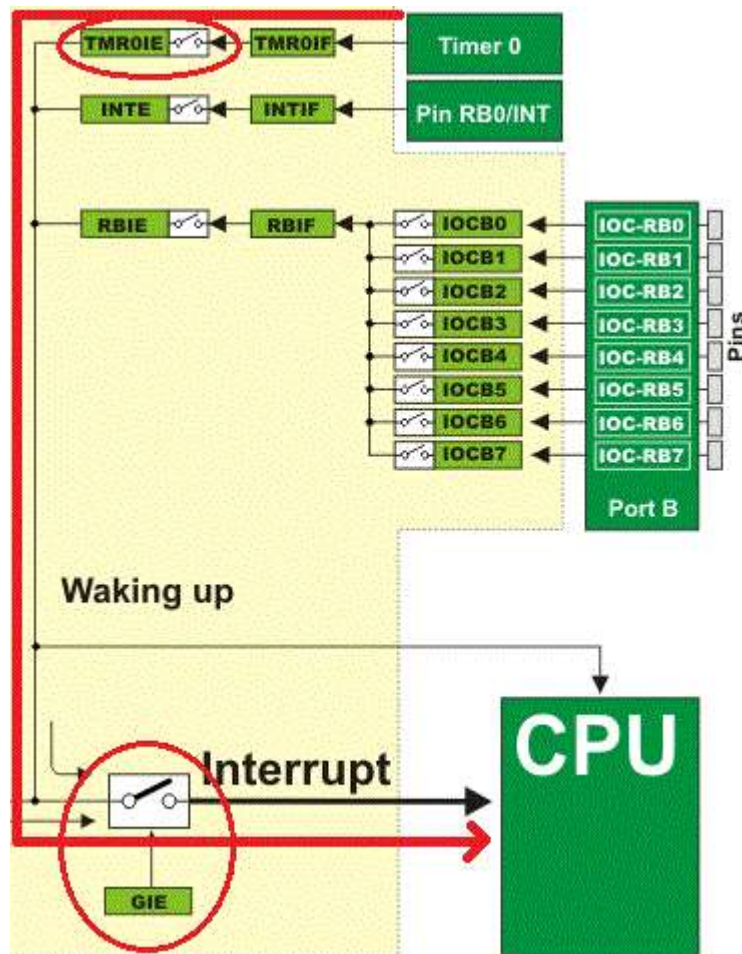


## Timer 0 Overflow Interrupt



TMR register ထဲက 8 bit တန်ဖိုးတွေ overflow ဖြစ်သွားတဲ့အခါမှာ TMR0IF ဆိုတဲ့ flag ကို enable ဖြစ်သွားစေပါတယ်။ TMR0IE နဲ့ GIE တို့မှာ enable လုပ်ထားမှသာ TMR overflow ဖြစ်တဲ့ အချိန် TMR0IF ထဲမှာ 1 ဖြစ်သွားပြီး ISR ကိုအလုပ်လုပ်စေမှာဖြစ်ပါတယ်။

TMR0 register ထဲမှာ overflow ဖြစ်မယ်ဆိုလို့ရှိရင် interrupt ဖြစ်ပါမယ်။ TMR0 ဆိုတာ Timer 0 module ရဲ့ register ဖြစ်ပါတယ်။ time တွေကို count လုပ်ပြီးမှတ်ထားနိုင်တဲ့ timer 0 ရဲ့ register တစ်ခုပါ။ TMR0 ဟာ 8 bit ဖြစ်တဲ့အတွက်ကြောင့် 0 ကနေ 255 အထိ တစ်ဖြေးဖြေး count လုပ်ပြီး တော့ တိုးတိုးသွားပါမယ်။ အဲဒီလိုမျိုး 0 ကနေ တဖြေးဖြေး count လုပ်လာလိုက်တာ 255 ပြည့်လို့ တိုး တဲ့ နေရာရောက်သွားရင် 8 bit ဘဲရှိတဲ့ အတွက် 256 ဆိုတာမတိုးနိုင်တော့ပါဘူး။ ဒါကို Timer Overflow ဖြစ်တယ်လို့ခေါ်ပါတယ်။ 255 ပြီး ထပ်တိုးလိုက်တဲ့အချိန်မှာ Timer က overflow ဖြစ်သွား ပြီးတော့ 0 ပြန်ဖြစ်သွားပါတယ်။

