



Features of 2.5D Network-on-Chip Router

ΤΗΜΜΥ ΑΠΘ | ΤΟΜΕΑΣ ΗΛΕΚΤΡΟΝΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ

Χαραλάμπους Νικόλαος | Επιβλέποντες: κ. Παυλίδης Βασίλειος, κ. Τσεκούρας
Αριστοτέλης

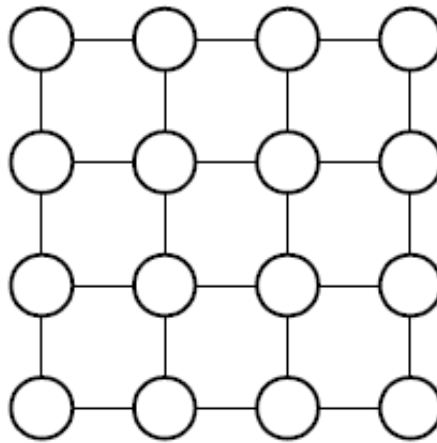
Εισαγωγή

Στο παρόν έγγραφο αναφέρονται επιγραμματικά **οι επιλογές σχεδίασης του 2.5D NoC Router** στα πλαίσια υλοποίησης της διπλωματικής μου εργασίας κατά το ακαδημαϊκό έτος 2025-2026 στον τομέα Ηλεκτρονικής & Υπολογιστών του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του ΑΠΘ. Σκοπός της παρόντος εγγράφου είναι **η συγκέντρωση και παρουσίαση των σχεδιαστικών επιλογών** που θα εφαρμοστούν στην υλοποίηση του κυκλώματος σε κώδικα RTL: SystemVerilog. Αξίζει να επισημανθεί πως **είναι πιθανή η διαρκής αλλαγή κι ανανέωση των επιλογών** αυτών στη διάρκεια του ακαδημαϊκού έτους, λόγω των μεταβολών που μπορεί να προκύψουν στο εγχείρημα της υλοποίησης. Αυτές οι αλλαγές, ωστόσο, θα αντανακλώνται στο παρόν έγγραφο μετά από κάθε τυχούσα ανανέωση.

1. Τοπολογία

Η τοπολογία αποτελεί **το πιο θεμελιώδες χαρακτηριστικό** ενός δικτύου διασυνδέσεων και σημαντικό παράγοντα για την απόδοση, την ταχύτητα και την συνολική δομή του δικτύου. Κάποιοι παράγοντες που λαμβάνονται υπόψη, κατά την επιλογή μιας τοπολογίας, είναι η **διάμετρος** (μέγιστη απόσταση ανάμεσα σε 2 κόμβους), ο **βαθμός διασύνδεσης** κάθε κόμβου, η **ποικιλία των μονοπατιών** (*path diversity*), το **μέσο μήκος ενός μονοπατιού** (εξαρτάται από τη διάμετρο), η **πολυπλοκότητα των διασυνδέσεων** και η **προσαρμογή σε διάφορες κλίμακες μεγέθους**. Λαμβάνοντας υπόψη τους παραπάνω παράγοντες, επιλέγουμε για την υλοποίηση του δικτύου την **τοπολογία πλέγματος** (*mesh topology*). Πιο συγκεκριμένα, επιλέγουμε **ένα 2D (δισδιάστατο) 4x4 πλέγμα (mesh) με 16 κόμβους**, το οποίο θα αποτελεί την τοπολογία του δικτύου διασύνδεσης μας. Έχοντας αυτό ως δεδομένο, ένας **δρομολογητής-router** αντιστοιχίζεται σε κάθε κόμβο του πλέγματος και πρέπει να σχεδιαστεί με αυτή την προοπτική. Για αυτόν το λόγο, κάθε κόμβος θα έχει **5 θύρες εισόδου-εξόδου**

(4 προς κάθε κατεύθυνση – Βόρεια, Νότια, Ανατολικά, Δυτικά και 1 τοπική προς τον πυρήνα του στοιχείου με το οποίο συνδέεται ο router).

