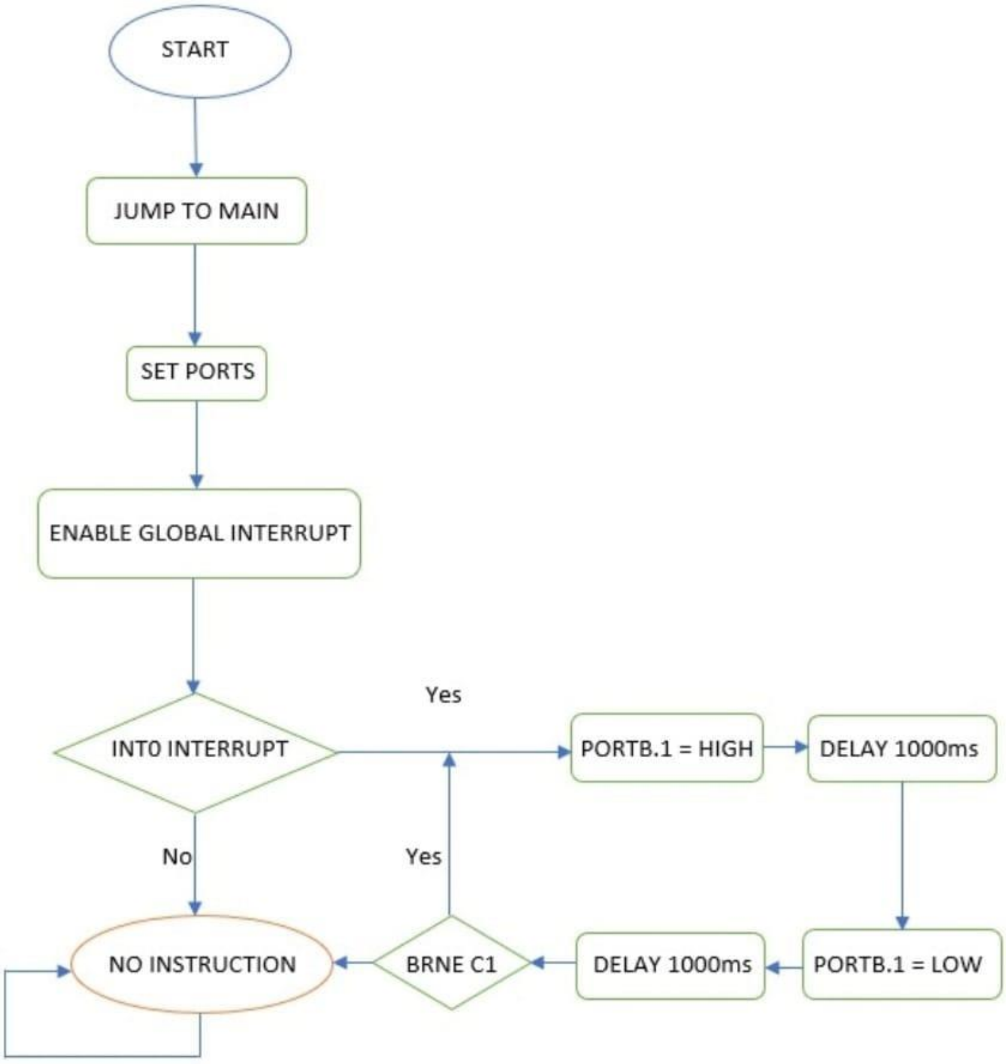
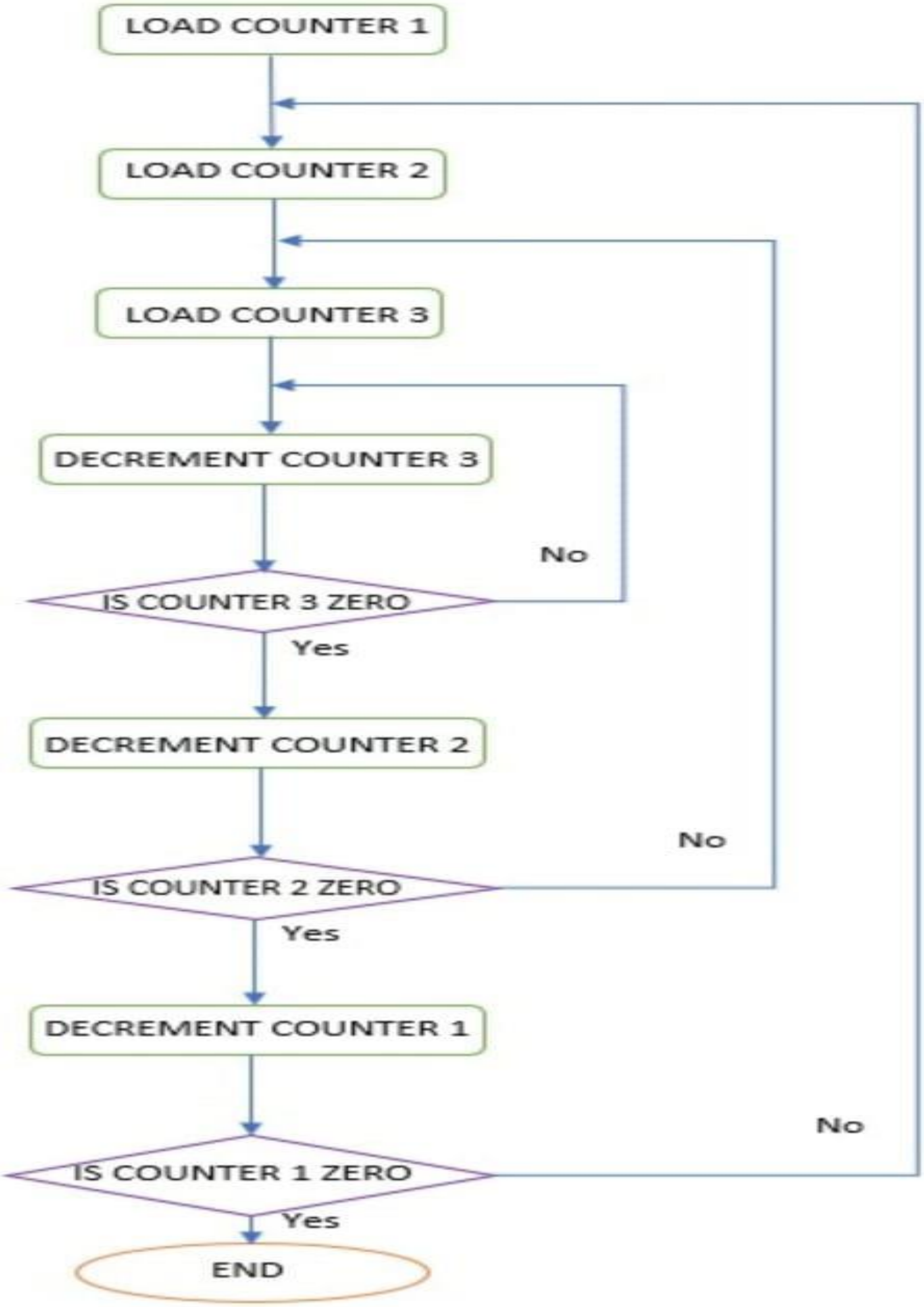
INTERRUPTS CODES FOR ATMEGA 8 IN ASM AND C PROGRAMMING

