



**MICROCHIP**

# **dsPIC30F Peripheral Module**

## **Input Capture Module**



# Input Capture

- Up to Eight Input Capture Channels
- Captures 16-bit timer value
  - ❖ At 30 MIPS resolution = 33 ns ( $T_{cy}$ )
  - ❖ At 30 MIPS with 16x pre-scale = 2.1 ns
- 4 deep buffer for each capture input
  - ❖ Interrupt on 1- 4 capture events
  - ❖ FIFO buffer overflow status
  - ❖ FIFO buffer empty status

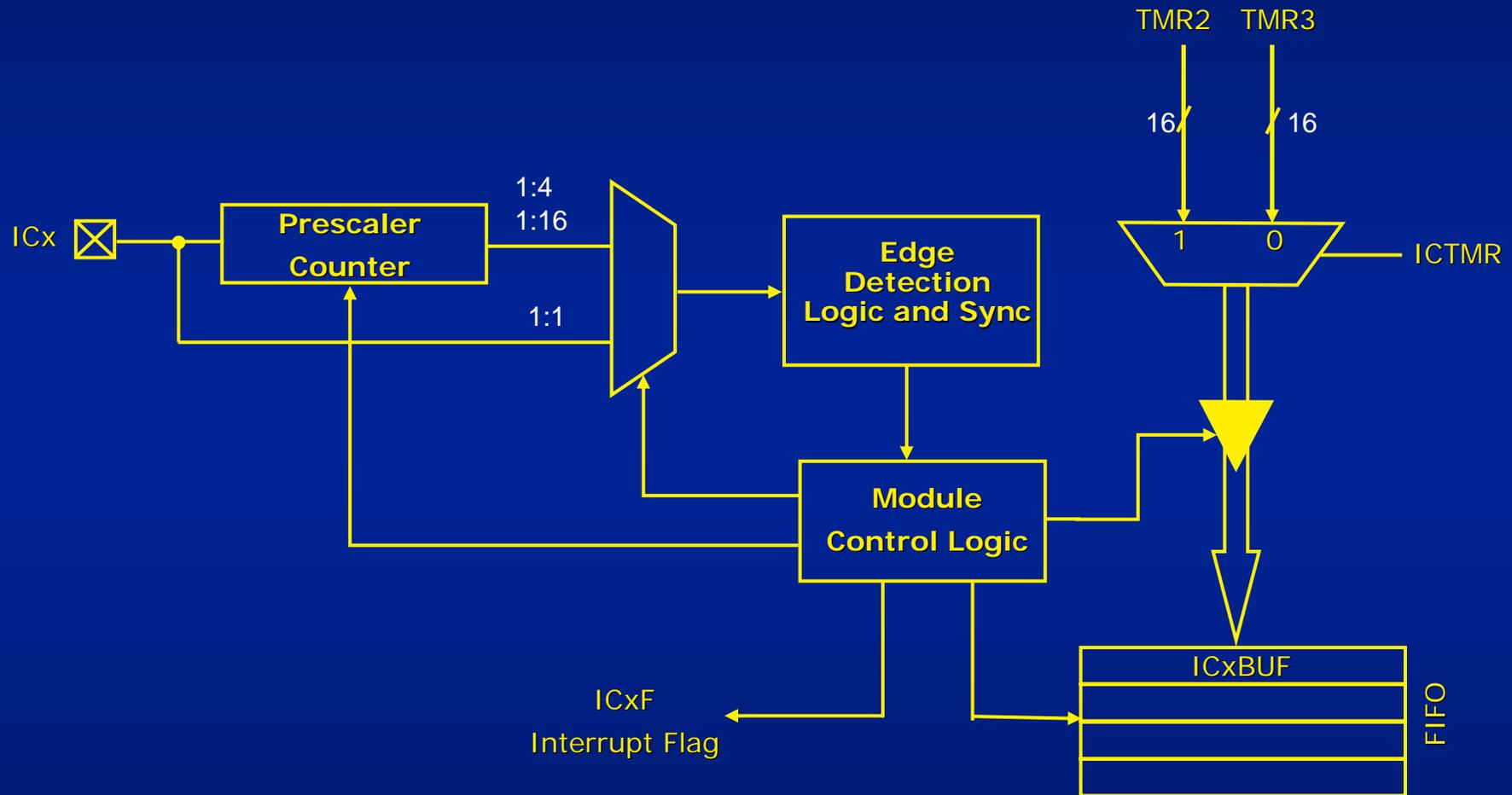


**MICROCHIP**

# Input Capture

- Timer 2 or Timer 3 as timebase
- Capture on:
  - ❖ ↑ edge at ICx pin
  - ❖ ↓ edge at ICx pin
  - ❖ Every 4th ↑ edge at ICx pin
  - ❖ Every 16th ↑ edge at ICx pin
  - ❖ ↑ edge and ↓ edge
    - ❖ Very useful for pulse and frequency measurement
    - ❖ Interface to hall sensors for rotor position feedback
    - ❖ Autobaud support for UART communications

