## Άσκηση 6 Α. Ασκήσεις πάνω στους χρονιστές(Timers)

**Να υπολογισθεί η περίοδος του κύκλου μηχανής στο παρακάτω διάγραμμα και η αρχική τιμή που πρέπει να δίνεται στον Timer0 έτσι ώστε να εκτελούνται διακοπές κάθε 10 ms. Η συχνότητα στην έξοδο του κρυσταλλικού ταλαντωτή είναι 48 MHz. Ο προγραμματιζόμενος διαιρέτης να τεθεί στην τιμή .**

Ταλαντωτής

Εσωτερικός διαιρέτης /4

Προγραμματιζόμενος Διαιρέτης (Prescaler)

Χρονιστής 0

(Timer 0)

Λόγος διαίρεσης από

1/2, 1/4, 1/8, 1/16, 1/32, 1/64, /1/128, 1/256

Διαίρεση 1/1…1/65535

Προαιρετικός

IRQ

Fclock=48 MHz

Τ κύκλος μηχανής(Machine Cycle)

Interrupt Request

Αίτημα διακοπής

**Στη συνέχεια να γραφεί πρόγραμμα με ρουτίνα διακοπών από τον Timer0 με το οποίο αναβοσβήνουν τα LED τα οποία συνδέονται στην πόρτα Β κάθε 200 ms. Προσοχή! Στο πρόγραμμα δεν θα χρησιμοποιηθούν καθόλου συναρτήσεις χρονοκαθυστέρησης.**



Κύκλωμα άσκησης 6Α (Αναβόσβημα των LED κάθε 200 ms με διακοπές από τον Timer0)

**Λύση:**

Α. Υπολογισμός της αρχικής τιμής του Timer0 ώστε να συμβαίνουν διακοπές κάθε 10 ms

Η συχνότητα στην είσοδο του προγραμματιζόμενου διαιρέτη, δηλαδή στην έξοδο του εσωτερικού διαιρέτη θα είναι:

Η συχνότητα στην έξοδο του εσωτερικού διαιρέτη θα είναι:

fέξοδος εσωτερικού διαιρέτη=

Η περίοδος στην έξοδο του εσωτερικού διαιρέτη ονομάζεται κύκλος μηχανής(MC Machine Cycle) και θα είναι:

Tέξοδος εσωτερικού διαιρέτη =0,08333μs=83,33 ns

Δηλαδή ένας κύκλος μηχανής είναι 83,33 ns

Η συχνότητα στην είσοδο του χρονιστή 0(Timer0), δηλαδή στην έξοδο του προγραμματιζόμενου διαιρέτη θα είναι:

fέξοδος προγραμματιζόμενου διαιρέτη=

Η περίοδος στην είσοδο του χρονιστή 0(Timer0), δηλαδή στην έξοδο του προγραμματιζόμενου διαιρέτη θα είναι:

T έξοδος προγραμματιζόμενου διαιρέτη=83,33 ns X 64 =5333ns

Μπορεί να υπολογισθεί και όπως παρακάτω

Τέξοδος προγραμματιζόμενου διαιρέτη

Θα πρέπει να υπολογισθεί η αρχική τιμή στον timer0 έτσι ώστε να συμβαίνει υπερχείλιση του timer0 κάθε 10 ms.

Υπερχείλιση του timer0 σημαίνει μετάβαση από την τιμή FFFF στην τιμή 0000.

O χρόνος που χρειάζεται ο timer0 για να μεταβεί από την αρχική τιμή που θα του δοθεί μέχρι να γίνει υπερχείλιση(και επομένως διακοπή) θα πρέπει να είναι:

10ms=10 000 μs=10 000 000 ns

Υπενθυμίζεται ότι (FFFF)h=(65535)d

Το πλήθος των βημάτων από την αρχική τιμή του Timer0 έως ότου γίνει υπερχείλιση θα είναι: 65536-(Αρχική τιμή του Timer0).

Δηλαδή θα πρέπει:[65536-(Αρχική τιμή του Timer0)]x5333 ns= 10 000 000 ns.

Υπολογίζουμε την αρχική τιμή του timer0.

65536-(Αρχική τιμή του Timer0)=

**(Αρχική τιμή του Timer0)=65536-1875 =63661**

B. διάγραμμα ροής

Αρχή

Αρχή ρουτίνας εξυπηρέτησης διακοπής από τον timer0

Αρχική τιμή στον timer0 ώστε η επόμενη διακοπή να συμβεί μετά από 10 ms

Τέλος ρουτίνας εξυπηρέτησης διακοπής από τον timer0

Συνθήκη πάντα αληθής

ΝΑΙ

ΟΧΙ

Ελάττωση του μετρητή διακοπών

κατά 1

Η τιμή του μετρητή διακοπών(counter1) έγινε 0;

Ο μετρητής διακοπών παίρνει την αρχική του τιμή (=20)

Αλλαγή της κατάστασης όλων των bit της πόρτας Β

ΟΧΙ

ΝΑΙ

Init()

Ενεργοποίηση διακοπών από τον Timer0

Απόδοση αρχικής τιμής στον timer0

Η πόρτα Β γίνεται έξοδος

Η πόρτα Β παίρνει αρχική τιμή 0x00

**Πρόγραμμα.**

#include <main.h>

#byte PORTB =0xF81 // καθορισμός του καταχωρητή δεδομένων της

//πόρτας Β

void init (void); //Δήλωση της ρουτίνας αρχικοποίησης

void timer0\_int(void); //Δήλωση της ρουτίνας διακοπών από τον timer0

int counter1=20; //Δήλωση μεταβλητής για μέτρηση των διακοπών.

//Στη μεταβλητή αυτή δίνεται η αρχική τιμή 20.

// Κύριο πρόγραμμα

void main() { // Ανοίγει η αγκύλη της main

init(); // Κλήση της ρουτίνας των αρχικών ρυθμίσεων

while (TRUE){

} // το κύριο πρόγραμμα δεν κάνει τίποτα. Εκτελεί έναν ατέρμονα βρόχο

} // κλείνει η αγκύλη του main

// Αρχή ρουτίνας εξυπηρέτησης της διακοπής…………………………………………..

#INT\_TIMER0 // Οδηγία ότι η επόμενη ρουτίνα είναι η ρουτίνα εξυπηρέτησης της

// από τον Timer0

void timer0\_int(void){

set\_timer0(63661); //αρχική τιμή του timer0 ώστε η επόμενη διακοπή να συμβεί σε

// χρόνο ίσο με 10 ms

counter1--;

if (counter1==0){

counter1=20;

PORTB=PORTB^0b11111111; // με την λογική πράξη του αποκλειστικού

//ή (Exclusive OR) ανάμεσα στην PORTB και

//την τιμή 11111111 αλλάζουμε την κατάσταση

//όλων των bit της PORTB

// Κάθε 20 διακοπές αλλάζουν όλα τα bit της πόρτας Β

} // κλείνει η αγκύλη του if

} // κλείνει η αγκύλη της ρουτίνας εξυπηρέτησης της

// διακοπής

// Τέλος ρουτίνας εξυπηρέτησης της διακοπής………………………………………

// Αρχή ρουτίνας αρχικών ρυθμίσεων…………………………………………………

void init (void) {

SETUP\_TIMER\_0(T0\_INTERNAL | T0\_DIV\_64 );// Ρύθμιση του προγραμματιζόμενου

// διαιρέτη(prescaler) στην τιμή

set\_timer0(63661); // Αρχική τιμή του timer0

// ώστε να συμβαίνουν διακοπές

// κάθε 10 ms

enable\_interrupts(INT\_TIMER0); // Ενεργοποίηση της // διακοπής από τον timer0

enable\_interrupts(GLOBAL); // Ενεργοποίηση του γενικού // διακόπτη των διακοπών

set\_tris\_b(0x00); //Η πόρτα Β γίνεται έξοδος

PORTB=0x00; //Αρχική τιμή 0 στην πόρτα Β

}

// Τέλος ρουτίνας αρχικών ρυθμίσεων………………………………………………….

**Ερωτήσεις:**

(Στις ερωτήσεις θεωρείστε ότι χρησιμοποιείται ο timer 0 σε λειτουργία 16 bit)

1. Να γραφεί η εντολή με την οποία ο prescaler(προγραμματιζόμενος διαιρέτης συχνότητας τοποθετείται στην τιμή .

2. Να γραφεί η εντολή με την οποία μετά από μια διακοπή ο timer τοποθετείται στην τιμή 32 000.

3. Αν ο προγραμματιζόμενος διαιρέτης(prescaler) είναι στην τιμή και το timer ξεκινάει από την τιμή 0 ανά πόσο χρονικό διάστημα θα συμβαίνουν διακοπές;

4. Αν ο προγραμματιζόμενος διαιρέτης είναι στην τιμή και το timer ξεκινάει από την τιμή 32768 ανά πόσο χρονικό διάστημα θα συμβαίνουν διακοπές;

5. Αν δεν χρησιμοποιείται προγραμματιζόμενος διαιρέτης(prescaler) και ο timer ξεκινάει από το 0 ανά πόσο χρονικό διάστημα(σε ms) θα γίνονται διακοπές;

6. Να γραφεί η εντολή με την οποία δεν χρησιμοποιείται ο prescaler(προγραμματιζόμενος διαιρέτης).

*Απάντηση: SETUP\_TIMER\_0(T0\_INTERNAL | T0\_DIV\_1 )*

7. Αν δεν χρησιμοποιείται prescaler να υπολογιστεί η αρχική τιμή στην οποία πρέπει να τοποθετείται ο timer0 έτσι ώστε να συμβαίνουν διακοπές κάθε 1 ms. Να γραφεί η εντολή με την οποία γίνεται αυτό.

8. Αν χρησιμοποιείται prescaler με τιμή να υπολογιστεί η αρχική τιμή στην οποία πρέπει να τοποθετείται ο timer0 έτσι ώστε να συμβαίνουν διακοπές κάθε 1 ms.

Α) Να γραφεί η εντολή με την οποία τοποθετείται ο timer0 στην τιμή αυτή.

Β) Να γραφεί η εντολή με την οποία ο prescaler τοποθετείται στην τιμή

9. Να γραφεί η εντολή με την οποία αλλάζουμε την κατάσταση μόνο του bit 2 της πόρτας Β.

Υπόδειξη: Κάνουμε Exclusive OR(στη γλώσσα C, ^ ) με μια συγκεκριμένη τιμή.

10. Να γραφεί η εντολή με την οποία μηδενίζουμε το bit 2 της πόρτας Β

Υπόδειξη: Κάνουμε AND (στη γλώσσα C, & ) με μια συγκεκριμένη τιμή.

11. Να γραφεί ρουτίνα διακοπής από τον timer0 με την οποία ο timer0 μετά από κάθε διακοπή τοποθετείται στην τιμή 32156 και κάθε (100)d  διακοπές αλλάζει η κατάσταση του ακροδέκτη 0 της πόρτας Β.

12. Αν ο prescaler είναι στην τιμή να υπολογισθεί η αρχική τιμή που πρέπει να δίνεται στον timer0 σε κάθε διακοπή έτσι ώστε να συμβαίνουν διακοπές κάθε 1 δευτερόλεπτο.