AN2152

Applications of the Peripheral Trigger Generator (PTG)

Author: *Ravikiran Shetty*
Microchip Technology Inc.

INTRODUCTION

The Peripheral Trigger Generator (PTG) module, in 16-bit dsPIC33 devices, is a user-programmable sequencer, which is capable of generating triggers with complex input signal sequences to coordinate the operation of other peripherals. This document will explain various applications that use PTG in conjunction with modules, such as an Analog-to-Digital Converter (ADC), Output Compare (OC), Pulse-Width Modulator (PWM), timers and interrupt controllers to achieve a complex sequence of events.

The PTG module supports 8-bit commands, called the Step commands, to the PTG Queue registers. Each 8-bit Step command is comprised of a 4-bit command code and a 4-bit option field.

These commands define a sequence of events for generating output trigger signals to the peripherals. The Step commands can also be used to generate the interrupt requests to the CPU.

For more information on the PTG module and its registers, refer to **"Peripheral Trigger Generator (PTG)"** (DS70669) in the *"dsPIC33/PIC24 Family Reference Manual"*.

Example usage cases discussed in this document are as follows:

- Application 1: Integrated PFC and Motor Control
- Application 2: Lighting Control
- Application 3: One-Shot Pulse Generation
- Application 4: Variable Width One-Shot Pulse Generation
- Application 5: Variable Frequency Waveform Generation
- Application 6: Constant Frequency Waveform Generation
- Application 7: ADC Module Control

Key Features of the PTG

- Behavior is Step Command Driven:
  - Step commands are 8 bits wide
- Commands are Stored in a Step Queue:
  - Queue depth is parameterized (8-32 entries)
  - Programmable Step execution time (Step delay)
- Supports the Command Sequence Loop:
  - Can be nested one-level deep
  - Can be conditional or unconditional loop
  - Two 16-bit loop counters
- Supports 16 Hardware Input Triggers:
  - Sensitive to either positive or negative edges, or a high or low level
- One sOftware Input Trigger
- Generates up to 32 Unique Output Trigger Signals
- Generates Two Types of Trigger Outputs:
  - Individual
  - Broadcast
- Strobed Output Port for Literal Data Values:
  - 5-bit literal write (literal part of a command)
  - 16-bit literal write (literal held in the PTGL0 register)
- Generates up to 16 Unique Interrupt Signals
- Two 16-Bit General Purpose Timers
- Flexible Self-Contained Watchdog Timer (WDT) to Set an Upper Limit to Trigger Wait Time
- Single Step Command Capability in Debug mode
- Selectable Clock (system, PWM or ADC)
- Programmable Clock Divider

APPLICATION 1: INTEGRATED PFC AND MOTOR CONTROL

In the integrated Power Factor Correction (PFC) and motor control application, a single Digital Signal Controller (DSC) device controls a Permanent Magnet Synchronous Motor (PMSM) using a Field Oriented Control (FOC) scheme, as well as the PFC converter.

This application requires three PWM channels to control the motor functioning and an additional PWM to control the PFC operation. An OC module can be used to augment the number of PWM channels available to the application, even beyond the number of high-speed PWM channels available on the device.

The PWM peripheral, in conjunction with an OC peripheral, can be used to generate the necessary PWM signals for motor control and PFC operation.

However, in an application such as PFC, optimal execution time is very important, and to achieve this objective, we have to achieve the following tasks within an optimal execution time:

- Synchronizing motor control PWM and PFC PWM.
- Triggering ADC for conversion, and switching ADC channels used for motor control and PFC feedback signals.

These application requirements can be achieved effectively using the PTG module, which can perform the following operations:

- Synchronizes high-speed PWM module and OC module.
- Generates ADC module triggers by monitoring high-speed PWM module edges.
- Monitors the “ADC conversion done” interrupt and generates appropriate interrupts, executing the FOC and digital PFC control code.
- Reduces CPU intervention, making the peripheral handling in this application core-independent.

In this application, the switching frequency of motor control PWM and PFC PWM is selected such that, it is in integral multiples.

The ADC in the dsPIC® DSC has four-channel simultaneous sampling capability. Both FOC and PFC algorithms have their own sets of analog channels that are to be sampled simultaneously, because the phase relationship of these signals is important from the control perspective.

In this application, the feedback signals of motor control and PFC are selected such that, by alternating between the ADC channel selections, both motor control and PFC signals are sampled. The motor control and PFC signals can be connected to the Sample-and-Hold (S&H) circuits before triggering the ADC, based on the edges of the PWM, by configuring CH0123SA = 0 or CH123SA = 1.

As shown in Table 1, the channels are configured in such a way that, at the end of a four-channel sample and conversion sequence, the conversion results for either FOC (if CH123SA (AD1CHS123<0>) = 0 and CH0SA<4:0> (AD1CHS0<4:0>) = 13) or PFC (if CH123SA (AD1CHS123<0>) = 1 and CH0SA<4:0> (AD1CHS0<4:0>) = 10) are available in their corresponding ADC1 Buffer registers (ADC1BUF0 to ADC1BUF3).

After setting the channel selection bits to connect the PFC feedback signals to the Sample-and-Hold circuit of an ADC, for every PFC PWM cycle, a trigger has to be generated. Similarly, for every motor control PWM cycle, an ADC trigger has to be generated after setting the channel selection bits to connect the motor control feedback signals to the Sample-and-Hold circuit of the ADC.

Note:	The ADC module trigger selection cannot be changed when ADON = 1.

Hence, the PTG module is configured to generate an ADC trigger by monitoring the edges of the motor control PWM and PFC PWM pulses.

