Introduction:

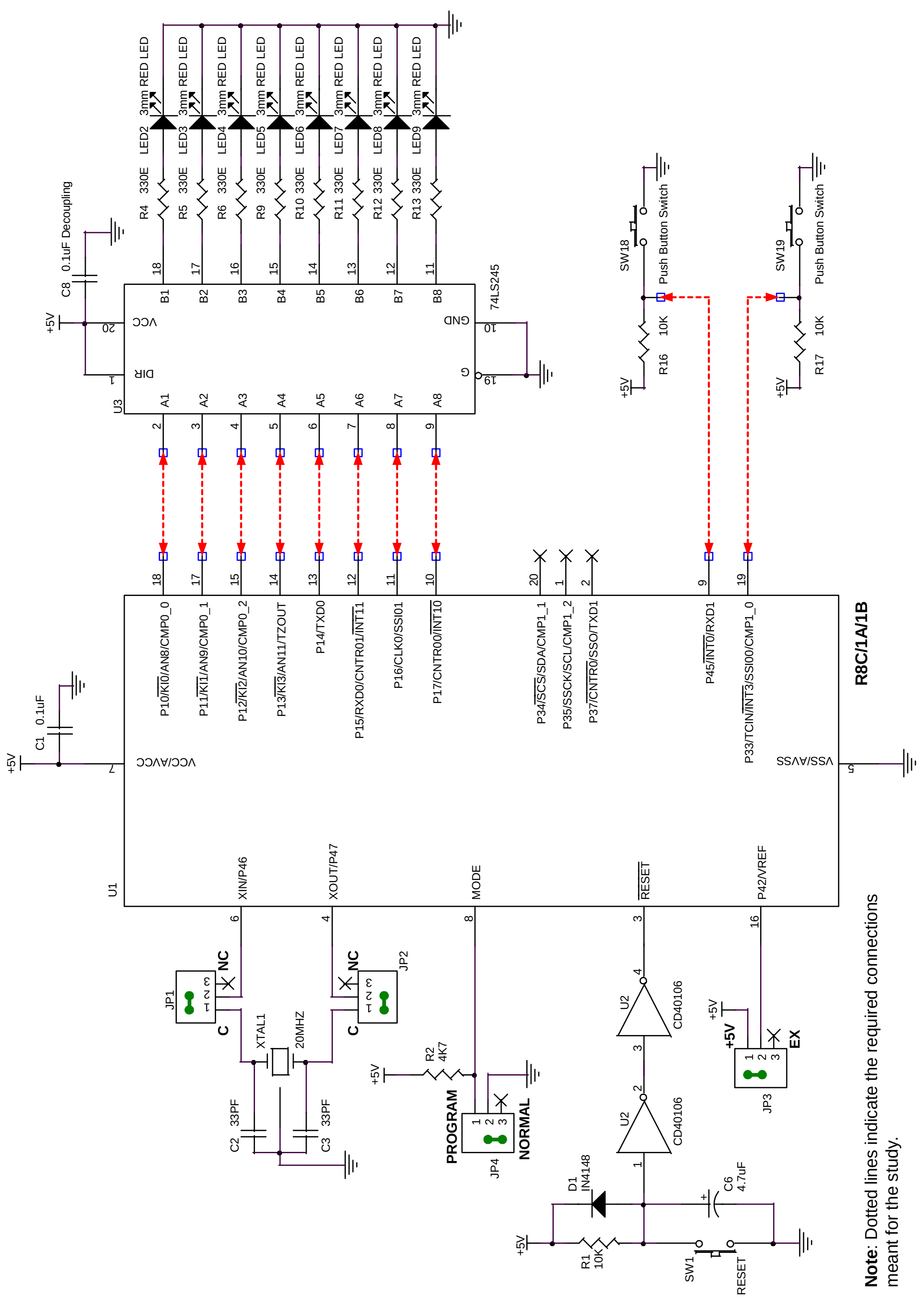
The interrupts INT0 and INT3 are considered for this study. A software counter is designed and the counter is incremented by one in the INT0 interrupt service routine. The counter value is decremented by one in the INT3 interrupt service routine. Point LEDs are used to display the count value.

Study Hardware:

A software counter is initialized in the program. For displaying the count value, 8 numbers of point LEDs are connected to the port lines, P10 to P17.

Two Push button switches are connected to INT0(P45) and INT2(P33) lines to trigger interrupt signals to the micon. The program is designed in such a way that the counter is incremented in the INT0 service routine and the same is decremented in the INT3 service routine.

Circuit:



Note: Dotted lines indicate the required connections meant for the study.

Connections:

- **LED**

1st LED	-> P10
2nd LED	-> P11
3rd LED	-> P12
4th LED	-> P13
5th LED	-> P14
6th LED	-> P15
7th LED	-> P16
8th LED	-> P17

- **Other Connections**

Pushbutton Switch	-> P45 (INT0)
Pushbutton Switch	-> P33 (INT3)

Functional Description:

Interrupt Mechanism:

All peripheral function Interrupts are maskable interrupts. Maskable interrupt has a bit, I flag, in the Flag register to enable or disable the corresponding maskable interrupt. Each maskable interrupt has a separate register, Interrupt Control Register to set a priority level for that interrupt. Seven levels of priority can be provided for the maskable interrupts individually.

Interrupt Sequence.

Flag register has a bit called I Flag which enables or disables the corresponding maskable interrupt. It should be set to 1 to enable the interrupt.

