

Θεόδωρος Μπάρκας ΑΜ:2016030050

1.Sampling / Polling

Memory map

0x00	lastInput	
0x01	stableSignalCounter	0x00 → disable else Counting from 1 to 11 (10 times). Γινετε Enable μονο οταν παρατηρουμε αλλαγη στο SPDT switch*

Προσέγγιση Προβλήματος

Αρχικά εκτελούμε δειγματοληψία για να δούμε αν η τιμή αλλάζει συγκριτικά με την προηγούμενη τιμή του διακόπτη .Η περίοδος δειγματοληψίας δίνετε από το TIMER0_OVF (Η παραμετροποίηση του οποίου αναλύετε παρακάτω).

Έστω ότι Abit είναι η είσοδος που λαμβάνουμε από το PA0 και A'bit η είσοδος που λαμβάνουμε από το PA2.

Καταλαβαίνουμε ότι έχει πατηθεί ο διακόπτης κατά τη δειγματοληψία όταν ισχύει ένα ή περισσότερα από τα παρακάτω :

- lastInputBit != ~A'bit
- lastInputBit != Abit

Σε αυτήν την περίπτωση θέτουμε την τιμή μνήμης stableSignalCounter στο 0x01 που σημαίνει ότι ψάχνουμε για τη σταθερή είσοδο η οποία λογικά θα έχει κατασταλάξει 10ms μετά. Επομένως μετράμε 10 timer0_counterOVF(διάρκειας 1ms ο καθένας). Κατόπιν αν έχουμε κατασταλάξει:

Θέτουμε τον stableSignalCounter=0x00(Disable).

Κωδικας TIMER0_OVF (INTERRUPT HANDLER)

```
//INTERRUPTS
ISR(TIMER0_OVF_vect){
    TCNT0= (unsigned char ) 100;
    unsigned char *lastInputMem = (unsigned char *)0x60;
    unsigned char *stableSignalCounterMem = (unsigned char *)0x61;
    unsigned char volatile bitA,nbitA;
    //Getting lastInput and stableSignalCounter from Memory
    unsigned char lastInput = *lastInputMem;
    unsigned char stableSignalCounter = *stableSignalCounterMem;
    //Getting Current Inputs Stable or Unstable
    //Abit
    bitA=PINA>>PINA0;
    bitA=bitA & 0x01;
    //A'bit
    nbitA=PINA>>PINA1;
    nbitA=nbitA & 0x01;
    //LOGIC
    if( stableSignalCounter==0x00 && (lastInput != bitA || lastInput != (~nbitA & 0x01)) ){
        stableSignalCounter++;
    }
    else if(stableSignalCounter==0x0A){
        stableSignalCounter = 0x00;
        *lastInputMem=bitA;
        PORTA = bitA<<PA2;
    }
    else if(stableSignalCounter>0x00){
        stableSignalCounter++;
    }
    *stableSignalCounterMem=stableSignalCounter;
}
```

TIMER0_OVF (και Initialization)

Οι παράμετροι (καταχωρητές που σχετίζονται) του timer0_OVF είχαν βρεθεί από το εργαστήριο 1 αλλά προκύπτουν εύκολα και από το προηγούμενο εργαστήριο όπου ο timer0_OVF ήταν στα 4ms και επόμενος άπλα κάνοντας τον prescaler από Cs2cs1cs0=100 → 011 περνούμε τον timer0_OVF κάθε 1ms .

- Cs2cs1cs0=011 , normal mode → TCCR=0x03;
- TCNT0=100(decimal) → TCNT0=100;
- TOIE0=1 → TIMSK=(1<<TOIE0);

2.Με εξωτερικά Interrupts

Σε αυτή τη πρόπτωση θα περνούμε Interrupt από έξω($INT0 \rightarrow A$, $INT1 \rightarrow A'$). Ξέρουμε ότι έστω παρακολουθώντας το A και στην περίπτωση που αυτό πάει από 1 σε 0 θα κάνει Bounce μερικές φορές, δίνοντας πολλαπλά Interrupts.

Αρκεί επομένως να αναγνωρίσουμε πιο σήμα πέφτει από 1 σε 0 (Falling Edge). Το σήμα το οποίο θα το κάνει αυτό μετά από μερικά Bounces θα καταλήξει στην τιμή 0.

Επομένως κάθε φορά που θα άρχετε το Falling Edge και ανάλογα το Interrupt το οποίο πυροδοτείτε (το $INT0$ -PD2(SET) ή $INT1$ -PD3(RESET)) θα καταλαβαίνουμε εύκολα την έξοδο.

Αυτός ο τρόπος λειτουργίας κατά κάποιο τρόπο αντιγράφει τη λειτουργία του μανταλωτή NAND-NAND(και λέγοντας αντιγράφει εννοούμε λειτουργικά).

Πιο συγκεκριμένα έχει την ίδια συμπεριφορά με το Truth Table του μανταλωτή NAND-NAND.

