11014 MS1

MPLAB® Simulator Fundamentals

Class Objective

- **When you finish this class you will be able to:**
  - Use MPLAB® Simulator to **test and debug** firmware
  - Create and apply simple **stimulus**
  - Use **Logic Analyzer** and **Instruction Trace**

Agenda

- **Understanding MPLAB® Simulator**
  - (demo)
- **Stimulus**
  - Asynchronous
  - Simple Synchronous
  - (Lab 1)
- **Tools of Simulator**
  - Stopwatch
  - Trace
  - Logic Analyzer
  - (Lab 2)

MPLAB® IDE and Simulator

What is MPLAB® Simulator?

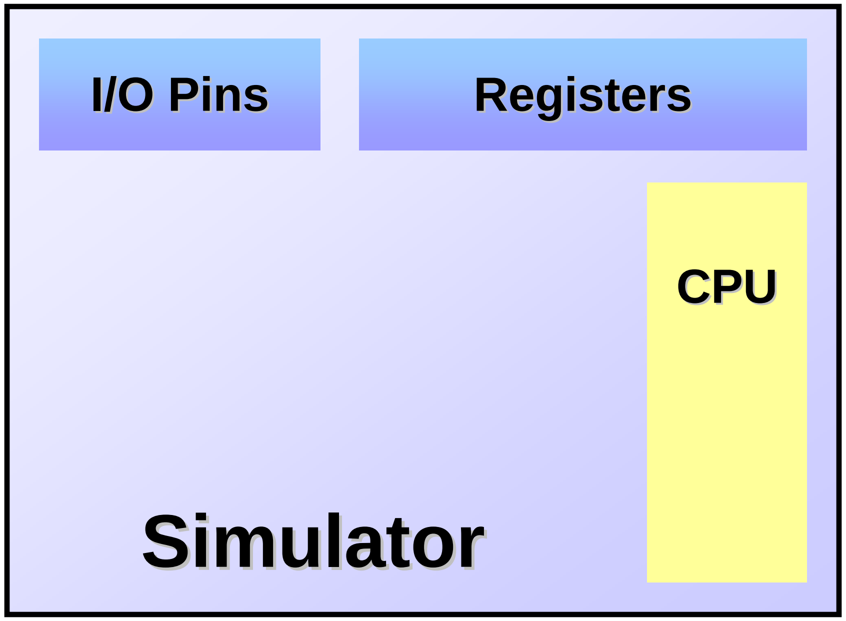
- One of the MPLAB® debuggers
- Supports all PIC® MCU & dsPIC® DSC families
  - Baseline
  - Mid-Range
  - PIC18
  - PIC24
  - dsPIC30
  - dsPIC33
- Speed
- Sophisticated stimulus engine

Why MPLAB® Simulator?

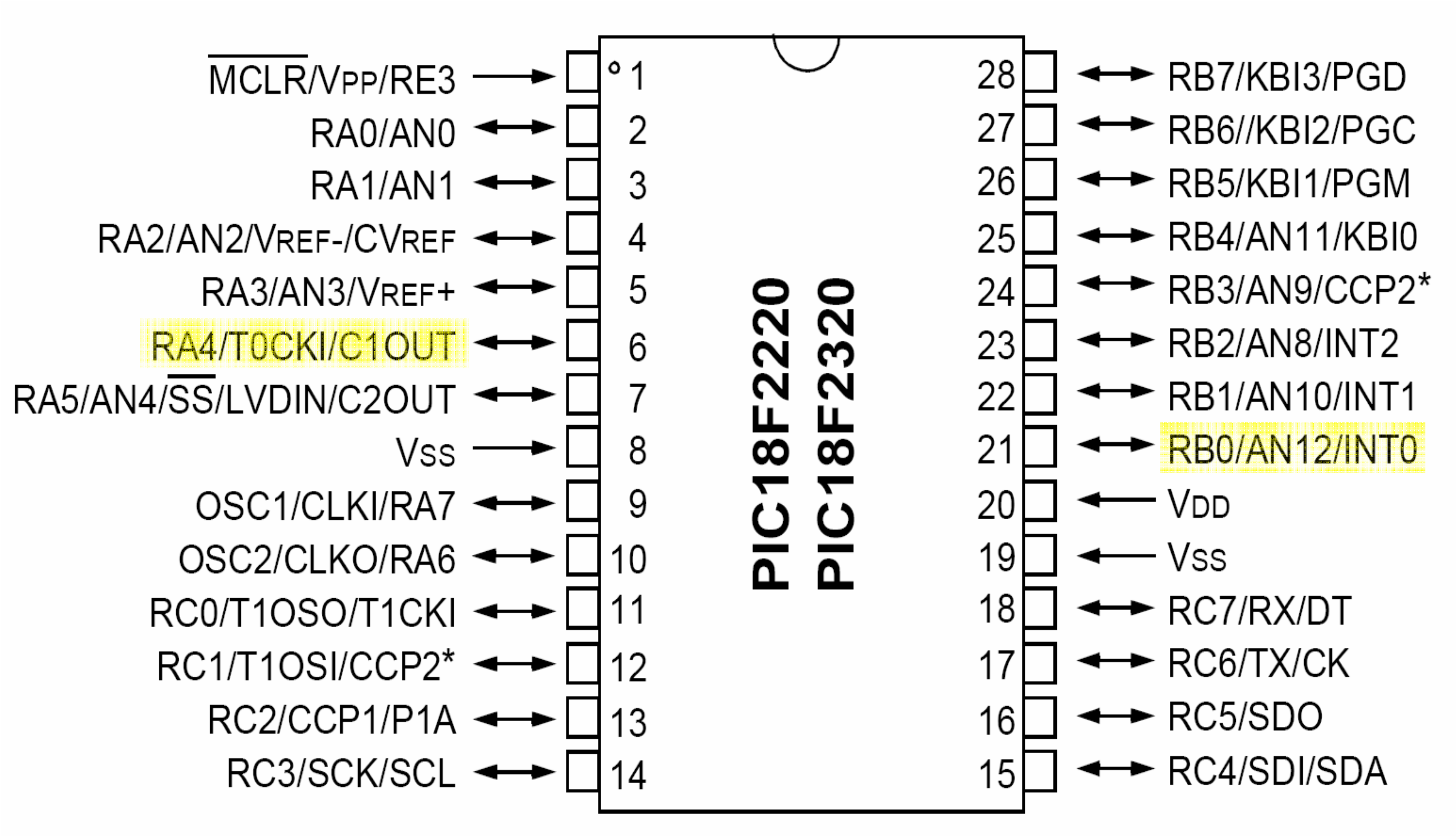
- **Pure software debugger**
  - Develop firmware before silicon arrives
  - Much faster than using ICD, ICE
- **Repeatable testing**
  - With pre-set stimulus
- **Load testing**
- **Free!**

Block Diagram of Simulator

- Cycle-based CPU simulation
- Registers
  - SFR (WREG, RXREG, T1CON)
  - GPR (2F0h)
- I/O Pins
  - RA0 (PORTA)
  - U1RX, T0CKI

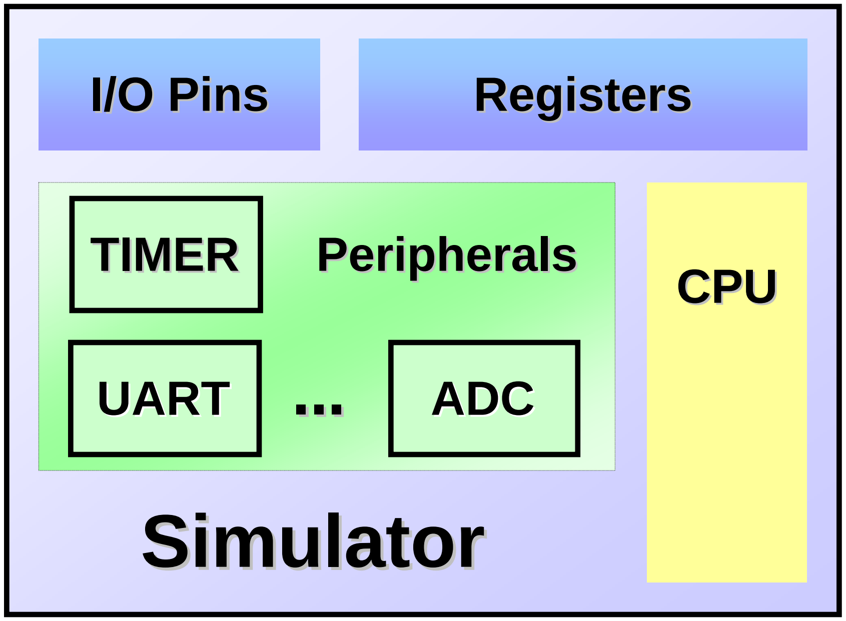


Alias Names of I/O Pins



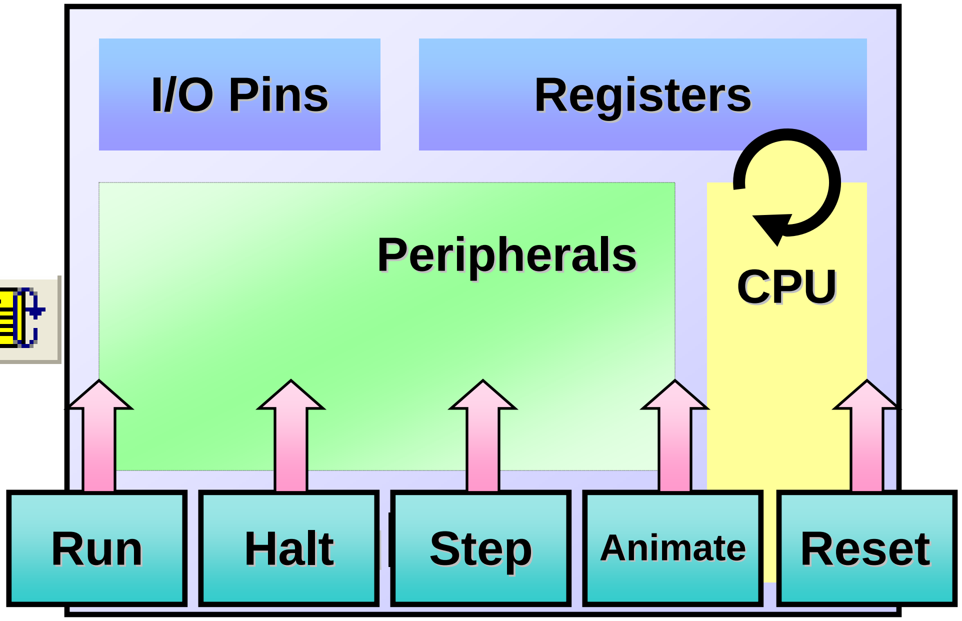
Block Diagram of Simulator

- Simulate behavior of peripherals
  - TIMER
  - UART, CCP, PWM, ADC
  - Input Capture
  - Output Compare
  - Comparator



Functions of Simulator

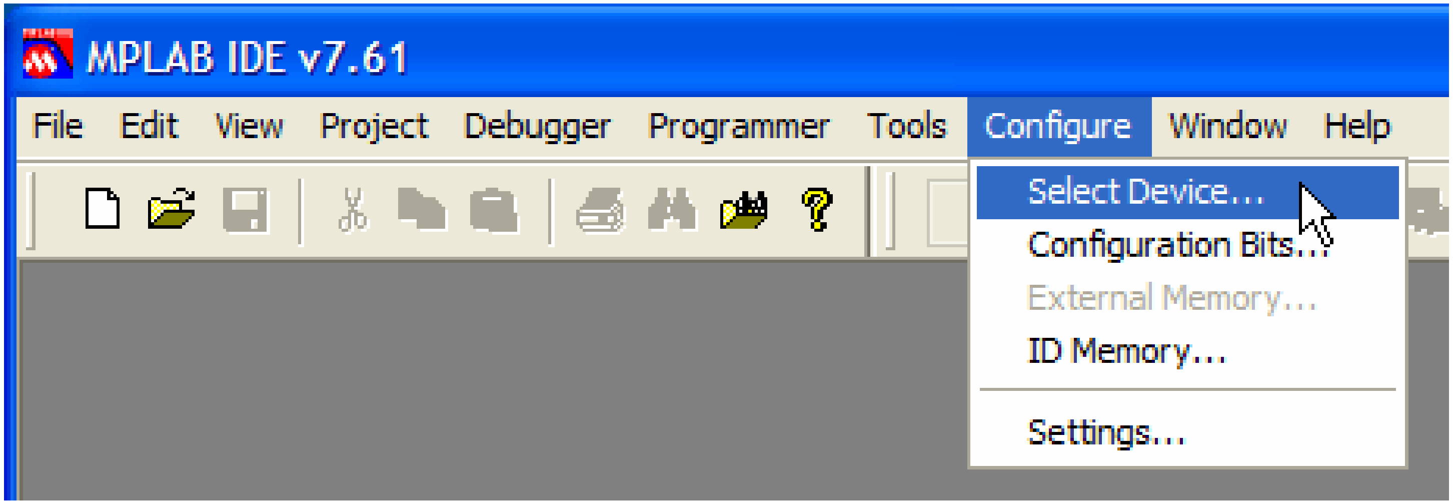
- Run: until breakpoint is reached ▶
- Halt: stop! ⏸
- Step Into {} Out {}
- Step Over {}
- Animate: slow ▶▶
- Reset: start over



Using Simulator

1. **Start MPLAB® IDE**
   Start>Programs>Microchip>MPLAB IDE >MPLAB IDE

2. **Select the right device**
   Configure>Select Device...



