

2.5D Network on Chip Router

Νίκος Χαραλάμπους

ΤΗΜΜΥ ΑΠΘ

4 Νοεμβρίου 2025

- 1 Συζητήθηκαν Προηγούμενως
- 2 Flit-Reservation Flow Control
- 3 Deadlock και Livelock
- 4 Συμπέρασμα

Συζητήθηκαν Προηγούμενως (Buffered Flow Control)

- Packet-Buffer Flow Control
 - Store-and-forward
 - Cut-through
- Flit-Buffer Flow Control
 - Wormhole
 - Virtual Channel

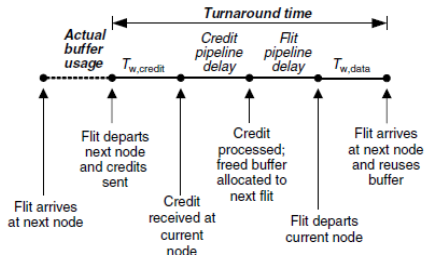
Συζητήθηκαν Προηγούμενως (Backpressure Mechanisms)

- Credit-Based Flow Control
 - Εύκολη υλοποίηση, ωστόσο υψηλό upstream signaling, (για κάθε flit εμπρός απαιτείται ένα credit προς τα πίσω).
- On/Off Flow Control
 - Μειονέκτημα: Χρειάζεται αριθμό buffers F_{on} και F_{off} πριν ξεκινήσουν ή σταματήσουν αντίστοιχα να δέχονται flits (Αεργία ή Υπερχείλιση). Ο αριθμός των buffers αυτών εξαρτάται από τον round trip χρόνο.
- Ack/Nack Flow Control
 - Για κάθε flit αποστέλλεται επιβεβαίωση προς τα πίσω.
Μειονέκτημα: Δεν υπάρχει μηχανισμός γνώσης υπερχείλισης. Αν ένα flit επανασταλεί λόγω αποτυχίας, πρέπει να υπάρχει μηχανισμός διάταξης των flit σε σειρά.

Flit-Reservation Flow Control

Σκεπτικό

Αυτή η πρακτική προτάθηκε για να μειώσει κατά το μέγιστο δυνατό τον χρόνο αεργίας ενός buffer λόγω του κύκλου αποστολής flits και της λήψης credits (turnaround time).

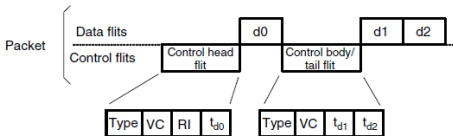


Σχήμα: Buffer Utilization

Flit-Reservation Flow Control

Σκεπτικό

Ως συνέπεια τα flits χωρίζονται σε control και data flits. Τα πρώτα έχουν ως σκοπό την προκράτηση και δέσμευση των πόρων στον κόμβο άφιξης πριν από την διέλευση των τελευταίων.



Σχήμα: Reservation Flits

Flit-Reservation Flow Control

Σκεπτικό

Ο χρονοπρογραμματισμός αποστολής και λήψης αποθηκεύεται σε output και input tables, τα οποία ανανεώνονται συνεχώς ανάλογα με τη διαθεσιμότητα των buffers και τη συμφόρηση του δικτύου.

Output channel \ Time		8	9	10	11	12	13	14	15	16	17
East channel	Channel busy			X							
	Free buffers on next node	2	1	1	0	1	2	3	4	4	4

(a)

Output channel \ Time		8	9	10	11	12	13	14	15	16	17
East channel	Channel busy			X		X					
	Free buffers on next node	2	1	1	0	0	1	2	3	3	3

(b)

Σχήμα: Output Scheduling

Συνοπτικά

Αποτελεί αποτελεσματική πρακτική έναντι της καθυστέρησης ενός pipelined router, καθώς όλα προγραμματίζονται εκ των προτέρων και τα credits λαμβάνονται σε προηγούμενο χρόνο (μόλις προγραμματιστεί η αναχώρηση των flits).

Πλεονεκτήματα

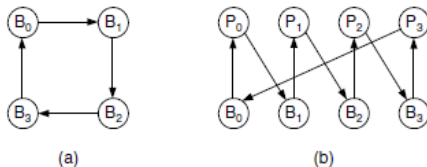
Μειώνει κατά το μέγιστο δυνατό την καθυστέρηση λόγω αναμονής flits και credits άρα και την αεργία των buffers.