If an interrupt occurs during execution of an instruction, the execution of current instruction is completed and then the processor determines its priority. If the interrupt has higher priority, then the control is then transferred to the interrupt sequence from the next cycle.

If an interrupt occurs during execution of any of these SMOVB, SMOVF, SSTR or RMPA instructions, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

The interrupt sequence:

1. The CPU gets interrupt information (interrupt number and interrupt request priority level) from the address 0x0000. The control then clears the IR bit of the corresponding interrupt control register to 0. User need not clear this bit.
2. The Flag register is saved to stack
3. The I, D, and U flags are cleared to 0.
4. The CPU's internal temporary registers are saved to the stack.
5. PC is saved to stack.
6. The interrupt priority level of the accepted interrupt is set in the IPL bits of the Flag register. This indicates that another interrupt of higher priority can occur during the service of this interrupt.
7. The start address of the relevant interrupt routine set in the interrupt vector is stored in the PC.

Once this interrupt sequence is completed, the micon will start executing the first instruction in the service routine.

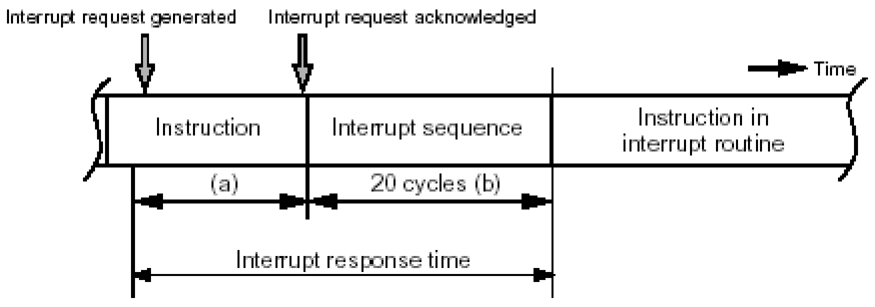
Interrupt Response Time:

The interrupt response time indicates a time taken for the controller to enter into the service routine after an interrupt is requested.

When an interrupt occurs, the controller completes the execution of the current instruction ((a) in the figure). This time depends on the instruction being executed. The DIVX instruction takes a maximum of 30 cycles.

Then the controller takes a time for the interrupt sequence ((b) in the figure), which is 20 cycles.

The interrupt response time of the controller is 50 cycles at the maximum and it may be less than that depending upon the instruction being executed.


Interrupt request generated
Interrupt request acknowledged
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
20 cycles (b)
Interrupt response time

(a) A time from when an interrupt request is generated till when the instruction then executing is completed. The length of this time varies with the instruction being executed. The DIVX instruction requires the longest time, which is equal to 30 cycles (without wait state, the divisor being a register).
(b) 21 cycles for address match and single-step interrupts.

The INT0 interrupt works on the falling edge and the INT3 works on the rising edge. These interrupts fall under the relocatable vector table area. Each interrupt has a separate control register, using which the priority level is set. Priority level of 1 is set for INT0 and the priority level is 2 for INT3.

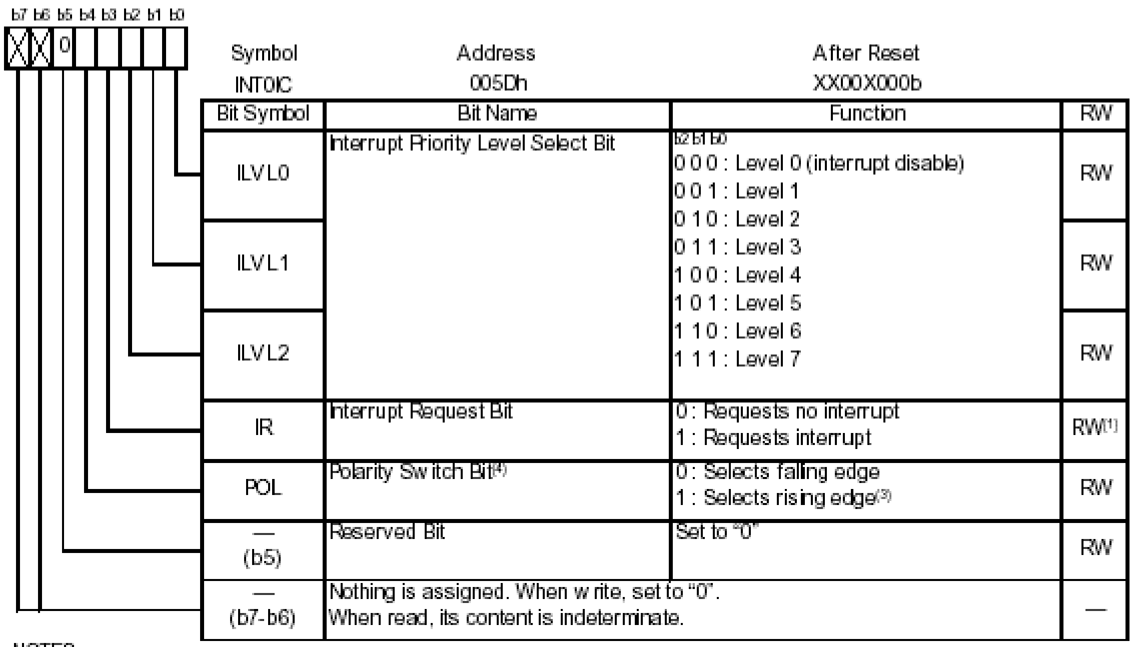
Flag register has a bit called I Flag which enables or disables the maskable interrupts. It should be set to 1 to enable the maskable interrupts.

