

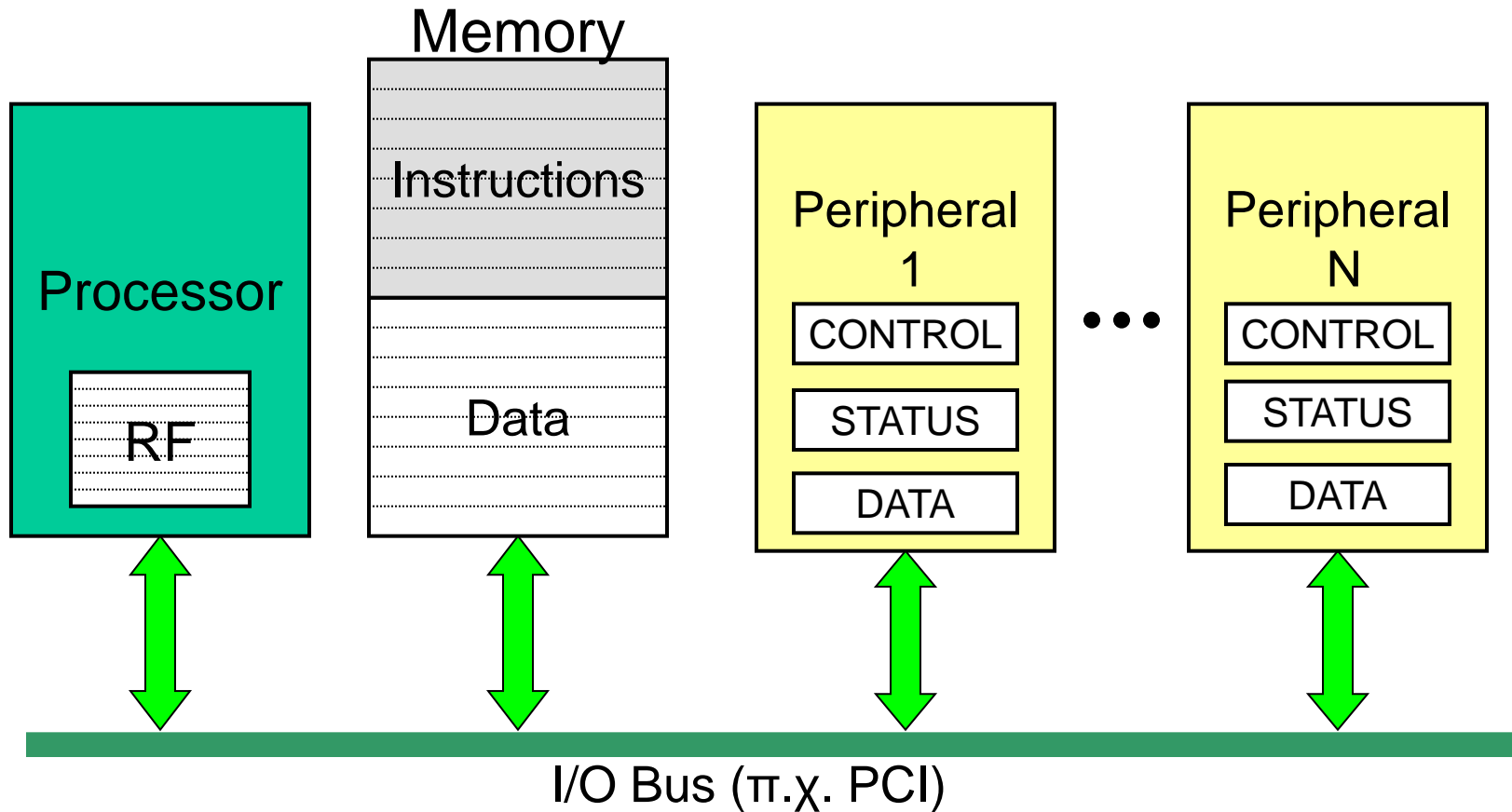
# ΗΡΥ 201– Ψηφιακοί Υπολογιστές

Είσοδος/Εξοδος, Διακοπές, DMA

<http://www.mhl.tuc.gr>

ΓΙΑΝΝΗΣ ΠΑΠΑΕΥΣΤΑΘΙΟΥ

# Απλοποιημένο Μοντέλο Η/Υ με Περιφερειακές Συσκευές



# Απλοϊκό Μοντέλο Συσκευής I/O

- Επικοινωνία με τον επεξεργαστή μέσω τριών καταχωρητών: Κατάστασης, Δεδομένων και Ελέγχου
- Καταχωρητής Κατάστασης (Status) :
  - Για συσκευές εισόδου δείχνει αν τα δεδομένα είναι έτοιμα για ανάγνωση
  - Για συσκευές εξόδου δείχνει αν η συσκευή έχει τελειώσει με τα παλιά και είναι έτοιμη να δεχτεί νέα δεδομένα
- Καταχωρητής Δεδομένων (Data) :
  - Τα δεδομένα για μετάδοση (από η προς τον επεξεργαστή)
- Καταχωρητής Ελέγχου (Control) :
  - Εδώ θέτουμε παραμέτρους της συσκευής και της δίνουμε εντολές που ελέγχουν τον τρόπο λειτουργίας της

# Παράδειγμα Συσκευής: Πληκτρολόγιο

- Καταχωρητής Κατάστασης:
  - Μηδέν αν δεν υπάρχει χαρακτήρας, 1 αν υπάρχει.
- Καταχωρητής Δεδομένων:
  - Ο χαρακτήρας που πατήθηκε (αν υπάρχει)
- Καταχωρητής Ελέγχου:
  - Ο επεξεργαστής γράφει 0 ή 1 στα τελευταία (LS) 3 bits του καταχωρητή για να ανάψει ή να σβήσει τα LED του πληκτρολογίου.
- Πώς διαβάζω μία συμβολοσειρά (string)?
  - Σε μια ανακύκλωση, ελέγχω τον καταχωρητή κατάστασης.
  - Εάν είναι μηδέν, δεν υπάρχει χαρακτήρας, και πρέπει να περιμένω.
  - Εάν είναι 1, τότε διαβάζω τον καταχωρητή δεδομένων.
  - Αν ο καταχωρητής δεδομένων είναι το πλήκτρο ENTER επιστρέφω την μέχρι τώρα συμβολοσειρά.
- Αυτή η διαδικασία ονομάζεται “Polling”

# Πώς Επικοινωνούν Επεξεργαστής και Περιφερειακά;

- Πώς «βλέπει» ο επεξεργαστής τους καταχωρητές των περιφερειακών συσκευών;
  - Χρησιμοποιεί ξεχωριστό δίαυλο επικοινωνίας (*I/O bus*) για τα περιφερειακά από αυτόν που έχει για την μνήμη.
    - Το κάθε περιφερειακό έχει την δική του διεύθυνση στον εξειδικευμένο δίαυλο αυτό.
    - Ειδικές εντολές μεταφέρουν δεδομένα μεταξύ καταχωρητών του επεξεργαστή και των περιφερειακών.
    - Επιπλέον κόστος για καλώδια, εντολές, κλπ.
  - Χρησιμοποιεί το δίαυλο (bus) επικοινωνίας και τις διευθύνσεις που έχει για την μνήμη (*Memory Mapped I/O*)