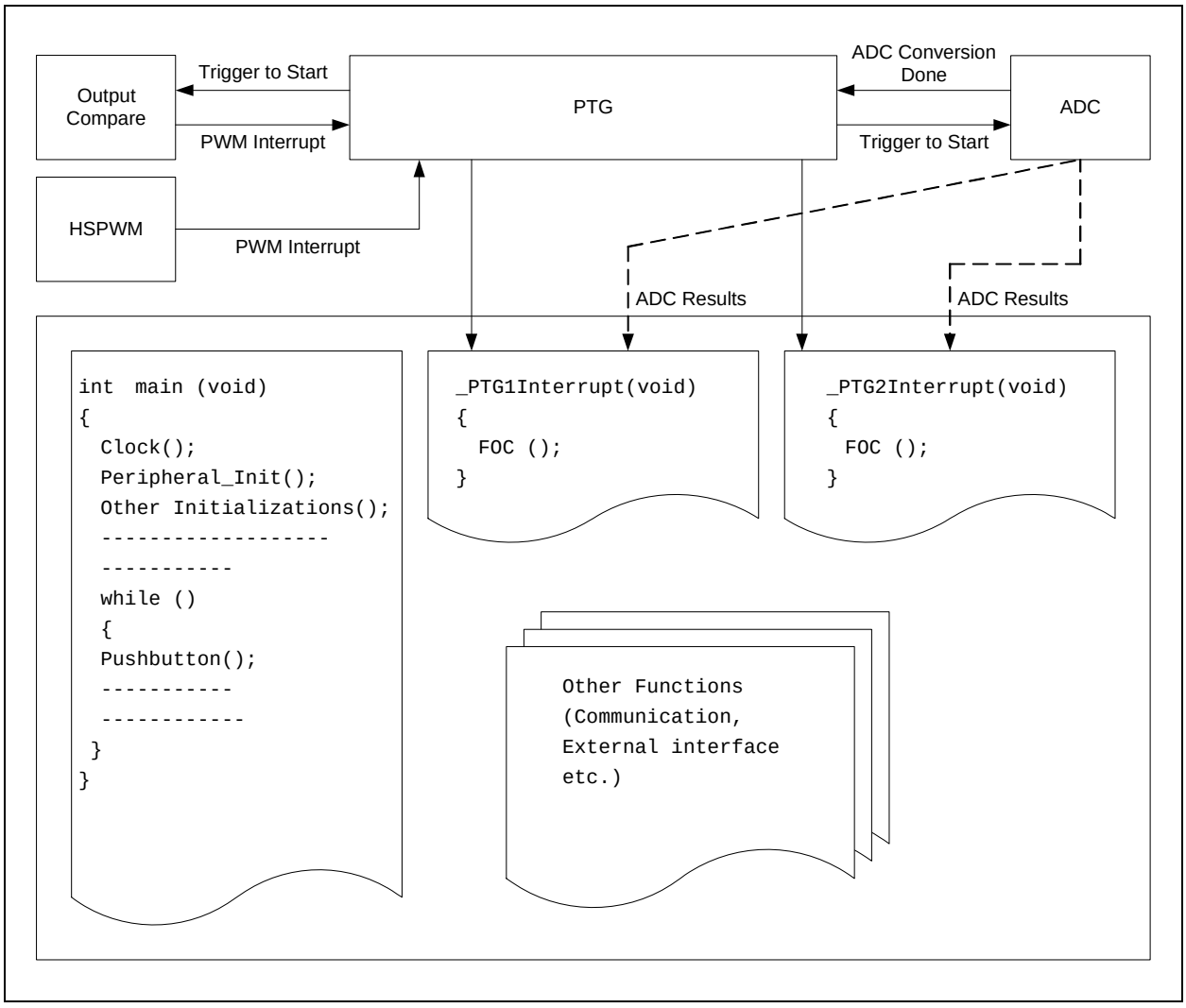
TABLE 1: ANALOG CHANNEL CONNECTIONS

Sample-and-Hold Circuit	Motor Control CH123SA = 0; CH0SA<4:0> = 13		PFC CH123SA = 1; CH0SA<4:0> = 10	
	Analog Feedback Signal	ADC Channel	Analog Feedback Signal	ADC Channel
CH0	Speed Reference	AN13	PFC Output Voltage	AN10
CH1	Phase Current 2	AN0	AC Input Voltage	AN3
CH2	Phase Current 1	AN1	NA	
CH3	Bus Current	AN2	AC Input Current	AN6

In addition, two PTG interrupts (PTG1 interrupt and PTG2 interrupt) are generated to execute the code for FOC and PFC, as shown in Figure 1.

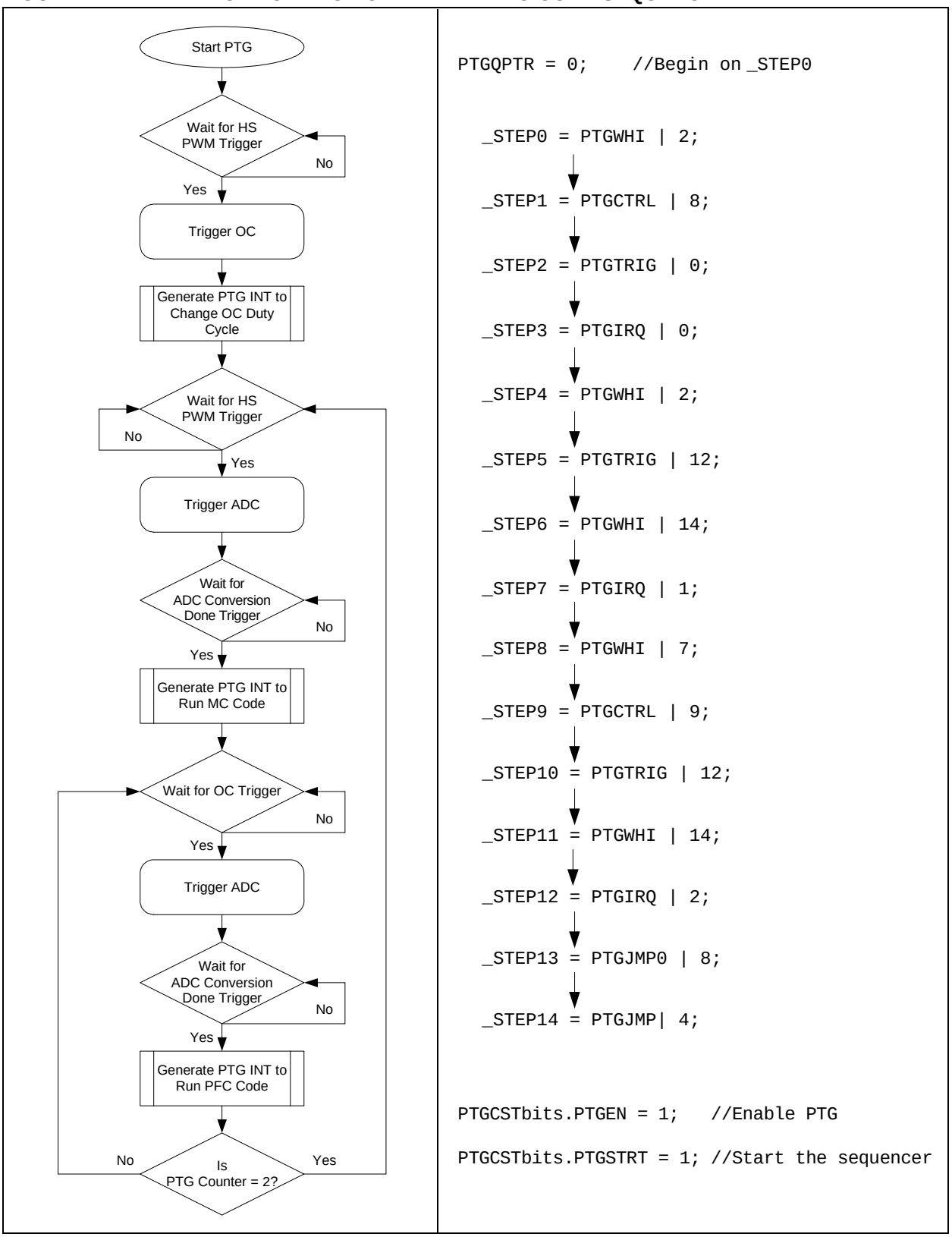
As seen in this example application, PTG simplifies the implementation by efficiently sequencing the use of the ADC and PWMs to achieve motor control and PFC implementation in one dsPIC33 device.

FIGURE 1: CODE EXECUTION USING PTG1/PTG2 INTERRUPTS BLOCK DIAGRAM



The flowchart of the application and PTG code sequence are illustrated in Figure 2.

FIGURE 2: APPLICATION FLOWCHART AND PTG CODE SEQUENCE



APPLICATION 2: LIGHTING CONTROL

In a light intensity control application, a PWM generator using an OC can be used to control the brightness of a light.

In this application, two OC modules are used and their duty cycles are controlled by inputs obtained from two separate ADC channels. Depending on each ADC value, the duty cycle is updated.

The PTG module supports a simpler way of synchronizing ADC and OC modules. In addition, the PTG helps to avoid a module deadlock to enhance the application safety.

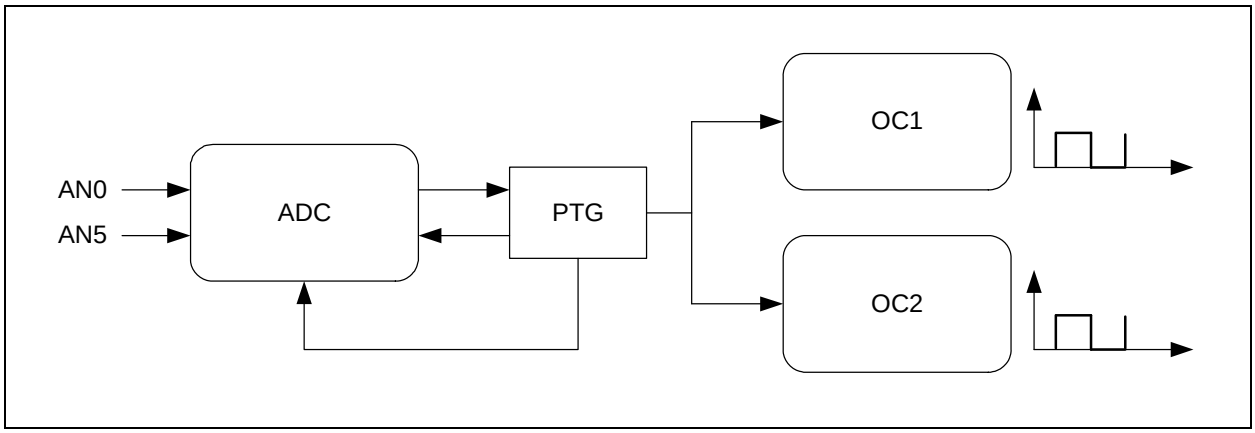
The following steps are performed for synchronization:

- Monitor the ADC and generate appropriate interrupts to change the OC duty cycle.
- Change the ADC channel without disturbing the CPU, as the PTG can do this independently.

As an additional safety feature, in the event of an unexpected failure, the PTG module has a dedicated Watchdog Timer (WDT) to monitor and perform the necessary required corrective actions.

A block diagram of this application is shown in Figure 3.

FIGURE 3: OUTPUT COMPARE DUTY CYCLE CONTROL USING PTG



Key Details of Implementation

ADC CHANNEL SELECTION

The Strobe output of the PTG module can be used to write to the AD1CHS0 register. The PTGCTRL command writes the PTGL0 register contents to the AD1CHS0 register. The PTGL0 register can be modified by using the PTGADD and PTGCOPY commands.

The ADC channels used to measure the input voltage will be selected using the PTGL0 register and modified by the PTGADD and PTGCOPY commands, so that the CPU will not be interrupted.

SYNCHRONIZATION OF ADC AND OC MODULES

On some devices, since the OC module cannot trigger the ADC and the ADC module trigger selection cannot be changed when ADON = 1, the PTG controls the ADC sampling and conversion processes. The ADC can trigger the PTG after the conversion is done, so that the PTG selects the next ADC channel for sampling and conversion. Hence, it synchronizes the sampling/conversion process with optimal execution time.

The OC duty cycle can be varied by updating the OCxR and OCxRS registers. Depending on the ADC value, the duty cycle is varied.

The PTG module can generate an individual interrupt pulse using the PTGIRQ command. The PTG module can generate up to four unique interrupt request signals. These signals are useful for generating different interrupts to change the duty cycles of different OC modules, so that executing complex functions within one Interrupt Service Routine (ISR) can be avoided.

PTG WATCHDOG TIMER AS A SAFETY FEATURE

A Watchdog Timer (WDT), within the PTG module, will prevent a situation where the PTG waits indefinitely for an external event when executing a command which waits for a hardware trigger high/low state.

In this application, the PTG will wait for an “ADC conversion done” trigger. Once enabled, the WDT starts counting when the command execution starts. It is disabled when the command completes execution. If an expected event fails to arrive before the WDT time-out period expires, the PTG module:

a) Aborts the (failing) command underway.
b) Halts the sequencer PTGSTRT (PTGCST<7>) = 0).
c) Sets PTGWDTO (PTGCST<6>) = 1.
d) Issues a Watchdog Timer error interrupt to the CPU.

This acts as a safety feature to recover from a situation where the ADC or PTG module stops working. These modules can be re-initialized and restarted within the Watchdog Timer error interrupt.

Advantages of Using PTG

a) The PTG will make the application core-independent by switching the ADC channels and monitoring modules without the CPU module's intervention.

   Also, saves CPU module's time which can be utilized for other tasks in the application.

   Figure 4 and Figure 5 depict the time consumption, with and without PTG, and illustrates the time available for the CPU to perform other operations.

b) The PTG alone will take care of all interactions within a module, which helps to reduce software complexity and also helps to maintain modularity.

c) The PTG module's WDT will help to recover from any catastrophic failure, thereby providing a more robust application.

d) If required, after the OC1 module triggers the PTG, the PTG, in turn, can trigger the OC2 module to generate a phase-shifted output.

FIGURE 4: TIME CONSUMPTION WITHOUT PTG

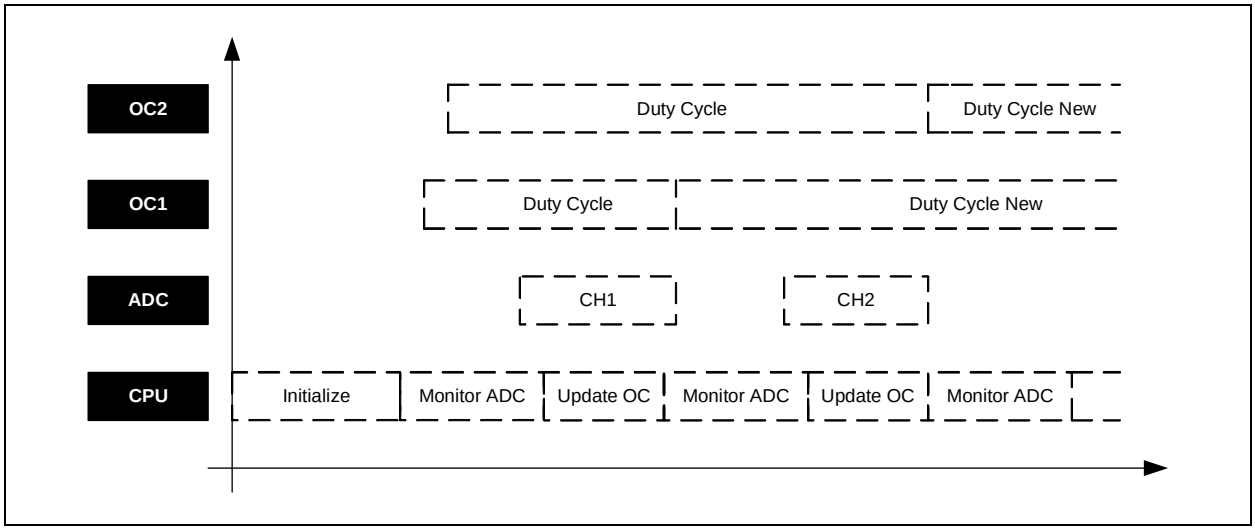
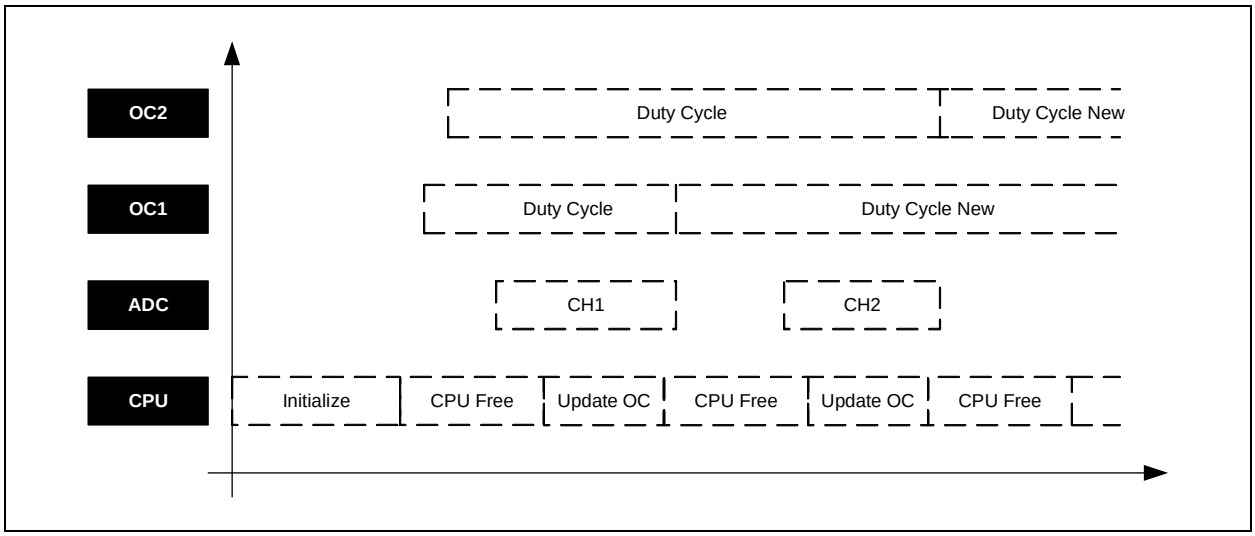
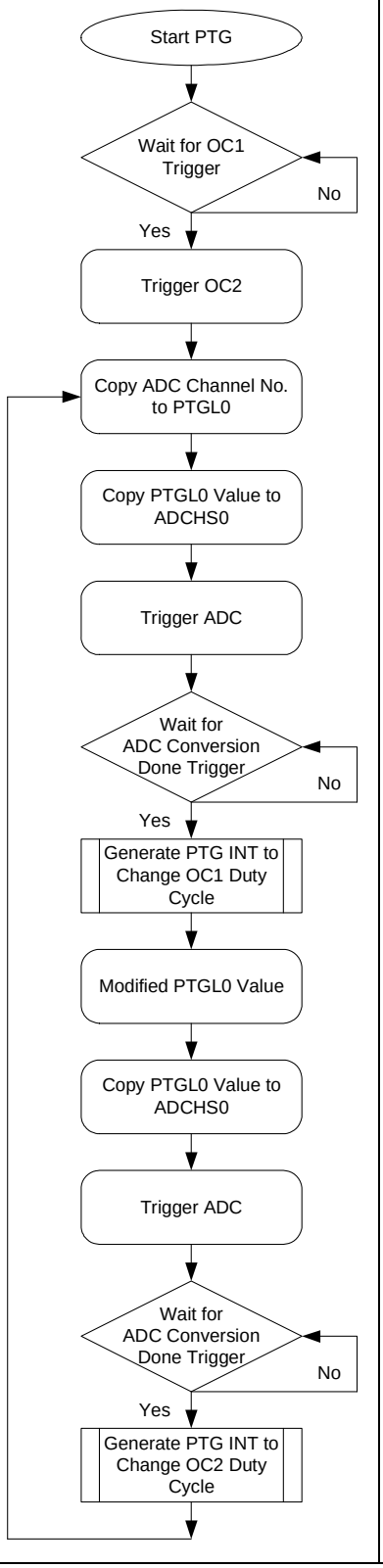


FIGURE 5: TIME CONSUMPTION WITH PTG



The flowchart of the application and the PTG code sequence are illustrated in Figure 6.

FIGURE 6: APPLICATION FLOWCHART AND PTG CODE SEQUENCE



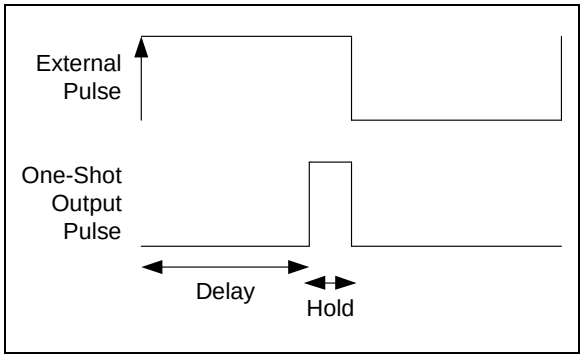

```
PTGQPTR = 0;    //Begin on _STEP0

PTGADJ = 0x2;

PTGHOLD = 0x3;

PTGL0 = 0x0;


_STEP0 = PTGWHI | 7;   //Wait for input trigger from OC1
         |
         V
_STEP1 = PTGTRIG | 2;  //Trigger OC2
         |
         V
_STEP2 = PTGCOPY | 13; //Copy ADC Channel number to PTGL0
         |
         V
_STEP3 = PTGCTRL | 14; //Change ADC Channel number
         |
         V
_STEP4 = PTGTRIG| 12;  //Trigger ADC
         |
         V
_STEP5 = PTGWHI | 14;  //Wait for ADC Conversion Done Trigger
         |
         V
_STEP6 = PTGIRQ| 1;    //Generate PTG interrupt, change OC1 duty cycle
         |
         V
_STEP7 = PTGADD | 5;   //ADD ADC Channel number to PTGL0
         |
         V
_STEP8 = PTGCTRL | 14; //Change ADC Channel number
         |
         V
_STEP9 = PTGTRIG| 12;  //Trigger ADC
         |
         V
_STEP10 = PTGWHI | 14; //Wait for ADC Conversion Done Trigger
          |
          V
_STEP11 = PTGIRQ| 2;   //Generate PTG interrupt, change OC2 duty
          |
          V
_STEP12 = PTGJMP| 2;   //Repeat the process


PTGCSTbits.PTGEN = 1;      //Enable PTG

PTGCSTbits.PTGSTRT = 1;    //Start the sequencer
```

APPLICATION 3: ONE-SHOT PULSE GENERATION

This section illustrates an application scenario where a one-shot pulse needs to be triggered by either a rising or falling edge of an external pulse. The width of the one-shot pulse can be changed dynamically by changing the delay and Wait time in the program loop. Another requirement would be that neither the external pulse nor the one-shot pulse should interrupt the CPU.

This application scenario is shown in Figure 7.

FIGURE 7: APPLICATION SCENARIO



The external pulse can be detected using an external interrupt feature or Change Notification (CN), but both methods would interrupt the CPU code execution.

Alternatively, a one-shot pulse can be generated, using an OC module, and an external pulse can synchronize the OC module. However, we also need to induce a delay before the one-shot pulse, which again interrupts the CPU code execution.

However, the PTG module can be used to accomplish this requirement efficiently without interrupting the CPU.

The PTG performs the following tasks:

- Synchronizing the external interrupt and OC modules.
- PTG timer can be used to induce delay after external pulse has been detected.
- PTG WDT will prevent PTG from waiting indefinitely for an external event, thereby avoiding deadlock conditions.

Key Details of Implementation

EXTERNAL INTERRUPT

The external interrupt can be configured for a positive or negative edge using the INTxEP bit and an external interrupt will act as a triggering source for the PTG module.

OUTPUT COMPARE

The OC module is configured for the Triggered Dual Compare Single-Shot mode. By setting the TRIGMODE bit, the OC module can be retriggered.

ROLE OF PTG

- PTG module will wait for a trigger from an external interrupt.
- When PTG gets triggered, it will wait until the PTG timer reaches its limit, adding the required delay before triggering the OC module.
- Trigger OC module, which generates a one-shot pulse with a configured pulse width.

There are two 16-bit General Purpose (GP) timers (PTGT0 and PTGT1), which can be used by the sequencer to wait for a specified period. All timers are cleared when the device is in the Reset state or when the PTG module is disabled (PTGEN = `0`). Step commands are available for loading, modifying or initializing the GP timers.

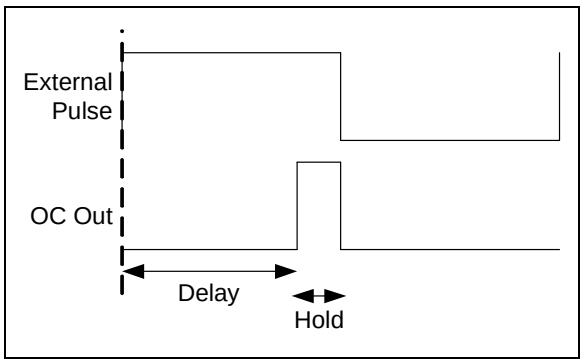
When waiting for a GP timer, the command will wait until the value of the timer (Timer0 or Timer1) reaches its limit value (PTGT0LIM or PTGT1LIM). On reaching the limit value, the Step command execution completes and the next command will start. The timer is also cleared for its next use.

Hence, the CPU will not be interrupted at any point. The delay-and-hold time can be changed within `main()`.

In the event of an unexpected failure, the WDT of the PTG module monitors and performs necessary corrective actions as a safety feature.

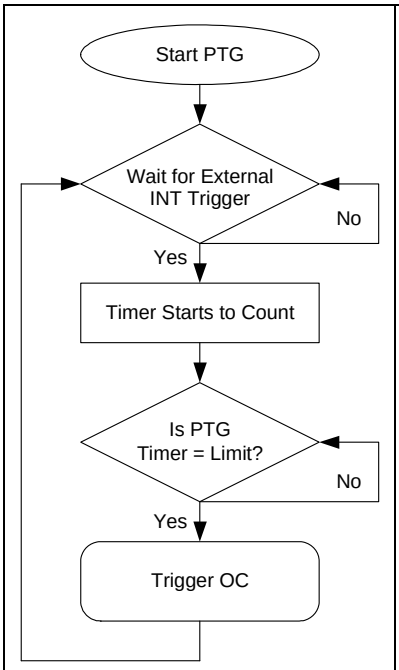
The application output is shown in Figure 8.

FIGURE 8: GENERATING SINGLE OC PULSE AFTER AN EXTERNAL PULSE



The flowchart of the application is illustrated in Figure 9.

FIGURE 9: APPLICATION FLOWCHART AND PTG CODE SEQUENCE

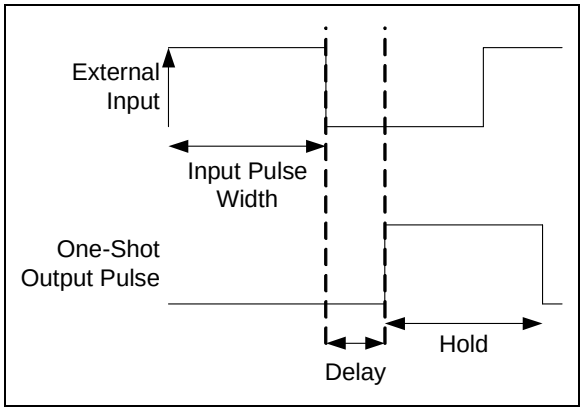



```
PTGQPTR = 0;       //Begin on _STEP0

PTGT0LIM = 0x2000; //Initialize delay required to the PTG timer



_STEP0 = PTGWHI | 15; //Wait for an external signal

_STEP1 = PTGCTRL | 8; //Starts PTGT0 timer and waits for time out

_STEP2 = PTGTRIG | 0; //Trigger OC module

_STEP3 = PTGJMP | 0;  //Jump to _STEP0 to wait for the next external signal



PTGCSTbits.PTGEN = 1;   //Enable PTG

PTGCSTbits.PTGSTRT = 1; //Start the sequencer
```

APPLICATION 4: VARIABLE WIDTH ONE-SHOT PULSE GENERATION

This section illustrates an application scenario, where a one-shot pulse needs to be triggered by the rising edge of an external pulse. The width of the one-shot pulse can be changed dynamically by changing the delay and wait time in the program loop; but the pulse width of an external pulse will be captured to decide the width (hold) of the one-shot pulse, as shown in Figure 10.

FIGURE 10: APPLICATION SCENARIO



Various peripheral features of the dsPIC® DSC can be used to effectively accomplish this task, a few of which are as follows:

- The external pulse can be detected using the external interrupt feature.
- The one-shot pulse can be generated using the OC module.
- Pulse width can be measured by using the IC module.

However, in this application, synchronizing the OC, IC and external interrupt modules, and capturing the pulse width of the input pulse using the IC module, are challenging tasks if performed only by the CPU.

The PTG module accomplishes this requirement efficiently with minimal CPU interruption by executing the following tasks:

- Synchronizing the external interrupt, OC modules and IC module.
- As soon as the external pulse is detected, the IC module will be triggered to measure the pulse width of the rising pulse only.
- The WDT of the PTG prevents the PTG from waiting indefinitely for an external event, thereby avoiding deadlock conditions.

Key Details of Implementation

EXTERNAL INTERRUPT

The external interrupt can be configured to interrupt on the positive edge using the INTxEP bit. This external interrupt will act as a triggering source for the PTG module.

INPUT CAPTURE (IC)

The IC module is configured for capture on every falling edge. The PTG is configured as its trigger/sync source.

OUTPUT COMPARE (OC)

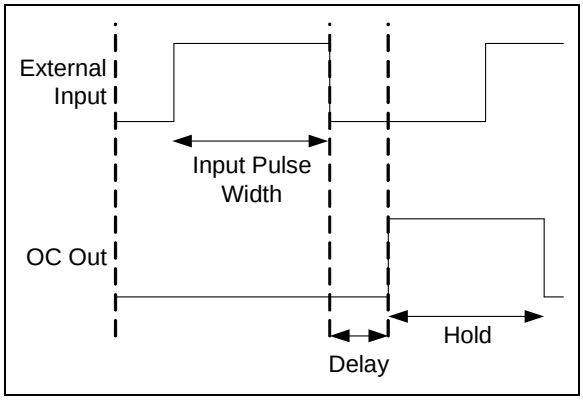
The OC module is configured for Dual Compare Single-Shot mode with the PTG configured as its trigger source.

ROLE OF PTG

- The PTG module will wait for a trigger from an external interrupt.
- When triggered, the PTG will trigger the IC module, which starts the IC timer.
- The PTG will wait for a trigger from the IC module.
- When the PTG gets triggered by the IC module, the IC buffer will capture the pulse width of the incoming external pulse.
- The PTG can generate unique interrupts, which will be used to change the OC duty cycle.
- A delay can also be added using the PTG timers.
- The PTG will trigger the OC module, which generates a one-shot pulse with a configured pulse width.
- In the event of an unexpected failure, the WDT of the PTG module monitors and resets the sequence.

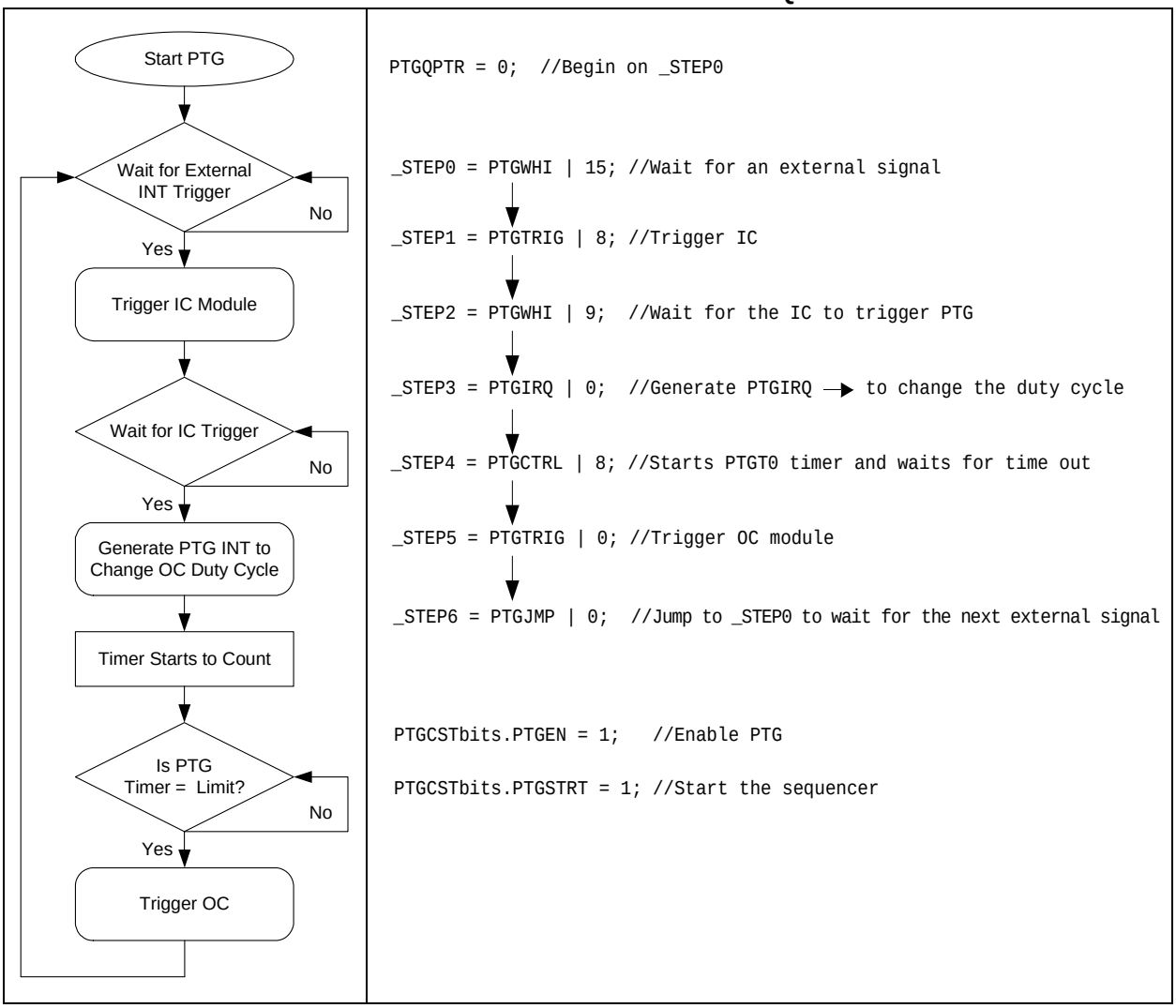
The application output waveform is shown in Figure 11.

FIGURE 11: GENERATING SINGLE OC PULSE AFTER AN EXTERNAL PULSE



The flowchart of the application is illustrated in Figure 12.

FIGURE 12: APPLICATION FLOWCHART AND PTG CODE SEQUENCE



APPLICATION 5: VARIABLE FREQUENCY WAVEFORM GENERATION

This application illustrates how the PTG can be used to generate a variable frequency signal generator.

The OC modules, along with DMA, can be used to generate variable frequency signals.

However, the PTG can provide an additional mechanism to generate a variable frequency signal, as described below.

The solution would use the following PTG features:

- The PTG has its own timer, which runs independently of the CPU.
- The PTG has ADD and COPY commands, which can write to the Timer Limit register.
- The PTG can generate its own interrupt.

The control techniques used in the application involve the following steps:

- Configuring the PTG with the timer limit value, as well as the adjust value, which will be added to the Timer Limit register.
- Generating the PTG ISR, where a general purpose I/O pin is toggled.
- Waiting for the timer to reach its limit value.
  - On reaching the limit value, the Step command execution completes, and in the next command, adds the PTGADJ value to the Timer Limit register.
  - The CPU will be interrupted only while toggling the I/O pin.

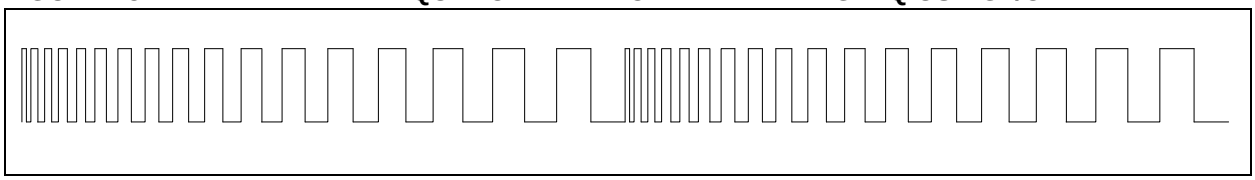
There are two 16-bit General Purpose timers (PTGT0 and PTGT1), which can be used by the sequencer to wait for a specified period. All timers are cleared when the device is in the Reset state or when the PTG module is disabled (PTGEN = 0). Step commands are available for loading, modifying or initializing the GP timers.

When waiting for the GP timer, the command will wait until the value of the timer (Timer0 or Timer1) reaches its limit value (PTGT0LIM or PTGT1LIM). On reaching the limit value, the Step command execution completes and the execution of the next command will start. The timer is also cleared for its subsequent use.

The timer limit value can be modified using the PTGADD or PTGCOPY commands. The PTGADD command will add the PTGADJ register contents to the PTGCxLIM register. The PTGCOPY command will copy the PTGHOLD register contents to the PTGCxLIM register.

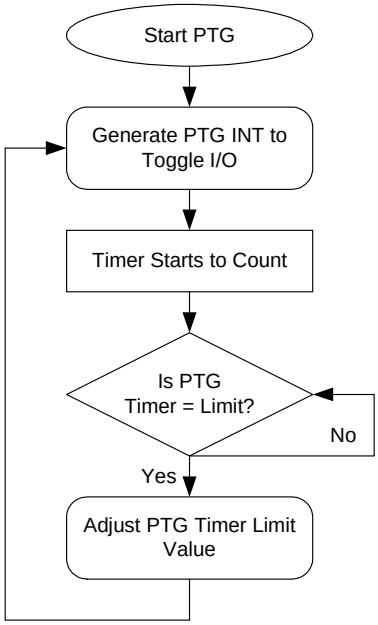
The application output is as shown in Figure 13.

FIGURE 13: VARIABLE FREQUENCY WAVEFORM WITHIN PTG IRQ USING I/O PIN



The flowchart of the application and the PTG code sequence are illustrated in Figure 14.

FIGURE 14: APPLICATION FLOWCHART AND PTG CODE SEQUENCE




```
PTGQPTR = 0;  //Begin on _STEP0

PTGADJ = 0x800;

PTGC0LIM = 0x800;


_STEP0 = PTGIRQ | 0x0;  //Generate PTG Interrupt 0 → Toggles GP I/O
          ↓
_STEP1 = PTGCTRL | 8;  //T0 = 0; While (T0++ ≠ PTGT0LIM)
          ↓
_STEP2 = PTGADD | 2;   //PTGT0LIM ≥ PTGADJ
          ↓
_STEP3 = PTGJMP | 0x0; //Jump to _STEP0


PTGCSTbits.PTGEN = 1;   //Enable PTG

PTGCSTbits.PTGSTRT = 1; //Start the sequencer
```

APPLICATION 6: CONSTANT FREQUENCY WAVEFORM GENERATION

This application illustrates how the PTG module can be used to generate a constant frequency signal, which can also act as a clock source.

The solution would use the following features of the PTG module's features:

- PTG can trigger a comparator, which acts as a mask input select
- The PTG module's trigger pulse width can be varied
- PTG has its own timer

The control techniques used in the application involve the following steps:

a) The PTG module trigger can work as a mask input select for the op amp/comparator, as shown in Figure 15.
   - Using this feature, a PTG output can be brought out through a comparator module.

b) The Comparator module is configured, as shown in Figure 16, where the inverting input is connected to ground and the non-inverting input is connected to an internal reference voltage.

c) The PTG trigger pulse will directly emerge as the comparator output. As long as the PTG generates triggers continuously, the comparator will generate a constant frequency waveform. The pulse width of the waveform will be one cycle of the PTG clock.

d) The on/off times can be controlled by the PTG timer and PTG Pulse-Width bits (PTGPWD<3:0>).
   - The PTG output pulse width (PTGPWD<3:0>) will decide the off time of the output waveform.
   - The PTG timer will decide the on time of the output waveform, which is the delay between the triggering comparator modules.
   - Depending on the comparator output polarity, the on/off time will be controlled by either the timer or PTGPWDx bits.

e) The output frequency can also be controlled by the PTGDIV register, which will act as a clock divider.

FIGURE 15: USER-PROGRAMMABLE MASKING FUNCTION

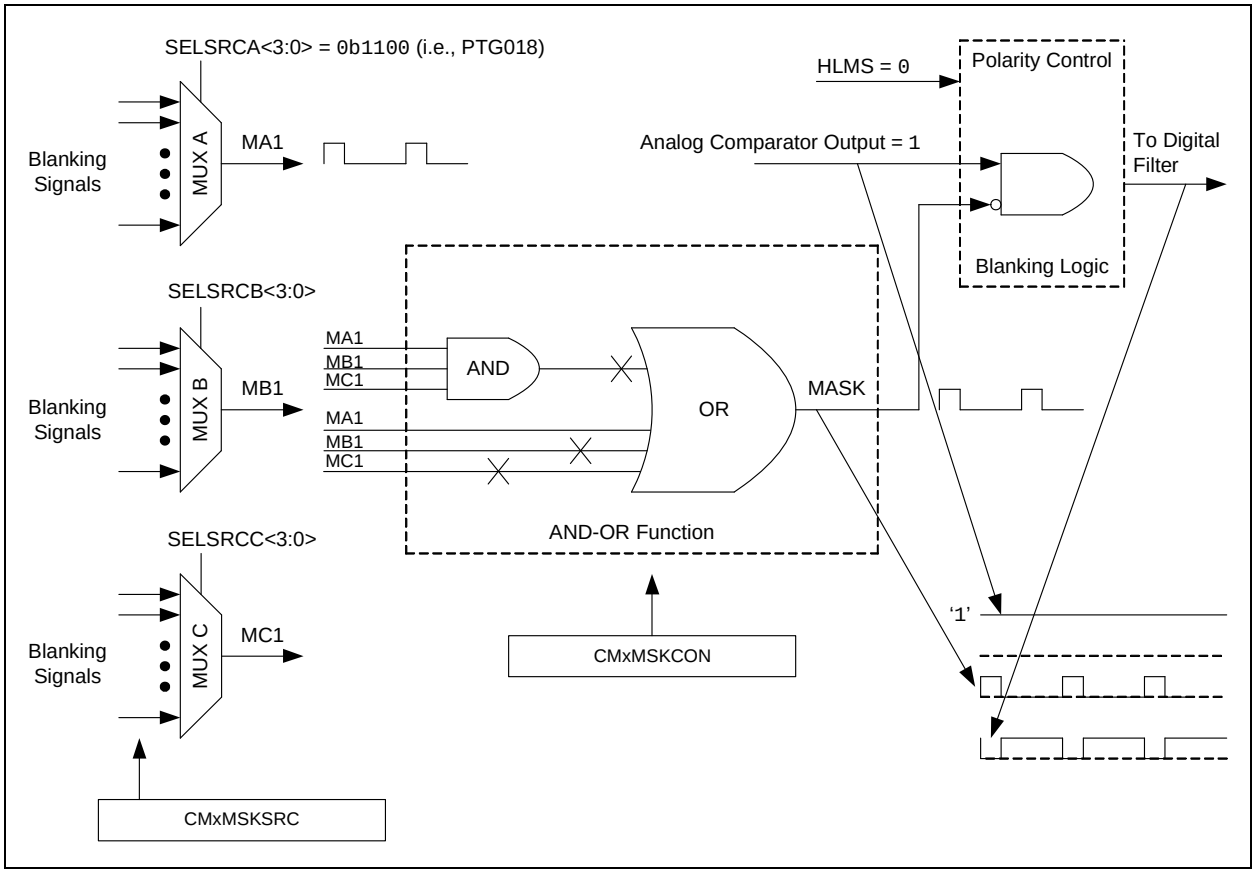
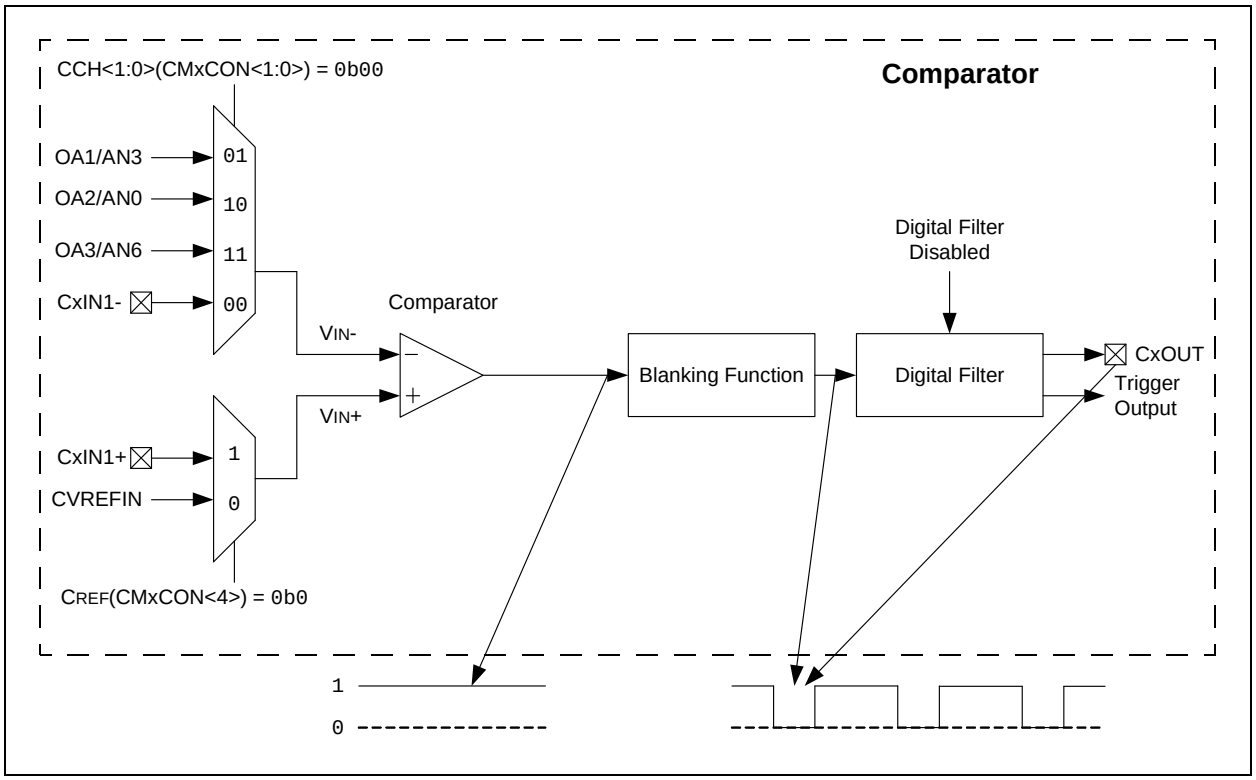
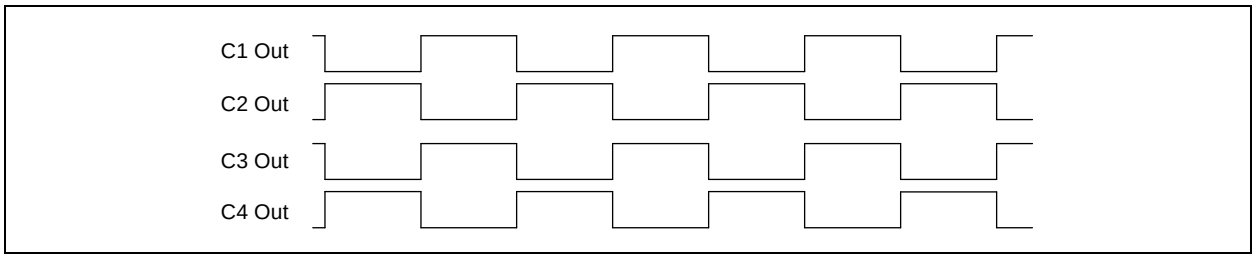


FIGURE 16: COMPARATOR CONFIGURATION



By changing the comparator output polarity, a complementary waveform can be generated using four comparator modules, as shown on Figure 17.

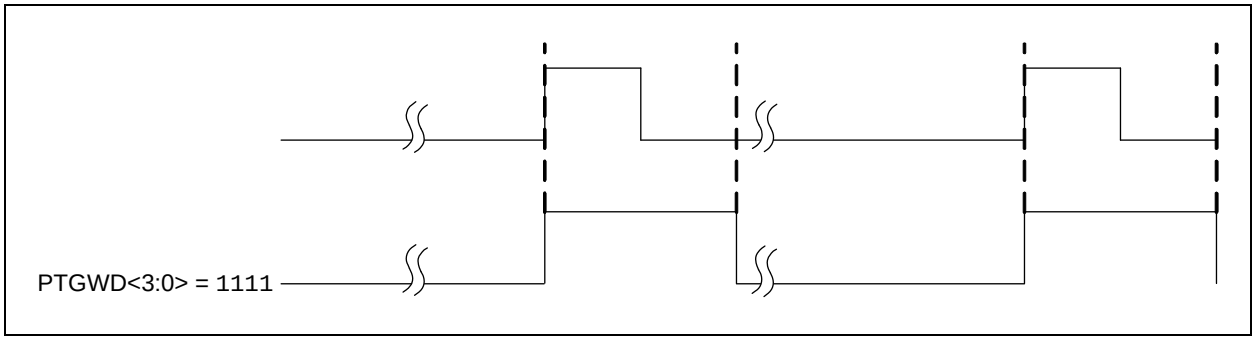
FIGURE 17: PTG OUTPUT THROUGH COMPARATOR



The width of the pulse can be modified using the PTGPWD<3:0> bits, which reduce the frequency of the output, as shown in Figure 18.

Hence, a constant waveform can be generated using the PTG and comparator modules.

FIGURE 18: PTG TRIGGER PULSE WIDTH

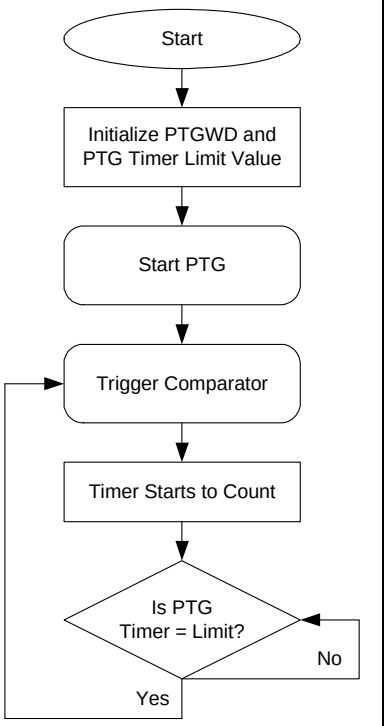


The advantages of using the PTG in this application are:

- The output can act as a constant clock source and run completely core-independent.
- Using more comparator modules, an even complementary waveform can be generated.
- The PTG works in power-saving modes, such as Idle and Sleep.

The flowchart of the application and the PTG code sequence are illustrated in Figure 19.

FIGURE 19: APPLICATION FLOWCHART AND PTG CODE SEQUENCE




```
PTGQPTR = 0;                  //Begin on _STEP0

PTGCONbits.PTGCLK = 0;        //PTG module clock source will be FP

PTGCONbits.PTGDIV = 9;        //PTG module clock prescaler

PTGCONbits.PTGPWD = 1;        //PTG Trigger output pulse width

PTGT0LIM = (_PTGPWD * 2) - 1; //Timer limit value



_STEP0 = PTGTRIG | 18; //Generate PTG trigger mask input
                         Select for Op Amp/Comparator (OFF time)
_STEP1 = PTGCTRL | 8;  //Wait for timer to reach limit value (ON time)

_STEP2 = PTGJMP | 0;   //Jump to _STEP0



PTGCSTbits.PTGEN = 1;   //Enable PTG

PTGCSTbits.PTGSTRT = 1; //Start the sequencer
```

APPLICATION 7: ADC MODULE CONTROL

The PTG module has many options to control the ADC, which is very helpful for synchronizing the ADC and other modules, as the PTG enables a large number of peripherals to trigger an ADC conversion. The PTG also enables the ADC trigger sources to be changed dynamically.

The PTG makes the process of controlling the ADC module core-independent, as the ADC can directly trigger the PTG after the conversion is done. If it were not for this feature, the ADC DONE bit would have been polled by the CPU or the ADC interrupt would have indicated the end of conversion.

The PTG can also change the ADC channel with its Strobe command. The strobe output of the PTG module can be used to write to the AD1CHS0 register (refer to Table 2).

TABLE 2: ADC CHANNEL MODIFICATION OPTIONS

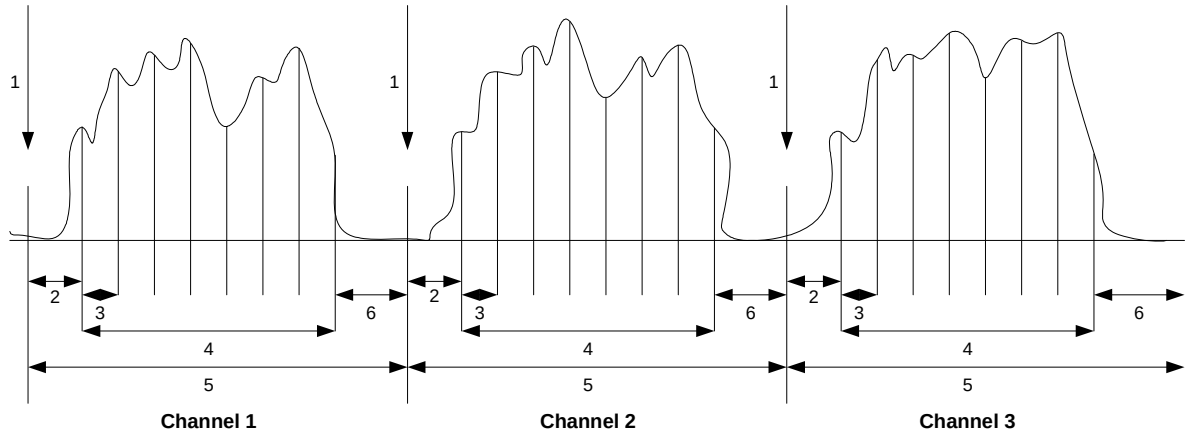
Command	Options	Description
PTGCTRL	0x1100	Writes the PTGL0 register contents to the AD1CHS0 register
PTGADD and PTGCOPY	—	Modifies the PTGL0 register
PTGCTRL	0x1100	Writes the PTGC0 register contents to the AD1CHS0 register
PTGCTRL	0x1101	Writes the PTGC1 register contents to the AD1CHS0 register

These features are also used in other applications, for example:

- In the **"Application 1: Integrated PFC and Motor Control"** section, the PWM and PFC channels are changed within the PTG ISR.
- In the **"Application 2: Lighting Control"** section, the PTGL0 with the PTGADD and PTGH0LD commands were used to switch the channel to be scanned without the CPU module's intervention.

The PTG provides the user with the flexibility to select any channel and specify the number of times each channel needs to be sampled; this is shown in Figure 20.

FIGURE 20: PTG AND ADC



Where:

1 = Trigger Input to PTG

2 = Delay before PTG Triggering ADC

3 = Delay between Each Sample and Conversion (PTG Triggering ADC)

4 = Repeat Step 3 Trigger Output (repeat 25x)

5 = Cycle Time for Each Channel

6 = Take Average of ADC Output within PTG ISR

Using this flexibility, the PTG can be used to automate the ADC sampling process using following steps:

a) The PTG is configured for an appropriate timer limit value to give enough time between triggers.

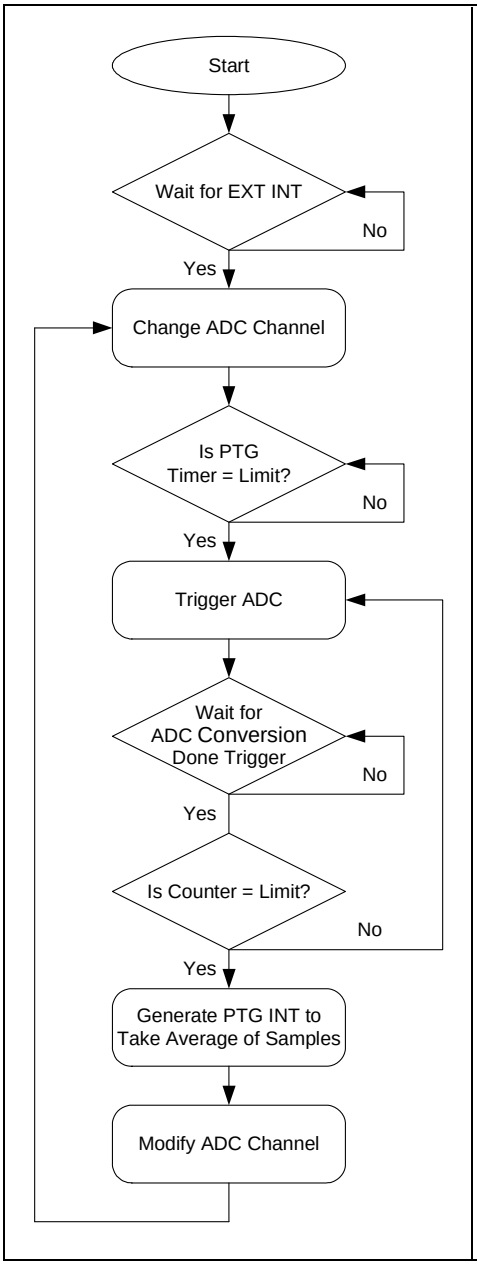
b) The ADC is configured with the PTG as a trigger source.

c) The PTG counter will be useful in repeating as many times as the ADC needs to be triggered.

d) The PTG ISR will take the average of the ADC outputs and also change the ADC channels by using the Strobe command.

The flowchart of the application and PTG code sequence are illustrated in Figure 21.

FIGURE 21: APPLICATION FLOWCHART AND PTG CODE SEQUENCE




```
PTGQPTR = 0;  //Begin on _STEP0


_STEP0 = PTGWHI|15;  //Waiting for input trigger from EXT INT
_STEP1 = PTGSTRB|14; //Write channel to ADCHS0
_STEP2 = PTGCTRL|8;  //Wait for timer to reach limit
_STEP3 = PTGTRIG|12; //Trigger ADC
_STEP4 = PTGWHI|14;  //Wait for ADC Conversion Done trigger
_STEP5 = PTGJMPC0|3; //Retrigger ADC
 _STEP6 = PTGIRQ|1;   //Generate PTG interrupt; Average of ADC output
 _STEP7 = PTGADD|5;  //Add ADC channel number to PTGL0, New channel
_STEP8 = PTGJMP|1;   //Change ADC channel number


PTGCSTbits.PTGEN = 1;   //Enable PTG

PTGCSTbits.PTGSTRT = 1; //Start the sequencer
```

SUMMARY

The PTG module in Microchip's DSC devices allows users to design complex application sequences with flexibility. A PTG allows various modules to interact with each other with minimum or no CPU interruption and also enhances the capabilities of the existing peripherals, thus expanding the horizon of applications a peripheral can accomplish.

With the PTG module, the applications can have faster response time and incur minimal software burden. The PTG also provides additional safety to the applications with built-in functions.

There are a wide range of applications that can be implemented using PTG, of which only a few are discussed in this document. Microchip encourages users to explore other possibilities of using PTG.

REFERENCES

1. **Peripheral Trigger Generator (PTG)** in the *"dsPIC33/PIC24 Family Reference Manual"*

   http://ww1.microchip.com/downloads/en/DeviceDoc/DS-70669A.pdf

2. Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW),

   http://ww1.microchip.com/downloads/en/AppNotes/01292A.pdf

3. **Analog-to-Digital Converter (ADC)** in the *"dsPIC33/PIC24 Family Reference Manual"*

   http://ww1.microchip.com/downloads/en/DeviceDoc/70621c.pdf

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Trademarks

ISBN: 978-1-5224-0651-8

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

07/14/15