AIM: Using Atmel AVR assembly language programming make codes for following points by using different different interrupts.

- Use int0 to redo the same in the demo program (duely filled in). Once the switch is pressed the LED should blink 10 times (ON (or OFF) - 1 sec, duty cycle could be 50 %). Demonstrate both the cases.
- Rewrite the program in 'C' (int1). Rewrite the C program for int0

FLOWCHART


START
JUMP TO MAIN
SET PORTS
ENABLE GLOBAL INTERRUPT
Yes
INT0 INTERRUPT
PORTB.1 = HIGH
DELAY 1000ms
No
Yes
NO INSTRUCTION
BRNE C1
DELAY 1000ms
PORTB.1 = LOW



QUESTIONS ASKED IN MANUAL

QUES 1: INT1 ASM >

```
#include "m8def.inc"
.org 0x0000
rjmp reset
.org 0x0004   ;set location vector for external interrupt 1
rjmp int1_ISR

.org 0x0100

reset:
  ;Loading stack pointer address
     LDI R16,0x70
  OUT SPL,R16
  LDI R16,0x00
  OUT SPH,R16

  LDI R16,0x01
```

```
OUT DDRB,R16


;Interface port B pin0 to be output
;so to view LED blinking

LDI R16,0x00
OUT DDRD,R16

;Set MCUCR register to enable low level interrupt
IN R16,MCUCR
ORI R16,0x00
OUT MCUCR,R16

;Set GICR register to enable interrupt 1
IN R16,GICR
ORI R16,0x80
```

```
  OUT GICR,R16

  LDI R16,0x00
  OUT PORTB,R16

  SEI
ind_loop:rjmp ind_loop

int1_ISR:IN R16,SREG
 PUSH R16

 LDI R16,0x0A
 MOV R0,R16
 ;Modify below loops to make LED blink for 1 sec
c1:  LDI R16,0x01  ;making led high
 OUT PORTB,R16
 LDI R16,5
a1:  LDI R17,200
a2:  LDI R18,250
   a3:  NOP        //(((4*250+3)*200)+3)*5=
1,000,000us=1sec
     ;//4 is for (1 nop cycle and 1 dec cycle and 2
brne cycle

    DEC R18
 BRNE a3
 DEC R17
 BRNE a2
 DEC R16
 BRNE a1

 LDI R16,0x00
 OUT PORTB,R16 ;making led low
 LDI R16,5
b1:  LDI R17,200
b2:  LDI R18,250
    b3:  NOP
     DEC R18
 BRNE b3
 DEC R17
 BRNE b2
 DEC R16
 BRNE b1



 DEC R0
 BRNE c1

 POP R16
 OUT SREG,R16

 RETI
```

QUES 2. INT0 ASM >

```
#include "m8def.inc"
.org 0x0000
rjmp reset
.org 0x0002
rjmp int0_ISR
.org 0x0100
reset:

LDI R16, 0x70 ;Loading stack pointer address
OUT SPL, R16
```

```
LDI R16, 0x00
OUT SPH, R16


LDI R16,0x01
OUT DDRB, R16
LDI R16,0x00
OUT DDRD, R16
OUT PORTD, R16 ; PORTD.2 as Push Button - Input

LDI R16,0x00 ; Set MCUCR register to enable low level interrupt
OUT MCUCR, R16

LDI R16,0x80 ; Set D6 Bit of GICR register to enable interrupt INT0
OUT GICR, R16
```

```
LDI R16,0x00 ; PORTB as Output
OUT PORTB, R16
SEI ;
ind_loop: rjmp ind_loop
int0_ISR: IN R16,SREG
PUSH R16
LDI R16,0x0A
MOV R0,R16

c1: LDI R16,0x01 ; To blink LED 10 times (R0 used)
OUT PORTB,R16 ; Making LED - HIGH

LDI R16,5
a1: LDI R17,200
a2: LDI R18, 250      ;( (4*250+3)*200 + 3)*5 = aprox 1sec
```

```
a3: NOP
DEC R18
BRNE a3
DEC R17
BRNE a2
DEC R16
BRNE a1

LDI R16,0x00
OUT PORTB,R16 ; Making LED - HIGH
LDI R16,5
b1: LDI R17,200
b2: LDI R18,250
b3:NOP
DEC R18
BRNE b3
```

```
DEC R17
BRNE b2
DEC R16
BRNE b1

DEC R0
BRNE c1

POP R16 ; Popping context from Stack
OUT SREG,R16
RETI
```

