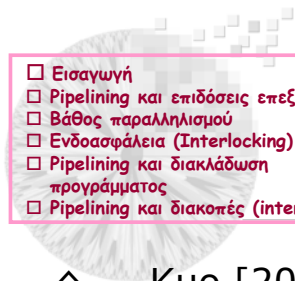


Παραλληλισμός Εντολών (Pipelining)

Τμήμα Επιστήμη και Τεχνολογίας
Τηλεπικοινωνιών

Πανεπιστήμιο Πελοποννήσου



- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Βιβλιογραφία Ενότητας



- ◇ Κuo [2005]: *Chapter 3: Section 3.4, Chapter 4: Sections 4.3-4.4*
- ◇ Lapsley [2002]: *Chapter 9*
- ◇ *TMS320C6000 CPU and Instruction Set Reference Guide*
- ◇ Σημειώσεις διδάσκοντα

★ Εισαγωγή

- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Εισαγωγή



- ◇ Ο παραλληλισμός εντολών γνωστός ως pipelining είναι ένα από τα βασικά χαρακτηριστικά των επεξεργαστών Ψ.Ε.Σ με τα οποία επιτυγχάνουν υψηλές επιδόσεις και εκτέλεση εντολών σε πραγματικό χρόνο. Τα άλλα βασικά χαρακτηριστικά είναι:
 - ◇ Αρχιτεκτονική Harvard της μνήμης
 - ◇ Υλοποίηση πολλαπλασιαστή σε υλικό
 - ◇ Υλοποίηση επαναληπτικών δομών σε υλικό (hardware looping)
- ◇ Ο παραλληλισμός εντολών τεμαχίζει τις εντολές σε διάφορα στάδια εκτέλεσης τα οποία εκτελούνται από διαφορετικά τμήματα του υλικού.
 - ◇ Με τον τρόπο αυτό πολλές εντολές μπορούν να εκτελούνται ταυτόχρονα εφόσον βρίσκονται σε διαφορετικά στάδια εκτέλεσης (και επομένως χρησιμοποιούν διαφορετικό υλικό)
 - ◇ Με τον τρόπο αυτό δεν υπάρχει μεγάλο ποσοστό χρόνου στο οποίο κάποιο τμήμα του υλικού παραμένει ανενεργό
- ◇ Το κόστος του pipelining είναι η δυσκολία προγραμματισμού:
 - ◇ Πολλές φορές αποδοτικός προγραμματισμός μπορεί να επιτευχθεί μόνο σε ASSEMBLY

☑ Εισαγωγή

- ★ Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Pipelining και επιδόσεις επεξεργαστή



- ◇ Για την επίδειξη της βελτίωσης στην ταχύτητα εκτέλεσης ενός προγράμματος που επιτυγχάνεται με τον παραλληλισμό εκτέλεσης εντολών θεωρούμε ότι η εκτέλεση κάθε μιας από τις εντολές περιλαμβάνει τα πιο κάτω στάδια:
 - ◇ **Φάση ανάκλησης εντολής από τη μνήμη (fetch – προσκόμιση εντολής).**
 - ◇ Στη φάση αυτή έχουμε μεταφορά της εντολής, της οποίας η διεύθυνση βρίσκεται στον καταχωρητή προγράμματος (program register), στον καταχωρητή εκτέλεσης εντολής (instruction register)
 - ◇ **Αποκωδικοποίηση εντολής (decode)**
 - ◇ Στη φάση αυτή έχουμε αναγνώριση του τι υποτίθεται ότι πρέπει να κάνει η εντολή και επομένως αναγνώριση του υλικού που εμπλέκεται στην εκτέλεση της εντολής (π.χ. πολλαπλασιαστής, ολισθητής, ALU κλπ)
 - ◇ **Ανάγνωση από ή εγγραφή στη μνήμη των ορισμάτων (read/write) της εντολής**
 - ◇ Τα ορίσματα της εντολής μεταφέρονται στις μονάδες εκτέλεσης της εντολής ή εγγράφονται στη μνήμη
 - ◇ **Εκτέλεση του ALU ή MAC τμήματος της εντολής (execute)**

