Eilat Avidan

CPE301 – SPRING 2018

Design Assignment X

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

The student understands that all required components should be submitted in complete for grading of this assignment.

NO	SUBMISSION ITEM	COMPLETED (Y/N)	MARKS (/MAX)
1	COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS		
2.	INITIAL CODE OF TASK 1/A		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E		
4.	SCHEMATICS		
5.	SCREENSHOTS OF EACH TASK OUTPUT		
5.	SCREENSHOT OF EACH DEMO		
6.	VIDEO LINKS OF EACH DEMO		
7.	GOOGLECODE LINK OF THE DA		

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

The components used for this assignment-

- Atmega328P
- Switch
- Breadboard
- Resistors (mainly 100 ohms)
- LED (red)
- Power supply

2. INITIAL/DEVELOPED CODE OF TASK 1/A (assembly)

```
.org 0
       LDI R16, (1<<2)
       SBI DDRB , 2                ;set PB2 as an output
       LDI R17 , 0                 ; using for out
       out PORTB, R17
begin:
       RCALL delay                 ;calling timer to wait 0.25 sec
       EOR R17, R16                ;xor to toggle the LED
       out PORTB, R17
       RJMP begin
delay:
       LDS R29, TCNT1H             ;loading upper bit of counter to R29
       LDS R28, TCNT1L             ;loading lower bit of counter to R28
       LDI R20, 0
       STS TCCR1A, R20             ;using control timer register
       LDI R20, 2                  ;set prescaler of 8
       STS TCCR1B, R20             ;start the clock
       CPI R28,0x08                ;comparing if lower is 0x08
       BRSH body
       RJMP delay
body:
       CPI R29,0x3D                ;comparing if upper bits is 0x3D
       BRSH done
       RJMP delay
done:
       LDI R20,0x00
       STS TCNT1H,R20              ;resetting the counter to 0 for next round
       LDI R20,0x00
       STS TCNT1L,R20              ;resetting the counter to 0 for next round
       RET
```

3. INITIAL/DEVELOPED CODE OF TASK 1/B (C)

```
#include <avr/io.h>
#define  F_CPU 5000000UL
#include <util/delay.h>


int main()
{
    DDRB = 0x04;            //set DE2 as an output
       PORTB = 0x00;        //initialize output to zero
    while (1)
```

```
    {
            PORTB |= (1<<2);            //turn on LED portb 2
            _delay_ms(250);             //delay of 0.25 seconds
            PORTB &= ~(1<<2);           //turn off the LED
            _delay_ms(250);             //delay of 0.25 seconds

    }
}
```

Calculation of TCNT value for task 1:

TCNT = (500,000/8)*0.25-1= 15624 (3D08)

4. INITIAL/DEVELOPED CODE OF TASK 2/A (assembly)

```
.org 0
        SBI DDRB, 2          ;PB2 as output
        LDI R17,0            ;needed to toogle led
        OUT PORTB,R17
        CBI DDRD, 2
        SBI PORTD, 2         ;port D2 pull up
        LDI R20,3            ;to set prescaler
        STS TCCR1B,R20       ;Prescaler: 64

start:
        SBIC   PIND, 2              ; skip when PIND2 is low
        RJMP   start
        SBI           PORTB, 2      ; Turn on the LED
        SBIC   PIND, 2              ; skip when PIND2 is low
        RJMP   start
        CALL   DELAY                ; Delay for 1 second
        CBI           PORTB, 2      ; Turn off the LED
        RJMP   start

delay:
        LDS R29, TCNT1H              ;loading upper bit of counter to R29
        LDS R28, TCNT1L             ;loading lower bit of counter to R28
        CPI R28,0x84                ;comparing if lower is 0x84
        BRSH body
        RJMP delay
body:
        CPI R29,0x1E
        BRSH done
        RJMP delay
done:
        LDI R20,0x00
        STS TCNT1H,R20              ;resetting the counter to 0 for next round
        LDI R20,0x00
        STS TCNT1L,R20              ;resetting the counter to 0 for next round
        RET
```