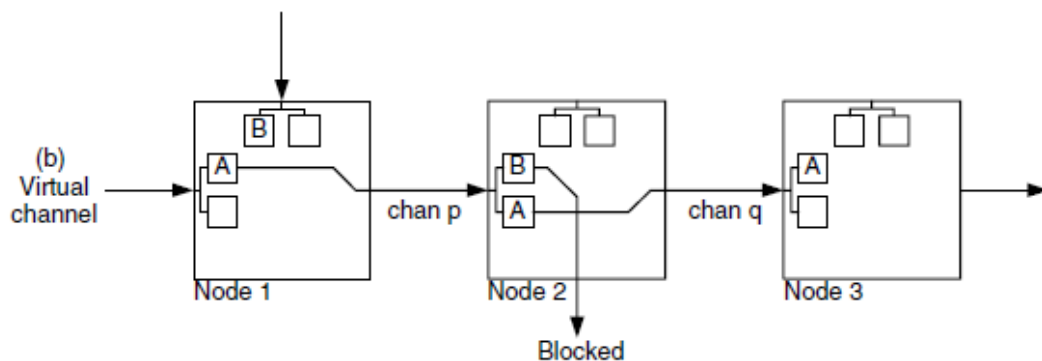


(b) A 4-ary 2-mesh

Εικόνα 1 - 2D 4x4 Τοπολογία Πλέγματος

2. Flow Control (Έλεγχος Ροής)

Το επόμενο βήμα στο πλαίσιο των σχεδιαστικών επιλογών, είναι να αποφασίσουμε με ποιον τρόπο θα πραγματοποιείται η ροή (αποστολή και λήψη) δεδομένων. Υπάρχουν αρκετές πολιτικές ροής ανάλογα με τη χρήση ή μη χρήση τοπικών **buffers** (σε κάθε κόμβο) για τη συγκράτηση δεδομένων και αρκετές βελτιωτικές πολιτικές προκειμένου αποφεύγεται το **blocking** ανάμεσα στα πακέτα. Αποφασίζοντας να προχωρήσουμε με μια **buffering** πολιτική και λαμβάνοντας υπόψη την δημιουργία **blocking** ανάμεσα στα πακέτα, επιλέγουμε την υλοποίηση ελέγχου ροής με τη χρήση **εικονικών καναλιών (Virtual**



Εικόνα 2 - Παράδειγμα χρήσης VCs

Channels ~ VCs), τα οποία αποτελούν οντότητες με ξεχωριστούς *buffers* που έχουν τη δυνατότητα να αξιοποιούν από κοινού μια **φυσική ζεύξη (φυσικό κανάλι)** έτσι ώστε, σε περίπτωση παύσης αποστολής δεδομένων τα οποία αφορούν ένα συγκεκριμένο **πακέτο**, να μπορεί να εκμεταλλευτεί τη φυσική ζεύξη ένα **εικονικό κανάλι** του οποίου τα δεδομένα μπορούν να προχωρήσουν στο δίκτυο. Πρακτικά, η πολιτική ροής ελέγχου με τη χρήση **εικονικών καναλιών**, αποτελεί μια πολιτική σε επίπεδο **flit** (καθώς οι *buffers* μπορεί να έχουν μέγεθος μικρότερο από ένα πακέτο, δηλαδή μερικών μόνο *flit*) και δημιουργήθηκε ακριβώς για να αντιμετωπιστεί η **αεργία** των φυσικών ζεύξεων που οφείλεται σε *blocking* ενός πακέτου από ένα άλλο που χρησιμοποιεί την ίδια ζεύξη αλλά δεν σημειώνει προσωρινά πρόοδο. Ωστόσο έως τώρα δεν διευκρινίστηκε με ποιον τρόπο θα είναι σε θέση να γνωρίζει ένας **κόμβος αποστολής** ότι ένας **κόμβος λήψης** θα διαθέτει ελεύθερο χώρο στους τοπικούς του *buffer* προκειμένου να υποδεχθεί τα απεσταλμένα *flits*. Για αυτόν τον λόγο χρειαζόμαστε έναν **μηχανισμό** ή μια πολιτική **προς-τα-πίσω πίεσης (backpressure mechanism)**. Ανάμεσα σε πολλούς που χρησιμοποιούνται σε **buffered flow control**, επιλέγουμε τον **credit-based backpressure mechanism**. Σύμφωνα με αυτή την πολιτική, κάθε κόμβος αποστολής έχει καταχωρημένο σε *registers* τον αριθμό των ελεύθερων *flits* στους *buffer* υποδοχής των **εικονικών καναλιών** στον κόμβο λήψης. Αν αυτός ο αριθμός είναι μεγαλύτερος του μηδενός, τότε ο **κόμβος αποστολής** δύναται να αποστέλλει *flits* μέχρι ο **μετρητής** αυτός να μηδενιστεί. Ωστόσο, πρέπει με κάποιον τρόπο οι δύο κόμβοι να επικοινωνούν προκειμένου όταν ελευθερώνεται μια θέση *flit* στον *buffer* λήψης, να ανανεώνεται ο μετρητής του προηγούμενου κόμβου. Για αυτό τον λόγο, ο **κόμβος λήψης**, οποτεδήποτε συμβαίνει αυτό, αποστέλλει ένα **credit** (ειδικό *flit* ελέγχου για αυτόν το σκοπό) προς τα πίσω στον **κόμβο αποστολής**, προκειμένου να αυξηθεί κατά 1 η τιμή του μετρητή και να είναι σε θέση να αποστείλει ξανά κάποιο *flit*. Αυτές οι 2 πολιτικές δίνουν ευελιξία στο σχεδιασμό μας, αν και δημιουργούν εν γένει μεγάλη **προς-τα-πίσω σηματοδότηση**, αλλά και θα χρησιμοποιηθούν όπως θα δούμε παρακάτω για την αναγνώριση της **συμφόρησης** του δικτύου με τη χρήση της **προς-τα-πίσω πίεσης**. Τέλος, το μέγεθος των *flits*, των **πακέτων** αλλά και ο αριθμός και το βάθος των *buffers*, όπως και ο αριθμός των **VCs** ανά ζεύξη, μπορούν να **παραμετροποιηθούν** για να έχουμε μια γενική εποπτεία του κυκλώματος.

Συμπερασματικά, επιλέγουμε :

- **Virtual Channel Flow Control**
- **Credit-Based Backpressure Mechanism**

3. Δρομολόγηση (Routing)

Ο τρόπος με τον οποίο θα αποφασίζεται η διαδρομή των flits και άρα των δεδομένων ενός πακέτου μέσω του δικτύου αφορά το κομμάτι της **δρομολόγησης**. Υπάρχει μια πληθώρα επιλογών σχετικά με τις πολιτικές δρομολόγησης, οι οποίες επηρεάζουν την **καθυστερήση (latency)** του δικτύου, αλλά και την **διεκπεραιωτικότητα** του (**throughput** ~ βαθμός χρήσης των ζεύξεων) όπως και μια σειρά από άλλους παράγοντες που με τη σειρά τους επηρεάζουν τη συνολική απόδοση του δικτύου. Λαμβάνοντας υπόψη τα παραπάνω και την προσπάθεια αποφυγής **deadlock** στο δίκτυο, στην υλοποίησή μας, θα ακολουθήσουμε το θεωρητικό πλαίσιο του **Duato**. Πιο συγκεκριμένα, θα εισάγουμε την έννοια των **εικονικών καναλιών διαφυγής (escape VCs)**, και κατόπιν θα χρησιμοποιούμε τα συγκεκριμένα εικονικά κανάλια ως **υποδίκτυο** προς αποφυγή ενός πιθανού **deadlock**. Με λίγα λόγια, μπορούμε να έχουμε, π.χ. δύο κλάσεις καναλιών (1 και 2) με την 1 να περιέχει τα εικονικά κανάλια διαφυγής και την 2 να περιέχει τα κανάλια στα οποία γίνεται μια πιο «ελεύθερη» δρομολόγηση. Δηλαδή, επιλέγουμε μέσω των καναλιών της κλάσης 2 να γίνεται μια **προσαρμοστική ελάχιστη δρομολόγηση (minimal adaptive routing)**, επιλογή σχεδίασης που ευθυγραμμίζεται με τις συνθήκες του **Duato** αλλά δεν αποτελεί απαίτησή του. Η δρομολόγηση αυτή αφενός δεν θα επιτρέπει το **misrouting** (αλλαγή πορείας και προσάυξηση απόστασης από τον προορισμό) κι αφετέρου θα είναι προσαρμοστική λαμβάνοντας υπόψη την διαθεσιμότητα των **downstream VC buffers** στους γειτονικούς κόμβους (μέσω του μηχανισμού των **credits**), άρα και τη συμφόρηση του δικτύου. Αυτού του είδους η δρομολόγηση θα λαμβάνει αποκλειστικά μέρος στα κανάλια κλάσης 2, ενώ οποτεδήποτε η διαθεσιμότητα είναι περιορισμένη, τα flits μπορούν να μεταβαίνουν από την κλάση 2 στην κλάση 1, αλλά όχι αντίστροφα. Από εκείνη τη στιγμή κι έπειτα θα δρομολογούνται αποκλειστικά μέσω της τελευταίας κλάσης μόνο με **XY Routing**, όπου έχει αποδειχτεί πως αποτελεί μια **deadlock-free** δρομολόγηση σε τοπολογία πλέγματος. Επομένως η κλάση 1 θα επιτελεί την απορρόφηση του δικτύου σε μια **deadlock-free** εγγυημένη δρομολόγηση, οποτεδήποτε η συμφόρηση δεν επιτρέπει την ελάχιστη προσαρμοστική δρομολόγηση στην κλάση 2. Όλα αυτά για την καταπολέμηση του **deadlock**, βασιζόμενοι στην γενικότερη θεωρητική θεμελίωση του **πρωτοκόλλου του Duato**. Επίσης, απαιτώντας η δρομολόγησή μας να είναι ελάχιστη, δημιουργούμε μια **livelock-free** δρομολόγηση. Στη θεωρία του **Duato** δεν απαιτείται η μόνιμη παραμονή ενός πακέτου στην κλάση διαφυγής. Ωστόσο, στην παρούσα υλοποίηση επιλέγεται η παραμονή στα **escape VCs** μέχρι την παράδοση του πακέτου. Έτσι απλοποιείται ο σχεδιασμός και διατηρείται χωρίς αμφιβολίες το **deadlock-free** χαρακτηριστικό του **υποδικτύου διαφυγής**. Τέλος, η δρομολόγηση ενός πακέτου δεν βασίζεται σε στατικούς πίνακες διαδρομών, αλλά υπολογίζεται **δυναμικά** και **αλγοριθμικά**