If an interrupt occurs during execution of an instruction, the execution of current instruction is completed and then the processor determines its priority. Control is then transferred to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instructions, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

Registers used:

INT0IC -	Interrupt Control Register for INT0
INT3IC -	Interrupt Control Register for INT3
INTEN -	External Input Enable Register
TCC0 -	Timer C Control Register 0
INT0F -	INT0 input filter select register

Interrupt control register for INT0 (INT0IC)



Symbol	Address	After Reset
INT0IC	005Dh	XX00X000b

Bit Symbol	Bit Name	Function	RW
ILVL0	Interrupt Priority Level Select Bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disable) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt Request Bit	0 : Requests no interrupt 1 : Requests interrupt	RW(1)
POL	Polarity Switch Bit(4)	0 : Selects falling edge 1 : Selects rising edge(3)	RW
— (b5)	Reserved Bit	Set to "0"	RW
— (b7-b6)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

NOTES :

1. Only "0" can be written to the IR bit. (Do not write "1".)
2. To rewrite the interrupt control register, rewrite it when the interrupt request which is applicable for its register is not generated. Refer to **12.5.6 Changing Interrupt Control Registers.**
3. If the INTOPL bit in the INTEN register is set to "1" (both edges), set the POL bit to "0" (selects falling edge).
4. The IR bit may be set to "1" (requests interrupt) when the POL bit is rewritten. Refer to **12.5.5 Changing Interrupt Factor.**

Using Interrupt Control Register for INT0 (INT0IC), the interrupt priority level for INT0 is selected as 1 along with the falling edge for sensing.

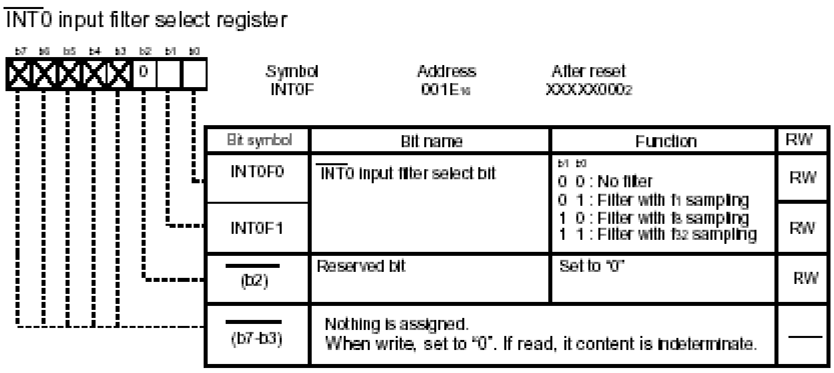
- Set ILVL0 bit to 1 to select the priority level as 1.
- POL bit is set to 0 to define falling edge sensing.

Interrupt Control Register for INT3 - INT3IC.

Bit symbol	Bit name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (Interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	RW^1
— (b7-b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

Similarly for interrupt INT3, the control register INT3IC is used. The value loaded is 0x02. Here priority level 2 is selected for INT3.

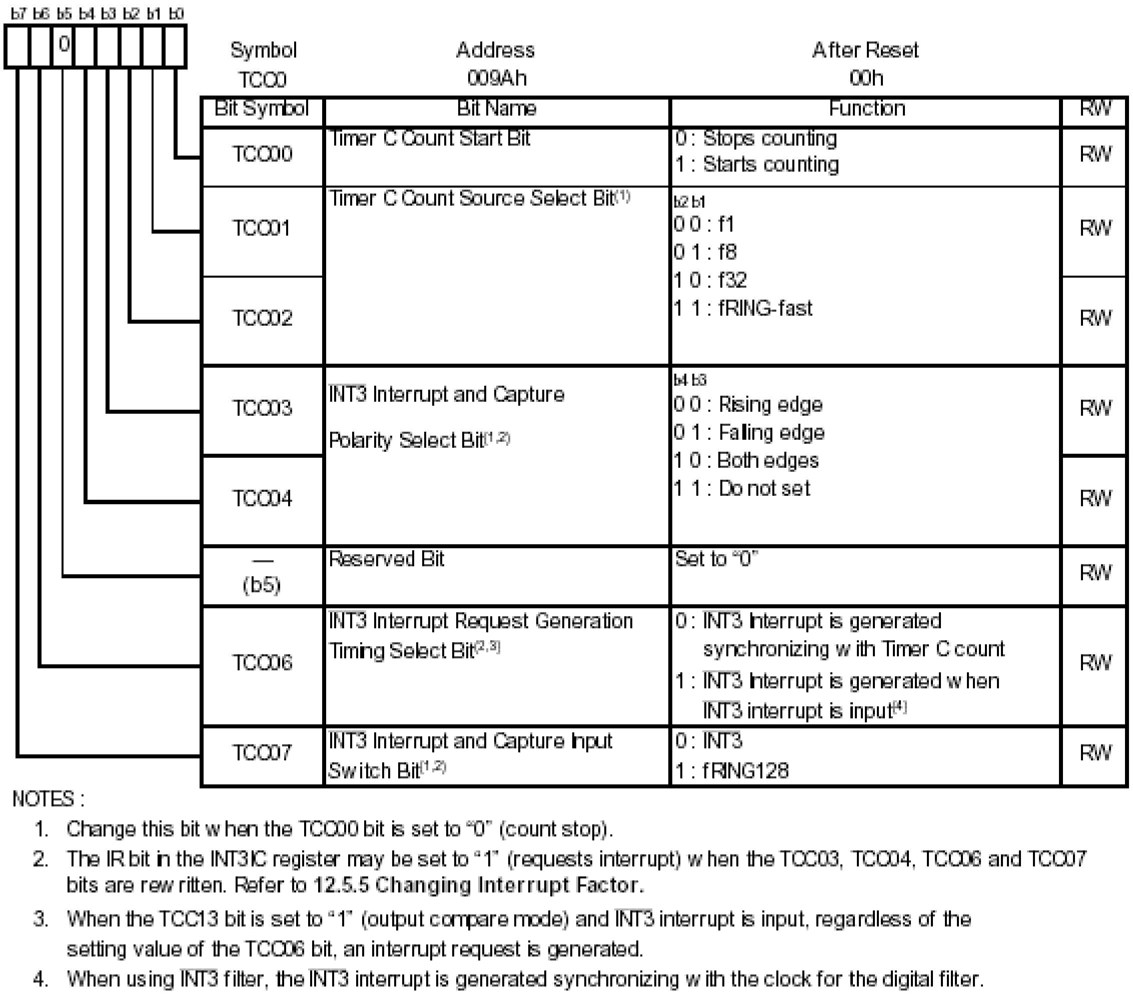
INT0 Input Filter Select Register - INTOF.