# Memory Mapped I/O

- Κάποιες από τις διευθύνσεις της μνήμης χρησιμοποιούνται για τις περιφερειακές συσκευές:
  - Ο χώρος διευθύνσεων του MIPS είναι  $2^{32}$ . Ένα μικρό κομμάτι αυτού του χώρου διατίθεται για να «βλέπει» ο επεξεργαστής τους καταχωρητές των περιφερειακών συσκευών.
  - Η ανάγνωση και εγγραφή από/προς τους καταχωρητές των συσκευών και του επεξεργαστή γίνεται με απλές εντολές lw/sw.
  - Απλούστερο, και πιο συνηθισμένο.

# Polling

Η τεχνική Polling βασίζεται στον περιοδικό έλεγχο των συσκευών για να διαπιστωθεί αν υπάρχουν νέα δεδομένα για ανάγνωση, και για το εάν οι συσκευές εξόδου είναι έτοιμες για νέα δεδομένα.

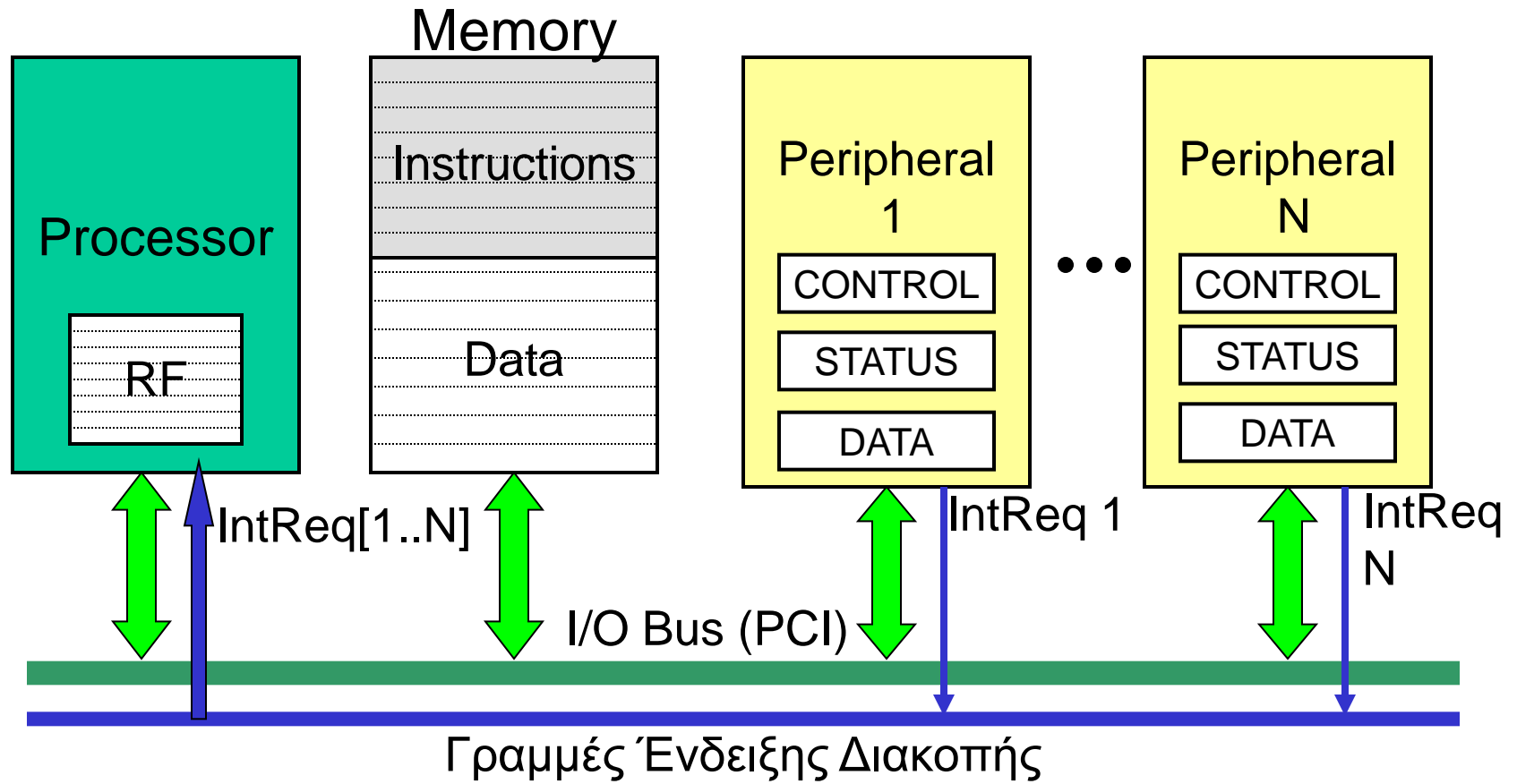
- Κάθε πότε πρέπει να ρωτάει ο επεξεργαστής τις συσκευές αν έχουν καινούργια δεδομένα γι' αυτόν;
  - Εξαρτάται από την ταχύτητα μετάδοσης της συσκευής. Για αργές συσκευές, όπως το πληκτρολόγιο, και μια φορά κάθε εκατοστό του δευτερολέπτου είναι υπέρ-αρκετό. Για γρήγορες συσκευές όμως πρέπει να ρωτάει πιο συχνά!
- Αν ρωτάμε πολύ συχνά, ξοδεύουμε τον χρόνο μας άδικα.
- Αν αργήσουμε να ρωτήσουμε, μπορεί να χαθούν δεδομένα
  - Π.χ. να πανωγραφτούν από τα επόμενα δεδομένα από μια συσκευή εισόδου πριν τα διαβάσουμε)

# Είσοδος/Εξοδος με Διακοπές

- Επειδή οι συσκευές εισόδου ξέρουν ακριβώς πότε έχουν δεδομένα για τον επεξεργαστή, και οι συσκευές εξόδου ξέρουν ακριβώς πότε μπορούν να δεχτούν νέα δεδομένα από τον επεξεργαστή:
  - Μπορούν να ενημερώσουν τον επεξεργαστή! Πώς όμως?
  - Δεν υπάρχει τρόπος να γράψουν κάτι σε καταχωρητές του επεξεργαστή!
- ΛΥΣΗ: Υποστήριξη διακοπών (interrupts) της κανονικής εκτέλεσης του προγράμματος από τις συσκευές.
  - Οι περιφεριακές συσκευές «μιλάνε» στον επεξεργαστή
  - ⇒ Τρόπος απόκτησης της προσοχής του επεξεργαστή!