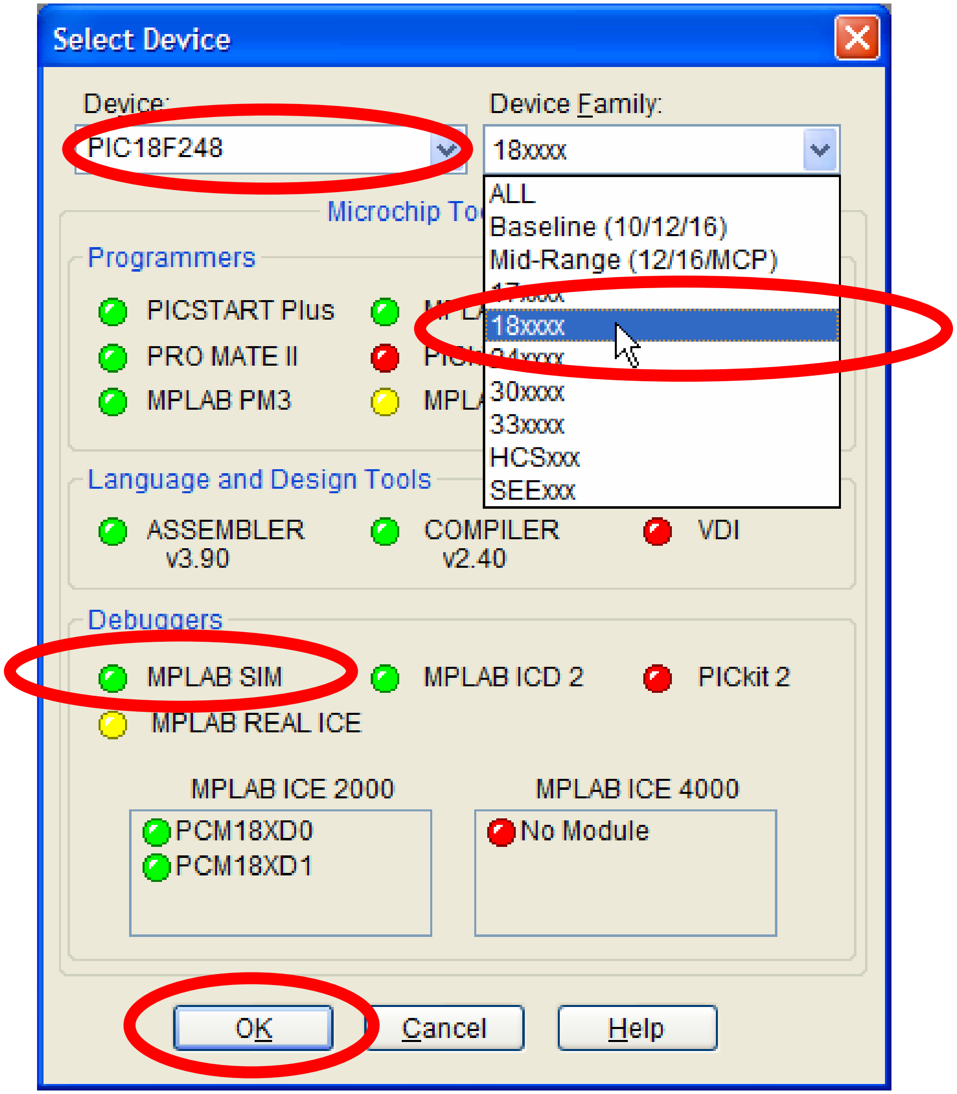
Select the Right Device

3. **Choose the Device Family**
4. **Then choose Device**
5. **Tool Support**

Green: *supported*

Yellow: *beta*

Red: *not ready*



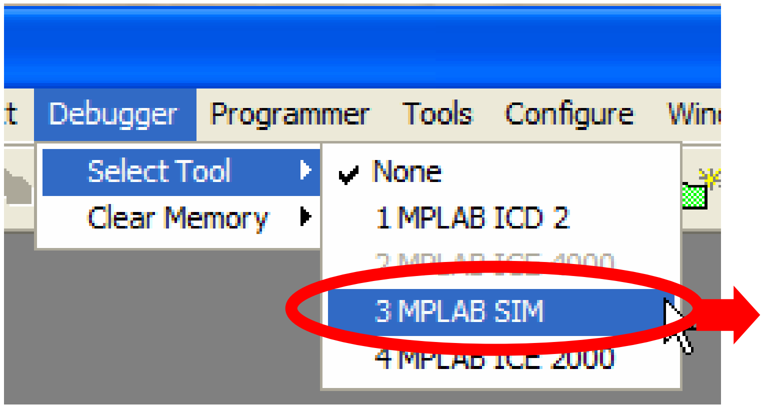
Select Device

Device: PIC18F248

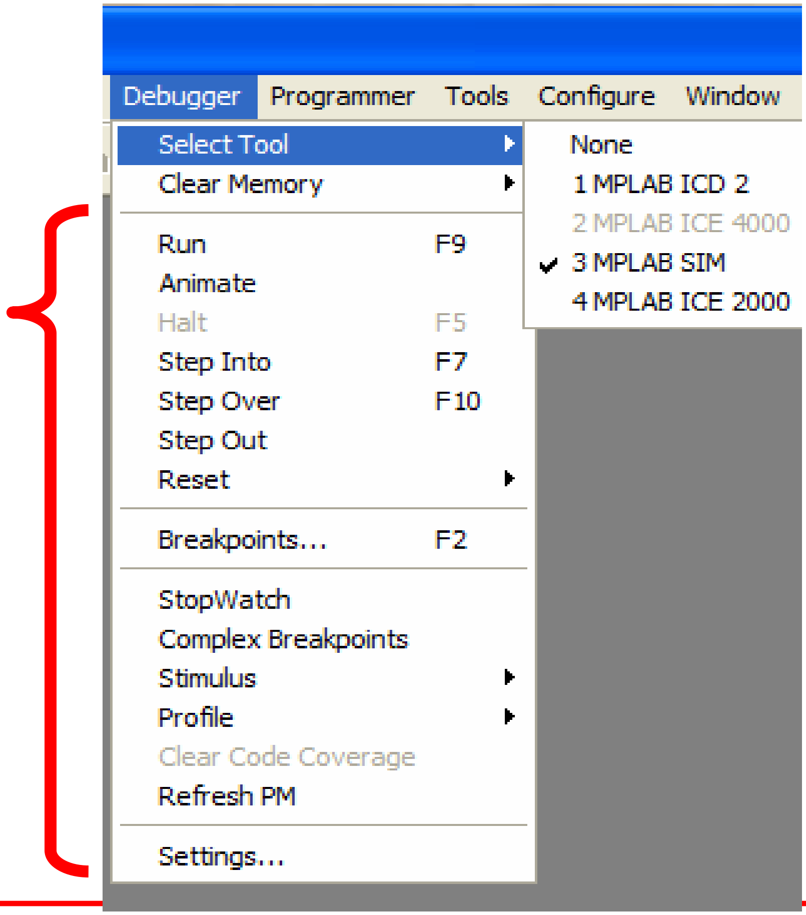
Device Family: 18xxxx

Simulator

6. **Select the Simulator**
   Debugger>Select Tool>MPLAB SIM

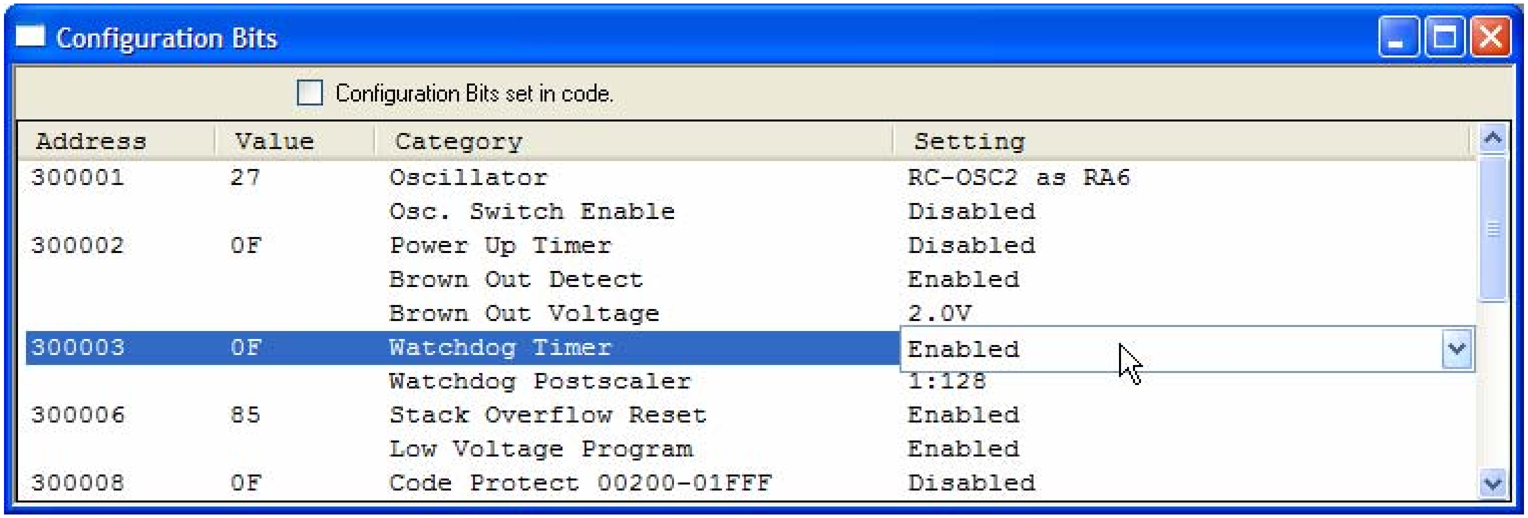


7. **Simulator functions available**



Simulator

8. **(Optional) Configure device**
   Configure>Configuration Bits...



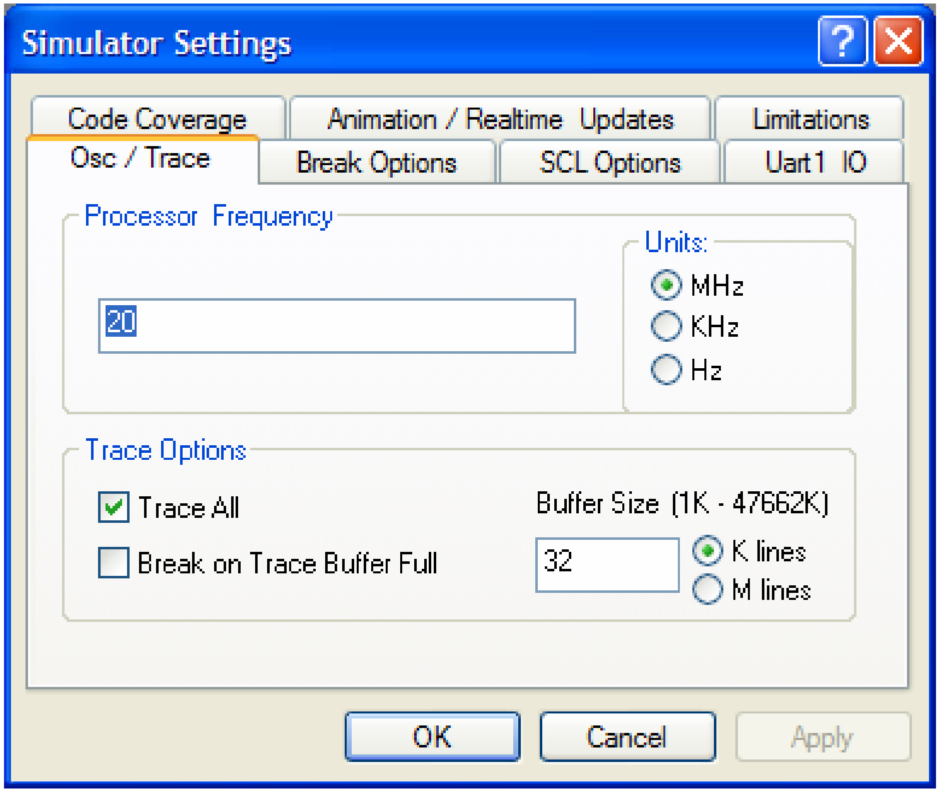
Address	Value	Category	Setting
300001	27	Oscillator	RC-OSC2 as RA6
		Osc. Switch Enable	Disabled
300002	0F	Power Up Timer	Disabled
		Brown Out Detect	Enabled
		Brown Out Voltage	2.0V
300003	0F	Watchdog Timer	Enabled
		Watchdog Postscaler	1:128
300006	85	Stack Overflow Reset	Enabled
		Low Voltage Program	Enabled
300008	0F	Code Protect 00200-01FFF	Disabled

- **Changes to configuration effective after next processor reset**
- **Alternatively, set these in code using __config directive**

Simulator

9. (Optional) Check Simulator Settings *Debugger>Settings...*

- Trace Options
- UART I/O
- Animation Speed
- Real time updates
- Break options
- Code Coverage



Processor Frequency

20

Units:
● MHz
○ KHz
○ Hz

Trace Options

☑ Trace All

□ Break on Trace Buffer Full

Buffer Size (1K - 47662K)

32
● K lines
○ M lines

Demo: Using MPLAB® Simulator

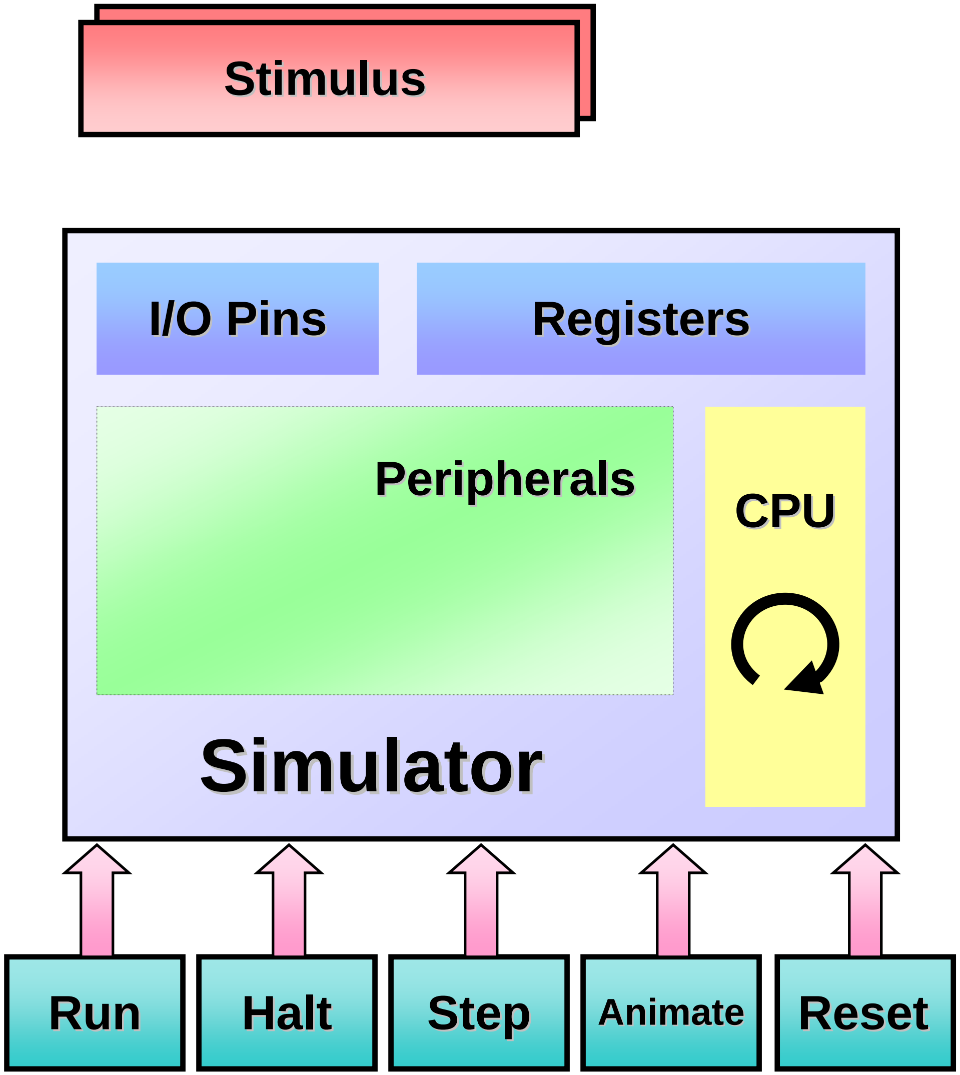
Stimulus

Stimulus

- A signal to stimulate a device through I/O ports / variables
- Signal can be:
  - UART data (communications)
  - ADC data (temperature sensor)
  - Digital I/O (switches)

Two Kinds of Stimulus

- Asynchronous
  - Any time
- Synchronous
  - Preset time



Asynchronous Stimulus

- User-initiated (click a button)
- Any time during simulation
- Apply to:
  - I/O pins
  - UART buffer registers (RCREG)
- Actions:
  - Set low / high
  - Pulse low / Pulse high
  - Toggle

Synchronous Stimulus

- **Pre-programmed when to happen**
- **Applicable to**
  - I/O Pins
  - All registers
- **Tied to events, e.g.**
  - When it is @ cycle 20 or @ 120 ms
  - When PC reaches address 0xC40
  - When ADCBUF is read in program
  - When RA4 is set