Bit symbol	Bit name	Function	RW
INT0F0	$\overline{\text{INT0}}$ input filter select bit	b1 b0 0 0 : No filter 0 1 : Filter with f1 sampling 1 0 : Filter with f8 sampling 1 1 : Filter with f32 sampling	RW
INT0F1			RW
— (b2)	Reserved bit	Set to "0"	RW
— (b7-b3)	Nothing is assigned. When write, set to "0". If read, it content is indeterminate.		—

No Filter is selected for INT0 interrupt.

Timer C Control Register 0 - TCC0.



b7 b6 b5 b4 b3 b2 b1 b0

0

Symbol	Address	After Reset
TCC0	009Ah	00h

Bit Symbol	Bit Name	Function	RW
TCC00	Timer C Count Start Bit	0 : Stops counting 1 : Starts counting	RW
TCC01	Timer C Count Source Select Bit[1]	b2 b1 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fRING-fast	RW
TCC02			RW
TCC03	INT3 Interrupt and Capture Polarity Select Bit[1,2]	b4 b3 0 0 : Rising edge 0 1 : Falling edge 1 0 : Both edges 1 1 : Do not set	RW
TCC04			RW
— (b5)	Reserved Bit	Set to "0"	RW
TCC06	INT3 Interrupt Request Generation Timing Select Bit[2,3]	0 : $\overline{\text{INT3}}$ interrupt is generated synchronizing with Timer C count 1 : $\overline{\text{INT3}}$ interrupt is generated when $\overline{\text{INT3}}$ interrupt is input[4]	RW
TCC07	INT3 Interrupt and Capture Input Switch Bit[1,2]	0 : $\overline{\text{INT3}}$ 1 : fRING128	RW

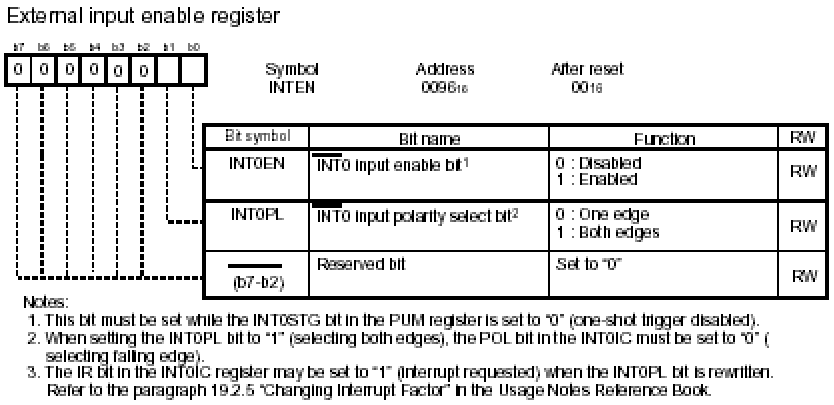
NOTES :

1. Change this bit when the TCC00 bit is set to "0" (count stop).
2. The IR bit in the INT3IC register may be set to "1" (requests interrupt) when the TCC03, TCC04, TCC06 and TCC07 bits are rewritten. Refer to **12.5.5 Changing Interrupt Factor.**
3. When the TCC13 bit is set to "1" (output compare mode) and $\overline{\text{INT3}}$ interrupt is input, regardless of the setting value of the TCC06 bit, an interrupt request is generated.
4. When using $\overline{\text{INT3}}$ filter, the $\overline{\text{INT3}}$ interrupt is generated synchronizing with the clock for the digital filter.

For INT3, the rising edge is selected using TCC0 register.

Set Bits TCC03 and TCC04 to 0 for rising edge sensing.

External Input Enable Register - INTEN.


