοιράζονται δυναμικά μνήμη
- Μεγάλη μνήμη για επαρκές bandwidth
- Υψηλή καθυστέρηση λόγω (de)serializers



(b) Μνήμη ανά φυσικό κανάλι (θύρα)

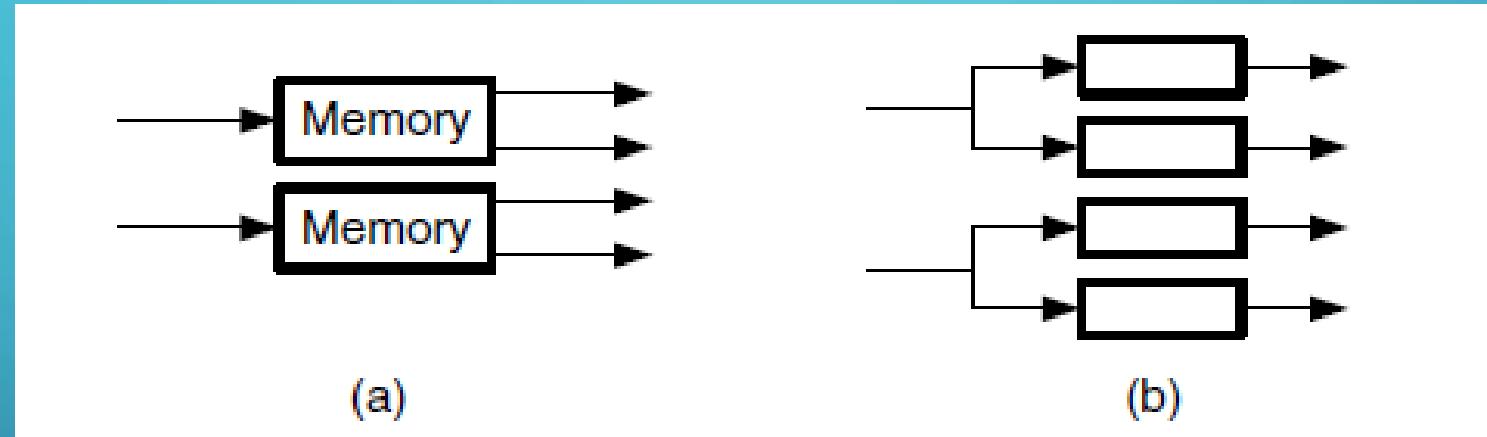
- Κάθε μνήμη χρειάζεται το διπλάσιο bandwidth μόνο της θύρας
- Τα εικ. κανάλια ανά θύρα μοιράζονται δυναμικά μνήμη



(c) Μνήμη ανά εικονικό κανάλι

- Δημιουργία *speedup* για καλύτερο *throughput*
- Κακή διαχείριση και χρήση μνήμης