UNIVERSITY OF MICROCHIP UOFM MASTERs 2007

Destination of Stimulus

- I/O Pins

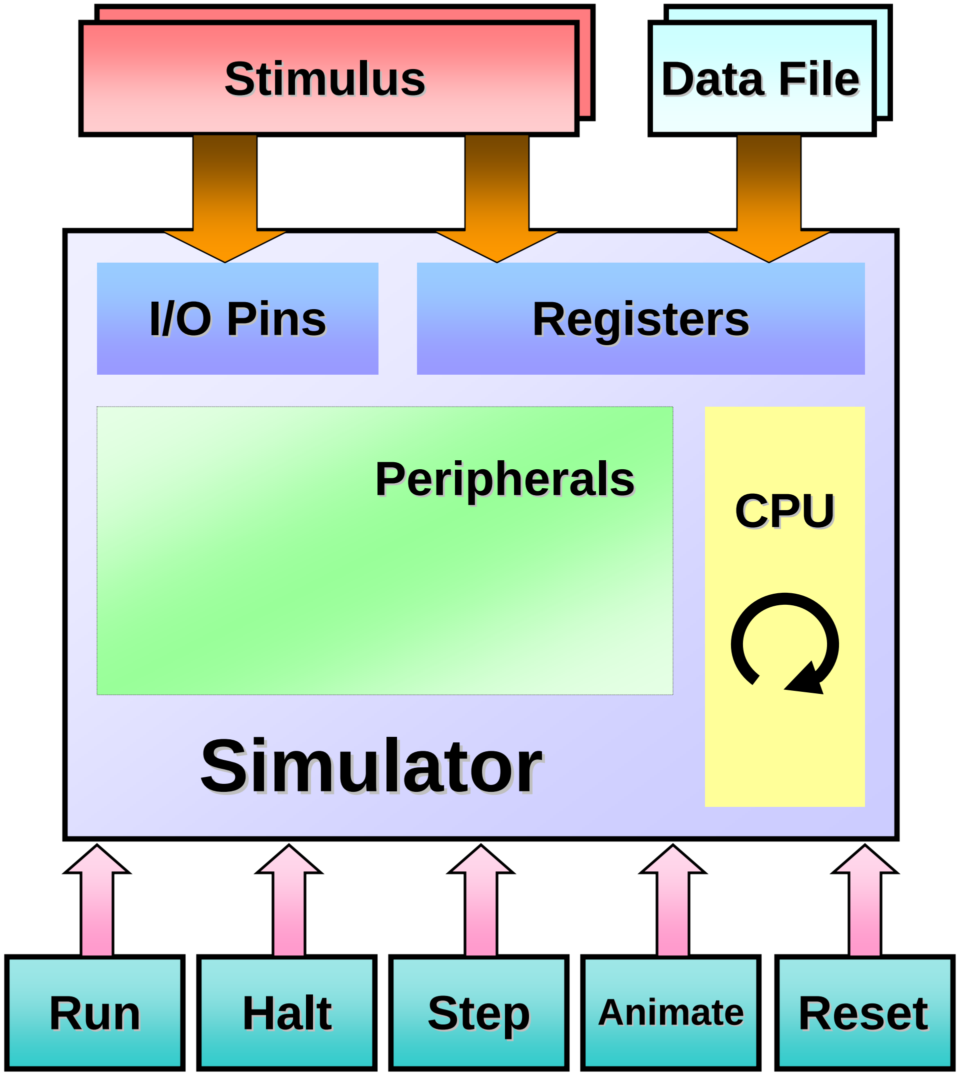
  Toggle pin

  T0CKI
- Registers

  UARTRX ← 123

  foo ← 0x5B
- Data

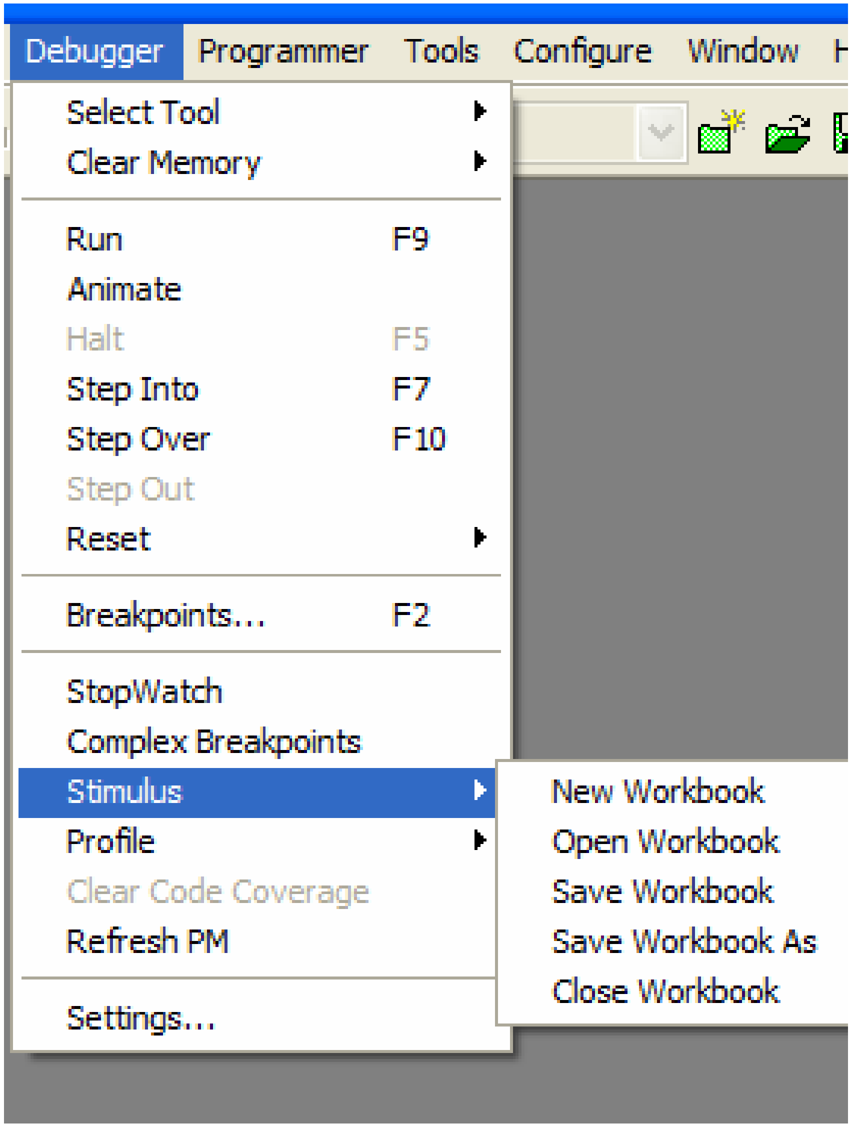
  Can be injected from data file



Using Stimulus GUI

Using Stimulus

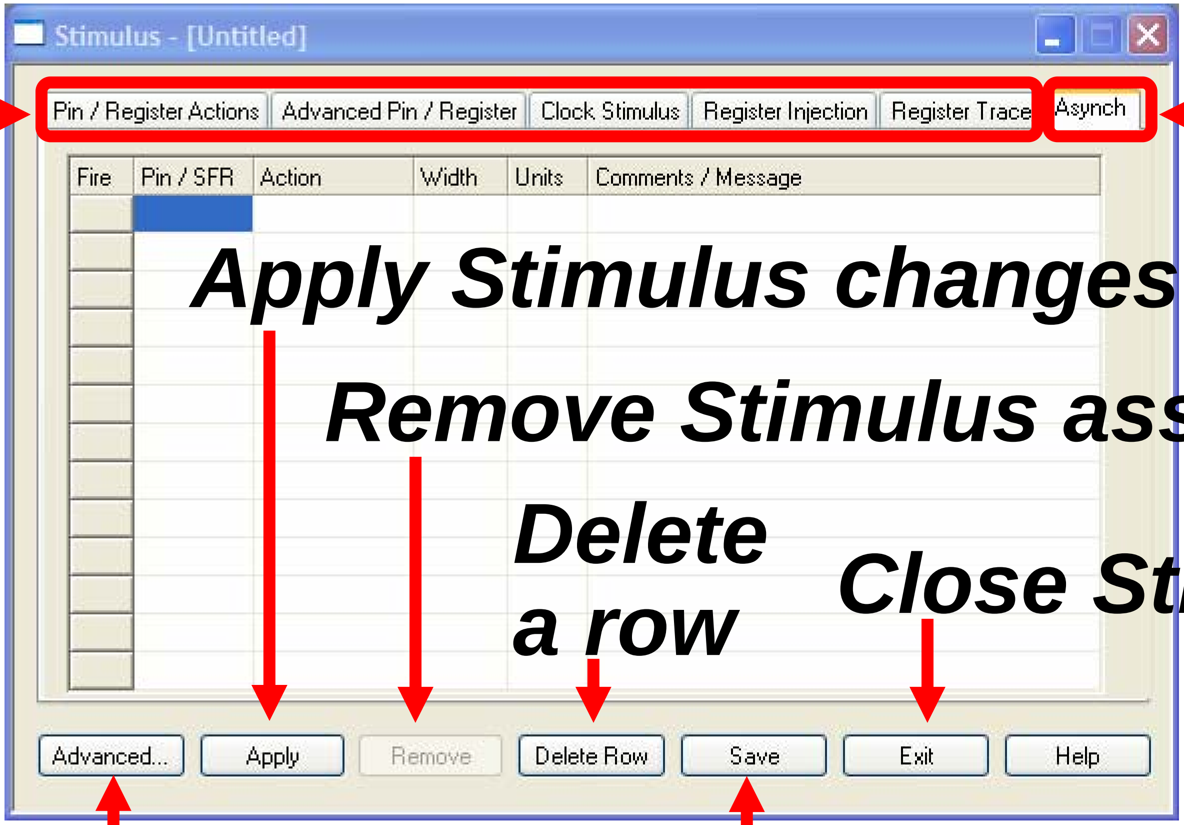
- **Stimuli are defined in workbook:**
  - New Workbook
  - Open Workbook
  - Save Workbook
  - Save Workbook As
  - Close Workbook



6 Tabs of Stimulus Window

Synchronous

Asynchronous



Apply Stimulus changes

Remove Stimulus association

Delete a row

Close Stimulus

Advanced SCL operations

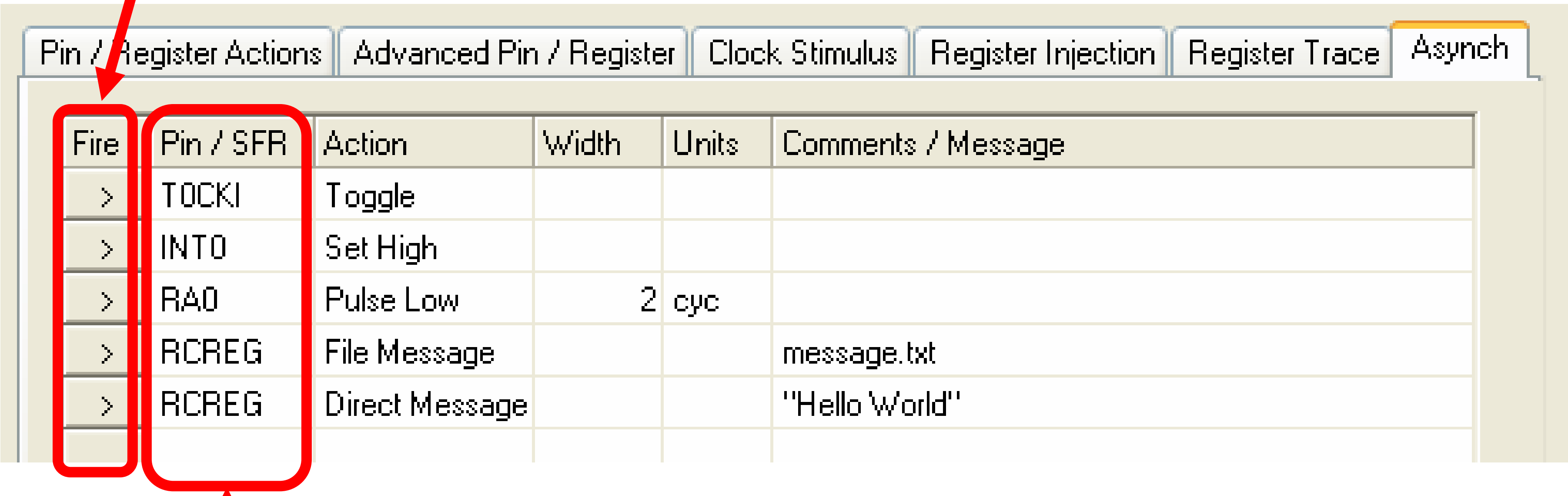
Save all stimulus definitions to workbook

Steps to Define/Use Stimulus

1. **Bring up stimulus window**
   Debugger>Stimulus>New Workbook
2. **Define stimulus**
3. **Save stimulus (optional)**
   Debugger>Stimulus>Save Workbook
4. **Apply stimulus**
5. **Stimulus window remains open**
6. **Start simulation**

Asynchronous Stimulus

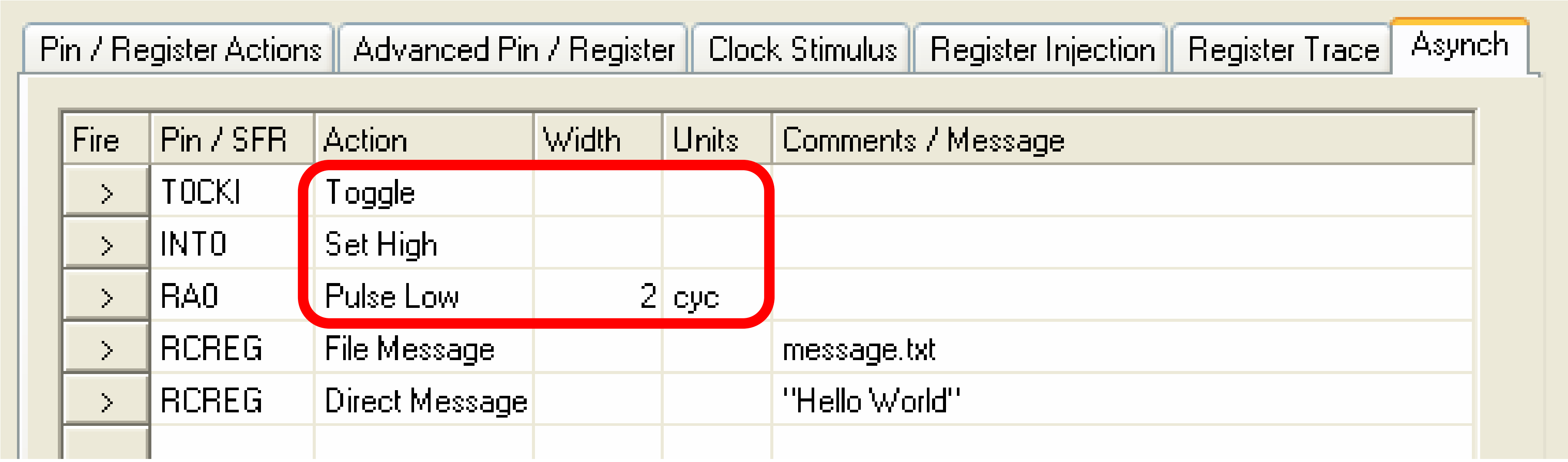
- Last tab in Stimulus Window
- Fire button for each row



Fire	Pin / SFR	Action	Width	Units	Comments / Message
>	T0CKI	Toggle			
>	INT0	Set High			
>	RA0	Pulse Low	2	cyc	
>	RCREG	File Message			message.txt
>	RCREG	Direct Message			"Hello World"

- Destination: I/O Pin or RCREG

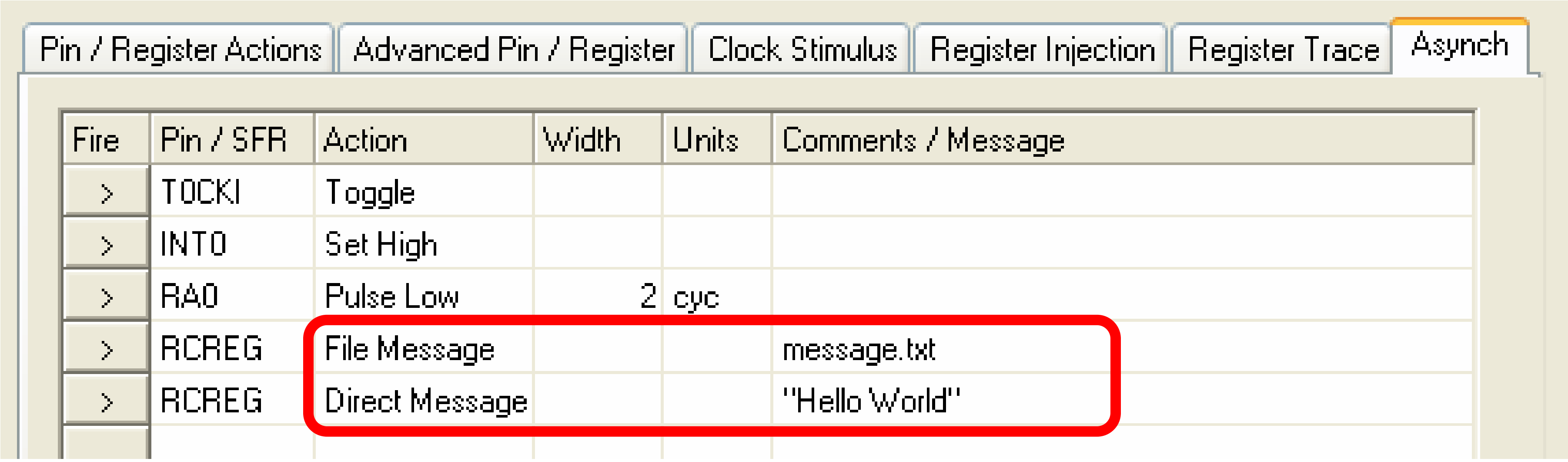
Asynchronous Stimulus



Fire	Pin / SFR	Action	Width	Units	Comments / Message
>	T0CKI	Toggle			
>	INT0	Set High			
>	RA0	Pulse Low	2	cyc	
>	RCREG	File Message			message.txt
>	RCREG	Direct Message			"Hello World"

- **Action for I/O Pin:**
  - Set High/Low
  - Toggle
  - Pulse High/Low for *2 cycles*

Asynchronous Stimulus



Fire	Pin / SFR	Action	Width	Units	Comments / Message
>	T0CKI	Toggle			
>	INT0	Set High			
>	RA0	Pulse Low	2	cyc	
>	RCREG	File Message			message.txt
>	RCREG	Direct Message			"Hello World"