Μειονεκτήματα

Χρειάζεται μνήμη για την αποθήκευση και διασύνδεση με tables. Είναι αναγκαία η έξτρα λογική για την ανανέωση και τον χρονοπρογραμματισμό δεδομένων. Αν υπάρξει δέσμευση πόρων, δεν διατίθεται μηχανισμός ακύρωσης ή τροποποίησης της κράτησης σε περίπτωση που βρεθεί αποδοτικότερη κατανομή.

Ορισμός

Είναι το φαινόμενο που προκύπτει όταν ομάδα πακέτων ή flits δεν μπορεί να μετακινηθεί στο δίκτυο, καθώς περιμένει άλλα πακέτα ή flits να απελευθερώσουν πόρους (κανάλια ή buffers), σχηματίζοντας κύκλο.

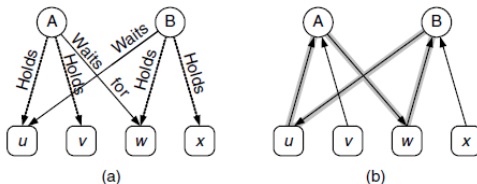


Σχήμα: Γράφοι εξάρτησης κι αναμονής για packet-buffer flow control

Εξάρτηση (Dependence)

Ορισμός

Οποτεδήποτε είναι πιθανό ένα πακέτο ή flit το οποίο δεσμεύει τον πόρο i να περιμένει τον πόρο $i+1$, τότε λέμε ότι υπάρχει εξάρτηση από τον πρώτο στον δεύτερο. Ένας κύκλος στον γράφο εξάρτησης είναι αναγκαία αλλά όχι ικανή συνθήκη για να προκύψει deadlock. Ωστόσο είναι κοινή πρακτική να αφαιρούνται οι κύκλοι στους γράφους εξάρτησης ώστε να είναι αδύνατο να εμφανιστεί.



Σχήμα: Wait-for and Hold relationship graphs

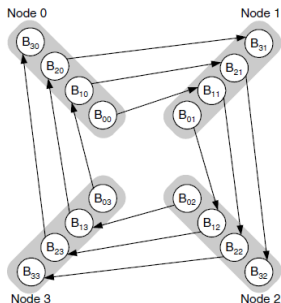
Κατάταξη Πόρων

Όπως είπαμε, το **deadlock** μπορεί να αποφευχθεί εξαφανίζοντας τους κύκλους στους γράφους εξάρτησης. Αυτό μπορεί να επιτευχθεί εφαρμόζοντας αρίθμηση στους πόρους με την προϋπόθεση ότι οι πόροι θα δεσμεύονται με αύξουσα σειρά. Έτσι είναι αδύνατο να προκύψει κύκλος, καθώς αυτό θα σήμαινε ότι τουλάχιστον έναν πακέτο αναμένει πόρο χαμηλότερης αρίθμησης ενώ παράλληλα διατηρεί πόρο υψηλότερης, το οποίο **εξ ορισμού απαγορεύεται**. Παρόλο που όλες οι τεχνικές αποφυγής χρησιμοποιούν κατάταξη πόρων, εκείνες διαφέρουν στο **πώς/αν επηρεάζουν τη δρομολόγηση (routing)**.

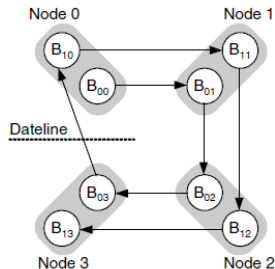
Κλάσεις Πόρων (Resource Classes)

Σκεπτικό

Μια κοινή πρακτική είναι να διαιρέσουμε τους πόρους σε κλάσεις με αρίθμηση προτεραιότητας.



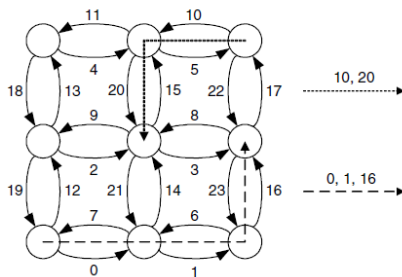
Σχήμα: Distance buffer classes



Σχήμα: Dateline buffer classes

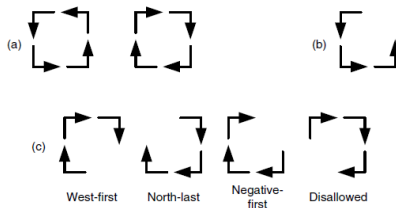
Περιορισμός Δρομολόγησης

Μια εναλλακτική προκειμένου να μειώσουμε το πλήθος των buffers που χρειάζονται για την αποφυγή deadlock, είναι να **περιορίσουμε** τη συνάρτηση δρομολόγησης με τέτοιο τρόπο ώστε να προκύπτει **μη-κυκλικός γράφος εξάρτησης** ανάμεσα στους πόρους.



