Introduction:

The interrupts INT0, INT1 and INT3 are considered for study purpose. Multiple interrupts are considered with priority levels.

Hardware:

Here 3 interrupts, INT0, INT2, and INT3 are considered in the study.

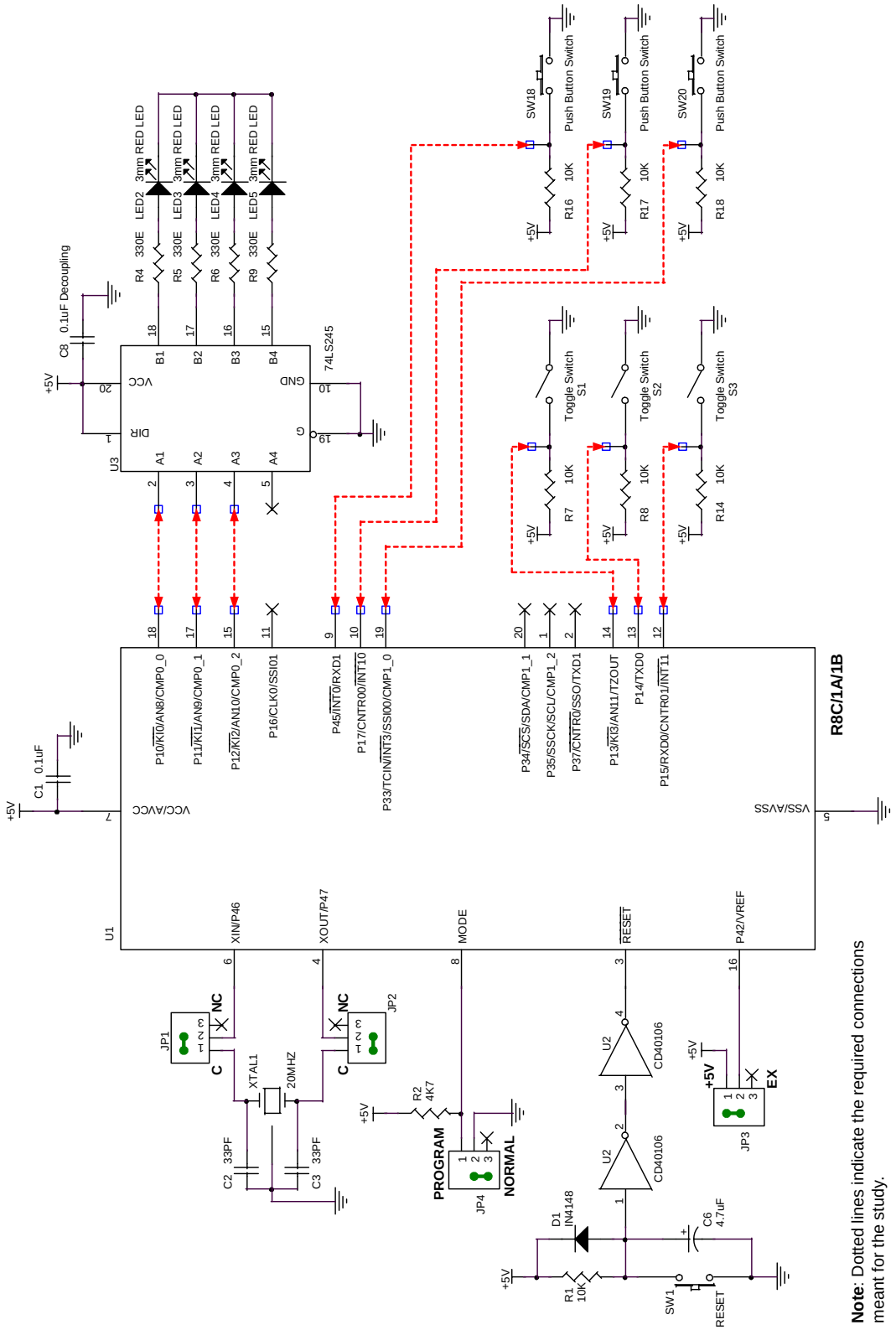
To each interrupt line, a push button switch is connected. Hence 3 push button switches are connected to the port lines P45 (INT0), P17(INT1) and P33(INT3).