# Input Capture Block Diagram





# Input Capture Control Register

- ICxCON
  - ❖ ICSIDL  $\Rightarrow$  Stop in Idle mode
  - ❖ ICTMR  $\Rightarrow$  Time Base Select for Input Capture
  - ❖ ICI<1:0>  $\Rightarrow$  Capture events per Interrupt select
  - ❖ ICOV  $\Rightarrow$  FIFO buffer overflow status
  - ❖ ICBNE  $\Rightarrow$  FIFO buffer Not Empty status
  - ❖ ICM<2:0>  $\Rightarrow$  Input Capture mode select

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
-	-	ICSIDL	-	-	-	-	-
bit15	14	13	12	11	10	9	bit8
R/W-0	R/W-0	R/W-0	R-0, HC	R-0, HC	R/W-0	R/W-0	R/W-0
ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>		
bit7	6	5	4	3	2	1	bit0

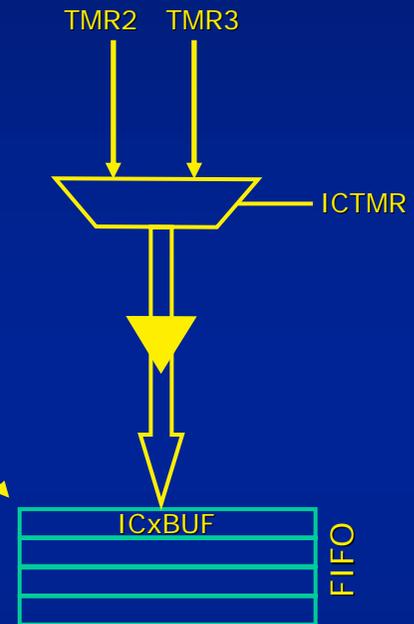


# Input Capture Mode Select

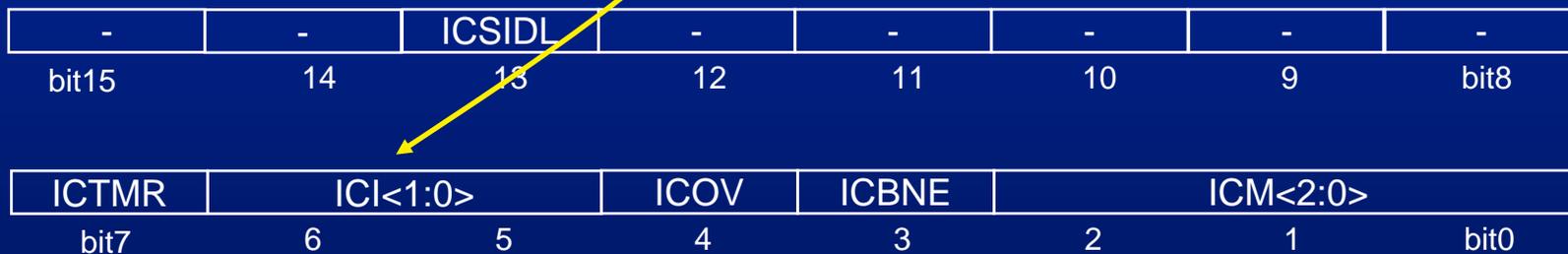
- ICM<2:0> : Input Capture Mode Select bits
  - ❖ 000 : Input Capture turned off
  - ❖ 001 : Capture every edge change
  - ❖ 010 : Capture every falling edge is coming
  - ❖ 011 : Capture every rising edge is coming
  - ❖ 101 : Capture every 4th rising edge
  - ❖ 110 : Unused (Disable)
  - ❖ 111 : Input Capture function as interrupt only, when device is in the SLEEP or IDLE mode (rising edge detect onlt)