# Μοντέλο Η/Υ με Υποστήριξη Διακοπών



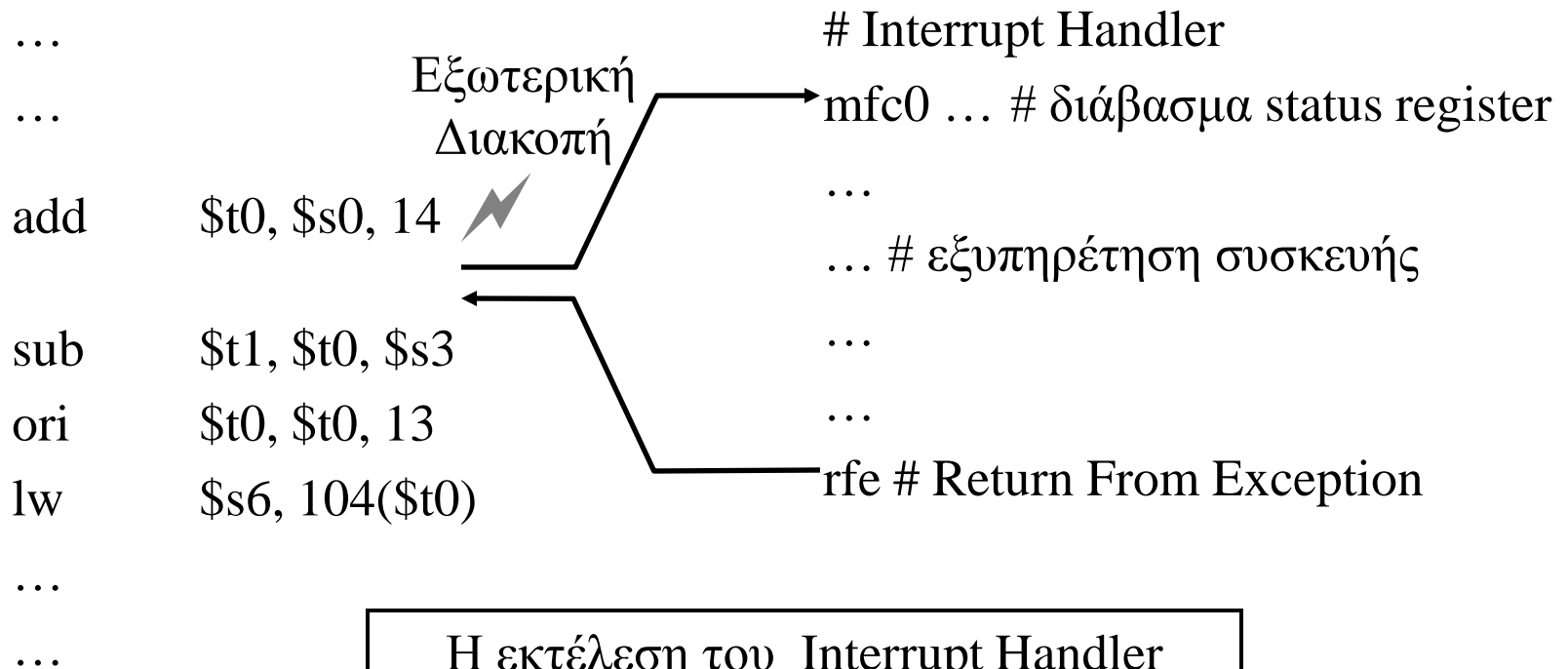
# Υποστήριξη Διακοπών

- Ο επεξεργαστής διαθέτει εισόδους αίτησης διακοπής (Interrupt Request) για ένα αριθμό συσκευών.
- Η κάθε συσκευή ειδοποιεί τον επεξεργαστή ενεργοποιώντας την γραμμή αίτησης που της αντιστοιχεί
- Η αίτηση διακοπής (συνήθως) απενεργοποιείται όταν ο επεξεργαστής διαβάσει τον status register
- Ο επεξεργαστής διακόπτει την κανονική επεξεργασία εντολών του χρήστη, και ξεκινάει να εκτελεί κώδικα (interrupt handler) ο οποίος θα βρει την συσκευή που αιτήθηκε την διακοπή, και θα διαβάσει (γράψει) τα νέα δεδομένα
- Τι γίνεται αν έχω περισσότερες συσκευές από ότι γραμμές;
  - Πολύπλεξη των γραμμών αίτησης διακοπής => μια γραμμή αντιστοιχεί σε παραπάνω από μία συσκευές
  - Χρειάζεται ένας ελεγκτής διακοπών (interrupt controller) ο οποίος θα παρέχει πολλές γραμμές αίτησης προς τις συσκευές
  - Ο interrupt handler πρέπει να βρει ποιά από τις πολυπλεγμένες συσκευές αιτήθηκε την διακοπή

# Διακοπές και Εξαιρέσεις

- Διακοπές και εξαιρέσεις (interrupts & exceptions) μπορούν να δημιουργηθούν από διάφορες πηγές:
  - Λάθη προγράμματος (παράνομες εντολές, παράνομες διευθύνσεις, κ.α.)
  - Αριθμητική: υπερχείλιση, διαίρεση με μηδέν, κ.α.
  - Συσκευές που χρειάζονται την προσοχή του λειτουργικού συστήματος
  - Σφάλματα υλικού
  - κ.α.
- Όλα τα παραπάνω απαιτούν ειδικό κώδικα (πρόγραμμα) για την εξυπηρέτησή τους, και κατόπιν την επιστροφή στην κανονική επεξεργασία

# Παράδειγμα Διακοπής προγράμματος



Η εκτέλεση του Interrupt Handler παρεμβάλλεται μεταξύ της add και της sub, και μετά η εκτέλεση του αρχικού προγράμματος συνεχίζεται κανονικά

# Βασικές Κατηγορίες Διακοπών

- Σύγχρονες/Ασύγχρονες
  - Σύγχρονες είναι οι διακοπές που η εξυπηρέτησή τους πρέπει να είναι άμεση γιατί αλλιώς η εκτέλεση του προγράμματος δεν μπορεί να συνεχίσει. Τέτοιου τύπου διακοπές είναι οι διακοπές εικονικής μνήμης, κ.α.
  - Ασύγχρονες είναι οι διακοπές όταν η εξυπηρέτηση τους μπορεί να αναβληθεί για κάποια (λογικό) χρονικό διάστημα. Οι εξωτερικές διακοπές από περιφερειακές συσκευές είναι συνήθως ασύγχρονες.
- Εσωτερικές/Εξωτερικές
  - Εσωτερικές είναι οι διακοπές που παράγονται στο εσωτερικό του επεξεργαστή (π.χ. Παράνομη διεύθυνση, υπερχείλιση, κ.α.)
  - Εξωτερικές είναι οι διακοπές που έρχονται από τον «έξω» κόσμο (περιφερειακά κλπ.)

# Επίπεδα Προστασίας και Διακοπές

- Για να προστατευθούν όλοι οι χρήστες σε ένα σύστημα, η πρόσβαση στις περιφερειακές συσκευές γίνεται κάτω από περιορισμούς. Για παράδειγμα, εάν μπορώ να διαβάσω οποιοδήποτε block του δίσκου, μπορώ να διαβάσω και αρχεία άλλων χρηστών, ακόμα και αν *δεν* έχω δικαίωμα να τα διαβάσω.
- Γι' αυτό, τα συστήματα διαχωρίζουν δύο (τουλάχιστον) επίπεδα δικαιωμάτων: του χρήστη (User) με πολλούς περιορισμούς και του συστήματος (Kernel)
- Για ασφάλεια μόνο το σύστημα μπορεί να εξυπηρετήσει διακοπές.
- Ένα ειδικό bit το «Kernel /User mode» δείχνει τα εκάστοτε δικαιώματα της διεργασίας που τρέχει στον επεξεργαστή.
- Το λειτουργικό σύστημα παρεμβάλλεται ανάμεσα από τους χρήστες και τις συσκευές, ώστε να επιβάλλει κανόνες σωστής χρήσης

# Εξυπηρέτηση Διακοπών

- Όταν συμβεί μια διακοπή ο επεξεργαστής πρέπει να:
  - Σώσει αρκετή πληροφορία για να μπορεί να επιστρέψει στην κανονική διεργασία αργότερα
  - Βρει την πηγή της διακοπής ανάμεσα στις πιθανά πολλαπλές συσκευές και λόγους διακοπής
  - Διακλαδιστεί στο κώδικα που εξυπηρετεί διακοπές (interrupt handler)

# Ποιά Διακοπή/Συσκευή Εξυπηρετούμε;

Για την εξυπηρέτηση μιας διακοπής χρειάζεται να διακλαδωθεί η εκτέλεση στον αντίστοιχο interrupt handler. Δυνατότητες:

- Vectored Interrupts:
  - Ο επεξεργαστής χρησιμοποιεί τον κωδικό της συσκευής σαν δείκτη σε ένα πίνακα από αρχικές διευθύνσεις όλων των interrupt handler. Κατόπιν διακλαδίζεται στην διεύθυνση που μόλις διάβασε. (π.χ. 80x86) (Μειονέκτημα: δυσκολία υλοποίησης)
- Κοινόχρηστος interrupt handler
  - Ο επεξεργαστής έχει μία και μόνη διεύθυνση για το interrupt handler. Εντολές μέσα στον interrupt handler θα διαπιστώσουν ποιά συσκευή δημιούργησε την διακοπή και θα την εξυπηρετήσουν. (π.χ. MIPS) (Μειονέκτημα: λίγο πιο αργό από τα Vectored Interrupts)
  - «Polling» των συσκευών για να βρεθεί η σωστή, εκτός αν υποστηρίζεται κάποια προσομοίωση των Vectored Interrupts!
  - Ο MIPS δίνει στον Cause register τον κωδικό της συσκευής  $\approx$  vector



# Εξυπηρέτηση Διακοπών MIPS

- Οι διακοπές διαχειρίζονται από τον «συνεπεξεργαστή 0» (coprocessor 0) , ο οποίος βρίσκεται μαζί με τον κύριο επεξεργαστή.
- Ο συνεπεξεργαστής είναι λογικά μια αυτόνομη μονάδα με τους δικούς της καταχωρητές.
  - \$12 = Status register (processor mode, κλπ.)
  - \$13 = Cause register (τύπος διακοπής κλπ.)
  - \$14 = EPC (Exception PC) (διεύθυνση διακοπής)
- Βρίσκεται όμως στο ίδιο ολοκληρωμένο με τον επεξεργαστή
- Επικοινωνεί με τον επεξεργαστή με τις εντολές mfc0 (move from coprocessor 0) και mtc0 (move to coprocessor 0)

# Εξυπηρέτηση Διακοπών MIPS #2

- Όταν έρθει μια διακοπή:
  - Αποθήκευση του PC στον καταχωρητή EPC
  - Αποθήκευση λόγου διακοπής στον καταχωρητή Cause Register
  - Απενεργοποίηση διακοπών
  - Σώσιμο του επιπέδου επεξεργασίας
  - Αλλαγή του επιπέδου επεξεργασίας σε «kernel»
  - Διακλάδωση στην διεύθυνση 0x80000080 (0x80000180 στον MIPS32)
- Ο interrupt handler
  - Βρίσκει την συσκευή που χρειάζεται προσοχή και την εξυπηρετεί
  - Σε κάποιο σημείο πρέπει να επανενεργοποιήσει τις διακοπές
- Στην επιστροφή από διακοπή (ειδική εντολή `rfe`: return from exception)
  - Επαναφορά επιπέδου επεξεργασίας
  - Διακλάδωση στην εντολή που διακόπηκε (χρήση του EPC)

# Κωδικοί Διακοπών MIPS

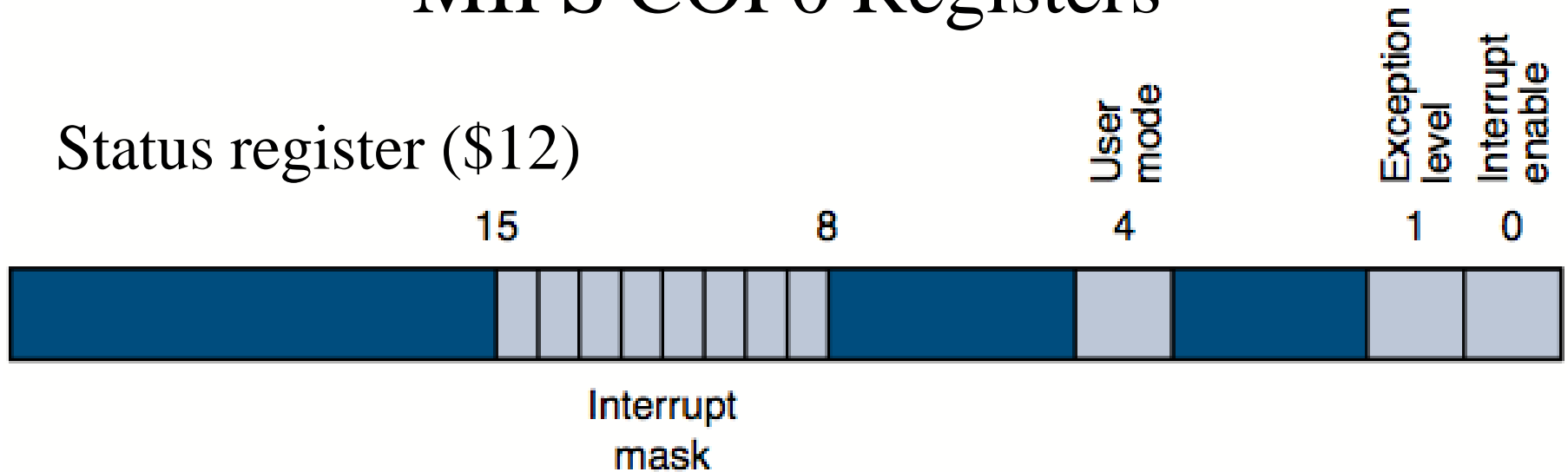
0	INT	Εξωτερική διακοπή (συσκευή I/O)
1,2,3		Εξαιρέσεις εικονικής μνήμης
4	ADDRL	Λάθος διεύθυνσης (ανάγνωση μνήμης)
5	ADDRS	Λάθος διεύθυνσης (εγγραφή μνήμης)
6	IBUS	Bus Error (ανάκληση εντολής)
7	DBUS	Bus Error (ανάκληση δεδομένων)
8	SYSCALL	Κλήση λειτουργικού συστήματος
9	BKPT	Breakpoint
10	RI	Reserved Instruction (παράνομη εντολή )
11	CPU	Λάθος εντολή συνεπεξεργαστή
12	OVF	Υπερχείλιση