Each interrupt uses one LED and one toggle switch in its service routine. For this purpose 3 LEDs are connected to the Port lines P10, P11 and P13. Similarly 3 toggle switches are connected to the port lines P13, P14 and P15.

The toggle switches and LEDs related to each interrupt is shown below:

Interrupts (Push Button Switches)	***Toggle Switches***	***LEDs***
INT0 (SW18)	P13 (S1)	P10 (LED2)
INT1 (SW19)	P14 (S2)	P11 (LED3)
INT3 (SW20)	P15 (S3)	P12 (LED4)

Circuit:



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

Connections:

- **LEDs**

  1st LED -> P10

  2nd LED -> P11

  3rd LED -> P12

- **Switches**

  Pushbutton Switch 1 -> P45 (INT0)

  Pushbutton Switch 2 -> P17 (INT1)

  Pushbutton Switch 3 -> P33 (INT3)

  Toggle Switch 1 -> P13

  Toggle Switch 2 -> P14

  Toggle Switch 3 -> P15

Functional Description:

Three interrupts INT0, INT1 and INT3 are considered for this study. All these three interrupts are set with three different priority levels.

These interrupts are provided with the relocatable vectors. Each interrupt has a separate control register, using which the priority level is set. Priority level is set low for INT0 which is 1. Priority level is 2 for INT1 and 3 for INT3.

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 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 maskable interrupt. It should be set to 1 to enable the maskable 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 new interrupt sequence from the next clock 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.

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

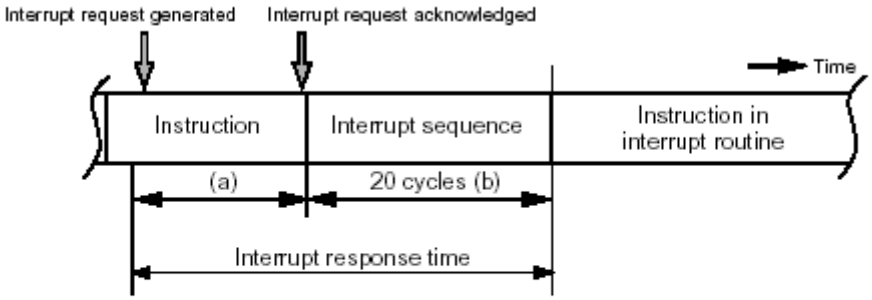
Interrupt Response Time:

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 less than that depending upon the instruction being executed.



(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.

Priority level:

All peripheral function interrupts are maskable. For each interrupt, priority can be set separately using a separate register. Since three bits are used, 7 levels of priority (1 to 7) can be used.

Condition 1:

If two or more interrupts, which has different priorities occur at the same time, in the resolving sequence, the interrupt which has highest priority will be serviced first. The remaining interrupts

will be kept pending. After completing the service of the highest priority interrupt, the pending interrupts will be considered.

Then the interrupt which has higher priority among the pending interrupts will be considered and serviced.