QUES 3: INT1 C >

```
#include <avr/io.h>
#define F_CPU 1000000
```

```
#include <util/delay.h>
#include <avr/interrupt.h>

ISR (INT1_vect)
{

for(int i=0; i<10; i++)
{
//PortB is set to 1 for 1 sec (ON State)
PORTB = 0x01;
for(int i=0; i<5;i++)
{
_delay_ms(200);
}
//PortB is set to 0 for 1 sec (ON State)
PORTB = 0x00;
for(int i=0; i<5;i++)
{
_delay_ms(200);
}
}
}

int main (void)
{
//i/o port declarations
DDRD = 0x00;
DDRB = 0x01;
MCUCR = 0x00;
GICR = 0x80;
PORTB = 0x00;

//set interrupt flag of SREG
```

```
sei();

while (1)
{
//for infinite loop
}
}
```

QUES 4: INT0 C >

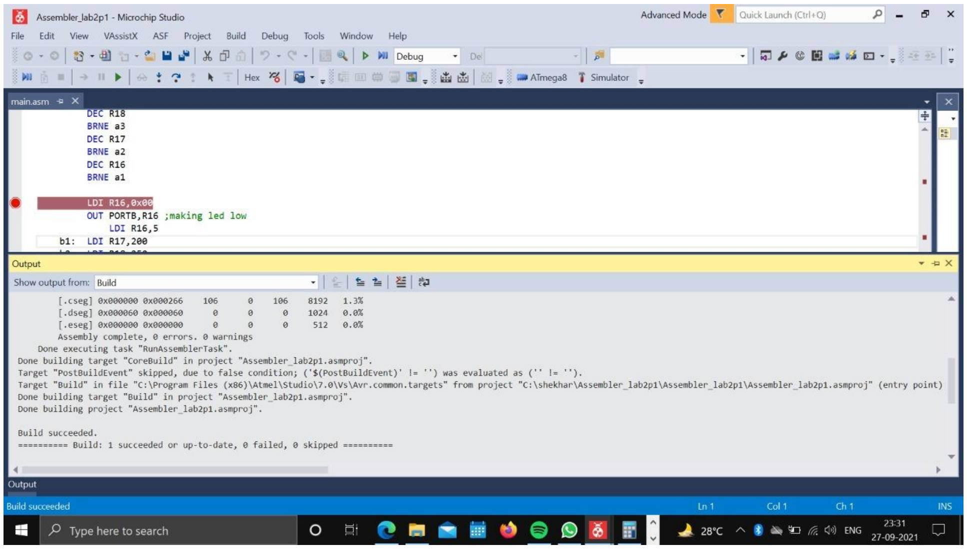
```
#include <avr/io.h>
#define F_CPU 1000000
#include <util/delay.h>
#include <avr/interrupt.h>

ISR (INT0_vect)
{
}

for(int i=0; i<10; i++)
{
//PortB is set to 1 for 1 sec (ON State)
PORTB = 0x01;
for(int i=0; i<5; i++)
{
_delay_ms(200);
}
//PortB is set to 0 for 1 sec (ON State)
PORTB = 0x00;
for(int i=0; i<5; i++)
{
_delay_ms(200);
}


}
```

```
}

int main (void)
{
// i/o port declarations
DDRD = 0x00;
DDRB = 0x01;
MCUCR = 0x00;
GICR = 0x40;
PORTB = 0x00;

//set interrupt flag of SREG
sei();

while (1)
{
//for infinite loop
}

}
```

INFERENCES

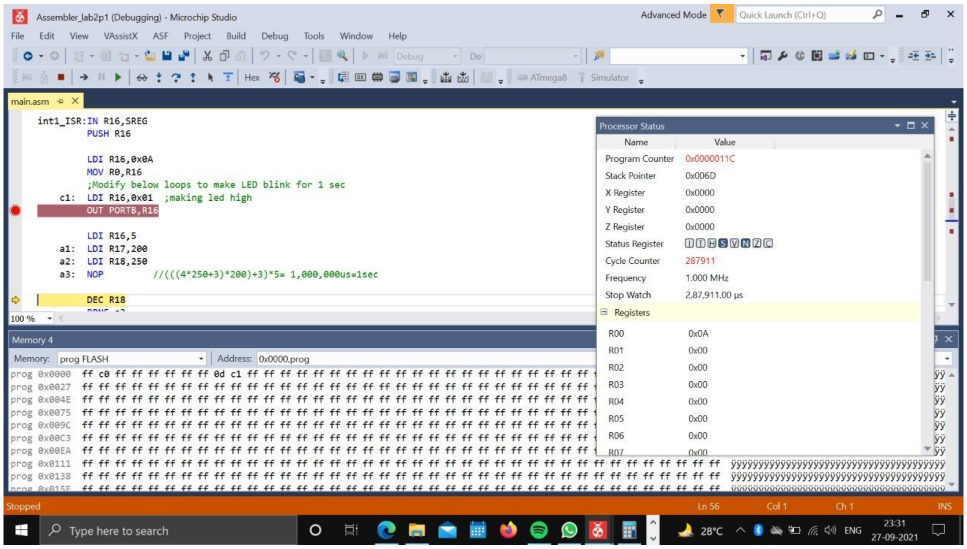
- Interrupts can be level or edge triggered.
- DDR Register is used to enable output and input modes of ports.
- By using Interrupts CPU need not to poll every device that needs service. So, it saves time of CPU.

INT 1 ASM



```
		DEC R18
		BRNE a3
		DEC R17
		BRNE a2
		DEC R16
		BRNE a1

		LDI R16,0x00
		OUT PORTB,R16 ;making led low
		    LDI R16,5
	b1:	LDI R17,200
```

```
        [.cseg] 0x000000 0x000266    106      0    106    8192   1.3%
        [.dseg] 0x000060 0x000060      0      0      0    1024   0.0%
        [.eseg] 0x000000 0x000000      0      0      0     512   0.0%
        Assembly complete, 0 errors. 0 warnings
    Done executing task "RunAssemblerTask".
Done building target "CoreBuild" in project "Assembler_lab2p1.asmproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\shekhar\Assembler_lab2p1\Assembler_lab2p1\Assembler_lab2p1.asmproj" (entry point)
Done building target "Build" in project "Assembler_lab2p1.asmproj".
Done building project "Assembler_lab2p1.asmproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```



Assembler_lab2p1 (Debugging) - Microchip Studio

main.asm

```
int1_ISR:IN R16,SREG
         PUSH R16

         LDI R16,0x0A
         MOV R0,R16
         ;Modify below loops to make LED blink for 1 sec
     c1: LDI R16,0x01  ;making led high
         OUT PORTB,R16

         LDI R16,5
     a1: LDI R17,200
     a2: LDI R18,250
     a3: NOP         //(((4*250+3)*200)+3)*5= 1,000,000us=1sec

         DEC R18
```

Processor Status

Name	Value
Program Counter	0x0000011C
Stack Pointer	0x006D
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	287911
Frequency	1.000 MHz
Stop Watch	2,87,911.00 µs
⊟ Registers	
R00	0x0A
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00

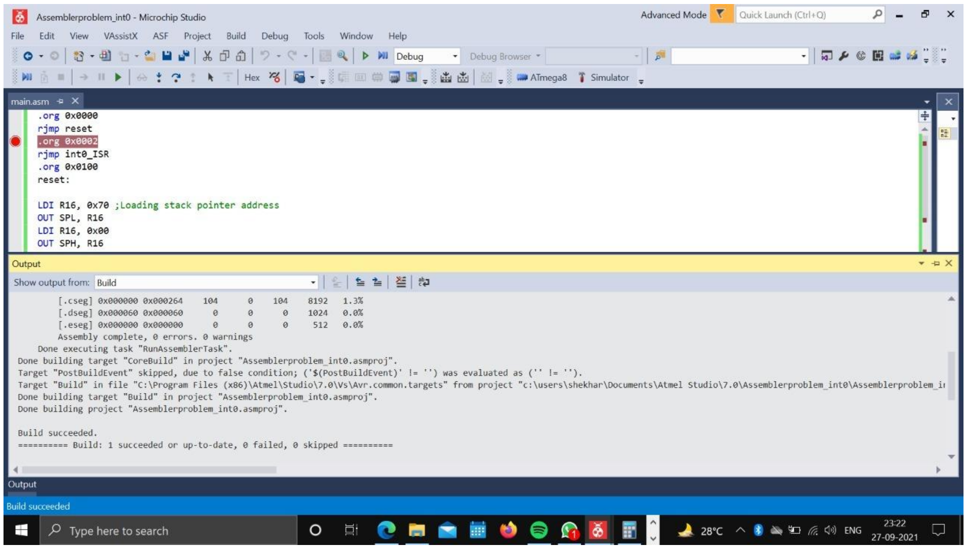
Memory 4

Memory: prog FLASH   Address: 0x0000,prog

```
prog 0x0000  ff c0 ff ff ff ff ff ff 0d c1 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ÿÿ
prog 0x0027  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ÿÿ
prog 0x004E  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ÿÿ
prog 0x0075  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ÿÿ
prog 0x009C  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ÿÿ
prog 0x00C3  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ÿÿ
prog 0x00EA  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ÿÿ
prog 0x0111  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0138  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Stopped   Ln 56   Col 1   Ch 1   INS

INT 0 ASM



Assemblerproblem_int0 - Microchip Studio

main.asm

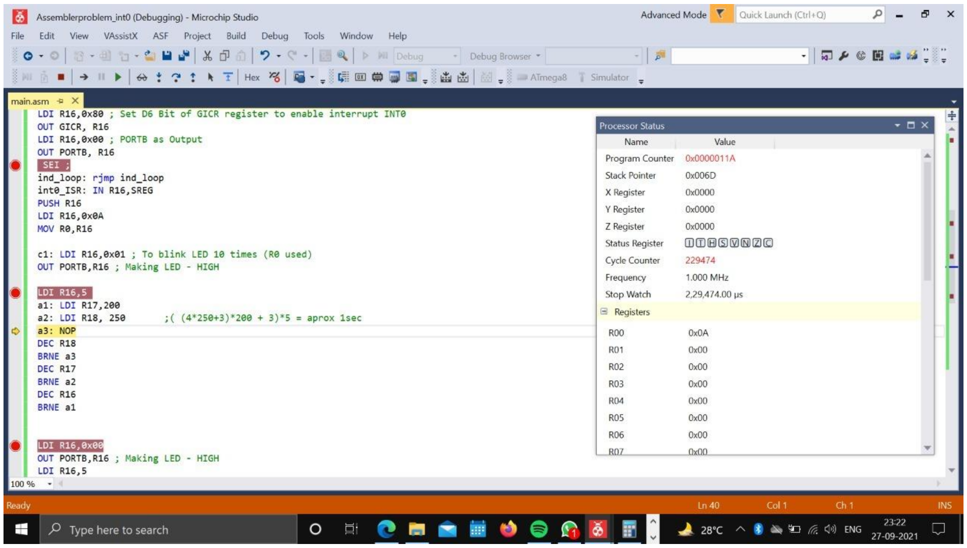
```
.org 0x0000
rjmp reset
.org 0x0002
rjmp int0_ISR
.org 0x0100
reset:

LDI R16, 0x70 ;Loading stack pointer address
OUT SPL, R16
LDI R16, 0x00
OUT SPH, R16
```