5. INITIAL/DEVELOPED CODE OF TASK 2/B (C)

```c
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 500000UL      //frequency 0.5MHz
#include <util/delay.h>

int main()
{
        DDRB |=(1<<2);
        DDRD &= ~(1<<2);

        while (1)
        {
                if (PIND & 0x04){                  //if switch pressed
                        PORTB |= (1<<2);     //turn on LED at port B2
                        _delay_ms (1000);    //wait one sec

                }
                else {
                        PORTB &= ~(1<<2);    //turn the LED off
                }

        }

}
```

Calculation of TCNT value for task 2:

TCNT = (500,000/64)*1-1= 7812 (1E84)

6. INITIAL/DEVELOPED CODE OF TASK 3/A (assembly)

```
.include "M328DEF.INC"
.org 0

        LDI R16, (1<<2)
        SBI DDRB , 2                ;set PB2 as an output
        LDI R17 , 0                 ; using for out
        OUT PORTB, R17
begin:
        RCALL delay                 ;calling timer to wait 0.25 sec
        EOR R17, R16                ;xor to toggle the LED
        out PORTB, R17
        RJMP begin
delay:
        LDI R28, 0x86               ;load 134 (TCNT value) to the register
        OUT TCNT0, R28              ;load the value to TCNT0
        LDI R20, 0
        OUT TCCR0A, R20
        LDI R20, 5                  ;5- 1024 prescaler (make the value less than 255- 8
bits)
        OUT TCCR0B, R20             ;Timer0, normal mode, int clk, 1024 prescaler
```

```
lp:
	IN R20, TIFR0		;check if TOV in TIFR register is set to one
	SBRS R20, TOV0		;if so, skip next instruction
	RJMP lp
done:
	LDI R20,0x00
	OUT TCCR0A, R20		;stop timer
	OUT TCCR0B, R20		;stop timer
	LDI R20, (1<<TOV0)
	OUT TIFR0, R20		;clear flag by writing 1 to TIFR
	RET
```

7. INITIAL/DEVELOPED CODE OF TASK 3/B (C)

```
#include <avr/io.h>
void delay();

int main()
{

	DDRB |= 0x04;		//set DE2 as an output
	PORTB = 0x00;		//initialize output to zero
	while (1)
	{
	delay();		// calling delay function
	PORTB ^= 0x04;		//xor to toggle LED portb 2
	}
}

void delay (){
	TCNT0 = 0x86;		//normal mode, counts up to 134
	TCCR0A |= 0;
	TCCR0B |= (1 << CS02) | (1 << CS00);	//1024 prescaler
	while ((TIFR0 & 0x01)== 0)
	{
		//DO nothing
	}
	TCCR0A = 0x00;			//stop timer
	TIFR0 |= (1<<TOV0);		//reset flag
}
```

8. INITIAL/DEVELOPED CODE OF TASK 4/A (assembly)

```
.include "M328DEF.INC"
.org 0
	jmp main

.org 0X20
	jmp TIMER0_OV_ISR

.org 0x100
main:
	LDI R20, HIGH(RAMEND)	;initalize the stack
	OUT SPH, R20
	LDI R20, LOW(RAMEND)
	OUT SPL, R20

	LDI R16, (1<<2)
	SBI DDRB , 2		;set PB2 as an output
	LDI R17 , 0		; using for out
	OUT PORTB, R17
begin:
	LDI R28, 134		;load 134 (TCNT value) to the register
	OUT TCNT0, R28		;load the value to TCNT0
	;timer int init
	LDI R20, (1<<TOIE0)
	STS TIMSK0, R20
	SEI			;enable interrupt

	LDI R20, 0
	OUT TCCR0A, R20
	LDI R20, 5		;5- 1024 prescaler (make the value less than 255- 8
bits)
	OUT TCCR0B, R20		;Timer0, normal mode, int clk, 1024 prescaler

lp:
	RJMP lp			;an infinit loop

TIMER0_OV_ISR:
	LDI R20, (1<<TOV0)
	OUT TIFR0, R20		;clear flag by writing 1 to TIFR
	EOR R17, R16		;xor to toggle the LED
	OUT PORTB, R17
	LDI R28, 134		;load 134 (TCNT value) to the register
	OUT TCNT0, R28		;load the value to TCNT0
	RETI
```