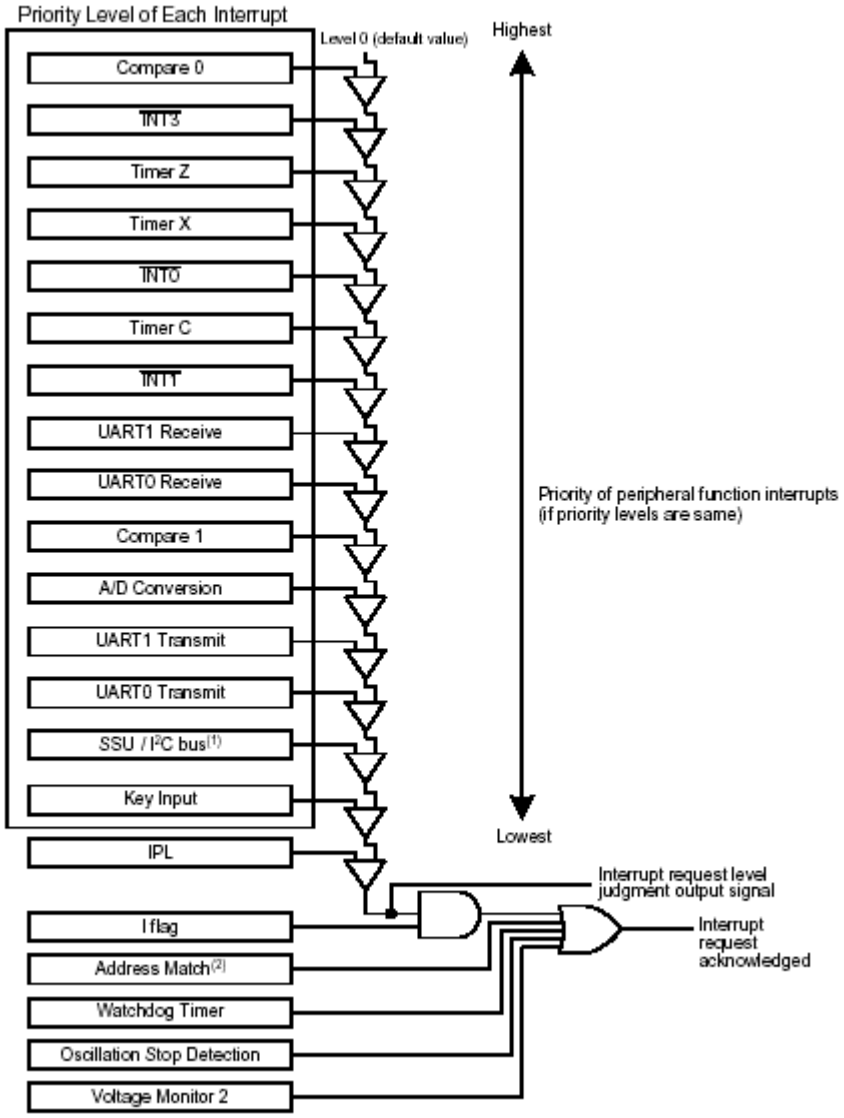
Condition 2:

While servicing an interrupt, if an interrupt with a higher priority occurs and at the same time the I-bit is enabled in the Flag register, the new interrupt will be serviced. After completing the new interrupt, the service of the current interrupt will be completed.

Condition 3:

If two or more interrupts occur at the same time and if they have same priority set using the ILVL0 to ILVL2 bits. A hardware priority resolver (The interrupt priority resolution circuit) is available in the controller which helps to resolve the priority of the maskable interrupts. The priority levels in the hardware is as follows:

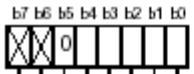
Compare 0 - Highest Priority
INT3
Timer Z
Timer X
INT0
Timer C
INT1
UART1 Reception
UART0 Reception
Compare 1
A/D Conversion
UART1 Transmission
UART0 Transmission
SSU/IIC Bus
Key Input - Lowest Priority


Priority Level of Each Interrupt
Level 0 (default value)
Highest
Compare 0
INT3
Timer Z
Timer X
INT0
Timer C
INT1
UART1 Receive
UART0 Receive
Compare 1
A/D Conversion
UART1 Transmit
UART0 Transmit
SSU / I²C bus(1)
Key Input
IPL
Priority of peripheral function interrupts
(if priority levels are same)
Lowest
Interrupt request level
judgment output signal
I flag
Address Match(2)
Watchdog Timer
Oscillation Stop Detection
Voltage Monitor 2
Interrupt
request
acknowledged

Registers used:

INT0IC - Interrupt Control Register for INTR0
INT1IC - Interrupt Control Register for INTR1
INT3IC - Interrupt Control Register for INTR3
INTEN - External Input Enable Register
TXMR - Timer Y,Z Mode Register
TYZMR - Timer Y,Z Mode Register
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 INT1 - INT1IC.

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	RW1
— (b7-b4)	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—

For INT1, the interrupt control register INT1IC is used. The value loaded is 0x02. INT1IC = 0x02. - priority level 2 is selected for INT1.

The interrupt control register for interrupt INT3 is INT3IC. The INT3IC is similar to the register INT1IC.

The data set in INT3IC is
INT2IC = 0x03 - priority level for INT3 is 3

Input Filter Select Register - INTOF.

$\overline{INT0}$ input filter select register

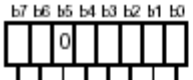
b7 b6 b5 b4 b3 b2 b1 b0 : X X X X X 0 [ ] [ ]

Symbol: INTOF | Address: 001E16 | After reset: XXXXX0002

Bit symbol	Bit name	Function	RW
INTOF0	$\overline{INT0}$ 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
INTOF1			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.



Symbol: TCC0 | Address: 009Ah | After Reset: 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 : INT3 Interrupt is generated synchronizing with Timer C count 1 : INT3 Interrupt is generated when INT3 interrupt is input(4)	RW
TCC07	INT3 Interrupt and Capture Input Switch Bit(1,2)	0 : 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 INT3 interrupt is input, regardless of the setting value of the TCC06 bit, an interrupt request is generated.
4. When using INT3 filter, the 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

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0		

Symbol: INTEN — Address: 0096₁₆ — After reset: 00₁₆

Bit symbol	Bit name	Function	RW
INT0EN	INT0 input enable bit[1]	0 : Disabled 1 : Enabled	RW
INT0PL	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.

Timer X Mode Register (TXMR).

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: TXMR — Address: 008Bh — After Reset: 00h

Bit Symbol	Bit Name	Function	RW
TXMOD0	Operating Mode Select Bit 0, 1[1]	b1 b0 0 0 : Timer mode or pulse period measurement mode 0 1 : Do not set 1 0 : Event count mode 1 1 : Pulse width measurement mode	RW
TXMOD1			RW
R0EDG	INT1/CNTR0 Polarity Switch Bit[2]	0 : Rising edge 1 : Falling edge	RW
TXS	Timer X Count Start Flag[3]	0 : Stops counting 1 : Starts counting	RW
TXOCNT	P3_7/CNTR0 Select Bit	Function varies depending on operating mode	RW
TXMOD2	Operating Mode Select Bit 2	0 : Other than pulse period measurement mode 1 : Pulse period measurement mode	RW
TXEDG	Active Edge Reception Flag	Function varies depending on operating mode	RW
TXUND	Timer X Underflow Flag	Function varies depending on operating mode	RW

NOTES :
1. When using INT1 interrupt, select modes other than pulse output mode.
2. The IR bit in the INT1IC register may be set to "1" (requests interrupt) when the R0EDG bit is rewritten. Refer to 12.5.5 Changing Interrupt Factor.
3. Refer to 14.1.6 Precautions on Timer X for precautions on the TXS bit.

For INT1 the rising edge is selected using TXMR register.

Set bit R0EDG to 0 for rising edge sensing.

Software Description:

Here we study the interrupt operations using the examples.

Example 1 - Higher priority interrupt INT1 is serviced during the service of INT0.

1. During the initialization of interrupts, the priority level of interrupts are set as follows using IPLV2 to IPLV0 bits. To each interrupt a push button switch is connected.

    1. INT2 - Highest priority
    2. INT1
    3. INT0 - lowest priority

    For each interrupt an LED, a push button switch and a toggle switch are used. The toggle switch and the LED helps to identify the interrupt. Keep all toggle switches at 0 level.