External input enable register

Symbol	Address	After reset
INTEN	0096_{16}	00_{16}

Bit symbol	Bit name	Function	RW
INT0EN	$\overline{\text{INT0}}$ input enable bit[1]	0 : Disabled 1 : Enabled	RW
INT0PL	$\overline{\text{INT0}}$ input polarity select bit[2]	0 : One edge 1 : Both edges	RW
(b7-b2)	Reserved bit	Set to "0"	RW

Notes:
1. This bit must be set while the INT0STG bit in the PUM register is set to "0" (one-shot trigger disabled).
2. When setting the INT0PL bit to "1" (selecting both edges), the POL bit in the INT0IC must be set to "0" (selecting falling edge).
3. The IR bit in the INT0IC register may be set to "1" (interrupt requested) when the INT0PL bit is rewritten. Refer to the paragraph 19.2.5 "Changing Interrupt Factor" in the Usage Notes Reference Book.

INT0EN bit in INTEN register is set to 1 to enable INT0 interrupt.

Software Description:

When the push button switch connected to INT0 is pressed, an interrupt occurs. Interrupt is triggered during the press of the key, which indicates that the interrupt is falling edge sensitive. A software counter is maintained and the count gets incremented and displayed on the point LEDs.

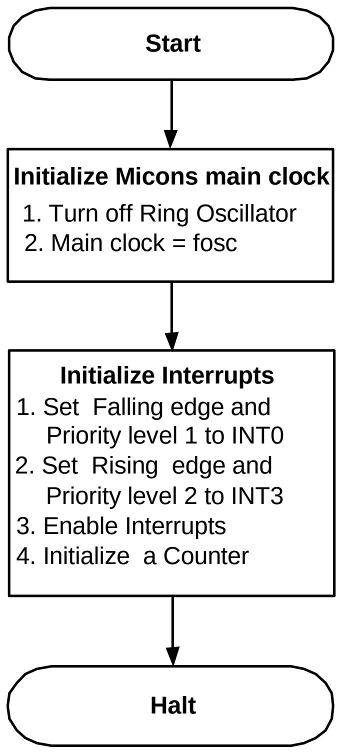
When the switch connected to INT3 is pressed and released, second interrupt occurs. Interrupt occurs during the release of the key, which indicates that the interrupt is rising edge sensitive. The same counter gets decremented and displayed on the point LEDs.

The functions in the file “**Demo10.C**” and short descriptions are listed below:

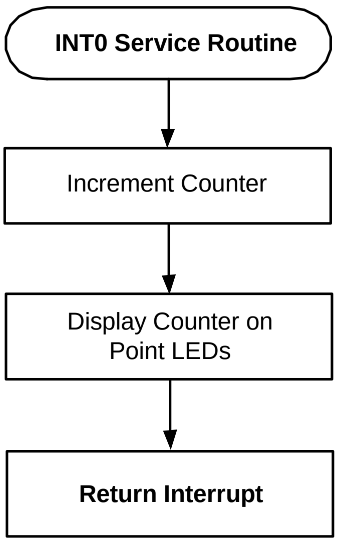
Functions	***Description***
main	Main routine which initializes the micon and the Interrupts INT0 and INT3.
MCUInitialize	Initializes Micon
Initialize INT0 and INT3	Initializes INT0 and INT3 with priorities 1 and 2 respectively.
Process INT0 interrupt	Interrupt service routine for INT0. Counter is incremented and displayed on Point LEDs.
Process INT3 interrupt	Interrupt service routine for INT3. Counter is decremented and displayed on Point LEDs

Program Flow:

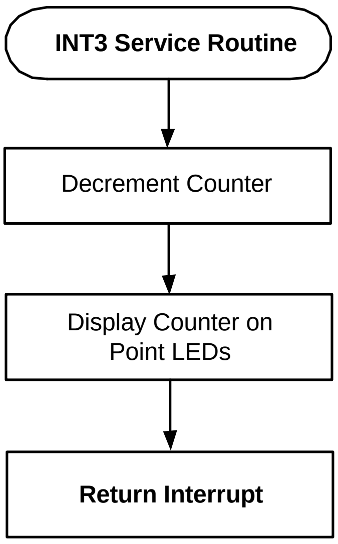
Main Program:


Start
Initialize Micons main clock
1. Turn off Ring Oscillator
2. Main clock = fosc
Initialize Interrupts
1. Set Falling edge and Priority level 1 to INT0
2. Set Rising edge and Priority level 2 to INT3
3. Enable Interrupts
4. Initialize a Counter
Halt

Interrupt INT0 Service Routine:



Interrupt INT3 service Routine:



Execute Study:

The 8 bit counter value is displayed on the Point LEDs.