- **Action for RCREG:**
  - **Message-based Stimulus**
    - File Message: **Inject from file**
    - Direct Message: **defined in GUI**
  - **More information in appendix**

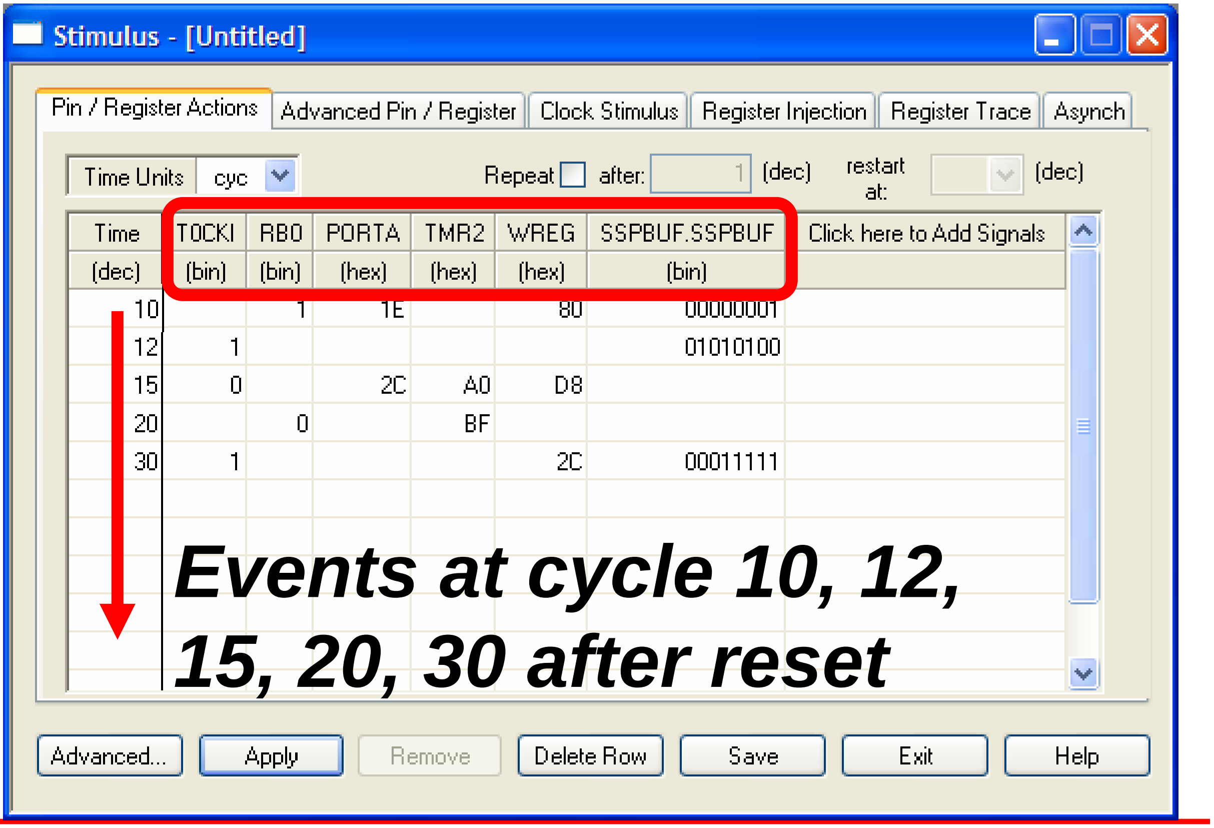
Asynchronous Summary

- **Quick stimulus to test application**
- **Possible uses:**
  - True random event (user key press)
  - Asynchronous UART message

- **Onto Synchronous Stimulus...**

Pin/Register Actions

- **Define Sequence of stimulus @ pre-set time**
- Pin
  T0CKI, RB0
- Port
  PORTA
- SFR
  TMR2, WREG
- Bit field
  SSPBUF



Time Units: cyc

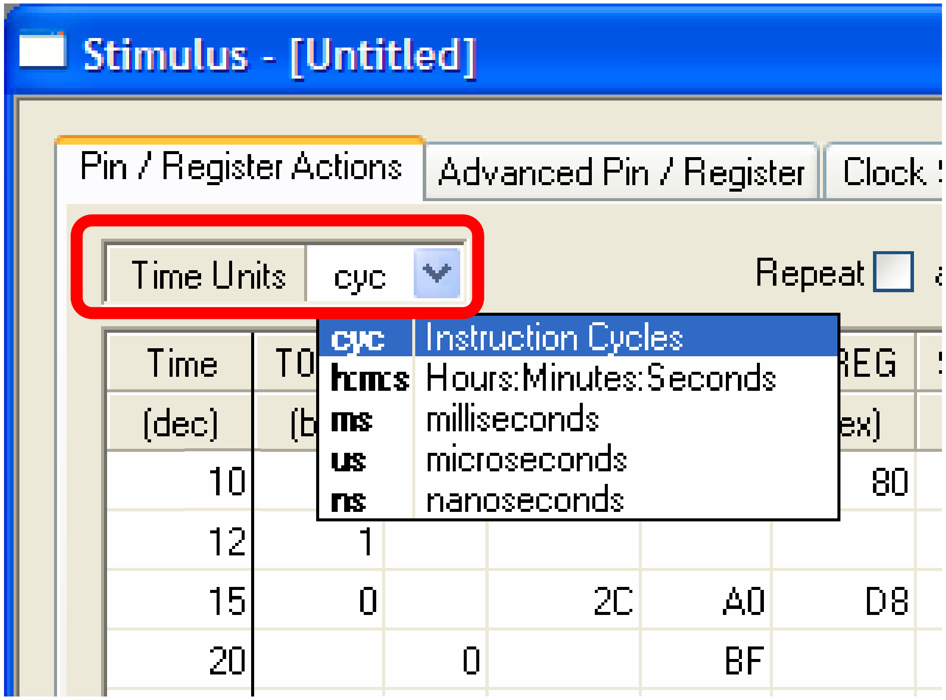
Repeat □ after: 1 (dec) restart at: (dec)

Time	T0CKI	RB0	PORTA	TMR2	WREG	SSPBUF.SSPBUF
(dec)	(bin)	(bin)	(hex)	(hex)	(hex)	(bin)
10		1	1E		80	00000001
12	1					01010100
15	0		2C	A0	D8	
20		0		BF		
30	1				2C	00011111

Events at cycle 10, 12, 15, 20, 30 after reset

Pin/Register Actions

- Time Unit
  - common to all events on this tab
  - Cycle
  - Hour:Min:Sec
  - ms
  - us
  - ns



Time Units cyc

Repeat □

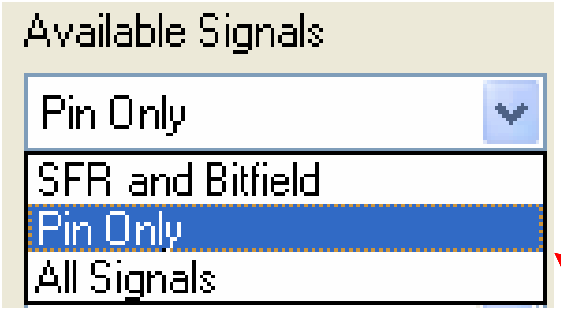
cyc	Instruction Cycles
h:m:s	Hours:Minutes:Seconds
ms	milliseconds
us	microseconds
ns	nanoseconds

Time (dec)	T0 (b				REG ex)
10					80
12	1				
15	0		2C	A0	D8
20		0		BF	

Pin/Register Actions

- **Add/Remove Signals**

Signal Filter:



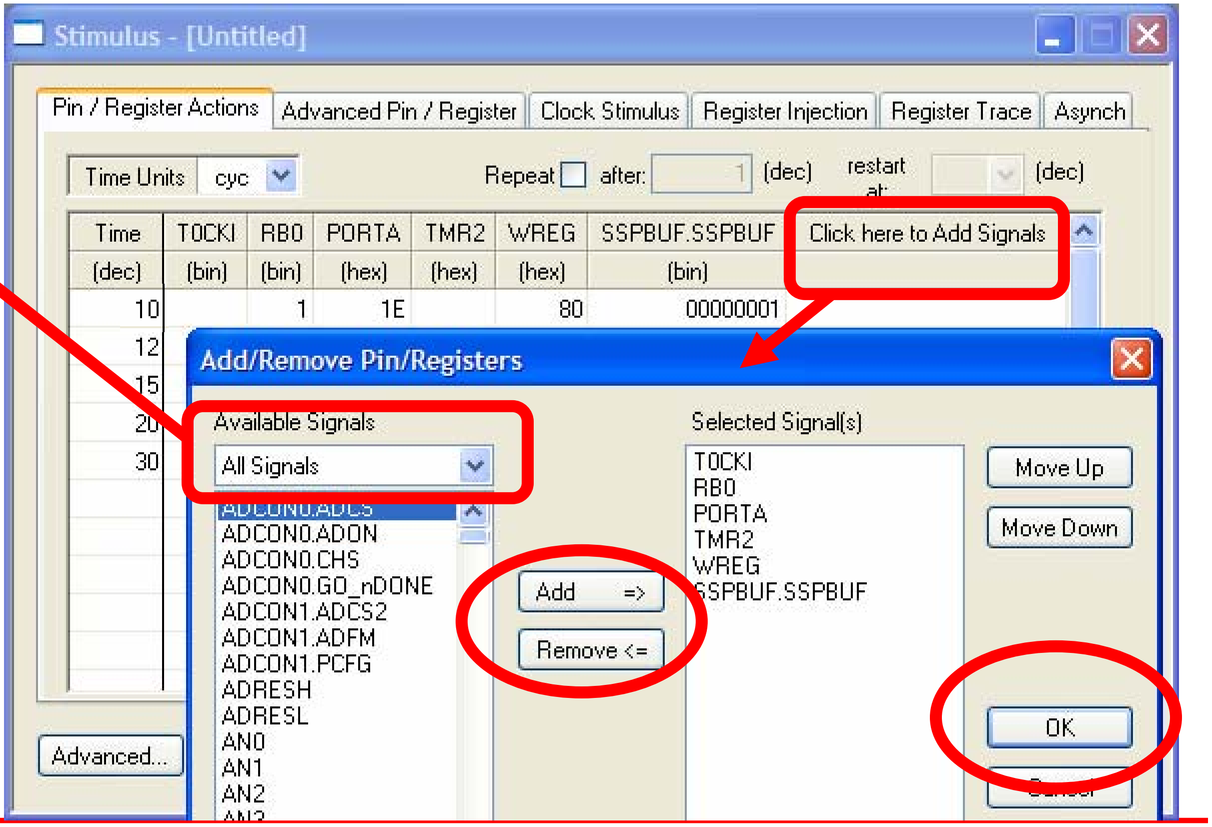
Available Signals

Pin Only

- SFR and Bitfield
- Pin Only
- All Signals

Add/Remove

Click OK



Stimulus - [Untitled]

Time Units: cyc

Repeat □ after: 1 (dec) restart at: (dec)

Time	T0CKI	RB0	PORTA	TMR2	WREG	SSPBUF.SSPBUF	Click here to Add Signals
(dec)	(bin)	(bin)	(hex)	(hex)	(hex)	(bin)	
10		1	1E		80	00000001	
12							
15							
20							
30							

Add/Remove Pin/Registers

Available Signals: All Signals

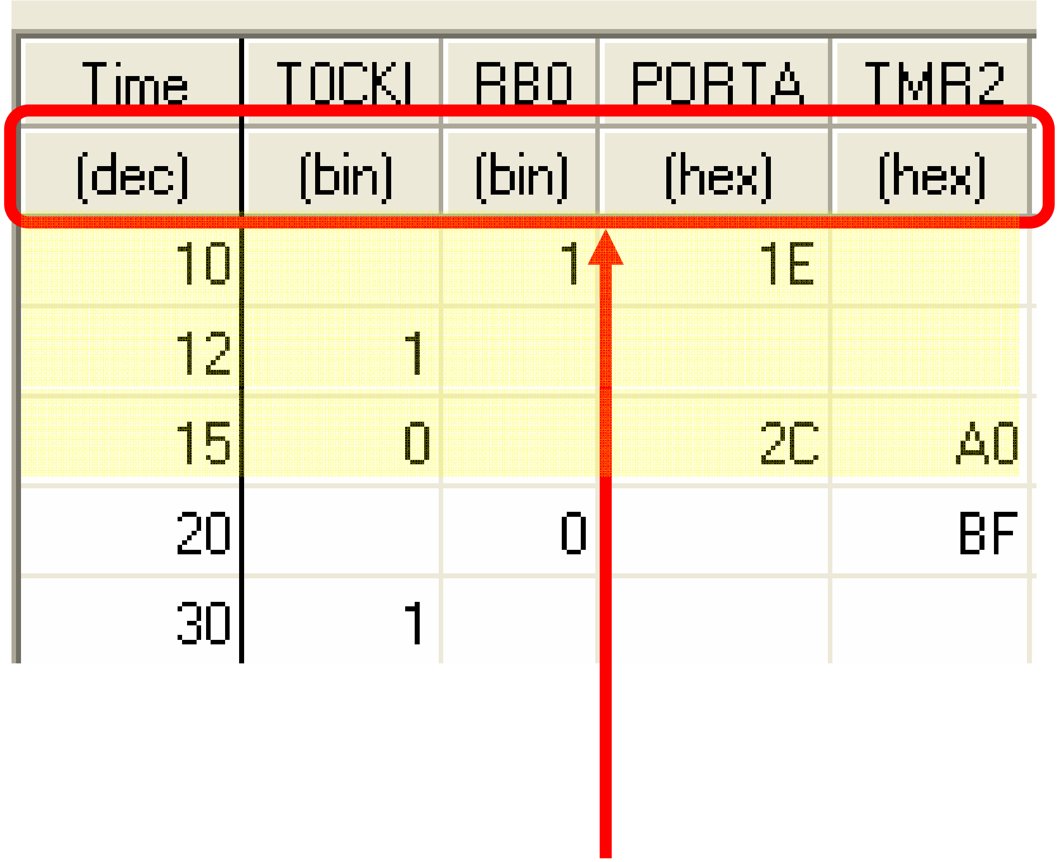
ADCON0.ADCS
ADCON0.ADON
ADCON0.CHS
ADCON0.GO_nDONE
ADCON1.ADCS2
ADCON1.ADFM
ADCON1.PCFG
ADRESH
ADRESL
AN0
AN1
AN2
AN3

Selected Signal(s):

T0CKI
RB0
PORTA
TMR2
WREG
SSPBUF.SSPBUF

What Happens at Run Time?

- At cycle 10:
  - RB0 ← high
  - PORTA ← 1E
- At cycle 12:
  - T0CKI ← high
- At cycle 15:
  - T0CKI ← low
  - PORTA ← 2C
  - TMR2 ← A0



Time	T0CKI	RB0	PORTA	TMR2
(dec)	(bin)	(bin)	(hex)	(hex)
10		1	1E	
12	1			
15	0		2C	A0
20		0		BF
30	1			

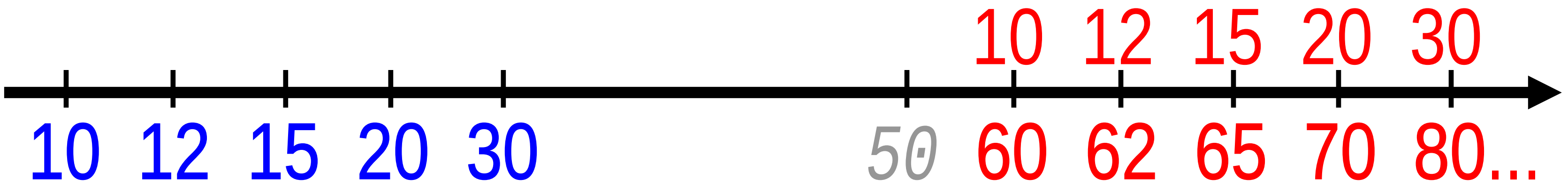
Radix for data entry

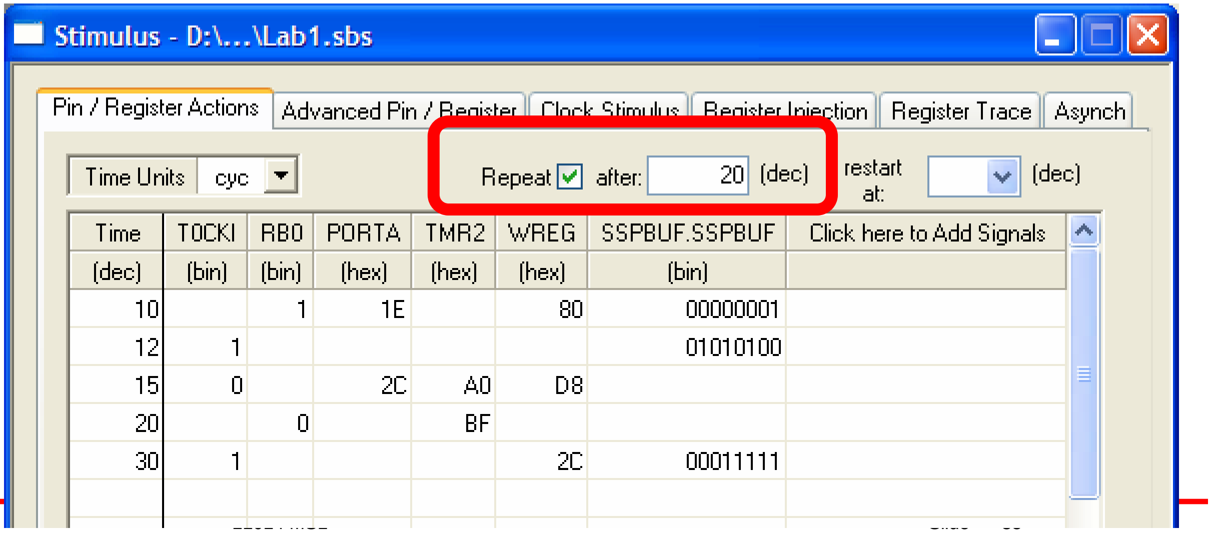
Might need to repeat stimuli

- Check “Repeat” and enter delay

10 12 15 20 30

[wait 20 cycles]





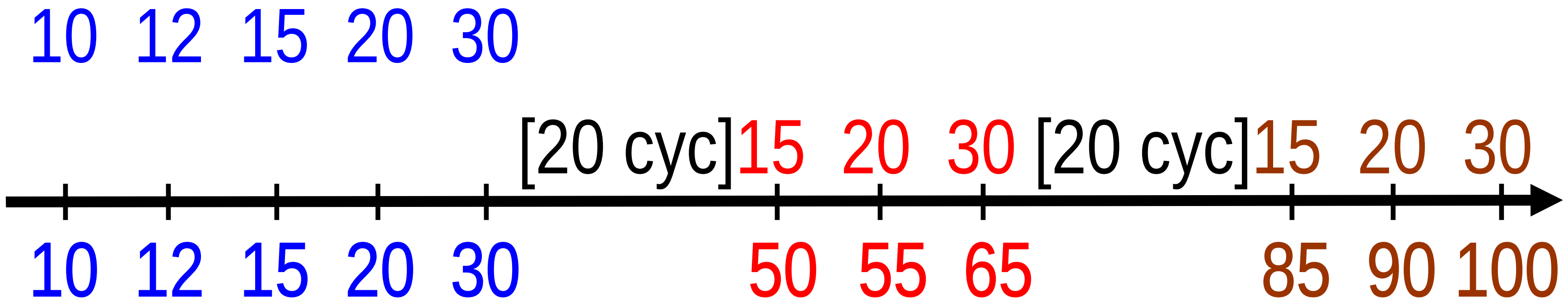
Time Units: cyc

Repeat ☑ after: 20 (dec) restart at: (dec)

Time (dec)	T0CKI (bin)	RB0 (bin)	PORTA (hex)	TMR2 (hex)	WREG (hex)	SSPBUF.SSPBUF (bin)
10		1	1E		80	00000001
12	1					01010100
15	0		2C	A0	D8	
20		0		BF		
30	1				2C	00011111

“Restart at” option

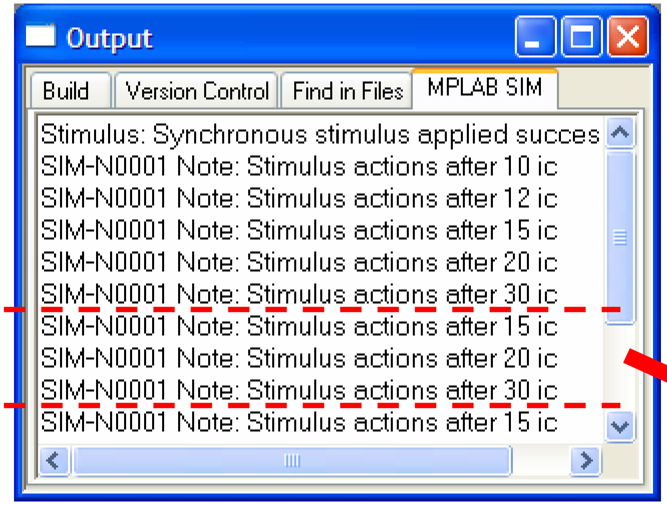
- **Selects which row to begin repeating after delay**



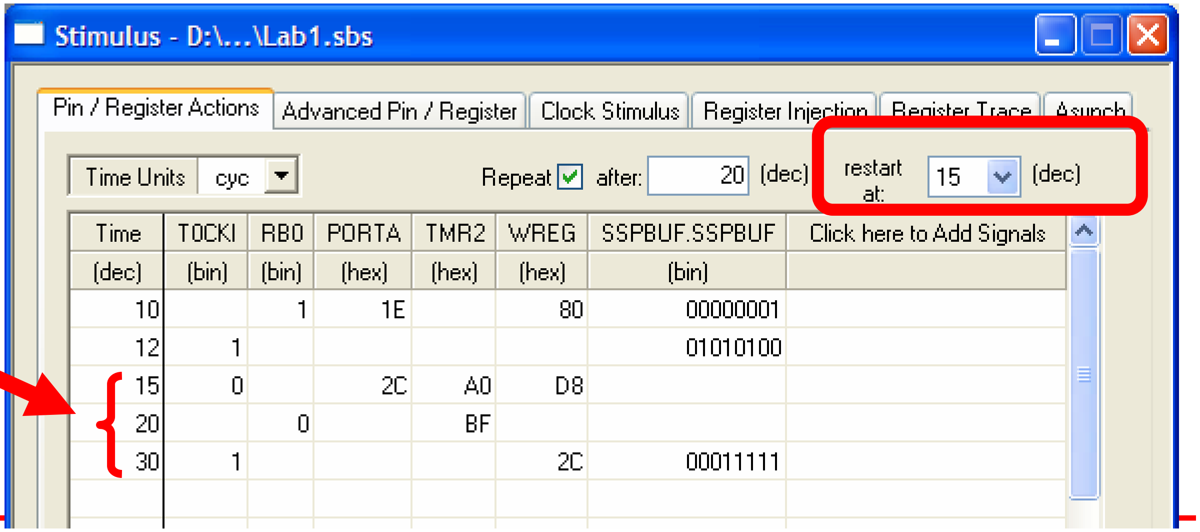
10 12 15 20 30

[20 cyc]15 20 30 [20 cyc]15 20 30

10 12 15 20 30 50 55 65 85 90 100



```
Stimulus: Synchronous stimulus applied succes
SIM-N0001 Note: Stimulus actions after 10 ic
SIM-N0001 Note: Stimulus actions after 12 ic
SIM-N0001 Note: Stimulus actions after 15 ic
SIM-N0001 Note: Stimulus actions after 20 ic
SIM-N0001 Note: Stimulus actions after 30 ic
SIM-N0001 Note: Stimulus actions after 15 ic
SIM-N0001 Note: Stimulus actions after 20 ic
SIM-N0001 Note: Stimulus actions after 30 ic
SIM-N0001 Note: Stimulus actions after 15 ic
```



Time Units cyc

Repeat ☑ after: 20 (dec) restart at: 15 (dec)

Time	TOCKI	RB0	PORTA	TMR2	WREG	SSPBUF.SSPBUF	Click here to Add Signals
(dec)	(bin)	(bin)	(hex)	(hex)	(hex)	(bin)	
10		1	1E		80	00000001	
12	1					01010100	
15	0		2C	A0	D8		
20		0		BF			
30	1				2C	00011111	

Pin/Register Summary

- Set / Clear pin, SFR, SFR bit field
- At specific time during simulation
- All data contained in the workbook
- Possible uses:
  - Setting two IRQ flags simultaneously
  - Simulating A/D buffer (ADRES)
  - Setting pulse to ext clock input T0CKI
  - Duration test (using “repeat”)

Lab 1: Simple Stimulus

C:>Masters>11014>Lab1>Lab1.mcw

Other Stimulus Tabs

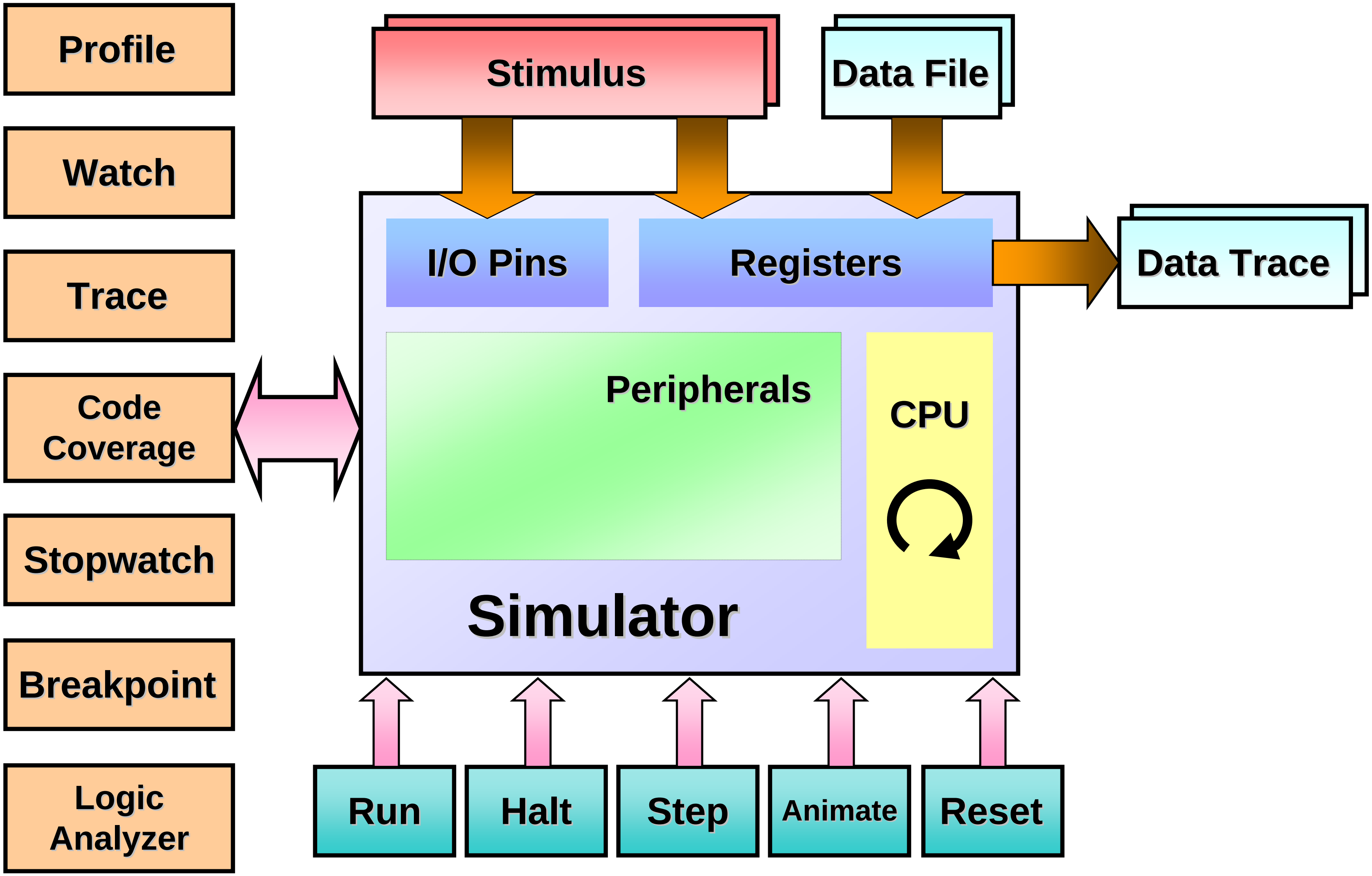
- **Register Injection**
  - Attach input data file to Register or GPR
- **Register Trace**
  - Attach output file to Register or GPR
- **Clock Stimulus**
  - Generates waveform to I/O pin

Other Stimulus Tabs (Cont’d)

- **Advanced Pin/Register**
  - Conditional Pin/Register stimulus
- **Other MASTERS classes**
  - Advanced Stimulus (11015 MS2)
  - Debugging Techniques (11016 MS3) *NEW!*

Tools in Simulator

Tools in Simulator



Tools in Simulator

- Profile
- Watch
- Trace
- Code Coverage
- Stopwatch
- Breakpoint
- Logic Analyzer

- **Profile**
  - Count # of times used by each instruction
- **Real-time Watch Window**
  - See contents of memory / variables during simulation
  - Drag and drop variables from editor

Execution Trace

Profile

Watch

Trace

Code Coverage

Stopwatch

Breakpoint

Logic Analyzer

- Record execution history of program
- Instruction tagged with time stamp (execution cycle)
- Disassembled Opcodes
- Source and Destination Data values
- Source and Destination Address

Execution Trace (cont'd)

Profile

Watch

Trace

Code Coverage

Stopwatch

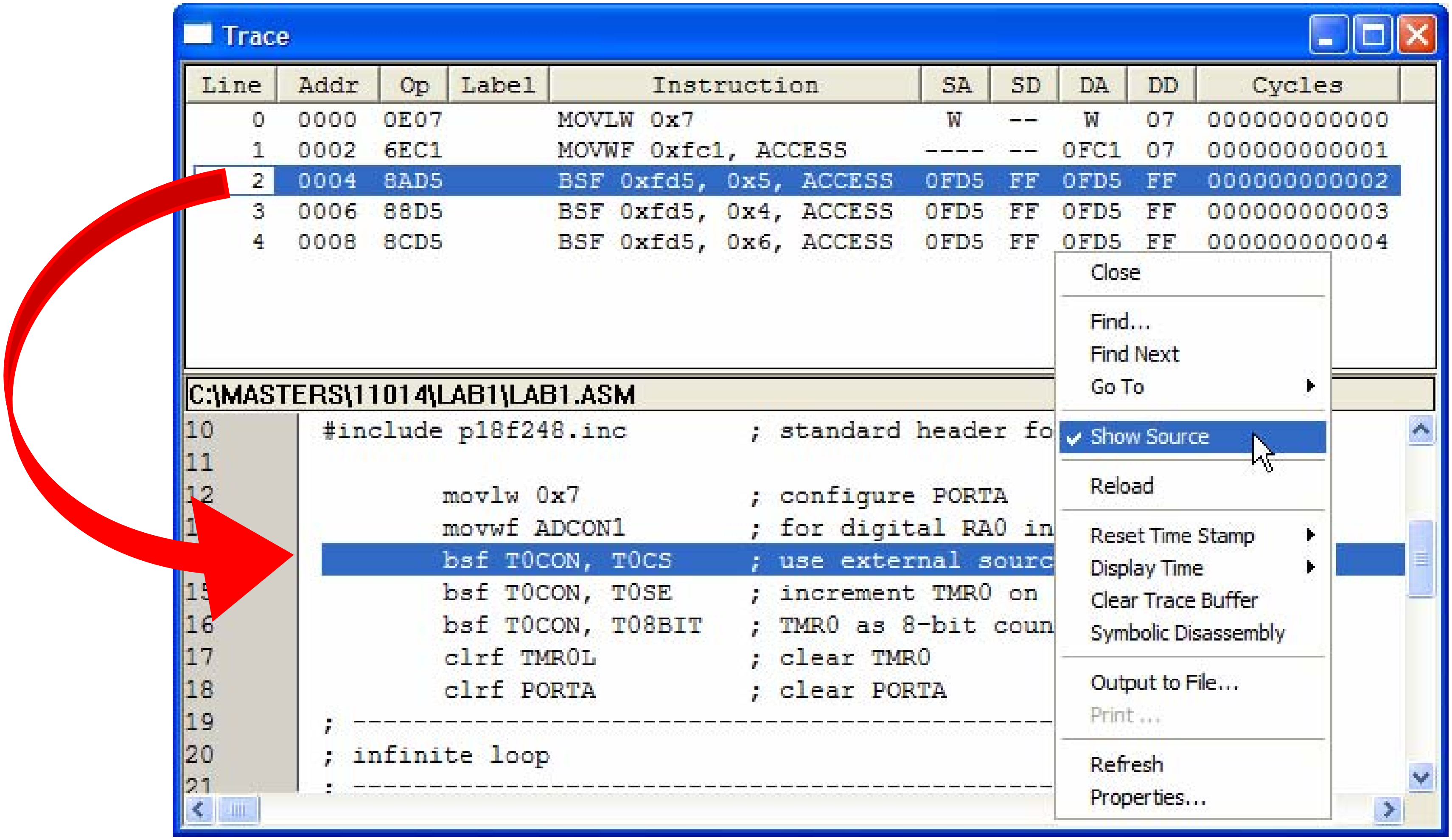
Breakpoint

Logic Analyzer

- Filter In/Out from source
- Trace to Source Correlation
- Select Menu *View>Simulator Trace*
- **Trace All** must be selected in debugger settings
- Trace buffer up to 2GB (one trace record ~40 bytes)

