Lab 07-2: Timer

What is Timer ?

- In general, Timer is a module that counts the number of clocks inside itself.
- The term "Timer" generally has two names, namely "Timer" and "Counter"
  - Period of clock is unknown: Counter
  - Period of clock is known: Timer
- The Timer of 8-bit MCU is incremental, which means it starts counting from 0. The module can set the count to notify the CPU after a certain number of clock cycles.

Interrupt & Timer

- Notification: It is Interrupt Flag. When it counts to the setting value, the module will set the interrupt flag as 1 and send Interrupt Request to CPU.
  - If GIE is clear, it relies on process to check whether the event has been occurred or not.
  - If GIE is set, it branches to ISR automatically.
- For example: the interrupt flag of Timer0 can be controlled at INTCON register.

REGISTER 9-1: 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/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

PIC18F4520 Timer Module

- **Timer0**:
  - Timer0 can be set as 8-bit or 16-bit mode.
  - The source of clock can be set as internal or external source.
  - Timer0 has an 8-bit prescalar.
  - When it comes to overflow, from FFh to 00h in 8-bit mode (or FFFFh to 0000h in 16-bit mode), it interrupts.

PIC18F4520 Timer0

REGISTER 11-1: T0CON: TIMER0 CONTROL 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
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT**: Timer0 8-Bit/16-Bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

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**: Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS<2:0>**: Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

Prescalar

- The counting range of 16-bit Timer is 0~65535. If the requirement of counting range exceed, it can be accomplished by prescalar.
- Prescalar can be used to widen the counting range.
- For example:
    - If the prescalar is set as ÷8 (1:8), then only one clock can access to TMRx for every 8 clocks, which means when TMRx counts to 1000, 8000 clocks has been generated.

PIC18F4520 Timer1 & Timer3

- **Timer1 & Timer3**:
  - 16-bit mode counter or timer. (using TMR1H:TMR1L)
  - Concatenated by two readable and writable 8-bit counter
  - The source of clock can be set as internal or external source.
  - There are four options for prescalar: ÷1, ÷2, ÷4, ÷8
  - When it comes to overflow, from FFFFh to 0000h, it interrupts.

Timer1 Control Register: T1CON

- Timer1 Clock Source:
  - Internal Clock (FOSC / 4)
  - Default frequency: 1MHz / 4
- T1CKPS<1:0>:
  - Timer Input Clock Prescale Select bit
- TMR1CS:
  - Timer1 Clock Source Select bit
- TMR1ON:
  - Timer1 On bit

REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of TImer1 in one 16-bit operation
0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 **T1RUN:** Timer1 System Clock Status bit
1 = Device clock is derived from Timer1 oscillator
0 = Device clock is derived from another source

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit
1 = Timer1 oscillator is enabled
0 = Timer1 oscillator is shut off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Select bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

PIC18F4520 Internal Oscillator

- Default Internal Oscillator Frequency: 1 MHz

REGISTER 2-2: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R/W-1	R/W-0	R/W-0	R(1)	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IDLEN:** Idle Enable bit
1 = Device enters an Idle mode on SLEEP instruction
0 = Device enters Sleep mode on SLEEP instruction

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits
111 = 8 MHz (INTOSC drives clock directly)
110 = 4 MHz
101 = 2 MHz
100 = 1 MHz(3)
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (from either INTOSC/256 or INTRC directly)(2)

bit 3 **OSTS:** Oscillator Start-up Timer Time-out Status bit(1)
1 = Oscillator Start-up Timer (OST) time-out has expired; primary oscillator is running
0 = Oscillator Start-up Timer (OST) time-out is running; primary oscillator is not ready

bit 2 **IOFS:** INTOSC Frequency Stable bit
1 = INTOSC frequency is stable
0 = INTOSC frequency is not stable

bit 1-0 **SCS<1:0>:** System Clock Select bits
1x = Internal oscillator block
01 = Secondary (Timer1) oscillator
00 = Primary oscillator

Note 1: Reset state depends on state of the IESO Configuration bit.
2: Source selected by the INTSRC bit (OSCTUNE<7>), see text.
3: Default output frequency of INTOSC on Reset.