Εισαγωγή

- ★ Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Pipelining και επιδόσεις επεξεργαστή (II)



- ◇ Στο επόμενο σχήμα παρουσιάζεται η εκτέλεση εντολών χωρίς pipelining. Έστω I1, I2 οι εντολές 1 και 2 αντίστοιχα. Αν κάθε κύκλος ρολογιού εκτελείται σε 10 ns τότε για την εκτέλεση κάθε μιας από τις εντολές χρειάζονται 40 ns.
- ◇ Το υλικό του επεξεργαστή που σχετίζεται με την εκτέλεση κάθε ενός από τα στάδια της εντολής παραμένει ανενεργό για το 75% του χρόνου.

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)							
	1	2	3	4	5	6	7	8
Προσκόμιση Εντολής	I1				I2			
Αποκωδικοποίηση Εντολής		I1				I2		
Ανάγνωση / Εγγραφή Δεδομένων			I1				I2	
Εκτέλεση Εντολής				I1				I2

 Εισαγωγή

- ★ Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Pipelining και επιδόσεις επεξεργαστή (II)



- ◇ Στο επόμενο σχήμα φαίνεται η εκτέλεση εντολών με παραλληλισμό (pipelining).
- ◇ Στον τέταρτο κύκλο ρολογιού η εντολή I4 βρίσκεται στη φάση ανάκλησης (fetch), η εντολή I3 βρίσκεται στο στάδιο αποκωδικοποίησης (decode), η εντολή I2 βρίσκεται στη φάση ανάγνωσης ή εγγραφής ορισμάτων (data read/write) ενώ η εντολή I1 βρίσκεται σε φάση εκτέλεσης από την ALU ή την μονάδα MAC.
- ◇ Σε σχέση με το προηγούμενο παράδειγμα ο μέσος χρόνος εκτέλεσης μιας εντολής είναι 10 ns (περίπου τέσσερις φορές μικρότερος).
- ◇ Το συγκεκριμένο παράδειγμα προσομοιάζει τη λογική pipelining της σειράς επεξεργαστών TMS320C3x

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)							
	1	2	3	4	5	6	7	8
Προσκόμιση Εντολής	I1	I2	I3	I4	I5	I6	I7	I8
Αποκωδικοποίηση Εντολής		I1	I2	I3	I4	I5	I6	I7
Ανάγνωση / Εγγραφή Δεδομένων			I1	I2	I3	I4	I5	I6
Εκτέλεση Εντολής				I1	I2	I3	I4	I5

- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- ★ Βάθος παραλληλισμού
- Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Βάθος παραλληλισμού (Pipelining depth)



- ◇ Από το προηγούμενο παράδειγμα είναι φανερό ότι όσο μεγαλύτερο βάθος παραλληλισμού έχουμε τόσο μικρότερος είναι ο χρόνος που απαιτείται για την εκτέλεση μιας εντολής.
- ◇ Για να μπορεί να αυξηθεί το βάθος παραλληλισμού χρειάζεται μεγαλύτερη ανεξαρτησία των επιμέρους μονάδων υλικού

DSP processor	Pipeline Phases
TMS320C2000	Fetch-Decode-Read/Write-eXecute (4 επίπεδα)
TMS320C3x	Fetch-Decode-Read/Write-eXecute (4 επίπεδα)
TMS320C54x	PreFetch-Fetch-Decode-DataAddressGeneration-Read/Write-eXecute (6 επίπεδα)
TMS320C55x	PF1-PF2-F-PreDecode (4 επίπεδα προσκόμισης εντολής) D-ADress-ACcess1-AC2-R-X-WriteResults (7 επίπεδα εκτέλεσης)
TMS320C6000	PG-PS-PW-PR-DP-DC-E1-E2-E3-E4-E5 (11 επίπεδα)

Όπως φαίνεται στο διπλανό σχήμα, για την οικογένεια επεξεργαστών της Texas Instruments, προοδευτικά το βάθος παραλληλισμού αυξάνει:

Από 4 επίπεδα στους παλαιότερους επεξεργαστές σε 11 επίπεδα στους νεότερους επεξεργαστές.

