

Άσκηση 13.4

α) Στο τέλος του φηδιδίου

Hex	Binary	Θέση	Hex	Binary	Θέση
B) 000	= 000000000000	λέξη μνήμης 0	70E	= 11100001110	Παράνομη
00E	= 00000001110	λέξη μνήμης 14	70F	= 11100001111	Παράνομη
100	= 001000000000	λέξη μνήμης 256	710	= 11100010000	Παράνομη
200	= 010000000000	λέξη μνήμης 512	780	= 111100000000	Παράνομη
2FF	= 01011111111	λέξη μνήμης 767	7C0	= 111110000000	status IN
300	= 011000000000	λέξη μνήμης 768	7C3	= 11111000011	data OUT
3FF	= 01111111111	λέξη μνήμης 1023	7C7	= 11111001111	Παράνομη
400	= 100000000000	Θέση 0 Buffer	7FF	= 11111111111	Παράνομη
4FF	= 10011111111	Θέση 255 Buffer	800	= 100000000000	λέξη μνήμης 0
500	= 101000000000	Θέση 256 Buffer	808	= 100000001000	λ.μν. 8
5FF	= 10111111111	Θέση 511 Buffer	80A	= 100000001010	λ.μν. 10
600	= 110000000000	Παράνομη	80E	= 100000001110	λ.μν. 14
608	= 11000001000	Παράνομη	80F	= 100000001111	λ.μν. 15
60A	= 11000001010	Παράνομη	C0E	= 110000001110	Θέση 14 Buffer
60E	= 11000001110	Παράνομη	C0F	= 110000001111	Θέση 15 Buffer
60F	= 11000001111	Παράνομη	E02	= 111000000010	Θέση 2 Buffer
610	= 11000010000	Παράνομη	E12	= 111000100010	Θέση
680	= 110100000000	Παράνομη	F02	= 111100000010	Παράνομη
6C0	= 110110000000	Παράνομη	F12	= 111100010010	Παράνομη
6C3	= 11011000011	Παράνομη			
6CF	= 11011001111	Παράνομη			
6FF	= 11011111111	Παράνομη			
700	= 111000000000	Παράνομη			
708	= 11100001000	Παράνομη			
70A	= 11100001010	Παράνομη			

Assn 13.5

a)

```
char read-kbd-busywait-char() {  
    int *status;  
    char *data;  
    status = 11111000000;  
    while (*status == 0) {  
        status = 11111000000;  
    }  
    *status = 0  
    data = 11111000000;  
    return *data;  
}
```

b)

```
char read-kbd-polling-char() {  
    int *status;  
    char *data;  
    status = 11111000000;  
    if (*status == 0) return '\0'  
    *status = 0  
    data = 11111000000;  
    return *data;  
}
```


Άσκηση 13.6

Ρολόι επεξεργαστή: 1 GHz

Overhead: 2000 κύκλοι ρολογιού

Χώστος δειγματοληψίας: 200 κύκλοι ρολογιού

α) 40 σημεία ταχύτητας 1ms, δηλαδή 1kHz

1 Διακοπή ανά 1ms

i) Για κάθε διακοπή ξοδεύουμε 2000 κύκλους ρολογιού

ii) Για τις 40 δειγματοληψίες ξοδεύουμε $40 \cdot 200 = 8000$ κύκλους ρολογιού

Αν αυτό επαναλαμβάνεται 1000 φορές το δευτερόλεπτο, κάθε second ξοδεύουμε $1000 \cdot (8000 + 2000) = 10.000.000$ κύκλους ρολογιού. Ο επεξεργαστής είναι 1GHz, πράγμα που σημαίνει ότι ο επεξεργαστής εκτελούνται 1.000.000.000 εντολές το δευτερόλεπτο. Από το ~~απώλει~~ κόστος το ποσοστό της υπολογιστικής δυναμικότητας του επεξεργαστή που αντιπροσωπεύει τους 10.000.000 κύκλους είναι:

~~Απώλει~~

$$\frac{10.000.000}{1.000.000.000} = 0,01 = 1\%$$

Είναι το 1% της υπολογιστικής δυναμικότητας της CPU.