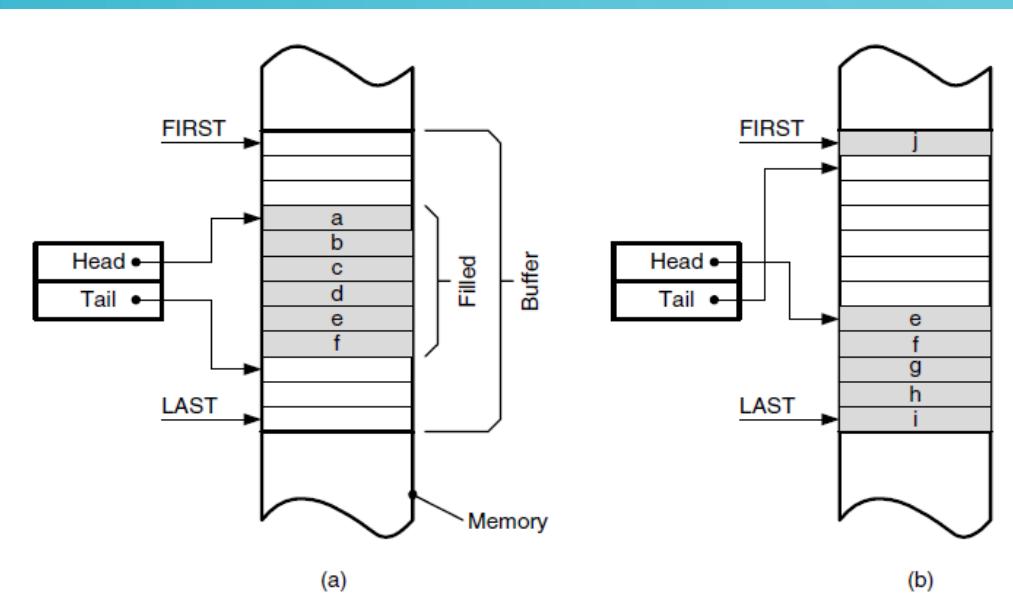
# Διαμέριση των Buffers



(a) Μνήμη ανά θύρα με πολλαπλές εξόδους

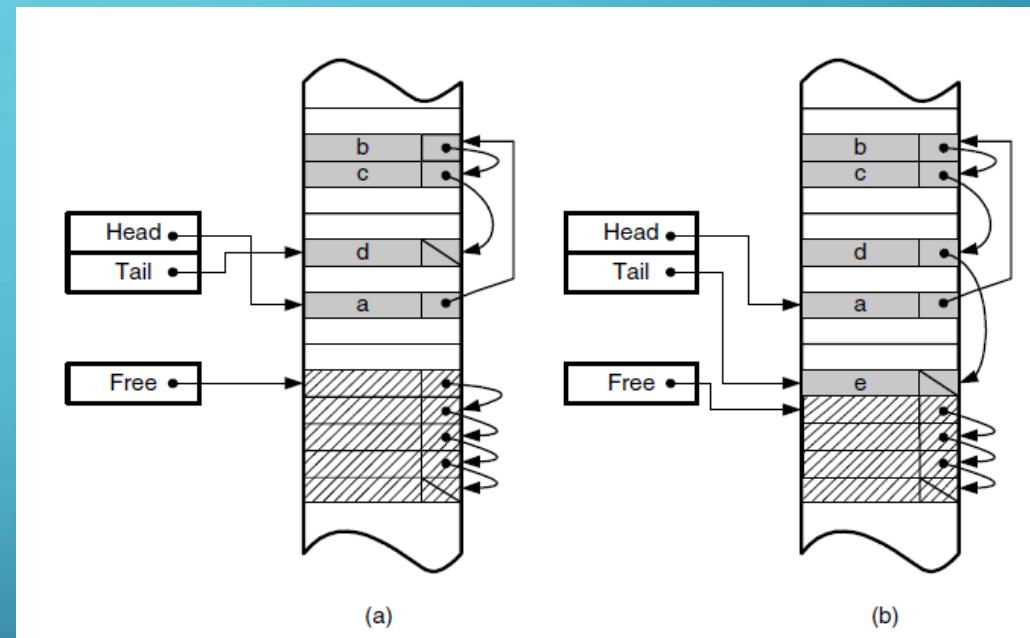
(b) Ομαδοποιημένη μνήμη με πολλαπλές εξόδους

# Δομές των Buffer εισόδου



Κυκλικός buffer

- Απλή κυκλική δομή
- Λιγότερη ευελιξία
- Για κοινή μνήμη, δεν συνηθίζεται



Συνδεδεμένη Λίστα

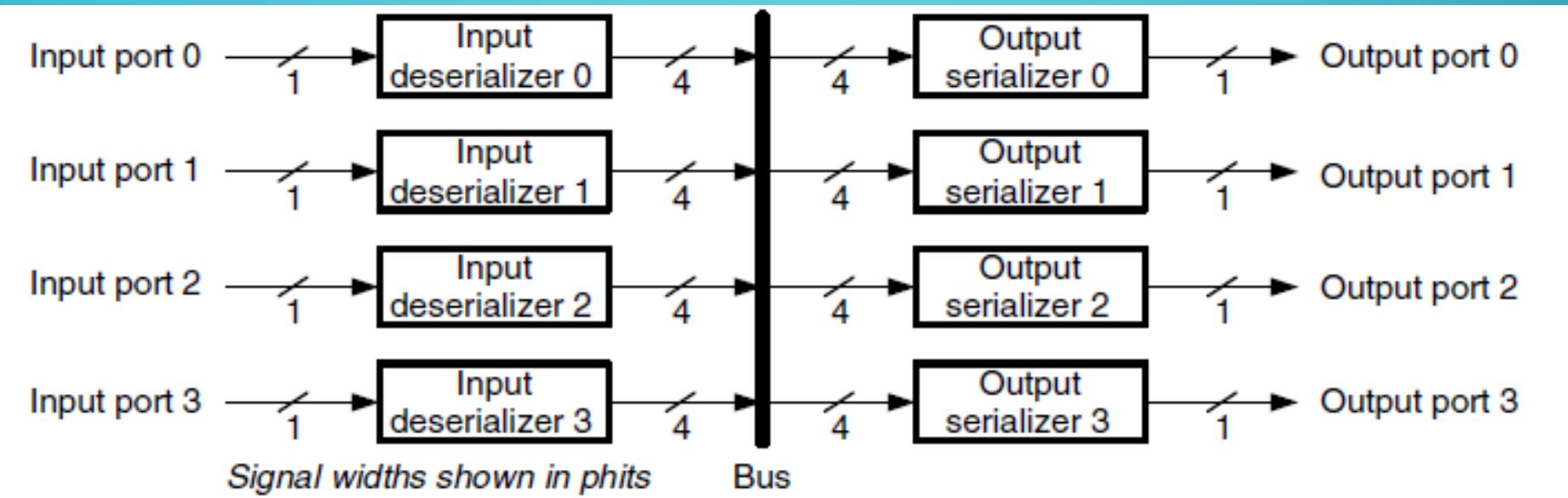
- Μεγαλύτερη ευελιξία (συνήθως για κοινή μνήμη)
- Δυσκολότερος χειρισμός και πιθανότητα σφαλμάτων