0 ကနေ 255 ရောက်အောင် တဖြည်းဖြည်းနဲ့ count လုပ်ပြီးတိုးတိုးသွားတယ်ဆိုတာကိုပြောခဲ့ ပါတယ်။ ဒါဆိုရင် 0 ကနေ 1 ဖြစ်ဖို့ 1 ကနေ 2 ဖြစ်ဖို့ ----- 254 ကနေ 255 ဖြစ်ဖို့ ဘယ်လောက် အချိန်

အတိုင်းအတာနဲ့ တိုးနေတာလဲဆိုတာပြောဖို့လိုပါတယ်။ အဲဒီ တစ်တိုးတဲ့ အချိန်ဟာ ကျွန်တော် တို့ PIC မှာတပ်ပြီးသုံးထားတဲ့ crystal အပေါ်မှာ မူတည်နေတာဖြစ်ပါတယ်။ အကယ်၍ 4 MHz Crystal ကိုသာသုံးခဲ့မယ် ဆိုရင် အဲဒီ frequency 4 MHz ကို အချိန်ပြန်ပြောင်းလိုက်တဲ့အခါမှာ  $1/f = 1/4\text{MHz} = 0.25 \mu\text{s}$  ရပါတယ်။ instruction တစ်ကြောင်းပြီးဖို့ရန်အတွက် clock cycle လေးခု လိုအပ်ပါတယ်။ ဒါကြောင့်မို့လို့ စောနကတွက်ခဲ့တဲ့ cycle တစ်ခုစာ  $0.25 \mu\text{s}$  ကို 4 နဲ့မြှောက်လိုက်တဲ့ အခါမှာ  $1 \mu\text{s}$  ရလာပါတယ်။ ဒါဟာ ကျွန်တော်တို့ရေးတဲ့ PIC Program ထဲမှာရှိတဲ့ instruction တစ်ခု ချင်းစီကို အလုပ်လုပ်ဖို့ရန်အတွက် ကြာတဲ့အချိန်ဖြစ်ပါတယ်။ Timer မှာလဲ ဒီလိုပါပဲ။ 0 ကနေ 1 ဖြစ်ဖို့ 1 ကနေ 2 ဖြစ်ဖို့ ----- 254 ကနေ 255 ဖြစ်ဖို့ ကြာတဲ့အချိန်ဟာ Crystal 4 MHz သုံးထား တယ်ဆိုရင်  $1 \mu\text{s}$  ကြာမှာဖြစ်ပါတယ်။ အကယ်၍ Crystal 20 MHz ကိုသုံးထားခဲ့မယ်ဆိုရင်တော့  $0.2 \mu\text{s}$  ကြာမှာဖြစ်ပါတယ်။

TMR0 မှာ 50 ဆိုတဲ့တန်ဖိုးရှိနေတယ်ဆိုပါစို့။ အဲဒီ 50 ဆိုတဲ့တန်ဖိုးကိုရဖို့ Crystal 4 MHz သုံးထားခဲ့တယ်ဆိုရင် အချိန်  $50 \mu\text{s}$  ကြာခဲ့ပါတယ်။

## REGISTER 2-2: OPTION\_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPUP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **RBPUP**: PORTB Pull-up Enable bit  
 1 = PORTB pull-ups are disabled  
 0 = PORTB pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG**: Interrupt Edge Select bit  
 1 = Interrupt on rising edge of INT pin  
 0 = Interrupt on falling edge of INT pin

bit 5 **T0CS**: Timer0 Clock Source Select bit  
 1 = Transition on T0CKI pin  
 0 = Internal instruction cycle clock ( $F_{OSC}/4$ )

bit 4 **T0SE**: Timer0 Source Edge Select bit  
 1 = Increment on high-to-low transition on T0CKI pin  
 0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit  
 1 = Prescaler is assigned to the WDT  
 0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>**: Prescaler Rate Select bits