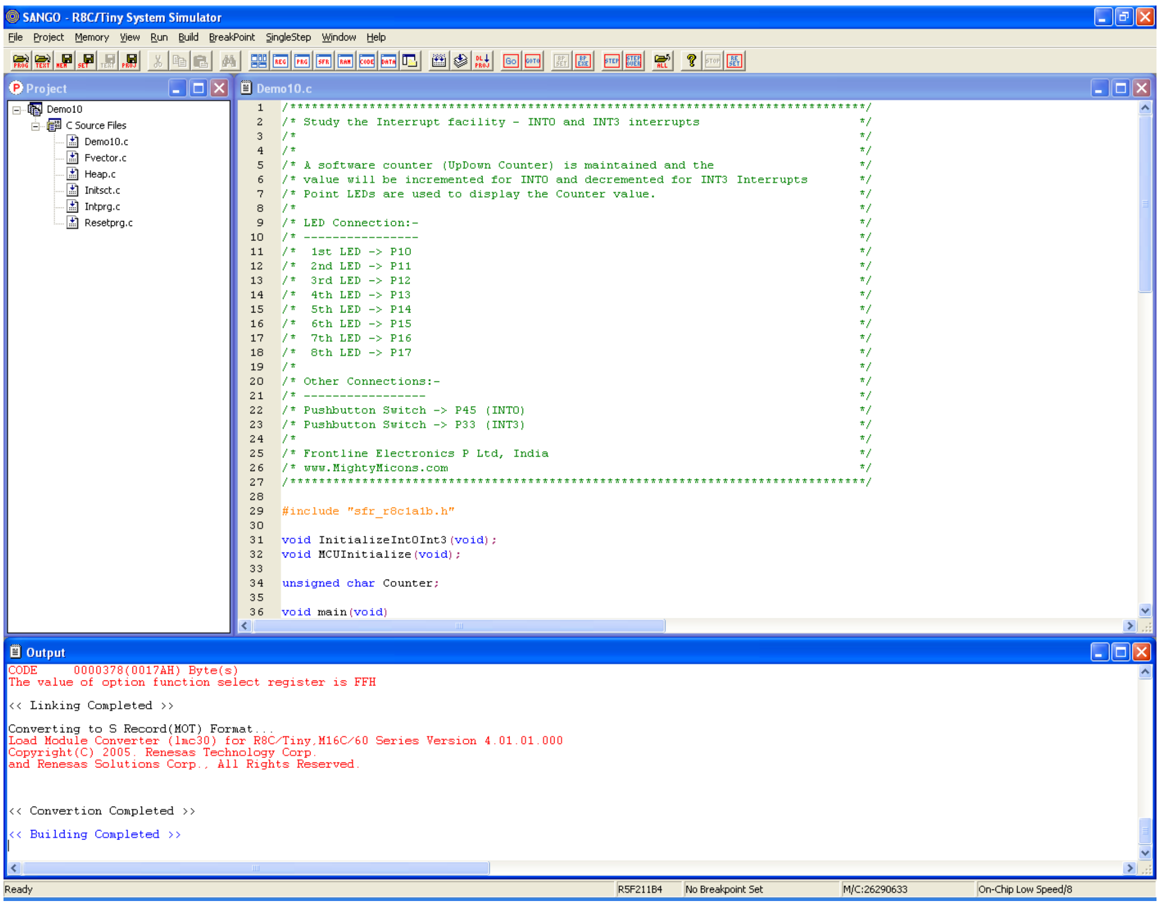
Press the switch connected to the INT0 pin. An interrupt occurs and the incremented count gets displayed on the LEDs.

Press the switch connected to the INT3 pin. An interrupt is activated and the decremented count gets indicated on the point LEDs.

Use Topview Simulator to Verify the Design.

Open the project Demo10 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo10.c file. Use **Build** option from **Build** menu to compile the project. An output window captures the compiler ouput.