2. Press SW18 to activate interrupt INT0. In the interrupt service an LED (LED2) will be switched ON. The control will come out of the service routine only if toggle switch S1 gets a 1 level. Since the level of the switch is set at 0 level, the control stays in the service routine itself.
3. Now press SW19 to use interrupt INT1. Now INT1 is serviced and LED3 will be ON. INT1 is serviced because INT1 has higher priority than INT0. In the service routine of INT1 the LED3 will be switched ON.
3. Change level of S2, (the switch connected with INT1) to 1. The LED3 will be switched OFF. The flow of control comes out of INT1 service and now it stars servicing INT0.
4. Change level of S1 to 1. The LED2 will be OFF. The control comes out of INT0 service and it goes to main routine.

Example 2- Lower priority interrupt INT0 is not serviced during INT1 service.

1. In the initialization of interrupts, the priority level of interrupts are set as follows using IPLV2 to IPLV0 bits. To each interrupt a push button switch is connected.

    1. INT3 - Highest priority
    2. INT1
    3. INT0 - lowest priority

    For every interrupt, an LED, a push button switch and a toggle switch are used. The toggle switch and the LED helps to identify the interrupt. Keep all toggle switches at '0' level.

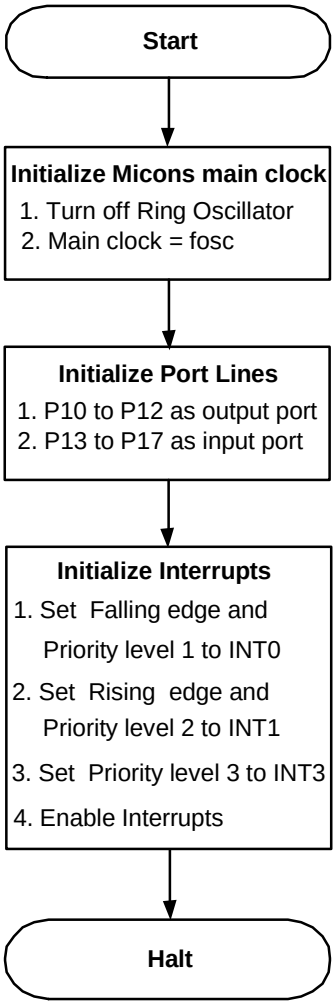
2. Press SW19 to interrupt the controller using INT1. In the interrupt service routine, LED3 is switched ON. The program control will come out of the service routine only if the switch S2 is set at 1 level.
3. Now press SW18(INT0) which has a lower priority than INT1. INT0 will not be serviced because it has a lower priority.
4. Change level of S2 to 1. The LED3 will be switched OFF. The control comes out of INT1 service and now pending interrupt INT0 is serviced and hence LED2 will be switched ON.
5. Change level of S1 to 1. The LED2 will be switched OFF and the control comes out of INT0 service.

The functions in the file "**Demo11.C**" and short descriptions are listed below:

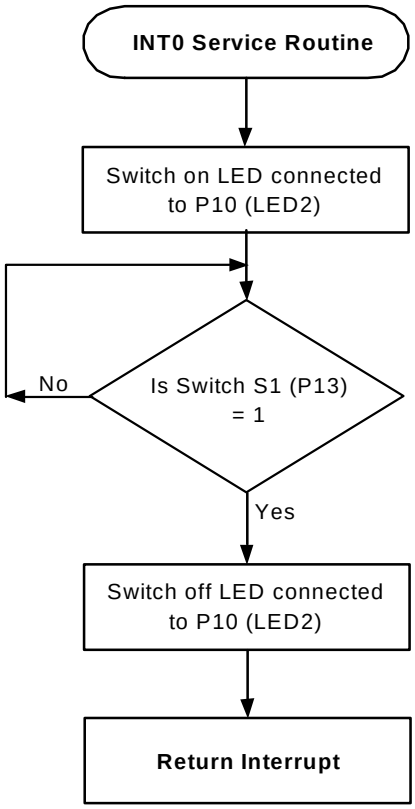
Functions	*Description*
main	Main routine which initializes the micon and the Interrupts INT0, INT1 and INT3.
MCUInitialize	Initializes Micon
Initialize INT0, INT1 and INT3	Initializes INT0, INT1 and INT3 with priorities 1, 2 and 3 respectively.
Process INT0 interrupt	Interrupt service routine for INT0.
Process INT1 interrupt	Interrupt service routine for INT1.
Process INT3 interrupt	Interrupt service routine for INT3.

