COMP551 – Interfacing

Fall 2017

Lab 5

Programming PIC18 Counters

Students:		
Student ID's:		
Section:	**01** **02**	**03** **04**

NOTE: Labs are due at the start of the next lab period. Only submit one lab per group of two students.

5. Introduction:

In lab 4, you used the PIC18 Timers to program time delays. The PIC18 Timers can also be used as counters to count events happening outside the PIC18.

When the timer is used as a timer, the PIC18's crystal is used as the frequency source. When the timer is used as a counter, a pulse, external to the PIC18 increments the TH and TL registers. In counter mode, the T0CON, TRM0H and TRM0L registers are the same as the ones used when the timer is used in timer mode.

5.1 Counter Programming

5.1.1 T0CS bit in T0CON Register

The T0CS (Timer0 Clock Source) bit in the T0CON register (Figure 5-1) decides the source of the clock for the timer. If T0CS = 0, the timer gets pulses from the crystal connected to the OSC1 and OSC2 pins (Fosc/4). If T0CS = 1, the timer is used as a counter and gets pulses from outside the PIC18. Therefore, when T0CS = 1, the counter counts up as pulses are fed into pin RA4 (PORTA.4). The pin is called T0CKI (Timer0 Clock Input).

FIGURE 5-1: T0CON (Timer 0 Control) Register

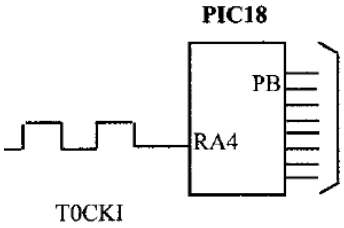
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

TMR0ON	D7	Timer0 ON and OFF control bit 1 = Enable (start) Timer0 0 = Stop Timer0
T08BIT	D6	Timer0 8-bit/16-bit selector bit 1 = Timer0 is configured as an 8-bit timer/counter. 0 = Timer0 is configured as a 16-bit timer/counter.
T0CS	D5	Timer0 clock source select bit 1 = External clock from RA4/T0CKI pin 0 = Internal clock (Fosc/4 from XTAL oscillator)
T0SE	D4	Timer0 source edge select bit 1 = Increment on H-to-L transition on T0CKI pin 0 = Increment on L-to-H transition on T0CKI pin
PSA	D3	Timer0 prescaler assignment bit 1 = Timer0 clock input bypasses prescaler. 0 = Timer0 clock input comes from prescaler output.
T0PS2:T0PS0	D2D1D0	Timer0 prescaler selector
	0 0 0 = 1:2	Prescale value (Fosc / 4 / 2)
	0 0 1 = 1:4	Prescale value (Fosc / 4 / 4)
	0 1 0 = 1:8	Prescale value (Fosc / 4 / 8)
	0 1 1 = 1:16	Prescale value (Fosc / 4 / 16)
	1 0 0 = 1:32	Prescale value (Fosc / 4 / 32)
	1 0 1 = 1:64	Prescale value (Fosc / 4 / 64)
	1 1 0 = 1:128	Prescale value (Fosc / 4 / 128)
	1 1 1 = 1:256	Prescale value (Fosc / 4 / 256)

Note: Use the Synchronous Clock Stimulus feature of the MPLAB Simulator to simulate the square wave. See the next section for information on how to use this feature.

Exercises:

1. Assume that an external clock signal is being fed into pin T0CKI (RA4). Write a program to use Counter0 in 8-bit mode to count up and display the state of the TMR0L count on PORTB. Start the count at 0H. Use the MPLAB simulator to verify the proper operation of the program.



2. Assume that an external clock signal is being fed into pin T1CKI. Write a program to use Counter1 in 16-bit mode to count the pulses and display the TMR0H and TMR0L registers on PORTB and PORTD, respectively. Start the count at 0H. Use the MPLAB simulator to verify the proper operation of the program.

Submit a hardcopy of the source code for each program, as well as a Logic Analyzer screenshot demonstrating the proper output of each program.

Using Stimulus

During simulation, the program being executed by the simulator may require stimuli from the outside. Stimulus is the simulation of hardware signals/data into the device. This stimulus could be a level change or a pulse to an I/O pin of a port. It could be a change to the values in an SFR (Special Function Register) or other data memory.

In addition, stimulus may need to happen at a certain instruction cycle or time during the simulation. Alternately, stimulus may need to occur when a condition is satisfied; for example, when the execution of program has reached a certain instruction address during simulation.

Basically, there are two types of stimulus:

- **Asynchronous** - A one-time change to the I/O pin or RCREG triggered by a firing button on the stimulus GUI within the MPLAB IDE.
- **Synchronous** - A predefined series of signal/data changes to an I/O pin, SFR or GPR (e.g., a clock cycle).

To define when, what and how external stimuli are to happen to a program, you would use the Stimulus Dialog (see below) tabs to create both asynchronous and synchronous stimulus on a stimulus workbook.

Stimulus Dialog

Use the Stimulus dialog to create asynchronous or synchronous stimuli. The Stimulus dialog allows you to enter stimulus information which is saved in a file called a workbook.