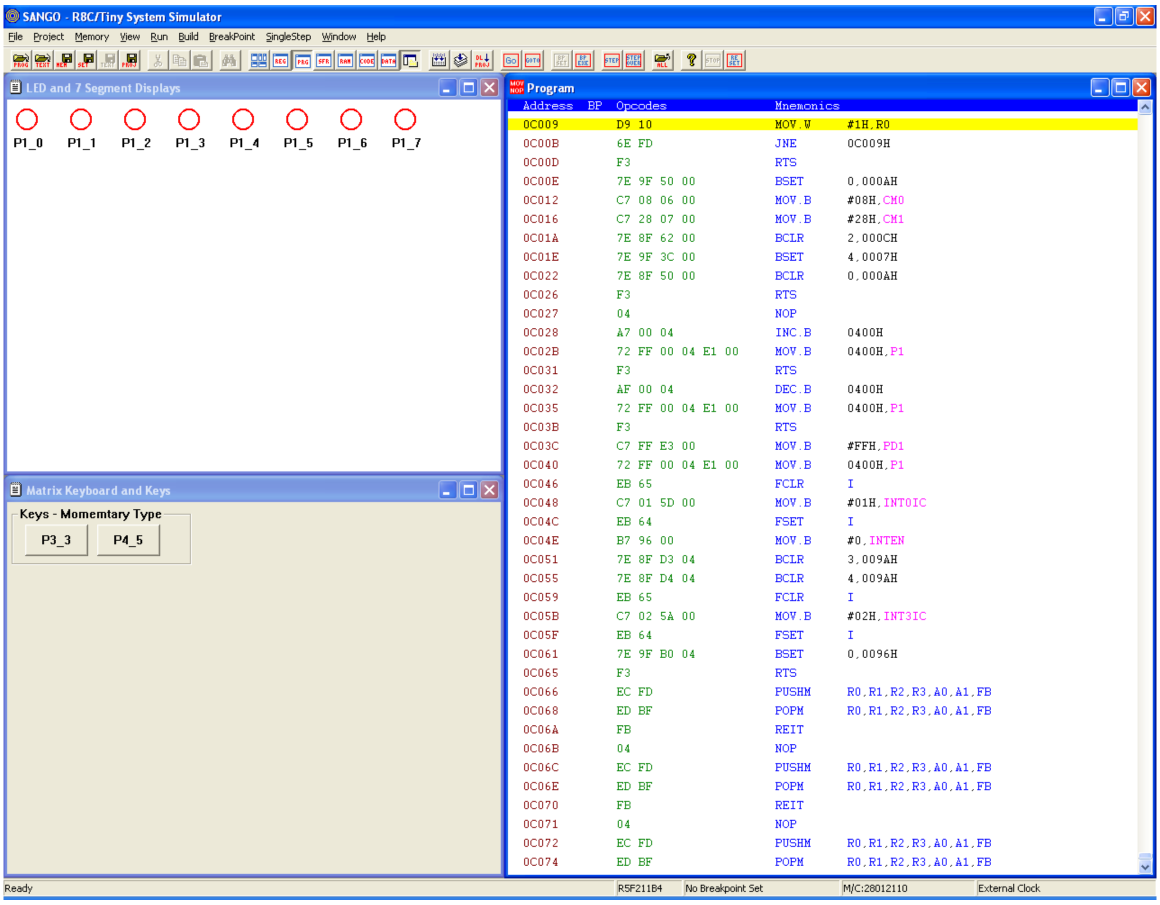
Use **Project** -> **Download Project** from main menu to download the .mot file into the simulator's memory for simulation.



```c
/******************************************************************************/
/* Study the Interrupt facility - INT0 and INT3 interrupts                    */
/*                                                                            */
/*                                                                            */
/* A software counter (UpDown Counter) is maintained and the                  */
/* value will be incremented for INT0 and decremented for INT3 Interrupts     */
/* Point LEDs are used to display the Counter value.                          */
/*                                                                            */
/* LED Connection:-                                                           */
/* ----------------                                                           */
/*  1st LED -> P10                                                            */
/*  2nd LED -> P11                                                            */
/*  3rd LED -> P12                                                            */
/*  4th LED -> P13                                                            */
/*  5th LED -> P14                                                            */
/*  6th LED -> P15                                                            */
/*  7th LED -> P16                                                            */
/*  8th LED -> P17                                                            */
/*                                                                            */
/* Other Connections:-                                                        */
/* -----------------                                                          */
/* Pushbutton Switch -> P45 (INT0)                                            */
/* Pushbutton Switch -> P33 (INT3)                                            */
/*                                                                            */
/* Frontline Electronics P Ltd, India                                         */
/* www.MightyMicons.com                                                       */
/******************************************************************************/

#include "sfr_r8c1a1b.h"

void InitializeInt0Int3(void);
void MCUInitialize(void);

unsigned char Counter;

void main(void)
```

```
CODE      0000378(0017AH) Byte(s)
The value of option function select register is FFH

<< Linking Completed >>

Converting to S Record(MOT) Format...
Load Module Converter (lmc30) for R8C/Tiny,M16C/60 Series Version 4.01.01.000
Copyright(C) 2005. Renesas Technology Corp.
and Renesas Solutions Corp., All Rights Reserved.



<< Convertion Completed >>

<< Building Completed >>
```

Connect point LEDs to the port line P10 to P17 using LED module setting and connect two push button switches to the port lines P45 and P33 to generate interrupt signals for INT0 and INT3. Open the LED and keyboard window and arrange them as shown for better visibility.


SANGO - R8C/Tiny System Simulator
File Project Memory View Run Build BreakPoint SingleStep Window Help
LED and 7 Segment Displays
P1_0 P1_1 P1_2 P1_3 P1_4 P1_5 P1_6 P1_7
Matrix Keyboard and Keys
Keys - Momemtary Type
P3_3 P4_5
Program
Ready
R5F211B4
No Breakpoint Set
M/C:28012110
External Clock