# Input Capture FIFO

- 4 deep buffer for each capture input
- Interrupt on 1-4 capture events
- FIFO buffer overflow status
- FIFO buffer empty status



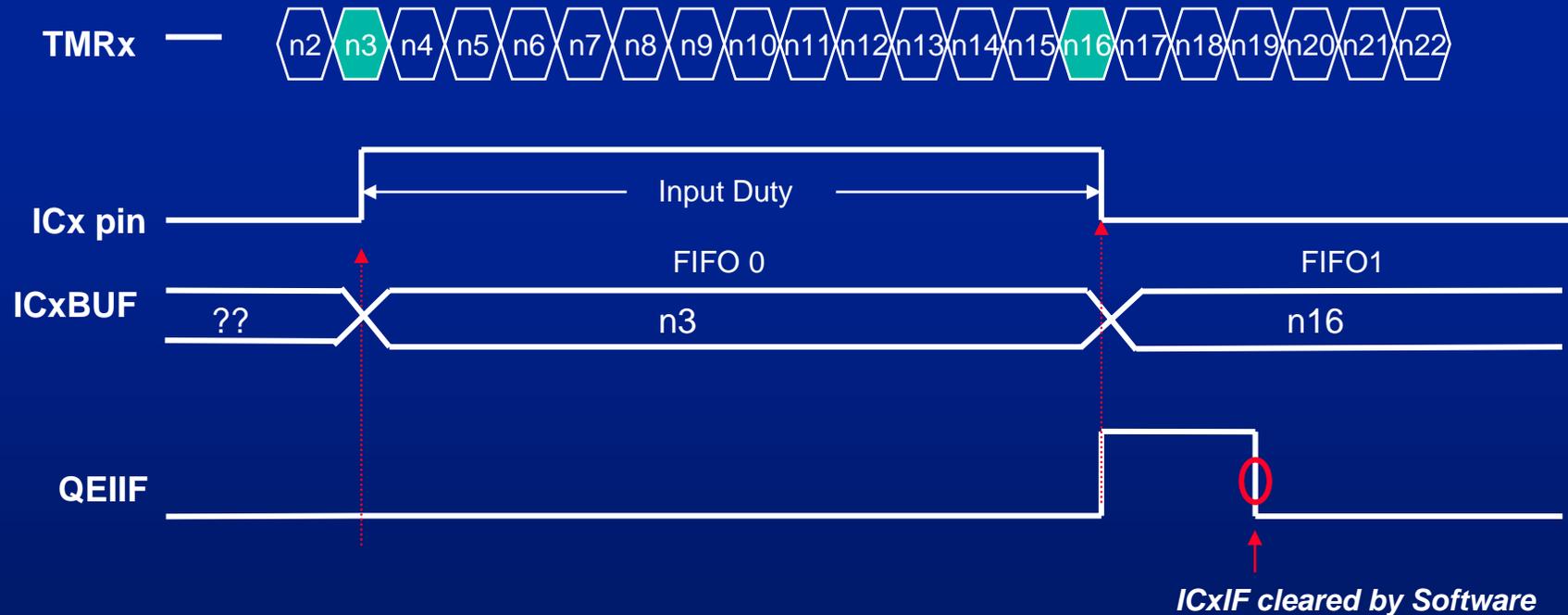
## ICxCON SFR





# Edge Detection Mode Calculate the Input Duty

- $ICM\langle 2:0 \rangle = 001$ , for capture every edge change
- $ICI\langle 1:0 \rangle = 01$ , for Interrupt on every second event
- $Input\ Duty = (n16 - n3) * TMRx\ input\ cycle$