Program Flow:

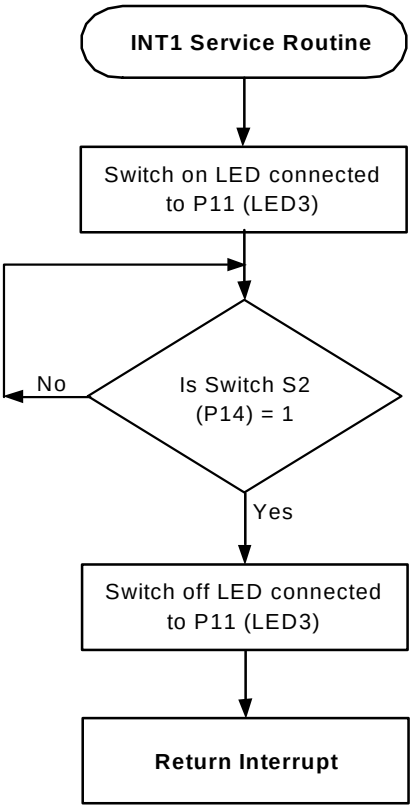
Main Program:



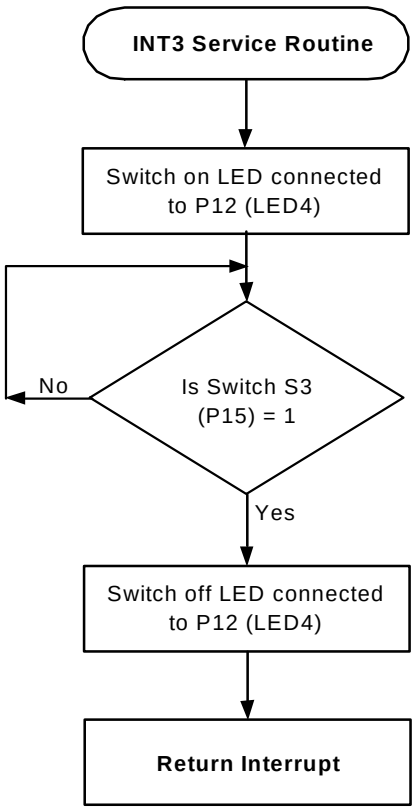
Interrupt INT0 service Routine:



Interrupt INT1 service Routine:



Interrupt INT3 service Routine:



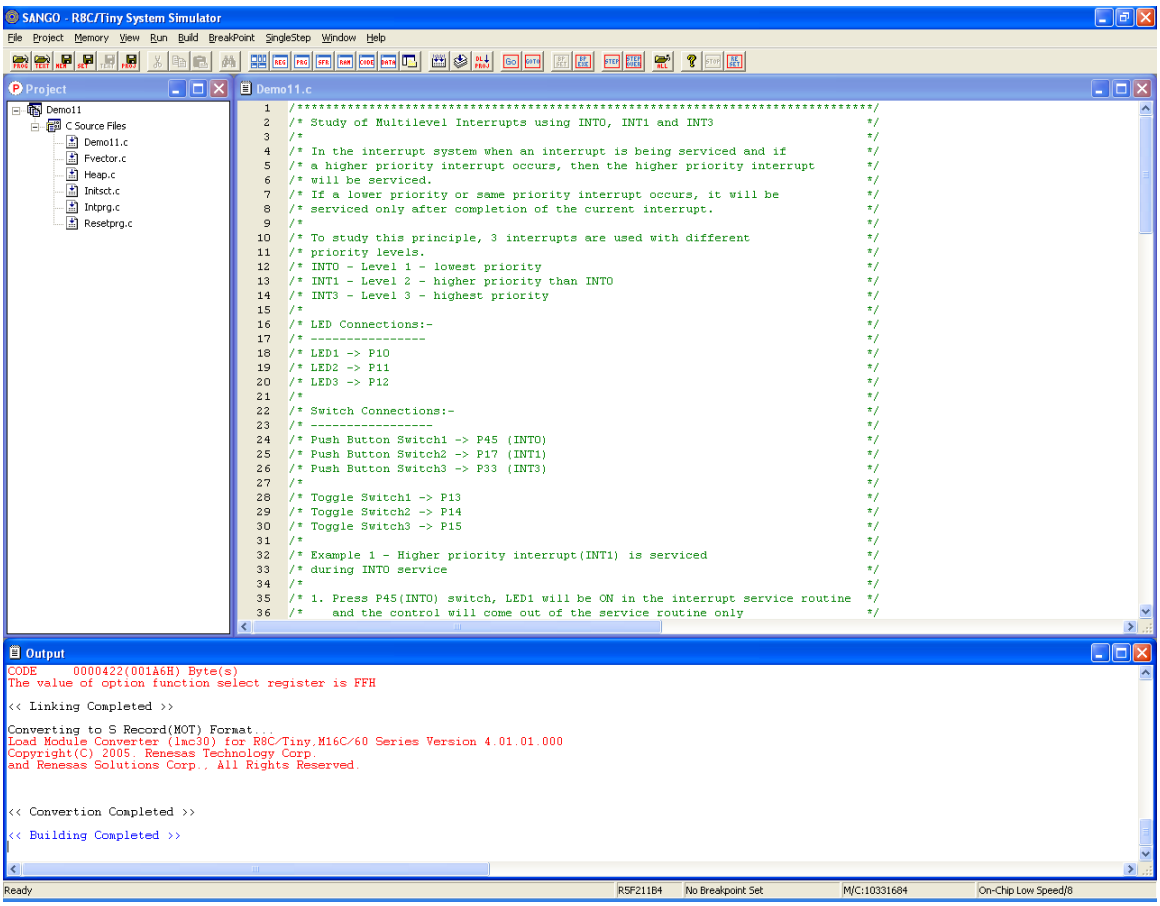
Execute the Study:

These examples help to study the priority levels of the interrupt mechanism.

