

Διασύνδεση Μνήμης

Σχεδιαστε το κυκλωμα που συνδεει έναν επεξεργαστη cpu με 8bit διαυλο με τεσσερις μνημες 16bit διαυλο έτσι ώστε να χρησιμοποιείται ο μισος συνολικος χώρος στο κύκλωμα.

Με άλλα λόγια :

1) Μικροϋπολογιστής με 16-bit δίαυλο διευθύνσεων και 8-bit δίαυλο δεδομένων, στον οποίο έπρεπε να συνδέσουμε τέσσερις μνήμες που συνολικά να πιάνανε το 50% της μέγιστης διαθέσιμης μνήμης.

Με άλλα λόγια :

1. 4 μνήμες 16 bit address bus να το συνδέσω με τυπικό μικροϋπολογιστή data bus 8 bits. Συνολική χωρητικότητα μνημών

Με άλλα λόγια :

1. 4 μνήμες 16 bit address bus να το συνδέσω με τυπικό μικροϋπολογιστή data bus 8 bits. Συνολική χωρητικότητα μνημών στο 50% 2.

Λύση :

Ο μικροϋπολογιστής μπορεί να διευθυνσιοδοτήσει $2^{16} = 64\text{KB}$ μνήμης . Οπότε θα χρειαστούμε τέσσερις μνήμες συνολικής χωρητικότητας $64/2 = 32\text{KB}$. Οπότε θα έχουμε μνήμες χωρητικότητας $32/4 = 8\text{KB}$ η καθεμία.

Πίνακας χαρτογράφησης

Μνήμη	Μέγεθος	Πεδίο Διεύθυνσης
RAM1	8KB	0000 0000 0000 0000 0001 1111 1111 1111
RAM2	8KB	0010 0000 0000 0000 0011 1111 1111 1111
RAM3	8KB	0100 0000 0000 0000 0101 1111 1111 1111
RAM4	8KB	0110 0000 0000 0000 0111 1111 1111 1111

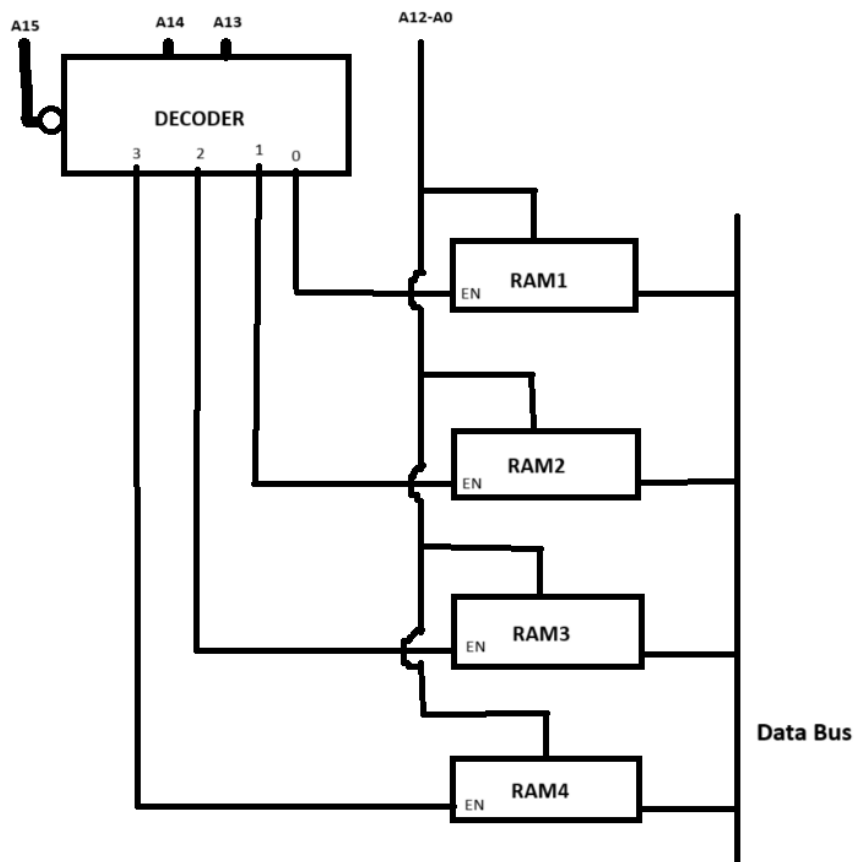
Αφου θα χρησιμοποιήσουμε την μισή μνήμη που μπορεί να διευθυνσιοδοτήσει ο μικροϋπολογιστής δεν θα χρειαστούμε το πιο σημαντικό ψηφίο.