# Καταχωρητές Cause και Status

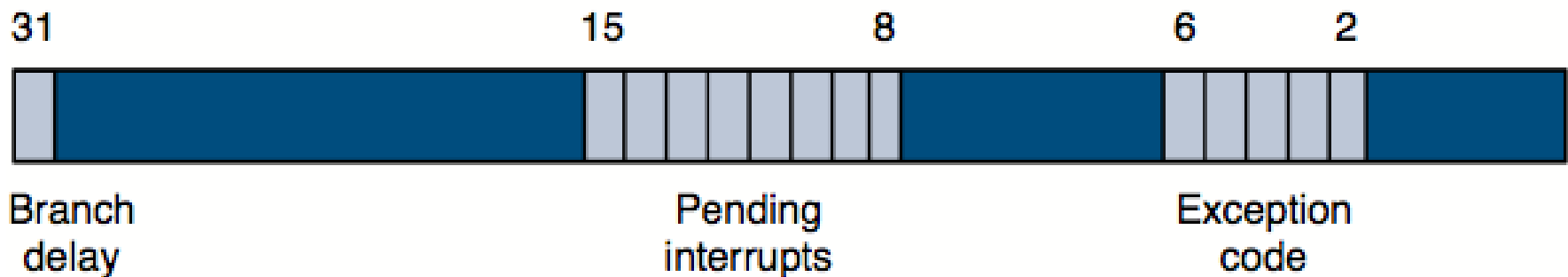
- Cause register (\$13)
  - Bits [6:2] = 5 bit κωδικός είδους διακοπής σύμφωνα με τον προηγούμενο πίνακα
  - Bits [15:8] = 8 bit κωδικός συσκευών που έχουν παράγει διακοπή (μπορεί να είναι  $\geq 1$  μέχρι και 8)
  - Τα υπόλοιπα bits δεν είναι χρησιμοποιούνται.
- Status register (\$12)
  - Bits για διάκριση user/kernel mode, interrupt enable (για όλες τις διακοπές, 8 bits μάσκα interrupt enable ανά συσκευή (ανάλογη με την μάσκα ανά συσκευή στον καταχωρητή Cause). Εάν το bit της μάσκας είναι 1 οι διακοπές από την συσκευή επιτρέπονται, αλλιώς όχι.

# MIPS COP0 Registers

Status register (\$12)



Cause register (\$13)

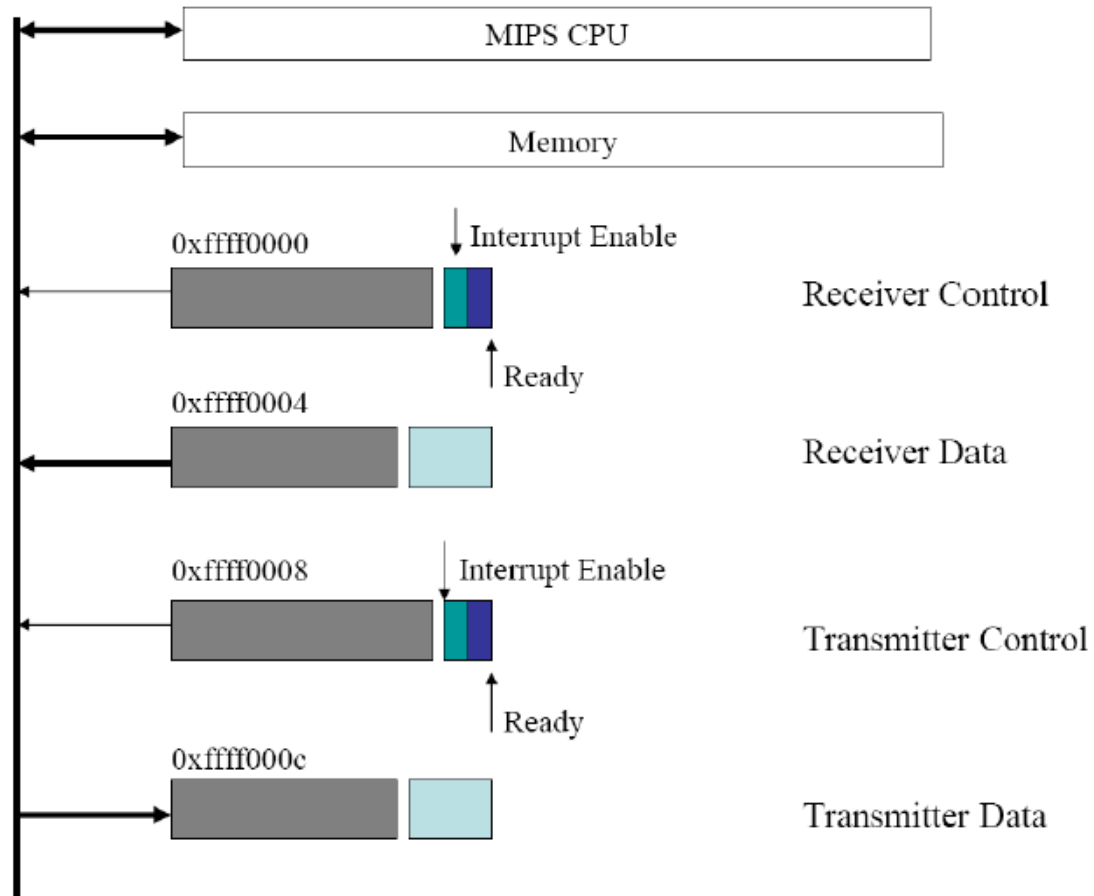


# Επικοινωνία CPU – I/O Devices (1)

- Απλές I/O Devices
  - Πληκτρολόγιο (Receiver)
  - Κονσόλα SPIM
- 2 ειδικές θέσεις στην μνήμη (Memory Mapped I/O)
  - Data
  - Control