### PS<2:0>: Prescaler Rate Select bits

Bit Value	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Timer တွေကို တိုးတိုးသွားဖို့ရန်အတွက် clock တွေလိုအပ်တယ်ဆိုတာကို အပေါ်မှာ ပြောခဲ့ ပါတယ်။ အဲဒီလိုမျိုး clock ကိုယူတဲ့နေရာမှာ Crystal တွေသုံးပြီးတော့ ယူလို့ရလို့။ အခြား အပြင်ကနေ လည်း clock source ပေးပြီးတော့ ယူလို့ရပါတယ်။ Internal Oscillator clock ကိုမသုံးဘဲနဲ့ ပြင်ပ T0CKI pin ကနေ clock source ကိုယူမယ်ဆိုရင် အဲဒီ Timer ကို Counter အနေနဲ့သုံးမယ်လို့ပြော လိုက်တာဖြစ်ပြီးတော့ Internal Clock ကနေ clock ယူမှသာ timer အနေနဲ့သုံးတာဖြစ်ပါတယ်။ T0CKI ဆိုတာ Timer 0 Clock Input ဖြစ်ပါတယ်။ အဲဒါကို OPTION Register ရဲ့ bit 5 မှာ သတ်မှတ် နိုင်ပါတယ်။ ဒီသင်ခန်းစာမှာ Timer အနေနဲ့သုံးမှာဖြစ်တဲ့အတွက်ကြောင့် T0CS (bit 5) ကို 0 ထားပြီး တော့ Internal Oscillator clock သုံးမယ်ဆိုပြီးတော့သတ်မှတ်လိုက်ပါမယ်။

ကျွန်တော်တို့ပုံမှန်ဆိုရင် clock လေးချက်ခုန့်မှ instruction တစ်ခုအလုပ်လုပ်လို့ပြီးတာဖြစ်ပါ တယ်။ ဆိုလိုတာက 4 MHz Crystal ရှိတဲ့ PIC မှာ 1 us ကြာမှ instruction တစ်ခုလုပ်လို့ပြီးတာပါ။ အဲဒီလိုမျိုးမလိုချင်လို့ ကိုယ်က 1 us အစား အခြားအချိန်တစ်ခုကြာမှ instruction တစ်ခု အလုပ်လုပ် တာပြီးစေချင်တယ်ဆိုရင် Prescale လုပ်ပေးရပါတယ်။ အပေါ်ကပုံမှာPrescale ဇယားပေးထားပါတယ်။ Option Register ရဲ့ bit 3 မှာ 0 ထားလိုက်ရင် အဲဒီ prescale ဇယားကိုTimer 0 ကနေသုံးနိုင်ပါတယ်။ 1 ထားလိုက်ရင် WDT ဆိုတဲ့ watchdog timer ကနေ သုံးနိုင်မှာဖြစ်ပါတယ်။ bit2-0 အထိက Prescale ဇယားထဲမှာပါတဲ့ တန်ဖိုးတွေကို စိတ်ကြိုက်ရွေးပြီးသတ်မှတ်နိုင်ဖို့ရန်အတွက်ဖြစ်ပါတယ်။

Prescaler Table ရဲ့ အလယ်ကော်လံမှာ Timer 0 အတွက်ထားနိုင်မယ့် Rate တွေကိုပေး ထားပါတယ်။ 1:2 ဆိုရင် 000 ဆိုပြီးတော့ Option register ရဲ့ bit 2-0 မှာပေးရမယ်။ 1:4 ဆိုရင် 001 ဆိုပြီးတော့ Option register ရဲ့ bit 2-0 မှာပေးရမယ်။ 1:8 ဆိုရင် 010 ဆိုပြီးတော့ Option register ရဲ့ bit 2-0 မှာပေးရမယ်။ 1:16 ဆိုရင် 011 ဆိုပြီးတော့ Option register ရဲ့ bit 2-0 မှာပေးရမယ်။ 1:32 ဆိုရင် 101 ဆိုပြီးတော့ Option register ရဲ့ bit 2-0 မှာပေးရမယ်။ 1:64 ဆိုရင် 101 ဆိုပြီးတော့ Option register ရဲ့ bit 2-0 မှာပေးရမယ်။ 1:128 ဆိုရင် 110 ဆိုပြီးတော့ Option register ရဲ့ bit 2-0 မှာပေးရမယ်။ 1:256 ဆိုရင် 111 ဆိုပြီးတော့ Option register ရဲ့ bit 2-0 မှာပေးရမယ်။