β) Έχουμε 40 συσχευτές (μολέριες) και κατά μέσο όρο έρχονται 40 εισοδοί ανά δευτερόλεπτο. Αρα για όλες τις γραμμές οι νέες εισοδοί ανά δευτερόλεπτο θα είναι: $40 \cdot 40 = 1600$ εισοδοί. Αν κάθε εισόδος προκαλεί μια διακοπή τότε θα έχουμε κατά μέσο όρο 1600 διακοπές, οι οποίες θα ξοδεύουν $1.600 \cdot 2.000 = 3.200.000$ κύκλους ρολογιού. Το ποσοστό της υπολογιστικής δυναμικότητας της CPU που αντιπροσωπεύει τους 3.200.000 κύκλους είναι $3.200.000 / 1.000.000.000 = 0,0032 = 0,32\%$. Σημφεροουν οι διακοπές. β), αφού $0,32\% < 1\%$.

γ) Τώρα θα έχουμε $40 \cdot 400 = 16.000$ εισόδους ανά δευτερόλεπτο. Αν έχουμε σε κάθε είσοδο και μια διακοπή 2000 κήνων, τότε ~~δευτερόλεπτο θα έχουμε~~ ~~ζευγαριά~~ συνολικά αυτές θα κοστίζουν στον επεξεργαστή $16.000 \cdot 2000 = 32.000.000$ κήνους ρολογιών. Αυτό αντιστοιχεί στο 3,2% της συνολικής ισχύος του επεξεργαστή. Σε αυτή, επομένως, την περίπτωση συμφέρει καλύτερα η δειγματοληψία α).

δ) Δεν κινδυνεύουμε να χάσουμε κάποια είσοδο (ότις περιπτώσεις (β) και (γ)) αν "πέσουν μαζεμένες" νέες εισοδοί από όλες τις γραμμές. Για να εξυπηρετήσει τις 40 αυτές διακοπές θα χρειαστεί $40 * 2000 = 80.000$ κύκλους ρολογιού. Επειδή έχουμε $clock = 1 \text{ GHz}$, κάθε κύκλος διαρκεί 1 ns , επομένως οι 80.000 κύκλοι θα διαρκούν ~~80.000 ns~~ $80.000 \text{ ns} = 80 \text{ μs}$. Το νωρίτερο που μπορεί να έρθει μια είσοδος είναι 1 ms , επομένως ~~απλά~~ ο επεξεργαστής ~~θα~~ θα προλάβει να τελειώσει πριν έρθει η επόμενη εντολή ^{δεν} άρα ^{δεν} κάνουμε δεδομένα.

Ε) Αντι για 40 εισόδους έχουμε μια η οποία μεταφέρει 32 Mbits ή 4 MBytes/s. Επιπρόσθετα, σύμφωνα και με την εκκρίνιση, μας έρχονται 100 k πακέτα το δευτερόλεπτο. Άρα αν έχουμε διακοπές, ο επεξεργαστής θα ζοδεύει κάθε δευτερόλεπτο: ~~100.000~~ $100.000 \cdot 2000 = 200.000.000$ κύκλους ρολογιού, δηλαδή το 20% της ισχύος της CPU για διακοπές. Αν από την άλλη, είχαμε δειγματοληψία, τότε κάθε δειγματοληψία θα κοστίζει $2000 + 200 = 2200$ κύκλους ρολογιού. Σύμφωνα με α) έχουμε 1000 δειγματοληψίες το δευτερόλεπτο επομένως η CPU ζοδεύει $2200 \cdot 1000 = 2.200.000$ κύκλους το δευτερόλεπτο για δειγματοληψία, που αντιστοιχεί στο 0.22% της ισχύος της CPU. Άρα η δειγματοληψία είναι πιο καλή επιλογή.

στ) Με περίπου 30 φορές χημωρότερη γραμμή δικτύου από αυτή
του ε) θα έχουμε 30 φορές περισσότερους κώδικους το δευτερόλεπτο,
δηλαδή $30 * 200.000.000 = 6.000.000.000$ κώδικους το δευτερόλεπτο.
Η CPU μας κάνει 1.000.000.000 κώδικους το δευτερόλεπτο, επομένως
ο επεξεργαστής δεν μπορεί να "ανταίρει" αυτήν την μέθοδο, με μια διακοπή
και αφιχνωμένο πακέτο.