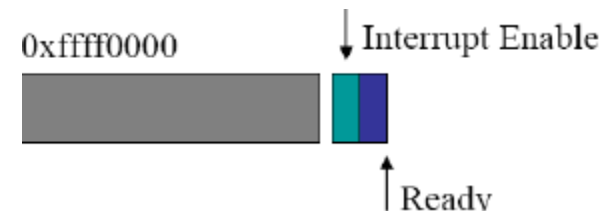
# Memory Mapped I/O

## Διευθύνσεις Memory Mapped I/O



# Control Register

- 32 bits width
  - Χρήσιμα: 2 least significant bits
- Ready bit
  - Είναι ενεργοποιημένο μόνο αν η συσκευή είναι έτοιμη να δεχτεί ή να διαβάσει έναν νέο χαρακτήρα
- Interrupt bit
  - Ενεργοποίηση των Interrupts για την συγκεκριμένη συσκευή





# Data Register

- 32 bits width
  - Χρήσιμα: 8 least significant bits
- 8 bits
  - Ο χαρακτήρας είτε που διαβάστηκε από το πληκτρολόγιο είτε που θα εμφανιστεί στην κονσόλα

0xffff000c



# Memory Mapped I/O Addresses

- Receiver ➔ Πληκτρολόγιο
- Transmitter ➔ Κονσόλα

Όνομα Καταχωρητή	Διεύθυνση
Receiver Control	0xffff0000
Receiver Data	0xffff0004
Transmitter Control	0xffff0008
Transmitter Data	0xffff000c

# Τεχνική Polling

- Π.χ. Να γραφτεί ένας χαρακτήρας στην κονσόλα
  - Διαβάζω τα περιεχόμενα της θέσης μνήμης 0xffff0008
  - Αν το Ready bit είναι 0 συνέχισε να διαβάζεις την παραπάνω θέση μνήμης
  - Αν το Ready bit είναι 1 αποθήκευσε τον χαρακτήρα (8 bits) στη θέση μνήμης 0xffff000c.
- Q1: Πώς κάνω ανάγνωση ενός χαρακτήρα από το πληκτρολόγιο?

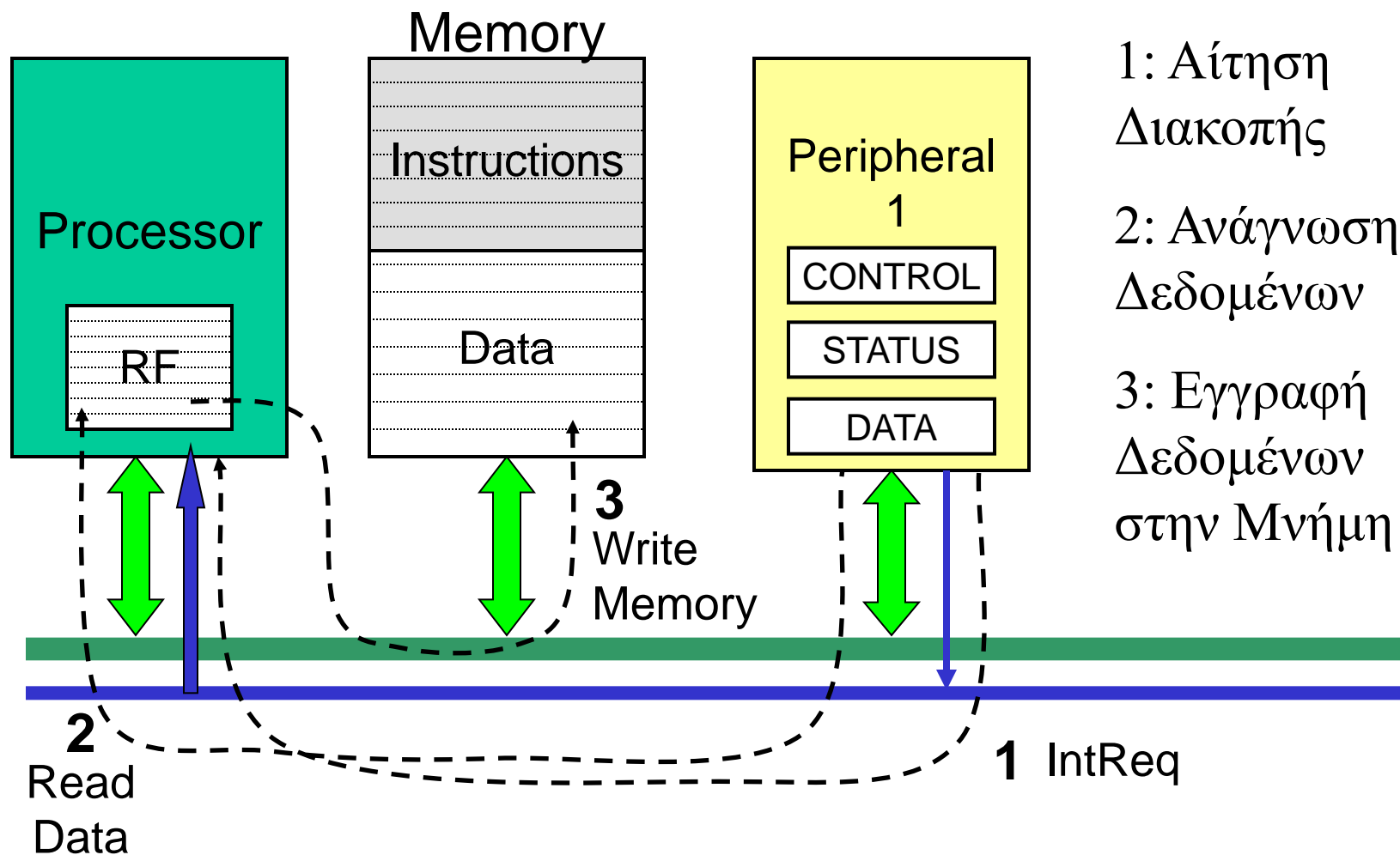
# Προχωρημένα Θέματα Διακοπών

- Κόστος Διακοπών:
  - Ο επεξεργαστής χρειάζεται αρκετούς κύκλους για να ξεκινήσει την εκτέλεση του interrupt handler.
  - Ο interrupt handler χρειάζεται αρκετές εντολές για να βρει την πηγή της διακοπής.
  - Συνολικό κόστος εξυπηρέτησης διακοπής πολλές δεκάδες ή λίγες εκατοντάδες εντολές.
- Κόστος Polling:
  - Λίγες εντολές για έλεγχο των συσκευών
- Σε ένα σύστημα με πολύ συχνές διακοπές, το κόστος των διακοπών μπορεί να είναι μεγαλύτερο από το κόστος του Polling! => Όταν έχω διακοπή, εξυπηρέτηση όλων των συσκευών που χρειάζονται προσοχή!

