Lesson 6
Counters

TOPICS

A. Counter Overview
B. Edge Counting
C. Pulse Generation
D. Pulse Measurement
E. Frequency Measurement
F. Position Measurement

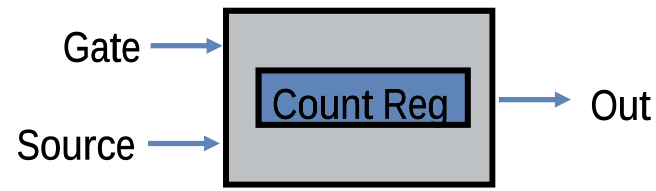
A. What is a Counter?

Two basic functions

- To count based on the comparison of input signals (gate, source)
- To generate pulses based upon inputs and register values

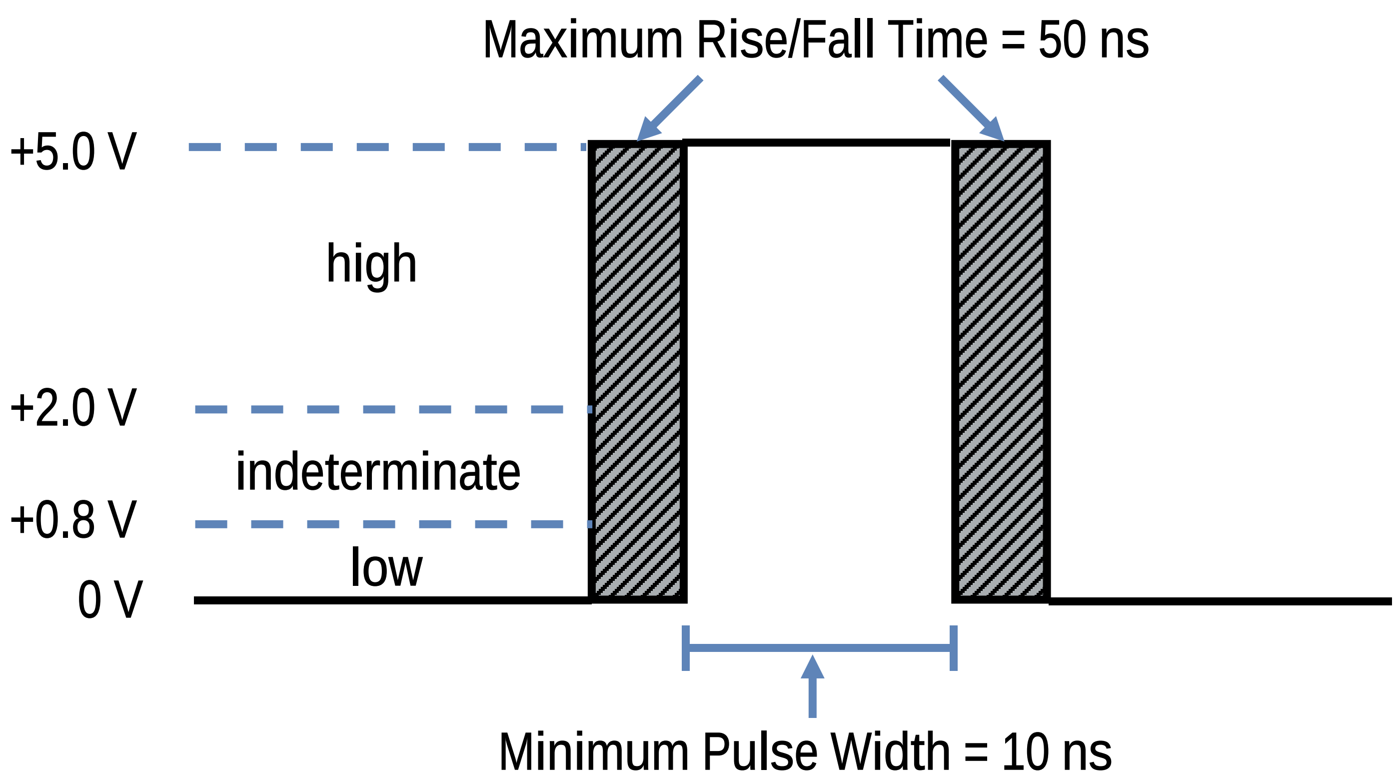
Many applications are derived from basic counting

- Edge counting, such as simple edge counting and time measurement
- Pulse, semi-period, and period width measurement
- Frequency measurement
- Single pulse and pulse train generation
- Position and velocity measurement

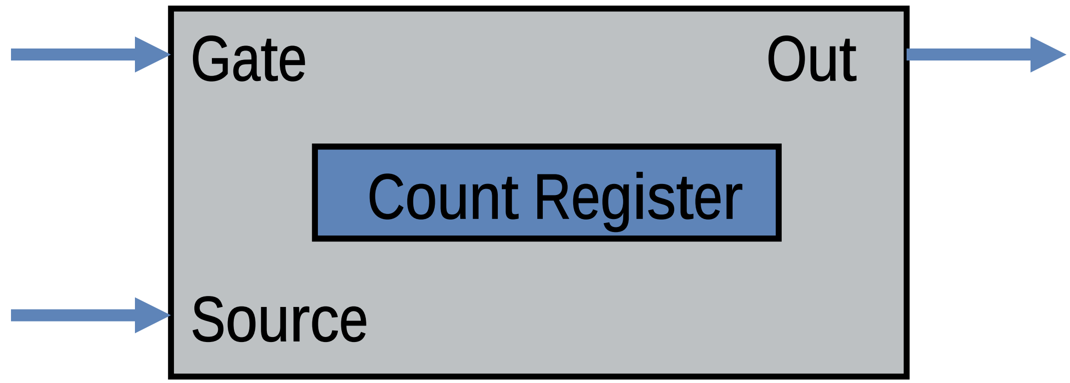


Counter Signals

Counters accept and generate TTL signals



Parts of a Counter



- Count Register
  - Stores the current count
- Source
  - Input signal that changes the current count
  - Active edge (rising or falling) of input signal changes the count
  - Choose if count increments or decrements on an active edge
- Gate
  - Input signal that controls when counting occurs
  - Counting can occur when gate is high, low, or between various combinations of rising and falling edges
- Out
  - Output signal used to generate pulses

Counter Pins

Counter gate and source are PFI pins

- PFI stands for Programmable Function Input
- Allows use of one pin for multiple applications

Example: Use pin 3 as digital trigger for analog input and counter gate

PFI0/TRIG1	11	45	EXTSTROBE*
PFI1/TRIG2	10	44	DGND
DGND	9	43	PFI2/CONVERT*
+5 V	8	42	PFI3/GPCTR1_SOURCE
DGND	7	41	PFI4/GPCTR1_GATE
PFI5/UPDATE*	6	40	GPCTR1_OUT
PFI6/WFTRIG	5	39	DGND
DGND	4	38	PFI7/STARTSCAN
PFI9/GPCTR0_GATE	3	37	PFI8/GPCTR0_SOURCE
GPCTR0_OUT	2	36	DGND

Source and Gate Selectors in DAQmx

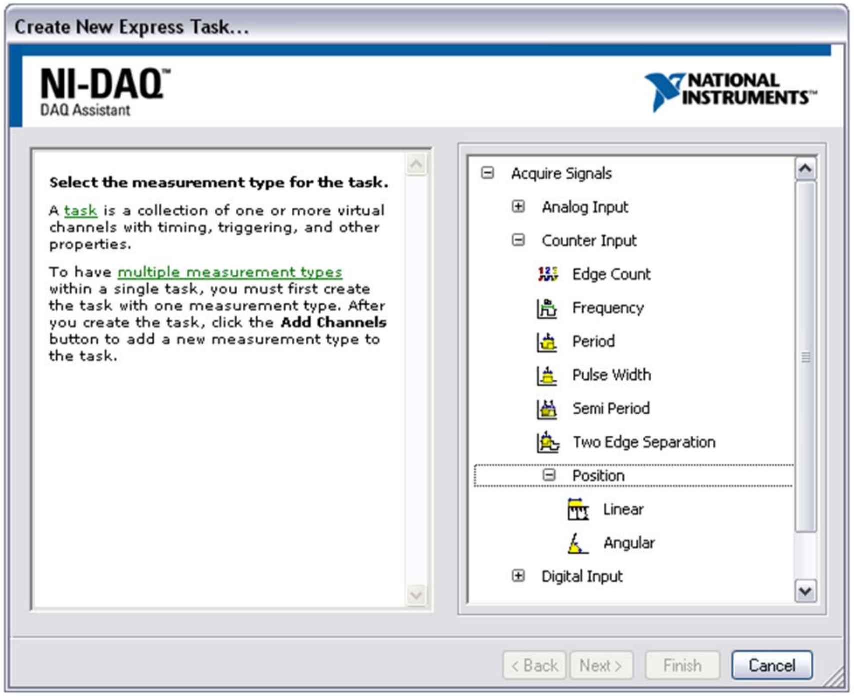
- Source and gate selection offers a great deal of flexibility
  - The source and gate pins for counters can be used for multiple applications
  - You only need to specify the input terminal for your measurement and DAQmx will connect it to the appropriate source or gate depending on the application
- Counter signals can be input on any PFI pins
- DAQ Assistant will give you the default PFI pin for the application

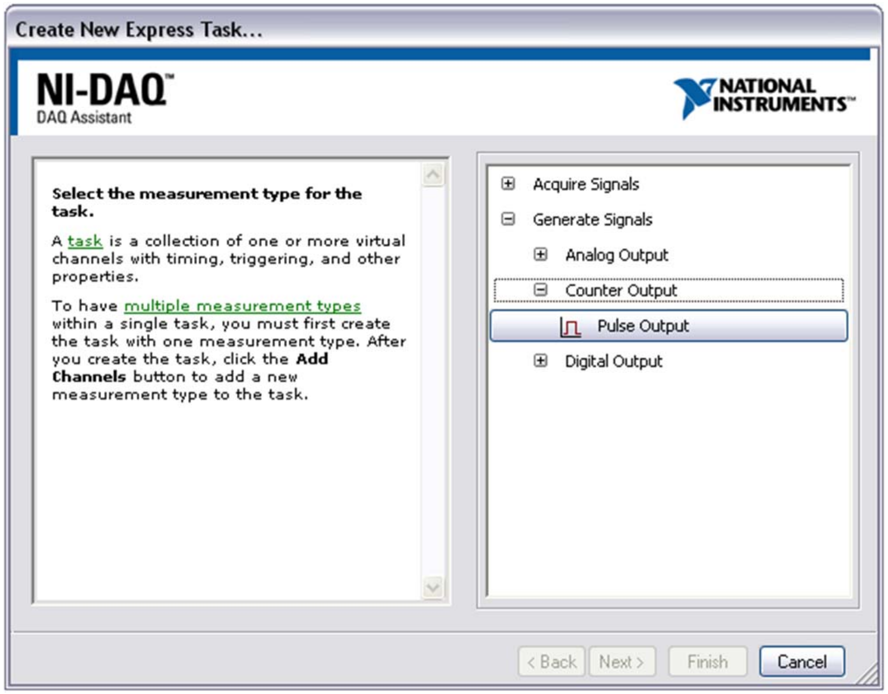
Counter Terminology

- Terminal Count
  - Value of the last count before a counter rolls over to 0
- Resolution
  - The size of the counter register specified in bits
  - Counter register size = $2^{(resolution)} – 1$
  - Typical resolutions - 16, 24, 32 bit
- Timebase
  - Internal signal that can be routed to the source
  - Common timebases – 100 kHz, 20 MHz, 80 MHz, 100 MHz

DAQ Assistant

Correct PFI line for each type of application is automatically chosen by NI-DAQmx





B. Edge Counting

Types of Edge Counting

- Simple
- Pause trigger (gated)
- Continuous buffered
- Finite buffered

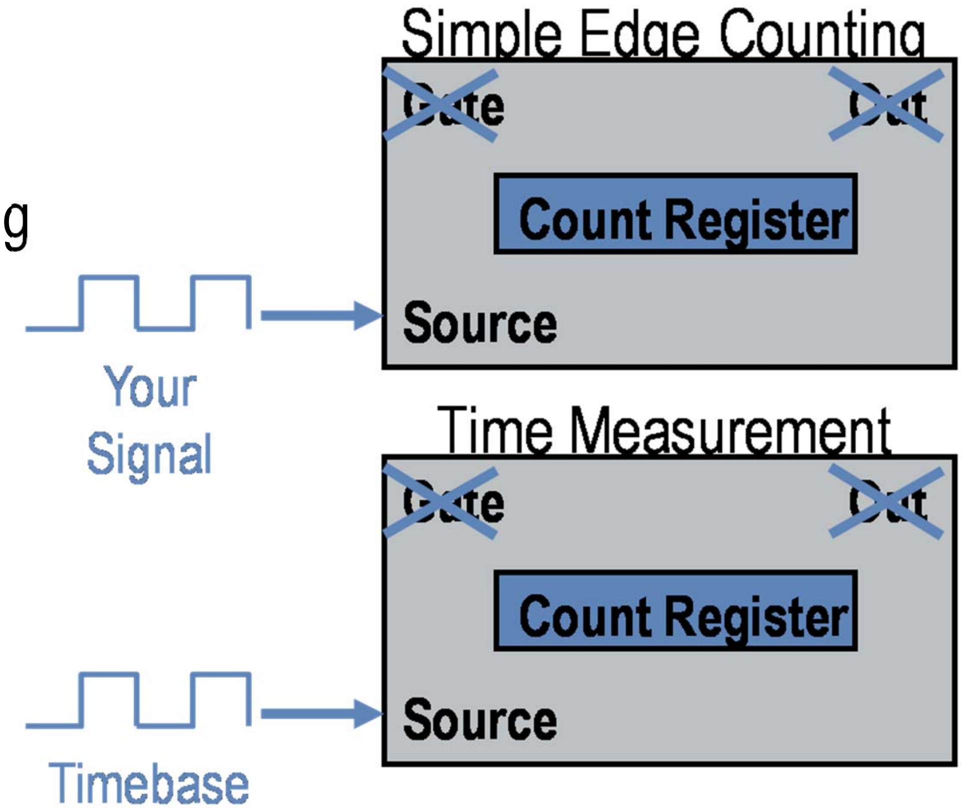
Edge Counting

Active edges on source signal increment the count

- Active edge can be either rising or falling

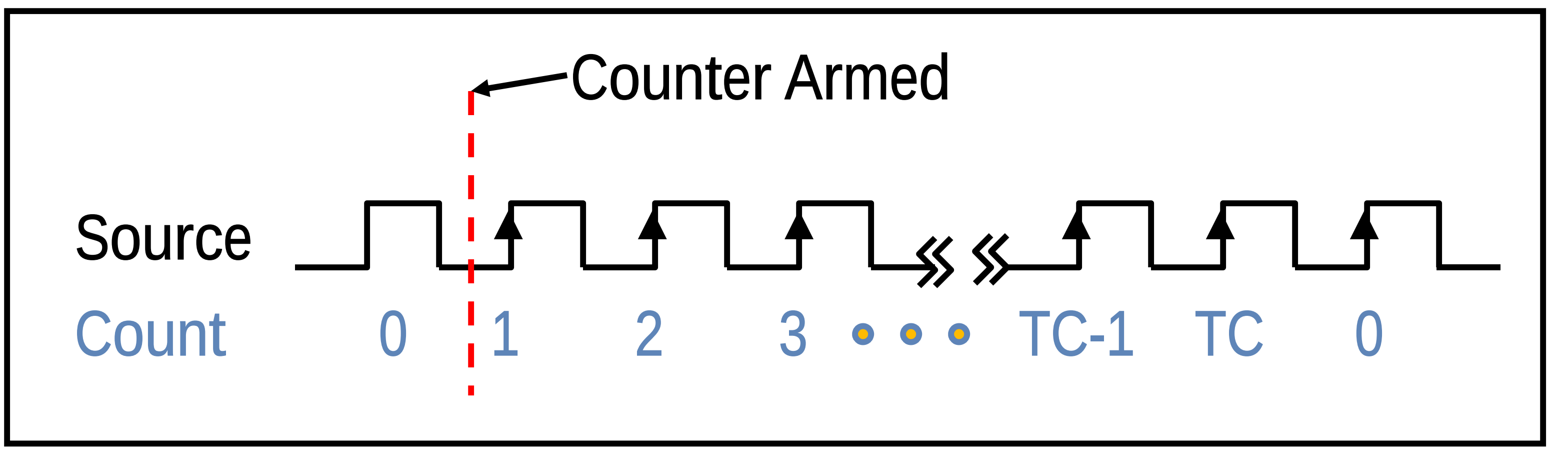
Known frequency timebase

- Time elapsed = (count) x (timebase period)