μέσω κάποιου αλγόριθμου δρομολόγησης, ο οποίος αποφασίζει την επόμενη μετάβαση σε κάποιον γειτονικό κόμβο (*hop*).

Συμπερασματικά, επιλέγουμε :

- **Minimal Adaptive Routing** (κλάση 2)
- **XY Routing** μέσω *escape VCs* (κλάση 1)

4. Πολιτικές Δέσμευσης εικονικών καναλιών

Διακρίνουμε τις πολιτικές (*επανα*)δέσμευσης ενός εικονικού καναλιού σε 2 βασικές κατηγορίες : *άμεση* (*aggressive*) και *συντηρητική* (*conservative*). Ας υποθέσουμε ότι ένας *buffer* εικονικού καναλιού μπορεί να περιέχει *flits* από δύο διαφορετικά πακέτα, π.χ. τα A και B. Επίσης, ας υποθέσουμε πως το πακέτο A έχει *flits* στον *upstream input VC buffer* του τρέχοντος κόμβου, όπως και το πακέτο B άλλωστε, αλλά έχει επιτυχώς δεσμεύσει το *downstream input VC* ενός γειτονικού κόμβου κι επομένως ένα μέρος των *flits* του πακέτου A βρίσκεται σε αυτόν τον *buffer*. Η *άμεση επαναδέσμευση* επιτρέπει τη δέσμευση του *downstream input VC* για λογαριασμό του πακέτου B, αμέσως μόλις η ουρά του πακέτου A αφήσει τον *upstream input VC buffer* και μεταβεί στο στάδιο της μεταγωγής. Τότε, αμέσως το πακέτο B μπορεί να δεσμεύσει το εικονικό κανάλι του γειτονικού κόμβου για λογαριασμό του. Βέβαια αυτό μπορεί να προκαλέσει περεταίρω προβλήματα αν το πακέτο A στον επόμενο κόμβο αντιμετωπίσει *blocking*, με συνέπεια να «σταματήσει» η αποστολή και του πακέτου B. Επιπρόσθετα, επειδή το εικονικό κανάλι θα περιέχει και τα 2 πακέτα, αλλά πρακτικά η κατάστασή του θα είναι δεσμευμένη από ένα μόνο πακέτο, θα πρέπει να έχει πραγματοποιήσει σύζευξη με το αντίστοιχο *εικονικό κανάλι εξόδου* (*output VC*), στο οποίο θέλει να μεταβεί το κάθε πακέτο. Επομένως, αυτό απαιτεί διπλασιασμό των διανυσμάτων κατάστασης του κάθε τρέχοντος *input VC*, γεγονός που απαιτεί μεγαλύτερη πολυπλοκότητα στον χειρισμό. Ωστόσο αν επιτρέψουμε τη χρήση ενός *input VC buffer* από ένα μοναδικό πακέτο, τότε η χρονική στιγμή που το επόμενο πακέτο μπορεί να δεσμεύσει ένα *downstream input VC*, είναι μόλις η ουρά π.χ. του πακέτου A αφήσει τον *downstream VC buffer* εξ ολοκλήρου άδειο κι έτοιμο για επαναδέσμευση από το πακέτο B, του οποίου η κεφαλή, εκείνη τη χρονική στιγμή θα βρίσκεται στον *upstream input VC buffer* του τρέχοντος κόμβου. Με αυτόν τον τρόπο, μπορεί να έχουμε μεγαλύτερη καθυστέρηση (*latency*), ωστόσο επιτυγχάνουμε μια πιο ασφαλή δέσμευση με περιθώρια για ευκολότερο χειρισμό. Επομένως στη σχεδιάσή μας θα προχωρήσουμε με :

- **Συντηρητική επαναδέσμευση εικονικού καναλιού**

5. Πρόσθετες Επιλογές

- Χρήση ξεχωριστών φυσικών ζεύξεων ανάμεσα σε flits & credits
- Χρήση ξεχωριστών κυκλικών buffer ανά εικονικό κανάλι
- Βασική αρχιτεκτονική 5×5 Crossbar με Separable Switch Allocator (Speedup = 1) (μετέπειτα βελτίωση)
- Πιθανή χρήση Speculation (βελτίωση κατά την υλοποίηση)
- Round-Robin Arbiters