Σχήμα: Dimension Order Routing σε τοπολογίες πλέγματος

Παραδείγματα Περιορισμένης Δρομολόγησης



Σχήμα: Μοντέλο στροφής για 2D τοπολογίες πλέγματος

Μειονεκτήματα

Περιορίζοντας τη συνάρτηση δρομολόγησης, μειώνουμε την ποικιλία μονοπατιών του δικτύου, γεγονός που επηρεάζει την απόδοση και την ανοχή σφαλμάτων στο δίκτυο. Στην περίπτωση του XY Routing η ποικιλία μηδενίζεται. Οι τεχνικές αυτές δεν μπορούν να εξαφανίσουν τους κύκλους σε τοπολογίες όπως ο τόρος (torus).

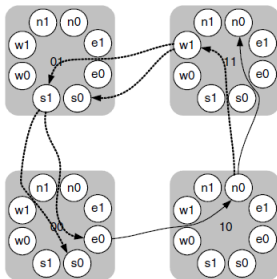
Πρόταση

Είναι δυνατή η ύπαρξη κύκλων στους γράφους εξάρτησης πόρων των αλγορίθμων προσαρμοστικής δρομολόγησης **χωρίς** ωστόσο να προκύπτει deadlock.

Σκεπτικό

Η βασική ιδέα πίσω από την παραπάνω πρόταση είναι η παροχή **μονοπατιών διαφυγής** για κάθε πακέτο σε πιθανή δημιουργία κύκλου. Υπό την προϋπόθεση ότι τα μονοπάτια διαφυγής είναι **deadlock-free** τα πακέτα μπορούν να κινηθούν ελεύθερα ακόμα κι όταν δημιουργούνται κυκλικές εξαρτήσεις ανάμεσα στα κανάλια.

Παράδειγμα Υλοποίησης Μονοπατιών Διαφυγής



Σχήμα: Κυκλική εξάρτηση εικονικών καναλιών με μονοπάτια διαφυγής

Παράδειγμα Δρομολόγησης

Χρησιμοποιώντας, για παράδειγμα, τους παρακάτω κανόνες δρομολόγησης δεν θα προκύψει deadlock ακόμα κι αν υπάρχει κύκλος στον γράφο εξάρτησης των virtual channels.

Παράδειγμα Υλοποίησης Μονοπατιών Διαφυγής

- ❶ Κάθε δρομολόγηση είναι ελάχιστη.
- ❷ Ένα πακέτο που βρίσκεται σε τρέχον εικονικό κανάλι κλάσης 1, μπορεί να δρομολογηθεί σε οποιοδήποτε άλλο εικονικό κανάλι διαθέσιμο στον επόμενο κόμβο.
- ❸ Ένα πακέτο που βρίσκεται σε τρέχον εικονικό κανάλι κλάσης 0, μπορεί να δρομολογηθεί σε οποιοδήποτε κλάσης 1 στον επόμενο κόμβο.
- ❹ Ένα πακέτο που βρίσκεται σε τρέχον εικονικό κανάλι κλάσης 0, μπορεί να δρομολογηθεί μόνο με XY Routing σε κάποιο άλλο κλάσης 0 στον επόμενο κόμβο.

Ανάλυση

Με λίγα λόγια, η δρομολόγηση δεν περιορίζεται όσο το εικονικό κανάλι αναχώρησης ή άφιξης είναι **κλάσης 1**. Ωστόσο όταν **και τα δύο** είναι **κλάσης 0** η δρομολόγηση πρέπει να γίνει με **XY Routing**. Πρακτικά, σε αυτό το παράδειγμα είμαστε σίγουροι πως υπάρχει μια **μη-κυκλική** επιλογή για κάθε πακέτο σε κάποιον κύκλο, επιτρέποντάς του πάντα να μεταβεί σε **dimension order routing** μέσω των εικονικών καναλιών κλάσης 0. **Χωρίς τον 1ο κανόνα**, παύει το δίκτυο να είναι **deadlock-free**, καθώς δημιουργούνται **έμμεσες εξαρτήσεις** κι έτσι τα κανάλια διαφυγής δεν είναι σε θέση να απορροφήσουν την κίνηση του δικτύου.

Εισαγωγή

Υποθέτοντας πως έχουμε μια **συνάρτηση δρομολόγησης** R μπορούμε να θεωρήσουμε μια **υποσυσχέτιση** δρομολόγησης R_1 τέτοια ώστε $R_1 \subseteq R$ πάνω σε ένα υποσύνολο των καναλιών $C_1 \subseteq C$. Έτσι η συμπληρωματική υποσυσχέτιση είναι $R_C = R - R_1$. Με $R(c, d)$ συμβολίζουμε ως ορίσματα το τρέχον κανάλι και τον προορισμό του πακέτου.

Παράδειγμα

Στο παράδειγμά μας η R αποτελεί το σύνολο των κανόνων 1-4 και C είναι το σύνολο των εικονικών καναλιών. Η R_1 είναι ο **κανόνας 4** και C_1 είναι το **σύνολο των καναλιών διαφυγής κλάσης 0**. Η R_C είναι το **σύνολο κανόνων 1-3**.

Άμεσες και Έμμεσες εξαρτήσεις

Άμεση Εξάρτηση

Αν υπάρχει κάποιος κόμβος x , τέτοιος ώστε το κανάλι ενός κόμβου j , $c_j \in R(c_i, x)$, δηλαδή το c_j να βρίσκεται στη διαδρομή προς τον κόμβο x , τότε υπάρχει μια (**άμεση**) εξάρτηση από το c_i στο c_j .

Έμμεση Εξάρτηση

Μια **έμμεση** εξάρτηση δημιουργείται επειδή σε **flit-buffer flow control** τα πακέτα επιτρέπεται να καταλάβουν έναν αυθαίρετο αριθμό virtual channels ταυτόχρονα. Αναλυτικότερα όταν ένα μονοπάτι προς τον κόμβο x χρησιμοποιεί ένα κανάλι $c_i \in C_1$ ακολουθούμενο από έναν αριθμό καναλιών μέσω της R_C και στη συνέχεια διαμέσου ενός καναλιού c_j της C_1 .

Packet-buffer Flow Control

Δεν δημιουργούνται **έμμεσες** εξαρτήσεις για **packet-buffer flow control**.

Πρωτόκολλο του Duato

Το πρωτόκολλο του Duato δίνει το θεωρητικό υπόβαθρο για τη δημιουργία deadlock-free συναρτήσεων δρομολόγησης τόσο για wormhole όσο και store-and-forward δίκτυα. **Αποτελείται από 3 βήματα.**

- 1 Δημιουργούμε ένα **δίκτυο διαφυγής** μέσω μιας συνάρτησης δρομολόγησης ώστε να είναι deadlock-free.
- 2 Δημιουργούμε έναν νέο εικονικό πόρο (κλάση) για κάθε πρωταρχικό φυσικό πόρο στο δίκτυο. Οι πρωταρχικοί πόροι χρησιμοποιούν τη συνάρτηση από το **Βήμα 1**, δηλ. την R_1 ενώ οι νέοι χρησιμοποιούν την R_C .
- 3 **Δεν υπάρχει περιορισμός** στην R_C για packet-buffer flow control. Για flit-buffer flow control η R_C σχεδιάζεται έτσι ώστε να μην υπάρχουν κυκλικές έμμεσες εξαρτήσεις.

Μηχανισμοί Επαναφοράς από Deadlock

Μηχανισμοί Επαναφοράς

Χωρίζονται σε μηχανισμούς **οπισθοδρομικής** και **προοδευτικής** επαναφοράς.

Οπισθοδρομική Επαναφορά

Όταν ένα πακέτο αποστέλλεται, ο κόμβος αποστολής ξεκινάει ένα **timeout** μέτρημα το οποίο επαναλαμβάνεται μόλις κάθε νέο flit διοχετεύεται στο δίκτυο. Όταν σταματήσει το μέτρημα πριν την διοχέτευση του τελευταίου flit ολόκληρο το πακέτο αφαιρείται από το δίκτυο κι επαναποστέλλεται.

Προοδευτική Επαναφορά

Υπάρχουν κοινή πόροι διαφυγής σε κάθε κόμβο του δικτύου. Μόλις δημιουργηθεί η υποψία για deadlock ο κοινός πόρος διαφυγής χρησιμοποιείται ως υποδοχή για τα πακέτα του κύκλου. Η δρομολόγηση που χρησιμοποιεί τον πόρο διαφυγής σχεδιάζεται ώστε να είναι

Ευχαριστώ!
Ερωτήσεις