Επίσης αναφέρετε ότι η έξοδος για την υλοποίηση με τα Interrupts θα “Βγαίνει” στην έξοδο PD4.

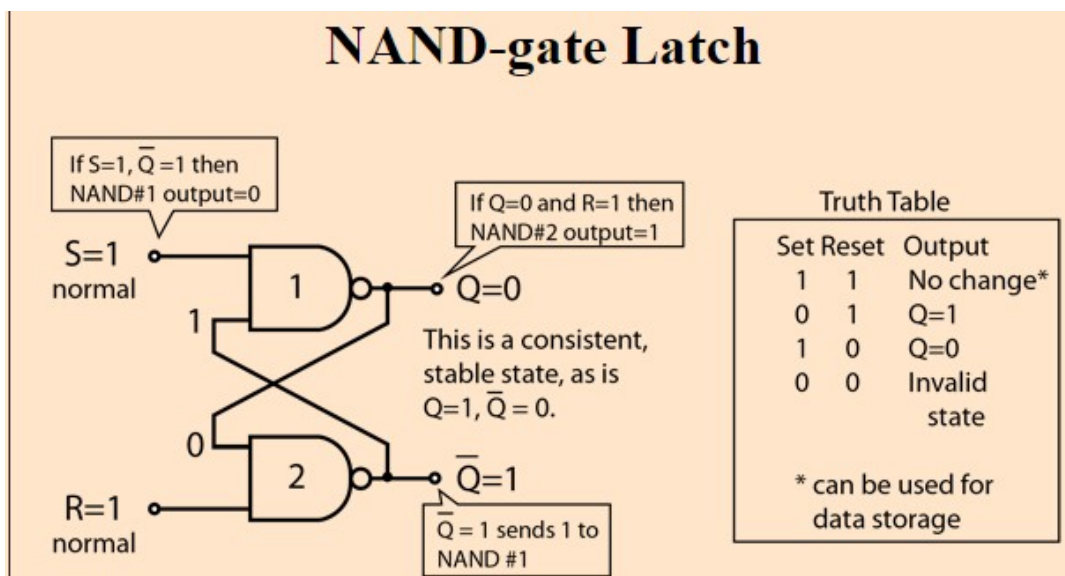


Figure 1: Πηγή: <http://hyperphysics.phy-astr.gsu.edu/hbase/Electronic/nandlatch.html>

Επομένως χρειάζεται να προβούμε σε ενέργειες μόνο κατά το Falling Edge του εκάστοτε Interrupt.

```
void EXTERNAL_INTERRUPTS_Init(void){
    unsigned char * QMem =(unsigned char *)0x62;
    *QMem=0x00;
    GICR|=(1<<INT0)|(1<<INT1);
    MCUCR |= 0b00001010; //isc11,isc10 and isc01,isc00 both set to -> 01 (Any logical change will trigger them)
}
```

Επομένως ο κώδικας εντός των συναρτήσεων (INTERRUPT HANDLER) είναι ο εξής :

```
ISR(INT0_vect){//PD2 is INT0->SET (FALLING EDGE MEANS PD4=0)
    PORTD=0<<PD4;
}

ISR(INT1_vect){//PD3 is INT1->RESET (FALLING EDGE MEANS PD4=1)
    PORTD|=1<<PD4;
}
```

[illegible]

- Σε περίπτωση που θέσουμε το δεύτερο stimFile: **2a_bouncingStart_INPUT=1.stim** που θέτει $A=1$ και $A'=0$ και στην δειγματοληψία Debouncer άλλα και στο τελικό αποτέλεσμα περιμένουμε να δούμε την αλλαγή άλλα μόνο μετά από 10 ms.

Όντως χτυπάμε πρώτα το break point 1 (εντοπισμός μέσω Polling/Sampling) μια φορά και κατόπιν χτυπάμε 9 φορές τον 3ο break point και κατόπιν μια το δεύτερο.

Name	Address	Value	Bits
I/O PINA	0x39	0x1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O DDRA	0x3A	0x7	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O PORTA	0x3E	0x1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

- Ύστερα εκτελούμε το 3ο stimFile: **3a_bouncingCase1.stim** που θέτει $A=1$ και $A'=1$ στην δειγματοληψία Debouncer και στο τελικό σήμα $A=0$ $A'=1$.

Όντως παρατηρούμε ακριβώς ίδια συμπεριφορά με το πάνω (ως αναφορά τους Break Point) καθώς και την αλλαγή εισόδου μετά από 10ms σε 0.

Name	Address	Value	Bits
I/O PINA	0x39	0x1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O DDRA	0x3A	0x7	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O PORTA	0x3E	0x1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

- Έπειτα ανοίγουμε το 4ο stimFile: **4a_bouncingCase2-FAULT.stim** το οποίο άπλα δίνει ίδιες τιμές εισόδου με τις προηγούμενη είσοδο $A=0$ και $A'=1$ και όπως είναι αναμενόμενο δεν παρατηρούμε καμία αλλαγή.