9. INITIAL/DEVELOPED CODE OF TASK 4/B (C)

```c
#include <avr/io.h>
#include <avr/interrupt.h>
volatile int counter = 0;


int main()
{
	DDRB = 0x02;		//set DE2 as an output
	PORTB = 0x00;		//initialize output to zero
	TCNT0 = 134;		//load 134 to the counter
	TIMSK0 = (1<<TOIE0); //start timer
	TCCR0A = 0;
	TCCR0B = 5;			//timer0, normal mode, prescaler of 1024
	sei();				//interrupt
	while (1)
	{
		// do nothing
	}
}

ISR (TIMER1_OVF_vect) {
	TIFR0 = (1<<0);
	if (counter == 134) {
		PORTB ^= (1<<2);		//toggle the LED portb 2
		counter = 0;
	}
	else
		counter++;
	}
```

10. INITIAL/DEVELOPED CODE OF TASK 5/A (assembly)

```
.include "M328DEF.INC"
.org 0
	jmp main
.org 0X02				;location for ext_int0
	jmp EX_int0
main:
	LDI R20, HIGH(RAMEND)	;initalize the stack
	OUT SPH, R20
	LDI R20, LOW(RAMEND)
	OUT SPL, R20
	SBI DDRB, 2			;PB2 as output
	LDI R17,0			;needed to toogle led
	OUT PORTB,R17
	LDI R20,3			;to set prescaler
	STS TCCR1B,R20		;Prescaler: 64
	LDI R20, (1<<INT0)
	STS EIMSK, R20		;set external interrupt
	SEI
here:
	jmp here
delay:
```

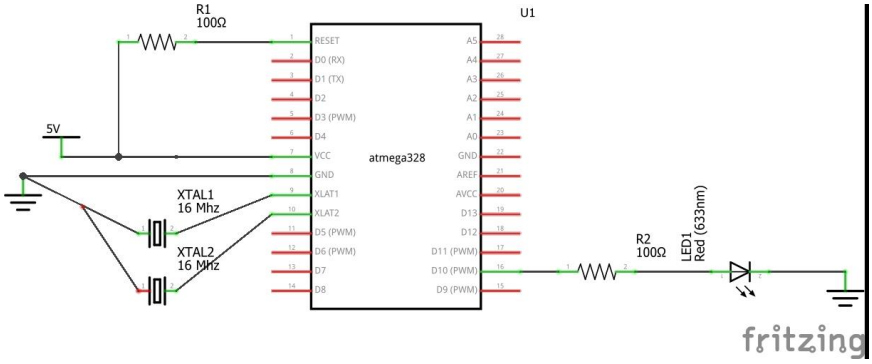
```
	LDS R29, TCNT1H		 ;loading upper bit of counter to R29
	LDS R28, TCNT1L		;loading lower bit of counter to R28
	CPI R28,0x84		;comparing if lower is 0x84
	BRSH body
	RJMP delay
body:
	CPI R29,0x1E
	BRSH done
	RJMP delay

EX_int0:
	SBI		PORTB, 2	; Turn on the LED
begin:
	SBIS	PIND2, 2
	RJMP begin
	call delay
	CBI		PORTB, 2	; Turn off the LED
	LDI R21, (1<<INTF0)
	STS EIFR, R21
	RETI
done:
	LDI R20,0x00
	STS TCNT1H,R20		;resetting the counter to 0 for next round
	LDI R20,0x00
	STS TCNT1L,R20		;resetting the counter to 0 for next round
	ret
```