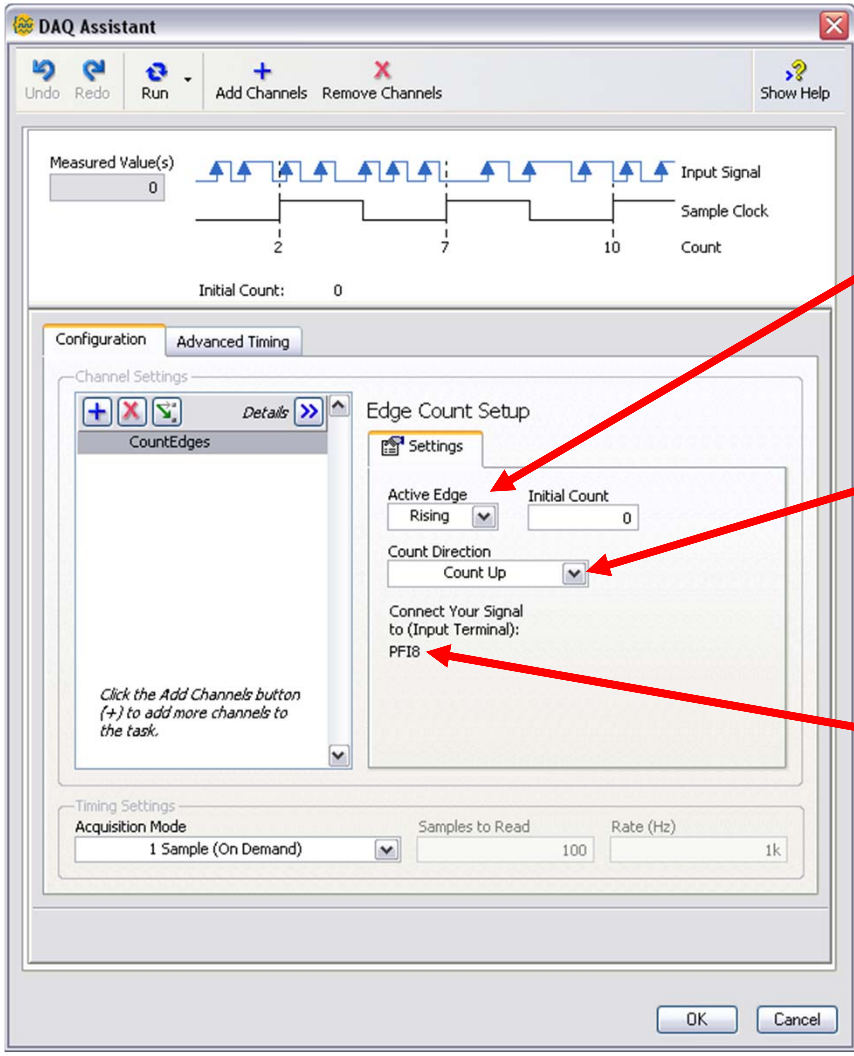


Edge Counting – Simple

- Count increments for the specified edge on the source
- You can change active edge to falling
- Counter will roll over when it reaches terminal count
  - Terminal count = $2^{\text{(Counter resolution)}} – 1$



Edge Counting – DAQ Assistant



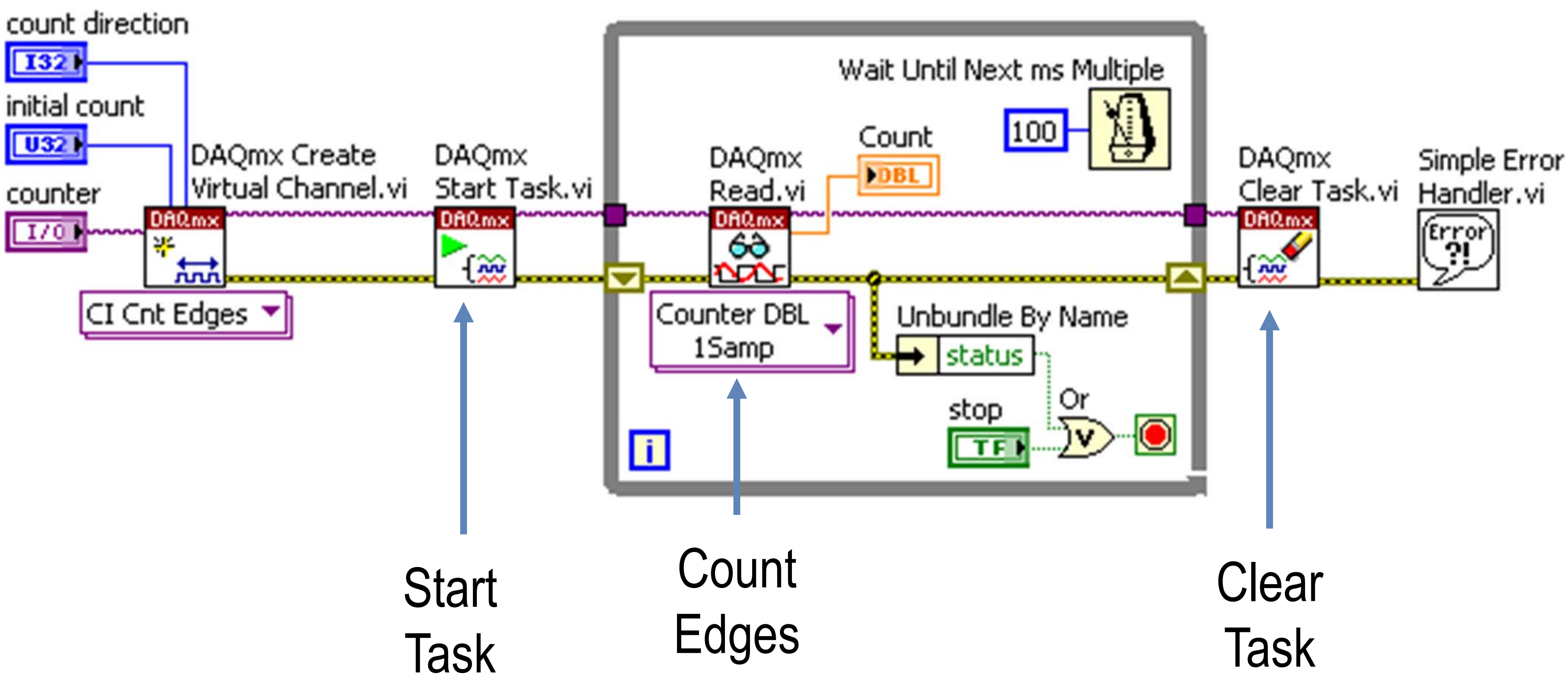
Select Rising or Falling Edge

Select Count Direction:
Count up
Count down
Externally controlled

Automatic PFI line selection

Edge Counting – Simple

Use the *Counter* instance of the DAQmx Read VI



Exercise 6-1: Simple Edge Counting

To create a VI to count the number of edges produced by rotating the knob of the Quadrature Encoder.

GOAL

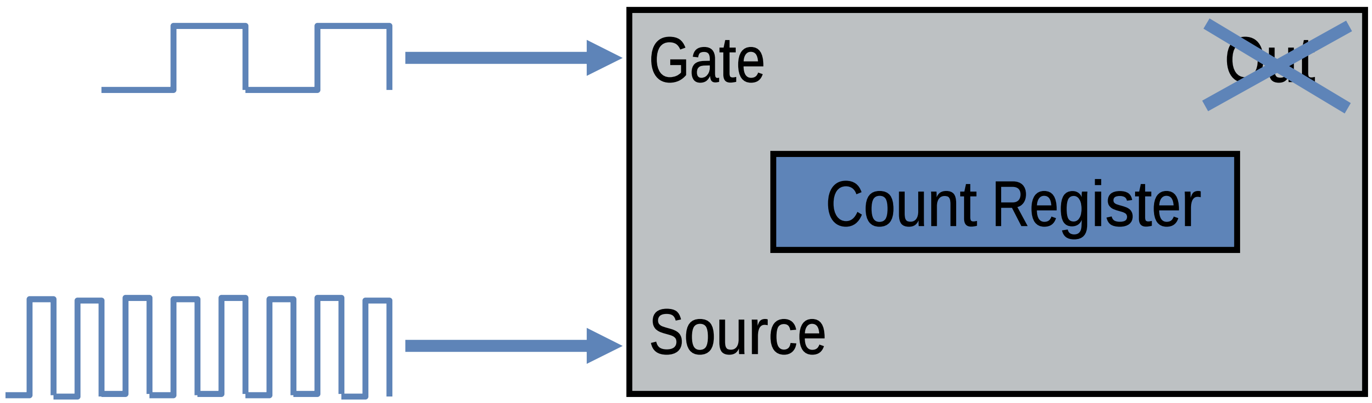
Exercise 6-1: Simple Edge Counting

- What is the highest number that will be output by the count indicator?

DISCUSSION

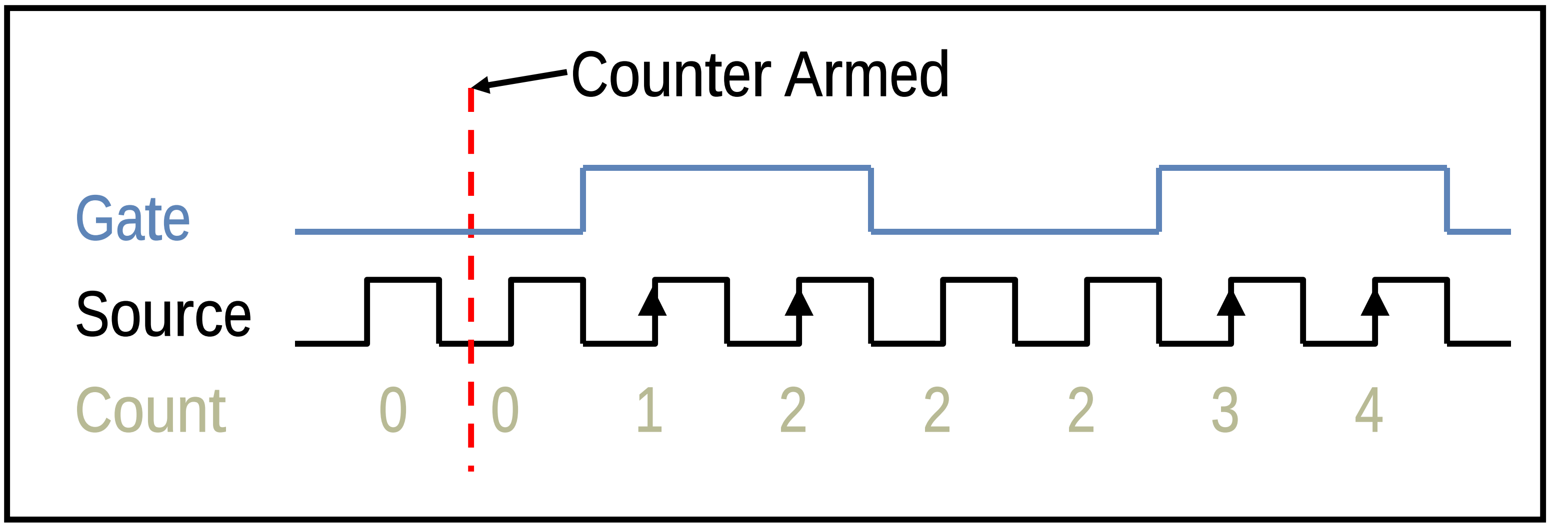
Edge Counting – Pause Trigger (Gated)

- Non-buffered
- Gate pauses increment/decrement of the count register
- Active edges on source increment register only when gate is enabled



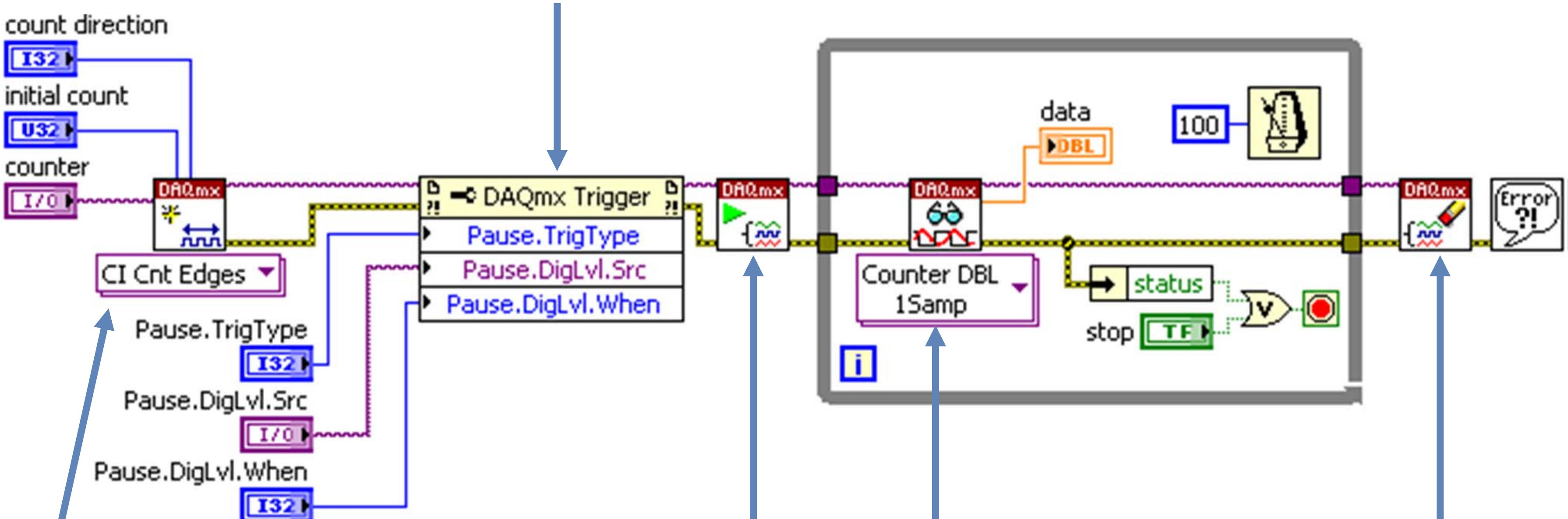
Edge Counting – Pause Trigger (Gated)

- Counter increments or decrements when gate is high or low (parameters are software selectable)
- Counter will pause counting while in the non-active state



Edge Counting – Pause Trigger Example

Pause trigger is set with DAQmx Trigger property node



Create Virtual Channel

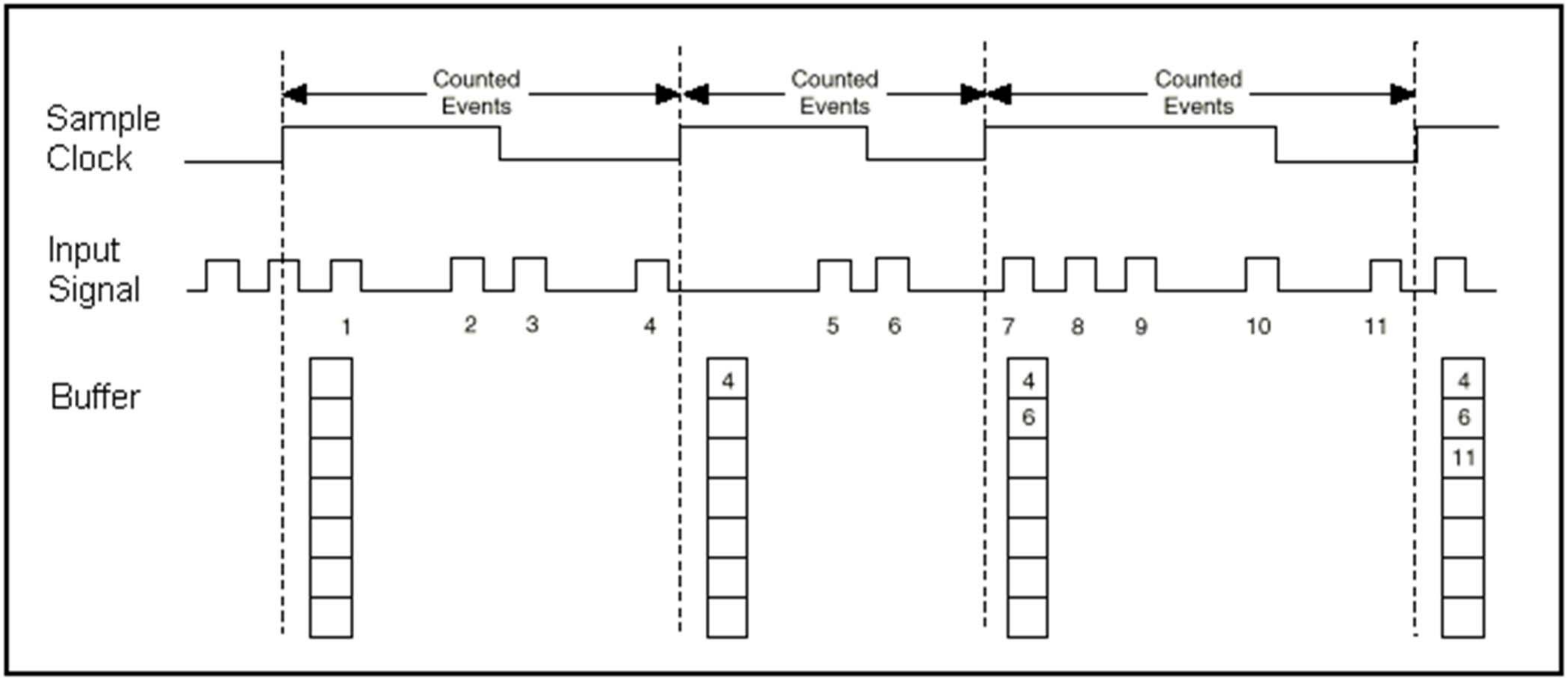
Start Task

Count Edges

Clear Task

Edge Counting – Continuous Buffered

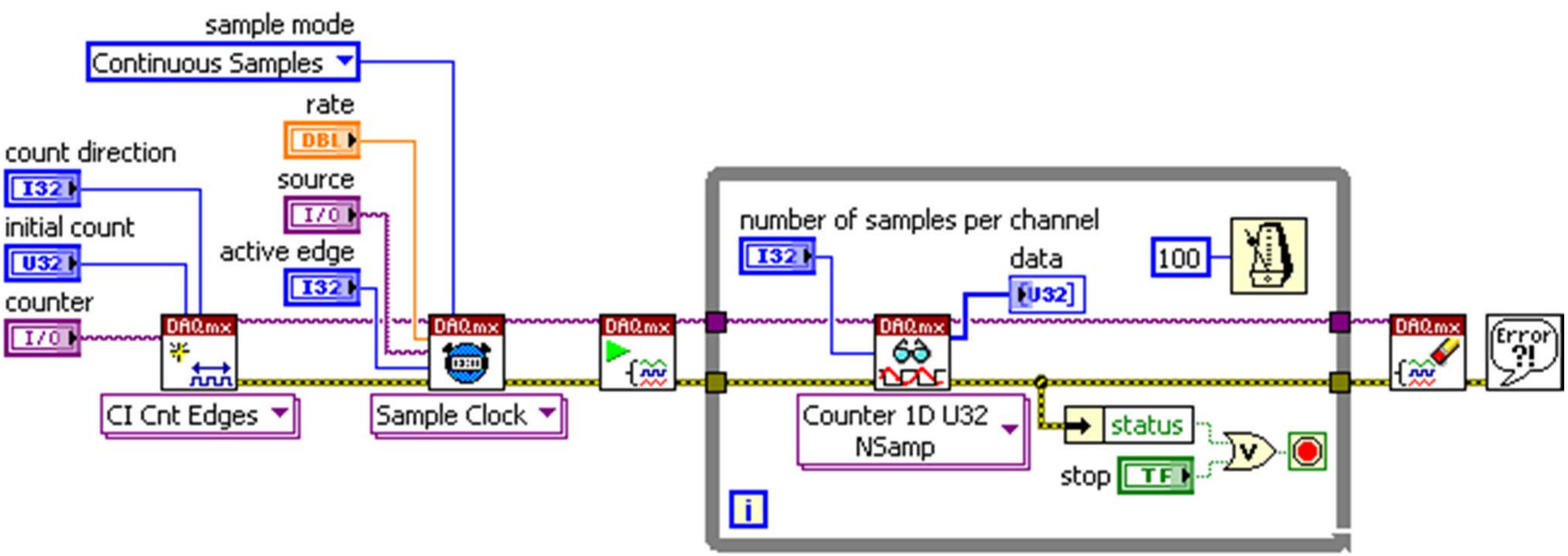
Device latches the number of edges counted onto each active edge of the sample clock and stores the number in the buffer



Edge Counting – Continuous Buffered

Use the DAQmx Timing VI

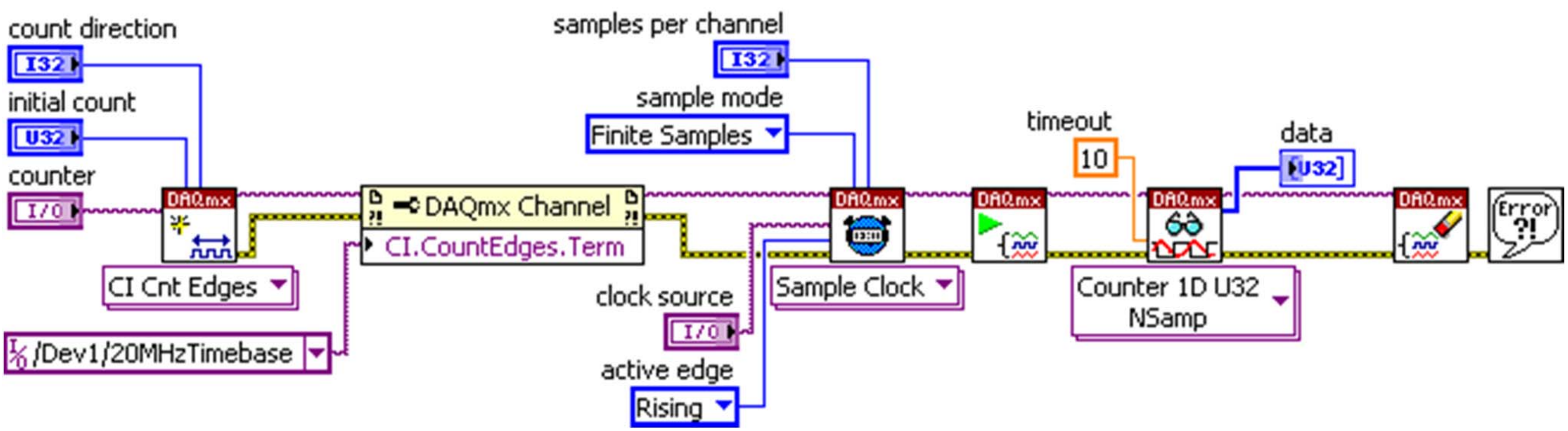
- Source determines when a sample is inserted into the buffer



Edge Counting – Finite Buffered

Use DAQmx Timing VI

- Source determines when a sample is inserted into the buffer
- Counter stops reading when the Samples per Channel value is reached



Exercise 6-2: Advanced Edge Counting

To use a pause trigger and finite buffered methods to perform edge counting.

GOAL

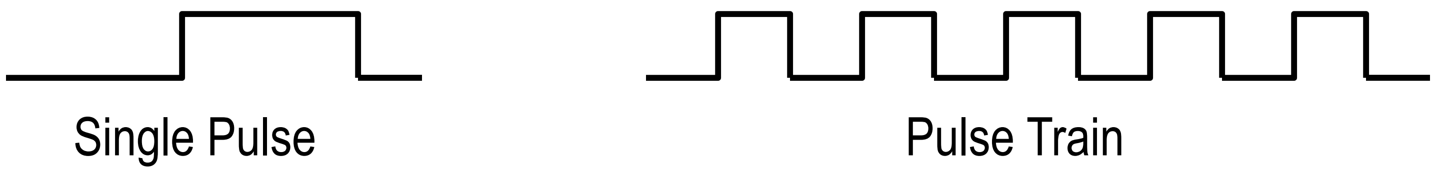
Exercise 6-2: Advanced Edge Counting

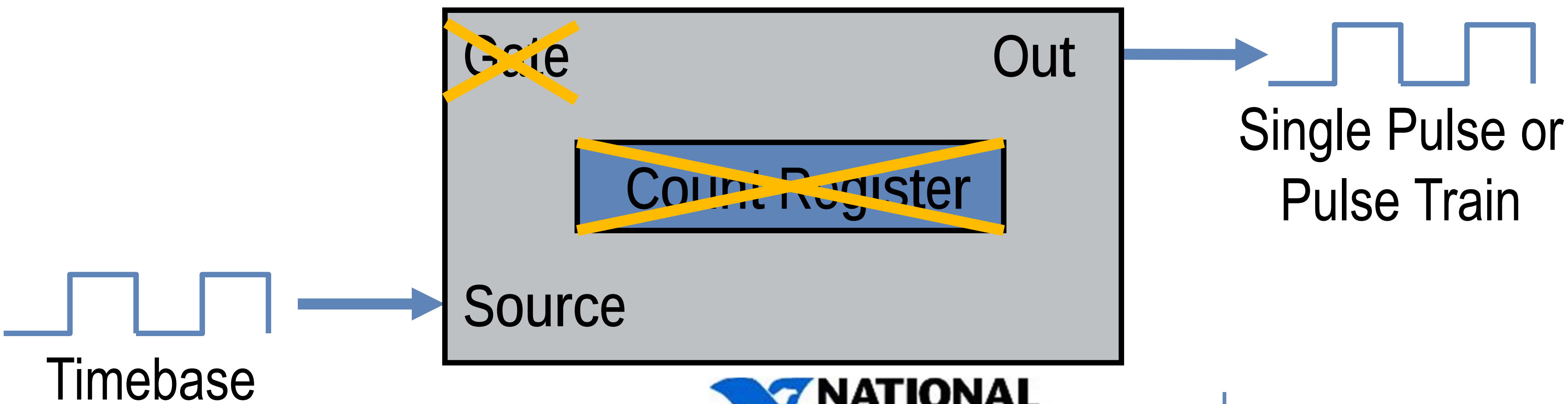
- When would finite buffered counting be helpful?

DISCUSSION

C. Pulse Generation

- A pulse is rapid change in amplitude of a signal from its idle value to an active value for a short period of time
- Generate TTL pulse or pulse train on the counter out pin

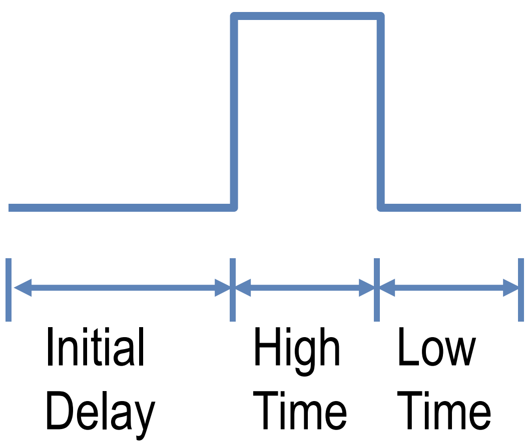




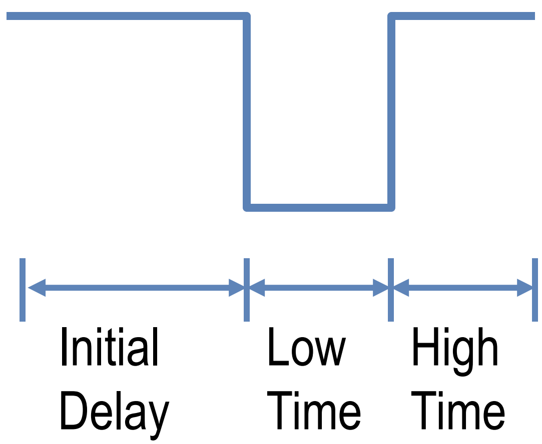
Pulse Characteristics

- Can have a low idle state or high idle state
- Consists of three parts
  - High time, low time, initial delay

Idle state is low

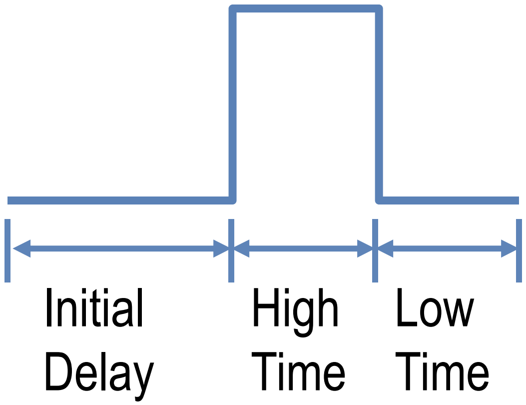


Idle state is high

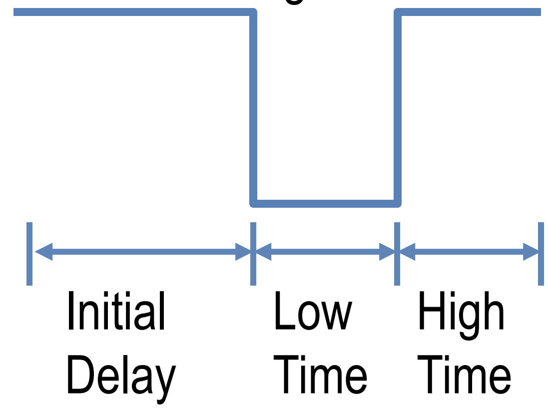


Pulse Characteristics

Idle state is low



Idle state is high

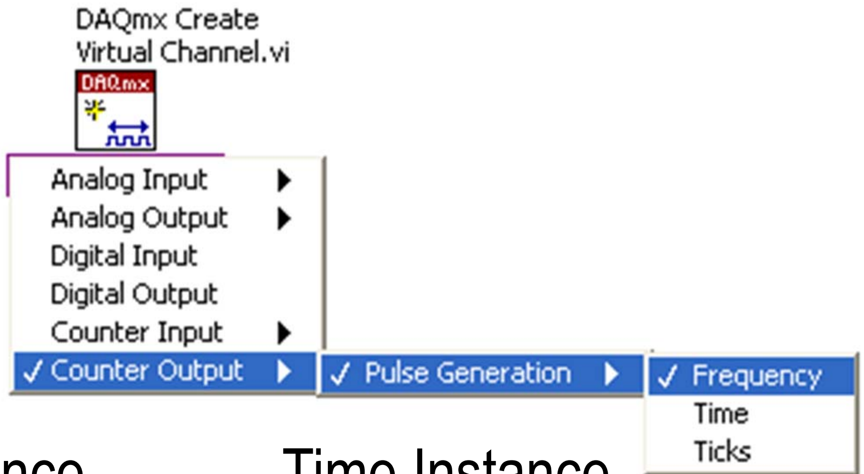


$$\text{Pulse Period} = \text{High Time} + \text{Low Time}$$

$$\text{Pulse Frequency} = \frac{1}{\text{Pulse Period}}$$

$$\text{Duty Cycle} = \frac{\text{High Time}}{\text{Pulse Period}}$$

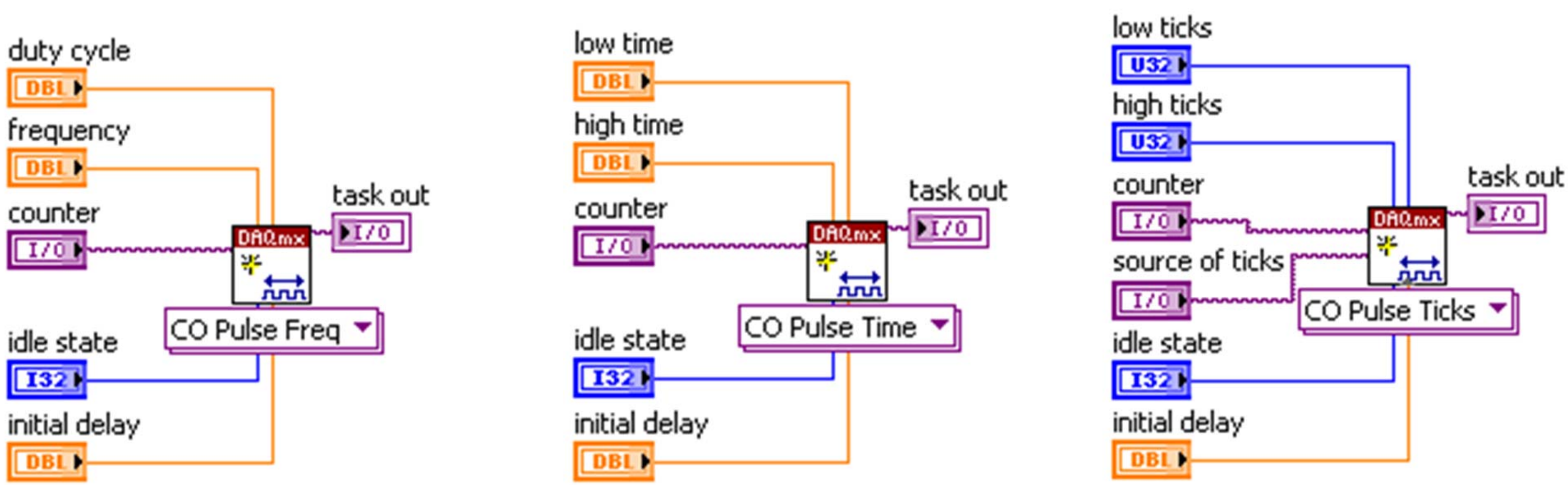
DAQmx Create Virtual Channel



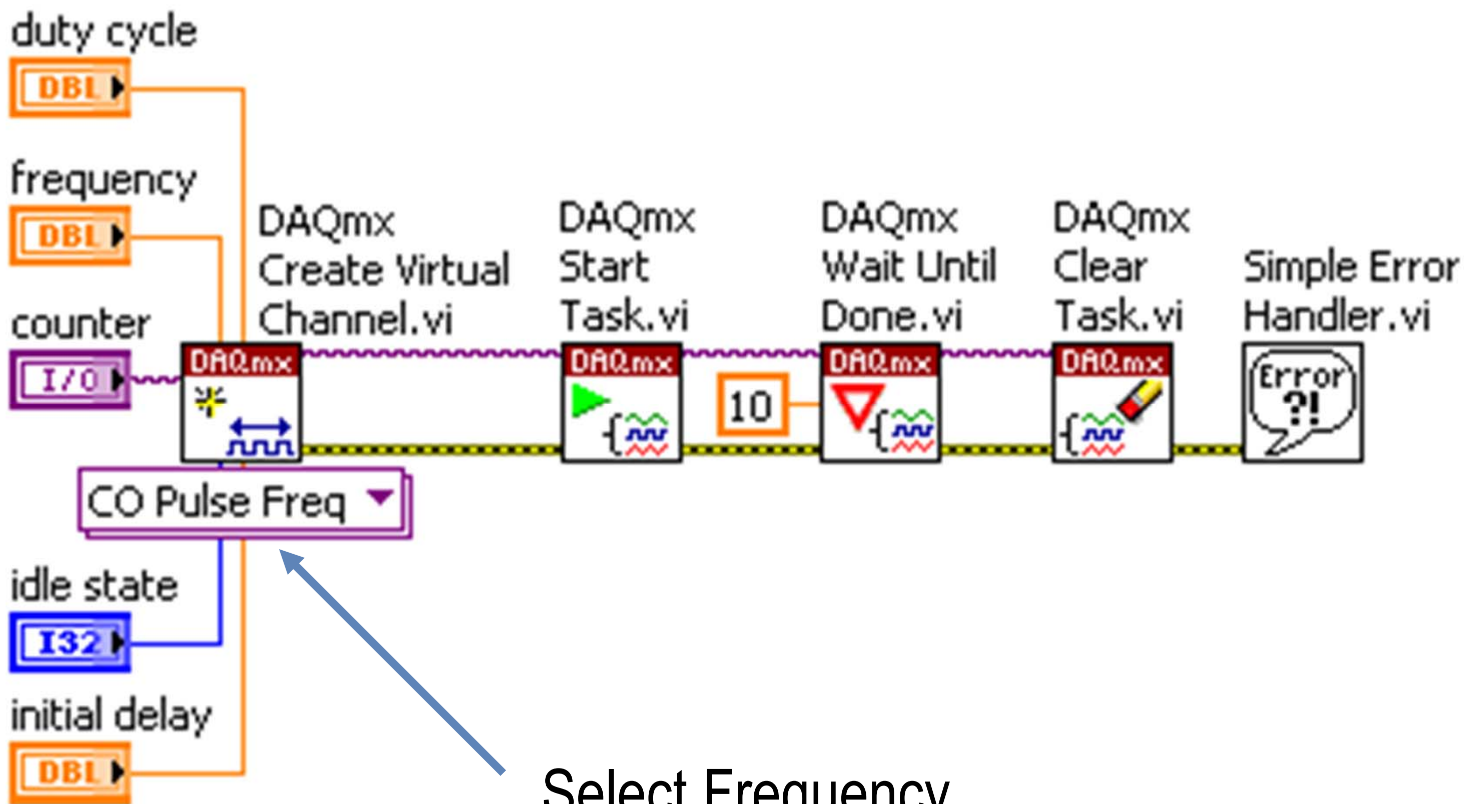
Frequency Instance

Time Instance

Ticks Instance



Single Pulse Generation



Select Frequency, Time, or Ticks

Exercise 6-3: Pulse Generation

To build a VI that generates a single pulse using a counter.

GOAL

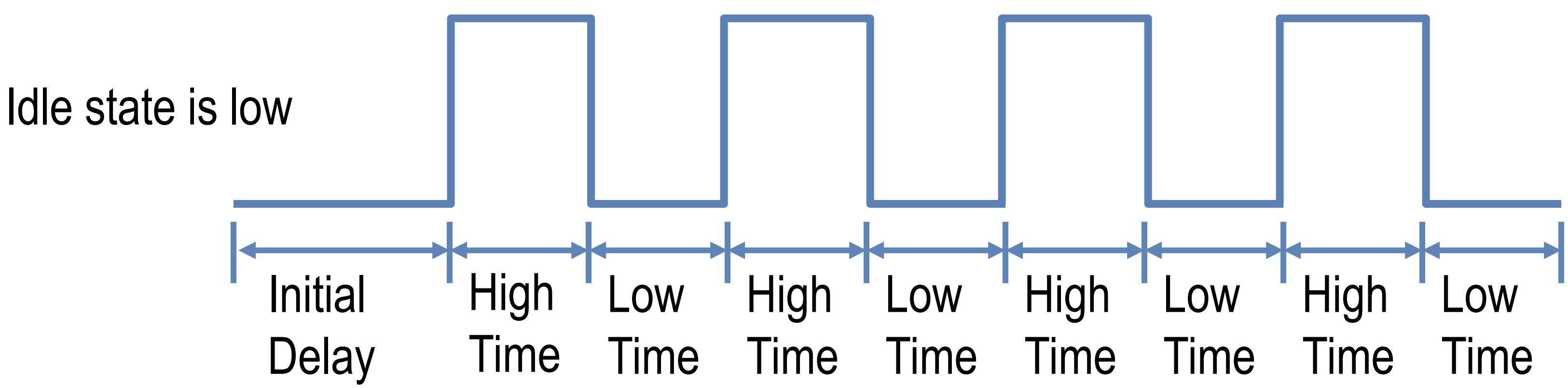
Exercise 6-3: Pulse Generation

- This VI defines the pulse by setting the duty cycle and frequency.
  How would you modify the VI to define the pulse by setting the high time and low time instead?

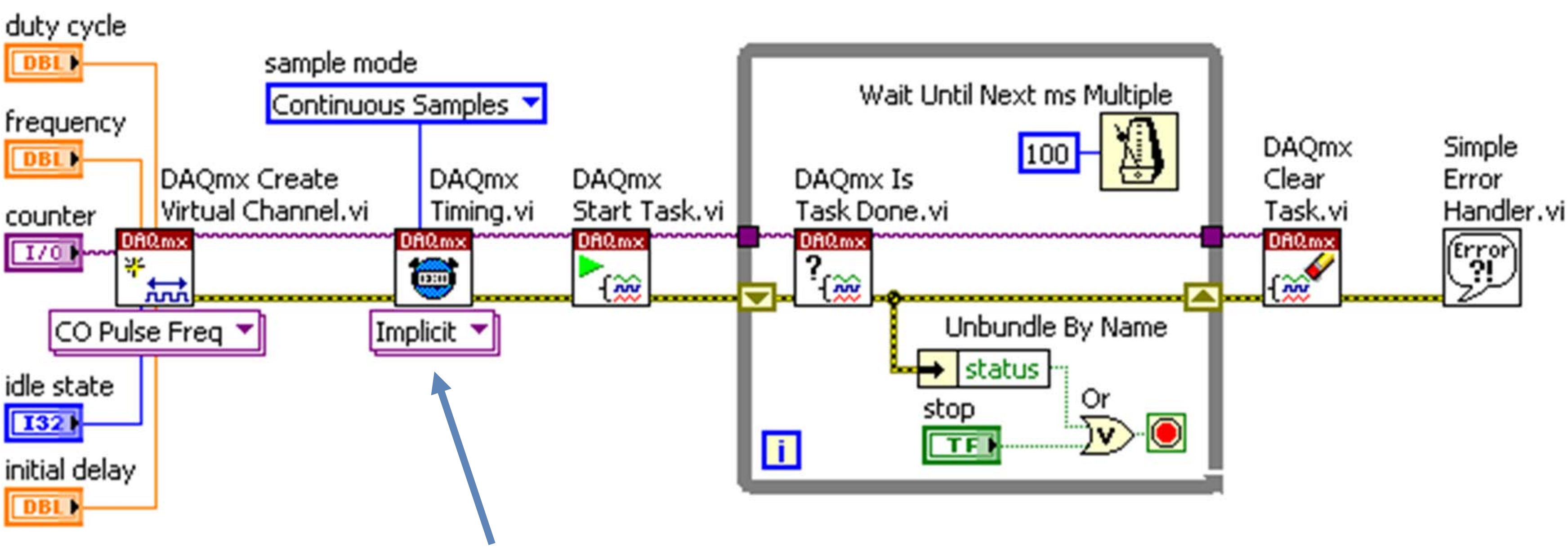
DISCUSSION

Pulse Train Generation

- Pulse Train – Generates more than one pulse
- Initial Delay – How long the output remains in the idle state before generation
- High Time – Amount of time the pulse is at a high level (5V)
- Low Time – Amount of time the pulse is at a low level (0V)



Pulse Train Generation



Timing VI:
Implicit and
continuous

Exercise 6-4: Pulse Train Generation

To create a VI to generate a pulse train.

GOAL

Exercise 6-4: Pulse Train Generation

- How would you modify this VI to only output 10 pulses?

DISCUSSION

Finite Pulse Train

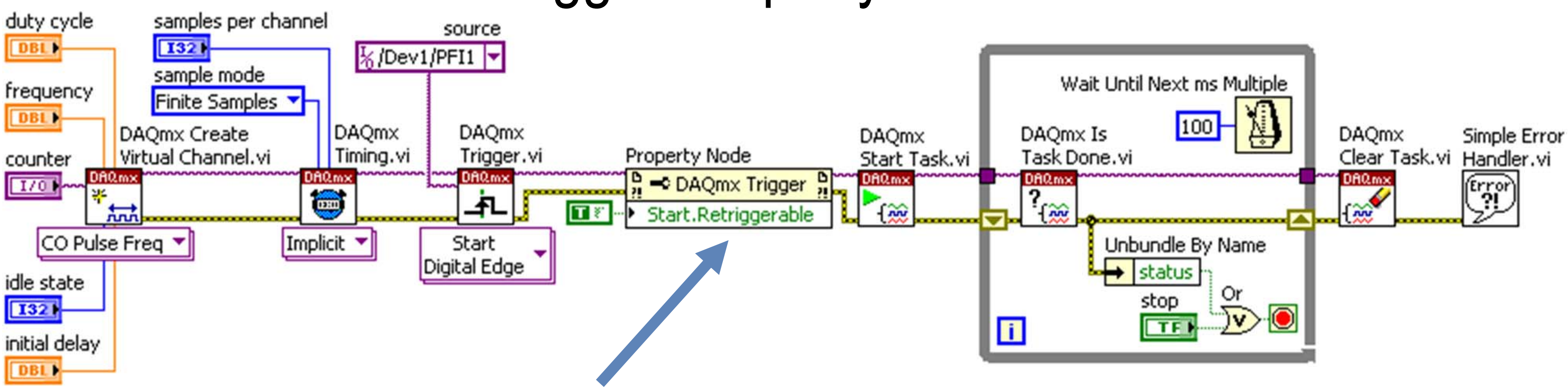
STC2-based devices require the use of 2 counter channels

- First counter generates continuous pulse train
- Second counter generates pulse that gates first counter
- Pulse width determines length of pulse train
- DAQmx uses pulse train settings to determine gating pulse width
  - Everything is managed by DAQmx
  - User simply specifies the counter output channel, pulse train characteristics, and number of pulses to generate

STC3-based devices only use 1 counter channel

Retriggerable Finite Pulse Train

- Similar to a Finite Pulse Train
- The counter pulse used for gating is retriggerable
  - Can use this property to create a retriggerable finite pulse train
- Set with DAQmx Trigger Property Node



DAQmx Trigger Property Node used for Retriggerable Operations

Exercise 6-5: Retriggerable Pulse Train Generation

To build a VI that generates a retriggerable finite pulse train.

GOAL

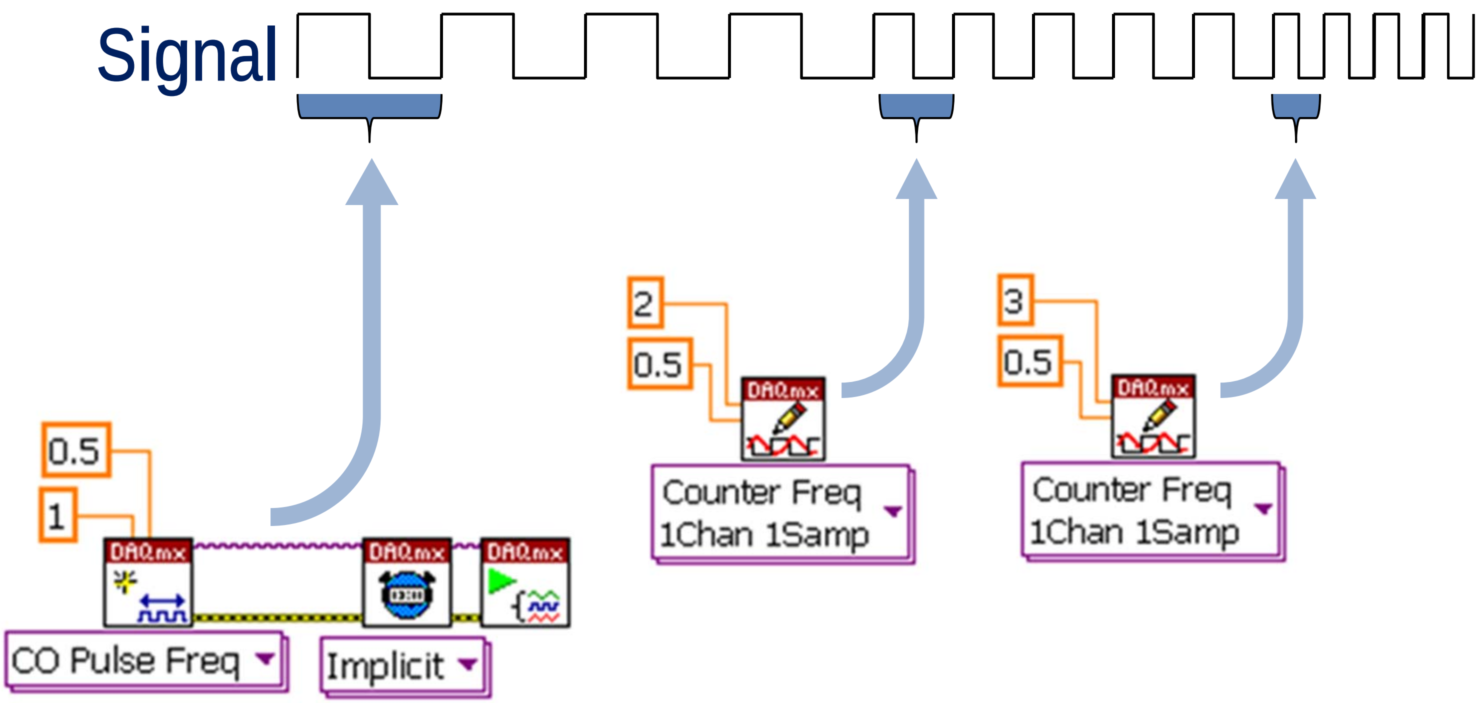
Exercise 6-5: Retriggerable Pulse Train Generation

- If the PFI1 line receives another trigger pulse while the VI is outputting 5 pulses, will the VI output another 5 pulses afterwards?

DISCUSSION

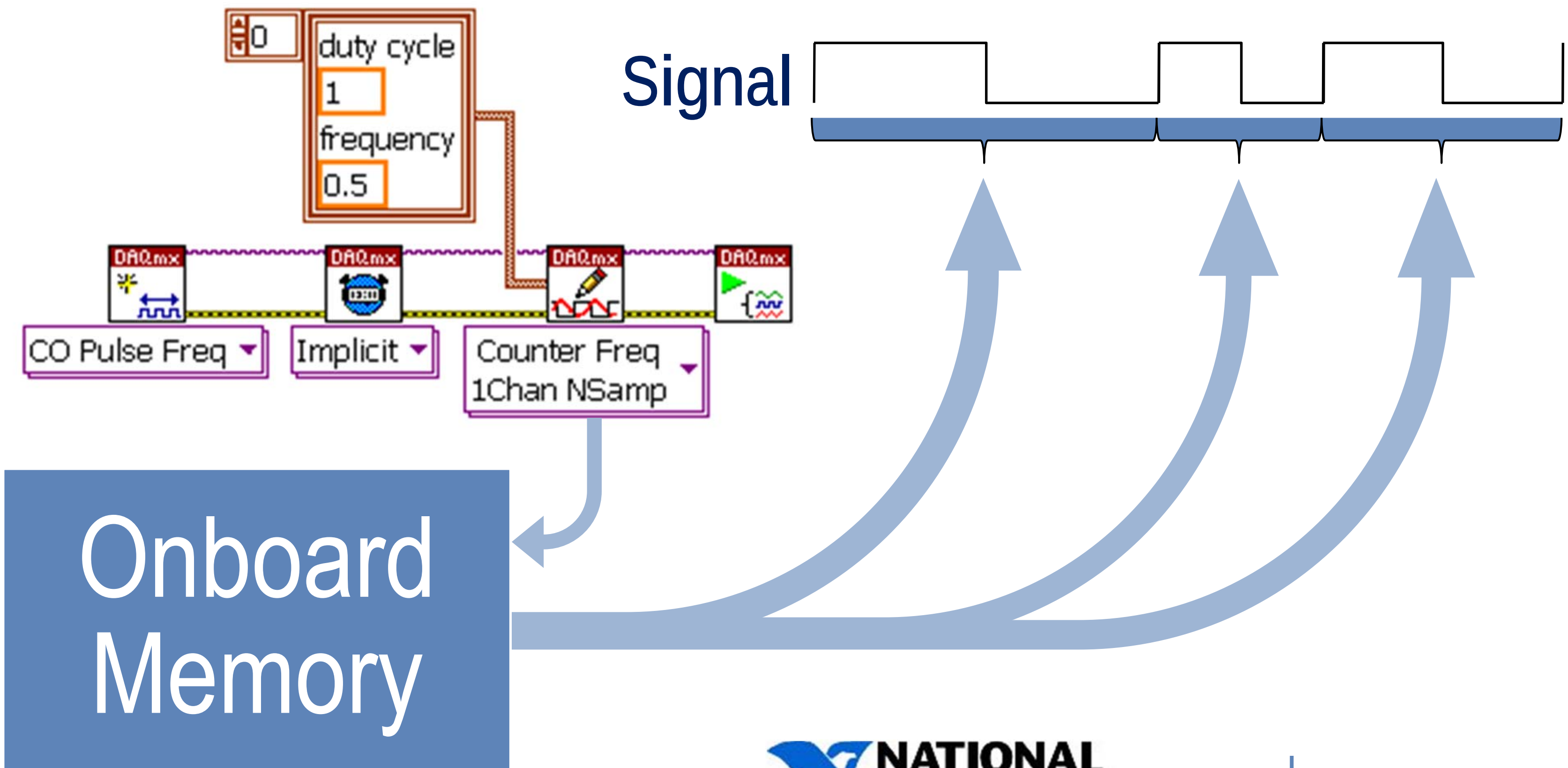
Non-Buffered Pulse Train (Implicit Timing)

- Supported in STC2-based and STC3-based DAQ devices
- Changing pulse parameters is software-timed



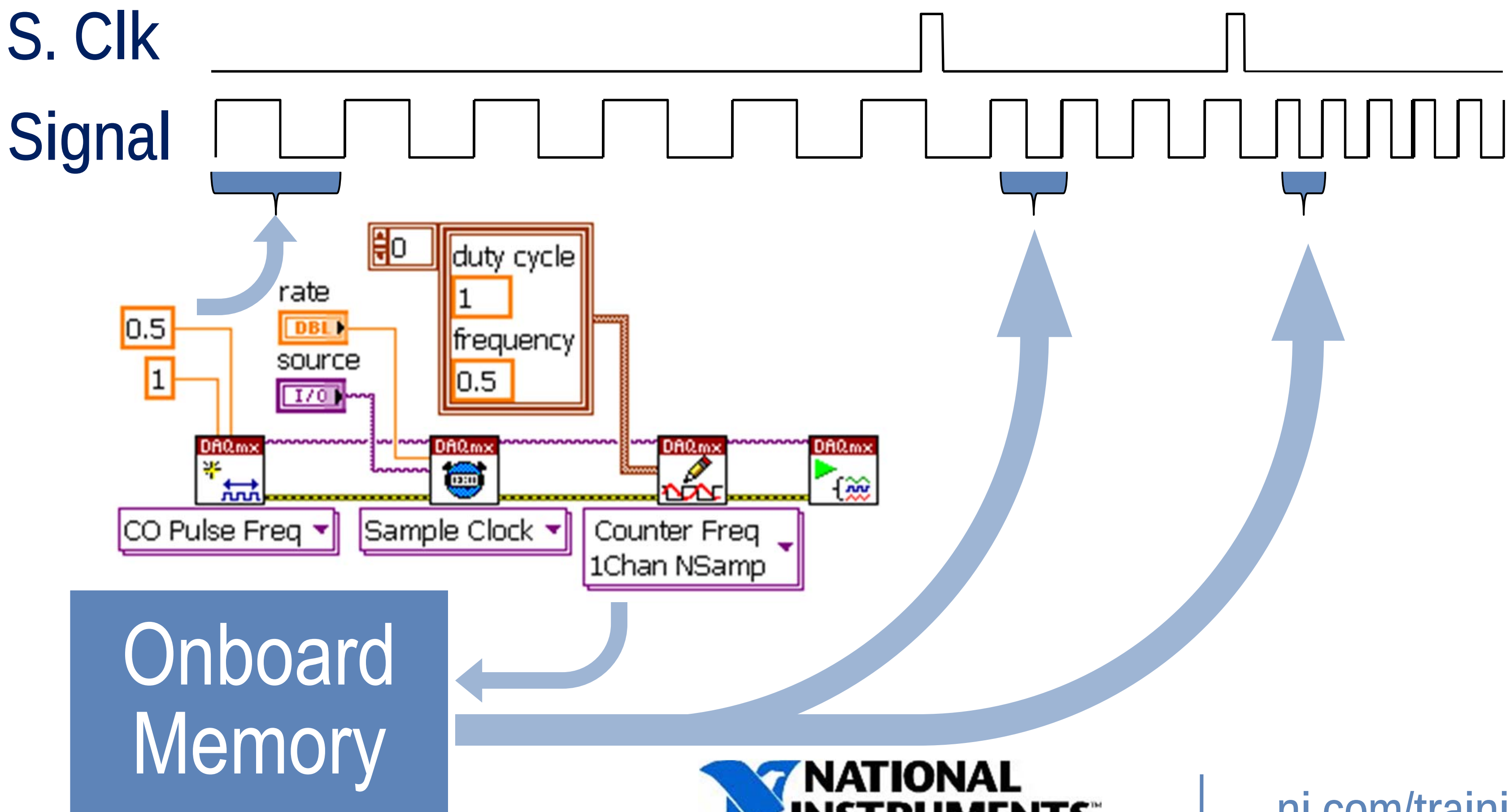
Buffered Pulse Train (Implicit Timing)

- Supported in STC3-based DAQ devices (X series)
- Changing pulse parameters is hardware-timed



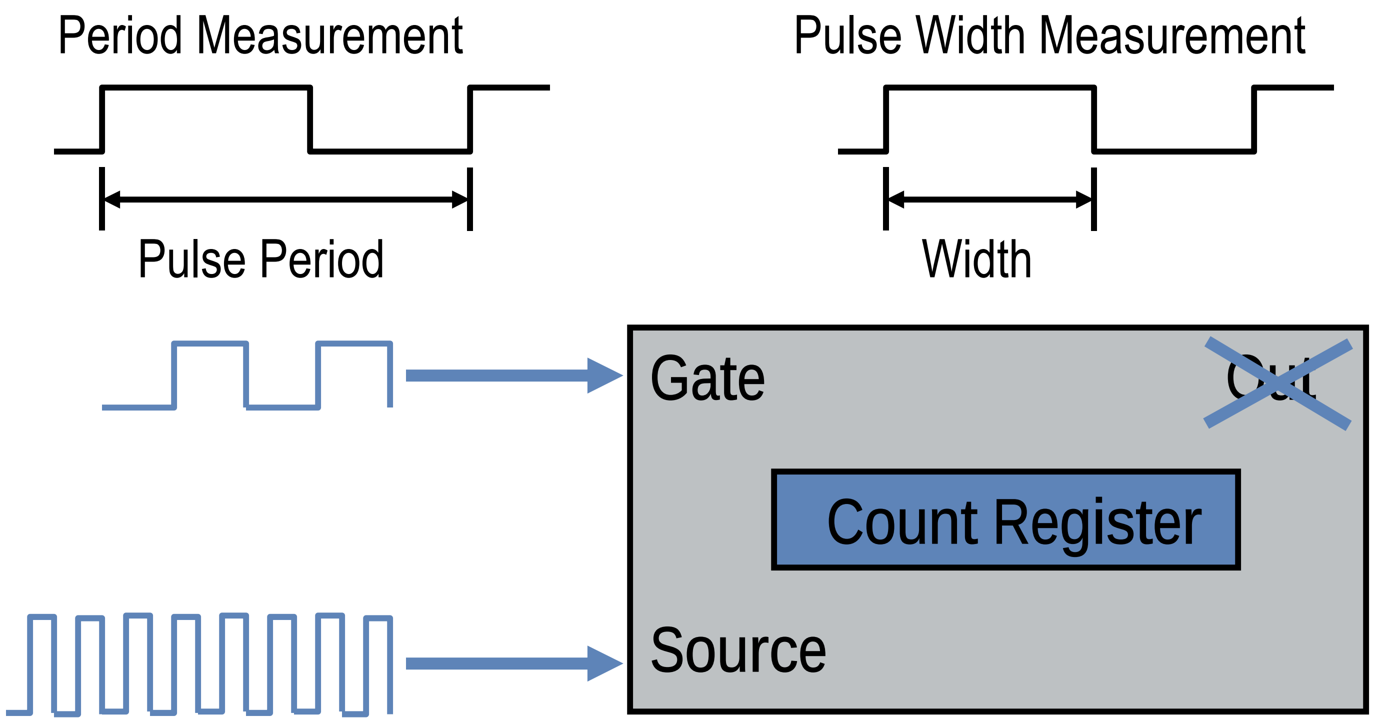
Buffered Pulse Train (Sample Clock Timing)

- Supported in STC3-based DAQ devices (X series)
- Changing pulse parameters is hardware-timed



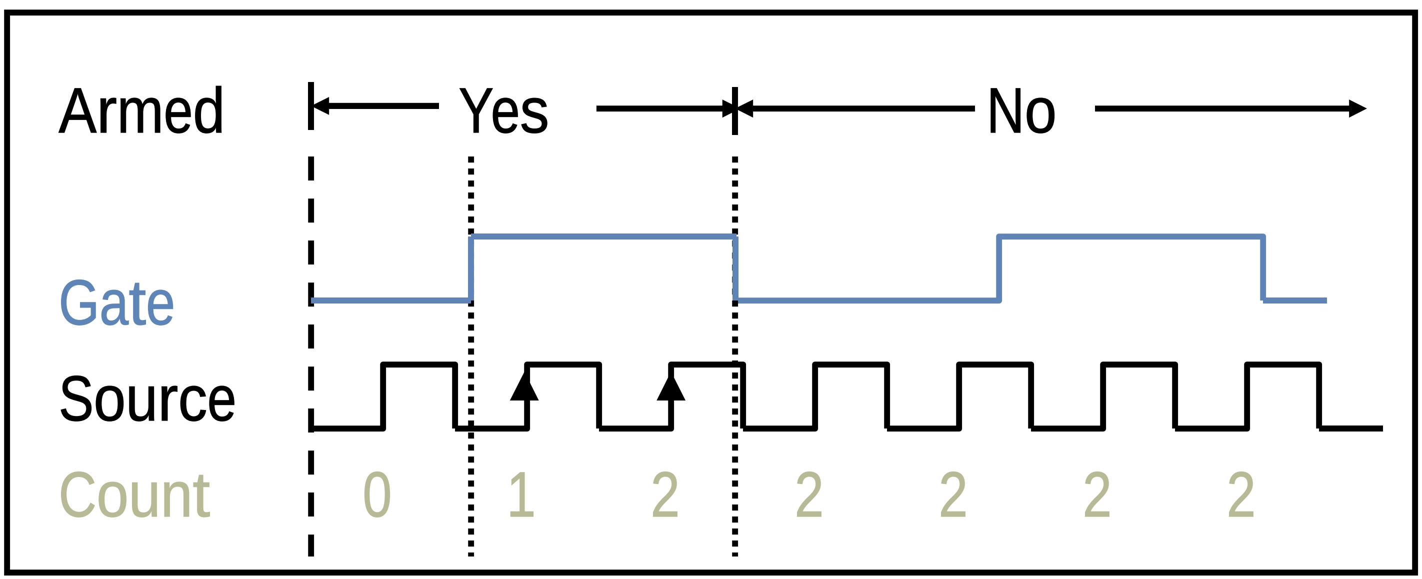
D. Pulse Measurement

Use a timebase with a known frequency to measure characteristics of a unknown signal

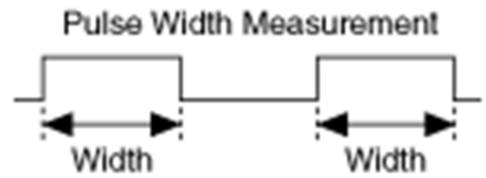


Pulse Width Measurement

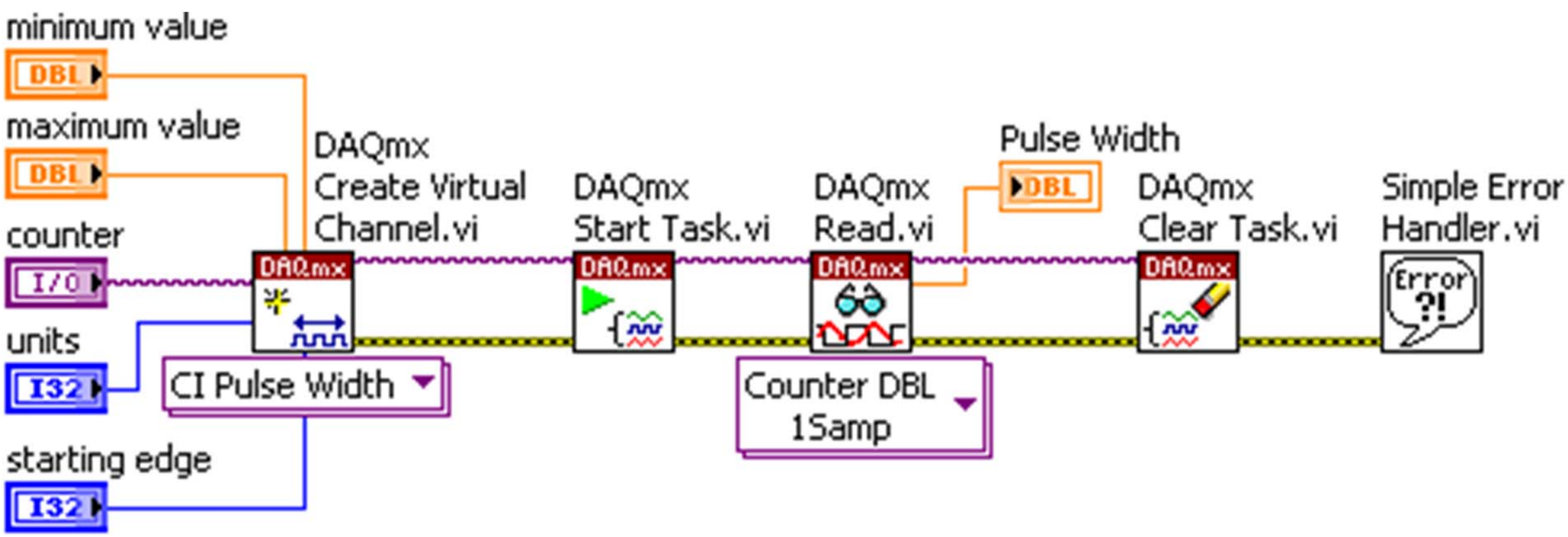
- Count will increment for each rising edge on source
  - Counting can start on either rising or falling edge
- Width of Gate = (Count) x (1/source frequency)