Πρέπει να σημειωθεί πάντως ότι δεν μπορούν να οριστούν για όλες τις εντολές 11 στάδια εκτέλεσης (κυρίως τα στάδια E1-E5)

- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- ★ Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Ενδοασφάλεια (Interlocking)



- ◇ Στα προηγούμενα παραδείγματα είχαμε πλήρη παραλληλισμό. Όλες οι μονάδες υλικού βρίσκονταν σε λειτουργία εκτελώντας διαφορετικές εντολές.
- ◇ Για να μπορεί να επιτευχθεί αυτό θα πρέπει διαφορετικές εντολές να μην χρειάζονται την ίδια χρονική στιγμή το ίδιο υλικό. Αυτό δεν είναι πάντοτε εφικτό.
- ◇ Στο επόμενο σχήμα υποδεικνύεται μια περίπτωση φιλονικίας (instruction contention) ανάμεσα σε δύο εντολές για πρόσβαση στη μνήμη. Συγκεκριμένα η εντολή I2 χρειάζεται δύο κύκλους ρολογιού για να διαβάσει τα δύο ορίσματα της από τη μνήμη. Ενώ κατά τη διάρκεια του πρώτου (κύκλος 4) από τους δύο κύκλους η εντολή I3 βρίσκεται στη φάση αποκωδικοποίησης στον επόμενο κύκλο (κύκλος 5) ζητά και αυτή πρόσβαση στη μνήμη.

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)							
	1	2	3	4	5	6	7	8
Προσκόμιση Εντολής	I1	I2	I3	I4	I5	I6	I7	I8
Αποκωδικοποίηση Εντολής		I1	I2	I3	I4	I5	I6	I7
Ανάγνωση / Εγγραφή Δεδομένων			I1	I2	I2/I3	I4	I5	I6
Εκτέλεση Εντολής				I1	I2	I3	I4	I5

Επίλυση φιλονικιών επιτυγχάνεται με την τεχνική της ενδοασφάλειας (interlocking)

Οι εντολές που βρίσκονται σε εκτέλεση κλειδώνονται στη φάση που βρίσκονται έως ότου επιλυθεί η φιλονικία

- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- ★ Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Ενδοασφάλεια (II)



- ◇ Στο επόμενο σχήμα φαίνεται η κατάσταση εκτέλεσης των εντολών για την επίλυση της φιλονικίας ανάμεσα στις εντολές I2 και I3.
 - ◇ Η εντολή I4 συγκρατείται στη φάση ανάκλησης (π.χ. παραμένει ίδια η διεύθυνση μνήμης στο μετρητή προγράμματος),
 - ◇ η εντολή I3 παραμένει σε κατάσταση αποκωδικοποίησης,
 - ◇ η εντολή I2 πραγματοποιεί τη δεύτερη ανάγνωση από τη μνήμη,
 - ◇ την ίδια χρονική στιγμή εκτελείται η εικονική εντολή NOP από την ALU.