Output

Show output from: Build

```
        [.cseg] 0x000000 0x000264    104      0    104    8192   1.3%
        [.dseg] 0x000060 0x000060      0      0      0    1024   0.0%
        [.eseg] 0x000000 0x000000      0      0      0     512   0.0%
        Assembly complete, 0 errors. 0 warnings
    Done executing task "RunAssemblerTask".
Done building target "CoreBuild" in project "Assemblerproblem_int0.asmproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "c:\users\shekhar\Documents\Atmel Studio\7.0\Assemblerproblem_int0\Assemblerproblem_ir
Done building target "Build" in project "Assemblerproblem_int0.asmproj".
Done building project "Assemblerproblem_int0.asmproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

Build succeeded



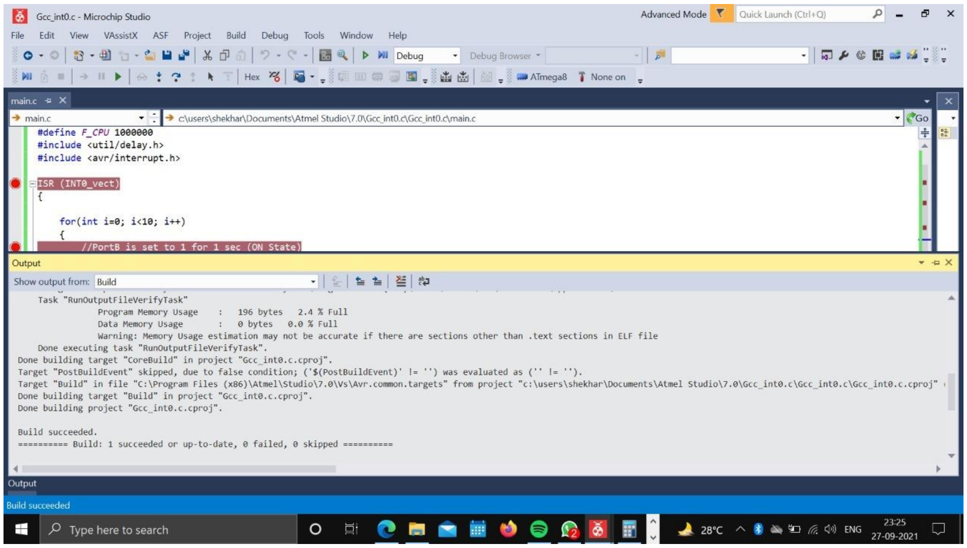
Assemblerproblem_int0 (Debugging) - Microchip Studio
Advanced Mode
Quick Launch (Ctrl+Q)
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Debug Browser
Hex
ATmega8
Simulator
main.asm

```
LDI R16,0x80 ; Set D6 Bit of GICR register to enable interrupt INT0
OUT GICR, R16
LDI R16,0x00 ; PORTB as Output
OUT PORTB, R16
 SEI ;
ind_loop: rjmp ind_loop
int0_ISR: IN R16,SREG
PUSH R16
LDI R16,0x0A
MOV R0,R16

c1: LDI R16,0x01 ; To blink LED 10 times (R0 used)
OUT PORTB,R16 ; Making LED - HIGH

LDI R16,5
a1: LDI R17,200
a2: LDI R18, 250       ;( (4*250+3)*200 + 3)*5 = aprox 1sec
a3: NOP
DEC R18
BRNE a3
DEC R17
BRNE a2
DEC R16
BRNE a1


LDI R16,0x00
OUT PORTB,R16 ; Making LED - HIGH
LDI R16,5
```

Processor Status

Name	Value
Program Counter	0x0000011A
Stack Pointer	0x006D
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	229474
Frequency	1.000 MHz
Stop Watch	2,29,474.00 µs
Registers	
R00	0x0A
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00

INT0 C



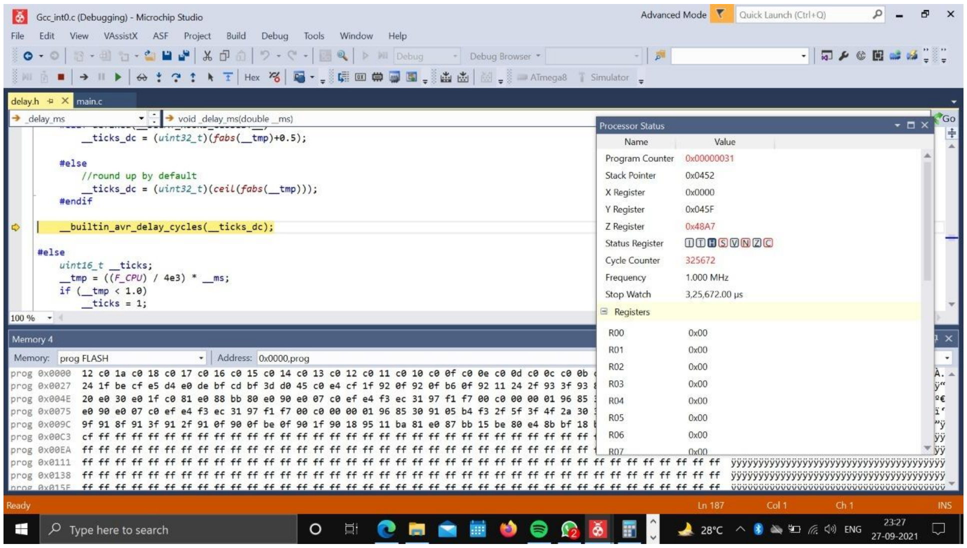
Gcc_int0.c - Microchip Studio
Advanced Mode
Quick Launch (Ctrl+Q)
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Debug Browser
Hex
ATmega8
None on
main.c
main.c
c:\users\shekhar\Documents\Atmel Studio\7.0\Gcc_int0.c\Gcc_int0.c\main.c
Go