# Μεταγωγείς (Switches)

Ο μεταγωγέας είναι η καρδιά του Router :

- Η βασικότερη παράμετρος είναι το *speedup*
- Ο λόγος *παρεχόμενου bandwidth* προς το *ελάχιστο* για *full throughput* μεταξύ θυρών εισόδου και εξόδου
- Μειωμένο *latency* και αυξημένο *throughput*
- Πολυπλεξία *χώρου* ή/και *χρόνου*

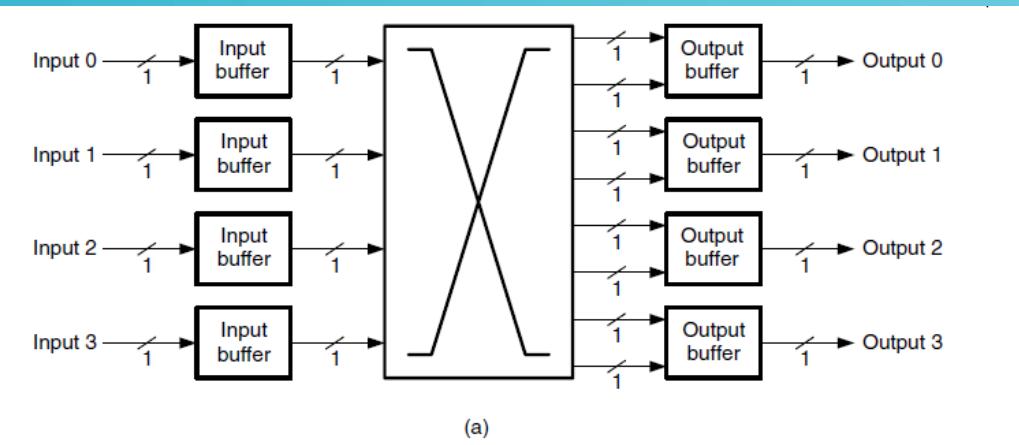
# Bus Switches



Ένας 4x4 bus μεταγωγέας χωρίς speedup, με deserializers/serializers. Κάθε flit αποτελείται από 4 phit.

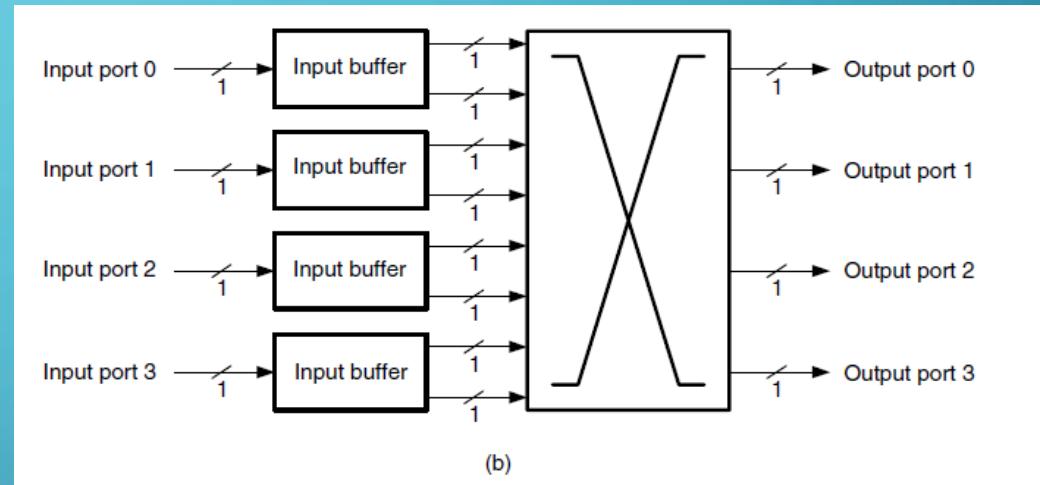
- Χρονική Πολυπλεξία ~ internal (phit) clock cycle (flit time  $\geq$  #ports)

# Crossbar Switches



(a)

(a) 4x4 Crossbar Switch με *output speedup* 2



(b)

(b) 4x4 Crossbar Switch με *input speedup* 2

- Πάλι χωρική ή χρονική πολυπλεξία
- Δεν υπάρχει νόημα για speedup > #ports
- Συνήθως προτιμάμε το input από το output speedup
- Το speedup διευκολύνει το έργο των allocators με μικρότερο κόστος



# Τέλος Παρουσίασης

Ερωτήσεις;