# Προχωρημένα Θέματα Διακοπών #2

- Τι γίνεται εάν μια διακοπή ζητηθεί ενώ μια άλλη εξυπηρετείται;
  - Η δεύτερη διακοπή περιμένει μέχρι η πρώτη να εξυπηρετηθεί. Απλό, αλλά σε συσκευές με μεγάλη ταχύτητα μεταφοράς η αναμονή πολλές φορές δεν είναι δυνατή!
  - Φωλιασμένη εξυπηρέτηση διακοπών: Η πρώτη διακοπή διακόπτεται για να ξεκινήσει η εξυπηρέτηση της δεύτερης
- Προτεραιότητες Διακοπών:
  - Ποιές διακοπές επιτρέπεται να διακοπούν? Τα συστήματα συνήθως ορίζουν προτεραιότητες στις διακοπές. Η εξυπηρέτηση μιας διακοπής μπορεί να διακοπεί μόνο από διακοπή υψηλότερης προτεραιότητας

# Μεταφορά Δεδομένων με Διακοπές



# Μεταφορά Δεδομένων με Διακοπές #2

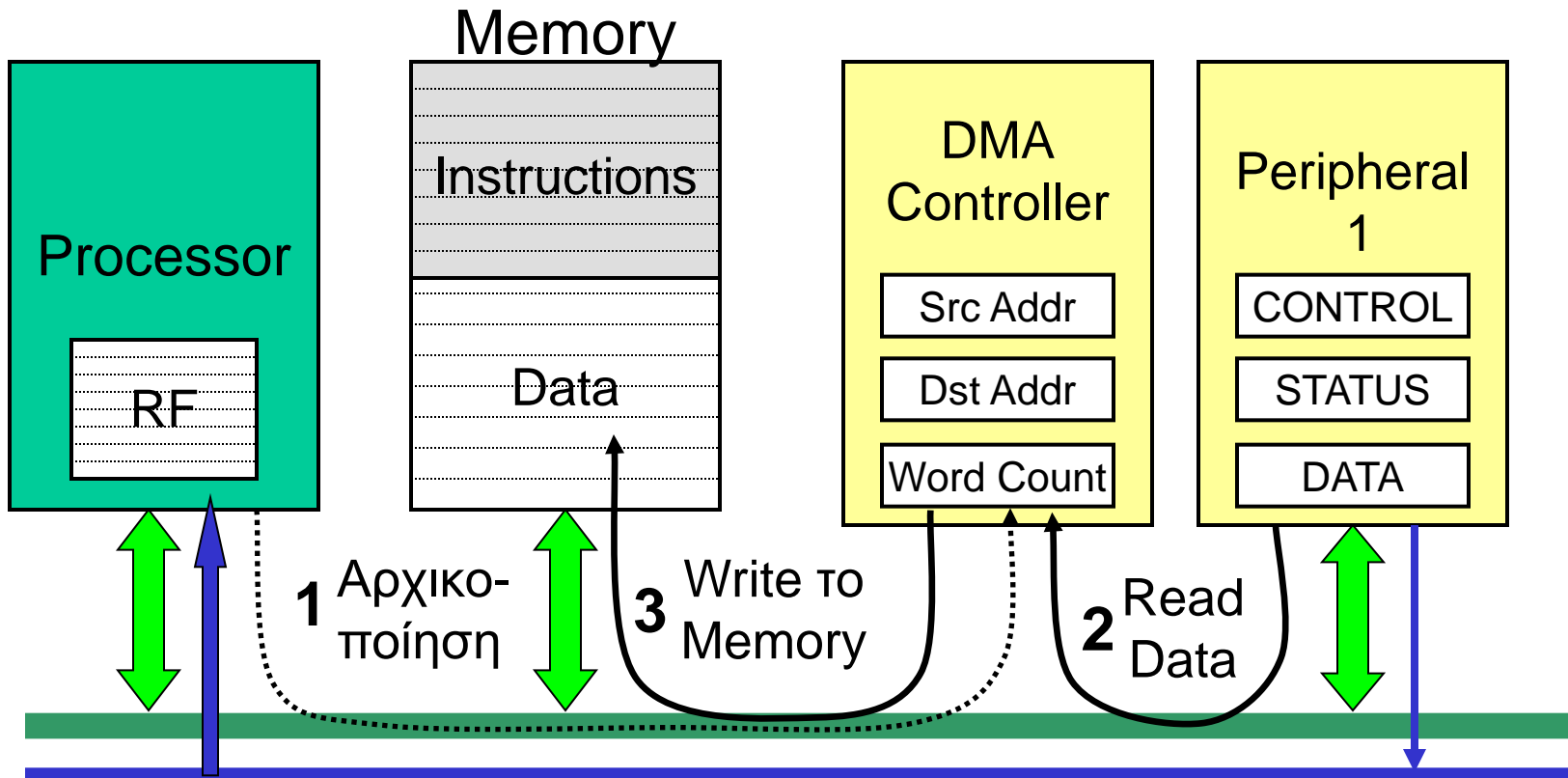
- Συνήθως ο interrupt handler δέχεται τα δεδομένα και να βάζει στην μνήμη για να χρησιμοποιηθούν αργότερα από τον χρήστη. Αυτό σημαίνει ότι για να διαβάσουμε μια λέξη, πρέπει: (α) να γίνει η αίτηση διακοπής, (β) να εξυπηρετηθεί η αίτηση και να διαβαστεί ο καταχωρητής δεδομένων, και (γ) να γραφτεί η λέξη στην μνήμη.
- Η διαδικασία ενώ είναι σε τελική ανάλυση απλή (μεταφορά μίας λέξης από το περιφερειακό στην μνήμη) κοστίζει δεκάδες ή εκατοντάδες εντολές και απασχολεί τον επεξεργαστή.
- ΛΥΣΗ: Άμεση Πρόσβαση Μνήμης (Direct Memory Access - DMA)

# Direct Memory Access

- Η Άμεση Πρόσβαση Μνήμης απαιτεί έναν ελεγκτή (DMA Controller), ο οποίος λειτουργεί σαν ένας μικρός και πολύ απλός επεξεργαστής.
- Το μόνο που κάνει είναι με δεδομένες δύο διευθύνσεις A και B και ένα αριθμό λέξεων N, να διαβάσει διαδοχικά N λέξεις ξεκινώντας από το A, και να τις αντιγράψει σε διαδοχικές διευθύνσεις ξεκινώντας από το B.
- Καταχωρητές DMA controller (ο οποίος εμφανίζεται σαν κανονική περιφερειακή συσκευή):
  - Source Address
  - Destination Address
  - Byte Count (ή Word Count)