Συγκεκριμένα μπορούμε να χρησιμοποιήσουμε τα 2^ο και 3^ο πιο σημαντικά ψηφία για να ξεχωρίσουμε τις μνήμες.

Πίνακας αληθείας σημάτων επιλογής

	A ₁₄	A ₁₃	Μνήμη
1	0	0	RAM1
2	0	1	RAM2
3	1	0	RAM3
4	1	1	RAM4

Κύκλωμα Λογικής



PIO

2) Στο Z80 PIO να γράψεις τις λέξεις που χρειάζεται για να τεθεί σε κατάσταση λειτουργίας ένα (Mode 1) και να εξηγήσεις τι κάνει το κάθε bit.

Με άλλα λόγια :

Pio mode 1 να βρω όλα τα words και να εξηγήσω κάθε bit

Με άλλα λόγια :

Pio mode 1 να βρω όλα τα words και να εξηγήσω κάθε bit

Λύση :

Για να θεσω το Z80 PIO σε Mode 1 θα χρειαστώ 3 λέξεις.

1. Λέξη ελέγχου κατάστασης : 01XX1111

Θέτω τα 2 πιο σημαντικά bit σε 01 ώστε να τεθεί σε mode 1 το PIO.

Τα 4 λιγότερο σημαντικά bit υποδεικνύουν τη λέξη κατάστασης που θα οριστεί.

2. Λέξη ελέγχου διακοπής : XXXX0111

Επειδή δεν αναφέρεται κάτι για interrupt control , δεν έχει σημασία τι θα βαλουμε στο πιο σημαντικό bit οπότε το αφήνουμε X

Τα επόμενα 3 πιο σημαντικά bit έχουν νόημα μόνο στο mode 3 του PIO.

Τα 4 λιγότερο σημαντικά bit υποδεικνύουν τη λέξη ελέγχου διακοπής.

3. Λέξη ελέγχου κάλυψης : XXXXXXXX

Τα bit καθορίζουν ποιο ακροδεκτες θα χρησιμοποιηθούν για να γίνει το interrupt , μπορούμε να θέσουμε όποιον θέλουμε.

Παρόμοια Άσκηση :

Προγραμματισμός PIO σε mode 1 και μεταβαση σε mode 2. Να γραφουν όλες οι λέξεις

Για να αλλάξει σε mode 2 θα χρειαστεί να εισάγουμε 1 λέξη , την λέξη ελέγχου κατάστασης όπου τα πρώτα δύο bit δείχνουν ότι θέσαμε το PIO σε mode 2 και τα τέσσερα τελευταία bit δείχνουν ότι η λέξη είναι λέξη ελέγχου κατάστασης.

Οπότε τα bit θα είναι 10XX1111

Prescaler

4ο θέμα Σχεδιάστε και περιγράψτε τη λειτουργία του prescaler στον μικροελεγκτή ATiny 2313

Με άλλα λόγια :

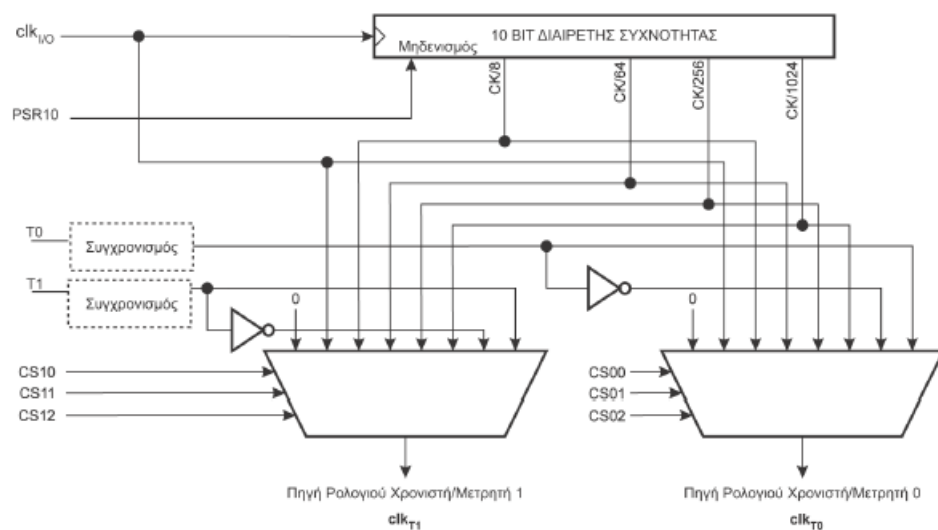
4 σχεδιασμός και εξήγηση prescaler Atiny