Single Pulse Width Measurement

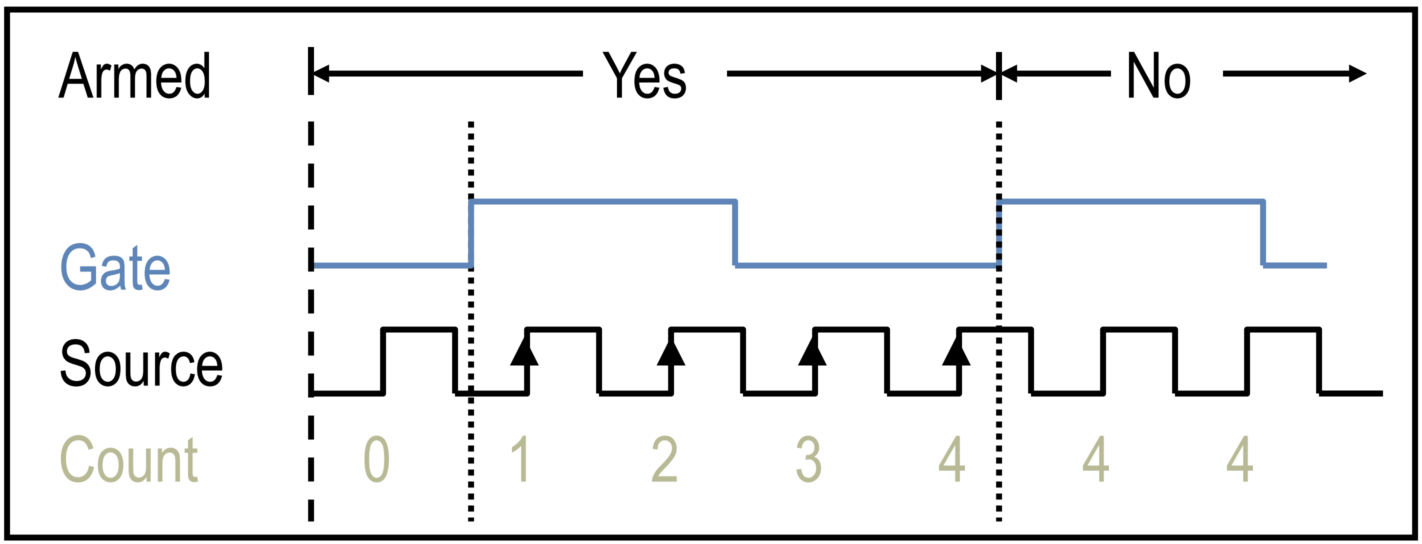


Set maximum and minimum values of the unknown pulse as accurately as possible. This allows NI-DAQmx to choose the best internal timebase.

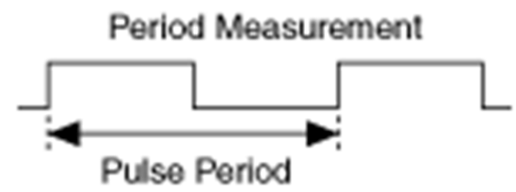


Period Measurement

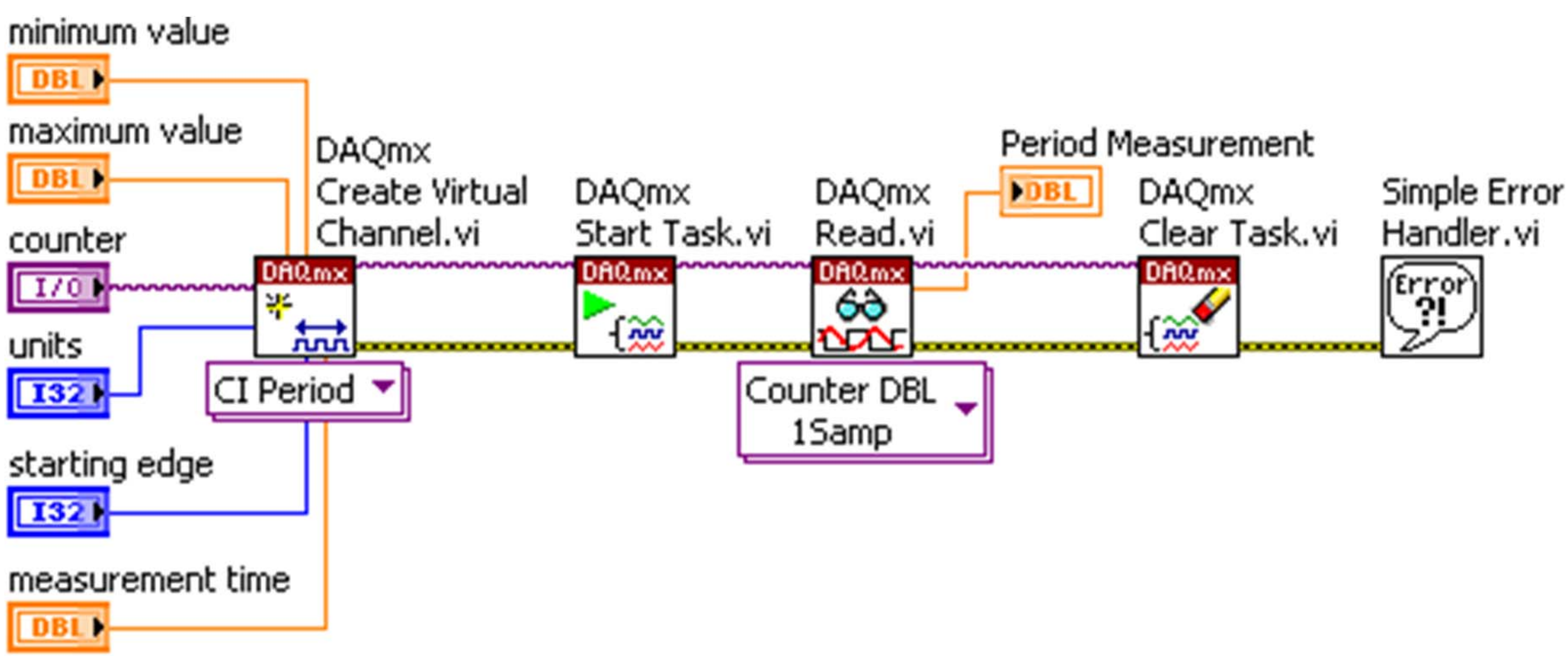
- Count will increment for each rising edge on source
  - Counting can either start and end on rising or falling edges
- *Period of Gate = (Count) x (1/source frequency)*



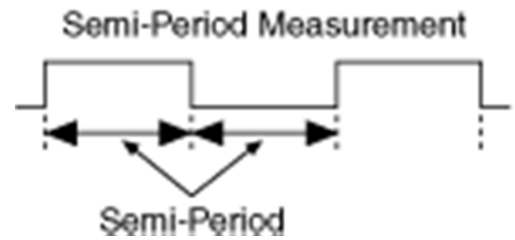
Single Period Measurement



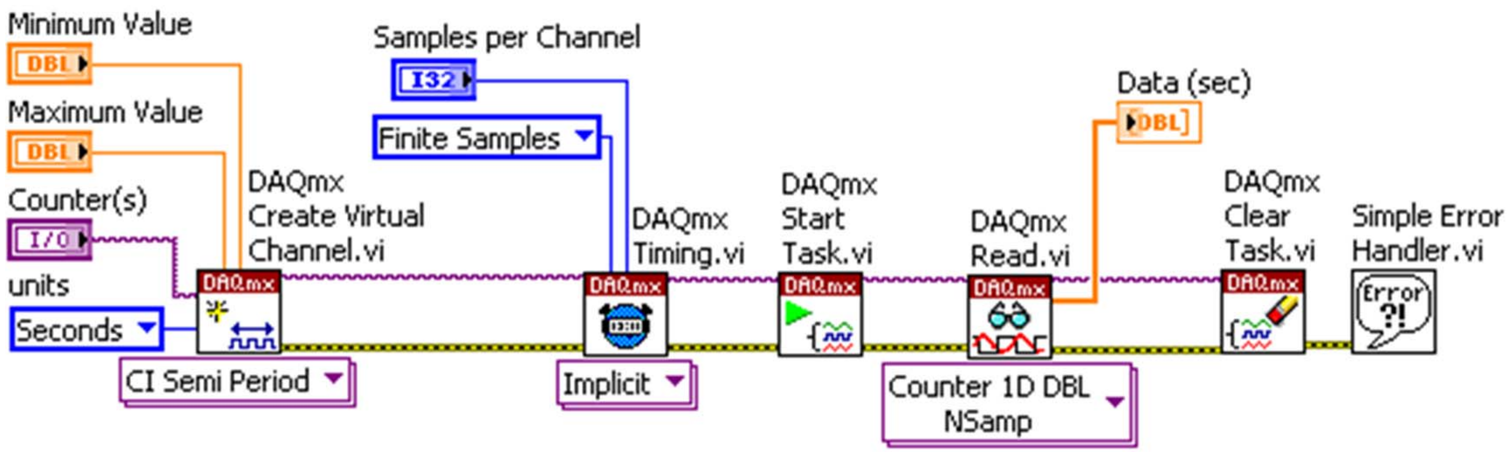
Like pulse width measurement, select **Counter Input»Period** from the Create Virtual Channel pull-down menu

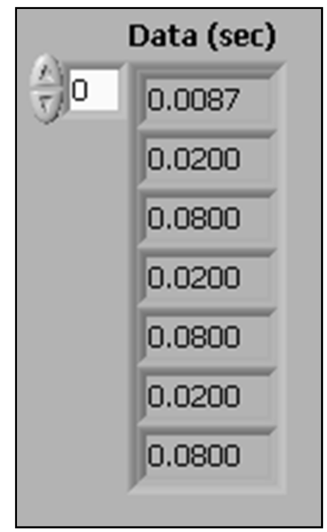


Semi Period Measurement



- Measures time between consecutive edges
- Select Counter Input»Semi Period from the Create Virtual Channel pull-down menu





Exercise 6-6: Pulse Width and Period Measurement

To build a VI to measure the pulse width and period of a pulse train.

GOAL

Exercise 6-6: Pulse Width and Period Measurement

- What measurement is the inverse of period?

GOAL

E. Frequency Measurements – STC2 Chip

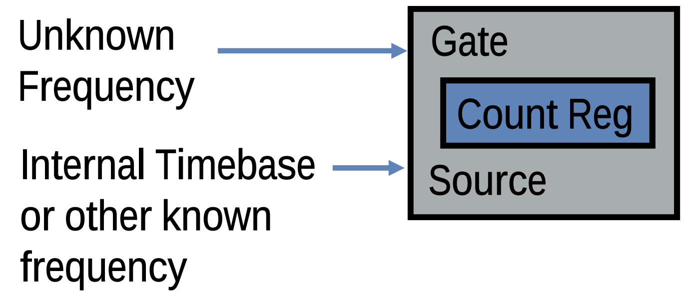
DAQ devices using NI-STC2 chip have three ways to measure frequency

- Low Frequency with 1 Counter
  - Good for low frequencies (depends on allowable error)
  - Uses 1 counter
- High Frequency with 2 Counters
  - Good for high frequencies (depends on allowable error)
  - Utilizes 2 counters, DAQmx automatically reserves the second
- Large Range with 2 Counters
  - Good for varying frequencies
  - Utilizes 2 counters, DAQmx automatically reserves the second

Frequency Measurement – Low Frequency with 1 Counter Method

Use period measurement and take the inverse

- Frequency = 1 / Period



Pros	Cons
• Only uses 1 counter • Good at low frequencies *(f < timebase / 100)*	• Can see large error at high frequencies due to phenomenon called quantization error

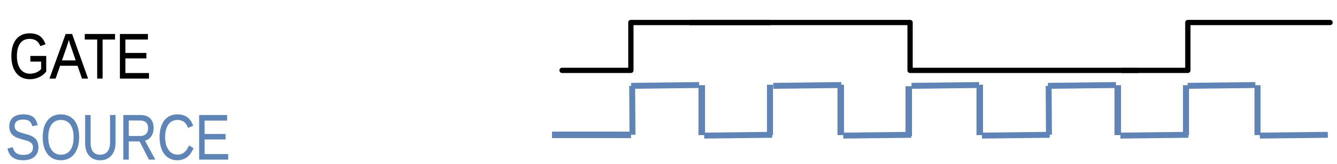
Quantization Error

Quantization error

- Inherent uncertainty in digitizing an analog value as a result of the finite resolution of the conversion process

Example

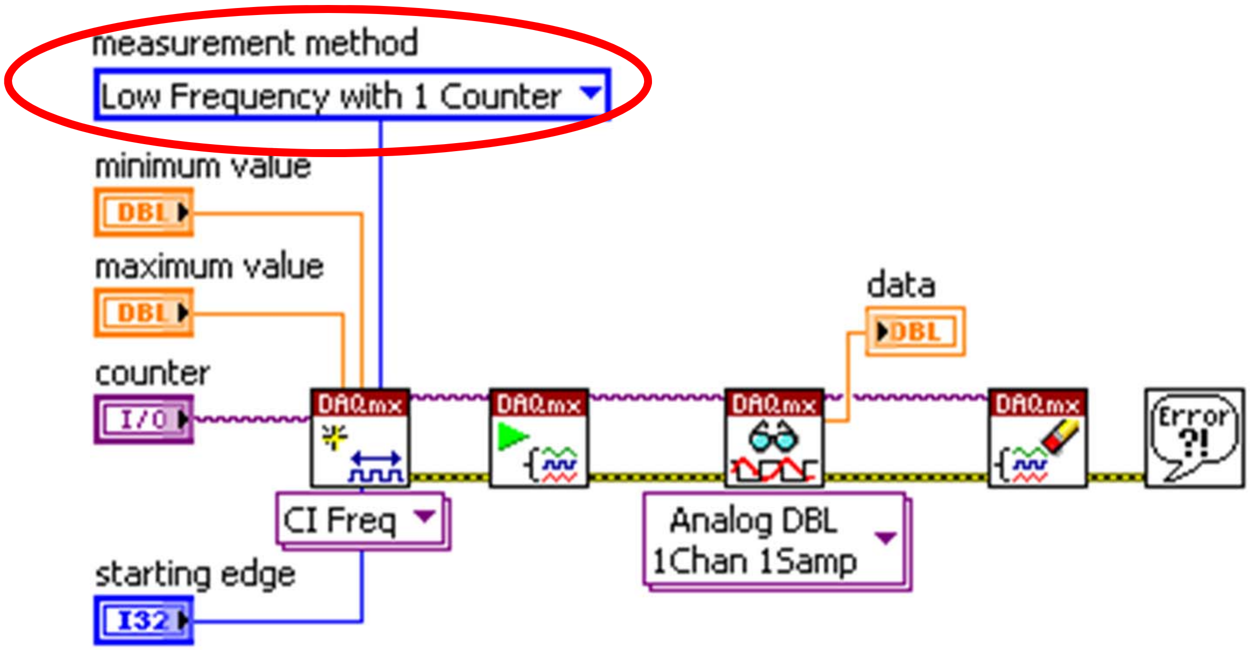
- Gate period is exactly four source cycles
- Measurement could be off by +/- 1 source cycles



Miss both edges	0	1	2	3	3
Miss one, catch one	0	1	2	3	4
Catch both edges	1	2	3	4	5

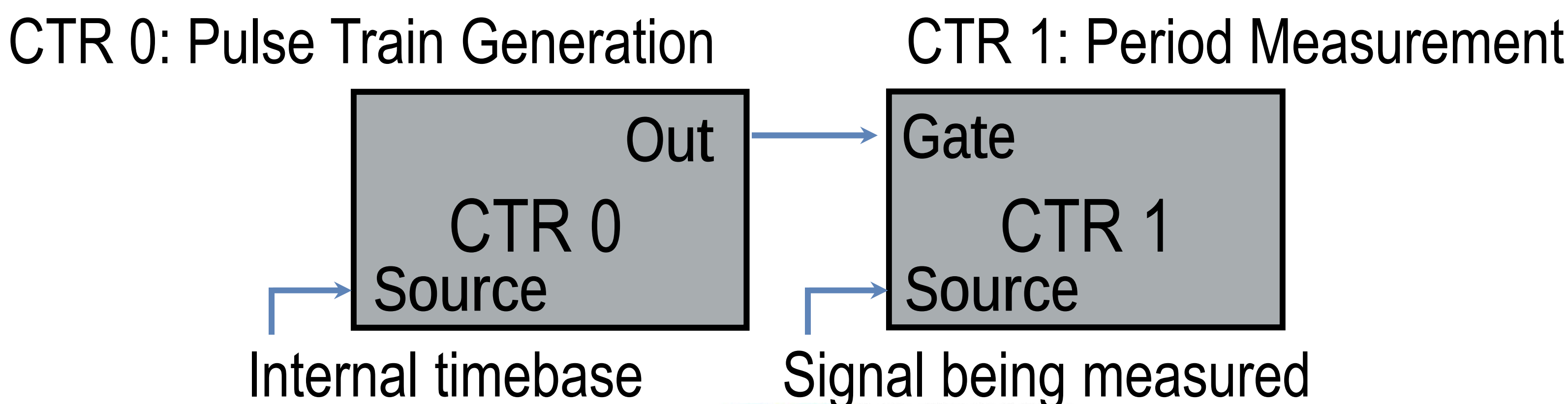
Frequency Measurement – Low Frequency with 1 Counter Method