1:2 ပေးလိုက်တဲ့အခါမှာ အရင်လိုမျိုး 1 us နေမှ တစ်တိုးတာမဟုတ်ဘဲနဲ့ 2 u နေမှ တစ်တိုး တာဖြစ်ပါတယ်။ 1:32 ဆိုလည်း 32 us နေမှ timer တန်ဖိုးကို တစ်တိုးတာဖြစ်ပါတယ်။ ဒါကြောင့် prescale တန်ဖိုးမြင့်လာလေ timer ကို အချိန်အကြာကြီးမှတ်လို့ရလာလေဖြစ်ပါတယ်။

Option register ရဲ့ bit 3 မှာ 1 ပေးခဲ့တယ်ဆိုရင် Prescaler ဇယားကို WDT(Watch Dog Timer) ကနေဘဲယူသုံးမှာဖြစ်တဲ့အတွက်ကြောင့် Timer0 ကနေ အဲဒီ ဇယားကိုသုံးလို့မရတော့ပါဘူး။ ဒါကြောင့် Timer0 တစ်တိုးရေတွက်မယ့် အချိန်ကို ပုံမှန် 1 us ဘဲရှိနေချင်တယ်ဆိုရင် အဲဒီ bit 3 နေရာ မှာ 1 ထားခဲ့လိုက်ပါ။

#### REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TOIE	INTE	RBIE <sup>(1,3)</sup>	TOIF <sup>(2)</sup>	INTF	RBIF
bit 7							bit 0

bit 7	<b>GIE:</b> Global Interrupt Enable bit 1 = Enables all unmasked interrupts 0 = Disables all interrupts
bit 6	<b>PEIE:</b> Peripheral Interrupt Enable bit 1 = Enables all unmasked peripheral interrupts 0 = Disables all peripheral interrupts
bit 5	<b>TOIE:</b> Timer0 Overflow Interrupt Enable bit 1 = Enables the Timer0 interrupt 0 = Disables the Timer0 interrupt
bit 4	<b>INTE:</b> INT External Interrupt Enable bit 1 = Enables the INT external interrupt 0 = Disables the INT external interrupt
bit 3	<b>RBIE:</b> PORTB Change Interrupt Enable bit <sup>(1,3)</sup> 1 = Enables the PORTB change interrupt 0 = Disables the PORTB change interrupt
bit 2	<b>TOIF:</b> Timer0 Overflow Interrupt Flag bit <sup>(2)</sup> 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow
bit 1	<b>INTF:</b> INT External Interrupt Flag bit 1 = The INT external interrupt occurred (must be cleared in software) 0 = The INT external interrupt did not occur
bit 0	<b>RBIF:</b> PORTB Change Interrupt Flag bit 1 = When at least one of the PORTB general purpose I/O pins changed state (must be cleared in software) 0 = None of the PORTB general purpose I/O pins have changed state

INTCON လို့ခေါ်တဲ့ Interrupt Control register မှာ GIE, PEIE , TOIE နဲ့ TOIF တို့ကိုလည်း enable လုပ်ပေးခဲ့ဖို့လိုအပ်ပါတယ်။

‘10000 us တိုင်းမှာ Interrupt ဖြစ်ပေါ်စေတဲ့ ပရိုဂရမ်ကိုအောက်မှာ ရေးသားထားပါတယ်။

**Join our PIC Microcontroller Online Class for more detail**

**:)**