Λύση :

Ένας prescaler είναι ένας διαιρέτης συχνότητας που μειώνει την συχνότητα του εσωτερικού ρολογιού σε έναν μικροελεγκτή.

Για μια συχνότητα $F(\text{CLOCK}) = 4\text{MHz}$, αν η τιμή του PRESCALER είναι FA0_{16} , τότε η συχνότητα του εσωτερικού ρολογιού θα είναι $F(\text{Internal Clock}) = 4\text{MHz} / 4000 = 1\text{kHz}$.
(αφού $\text{FA0}_{16} = 4000_{10}$)

Prescaler: ATtiny2313

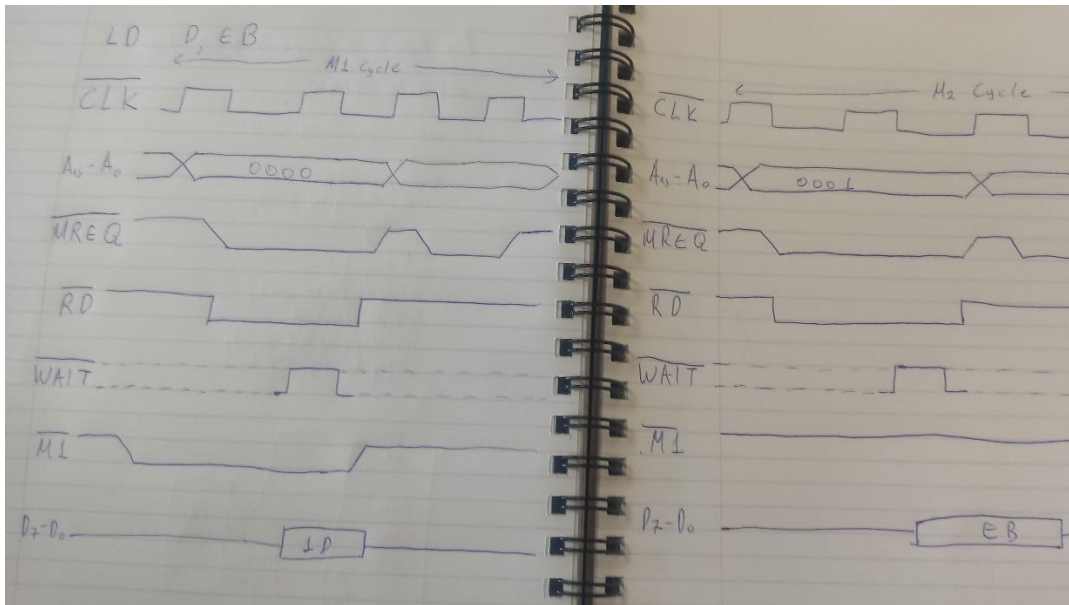


Σήμα εντολής

Σχεδιάστε και εξηγήστε το σήμα της εντολής LD D, EB

Λύση :

