Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

6ο Εξάμηνο

Συστήματα Μικροϋπολογιστών

4η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ

[redacted]

Χατζή Ήβη

Άσκηση 1

```
.include “m16def.inc”
reset:
ldi r24,low(RAMEND)  ; Initialize stack pointer
out SPL,r24
ldi r24,high(RAMEND) ;
out SPH,r24
ser r24              ; Initialize PORTA for output
out DDRA,r24
clr r24
out DDRB,r24         ; Initialize PORTB for input

main:
ldi r26,01           ; Start from LED0
rcall move_left
nop
rcall move_right
rjmp main

move_left:
in r24,PINB          ; Input PB0
andi r24,01          ; Keep LSB
cpi r24,01           ; If PB0 = 1 don’t move
breq move_left
out PORTA,r26
cpi r26,80           ; Check if reached MSB
brcc end_left        ; If MSB return
lsl r26              ; Else shift left and continue
rjmp move_left

end_left: ret

move_right:
in r24,PINB          ; Input PB0
andi r24,01          ; Keep LSB
cpi r24,01           ; If PB0 = 1 don’t move
breq move_right
out PORTA,r26
cpi r26,01           ; Check if reached LSB
breq end_right       ; If LSB return
lsr r26              ; Else shift right and continue
rjmp move_right

end_right: ret
```

Άσκηση 2

```c
/*
 * ex2.c
 */

#include <avr/io.h>


int main(void)
{
	char input,A,B,C,D,F0,F1;
	DDRB=0xFF;            // initialise PORTB as output
	DDRA=0x00;           // initialise PORTA as input
	while(1)
	{
		input = PINA & 0x0F; //input bits 0-3 from PORTA
		A = input & 0x01;                  // A is bit 0
		B = (input & 0x02) >> 1;    // B is bit 1, shift once
		C = (input & 0x04) >> 2;    // C is bit 2, shift twice
		D = (input & 0x08) >> 3;    // D is bit 3, shift thrice

		F0 = ~((A & B & ~C)|(C & D));      // calculate F0
		F0 = F0 & 0x01;             // isolate bit 0
		F1 = ((A | B) & (C | D));   // calculate F1
		F1 = (F1 << 1) & 0x02;      // shift F1 once to the left and isolate bit 1

		PORTB = F1 | F0; // output at PORTB
	}
	return 0;
}
```

Άσκηση 3

```
/*
*ex3.c
*/

#include<avr/io.h>

char x,input;

int main(void){

    DDRA = 0xFF;                    // Initialise PORTA as output
    DDRC = 0x00;                   // Initialise PORTC as input
    x = 1;                         // Initialise x for

    while(1){
        input=PINC;
        if(input == 1)              // SW0 is pressed
              if(x == 128) x = 1;   // If 10000000 then 00000001 else
              else x = x << 1;      // Shift left once
        else if(input == 2)         // SW1 is pressed
              if(x == 1) x = 128;   // If 00000001 then 10000000 else
              else x = x >> 1;      // Shift right once
        else if(input == 4)         // SW2 is pressed
              x=128;                // Led7 (MSB)
        else if(input == 8)         // SW3 is pressed
              x = 1;                // Led0 (LSB)
        while(PINC != 0);           // Wait until push-button is unpressed
        PORTA = x;                  // Apply the change of the output
    }
    return 0;
}
```