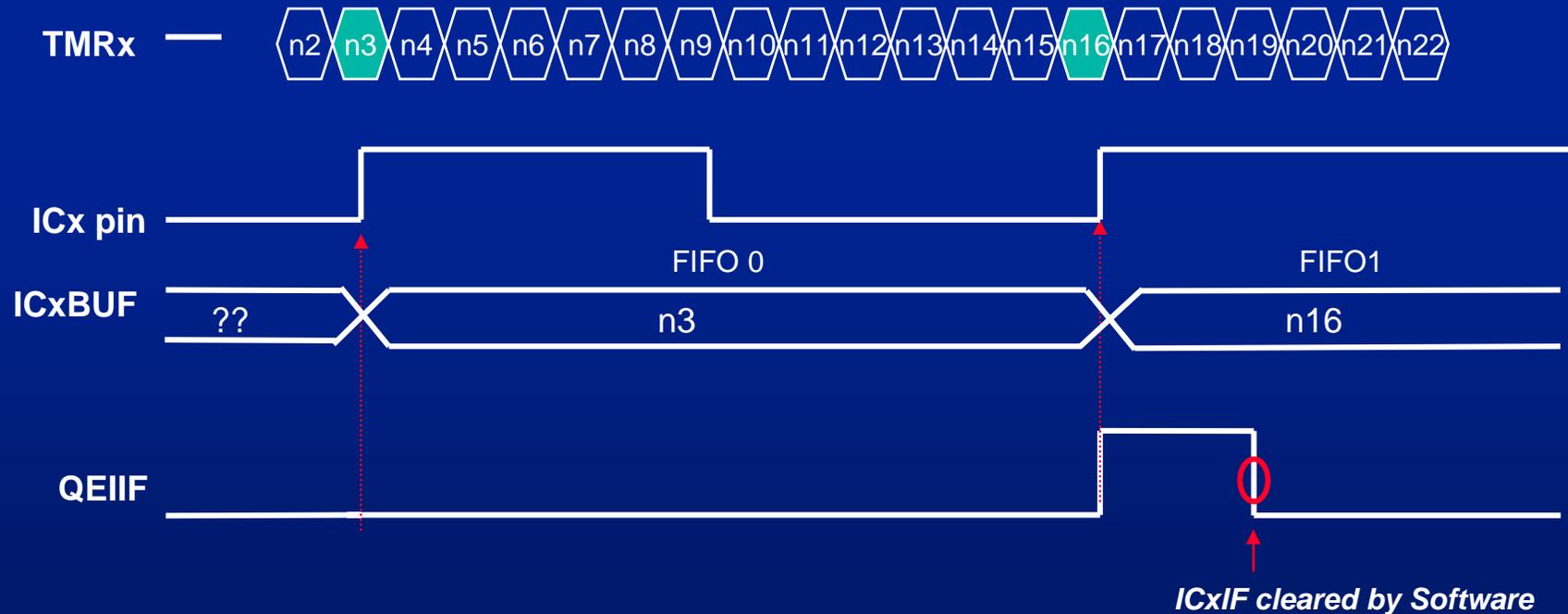




MICROCHIP

# Edge Detection Mode Calculate the Input Frequency

- $ICM\langle 2:0 \rangle = 011$ , for capture every rising edge
- $ICI\langle 1:0 \rangle = 01$ , for Interrupt on every second event
- $Input\ Freq. = 1 / [(n16 - n3) * TMRx\ input\ cycle]$





**MICROCHIP**

# Input Capture Lab1

- Input Capture 7 (IC7/RB4/AN4) connect to PIC12F675 to measure the input frequency
- Close the SW1 of DIPSW4 (IC7 tie to QEA output)
- Use the Input Capture mode with 2 FIFO interrupt to capture the input signal
- Every 200mS show the frequency & period on LCD
- Change VR3 to see frequency result, use a scope to conform the input frequency (0.8K ~ 3.8KHz)

Freq. = 1238 Hz

Period= 807 uS

LCD Module display items



**MICROCHIP**

# Hint Lab1

## Getting the Time Period

- Input Capture 7 (IC7) Interrupt Function

```
void _ISR_IC7Interrupt(void)           // Interrupt Function for the IC7
{
    ReadCapture7( &timer_edge[0]);     // Read Timer count from FIFO 0
    ReadCapture7( &timer_edge[1]);     // Read Timer count from FIFO 1
    Int_flag = 1;                       // Set IC7 process Flag
    IFS1bits.IC7IF = 0;                 // Clear the IC7 interrupt flag
}
```



**MICROCHIP**

## Hint Lab3

# Calculate the Time Period

- Need to check value for both FIFO 0 and FIFO 1 to make sure the rollover issue.

```
while (!Int_flag);           // Get two input signal edge
DisableIntIC7;              // Disable Interrupt of Capture 7
Int_flag = 0;

/* calculate time count between two capture events */

if ( timer_edge[1] >= timer_edge[0])
    period = timer_edge[1] - timer_edge[0];
else
    period = 65536 + timer_edge[1] - timer_edge[0];

frequency= FCY/period;      // Calculate the frequency
period = period / 14.745600 ; // Calculate the period
```