2 κύκλοι

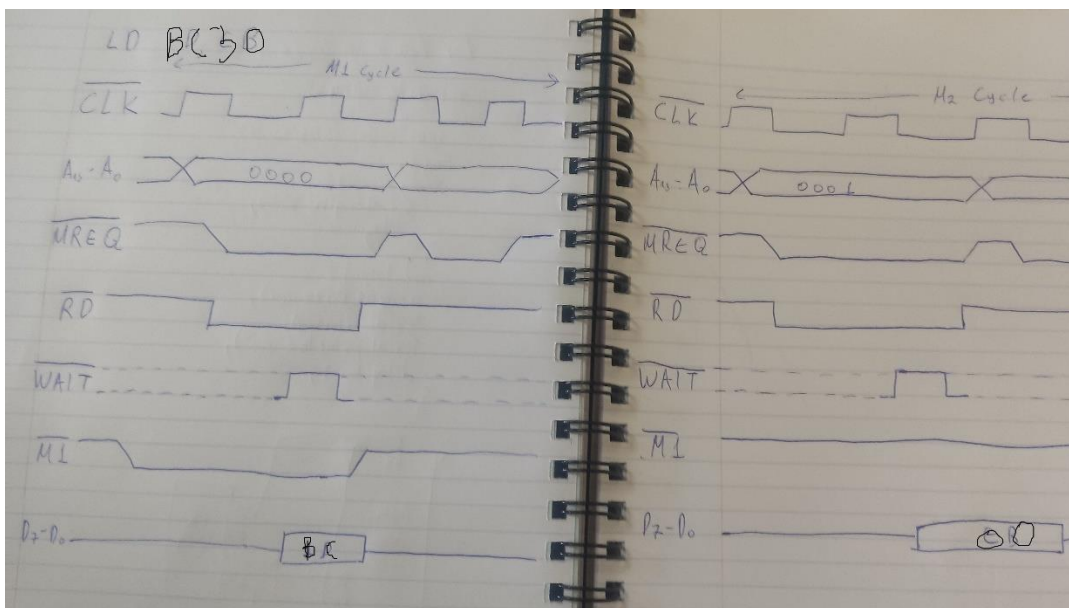


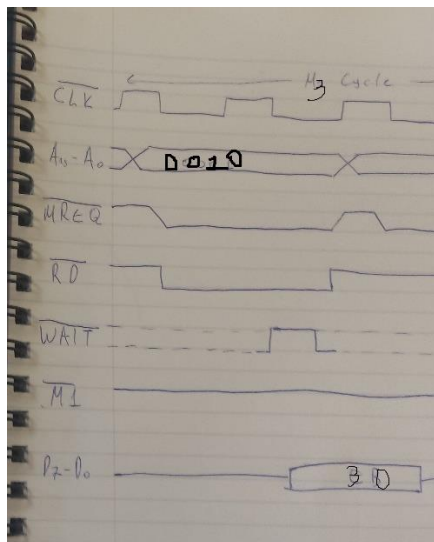
Παρόμοια Άσκηση :

Na σχεδιάσω και να εξηγήσω εντολή LD BC,30h

Λύση :

3 κύκλοι



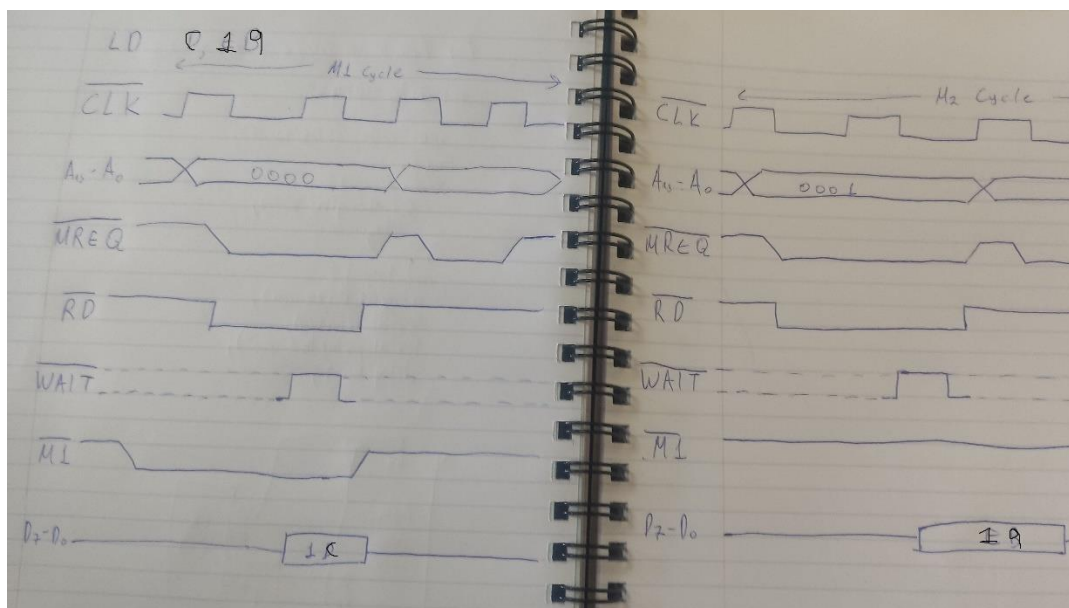


Παρόμοια Άσκηση :