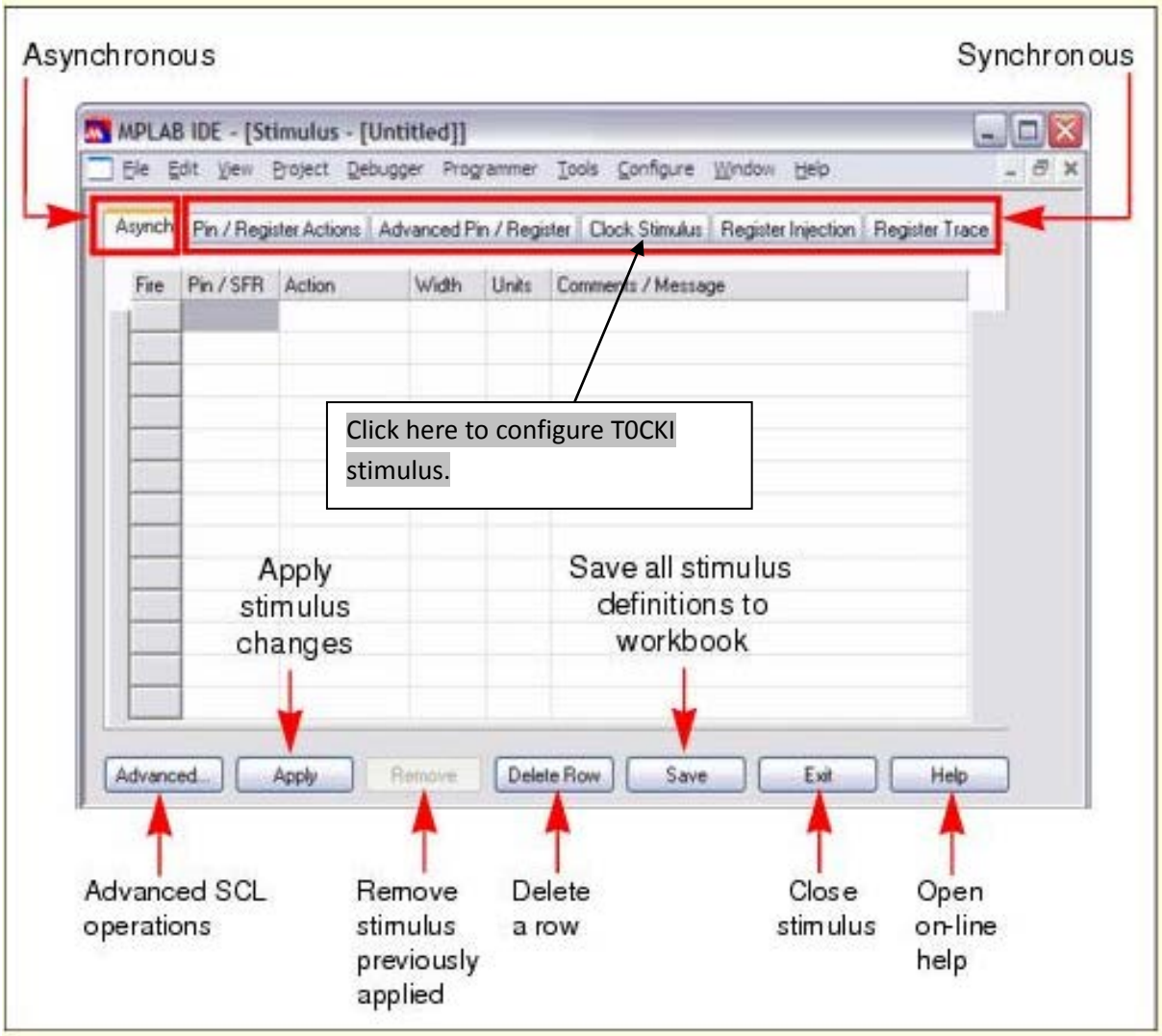
To open a new workbook, select ***Debugger->Stimulus->New Workbook***.

The Stimulus dialog (see Figure: Stimulus Dialog) has these tabs.

- Asynch Tab
- Pin/Register Actions Tab
- Advanced Pin/Register Tab
- **Clock Stimulus Tab**
- Register Injection Tab
- Register Trace Tab

Note: The Stimulus dialog must be open for stimulus to be active during simulation.

Figure: Stimulus Dialog



Clock Stimulus Tab

Pulse (high and low) values applied to a pin, is clocked stimulus as shown in Table: Definitions of Clock Stimulus. Clocked stimulus may be entered here.

Figure: Definition of Clock Stimulus

Item	Definition	
Label	Unique name of the clock stimulus you are specifying (optional).	
Pin	Choose the pin on which you will apply clocked stimulus.	
Initial	Enter the initial state of the clocked stimulus, either low or high.	
Low Cycles	Enter a value for the number of low cycles in a clock pulse.	
High Cycles	Enter a value for the number of high cycles in a clock pulse.	
Begin	Click here to activate the selection in the Begin section:	
	Always (default)	Begin stimulus immediately on program run.
	PC=	Begin stimulus when the program counter equals the entered value.
	Cycle=	Begin stimulus when the instruction cycle count equals the entered value. *absolute time* (cyc) - relative to the beginning of simulation. *after last clock* (cyc+) - relative to the End Cycle (see below). After the last clock relative to the End Cycle (see below)
	Pin=	Begin stimulus when the selected pin has the selected value (low or high).
End	Click here to activate the selection in the End section:	
	Never (default)	Apply stimulus until program halt.
	PC=	End stimulus when the program counter equals the entered value.
	Cycle=	End stimulus when the instruction cycle count equals the entered value. *absolute time* (cyc) - relative to the beginning of simulation. *from clock start* (cyc+) - relative to the Begin Cycle (see above).
	Pin=	End stimulus when the selected pin has the selected value (low or high).
Comments	Add descriptive information about the stimulus.	