Name	Address	Value	Bits
I/O PINA	0x39	0x1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O DDRA	0x3A	0x7	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O PORTA	0x3E	0x1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

- Ύστερα εκτελούμε το 5ο stimFile: **5a_bouncingCase3=1.stim** που θέτει $A=0$ και $A'=0$ στην δειγματοληψία Debouncer και στο τελικό σήμα $A=0$ $A'=1$.

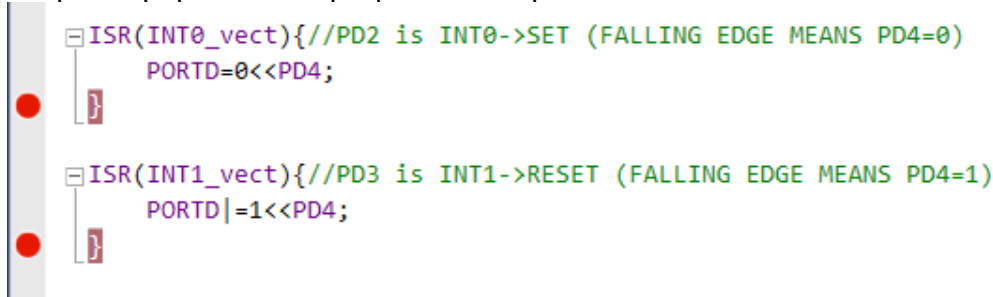
σε αυτήν την περίπτωση μπαίνουμε στο πρώτο if και παρατηρούμε παρόμοια συμπεριφορά με το 2ο και το 3ο stimFile και τελικά έχουμε:

Name	Address	Value	Bits
I/O PINA	0x39	0x1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O DDRA	0x3A	0x7	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O PORTA	0x3E	0x1	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

External Interrupts

Για την δοκιμή του External Interrupt φτιάξαμε τα stimFile: **b1_BOUNCES_0_to_1_and_1_to_0.stim**
Σε αυτό το σημείο παρατηρούμε για την κατανόηση και σωστή διεκπεραίωση του Testing:

- την πυροδότηση των Interrupts μέσω break points .



- Τα stimFile που εκτελούνται (από τη καρτέλα output και στο dropdown menu την επιλογή “fileStimuliProvider”), το stimFile εμπεριέχει σχόλια με το τι αναμένουμε να δούμε από την κάθε αλλαγή που δεχόμαστε από το stimFile (Τιμές που περνούμε κατά τη διάρκεια του Bouncing), τα σχόλια φαίνονται στο runtime στο fileStimuliProvider .

```
1 //SWITCH from 0 to 1 RESET BOUNCES
2 //-----//-----COMMENTS-----TRIGGER-----C_EDGE-----SWITCH A
3 PIND = 0b0001000 //INT1=RESET=1 INT0=SET=0 ->Interrupt NOT TRIGGER -RISING EDGE -INITIALIZATION ->0
4 #1000
5
6 PIND = 0b0001000 //INT1=RESET=1 INT0=SET=0 ->Interrupt NOT TRIGGER -SAME AS BEFORE -no changes
7 #1000
8
9 PIND = 0b0001100 //INT1=RESET=1 INT0=SET=1 ->Interrupt NOT TRIGGER -RISING EDGE "11" -no changes
10 #1000
11
12 PIND = 0b0000100 //INT1=RESET=0 INT0=SET=1 ->Interrupt TRIGGER -FALLING EDGE RESET ->1
13 #1000
14
15 PIND = 0b0001100 //INT1=RESET=1 INT0=SET=1 ->Interrupt NOT TRIGGER -RISING EDGE "11" -no changes
16 #1000
17
18 PIND = 0b0000100 //INT1=RESET=0 INT0=SET=1 ->Interrupt TRIGGER -FALLING EDGE RESET -no changes"same as before" ->1
19 #1000
20
21 PIND = 0b0001100 //INT1=RESET=1 INT0=SET=1 ->Interrupt NOT TRIGGER -RISING EDGE "11" -no changes
22 #1000
23
24 PIND = 0b0000100 //INT1=RESET=0 INT0=SET=1 ->Interrupt TRIGGER -FALLING EDGE RESET -no changes"same as before" ->1
25 #1000
26
27 PIND = 0b0000100 //INT1=RESET=0 INT0=SET=1 ->Interrupt NOT TRIGGER -SAME AS BEFORE -no changes
28 #1000
29
30 //SWITCH from 1 to 0 SET BOUNCES
31 #50000
32 //-----//-----COMMENTS-----TRIGGER-----C_EDGE-----SWITCH A
33 PIND = 0b0000100 //INT1=RESET=0 INT0=SET=1 ->Interrupt NOT TRIGGER -SAME AS BEFORE -no changes ->1
34 #1000
35
```

Εδώ φαίνεται μέρος του stimFile

- Καθώς και την έξοδο PD4 .

Βλέπουμε ότι η συμπεριφορά όταν τρέχουμε την προσομοίωση είναι ίδια με αυτή που περιμένουμε να δούμε και που εξηγείτε από τα σχολεία του stimFile.