1 χρονοδιάγραμμα LD C , 19H

Λύση :

2 κύκλοι

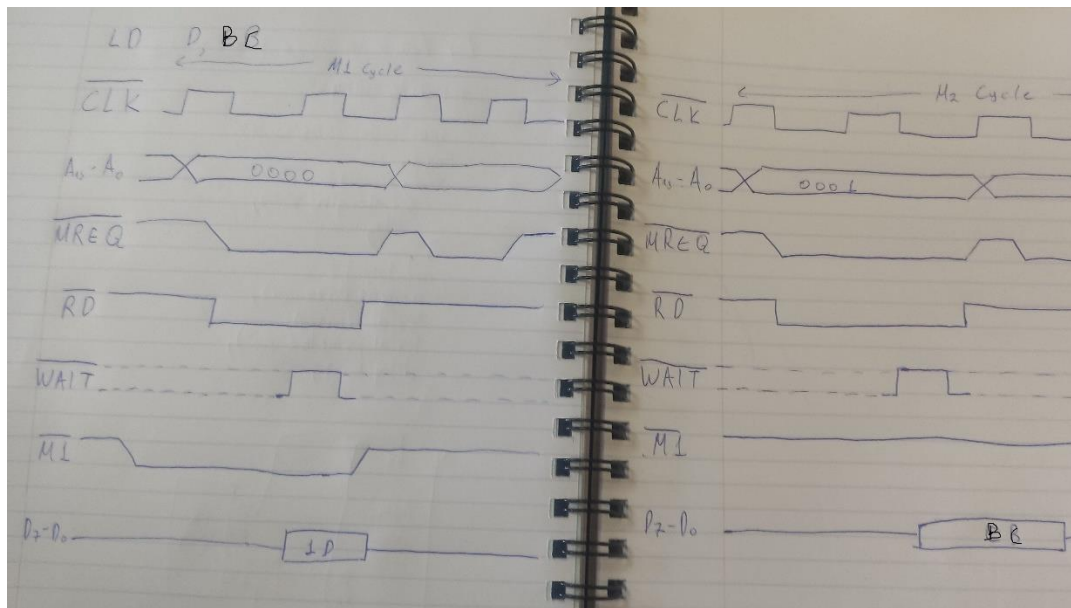


Παρόμοια Άσκηση :

3) Να σχεδιαστούν και να εξηγηθούν τα διαγράμματα χρονισμού στο Z80 για την εντολή LD D, 0xBC.

Λύση :