Trace to Source Correlation

- Right click → select *Show Source*



```
Line  Addr  Op    Label  Instruction                SA    SD  DA    DD  Cycles
   0  0000  0E07         MOVLW 0x7                  W     --  W     07  000000000000
   1  0002  6EC1         MOVWF 0xfc1, ACCESS        ----  --  0FC1  07  000000000001
   2  0004  8AD5         BSF 0xfd5, 0x5, ACCESS     0FD5  FF  0FD5  FF  000000000002
   3  0006  88D5         BSF 0xfd5, 0x4, ACCESS     0FD5  FF  0FD5  FF  000000000003
   4  0008  8CD5         BSF 0xfd5, 0x6, ACCESS     0FD5  FF  0FD5  FF  000000000004
```

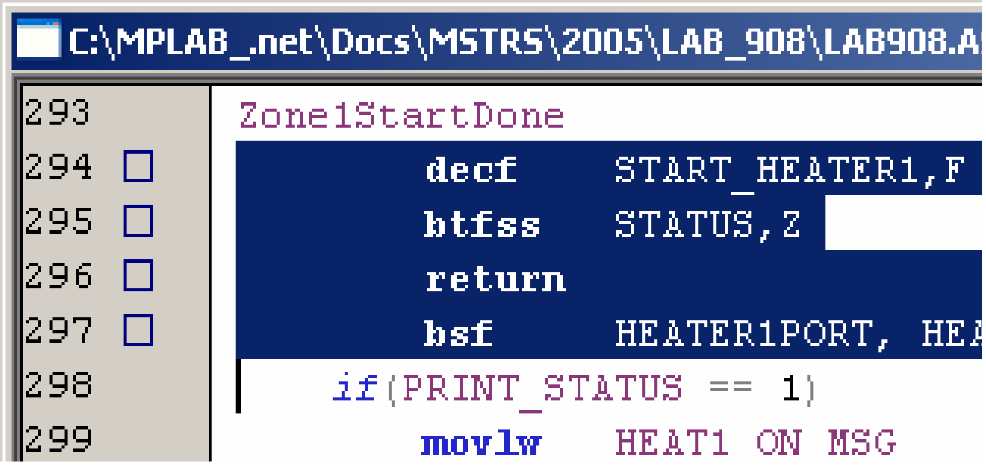
C:\MASTERS\11014\LAB1\LAB1.ASM

```
#include p18f248.inc        ; standard header fo

        movlw 0x7           ; configure PORTA
        movwf ADCON1        ; for digital RA0 in
        bsf T0CON, T0CS     ; use external sourc
        bsf T0CON, T0SE     ; increment TMR0 on
        bsf T0CON, T08BIT   ; TMR0 as 8-bit coun
        clrf TMR0L          ; clear TMR0
        clrf PORTA          ; clear PORTA
; ----------------------------------------------
; infinite loop
; ----------------------------------------------
```

Filter Trace – selective

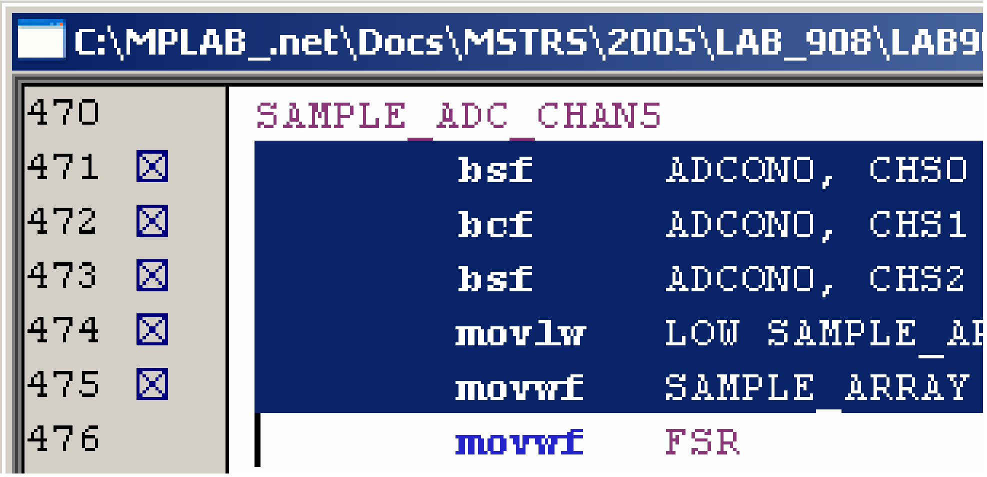
- **Trace certain lines, e.g. loop** right click→ *Add Filter-in Trace*



C:\MPLAB_.net\Docs\MSTRS\2005\LAB_908\LAB908.A

```
  Zone1StartDone
☐         decf      START_HEATER1,F
☐         btfss     STATUS,Z
☐         return
☐         bsf       HEATER1PORT, HEA
      if(PRINT_STATUS == 1)
          movlw     HEAT1 ON MSG
```

- **Not trace some e.g. eliminate wait loops** right click→ *Add Filter-out Trace*



C:\MPLAB_.net\Docs\MSTRS\2005\LAB_908\LAB9

```
  SAMPLE_ADC_CHANS
☒         bsf       ADCON0, CHS0
☒         bcf       ADCON0, CHS1
☒         bsf       ADCON0, CHS2
☒         movlw     LOW SAMPLE_AR
☒         movwf     SAMPLE_ARRAY
          movwf     FSR
```

Code Coverage

Profile

Watch

Trace

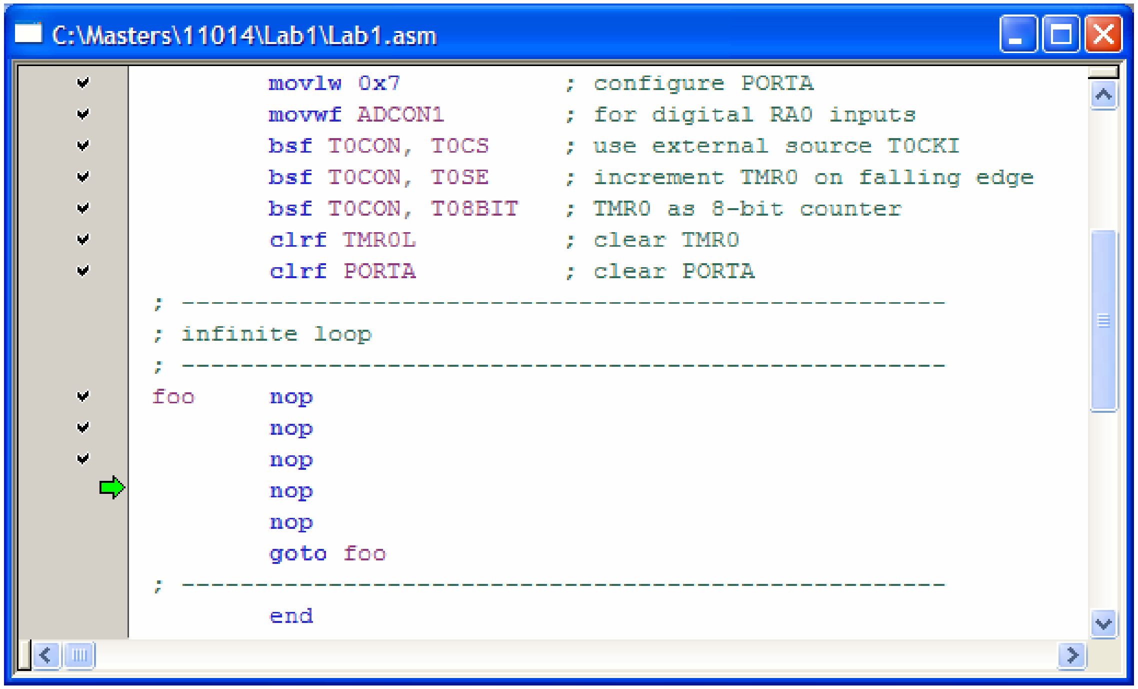
Code Coverage

Stopwatch

Breakpoint

Logic Analyzer

- Visually shows executed code in editor window (checked)
- Identifies unreachable code



C:\Masters\11014\Lab1\Lab1.asm

```
✓          movlw 0x7            ; configure PORTA
✓          movwf ADCON1         ; for digital RA0 inputs
✓          bsf T0CON, T0CS      ; use external source T0CKI
✓          bsf T0CON, T0SE      ; increment TMR0 on falling edge
✓          bsf T0CON, T08BIT    ; TMR0 as 8-bit counter
✓          clrf TMR0L           ; clear TMR0
✓          clrf PORTA           ; clear PORTA
   ; -----------------------------------------------------
   ; infinite loop
   ; -----------------------------------------------------
✓  foo     nop
✓          nop
✓          nop
           nop
           nop
           goto foo
   ; -----------------------------------------------------
           end
```

Stopwatch

Profile

Watch

Trace

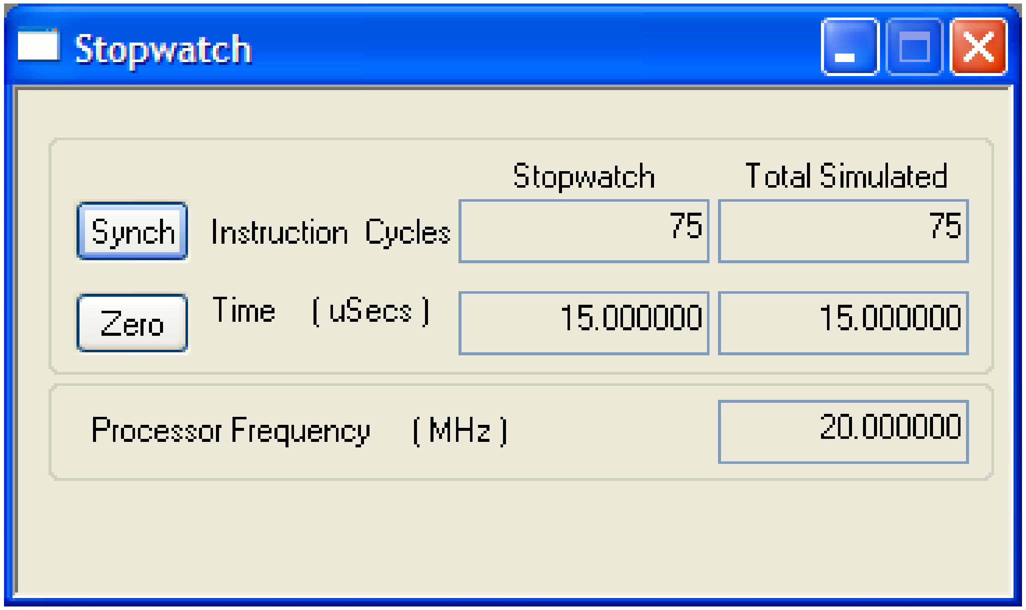
Code Coverage

Stopwatch

Breakpoint

Logic Analyzer

- Measures in cycles & time
- Reset to zero upon reset



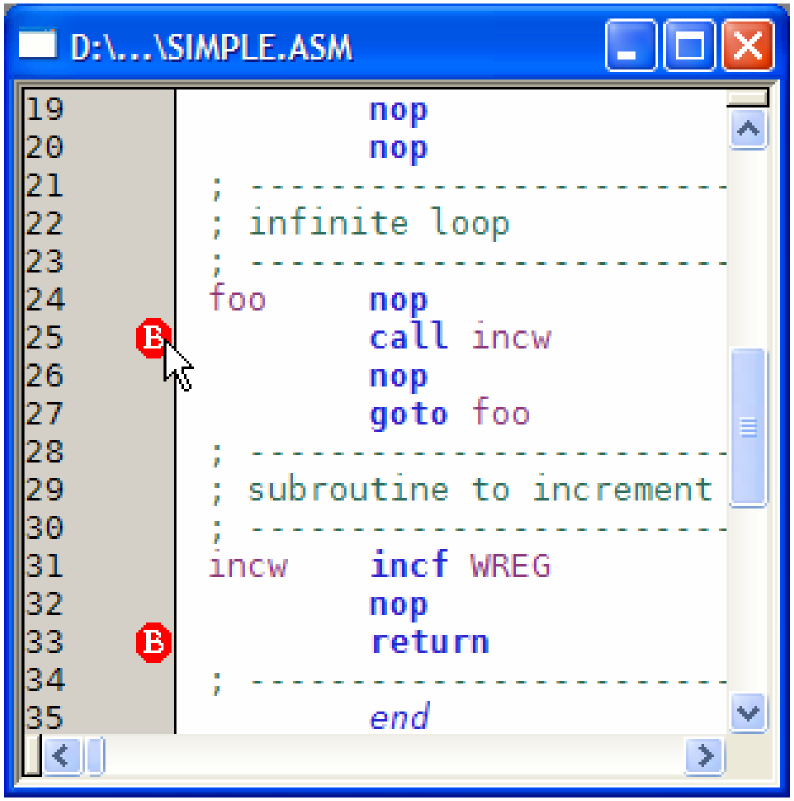
		Stopwatch	Total Simulated
Synch	Instruction Cycles	75	75
Zero	Time (uSecs)	15.000000	15.000000

Processor Frequency (MHz) 20.000000

Breakpoint

- Profile
- Watch
- Trace
- Code Coverage
- Stopwatch
- Breakpoint
- Logic Analyzer

- Set unlimited # of breakpoints
- Double click in editor margin to set



```
            nop
            nop
; -----------------------
; infinite loop
; -----------------------
foo         nop
            call incw
            nop
            goto foo
; -----------------------
; subroutine to increment
; -----------------------
incw        incf WREG
            nop
            return
; -----------------------
            end
```

Complex Breakpoints

Profile

Watch

Trace

Code Coverage

Stopwatch

Breakpoint

Logic Analyzer

- Provides great flexibility when to halt execution
- Break on Program Memory or Data Memory R/W
- Anded Breakpoints

  A & B must happen same time
- Sequenced Breakpoints

  A then B must happen in order

Logic Analyzer

- Profile
- Watch
- Trace
- Code Coverage
- Stopwatch
- Breakpoint
- Logic Analyzer

- Display I/O port data graphically
- Uses data in trace buffer
- Trace MUST be enabled (Trace All selected in settings)
- Max. 32 channels
- Bus (collection of I/O pins) e.g. PORTB = RB0...RB7