```
#define F_CPU 1000000
#include <util/delay.h>
#include <avr/interrupt.h>

ISR (INT0_vect)
{

    for(int i=0; i<10; i++)
    {
        //PortB is set to 1 for 1 sec (ON State)
```

Output

Show output from: Build

```
    Task "RunOutputFileVerifyTask"
                Program Memory Usage    :   196 bytes   2.4 % Full
                Data Memory Usage       :   0 bytes    0.0 % Full
                Warning: Memory Usage estimation may not be accurate if there are sections other than .text sections in ELF file
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "Gcc_int0.c.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "c:\users\shekhar\Documents\Atmel Studio\7.0\Gcc_int0.c\Gcc_int0.c\Gcc_int0.c.cproj"
Done building target "Build" in project "Gcc_int0.c.cproj".
Done building project "Gcc_int0.c.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```



Gcc_int0.c (Debugging) - Microchip Studio
Advanced Mode
Quick Launch (Ctrl+Q)
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Debug Browser
Hex
ATmega8
Simulator
delay.h
main.c
_delay_ms
void _delay_ms(double __ms)
Go
100 %

```
        __ticks_dc = (uint32_t)(fabs(__tmp)+0.5);

    #else
        //round up by default
        __ticks_dc = (uint32_t)(ceil(fabs(__tmp)));
    #endif

    __builtin_avr_delay_cycles(__ticks_dc);

#else
    uint16_t __ticks;
    __tmp = ((F_CPU) / 4e3) * __ms;
    if (__tmp < 1.0)
        __ticks = 1;
```

Processor Status

Name	Value
Program Counter	0x00000031
Stack Pointer	0x0452
X Register	0x0000
Y Register	0x045F
Z Register	0x48A7
Status Register	I T H S V N Z C
Cycle Counter	325672
Frequency	1.000 MHz
Stop Watch	3,25,672.00 µs
⊟ Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00

Memory 4

Memory: prog FLASH   Address: 0x0000,prog

```
prog 0x0000  12 c0 1a c0 18 c0 17 c0 16 c0 15 c0 14 c0 13 c0 12 c0 11 c0 10 c0 0f c0 0e c0 0d c0 0c c0 0b
prog 0x0027  24 1f be cf e5 d4 e0 de bf cd bf 3d d0 45 c0 e4 cf 1f 92 0f 92 0f b6 0f 92 11 24 2f 93 3f 93
prog 0x004E  20 e0 30 e0 1f c0 81 e0 88 bb 80 e0 90 e0 07 c0 ef e4 f3 ec 31 97 f1 f7 00 c0 00 00 01 96 85
prog 0x0075  e0 90 e0 07 c0 ef e4 f3 ec 31 97 f1 f7 00 c0 00 00 01 96 85 30 91 05 b4 f3 2f 5f 3f 4f 2a 30
prog 0x009C  9f 91 8f 91 3f 91 2f 91 0f 90 0f be 0f 90 1f 90 18 95 11 ba 81 e0 87 bb 15 be 80 e4 8b bf 18
prog 0x00C3  cf ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x00EA  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
prog 0x0111  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
prog 0x0138  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```