11. INITIAL/DEVELOPED CODE OF TASK 5/B (C)

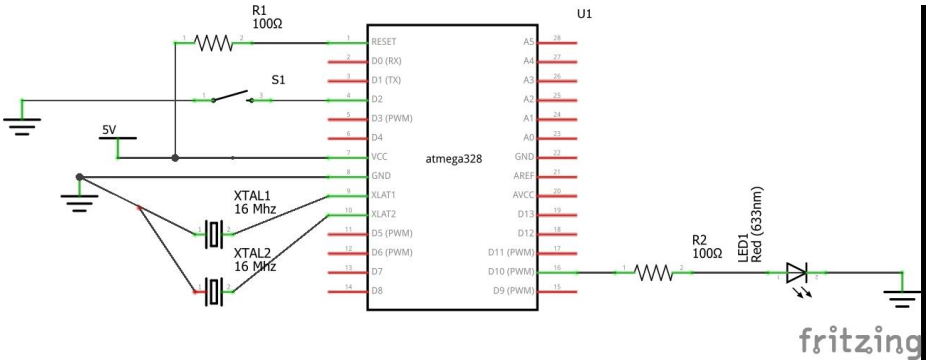
```
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 1000000UL     //frequency 1MHz
#include <util/delay.h>

int main()
{
	DDRB |=(1<<2);			//portb sets as an output
	DDRD &= ~(1<<2);		//protd as an input
	EIMSK = (1<<INT0);		//int0 is pin B2
	EICRA = (1<<ISC01) | (1<<ISC00); //rising edge
	sei ();				//enable interrupt
	while (1)
	{
		//do nothing
	}

}
ISR (INT0_vect)			//interrupt, turn on LED
{
	EIFR |=(1<<INTF0);
	PORTB |= (1<<2);	//turn on LED at port B2
	_delay_ms (1000);	//wait one sec
	PORTB &= ~(1<<2);	//turn the LED off


	}
```

12. SCHEMATICS

Task 1:



Task 2:



13. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 1:

Name	Address	Value	Bits
I/O PINB	0x23	0x00	☐☐☐☐☐☐☐☐
I/O DDRB	0x24	0x04	☐☐☐☐☐■☐☐
I/O PORTB	0x25	0x00	☐☐☐☐☐☐☐☐

Figure 1 toggle LED

Program Counter	0x00000005
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x3D08
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	125039
Frequency	0.500 MHz
Stop Watch	250,070.00 µs

Figure 2 required delay of 0.25 after reset stop watch

Task 1b:

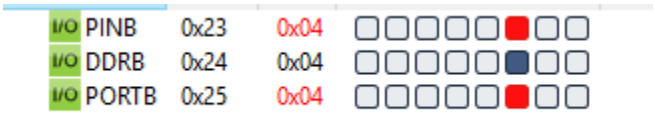


Figure 3 toggle LED

Program Counter	0x0000004D
Stack Pointer	0x08FD
X Register	0x0000
Y Register	0x08FF
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	1250017
Frequency	0.500 MHz
Stop Watch	2,500,004.00 µs

Figure 4 delay of 2.5 second after resetting the stop watch

Task 2:

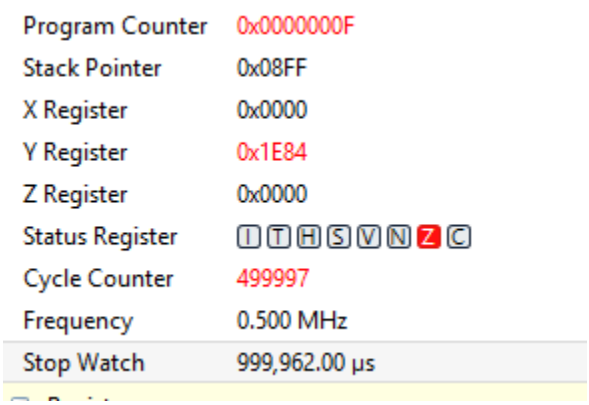


Figure 5 delay of 1 second after resetting the watch

Task 3:

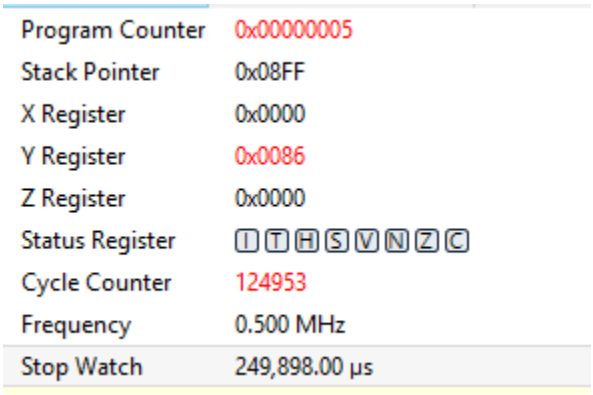


Figure 6 delay of 0.25 sec after resetting the stop watch

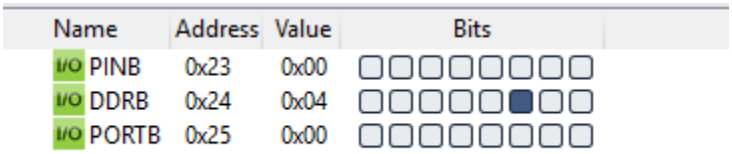


Figure 7 toggle LED

Task 3- C code (measuring the delay after resetting the stop watch)

Program Counter	0x00000049
Stack Pointer	0x08FB
X Register	0x0000
Y Register	0x0804
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	124958
Frequency	0.500 MHz
Stop Watch	249,868.00 µs

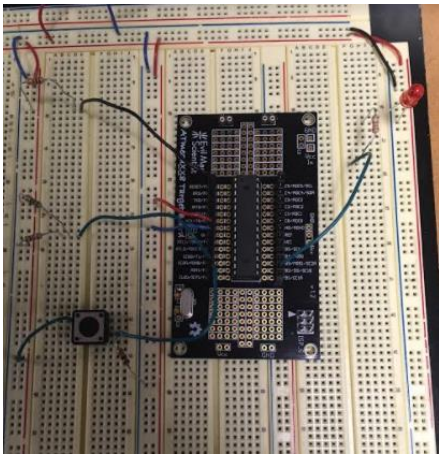
Name	Address	Value	Bits
I/O PINB	0x23	0x00	
I/O DDRB	0x24	0x04	
I/O PORTB	0x25	0x00	

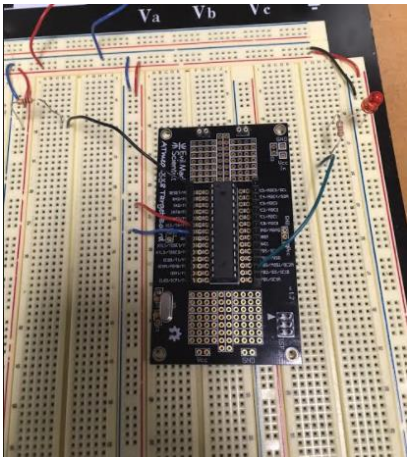
Task 4 (C code) – results of the delay (0.25) seconds and toggle LED

Program Counter	0x00000050
Stack Pointer	0x08FD
X Register	0x0102
Y Register	0x08FF
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	125022
Frequency	0.500 MHz
Stop Watch	249,964.00 µs

Name	Address	Value	Bits
I/O PINB	0x23	0x00	
I/O DDRB	0x24	0x02	
I/O PORTB	0x25	0x00	

14. SCREENSHOT OF EACH DEMO (BOARD SETUP)





15. VIDEO LINKS OF EACH DEMO

Task 1- https://youtu.be/4F8kpBeYBOw
Task 2- https://youtu.be/73utptmEqtw

16. GITHUB LINK OF THIS DA

git@github.com:EilatAvidan/microcon.git

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Eilat Avidan