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)							
	1	2	3	4	5	6	7	8
Προσκόμιση Εντολής	I1	I2	I3	I4	I4	I5	I6	I7
Αποκωδικοποίηση Εντολής		I1	I2	I3	I3	I4	I5	I6
Ανάγνωση / Εγγραφή Δεδομένων			I1	I2	I2	I3	I4	I5
Εκτέλεση Εντολής				I1	NOP	I2	I3	I4

Στο συγκεκριμένο παράδειγμα αν η εντολή I3 δεν χρειαζόταν πρόσβαση στη μνήμη τότε δεν θα είχαμε πρόβλημα φιλονικίας. Επομένως η σειρά των εντολών κατά τη συγγραφή του προγράμματος έχει μεγάλη σημασία για την αποφυγή φιλονικίας

- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- ★ Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Παράδειγμα



- ◇ Υπάρχουν και άλλες περιπτώσεις στις οποίες μπορεί να έχουμε φιλονικία, π.χ.:
 - ◇ Άμεση πρόσβαση σε δεδομένα εντός εντολών η οποία μπορεί να απαιτεί δεύτερη ανάκληση από τη μνήμη προγράμματος
 - ◇ Σε ορισμένες περιπτώσεις η φιλονικία μπορεί να μην είναι αντιληπτή αλλά η διαδοχή εντολών μπορεί να μην επιφέρει τα αναμενόμενα αποτελέσματα όπως φαίνονται στο επόμενο παράδειγμα (από τον επεξεργαστή Motorola DSP5600)
 - ◇ Οι επόμενες εντολές στοχεύουν στη μετακίνηση των περιεχομένων της θέσης μνήμης 1234h στον καταχωρητή X0.

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)			
	1	2	3	4
Program Memory	Fetch MOVE #1234,R0	Fetch MOVE X:(R0),X0	I3	I4
Decode Unit		Decode MOVE #1234,R0	Decode MOVE X:(R0),X0	I3
Address Generation Unit			Generate Data Address for X:(R0), access (i.e., 5678)	R0 now equals 1234
Execution Unit			Execute MOVE #1234,R0	Execute MOVE X:(R0),X0
X Data Memory				Read data from address 5678

MOVE #1234,R0
MOVE X:(R0),X0

Η πρώτη εντολή μεταφέρει τα άμεσα δεδομένα (διεύθυνση 1234h) στον καταχωρητή R0.

Η δεύτερη εντολή το περιεχόμενο της θέσης μνήμης που υποδεικνύεται από τον καταχωρητή R0 στον καταχωρητή X0.

- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- ★ Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Παράδειγμα (συν.)



- ◇ Η διαδικασία pipelining για τις δύο εντολές φαίνεται στο σχήμα που ακολουθεί:
 - ◇ Ανάκληση της πρώτης εντολής (1^{ος} κύκλος ρολογιού)
 - ◇ Ανάκληση της δεύτερης εντολής και αποκωδικοποίηση της πρώτης η οποία περιλαμβάνει και πρόσβαση στα άμεσα δεδομένα (2^{ος} κύκλος ρολογιού)
 - ◇ Αποκωδικοποίηση δεύτερης εντολής και ταυτόχρονη δημιουργία της διεύθυνσης πρόσβασης στα δεδομένα με τη βοήθεια του τρέχοντος περιεχομένου του καταχωρητή R0, π.χ. 5678 (3^{ος} κύκλος ρολογιού)
 - ◇ Εκτέλεση πρώτης και δεύτερης εντολής (4^{ος} κύκλος ρολογιού)

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)			
	1	2	3	4
Program Memory	Fetch MOVE #1234,R0	Fetch MOVE X:(R0),X0	I3	I4
Decode Unit		Decode MOVE #1234,R0	Decode MOVE X:(R0),X0	I3
Address Generation Unit			Generate Data Address for X:(R0), access (i.e.,5678)	R0 now equals 1234
Execution Unit			Execute MOVE #1234,R0	Execute MOVE X:(R0),X0
X Data Memory				Read data from address 5678

Είναι φανερό ότι κατά την εκτέλεση της παραπάνω διαδοχής εντολών τελικά στον καταχωρητή X0 μεταφέρεται το περιεχόμενο της θέσης μνήμης που υποδεικνύεται από την τρέχουσα τιμή του καταχωρητή R0. Το παράδειγμα αυτό μας δίνει μια περίπτωση pipeline hazard δηλαδή προβλήματος εξαιτίας της διαδικασίας pipelining.

- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- ★ Ενδοασφάλεια (Interlocking)
- Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Παράδειγμα (II)



- ◇ Η Texas Instruments στους επεξεργαστές της χρησιμοποιεί interlocking για την προστασία από το προηγούμενο πρόβλημα.
 - ◇ Συγκεκριμένα εφόσον αναγνωρισθεί ότι υπάρχει προσπάθεια εγγραφής στους καταχωρητές διευθύνσεων (AR_i, i = 1, ..., 7) τότε όλες οι εντολές οι οποίες χρησιμοποιούν έμμεση διευθυνσιοδότηση κλειδώνονται στο στάδιο στο οποίο βρίσκονται μέχρι την ολοκλήρωση της διαδικασίας εγγραφής στους βοηθητικούς καταχωρητές
 - ◇ Η εντολή LDI (load immediate) φορτώνει στον ενεργό βοηθητικό καταχωρητή μια τιμή που δηλώνεται στο όρισμα της εντολής
 - ◇ Η εντολή MPYF (floating point multiplication) χρησιμοποιεί τη διεύθυνση μνήμης που υποδεικνύεται από τον ενεργό βοηθητικό καταχωρητή για τα δεδομένα ενός από τα ορίσματά της.

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)							
	1	2	3	4	5	6	7	8
Προσκόμιση Εντολής	LDI	MPYF	I3	I3	I3	I4	I5	I6
Αποκωδικοποίηση Εντολής		LDI	MPYF	MPYF	MPYF	I3	I4	I5
Ανάγνωση / Εγγραφή Δεδομένων			LDI	---	---	MPYF	I3	I4
Εκτέλεση Εντολής				LDI	NOP	NOP	MPYF	I3

Η εντολή MPYF παραμένει στο στάδιο αποκωδικοποίησης μέχρι την ολοκλήρωση της εντολής LDI.

- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- Ενδοασφάλεια (Interlocking)
- * Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Pipelining και διακλάδωση προγράμματος



- ◇ Ειδική μνεία για τη διαδικασία pipelining χρειάζεται στην περίπτωση διακλάδωσης ή εν γένει αλλαγής της ροής του προγράμματος
- ◇ Το σχήμα που ακολουθεί επιδεικνύει τη διαδικασία σε αυτή την περίπτωση:
 - ◇ Στο στάδιο της αποκωδικοποίησης της εντολής BR (branch) γίνεται αντιληπτό ότι έχουμε περίπτωση αλλαγής της ροής του προγράμματος επομένως η εκτέλεση θα συνεχιστεί από το νέο σημείο (εντολή N1) αντί της επόμενης εντολής που ακολουθεί την BR (εντολή I2) η οποία όμως έχει ήδη ανακληθεί από τη μνήμη
- ◇ Ένας τρόπος αντιμετώπισης είναι η εκκένωση της ουράς pipelining που όμως οδηγεί κάθε εντολή BR να εκτελείται σε τέσσερις κύκλους ρολογιού κατά μέσο όρο αντί για ένα που απαιτείται για τις υπόλοιπες εντολές

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)							
	1	2	3	4	5	6	7	8
Προσκόμιση Εντολής	BR	I2	---	---	N1	N2	N3	N4
Αποκωδικοποίηση Εντολής		BR	---	---	---	N1	N2	N3
Ανάγνωση / Εγγραφή Δεδομένων			BR	---	---	---	N1	N2
Εκτέλεση Εντολής				BR	NOP	NOP	NOP	N1

Η ανωτέρω τεχνική χειρισμού διακλαδώσεων προγράμματος ονομάζεται multicycle branch, καθότι η εκτέλεση μιας εντολής διακλάδωσης χρειάζεται τόσους κύκλους ρολογιού να εκτελεστεί όσο το βάθος παραλληλισμού (pipeline depth)

- Εισαγωγή
- Pipelining και επιδόσεις επεξεργαστή
- Βάθος παραλληλισμού
- Ενδοασφάλεια (Interlocking)
- * Pipelining και διακλάδωση προγράμματος
- Pipelining και διακοπές (interrupts)

Διακλάδωση με καθυστέρηση



- ◇ Οι περισσότεροι επεξεργαστές Ψ.Ε.Σ ακολουθούν τη φιλοσοφία multicycle branch. Μερικοί όμως υιοθετούν την τεχνική της καθυστερημένης διακλάδωσης η οποία όμως προσθέτει μεγαλύτερες δυσκολίες στον προγραμματισμό
 - ◇ Όπως φαίνεται στο επόμενο σχήμα παρά την αναγνώριση της εντολής BR στο δεύτερο κύκλο ρολογιού η ουρά pipeline δεν εκκενώνεται αλλά συνεχίζεται η εκτέλεση των εντολών I2, I3, I4 που ακολουθούν την BR.
 - ◇ Μετά την εκτέλεση της εντολής BR η επόμενη εντολή που ανακαλείται από τη μνήμη είναι η N1, δηλαδή η πρώτη εντολή στην οποία παραπέμπει η εντολή BR.

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)							
	1	2	3	4	5	6	7	8
Προσκόμιση Εντολής	BR	I2	I3	I4	N1	N2	N3	N4
Αποκωδικοποίηση Εντολής		BR	I2	I3	I4	N1	N2	N3
Ανάγνωση / Εγγραφή Δεδομένων			BR	I2	I3	I4	N1	N2
Εκτέλεση Εντολής				BR	I2	I3	I4	N1

Συνέπεια της καθυστερημένης διακλάδωσης είναι στο επόμενο τμήμα προγράμματος οι εντολές INST 1, INST 2, INST 3 να εκτελεστούν πριν πραγματικά συμβεί η διακλάδωση
BR new_addr
INST 1
INST 2
INST 3

- ☑ Εισαγωγή
- ☑ Pipelining και επιδόσεις επεξεργαστή
- ☑ Βάθος παραλληλισμού
- ☑ Ενδοασφάλεια (Interlocking)
- ☑ Pipelining και διακλάδωση προγράμματος
- ★ Pipelining και διακοπές (interrupts)

Pipelining και διακοπές (interrupts)



- ◇ Οι διακοπές (interrupts) για εξυπηρέτηση από τον επεξεργαστή αιτημάτων από περιφερειακές συσκευές (όπως η σειριακή θύρα, οι AD converters κλπ) επιφέρουν και αυτές αλλαγές στη ροή του προγράμματος.
- ◇ Μετά την αναγνώριση της διακοπής ο επεξεργαστής:
 - ◇ Αφήνει να ολοκληρωθεί ο τρέχων κύκλος ρολογιού
 - ◇ Εισάγει μια εντολή INTR η οποία οδηγεί τον επεξεργαστή στο κατάλληλο διάνυσμα διακοπής για εξυπηρέτηση της διακοπής
 - ◇ Αφήνει τις εντολές που βρίσκονται στο pipeline σε στάδιο αποκωδικοποίησης και μετά να εκτελεστούν.

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)									
	1	2	3	4	5	6	7	8	9	10
Προσκόμιση Εντολής	I4	I5	I6	---	---	---	V1	V2	V3	V4
Αποκωδικοποίηση Εντολής	I3	I4	I5	INTR	---	---	---	V1	V2	V3
Ανάγνωση / Εγγραφή Δεδομένων	I2	I3	I4	I5	INTR	---	---	---	V1	V2
Εκτέλεση Εντολής	I1	I2	I3	I4	I5	INTR	NOP	NOP	NOP	V1

↑ Interrupt processing begins

Στο παράδειγμα του σχήματος η εντολή INTR εισάγεται μετά την I5 (η οποία ήταν σε φάση αποκωδικοποίησης) κατά την αναγνώριση της διακοπής. Η εξυπηρέτηση της διακοπής αρχίζει στον κύκλο ρολογιού 7 αφού έχει ολοκληρωθεί η εκτέλεση των εντολών I3, I4, I5 και της INTR.