Registers Associated with Timer1

- TMR1L, TMR1H: registers for Timer1. When it comes to overflow, it interrupts.
- PTR1: (TMR1IF) TMR1 Overflow Interrupt Flag bit
- PIE1: (TMR1IE) TMR1 Overflow Interrupt Enable bit
- PTR1: (TMR1IP) TMR1 Overflow Interrupt Priority bit

TABLE 12-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TMR1L	Timer1 Register Low Byte								50
TMR1H	Timer1 Register High Byte								50
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	50

PIC18F4520 Timer2

- **Timer2**:
  - An 8-bit mode timer or counter having both prescalar and postscalar.
  - Timer2 automatically increment by 1 and compare with the presetting value. If the value is the same, Timer2 sends the signal to postscalar or interrupts. After sending signal, Timer2 clears the register and restarts.

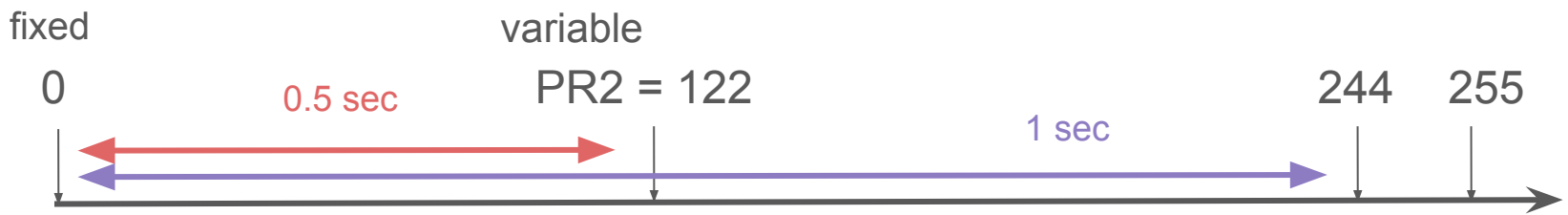
How to get 0.5 second using Timer2

Assume that postscalar and prescalar of Timer 2 are both 1:16 and FOSC = 250kHz

instruction clock frequency = FOSC / 4 = 250kHz / 4 = 62.5 kHz => #instr clocks / second = 62.5k

increment / second = (#instr clocks / second) / 16 / 16 = (62.5k) / 16 / 16 = 244

increment in 0.5 second = (# increment / second) / 2 = 122



How to get 0.5 second using Timer1

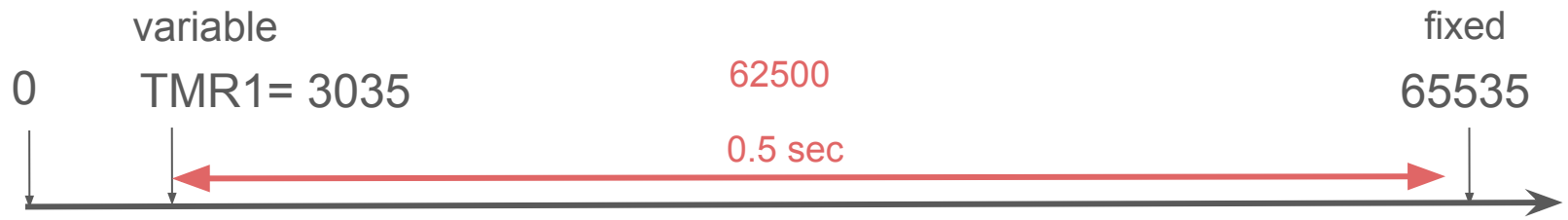
Assume that prescalar Timer1 is 1:2 and FOSC = 1MHz

instruction clock frequency = FOSC / 4 = 1MHz / 4 = 250 kHz => #instr clocks / second = 250k

increment / second = (#instr clocks / second) / 2 = (250k) / 2 = 125000

increment in 0.5 second = (# increment / second) / 2 = 62500

TMR1 = 65535 - 62500 = 3035 (0xBDB in hexadecimal)



Summary

	Timer0	Timer1	Timer2	Timer3
Size	8 or 16 bits	16 bits	8 bits	16 bits
Prescalar options	8 options (1:256)	4 options (1:8)	3 options (1:16)	4 options (1:8)
Postscalar options	None	None	16 options (1:16)	None
When to Interrupt	Overflow	Overflow	Equal to presetting value	Overflow

Reference

PIC18F4520 Datasheet: https://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf