

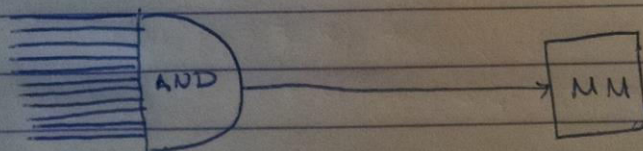
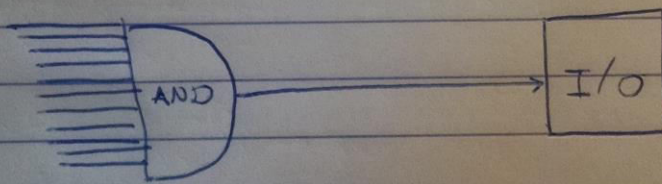
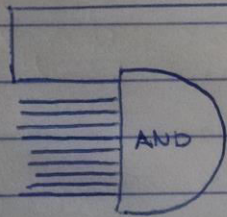
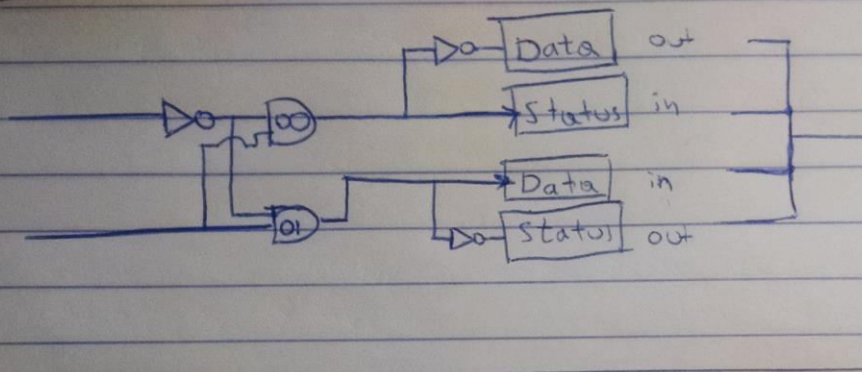
ΜΑΡΑΝΤΟΣ ΑΛΕΞΑΝΔΡΟΣ

ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ 13

3329

Assignment 13.4

Proc



e)

Hex	Bin	Adress (MM)
000	0000 0000 0000	Adress 0 (MM)
00E	0000 0000 1110	Adress 14 (MM)
100	0001 0000 0000	Adress 256 (MM)
200	0010 0000 0000	Adress 512 (MM)
2FF	0010 1111 1111	Adress 767 (MM)
300	0011 0000 0000	Adress 768 (MM)
3FF	0011 1111 1111	Adress 1023 (MM)
400	0100 0000 0000	buffer 0
4FF	0100 1111 1111	buffer 255
500	0101 0000 0000	buffer 256
5FF	0101 1111 1111	buffer 511
600	0110 0000 0000	napisano
608	0110 0000 1000	>
60A	0110 0000 1010	>
60E	0110 0000 1110	>
60F	0110 0000 1111	>
610	0110 0001 0000	>
680	0110 1000 0000	>
6CD	0110 1100 0000	>
6C3	0110 1100 0011	>
6CF	0110 1100 1111	>
6FF	0110 1111 1111	>
700	0111 0000 0000	>
708	0111 0000 1000	>
70A	0111 0000 1010	>
70E	0111 0000 1110	>
70F	0111 0000 1111	>
710	0111 0001 0000	>
780	0111 1000 0000	>

7C0	0111 1100 0000	IN (status)
7C3	0111 1100 0011	OUT (data)
7CF	0111 1100 1111	παράκληση
7EF	0111 1111 1111	↗
800	1000 0000 0000	E/E θύρα 0
808	1000 0000 1000	E/E θύρα 9
80A	1000 0000 1010	E/E θύρα 10
80E	1000 0000 1110	E/E θύρα 14
80F	1000 0000 1111	E/E θύρα 15
C0E	1100 0000 1110	παράκληση
C0F	1100 0000 1111	παράκληση
F02	1111 0000 0010	παράκληση
F0B	1111 0001 0010	παράκληση

ΑΣΚΗΣΗ 13.5

a)

```

read_kbd_busywait_char( *character){
    while ( load_StatusReg){
        char New;
        check (load_StatusReg);
        if(check){
            New = get.char();
            return New;
        }
    }
    return *character

```

b)

```

read_kbd_polling_char(*character)

    check (load_StatusReg);
    if (check)

```

```

{
    return *character;
}
else
{
    return '\0';
}

```

ΑΣΚΗΣΗ 13.6

Συχνότητα ρολογιου : 1GHz

Overhead : 2000 cycles

Κοστος δειγματοληψιας : 200 cycles

a) Καταγραφη 40 σηματον

Μεγιστος ρυθμος κάθε εισοδου 1 kHz

Κάθε ένα εκ των 40 λαμβανει νεα εισοδο λοιπον κάθε 1 ms

$1 \text{ (διακοπη)} / 1 \text{ ms} = 0.001 \text{ s}$

i) 2000 cycles ακομα

ii) $200 * 40 \text{ (σηματα)} = 8000 \text{ cycles}$

Συνολο $2000 + 8000 = 10000 \text{ cycles}$

Αρα $10000 * 1000 = 10^7 \text{ cycles}$

Δηλαδη $10^7 / 10^9 = 0.01$

Δηλαδη 1%

b)

Καινουριες εισοδοι = $40 \text{ (παλιοι)} * 40 \text{ (νεοι)} = 1600 \text{ εισοδοι}$

Οι κυκλοι τωρα είναι : $= 40 * \text{το κοστος των διακοπων}$

$= 40 * 40 * 2000$

$= 3200000 \text{ κυκλοι}$

Ενώ το ποσοστο = $3200000 / 10^9 = 0.32 \%$

Αρα η διακοπη συμφερει

c)

Σε κάθε γραμμή έχουμε 500 εισόδους

$$\begin{aligned}\text{Αρα οι κυκλοι είναι :} &= 500 * 40 * 2000 \\ &= 40000000\end{aligned}$$

$$\text{Ενώ το ποσοστο} = 40000000 / 10^9 = 4\%$$

Αρα η δειγματοληψια συμφερει τωρα

d)

Αν ισχυσει η περιπτωση β τότε υπαρχει πιθανοτητα καποια σηματα να χαθουν αν δεν γινει γρηγορα το polling

Ενώ αντιθετα στο γ δεν υπαρχει τετοιος κινδυνος απου τα σηματα καθυστερουν σημαντικα να φτασουν

$$\text{Κοστος} = 40 * 2000 = 80000 \text{ κυκλοι}$$

$$\text{Χρονος} = 80000 / 1069 = 0.08 \text{ ms}$$

Αφου για καθε νεα εισοδο χρειαζεται 1 ms και ο χρονος εκτελεσης είναι 0.08 ms , οι 40 διακοπες που επονται προλαβαινουν να εκτελεστουν

e)

$$\text{Κάθε εισοδος } 32 \text{ Mbits/s} \rightarrow 4 \text{ Mbytes/s}$$

Buffer του ενος πακετου

Μικροτερο δυνατο πακετο 40 Bytes

$$\text{Ρυθμος αφιξης } 100 \text{ K (πακετα)}/s$$

$$\text{Ενώ οι διακοπες είναι} = 100 \text{ (πακετα)} / s * 1000 = 100000 \text{ διακοπες} / s$$

$$\text{Ενώ και οι κυκλοι είναι: } 100000 * 2000 = 2 * 10^8 \text{ κυκλοι}$$

$$\text{Αρα το ποσοστο είναι : } 2 * 10^8 / 10^9 = 20\%$$

f)

$$1 \text{ Gbits/sec} = 0.125 \text{ Gbytes/s}$$

$$0.125 \text{ Gbytes/s} = 125 \text{ Mbytes/s} = 125000 \text{ Kbytes/s}$$

$$\text{Ο ρυθμος αφιξης των πακετων είναι } 125000 \text{ KBytes} / 40 \text{ Bytes} = 3125 * 10^3 \text{ πακετα}$$

$$\text{Ενώ } 3125 * 10^3 * 1000 = 3125 * 10^6 \text{ είναι οι διακοπες} / s$$

$$\text{Ο αριθμος των κυκλων είναι } 3125 * 10^6 * \text{overhead} = 625 * 10^7 \text{ κυκλοι}$$

Χρειαζεται λοιπον ρολοι 6.25 Ghz ο υπολογιστης

ΑΣΚΗΣΗ 13.7

A)

$$\begin{aligned} 4 \text{ Bytes} / 80 \text{ ns} &= 1/20 \text{ bytes/ns} = 0.05 \text{ Bytes} / 10^{-9} \text{ s} \\ &= 5 * 10^{-8} \text{ Mbyte} / 10^{-9} \text{ s} = 50 \text{ MB/s} \quad / \quad 400 \text{ Mbits/s} \\ 2 * 10 \text{ Mbytes} / \text{s} \end{aligned}$$

$$100 \text{ Mbits} / \text{s} = 12.5 \text{ Mbytes} / \text{s}$$

$$\text{Δηλαδή } 20 \text{ MB/s} + 12.5 \text{ MB} = 32.5 \text{ MB}$$

$$\text{Ποσοστό λοιπόν } 32.5/50 = 65 \%$$

B)

$$\text{PCI-E των } 2.5 \text{ Gbits/s} = 2500 \text{ Mb/s} = 312 \text{ MB/s}$$

i)

$$\text{ρυθμός} = 200 \text{ MB/s} = 1600 \text{ Mbit/s}$$

ii)

$$64 \text{ Bytes} = 64 * 10^{-6} \text{ MB}$$

$$256 \text{ ns} = 256 * 10^{-9} \text{ s}$$

$$\text{Αρα ρυθμός} = 64 * 10^{-6} / 256 * 10^{-9} = 0.25 * 10^3 \text{ MB/s} = 2000 \text{ Mb/s}$$

Γ)

$$32.5 \text{ MB/s} / 256 \text{ MB/s} = 0.126 = 12.6 \%$$

Η συνολική μείωση της απασχόλησης στη χρήση του επεξεργαστή DMA είναι λογική και αναμενομένη καθώς εκτελεί άλλες ανεξαρτητες εντολές παραλληλα. Επίσης απασχολείται λιγότερο ο επεξεργαστής καθώς το DMA μεταφέρει μεγαλύτερα πακέτα