- Specify the counter input frequency measurement as the application with "Low Frequency with 1 Counter" selected.
- This is an example of a non-buffered, single measurement



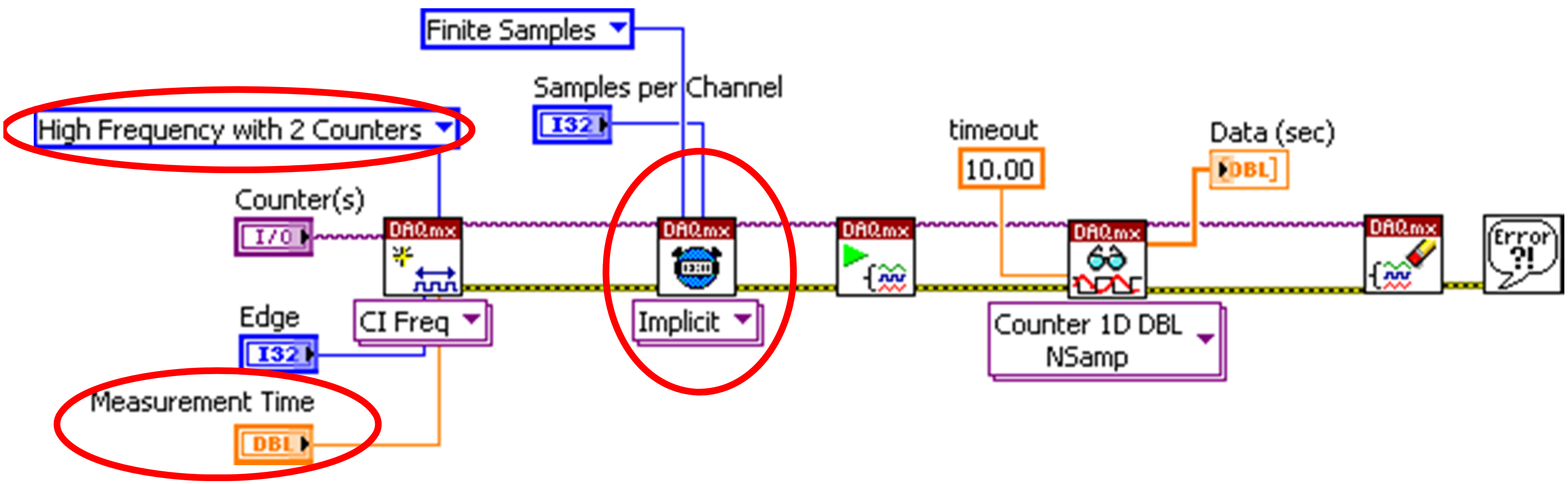
Frequency Measurement – High Frequency with 2 Counters Method

- Count the number of edges of the unknown signal during a known period of time (measurement time)
- The longer the period of the known pulse, the less significant the quantization error
- Unknown frequency is calculated by dividing the number of pulses by the known period



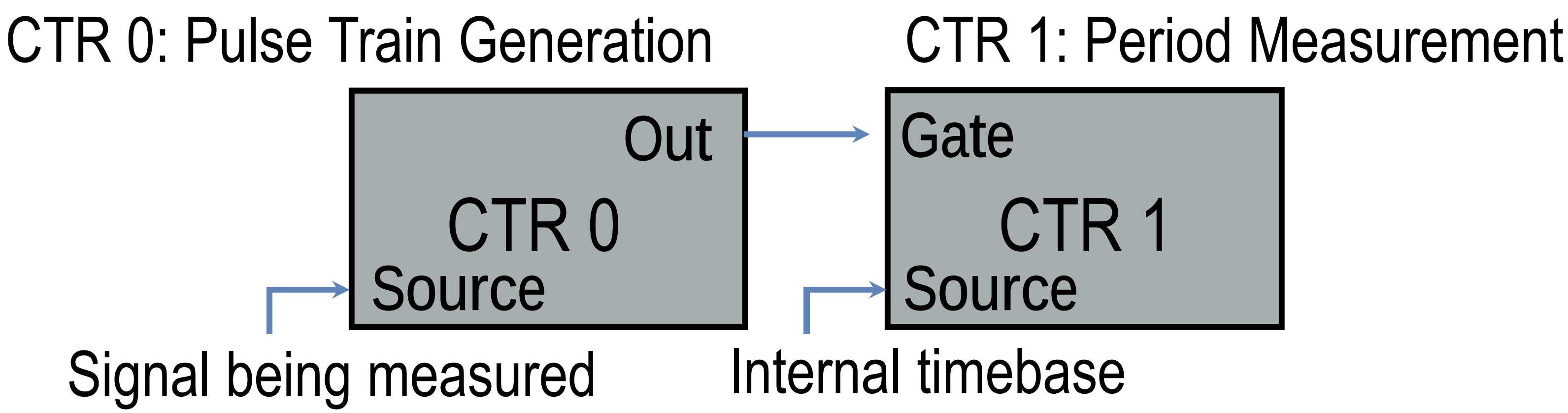
Frequency Measurement – High Frequency with 2 Counters Method

- Specify the counter input frequency measurement as the application with “High Frequency with 2 Counters” selected
- Measurement Time specifies length of time for the measurement



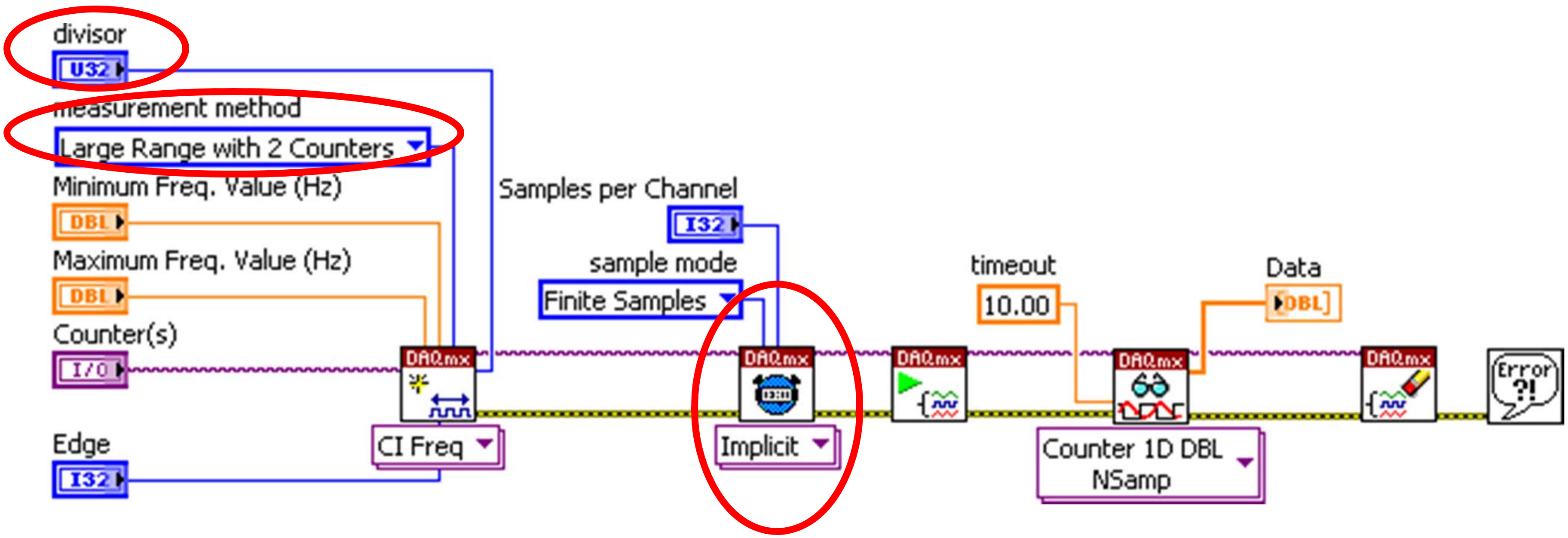
Frequency Measurement – Large Range with 2 Counters Method

- Divide the unknown signal down to a slower frequency
- Perform period measurement on that slower frequency
- Multiply by divisor value to obtain the correct frequency
- The larger the divisor, the slower it takes to perform period measurement and the more accurate the measurement



Frequency Measurement – Large Range with 2 Counters Method

- Specify the counter input frequency measurement as the application with “Large Range with 2 Counters” selected
- Divisor specifies value to divide the input signal by an integer

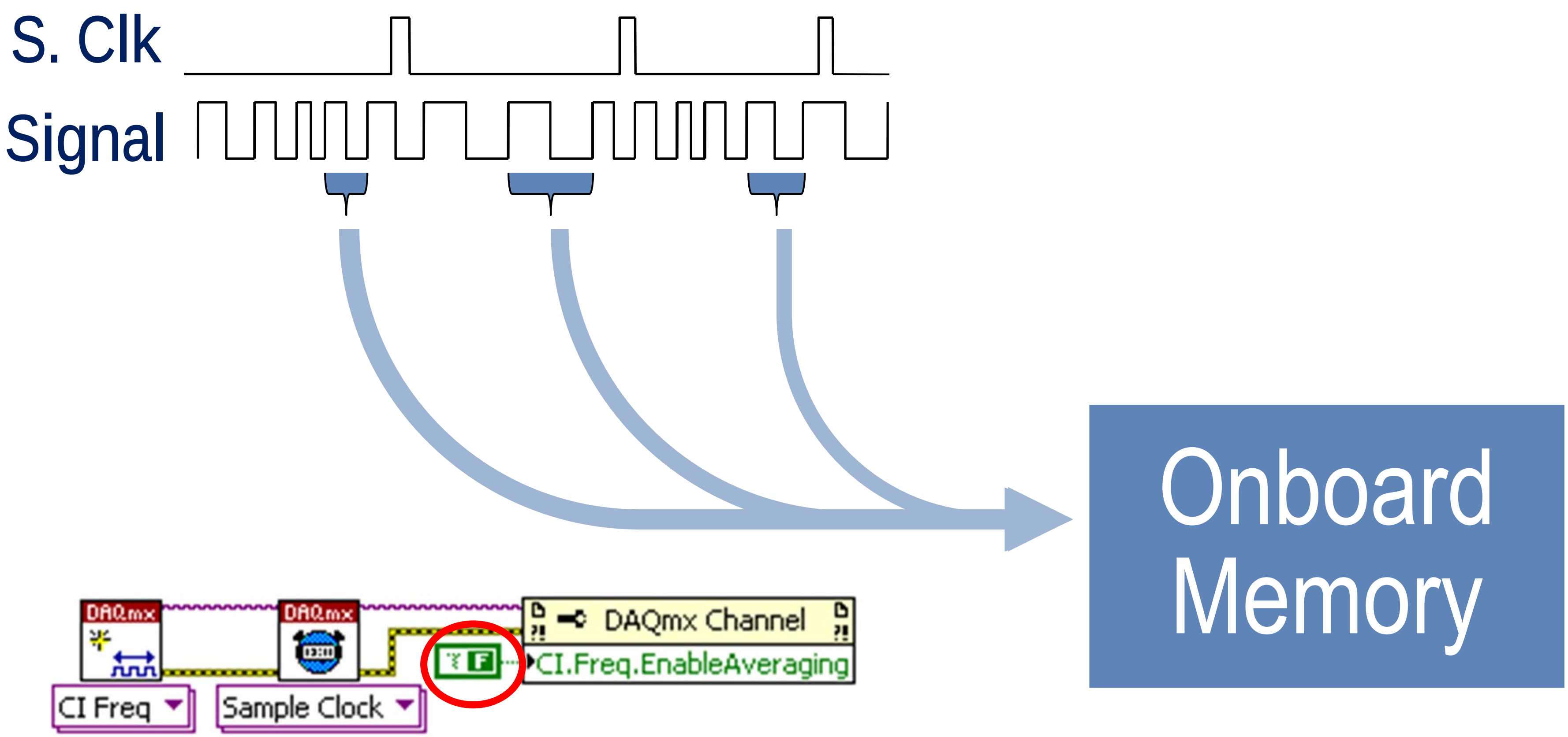


Frequency Measurements – STC3 Chip

DAQ devices using NI-STC3 chip have two additional ways to measure frequency

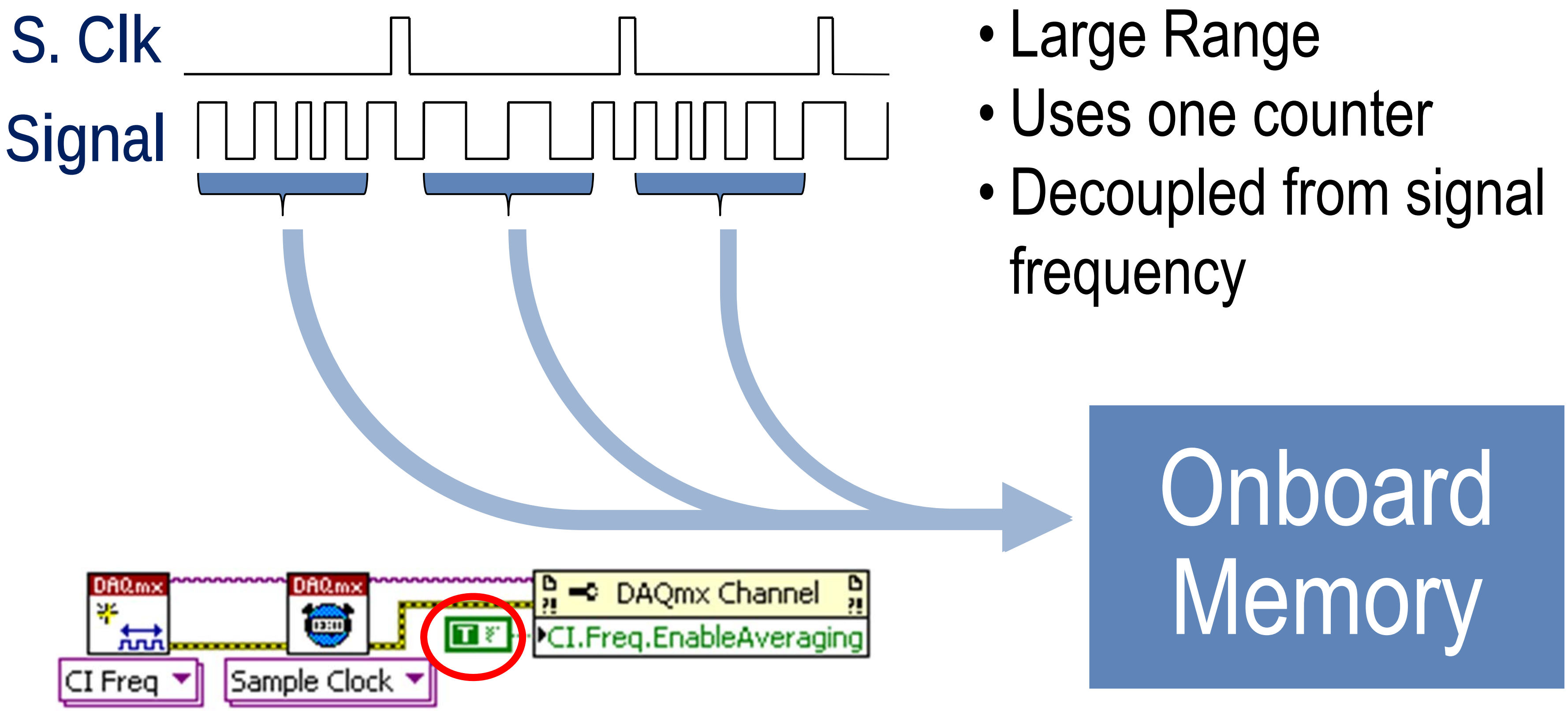
Sample Clock timing (averaging disabled)	Sample Clock timing (averaging enabled)
Good for low frequencies	Good for high frequencies and varying range of frequencies
Uses 1 counter	Uses 1 counter

Sample Clock Timing (Averaging Disabled) with STC3



Sample Clock Timing (Averaging Enabled) with STC3

- High resolution
- Large Range
- Uses one counter
- Decoupled from signal frequency



Exercise 6-7: Frequency Measurement

To build a VI to measures frequency using a counter.