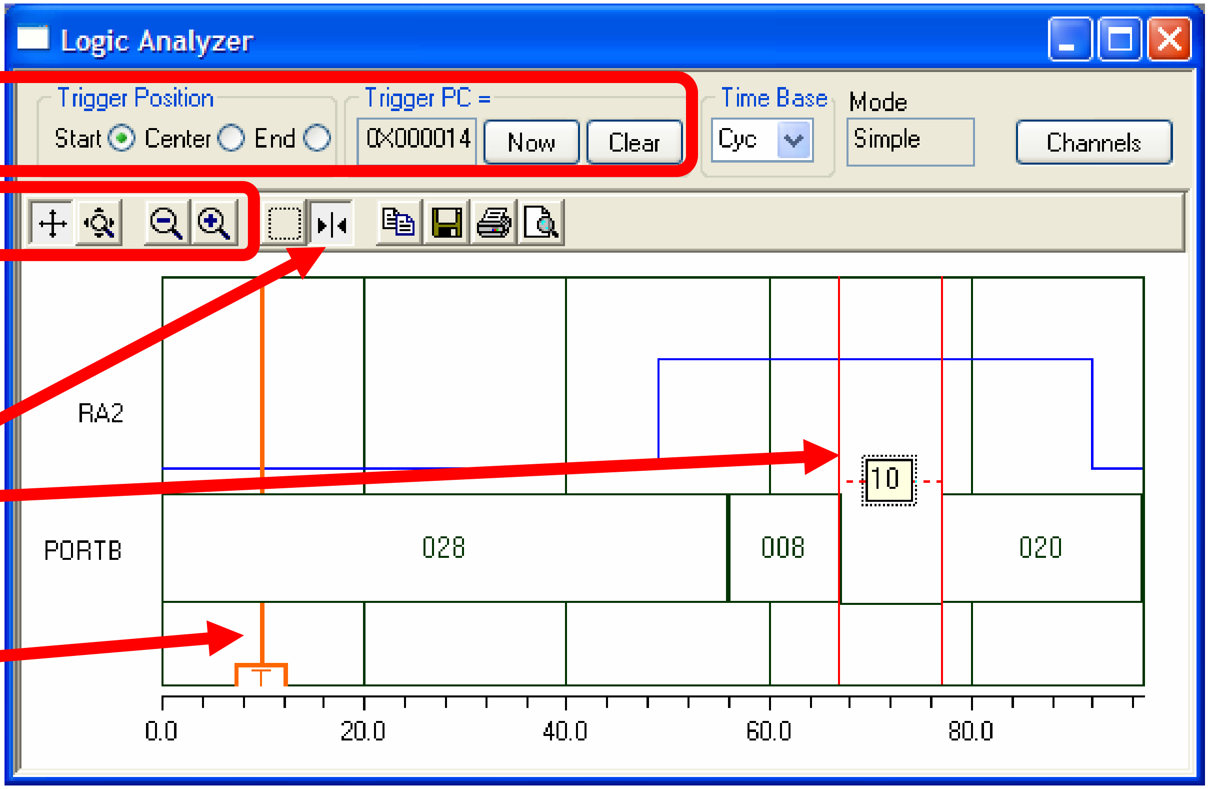
Logic Analyzer (Cont’d)

- Profile
- Watch
- Trace
- Code Coverage
- Stopwatch
- Breakpoint
- Logic Analyzer

- Simple Trigger on a PC
  - Start: history after trigger
  - End: history before trigger
  - Center: history before and after trigger
- May stop at trace buffer full
- Cursors for time measurement
- Printable displays

Logic Analyzer (Cont'd)

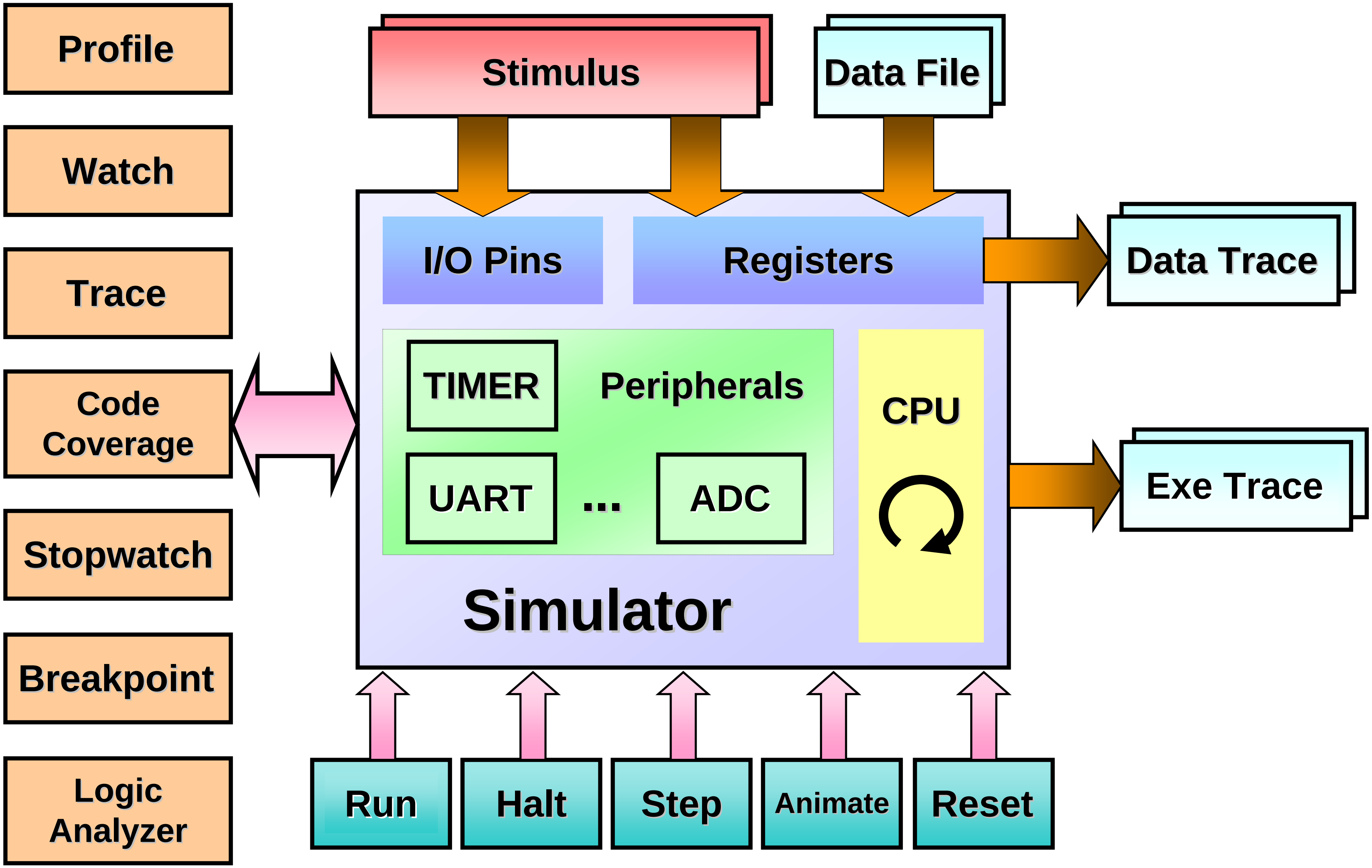
- *View>Simulator Logic Analyzer*
- Simple Trigger Definition
- Zoom
- Cursor
- Trigger



Lab 2: Logic Analyzer

C:>Masters>11014>Lab2>Lab2.mcw

Summary


Profile
Watch
Trace
Code Coverage
Stopwatch
Breakpoint
Logic Analyzer
Stimulus
Data File
I/O Pins
Registers
Data Trace
TIMER
Peripherals
UART
...
ADC
CPU
Exe Trace
Simulator
Run
Halt
Step
Animate
Reset

Summary

- **MPLAB® IDE and Simulator**
- **Stimulus**
  - Asynchronous
  - Simple Synchronous
- **Tools in Simulator**
  - Trace
  - Logic Analyzer

Where to find out more

- **MPLAB® IDE Help**
  - Topic: MPLAB SIM
- **Forums / Webinars**
  - **http://forum.microchip.com**
  - **http://techtrain.microchip.com/webseminars**

Useful Links

- Microchip Change Notification (good way to keep up to date on latest MPLAB® IDE and C18/C30 releases, as well as important Dev Tool notifications);
- http://cn.microchip.com/sales/product_change.nsf

- Microchip Dev Tools Getting Started (series of many tutorials and overviews):
- http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2122

- Microchip archives:
- http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023073

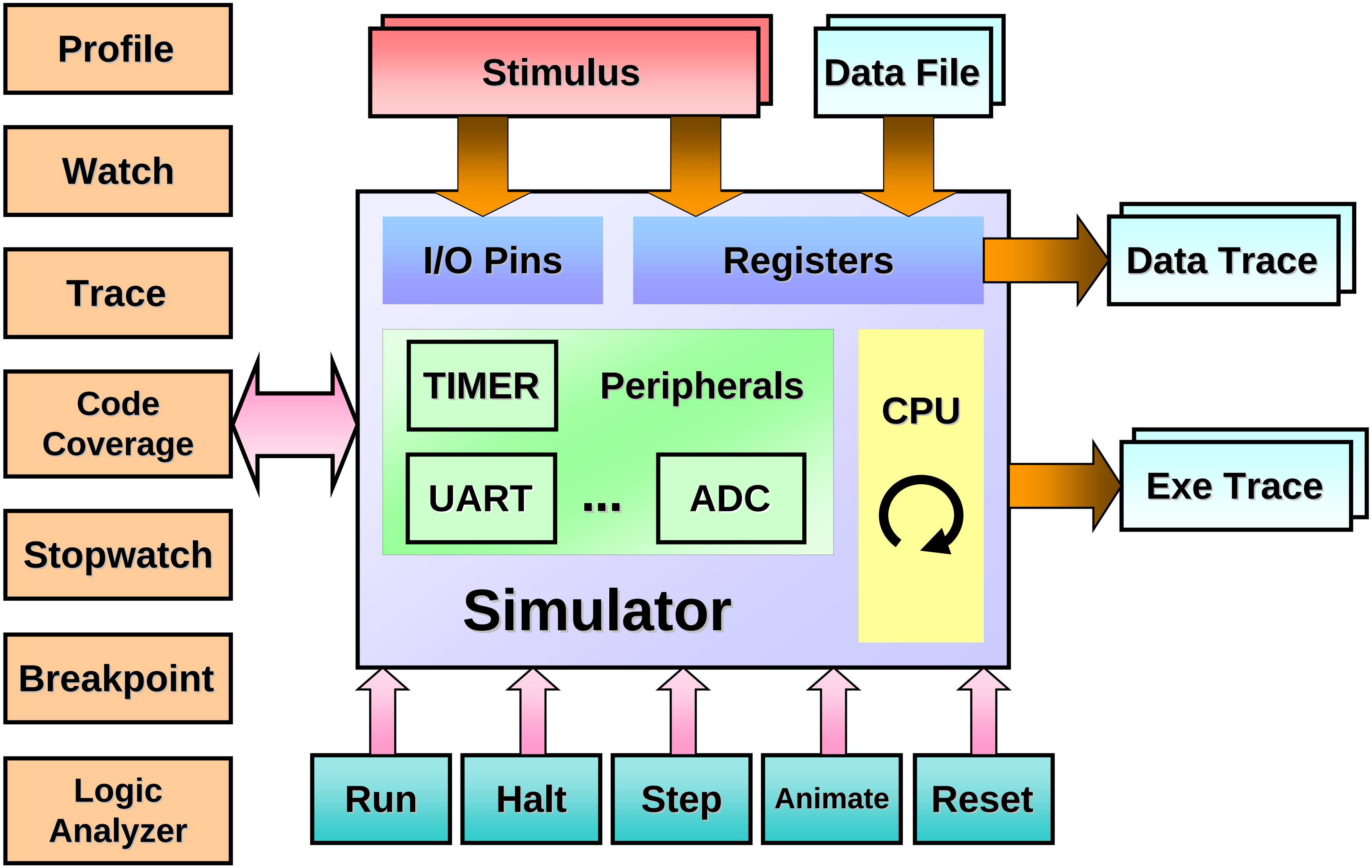
- Development Tool Selector (to find out tool support, accessories, adapteres, etc.):
- http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1496

- Third Party Development Tools:
- http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1926&type=-1&label=A

- MPLAB download page:
- http://www.microchip.com/mplab

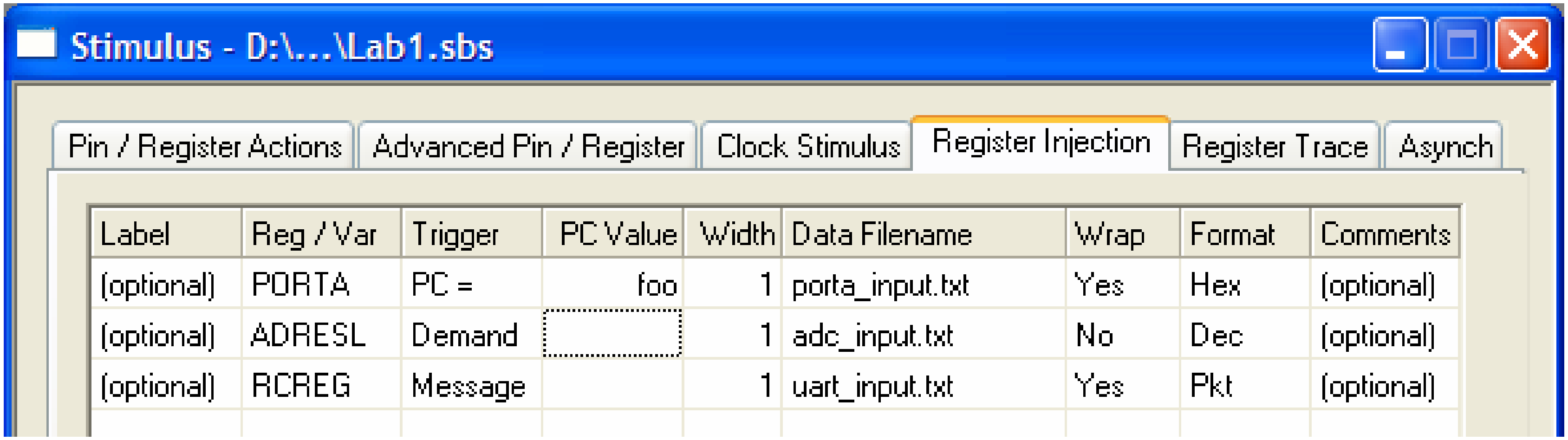
THANK YOU!

Block Diagram of Simulator



Register Injection

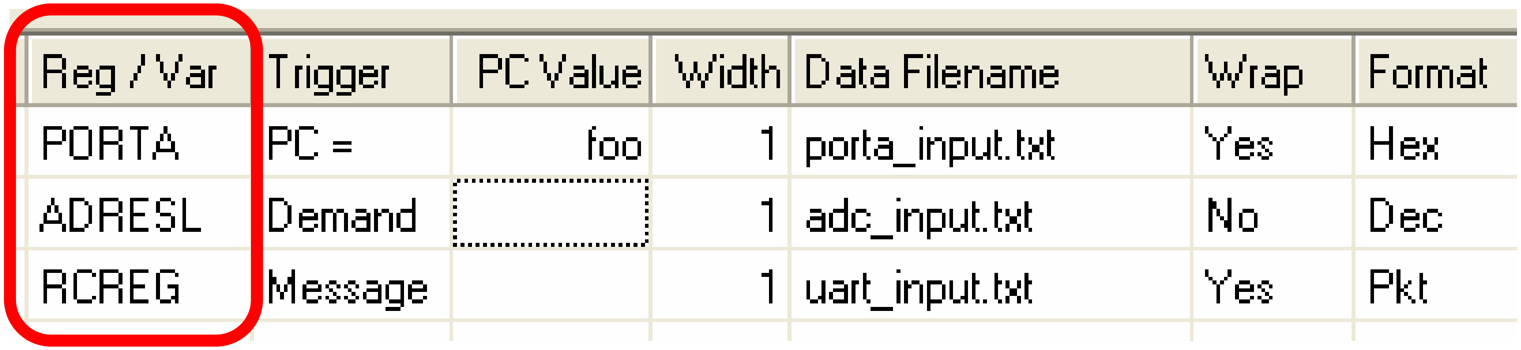
- **Inject values from data file**



Label	Reg / Var	Trigger	PC Value	Width	Data Filename	Wrap	Format	Comments
(optional)	PORTA	PC =	foo	1	porta_input.txt	Yes	Hex	(optional)
(optional)	ADRESL	Demand		1	adc_input.txt	No	Dec	(optional)
(optional)	RCREG	Message		1	uart_input.txt	Yes	Pkt	(optional)

- **Can inject a message to UART**
  Select Trigger = Message
- **Can inject to GPR (variable) also**

