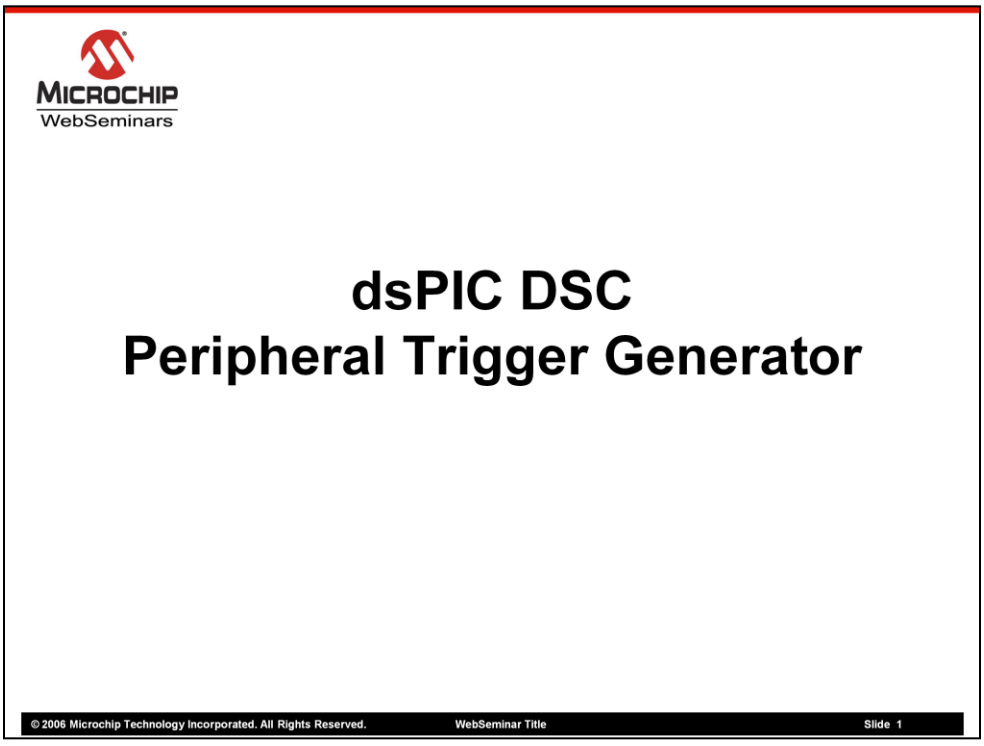
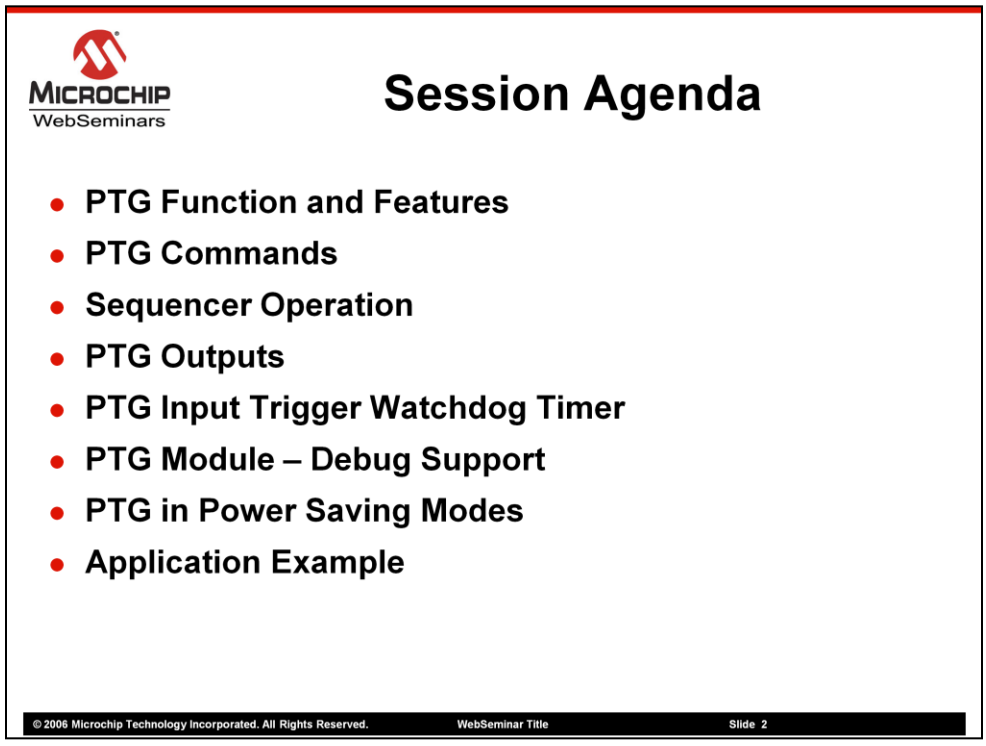


I am Jenny Puthusseri from Microchip Technology, and Welcome to **dsPIC DSC Peripheral Trigger Generator** Web seminar.



Here is the agenda for today's session

We will see what does Peripheral Trigger Generator do ? and what are its features ?

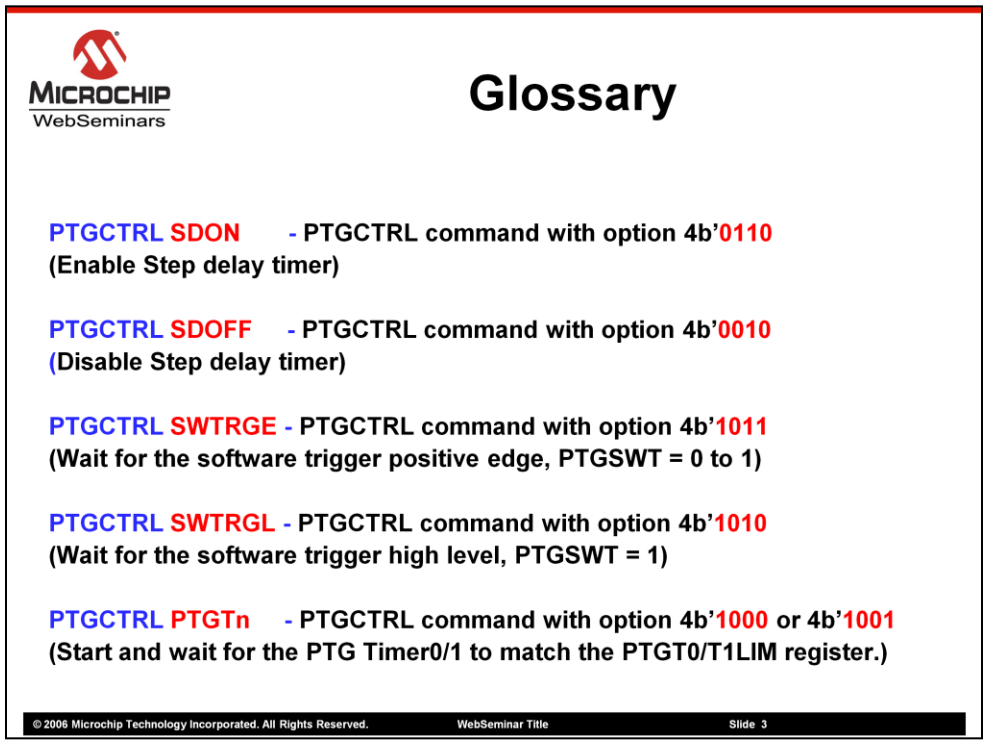
And we will have an overview of PTG Step commands and understand operation of the PTG module

Then , we will discuss about various features of this module and its outputs in detail

We will then move on to Dedicated PTG Watchdog Timer, intended to prevent PTG lock-up

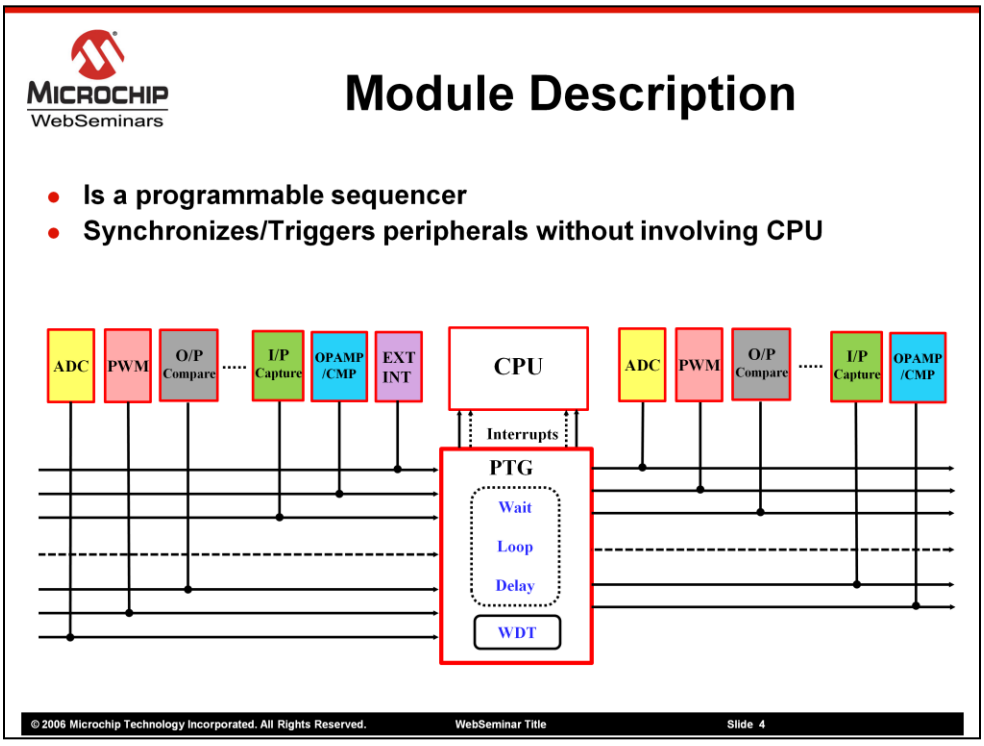
We will also see how the peripheral behaves in power save modes

At the end ,we will discuss about an application example



These terms may be referenced in next slides. They are representation used for better readability. And its actual representation in bits is displayed .

For more details refer Family Reference Manual of PTG and is available from microchip website.



Function of PTG module is to generate sequences of trigger signals to co-ordinate the operation of other peripherals

such as ADC,PWM etc as defined by the user

This module can synchronize and generate triggers without CPU intervention. Thus saving CPU execution time

That is it can select from several input triggers and generate related events such as output triggers or interrupts to

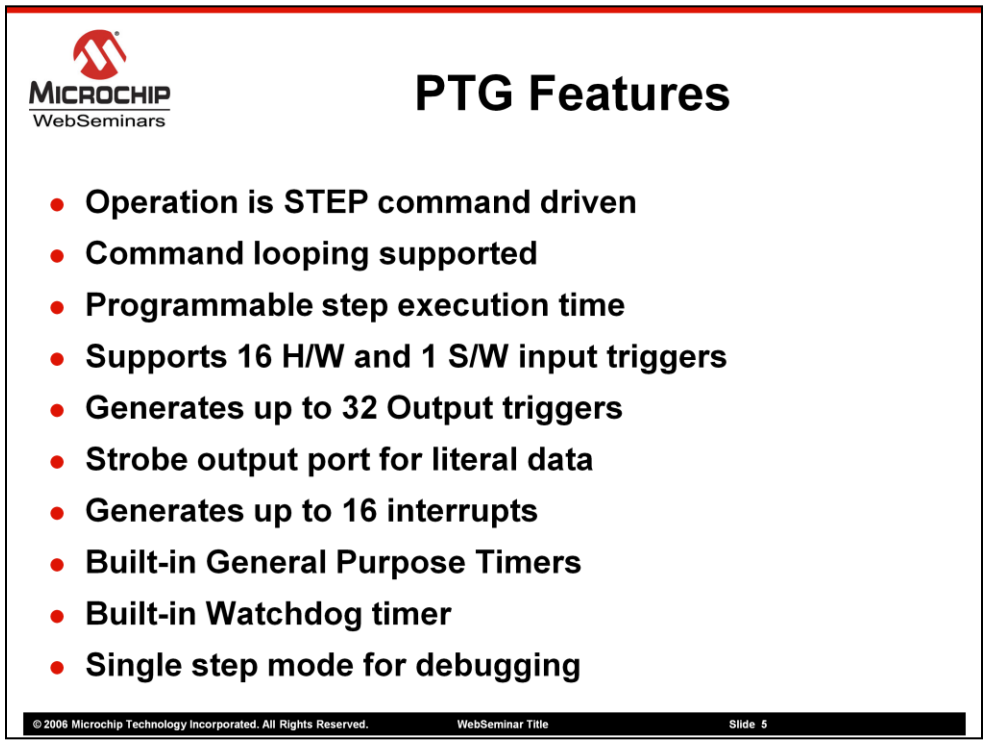
other device peripherals and processor in sequential and repeatable fashion

Possible Input triggers to the module can be from ADC, PWM, Output Compare, Input Capture, Op-Amp/Comparator, External Interrupt etc.

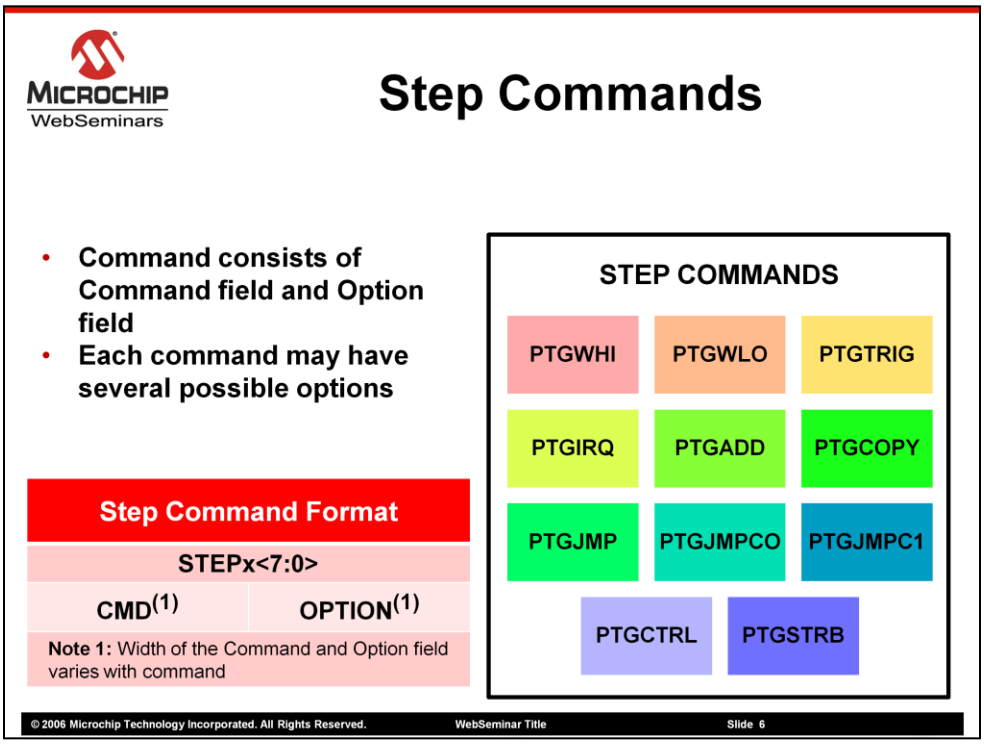
And it can trigger or synchronize peripherals like Output Compare, Input Capture, ADC, PWM, Op-Amp/Comparator etc.

Built-in counters, timers and sophisticated step commands enables PTG module for performing complex operations involving

repeated trigger generation at different intervals



These are the features of PTG Module implemented in Microchip dsPIC Controllers.



As discussed, this module operation sequence is user programmable through PTG Commands.

And is done by writing 8-bit command called Steps into PTG Queue registers.
And each command consists of Command field and option field.

Option field is used to mention exactly which one of the several possible input/output options the command should act upon.
It specifies which trigger input to wait for and which trigger or interrupt to be generated
Now ,we will discuss about PTG step commands

The PTGWHI/PTGWLO waits for level or edge sensitive triggers on any of the available hardware inputs

PTGWHI – Wait for Positive edge or high level of trigger, whereas
PTGWLO – Wait for Negative edge or low level of trigger
PTGTRIG command generates selected trigger signal on the trigger bus
PTGIRQ generates specified Interrupt request signal to the CPU

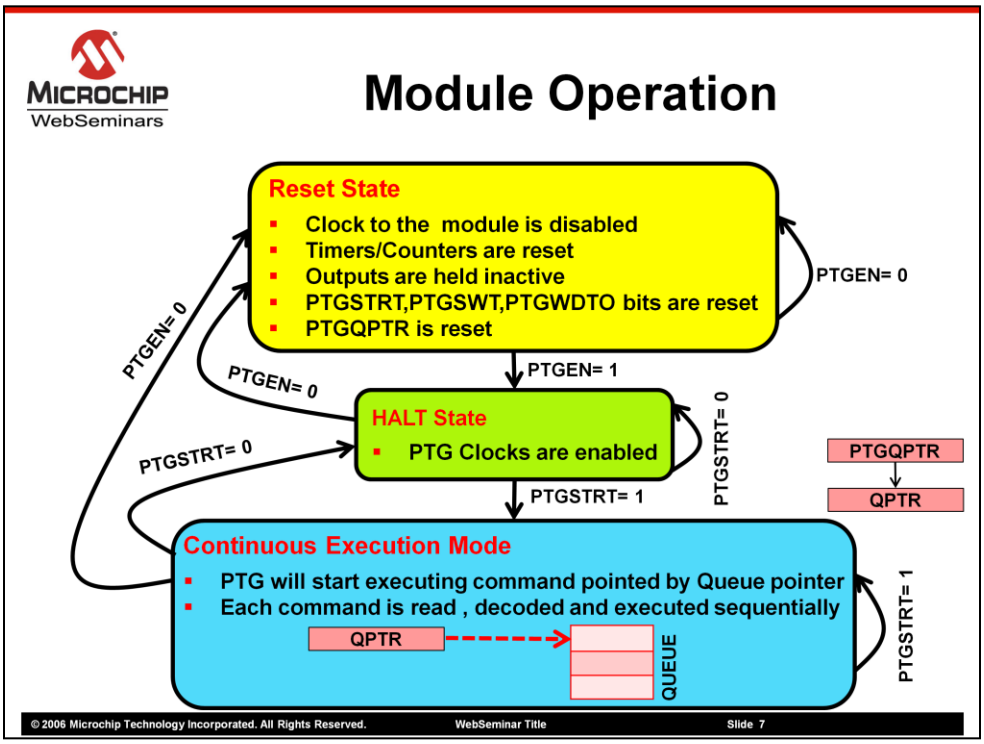
PTGADD/PTGCOPY commands are used to modify the value of Counter/Timer Limit Registers, Step Delay Limit Register or Literal register during sequencer execution
PTGADD - Add the contents of PTGADJ register to the specified source register and stores the result in the same register

PTGCOPY- Copy the contents of PTGHOLD register to the specified destination register

Commands PTGJMP,PTGJMPC0 and PTGJMPC1 changes the flow of execution

PTGJMP - Performs an Unconditional jump to the specified queue location/step
Where as PTGJMPC0 and PTGJMPC1 are conditional Jump commands.

PTGCTRL – Executes the control function defined by the 4bit Option field
Using this command with appropriate option ,user can
Start and Wait for General Purpose Timers,
Enable /Disable Step Delay Timer,
Generate 16 bit Strobe outputs,
Generate multiple triggers,
Wait for edge or level sensitive Software trigger.
PTGSTRB command is used to generate 5-bit strobe output



User initializes STEP Queue registers with the step commands(8bit) and These step Commands define the sequence of events

We will look at the state flow diagram of the PTG Sequencer, to understand its operation

When the module is disabled(PTGEN = 0),the clocks to the module is disabled for maximum power savings

As shown in this state, all the timers/counters are reset. And all the PTG Outputs are driven inactive. The queue pointer will also be reset.

When the module is enabled by setting PTG Enable bit (PTGEN),it enters halt state. In this state all module clock are enabled.

Module will continue in this state, if the PTG Sequencer Start bit(PTGSTRT) is 0.

When PTG Sequencer Start bit(PTGSTRT) is set while in halt state,sequnecer will enter continuous execution mode.

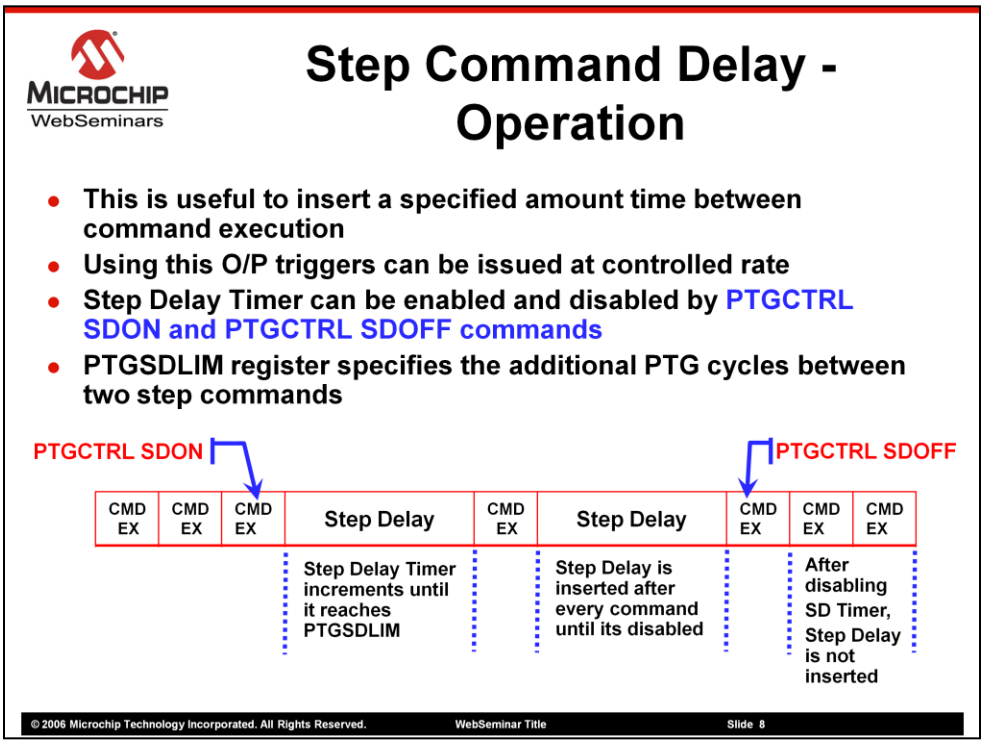
When entering this mode Queue pointer SFR register value is transferred to the internal queue pointer.

You must first set PTG Enable bit (PTGEN), and subsequently set PTG Sequencer Start bit(PTGSTRT) ,to enter continuous execution mode.

After entering this mode, module will read, decode and execute the command pointed by the internal queue pointer.

And after executing each command internal queue pointer is incremented, except if it is executing jump queue commands.

And the sequencer will exit this mode, if the module is disabled , if PTG Sequencer Start bit(PTGSTRT) is cleared or if Watchdog time out event occurs



Step Command delay is useful to insert a specified amount of time between command execution

By using this feature O/P triggers can be issued at controlled rate ,to avoid overloading of the target peripherals

For example ,consider PTG is configured for generating output triggers to a slower external device.

Then this delay can be inserted between the triggers to match its clock rate.

Step Delay Timer can be enabled or disabled by PTGCTRL command

As shown in the diagram, Executing the PTGCTRL SDON command, inserts specified step delay after executing every command.

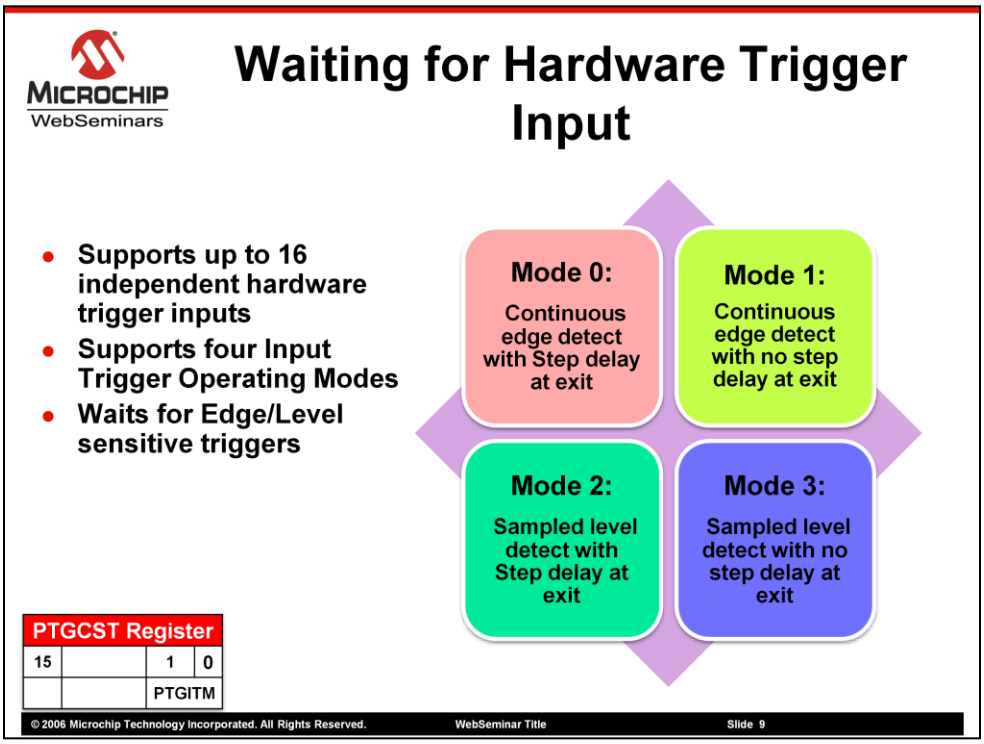
The PTG Step delay timer(SDLY) increments at PTG Clock rate after completion of a command execution.

When this matches with Step delay timer limit value(PTGSDLIM),

execution of next command will begin and timer will be reset.

And if PTGCTRL SDOFF is executed the step delay is not inserted any more.

PTG Step Delay Limit Register(PTGSDLIM), may be modified by PTGADD/PTGCOPY commands



This module supports up to 16 independent hardware trigger inputs

Example for hardware trigger inputs are ADC conversion done Interrupt,PWM Interrupts, External Interrupt and PWM Master Synchronization output

PTG module can be configured to wait either for positive or negative edge transitions or for high or low level sensitive triggers

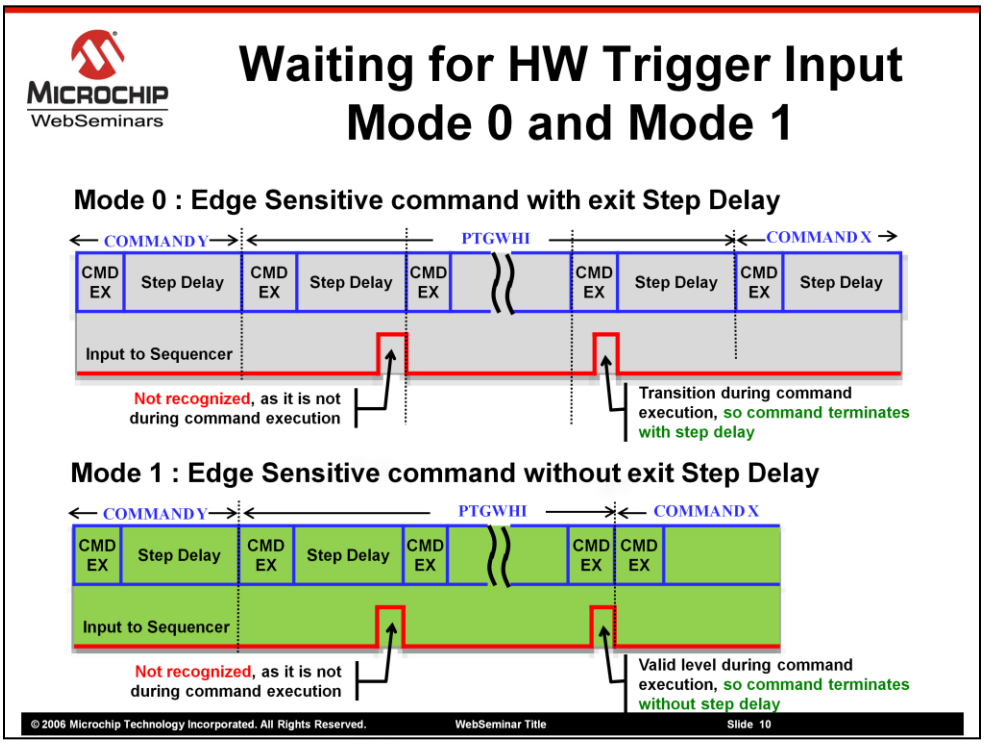
The two commands that enables waiting of hardware input triggers are :
PTGWHI and PTGWLO

While executing these commands, sequencer will repeat execution until valid edge or level is detected on the selected trigger input

There are four input trigger operating modes and is shown here.

Operating Modes are specified by writing to PTG Input Trigger Command Operating Mode bits (PTGITM[1:0])

Next slides explains Waiting modes in detail



We will now discuss about hardware input trigger waiting modes in detail.

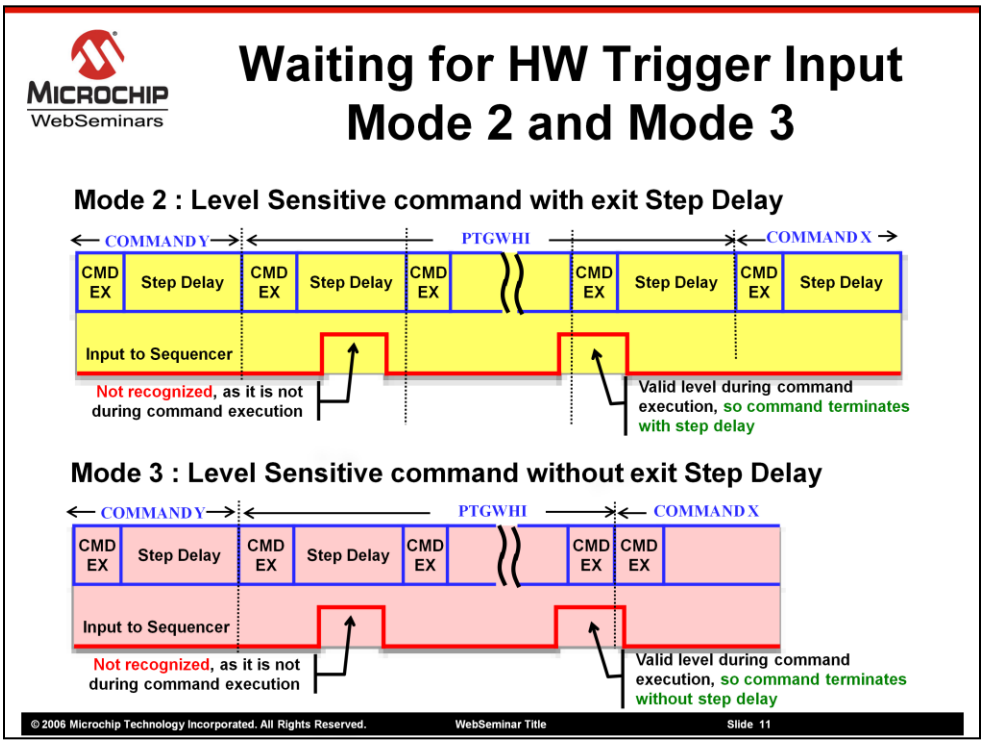
Look at the example timing diagram ,executing PTGWHI command in Mode 0 and Mode 1.

Both Mode 0 and Mode 1 are edge sensitive trigger input modes.

In Mode 0 if the valid edge is detected, command execution completes. If the step delay timer is enabled , the step delay is inserted before executing next instruction.

Where as in Mode 1,if a valid edge is detected ,whether step delay timer is enabled or not ,step delay is not inserted after command execution has completed.

So, if the step delay timer is disabled ,both modes are identical in operation.



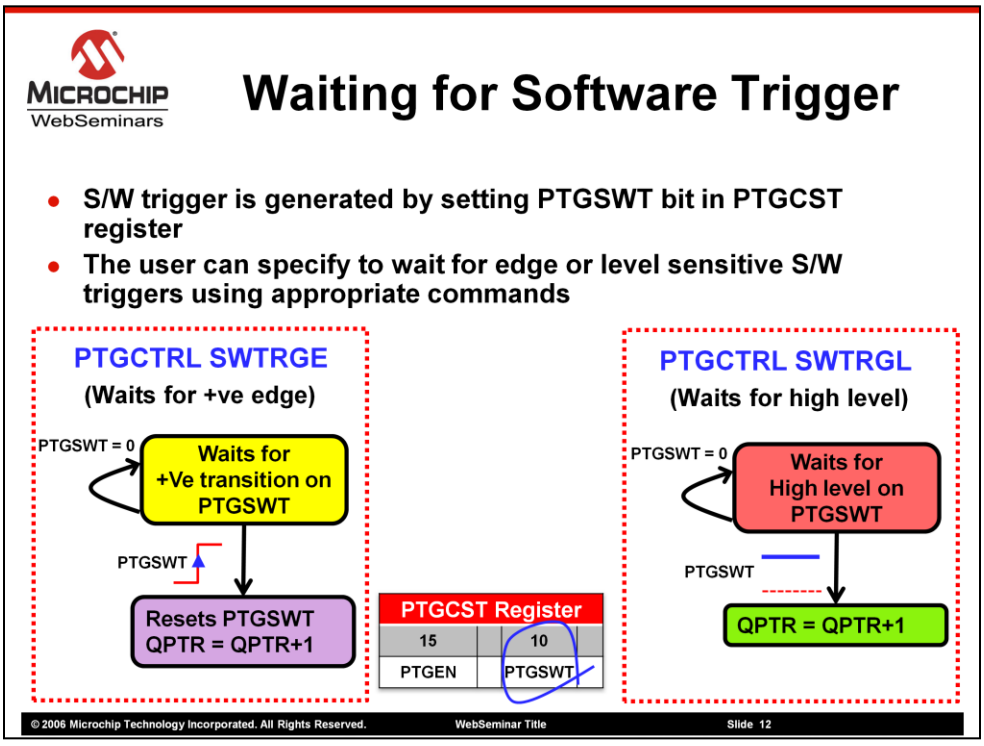
Look at the example timing diagram ,executing PTGWHI command in Mode 2 and Mode 3.

Both Mode 2 and Mode 3 are level sensitive trigger input modes.

In Mode 2 if the valid level is detected, command execution completes. If the step delay timer is enabled , the step delay is inserted before executing next command.

Where as in Mode 3,if a valid level is detected ,whether step delay timer is enabled or not ,step delay is not inserted after command execution has completed.

So, if the step delay timer is disabled ,both modes are equivalent in operation.



S/W trigger wait operation are similar to hardware trigger wait operation.

S/W trigger is generated by setting PTG Software Trigger bit(PTGSWT)

The user can specify to wait for edge or level sensitive triggers using PTGCTRL command with appropriate option

PTGCTRL SWTRGE(b'1011) command is sensitive only to positive transition(0 to 1) on PTG Software Trigger bit(PTGSWT)

This transition must occur during command execution, otherwise command will continue to wait

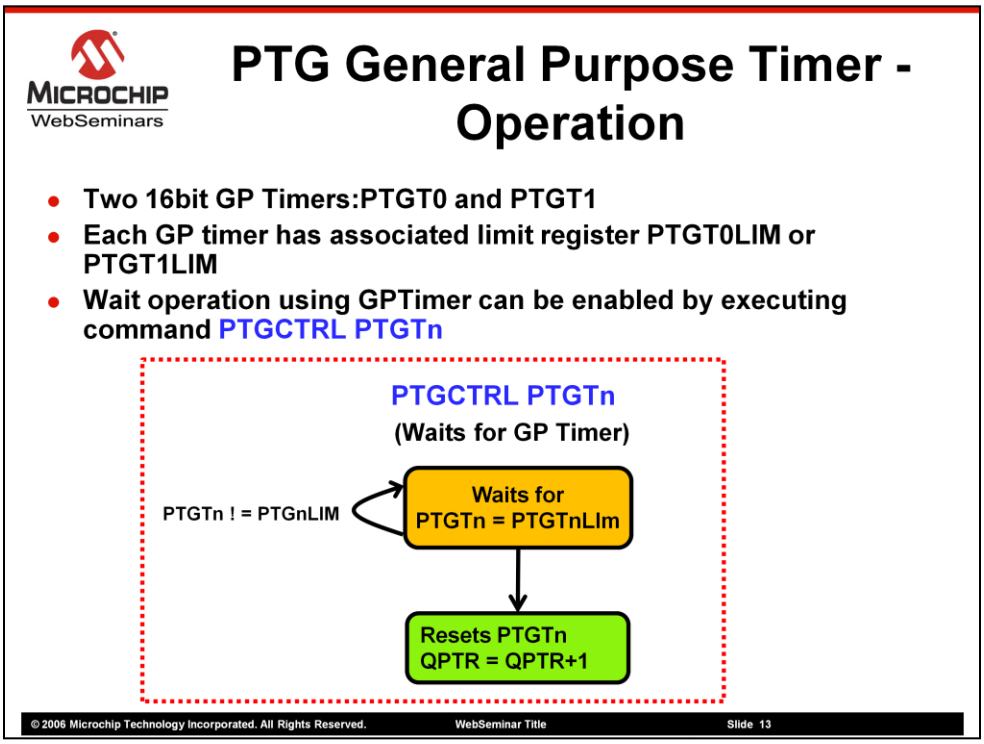
PTG Software Trigger bit(PTGSWT) is cleared automatically upon completion of command.

PTGCTRL SWTRGL(b'1010) command is sensitive only to high level of PTG Software Trigger bit(PTGSWT).

Command will wait until it observes PTG Software Trigger bit(PTGSWT) is 1.

Note that PTG Software Trigger bit(PTGSWT) is not cleared in this case,

So, If desired this bit may be cleared by the user



We will discuss about the operation of General Purpose timers in the PTG module.

This PTG feature will be very useful, to create delay and execute wait operation between two subsequent events.

For example ,After receiving an external interrupt from a sensor ,if you would like to wait for a certain amount of time before measuring a sensor signal through ADC. Then the timers can be configured to insert the required amount of delay between external interrupt and ADC sample and conversion.

There are two 16bit General purpose Timers namely PTGT0 and PTGT1.

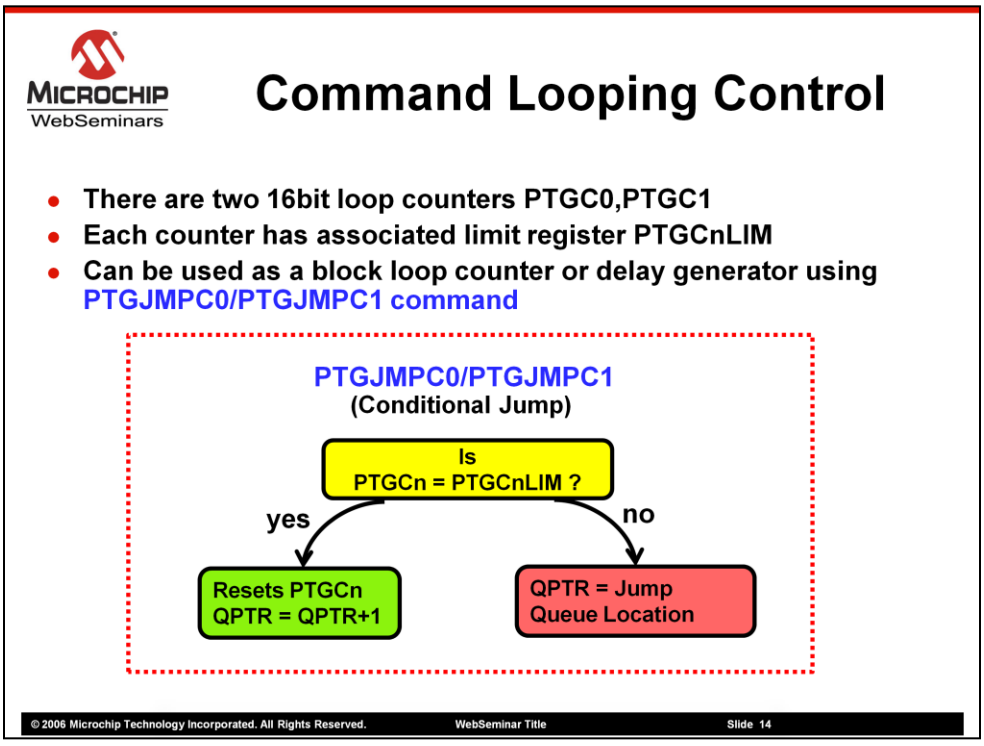
Each General Purpose timer has an associated internal counter and limit register (PTGTnLIM)

Wait operation using General Purpose Timers can be enabled by executing command PTGCTRL with appropriate option

While executing this command, associated timer increments on every PTG clock rising edge

If timer value is not equal to limit register (i.e PTGTn!= PTGTnLIM) ,then PTG Sequencer will continue to execute the same command and remain in this step

When contents of timer is equal to limit register(PTGTn = PTGTnLIM),the command is complete and the counter is reset. And next command will begin to execute



Command looping Control Enables the PTG sequencer to execute block of commands repeatedly.

Module has two 16bit loop counters PTGC0 and PTGC1.No of iterations are specified by writing to their associated limit registers

(PTGC0LIM and PTGC1LIM).Block loops are implemented through commands PTGJMPC0 and PTGJMPC1 in association with these

Registers.

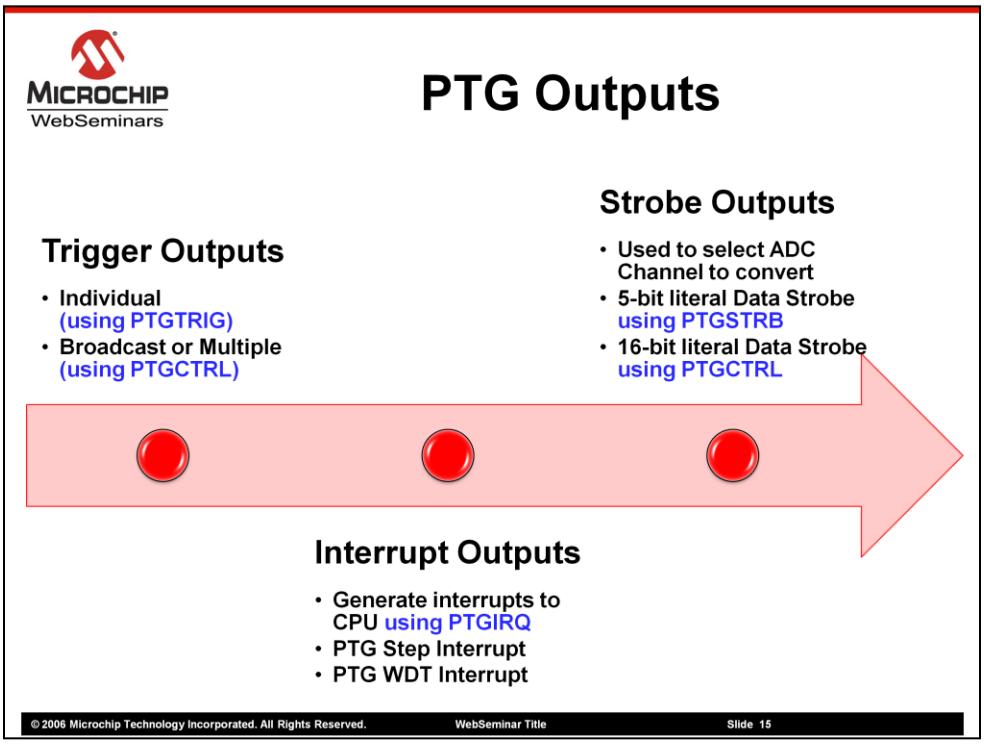
The provision of two loop counters and associated jump(PTGJMPCn) commands ,allows one level deep nested loops

We will now look at implementation of looping control with in the module.

Each time the PTGJMPCn command is executed the corresponding internal loop counter is compared to its limit value

If the counter is not reached the limit value ,the queue pointer is loaded with target Jump queue location and counter is incremented by 1

If the counter reaches the limit value ,the sequencer will move to next step command and resets loop counter



Now, we will see module outputs generated through execution of various commands.

Module can generate up to 32 Trigger outputs. It can generate Individual or Multiple(Broadcast) triggers.

Individual trigger can be generated using PTGTRIG command. For example generating a trigger for ADC sample and conversion to begin.

The Broadcast trigger capability is specified by PTG Broadcast Trigger Enable Register(PTGBTE) ,each bit in this register corresponds to each individual trigger output.

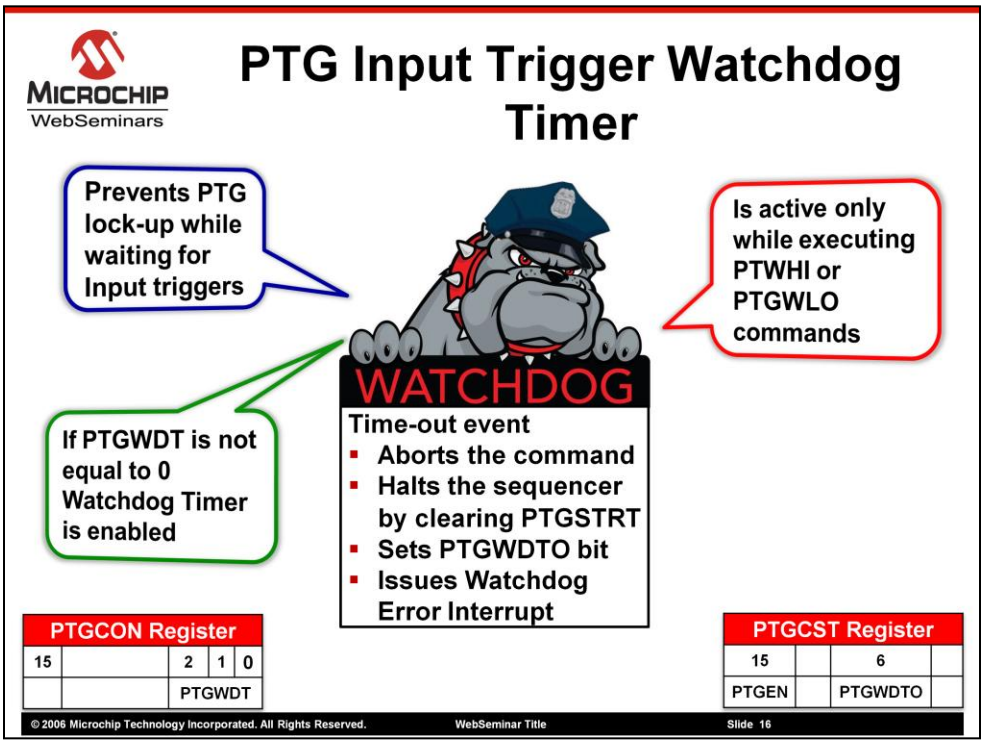
If a bit was set in the PTG Broadcast Trigger Enable Register(PTGBTE),
then while executing PTGCTRL command with Broadcast trigger option ,
the corresponding individual trigger output is asserted.

This is useful in generating multiple triggers simultaneously.

Module can generate up to 16 interrupt request signals to CPU.This is useful in performing more complex operation in coordination with CPU.
In addition to this ,it can generate two module operation interrupts:
PTG Step Interrupt and PTG Watch Dog timeout interrupt

Module can generate 16-bit wide Strobe data Outputs. This is useful writing a data to other module registers.
For example ,the specified data can be written to ADC Channel Select Register (AD1CHSO),so that PTG can select which channel to be converted

PTGSTRB command will drive 5bit data literal/option embedded in the command to the Data output bus.
Using PTGCTRL command with suitable option can copy contents of PTG Counters/PTG Literal register(PTGC0,PTGC1 or PTGLO) to ADC Channel select register AD1CSHO



Now, we will discuss about PTG Input Trigger Watchdog Timer.

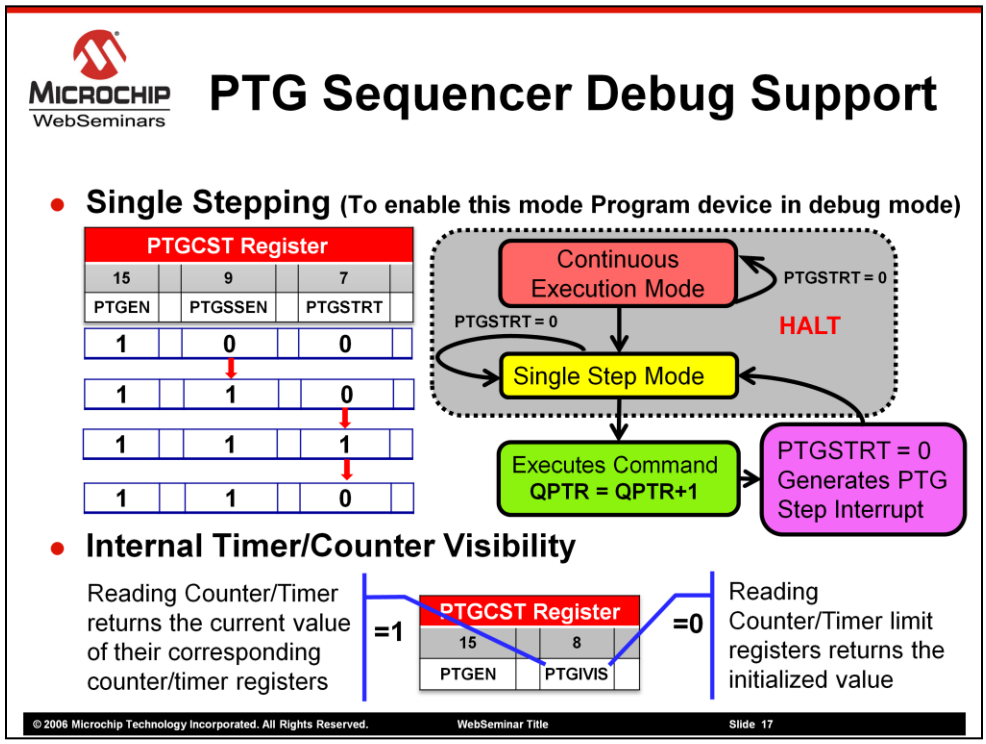
This is intended to prevent PTG Sequencer lock-up ,if input trigger event never arrive

Note that ,Watch Dog Timer is active only if the PTG sequencer is waiting for hardware input triggers.

Watch Dog timer configuration bits are present in PTG Control register(PTGCON) .
You can initialize PTG Watch Dog Time-out Count value bits(PTGWDT[2:0]),
with the total number of PTG Cycles that the sequencer may wait for an input trigger occurrence.

And if these bits are initialized with any value other than zero, watch dog timer is enabled.

PTG performs these actions, If the expected trigger input fails to arrive before Watch dog timer time-out period expires.



PTG Module support two features to enable user to debug the step command sequence

➢First one is Single Stepping

Module can be made to single step through the command sequence when the device is in debug mode

If PTG Enable and PTG Single Step bits are set(i.e PTGEN = 1,PTGSSEN =1) when device is in debug mode ,PTG operates in single step mode.

Module waits until PTG Sequencer Start bit(PTGSTRT) is set by the user, to begin executing command

When PTG Sequencer Start bit(PTGSTRT) is set, PTG Module will execute command in the queue and increments queue pointer.

After executing the command PTG Sequencer Start bit (PTGSTRT)is automatically cleared and PTG single step interrupt is generated

User may thus step through the step queue one command at a time

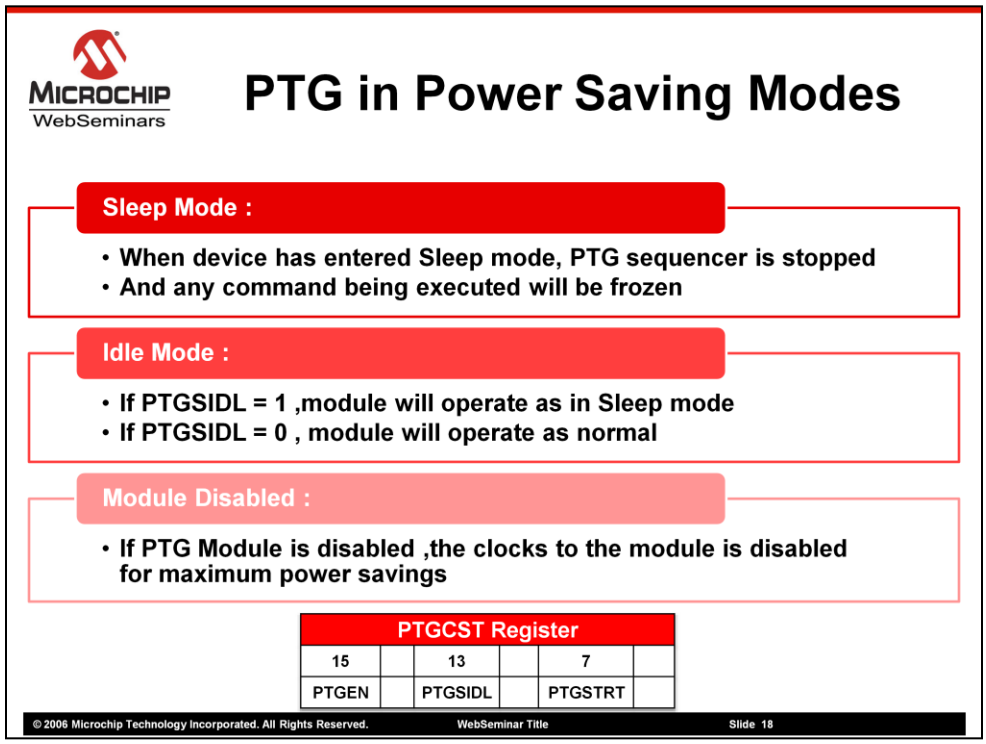
➢And other debugging support is allowing Visibility of Internal Timer/Counter

PTG Timer/Counter Visibility bit(PTGIVIS) enables the user to read the contents of
General Purpose timers(PTGT0 & PTGT1),
Loop Counters (PTGC0 & PTGC1) and Step Delay Timer(PTGSD)

When PTG Timer/Counter Visibility bit(PTGIVIS) is set as 1,a read of associated limit registers will return the current value of corresponding timer/counter at the time of read

When PTG Timer/Counter Visibility bit(PTGIVIS) is set as 0,a read of limit registers returns the value written to it at prior time

And is available irrespective of device operating modes



Let us study the operation of PTG in device power saving modes

If device enters sleep mode, all clocks are suspended. If PTG module is enabled while device entering sleep mode,

sequencer will be stopped in its current state. And any command being executed will be frozen until the clock resumes

It is recommended to shut down the module prior to entering Sleep mode, to avoid any unexpected operation

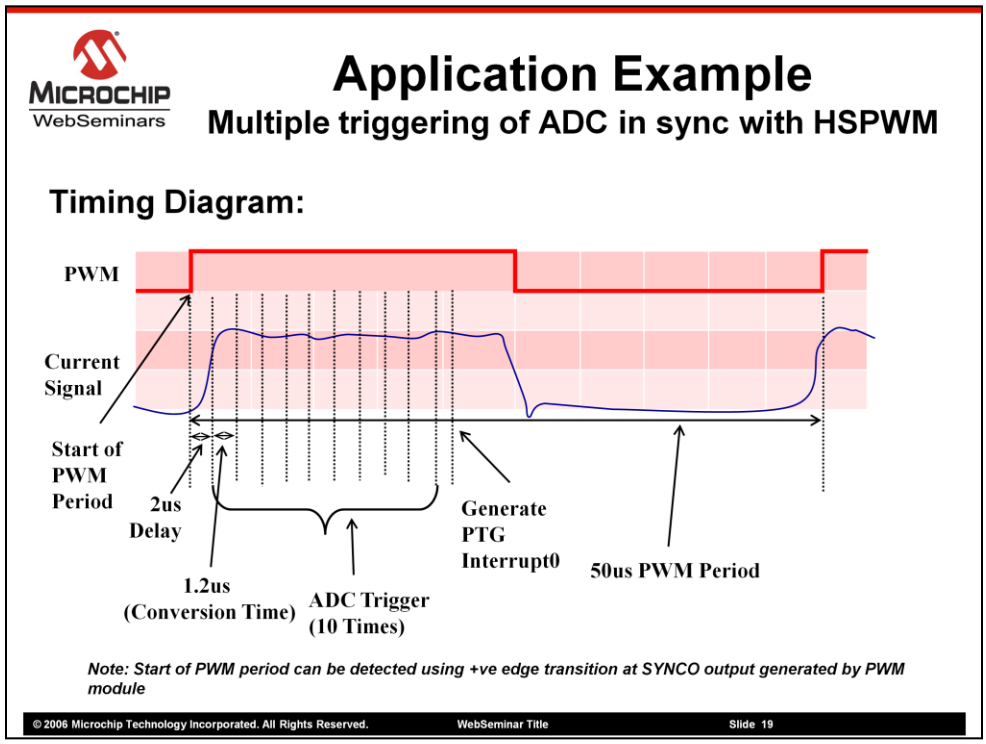
In Idle mode ,PTG can be completely functional according to the value of PTG Stop in Idle Mode bit(PTGSIDL))

If PTG Stop in Idle Mode bit(PTGSIDL) is set, module operation will be discontinued .And will operate as in Sleep Mode

If PTG Stop in Idle Mode bit(PTGSIDL) is reset, unit will operate normally

If the module is disabled(PTGEN = 0)

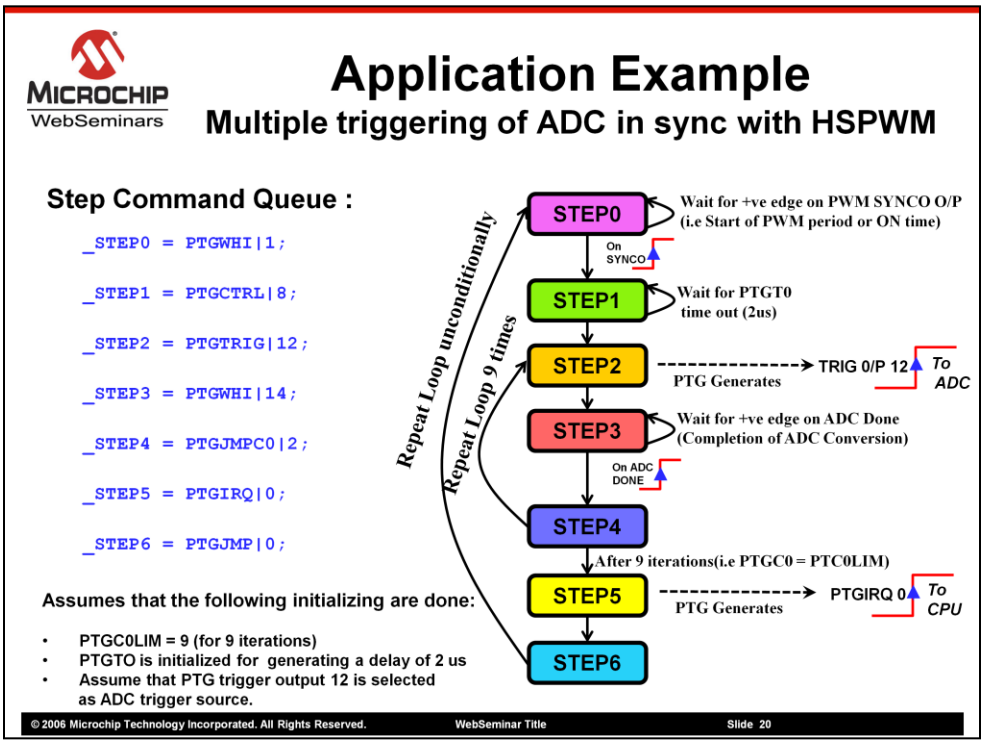
Then all internal PTG clocks are disabled and the module enters its lowest power mode



Now will consider an application example to trigger ADC multiple times synchronizing with High Speed PWM Module.

Let us first look at the timing diagram of the application. As shown, we have a PWM of period 50us generated using High Speed PWM module. From the start of PWM ON time waits for 2 us, and then generate ADC triggers to sample the current signal ten times. And after generating ADC trigger, sequencer is will wait for ADC conversion to complete ,before generating next trigger. At the end generates a PTG Interrupt(PTG interrupt0) to process the acquired current samples.

In the next slide we will see how it can be implemented using PTG commands.



We will have the following initialization done ,

Initialize the timer PTG Timer 0 to wait for 2us.

Initialize PTG Counter Limit register 0 with 9 to repeat the ADC trigger generation 9 times.

And configure PTG Trigger Output 12 as ADC trigger source, in ADC Configuration register.

Now, we will look at the Sequencer Execution Steps.

The Step Queue initialization commands and State flow is shown in the diagram.

Every time a step execution completes,sequecer will begin to execute next step.

Step 0 of the command Queue is initialized to wait for Master Synchronization output, to transition from low to high as an indication of start of PWM ON time and period.

Step 1 starts PTGTimer 0 (PTGT0) and waits for time out creating a delay of 2us before starting adc sampling.

Step 2 triggers ADC conversion through PTG trigger output 12 ,which is selected ADC trigger source.

Step 3 Waits for ADC conversion to complete. When AD conversion is done, the sequencer will execute next command in the queue.

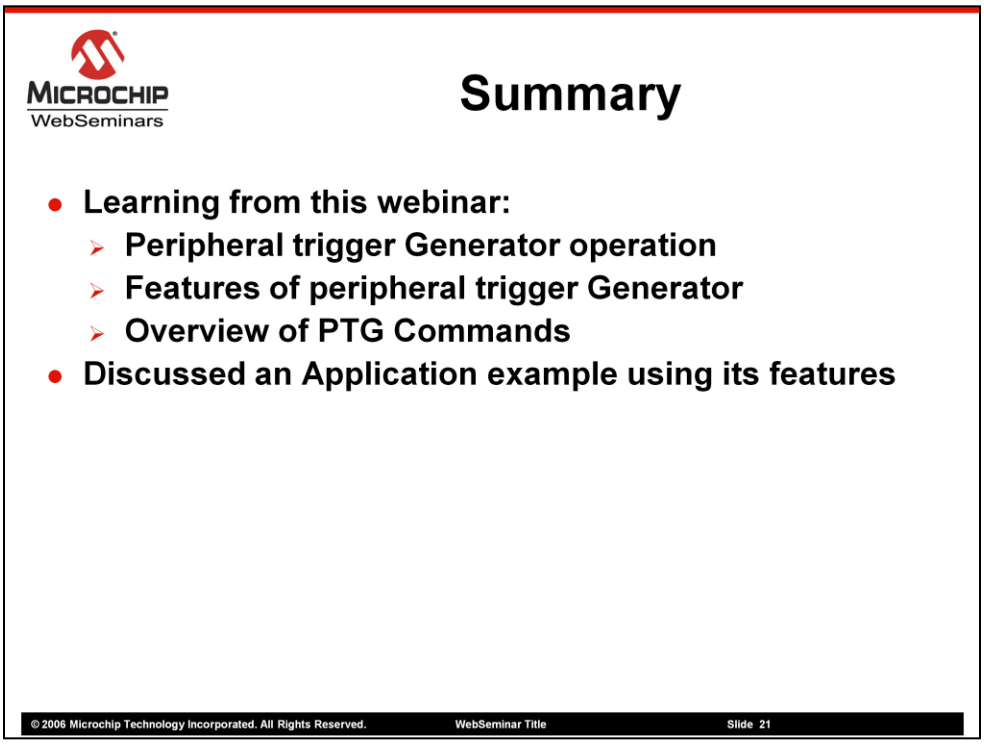
Step 4 is a jump queue command, to enable multiple triggering of ADC.This command enables the sequencer to jump to Step 2 ,nine times.

Thus triggering ADC additionally nine times.

Step5 is command to generate PTG Interrupt 0,to process the samples.

Step 6 is an unconditional jump to repeat the queue so that Sequencer will continue from Step0 .

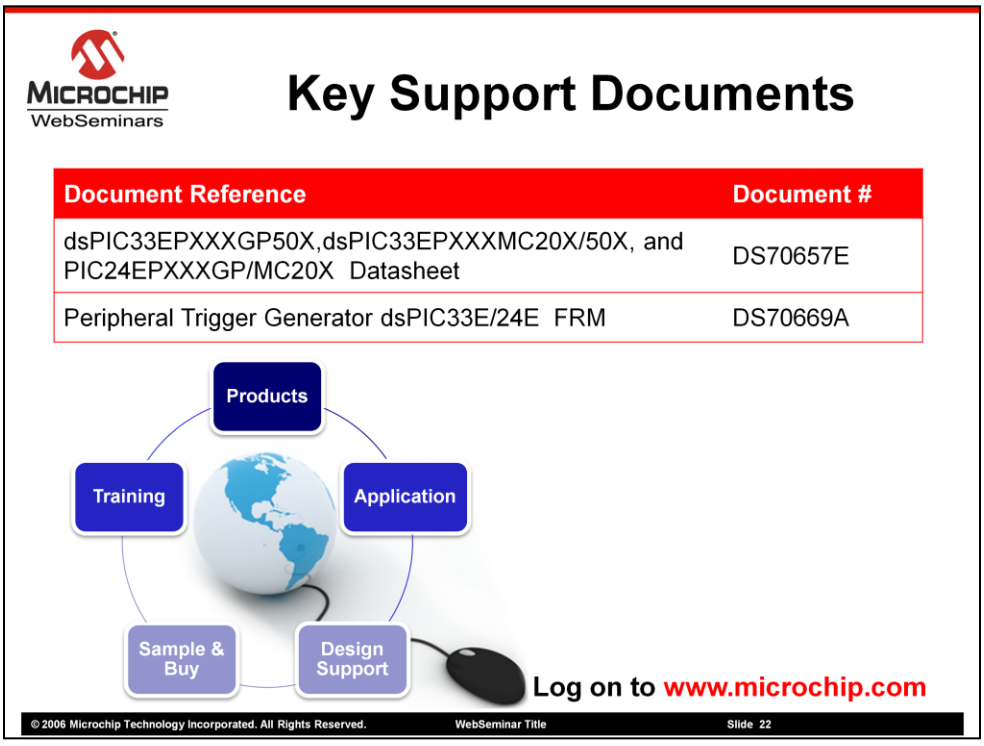
This simple application example demonstrates the capability of the PTG Module.



Now will summarize ,what we learned from this webinar

We had an overview of PTG Commands
We covered operation of peripheral trigger generator module
And we also looked at its features in detail

Finaly,looked at an application example using PTG module features



For more information, here are references to some important documents

The Family Reference Manual contains detailed information about the the peripheral

For device-specific information such as pin out diagrams, packaging and electrical characteristics, the device datasheets are the best source of information.

All these documents can be obtained from the Microchip web site www.microchip.com.Continue to check this website for new product releases and updates.

This wraps up the webinar on Peripheral trigger Generator on the dsPIC Digital Signal Controllers.

I thank you for your interest in Microchip Products.