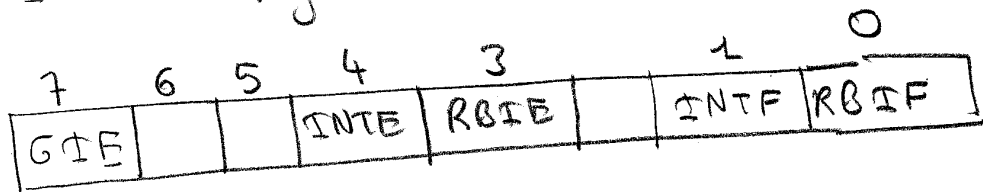


①

## INTCON Register



RBIE → RB Port Change Interrupt Enable bit.  
1 = Enables the RB port change interrupt  
0 = Disables the RB port change interrupt

RBIF → RB Port Change Interrupt Flag bit.  
1 = At least one of the RB7:RB4 pins changed state  
0 = None of the RB7:RB4 pins have changed state

→ command

sleep → The processor is put into sleep mode with the oscillator stopped.

The device can wake-up from sleep through one of the following events

- 1) External RESET input on MCLR pin
- 2) WDT wake up (if WDT was enabled)
- 3) Interrupt from RB0/INT pin

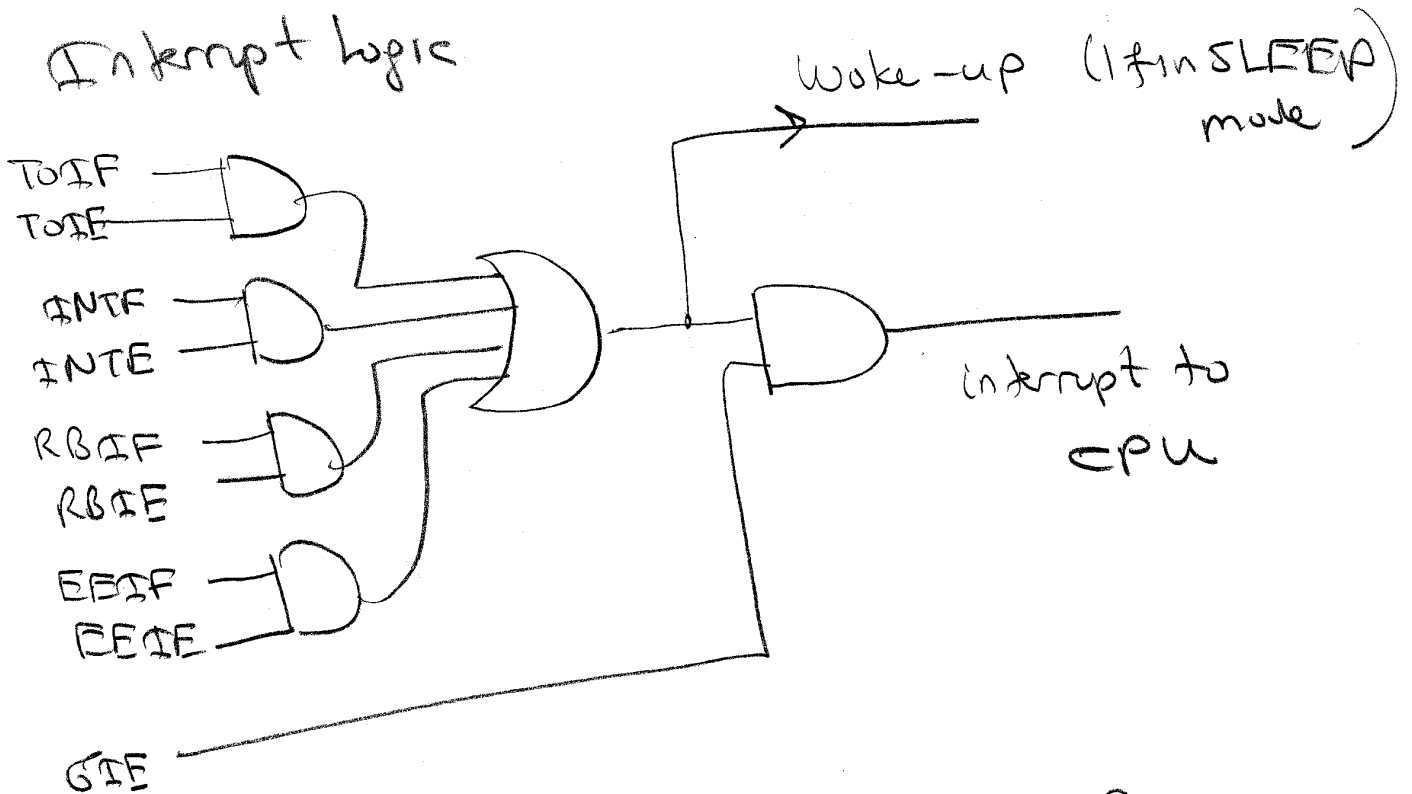
RB port change, or data EEPROM write complete

②

## PORTB Interrupts

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<5>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<6>)

### Interrupt Logic



- Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits.
- The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid infinite interrupt requests.

③

- The INT interrupt can wake the processor from SLEEP only if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether the processor branches to the interrupt vector following wake-up.

Ex<sup>o</sup>

The following program checks all the interrupt sources and executes the related interrupt-subroutines

```
org 0x000  
goto main  
org 0x004,  
btfsc INTCON, INTF  
goto intsubr-RB;  
btfsc INTCON, RBIF;  
goto intsubr-RBO-INT;  
btfsc INTCON, TOIF;  
goto intsubr-Timer;  
btfsc INTCON, EEIF;  
goto intsubr-Eprom;  
goto int-end-sbr
```

two interrupt sources may have received signal at the same time

4

main

```

bsf INTCW, GIE;
bsf INTCW, RBIE;
bsf INTCW, INTE;
bsf INTCW, TOIE;

```

} Enable  
all  
the  
interrupt  
sources

```

loop goto loop;

```

intsbr\_RB

```

bcf INTCW, RBIF;

```

```

goto 0x004;

```

intsbr\_RB0\_INT

```

bcf INTCW, INTF;

```

```

goto 0x004;

```

intsbr\_Timer

```

bcf INTCW, TOIF;

```

```

goto 0x004;

```

intsbr\_Epnum

```

bcf INTCW, EEIF;

```

```

goto 0x004;

```

int\_end\_sbr

```

retfie;

```