# Μεταφορά Δεδομένων με DMA



# Direct Memory Access #2

- Ο επεξεργαστής αρχικοποιεί τους καταχωρητές του ελεγκτή DMA, και μετά **δεν** ασχολείται με την μεταφορά, αλλά εκτελεί το πρόγραμμα του χρήστη.
- Ο ελεγκτής DMA διαβάζει από το περιφερειακό και γράφει στην μνήμη (ή το αντίθετο)
- Όταν τελειώσει η μεταφορά, ο ελεγκτής DMA ειδοποιεί τον επεξεργαστή με μια διακοπή
- Ο ελεγκτής DMA μπορεί να ξεκινήσει πρόσβαση στον δίαυλο, δηλαδή να είναι «bus master»
- Μόνο κόστος για τον επεξεργαστή και το πρόγραμμα του χρήστη ότι σε προσβάσεις στην μνήμη, ο επεξεργαστής μπορεί να βρει την μνήμη απασχολημένη από το DMA, λόγω κοινής χρήσης του δίαυλου (bus)

# Πρόσβαση στον Δίαυλο από τον Ελεγκτή DMA

- Ο διάυλος (Bus) είναι κοινόχρηστος => μόνο ένας μπορεί να γράφει ή να διαβάζει κάθε χρονική στιγμή
- Αίτηση για ανάγνωση/εγγραφή μπορούν να κάνουν όλοι οι «bus masters», δηλαδή ο επεξεργαστής και ο ελεγκτής DMA
- Τι γίνεται όταν και οι δύο ζητήσουν ταυτόχρονα να διαβάσουν για παράδειγμα κάτι από την μνήμη;
- Διαιτησία (arbitration) για να επιλεγθεί ένας master ο οποίος θα έχει πρόσβαση στο διάυλο. Ο άλλος master θα περιμένει μέχρι η πρώτη πρόσβαση να τελειώσει, και θα ξανακάνει αίτηση για τον διάυλο
- Πώς επιλέγω έναν από τους bus-masters?
  - **Προτεραιότητες:** Π.χ. Ο επεξεργαστής είναι πιο σημαντικός από το DMA, οπότε θα διαλέγω τον επεξεργαστή πρώτα
  - **Round-Robin:** εναλλάσσω την σειρά ώστε όλοι να αποκτούν πρόσβαση ο ένας μετά τον άλλο

# Παράδειγμα

**ΔΕΔΟΜΕΝΑ:** CPU 100MHz (διάρκεια κύκλου ρολογιού 10 nsec). Οι εντολές χρειάζονται 1 κύκλο εκτός από lw η οποία χρειάζεται 2 κύκλους. 33% όλων των εντολών είναι lw/sw (πρόσβαση μνήμης), και τα lw είναι διπλάσια από τα sw. Η εγγραφή στη μνήμη χρησιμοποιεί τον δίαυλο για 1 κύκλο, ενώ η ανάγνωση για 2 κύκλους (διεύθυνση στον πρώτο, απάντηση με τα δεδομένα στον δεύτερο). Ο δίαυλος έχει πλάτος μία λέξη. Αγνοήστε την ανάκληση εντολών.

**ΖΗΤΟΥΜΕΝΟ:** Ποιά είναι η μέγιστη ταχύτητα μεταφοράς που μπορεί να επιτύχει η Άμεση Πρόσβαση Μνήμης (DMA);

# Παράδειγμα

**ΔΕΔΟΜΕΝΑ:** CPU 100MHz (διάρκεια κύκλου ρολογιού 10 nsec). Οι εντολές χρειάζονται 1 κύκλο εκτός από lw η οποία χρειάζεται 2 κύκλους. 33% όλων των εντολών είναι lw/sw (πρόσβαση μνήμης), και τα lw είναι διπλάσια από τα sw. Η εγγραφή στη μνήμη χρησιμοποιεί τον δίαυλο για 1 κύκλο, ενώ η ανάγνωση για 2 κύκλους (διεύθυνση στον πρώτο, απάντηση με τα δεδομένα στον δεύτερο). Ο δίαυλος έχει πλάτος μία λέξη. Αγνοήστε την ανάκληση εντολών.

**ΖΗΤΟΥΜΕΝΟ:** Ποιά είναι η μέγιστη ταχύτητα μεταφοράς που μπορεί να επιτύχει η Άμεση Πρόσβαση Μνήμης (DMA);

**ΑΠΑΝΤΗΣΗ:** Σε εκτέλεση 9 εντολών του επεξεργαστή, έχω 2 lw, 1 sw και 6 απλές εντολές (add κ.α.). Η εκτέλεση αυτών των εντολών θα χρειαστούν:  $(6+1) * 1 + 2(lw) * 2 = 11$  κύκλοι = 110 nsec.

Στη διάρκεια των 11 κύκλων, ο επεξεργαστής χρησιμοποιεί τον δίαυλο για 1 sw \* 1 κύκλο + 2 lw \* 2 κύκλους = 5 κύκλοι. Δηλαδή από τους 11 κύκλους οι 11 – 5 = 6 κύκλοι είναι διαθέσιμοι για DMA.

Η μεταφορά μίας λέξης απαιτεί μία ανάγνωση και μία εγγραφή, δηλαδή 2 + 1 = 3 κύκλους. Συνεπώς κάθε 11 κύκλους μπορώ να μεταφέρω με DMA:  $6 / 3 = 2$  λέξεις = 8 bytes. Ο μέγιστος ρυθμός μεταφοράς DMA είναι 8 bytes κάθε 11 κύκλους, δηλαδή  $8 \text{ bytes} / 110 \text{ nsec} = 72.727 \text{ Mbytes/sec}$ .

# Προχωρημένα Θέματα DMA

- Σε ένα υπολογιστή υπάρχουν πολλές περιφερειακές συσκευές, οι οποίες μπορούν να δουλεύουν ταυτόχρονα (π.χ. Δίσκος, και κάρτα δικτύου).
  - ⇒ Ανάγκη για πολλαπλά «κανάλια» DMA, ένα για κάθε μεταφορά. Το κάθε κανάλι έχει τους δικούς του καταχωρητές src και dest address, και word count, και λειτουργεί αυτόνομα. Συνήθως 4 κανάλια DMA είναι αρκετά για απλά συστήματα (σταθμοί εργασίας, όχι servers).
- Πώς ξέρει ο DMA controller πότε είναι έτοιμη μια συσκευή;
  - Μπορεί να ξέρει κάθε πότε είναι έτοιμη η συσκευή από τον ρυθμό μεταφοράς της (για κάποιες συσκευές είναι δυνατό). Ένας καταχωρητής wait\_time δείχνει στον ελεγκτή πόσο πρέπει να περιμένει για να διαβάσει (γράψει) την επόμενη λέξη.
  - Ο DMA controller μπορεί να διαβάζει το status register (polling)
    - ⇒ Θυμίζει μικρό και απλό επεξεργαστή!
    - ⇒ Εκτελεί προγράμματα; Σε (παλαιά/) μεγάλα συστήματα (mainframes) **ΝΑΙ** και λέγεται *I/O Processor*!
- Απλοί DMA controllers **δεν** έχουν αυτές τις δυνατότητες