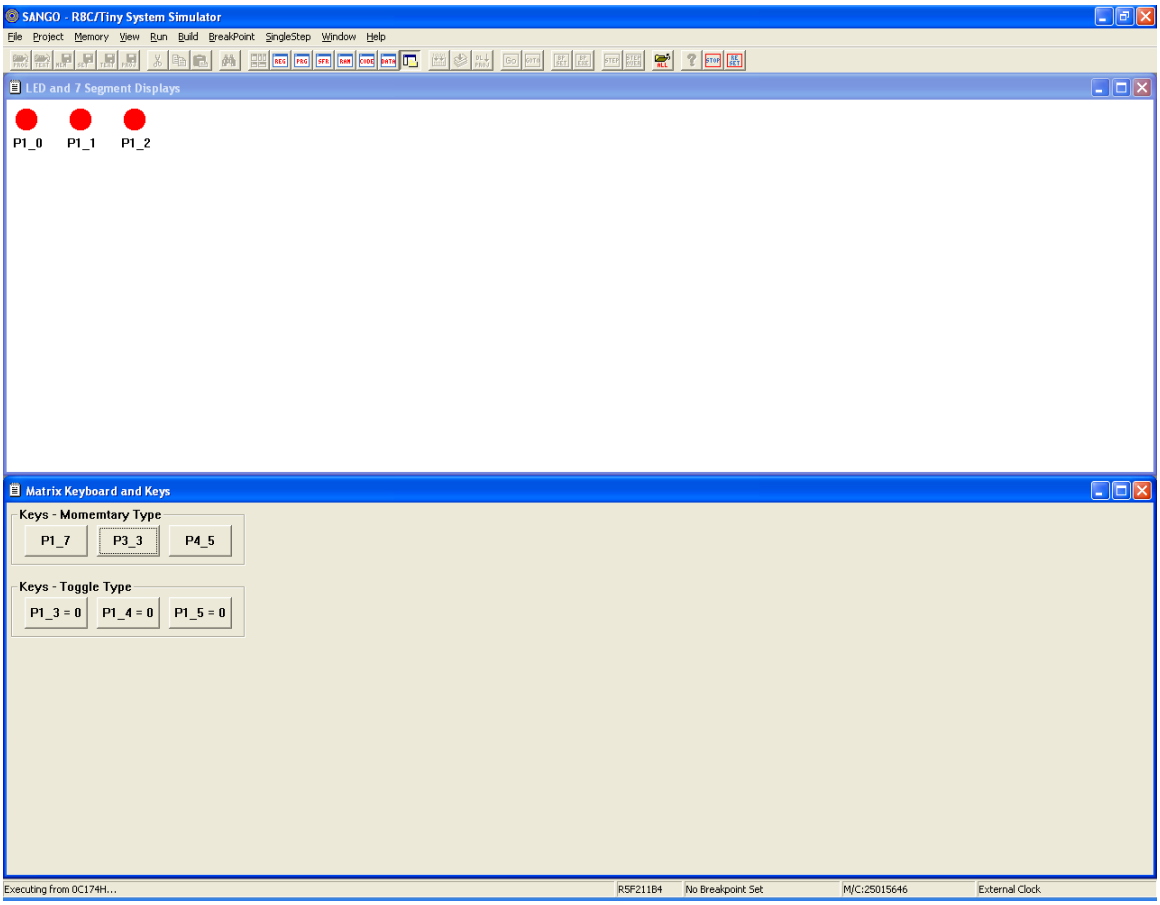
Use Topview Simulator to Verify the Design.

Open the project Demo11 in the R8C/Tiny System Simulator using **Open Project** option from **Project menu**. The project window opens up along with the Demo11.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.



Connect point LEDs to the port line P10 to P12 using LED module setting, connect three push button switches to the port lines P45, P17 & P33 and connect three numbers of toggle switches to the port lines P13 to P15. Open the LED and Keyboard window and arrange them as shown for better visibility.



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

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

Example 1 - Higher priority interrupt(INT1) is serviced during INT0 service.

1. Press P45(INT0) switch, LED1 will be ON in the interrupt service routine and the control will come out of the service routine only if switch toggle switch at port line P13 level is 1.
2. Now press P17(INT1)switch. INT1 is serviced and LED2 will be ON.
3. Change level of toggle switch at port line P14 to 1. The LED2 will be OFF. The control comes out of INT1 service and now it is in INT0.
4. Change level of toggle switch at P13 to 1. The LED2 will be OFF. The control comes out of INT0 service.

Example 2 - Lower priority interrupt(INT0) is not serviced during INT1 service.

1. Press push button switch at P17(INT1), LED2 will be ON in the interrupt service routine and the control will come out of the service routine only if toggle switch connected at P14 level is 1.
2. Now press push button switch P45(INT0). INT0 will not be serviced because it has a lower priority.
3. Change level of toggle switch at P14 to 1. The LED3 will be OFF. The control comes out of INT1 service and now INT0 is serviced and hence LED1 will be switched ON.
4. Change level of toggle switch connected to portline P13 to 1. The LED1 will be OFF. The control comes out of INT0 service.