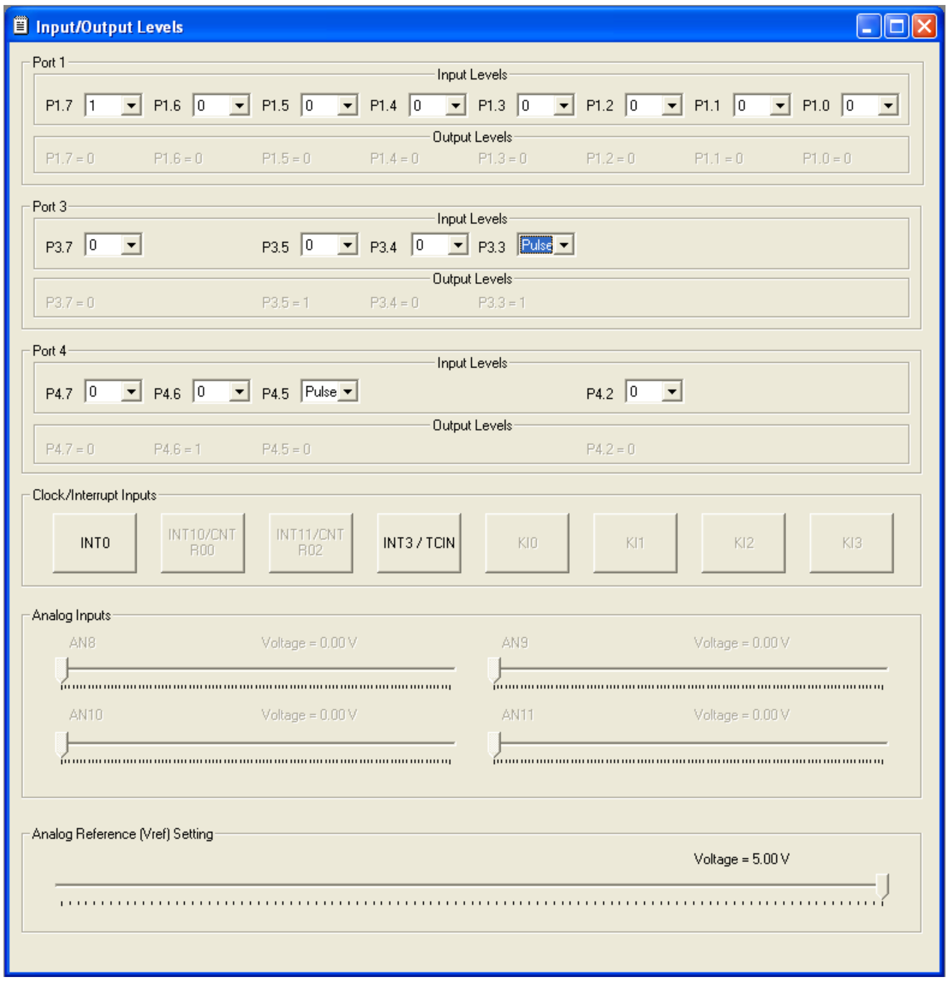
```
Address  BP  Opcodes              Mnemonics
0C009        D9 10                MOV.W     #1H,R0
0C00B        6E FD                JNE       0C009H
0C00D        F3                   RTS
0C00E        7E 9F 50 00          BSET      0,000AH
0C012        C7 08 06 00          MOV.B     #08H,CM0
0C016        C7 28 07 00          MOV.B     #28H,CM1
0C01A        7E 8F 62 00          BCLR      2,000CH
0C01E        7E 9F 3C 00          BSET      4,0007H
0C022        7E 8F 50 00          BCLR      0,000AH
0C026        F3                   RTS
0C027        04                   NOP
0C028        A7 00 04             INC.B     0400H
0C02B        72 FF 00 04 E1 00    MOV.B     0400H,P1
0C031        F3                   RTS
0C032        AF 00 04             DEC.B     0400H
0C035        72 FF 00 04 E1 00    MOV.B     0400H,P1
0C03B        F3                   RTS
0C03C        C7 FF E3 00          MOV.B     #FFH,PD1
0C040        72 FF 00 04 E1 00    MOV.B     0400H,P1
0C046        EB 65                FCLR      I
0C048        C7 01 5D 00          MOV.B     #01H,INT0IC
0C04C        EB 64                FSET      I
0C04E        B7 96 00             MOV.B     #0,INTEN
0C051        7E 8F D3 04          BCLR      3,009AH
0C055        7E 8F D4 04          BCLR      4,009AH
0C059        EB 65                FCLR      I
0C05B        C7 02 5A 00          MOV.B     #02H,INT3IC
0C05F        EB 64                FSET      I
0C061        7E 9F B0 04          BSET      0,0096H
0C065        F3                   RTS
0C066        EC FD                PUSHM     R0,R1,R2,R3,A0,A1,FB
0C068        ED BF                POPM      R0,R1,R2,R3,A0,A1,FB
0C06A        FB                   REIT
0C06B        04                   NOP
0C06C        EC FD                PUSHM     R0,R1,R2,R3,A0,A1,FB
0C06E        ED BF                POPM      R0,R1,R2,R3,A0,A1,FB
0C070        FB                   REIT
0C071        04                   NOP
0C072        EC FD                PUSHM     R0,R1,R2,R3,A0,A1,FB
0C074        ED BF                POPM      R0,R1,R2,R3,A0,A1,FB
```

Down load the program using **Download Project** command in **Project** menu.

Run the program using **Go** command in **Run** menu.

Now all the LEDs connected to the port P1 will be in off condition to indicate the binary value for “00H”. To increment the counter value give one interrupt pulse to INT0 by pressing the push button switch connected to the port line P45. To decrement the count value give one pulse to INT3 interrupt using another push button switch.

In Topview Simulator there is another facility provided in the I/O window to generate interrupt clock input pulses to the timer inputs. You can also this facility to generate interrupt signals.


Input/Output Levels
Port 1
Input Levels
P1.7 1
P1.6 0
P1.5 0
P1.4 0
P1.3 0
P1.2 0
P1.1 0
P1.0 0
Output Levels
P1.7 = 0
P1.6 = 0
P1.5 = 0
P1.4 = 0
P1.3 = 0
P1.2 = 0
P1.1 = 0
P1.0 = 0
Port 3
Input Levels
P3.7 0
P3.5 0
P3.4 0
P3.3 Pulse
Output Levels
P3.7 = 0
P3.5 = 1
P3.4 = 0
P3.3 = 1
Port 4
Input Levels
P4.7 0
P4.6 0
P4.5 Pulse
P4.2 0
Output Levels
P4.7 = 0
P4.6 = 1
P4.5 = 0
P4.2 = 0
Clock/Interrupt Inputs
INT0
INT10/CNTR00
INT11/CNTR02
INT3 / TCIN
KI0
KI1
KI2
KI3
Analog Inputs
AN8
Voltage = 0.00 V
AN9
Voltage = 0.00 V
AN10
Voltage = 0.00 V
AN11
Voltage = 0.00 V
Analog Reference (Vref) Setting
Voltage = 5.00 V