2 κύκλοι



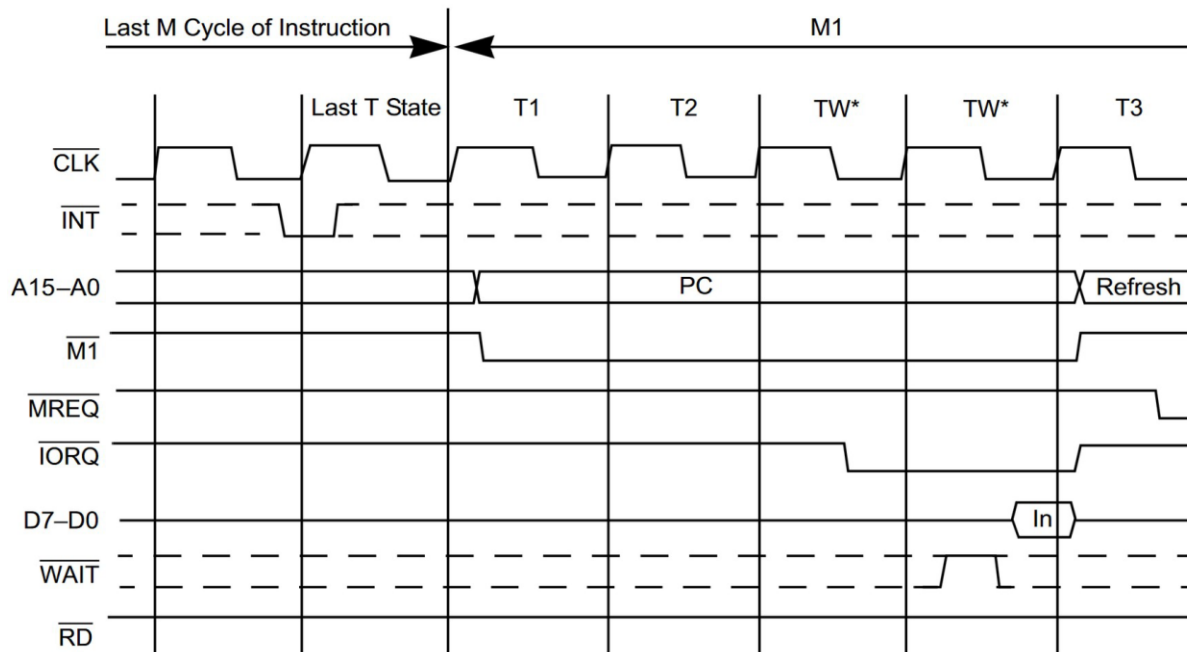
Interrupt

Σχεδιασμός κι επεξήγηση χρονοδιαγράμματος για interrupt

Με άλλα λόγια :

2 χρονοδιάγραμμα σήμα διακοπή INT~

Λύση :

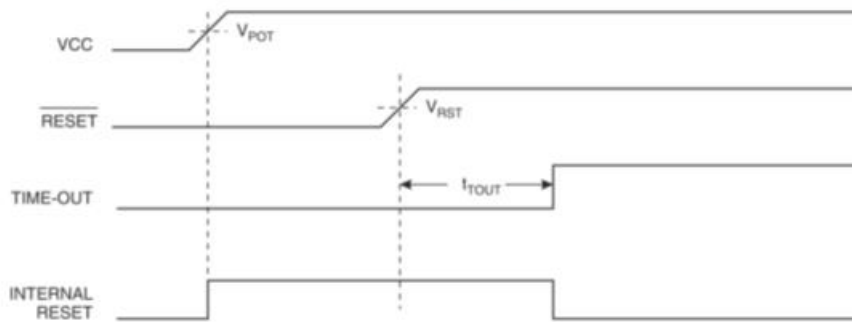


reset

Σχεδιασμός διαγράμματος κι επεξήγηση reset συνδεδεμένο με αντίσταση στην τροφοδοσία και με πυκνωτή στο εδαφος

Λύση :

Reset που καθυστερεί σε σχέση με την τροφοδοσία



- Το Reset- είναι συνδεδεμένο μέσω μιας αντίστασης στην τάση τροφοδοσίας και μέσω ενός πυκνωτή στη γη.
- Αυτό συμβαίνει προκειμένου να διασφαλίζεται ότι η τάση τροφοδοσίας δεν φτάνει απότομα στην τελική τιμή της.

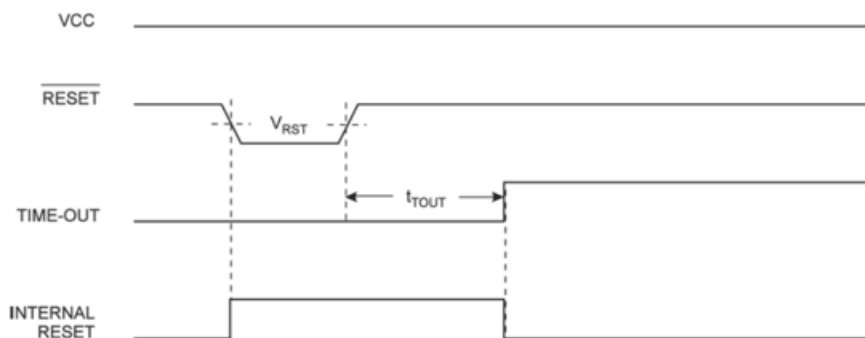
Παρόμοιο θέμα :

3ο θέμα να περιγράψετε πως κάνουμε επανεκκίνηση σε έναν μικροελεγκτή με εξωτερική λογική

Με άλλα λόγια :

3 εξωτερική λογική για reset μικροελεγκτή

Λύση :



Προγραμματισμός Arduino

Ultrasonic sensor module που υπολογίζει την απόσταση από αντικείμενα μπροστά του.