Άσκηση 13.7

Ρολόι CPU: 1 GHz

Ρυθμός μετάδοσης: 1 Byte κάθε 3.2 ns (κάθε 3.2 κώδικους)

Κάθε πακέτο έχει μια επικεφαλίδα μεγέθους 12 Bytes

α) Ρυθμός αντιγραφής 4 Bytes ανά 80 ns, ~~αφαιρείται~~

Σε 1 second ο επεξεργαστής κάνει 1.000.000.000, δηλαδή
 $1 \text{ sec} = 1.000.000.000 \text{ ns}$. Σε ένα second χωράνε τα 80 ns:

$1.000.000.000 / 80 = 12.500.000$ φορές. Αυτό σημαίνει ότι
το δευτερόλεπτο έχουμε ~~12500000~~ πακέτα των 4 Byte. Κάθε
δευτερόλεπτο δηλαδή έχουμε $4 * 12.500.000 = 50.000.000 \text{ Bytes}$
που ~~προσδυναί~~ με: 47,684 MB/s ή 381,470 Mbits/s. Για
ευκολία μπορούμε να πούμε πως περίπου έχουμε 50 MB/s ή 400 Mbits/s.
Ρυθμό αντιγραφής. Εάν η CPU τροφοδοτεί δύο δίσκους με παροχή
10 MBytes/s τότε χρειάζεται in total $2 * 10 \text{ MB/s} = 20 \text{ MB/s}$. Επιπρόσθετα,
τροφοδοτούμε και μία γραμμή ethernet των 100 Mbits/s που είναι περίπου
160 με 12500 MB/s. Άρα συνολικά χρειάζεται $20 + 12.5 = 32.5 \text{ MB/s}$
που είναι το $32.5 / 50 = 65\%$ της ισχύος του.

β) i) Γνωρίζουμε ότι ο ρυθμός μετάδοσης είναι 1 Byte ανά 3.2 ns και κάθε πακέτο έχει 1 header μεγέθους 12 Bytes, ορα εδώ τα πακέτα είναι μεγέθους 16 Bytes.

1 πακέτο είναι 16 Bytes, ορα χρειάζεται 51.2 ns. Έχουμε συνολικά ανά 51.2 ns ένα πακέτο, που είναι ίσο με (19.531.250)

19,5 M πακέτα το δευτερόλεπτο. Αφαιρούμε τα headers από

~~κάθε πακέτο, που σημαίνει ότι τώρα κάθε πακέτο είναι~~

~~μεγέθους 4 Bytes, χρειάζονται 12,8 ns. Ως εκ τούτου Αφαιρούμε~~

Επίσης, γνωρίζουμε ότι ο ρυθμός μετάδοσης του bus είναι

περίπου 312,5 MB/s. Αφαιρούμε τα headers από κάθε

πακέτο και έχουμε: ~~312,5 - (19,5 * 12)~~

$$312,5 - (19,5 * 12) \approx 78 \text{ MB/s} \approx 624 \text{ Mbits/s}$$

ii) 76 Bytes κάθε πακέτο

4 M πακέτα ανά δευτερόλεπτο

Αρα οα έχουμε

$$312,5 - (4 * 12) \approx 264,5 \text{ MB/s} \approx 2.116 \text{ Mbits/s}$$

γ) Γνωρίζουμε πως για την εξυπηρέτηση των σημάτων δίκλων και του δικτύου χρειαζόμαστε 32,5 MB/s, που είναι το:

$$32,5 / 312,5 = 0.104 = 10,04\% \text{ της αρτηρίας/bus.}$$

Η μεγάλη μείωση της απασχόλησης σε σχέση με το α), οφείλεται στο γεγονός ότι πλέον μεταφέραμε μεγαλύτερα πακέτα κάθε φορά και όχι μια λέξη τη φορά, καθώς και στην μεγαλύτερη καταπόνηση της αρτηρίας - bus.