GOAL

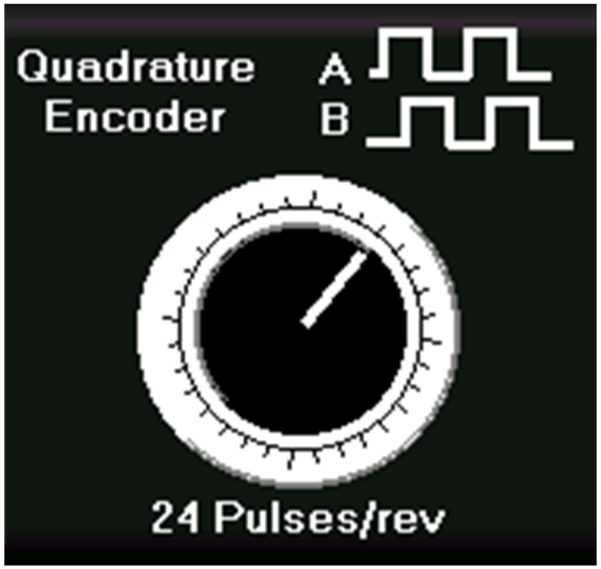
Exercise 6-7: Frequency Measurement

- Can you use the frequency measurement methods discussed for period measurements? (i.e. Low Frequency with 1 Counter, High Frequency with 2 Counters, etc)

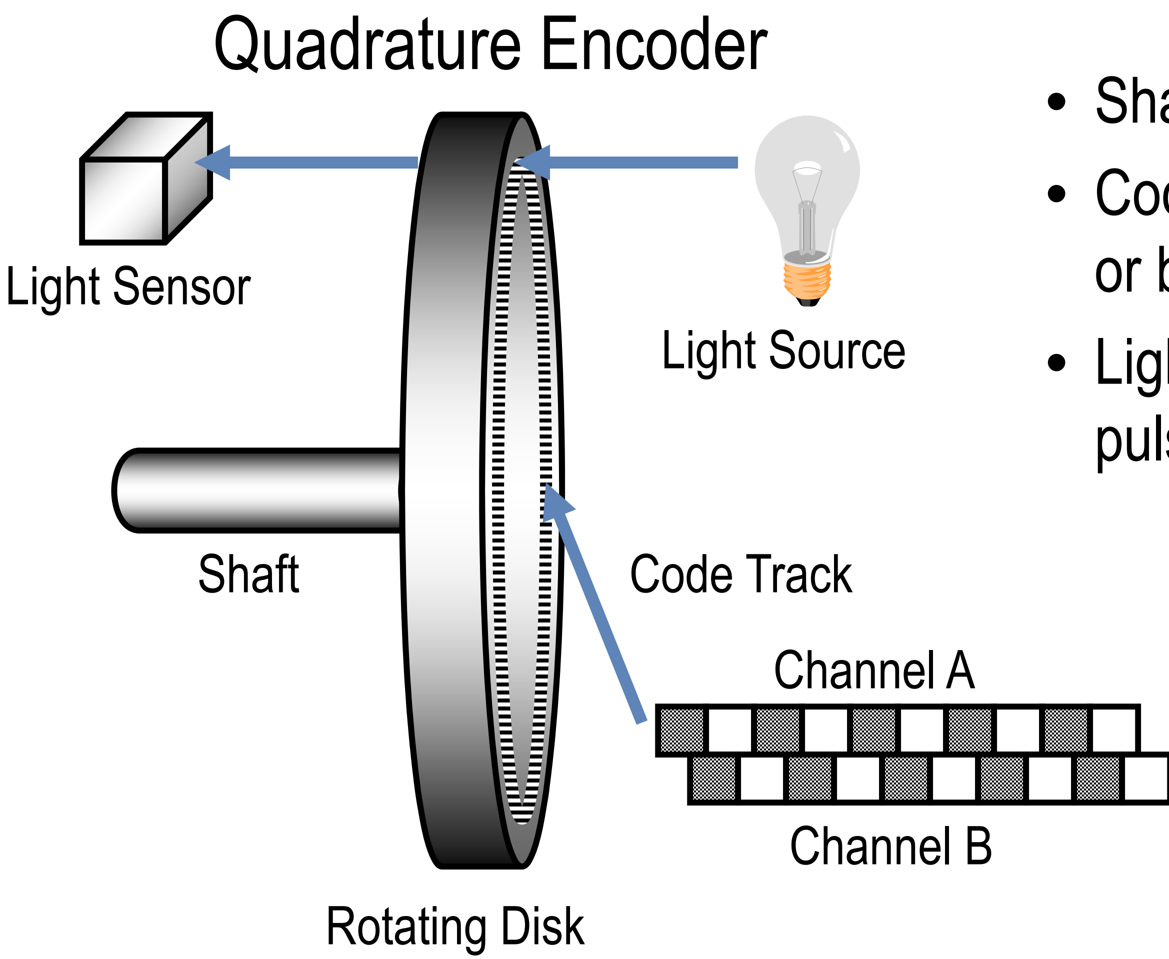
DISCUSSION

F. Position Measurement

- You can measure position with a quadrature encoder
- BNC-2120 has a quadrature encoder
- NI-TIO, NI-STC2, and NI-STC3 chips directly support quadrature encoders

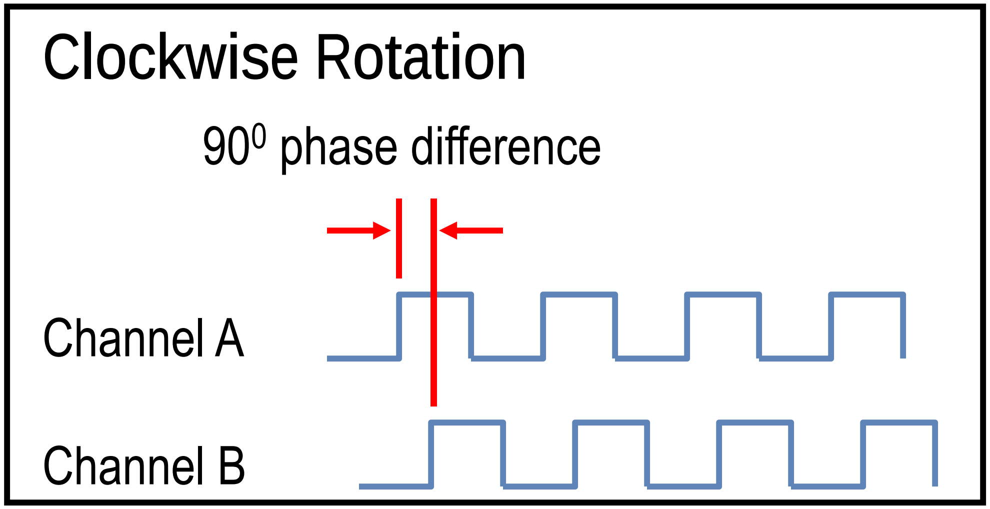


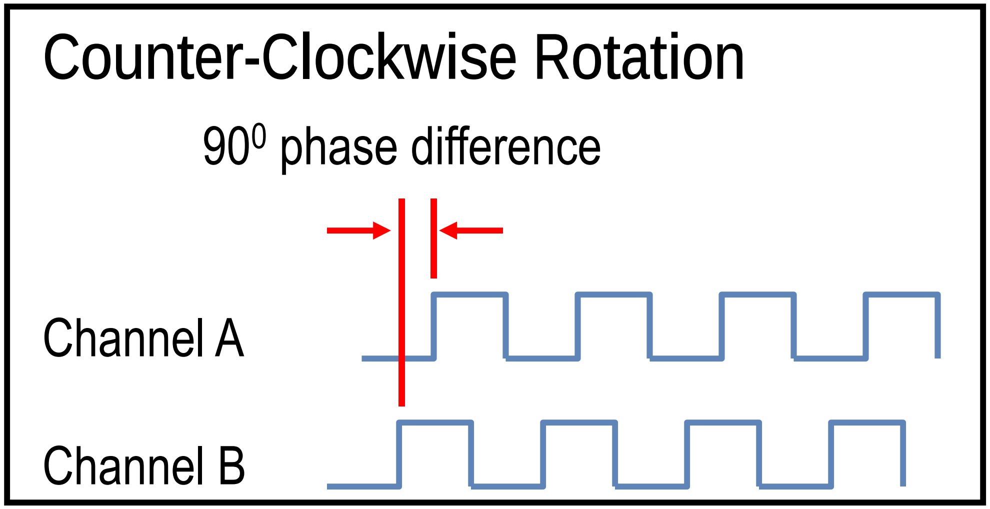
How Does an Encoder Work?



- Shaft and disk rotate
- Code track either passes or blocks light to sensor
- Light sensor creates two pulse trains

Quadrature Encoder





Quadrature encoders produce two pulse trains 90 degrees out of phase

Clockwise rotation

- Channel A leads Channel B

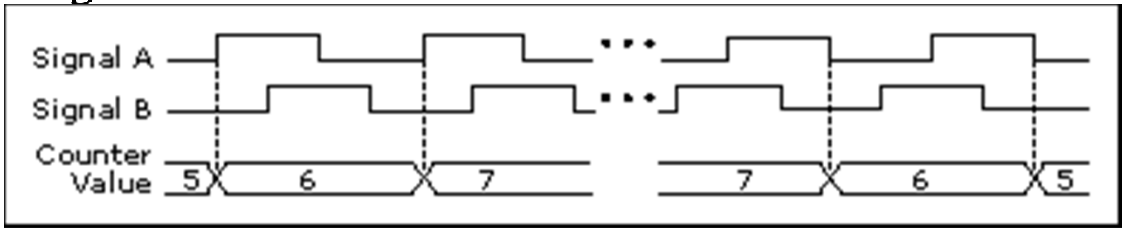
Counter-Clockwise rotation

- Channel B leads Channel A

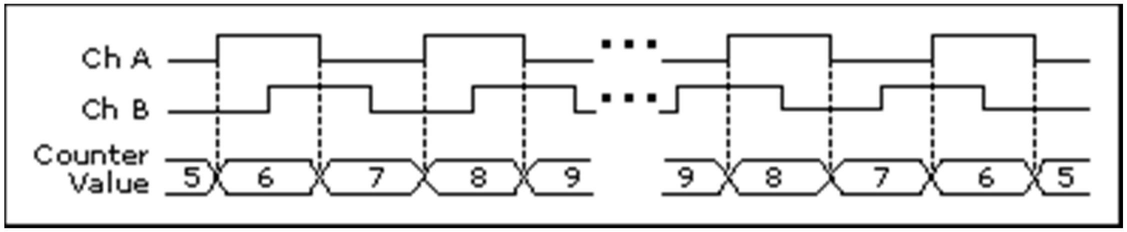
Quadrature Encoder – Decoding

Types of decoding

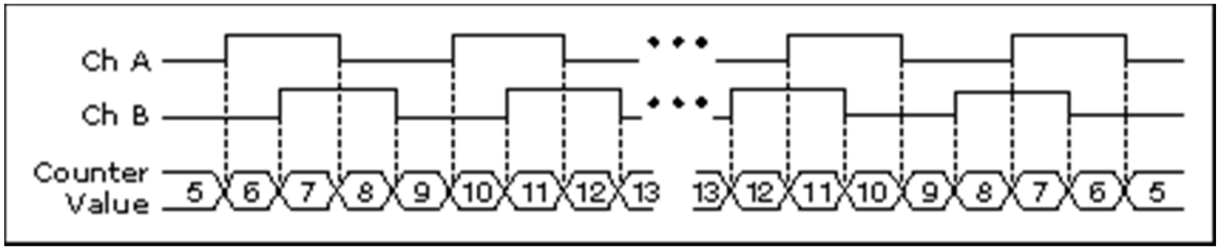
- X1 – Counter increments on rising edge of A if A leads and decrements on falling edge of A if B leads



- X2 – Counter increments/decrements on rising and falling edges of A depending if A or B leads



- X4 – Counter increments/decrements on rising and falling edges of both A and B depending if A or B leads



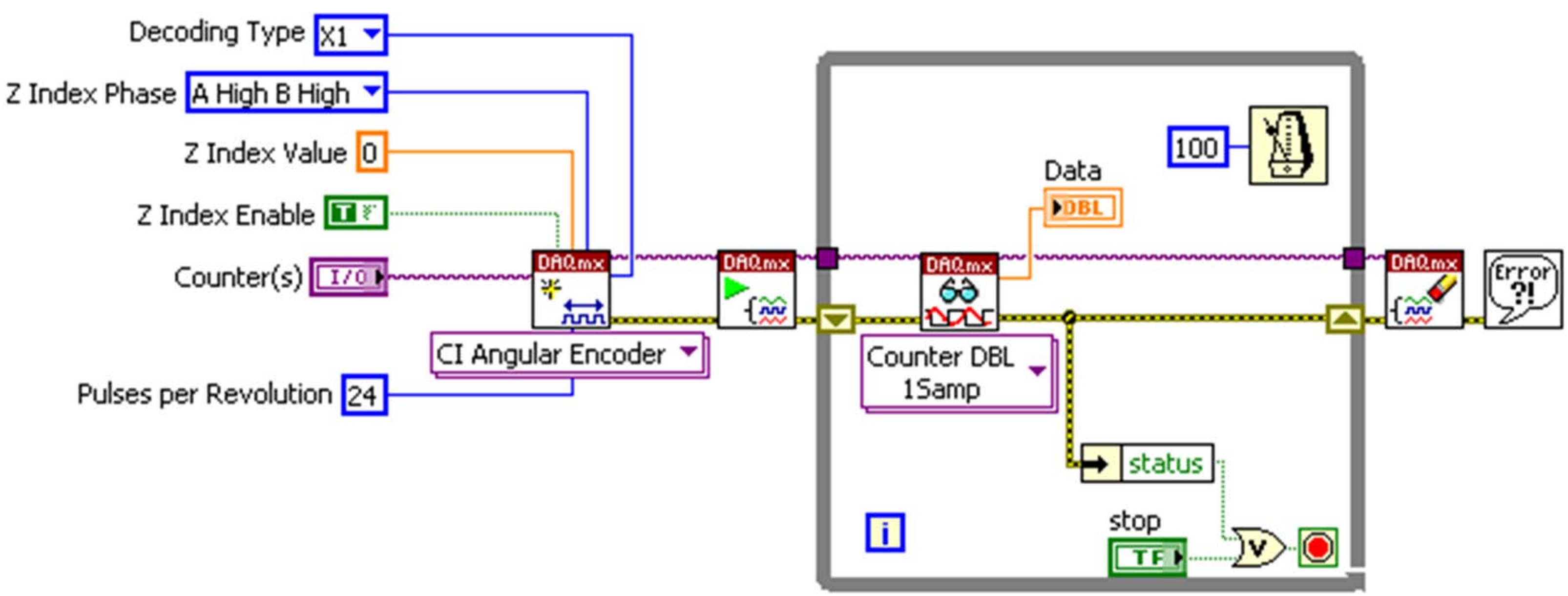
Quadrature Encoder – Z Indexing

Many encoders use a third signal for Z indexing

- Produces a pulse at fixed positions
  - For example, at 0 degrees or 45 degrees on an angular encoder
  - Refer to the documentation for an encoder to obtain the timing of signal Z in relation to the A and B signals
- Use for precise determination of a reference position

Quadrature Encoder Example

Use the **Counter Input»Angular Encoder** or **Counter Input»Linear Encoder** instance of the DAQmx Create Virtual Channel VI



Sample Clock Timing with NI-STC3

DAQ devices with NI-STC3 chip can use sample clock timing with the following measurements

- Pulse width measurement
- Pulse measurement
- Frequency and period measurements
- Position measurement
- Two-signal separation measurement

Summary—Quiz

1. Which of the following are components of a counter?
   a) Source
   b) Gate
   c) Multiplexer
   d) Register
   e) Output

Summary—Quiz Answer

1. Which of the following are components of a counter?
   a) Source
   b) Gate
   c) Multiplexer
   d) Register
   e) Output

Summary—Quiz

2. What is the terminal count of a 24-bit counter?

Summary—Quiz Answer

2. What is the terminal count of a 24-bit counter?

   $2^{24} - 1 = 16777216 - 1 = 16,777,215$

Summary—Quiz

3. What error occurs when the frequency being measured by the Low Frequency with 1 Counter method approaches the timebase of the DAQ device?

Summary—Quiz Answer

3. What error occurs when the frequency being measured by the Low Frequency with 1 Counter method approaches the timebase of the DAQ device?

   Quantization Error