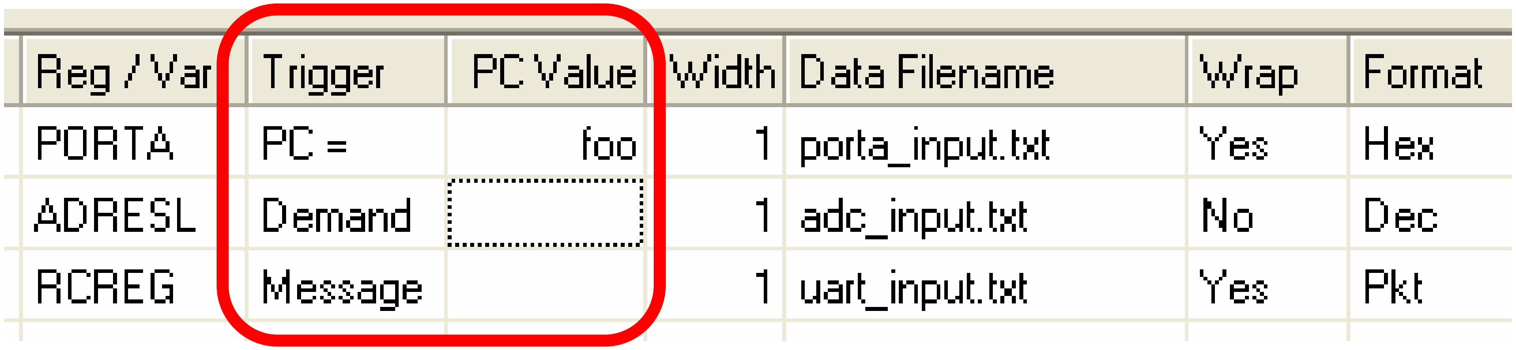
Register Injection



Reg / Var	Trigger	PC Value	Width	Data Filename	Wrap	Format
PORTA	PC =	foo	1	porta_input.txt	Yes	Hex
ADRESL	Demand		1	adc_input.txt	No	Dec
RCREG	Message		1	uart_input.txt	Yes	Pkt

- **Destination**
  - SFR (e.g. ADRESL, RCREG), or
  - GPR (aka. Variable)
    - **Select variable from drop down**
    - **Type address directly (2C0) in hex**

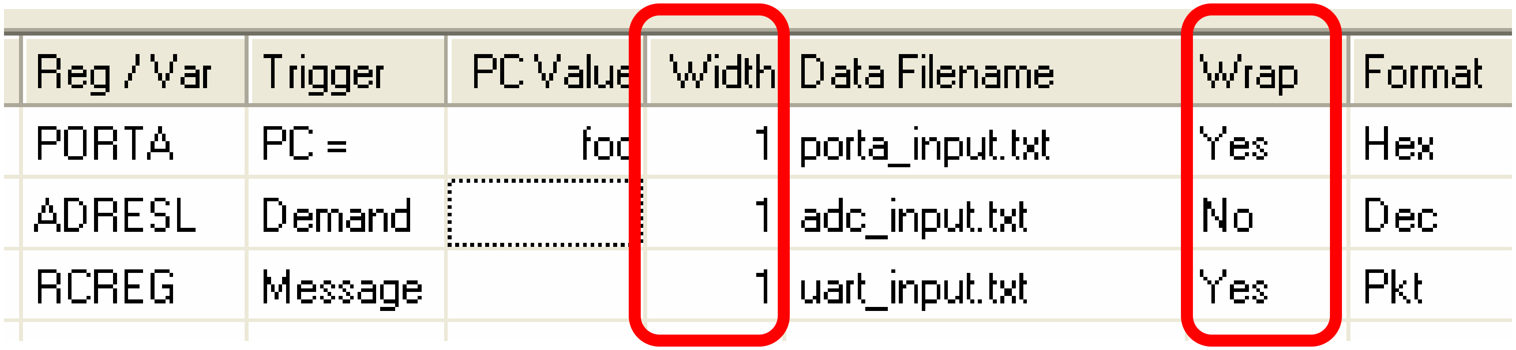
Register Injection



Reg / Var	Trigger	PC Value	Width	Data Filename	Wrap	Format
PORTA	PC =	foo	1	porta_input.txt	Yes	Hex
ADRESL	Demand		1	adc_input.txt	No	Dec
RCREG	Message		1	uart_input.txt	Yes	Pkt

- **Trigger**
  - **PC=**
    when PC reaches a location, e.g. *foo*
  - **On Demand**
    when the program reads register/variable
  - **Message (UART only)**
    data injected to RCREG per baud rate

Register Injection



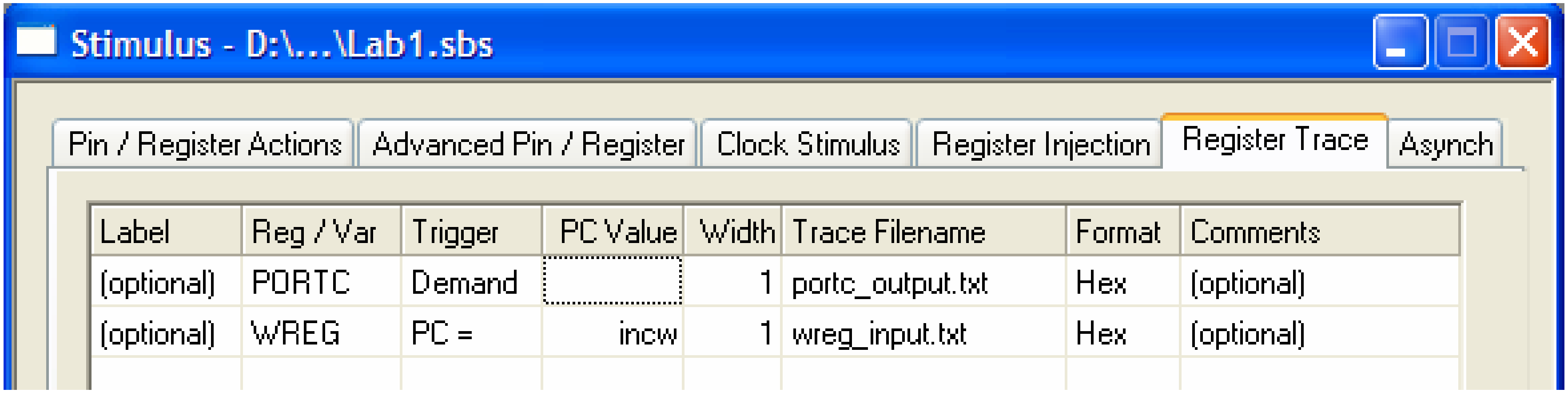
Reg / Var	Trigger	PC Value	Width	Data Filename	Wrap	Format
PORTA	PC =	foc	1	porta_input.txt	Yes	Hex
ADRESL	Demand		1	adc_input.txt	No	Dec
RCREG	Message		1	uart_input.txt	Yes	Pkt

- **Width (bytes)**
  - SFR (dsPIC® DSC: 2, others: 1)
  - GPR (user-specified)
- **Wrap**
  - Wrap-around at the end of input file

Register Injection Summary

- **Difference than Pin/Reg Actions Tab**
  - Data is not defined in the workbook
  - Use Excel-generated data
  - Can switch test data set easily
- **Possible uses:**
  - A/D simulation (ADRES)
  - Zero-crossing – covered in **Advanced Stimulus (11015 MS2)**

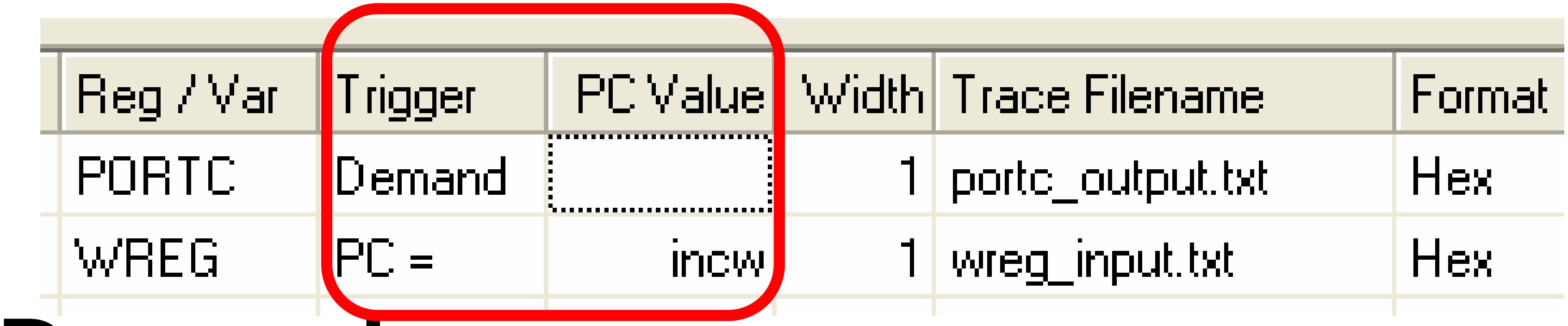
Register Trace



Label	Reg / Var	Trigger	PC Value	Width	Trace Filename	Format	Comments
(optional)	PORTC	Demand		1	portc_output.txt	Hex	(optional)
(optional)	WREG	PC =	incw	1	wreg_input.txt	Hex	(optional)

- **Similar to Register Injection Tab**
- **Sends register value to output file**
- **Supports GPR / variable**
- **No Wrap column**

Register Trace



Reg / Var	Trigger	PC Value	Width	Trace Filename	Format
PORTC	Demand		1	portc_output.txt	Hex
WREG	PC =	incw	1	wreg_input.txt	Hex

- **Demand**
  - Every time the program changes the register/variable
- **PC=**
  - Every time PC reaches a location
  - e.g. label *incw* or address 2C0

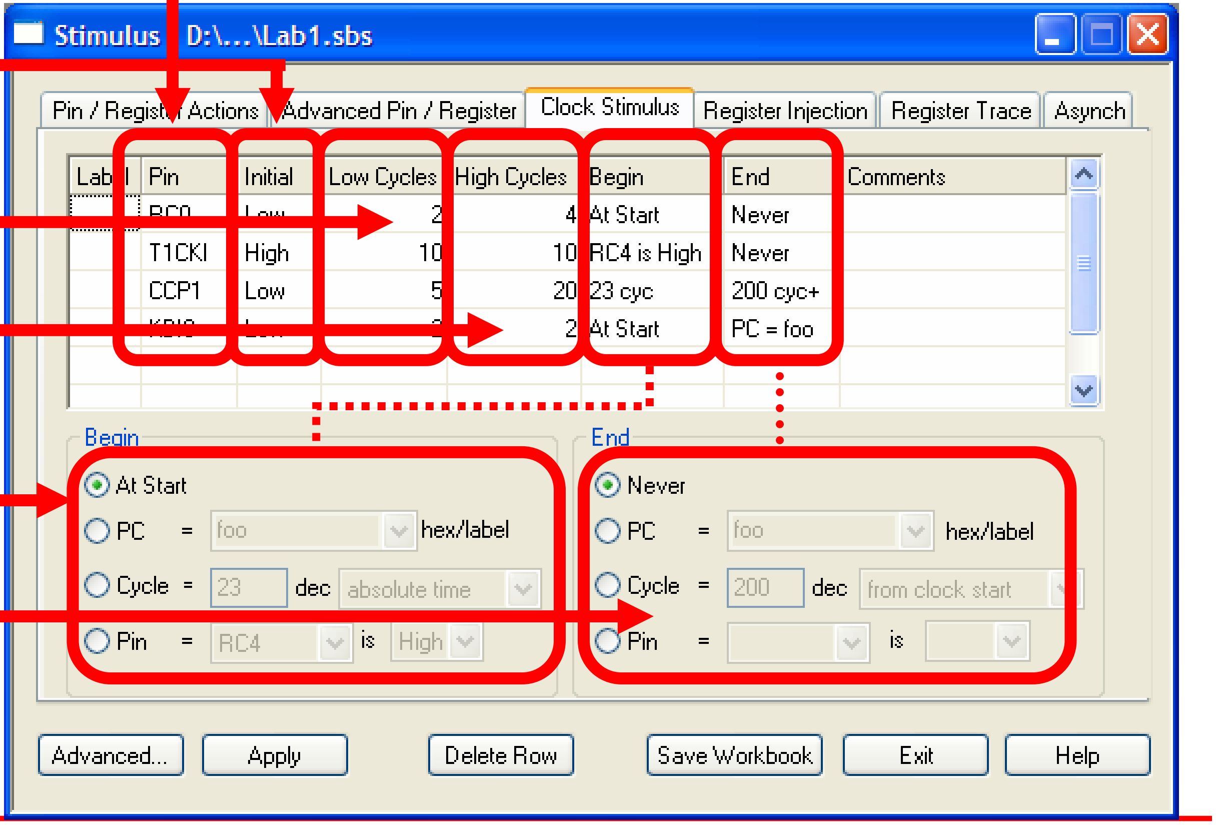
Register Data File Formats

- Format
  - **Hex**: white space delimited hex value e.g. **`1FF  AC  53`**
  - **Dec**: white space delimited dec value e.g. **`128  75  145`**
  - **Raw**: binary data recorded as-is
  - **Packet:** only for UART buffer
    - **Combo of Hex and “string”**
    - **Wait time**

```
wait 2 ms
61 62 62 20
"abc"
```

Clock Stimulus

- **Generates waveform to pin**
- Target Pin
- Initial State
- Low Cycles
- High Cycles
- Begin cond.
- End cond.



Label	Pin	Initial	Low Cycles	High Cycles	Begin	End	Comments
	RC0	Low	2	4	At Start	Never	
	T1CKI	High	10	10	RC4 is High	Never	
	CCP1	Low	5	20	23 cyc	200 cyc+	
	KBI0	Low	2	2	At Start	PC = foo	

Begin
● At Start
○ PC = foo hex/label
○ Cycle = 23 dec absolute time
○ Pin = RC4 is High

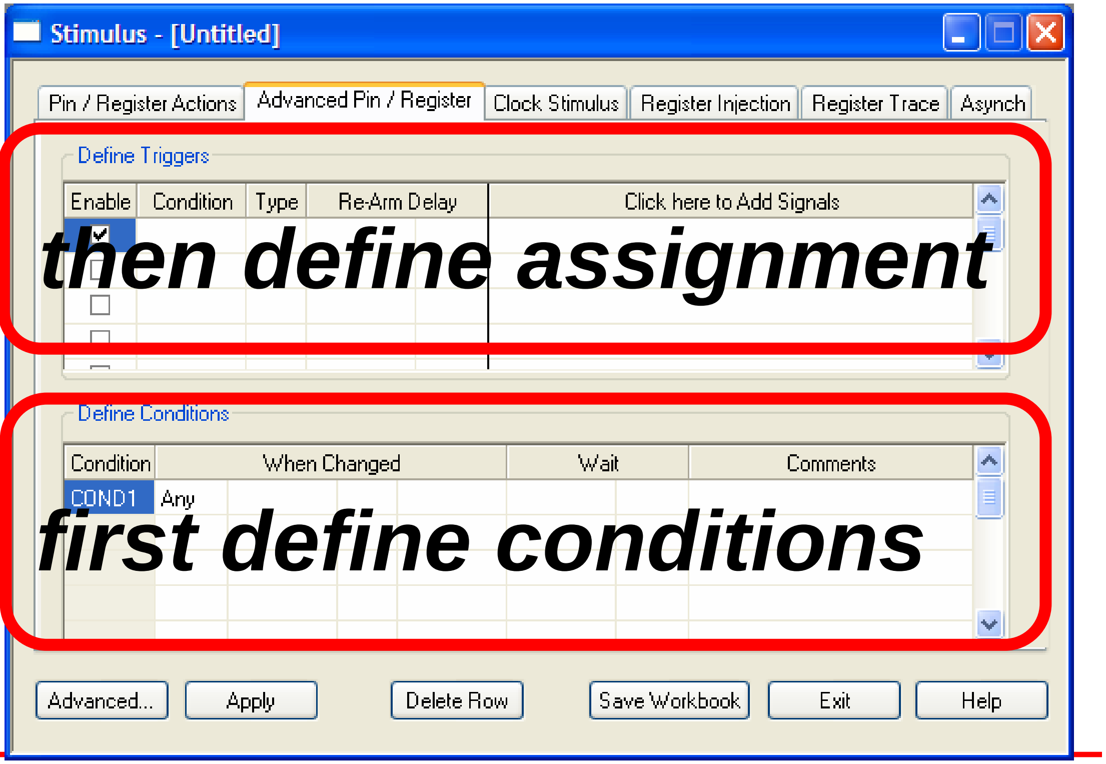
End
● Never
○ PC = foo hex/label
○ Cycle = 200 dec from clock start
○ Pin = is

Clock Stimulus

- **Any type of clocking signals**
- **Possible uses:**
  - TMR external clock signals T0CKI
  - Key bounce – covered in **Advanced Stimulus (11015 MS2)**

Advanced Pin/Register

- Pin/Register assignment with **conditional** control
- Condition
  - -If TMR2 is E0
  - -If INT1IF is ‘1’
- Assignment
  - -Triggered once
  - -Whenever cond happens



Advanced Pin/Register

- **Provides flexibility with gating**
  - Complex pin and register stimulus based on other signals
- **Possible uses:**
  - Set IRQ flags based on clock stimulus
  - Pulse train – covered in **Advanced Stimulus (12025 MS2)**

11014 MS1

MPLAB Simulator Fundamentals

(END)

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.