4) Έστω ένα Ultrasonic Sensor Module για τη μέτρηση αποστάσεων, στο οποίο γράφεις κώδικα για Arduino Uno και κάνεις ένα γρήγορο διάγραμμα του κυκλώματος

Πρόγραμμα arduino να υπολογίσω την απόσταση που λαμβάνουν αισθητήρες

Πρόγραμμα arduino να υπολογίσω την απόσταση που λαμβάνουν αισθητήρες

Λύση :

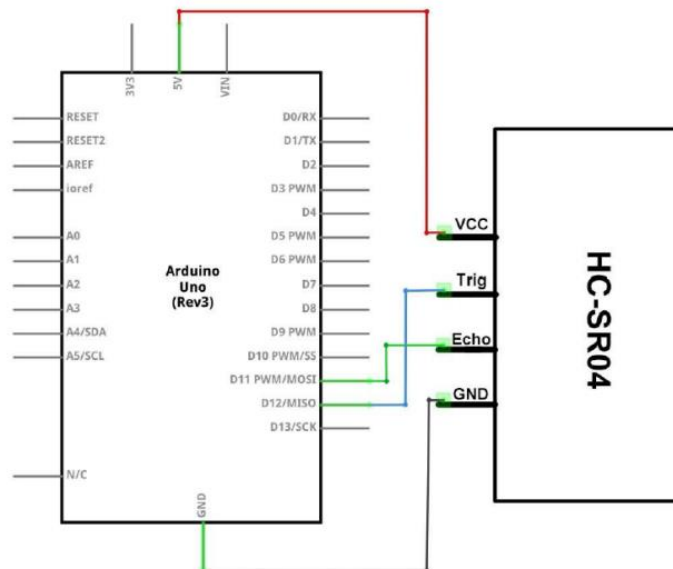
```
#include "SR04.h"
```

```
SR04 sr04 = SR04(11,12);
```

```
void setup () {  
    Serial.begin( 9600 );  
}
```

```
long distance;
```

```
void loop() {  
    distance = sr04.Distance();  
    Serial.print( distance );  
    Serial.println( "cm" );  
}
```



Παρόμοια Άσκηση

Προγραμματισμός Arduino Uno για ένα απλό διακοπτή (αναγνώριση φωνής)

Λύση :

[δεν ξέρω]

Stack pointer

5. Να εξηγήσω πως μεταβάλλεται ο stack pointer με εντολές Atiny (rcall push ldi)

Λύση :

Ο rcall μειώνει τον δείκτη στοίβας κατά 2

Το push μειώνει τον δείκτη στοίβας κατά 1

Ο LDI δεν επηρεάζει τον δείκτη στοίβας

Άρα ο δείκτης στοίβας θα είναι 3 μονάδες κάτω από την αρχική του τιμή.

Παρόμοια άσκηση :

5) Μικρό πρόγραμμα με τη στοίβα σε μικροελεγκτές της οικογένειας ATtiny, στο οποίο γίνονται δύο (2) κλήσεις ρουτίνας, ένα (1) LD και ένα (1) PUSH και να γράψεις πώς επηρεάζει τη στοίβα.

Λύση :

Ο LD δεν επηρεάζει τον δείκτη στοίβας

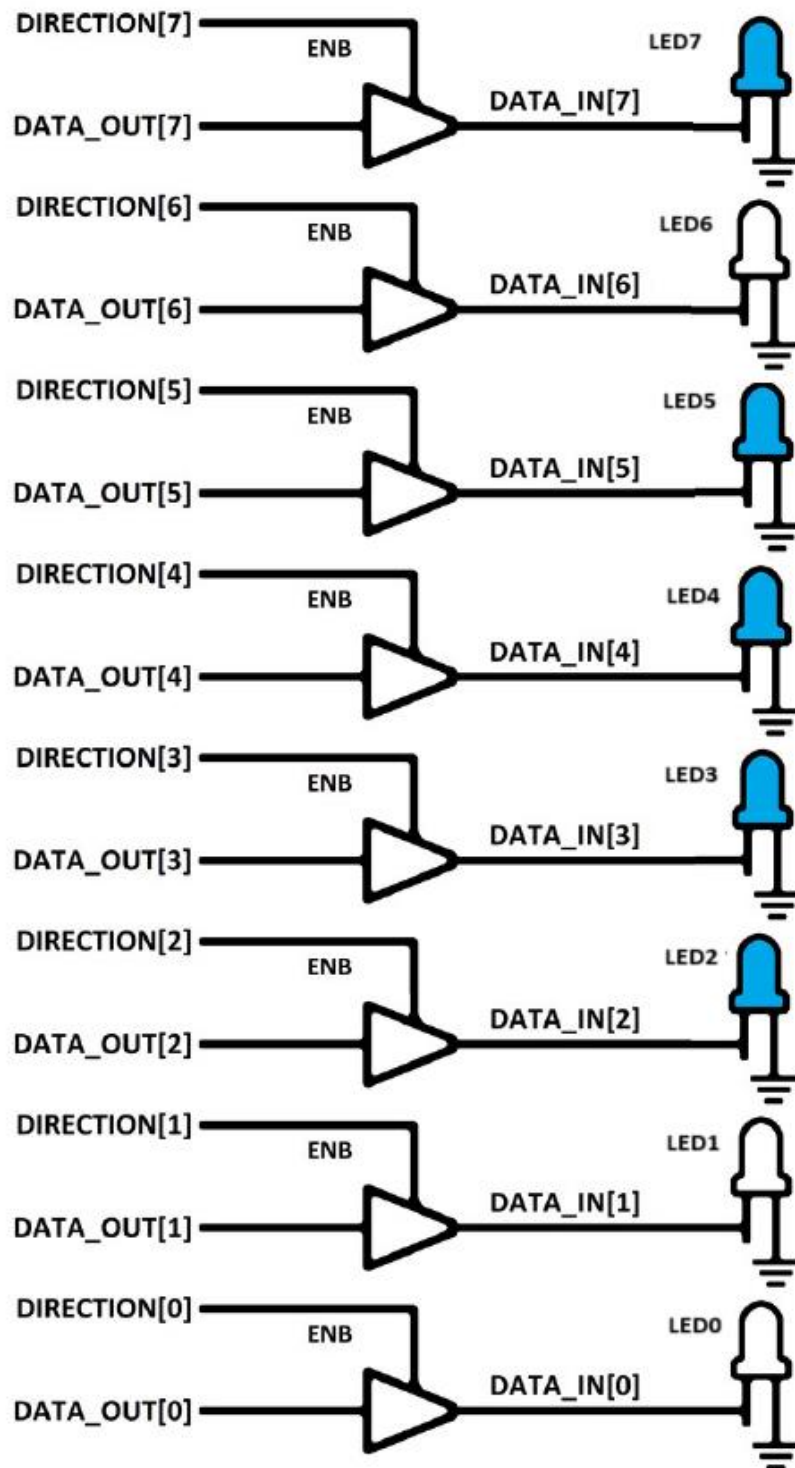
Το Push μειώνει τον δείκτη στοίβας κατά 1

Άρα ο δείκτης στοίβας θα είναι 1 μονάδα κάτω από την αρχική του τιμή.

Παράλληλη Θύρα

Δίνονται μικροϋπολογιστικό σύστημα με παράλληλη θύρα των 8bit με συνδεδεμένα 8 LED όπως φαίνεται στο παρακάτω σχήμα. Ο register A περιέχει την τιμή 0x32 και το DATA_OUT την τιμή 0xBC.

Να γραφτεί πρόγραμμα σε Assembly κώδικα που να ανάβει μόνο τα περιττά LED .



Λύση :

Βάζουμε την τιμή που μας δίνει μέσα στον καταχωρητή A , κάνουμε μια μάσκα OR για να ανάψουμε το LED1 και μετά κάνουμε μια μάσκα AND για να σβήσουμε το LED4 και το LED2

IN A , (DATA_IN)	; A	=	10111100 = 0xBC
	;		00000010 = 0x02
OR 0x02	; A OR 0x02	=	10111110
	; A	=	10111110
AND 0xEB	;		11101011 = 0xEB
	; A AND 0xEB	=	10101010 = 0xAA
OUT (DATA_OUT) , A	;		