- ☑ Εισαγωγή
- ☑ Pipelining και επιδόσεις επεξεργαστή
- ☑ Βάθος παραλληλισμού
- ☑ Ενδοασφάλεια (Interlocking)
- ☑ Pipelining και διακλάδωση προγράμματος
- ★ Pipelining και διακοπές (interrupts)

Παράδειγμα



- ◇ Μια εναλλακτική υλοποίηση επεξεργασίας διακοπών υιοθετείται από τους επεξεργαστές Motorola. Σε αυτή την περίπτωση αντί της εκτέλεσης της εντολής INTR εισάγεται κατευθείαν η κλήση της υπορουτίνας εξυπηρέτησης της διακοπής (JSR) με αποτέλεσμα κατά τη φάση αποκωδικοποίησης της JSR να προσκομίζεται η πρώτη εντολή (V3) εξυπηρέτησης της διακοπής.
- ◇ Επομένως η εξυπηρέτηση της διακοπής αρχίζει πρακτικά στον 5ο κύκλο ρολογιού.

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)									
	1	2	3	4	5	6	7	8	9	10
Προσκόμιση Εντολής	I3	I4	JSR	---	V3	V4	V5	V6	V7	V8
Αποκωδικοποίηση Εντολής	I2	I3	I4	JSR	---	V3	V4	V5	V6	V7
Εκτέλεση Εντολής	I1	I2	I3	I4	JSR	---	V3	V4	V5	V6

↑ Interrupt processing begins

- ☑ Εισαγωγή
- ☑ Pipelining και επιδόσεις επεξεργαστή
- ☑ Βάθος παραλληλισμού
- ☑ Ενδοασφάλεια (Interlocking)
- ☑ Pipelining και διακλάδωση προγράμματος
- ★ Pipelining και διακοπές (interrupts)

Ταχεία εξυπηρέτηση διακοπής



- ◇ Η εξυπηρέτηση διακοπών αρχίζει με την εκτέλεση εντολών στο διάνυσμα διακοπών. Στο διάνυσμα αυτό αποθηκεύονται το πολύ δύο εντολές για κάθε διακοπή (V1, V2). Αν οι εντολές αυτές επαρκούν για την εξυπηρέτηση της διακοπής τότε έχουμε **‘ταχεία εξυπηρέτηση διακοπής’**
- ◇ Αν χρειάζονται περισσότερες από δύο εντολές για την εξυπηρέτηση της διακοπής τότε η μία εξ’ αυτών πρέπει να είναι υποχρεωτικά μια εντολή κλήσης υπορουτίνας για την εξυπηρέτηση της διακοπής

	Κύκλος Ρολογιού (κύκλος εκτέλεσης εντολής)									
	1	2	3	4	5	6	7	8	9	10
Προσκόμιση Εντολής	I3	I4	V1	V2	I5	I6	I7	I8	I9	I10
Αποκωδικοποίηση Εντολής	I2	I3	I4	V1	V2	I5	I6	I7	I8	I9
Εκτέλεση Εντολής	I1	I2	I3	I4	V1	V2	I5	I6	I7	I8

↑ Interrupt processing begins

Στο διπλανό σχήμα φαίνεται η διαδικασία ‘ταχείας εξυπηρέτησης διακοπής’. Είναι φανερό ότι σε αυτή την περίπτωση δεν έχουμε κάποια συνέπεια στο pipeline καθώς οι δύο εντολές απλά εισάγονται στο pipeline χωρίς αυτό να χρειαστεί να εκκενωθεί