LXI IVI Programming Model for Synchronization and Triggering

Lynn Wheelwright – lynnw@sonic.net

September 27, 2005

LAN eXtensions for Instrumentation

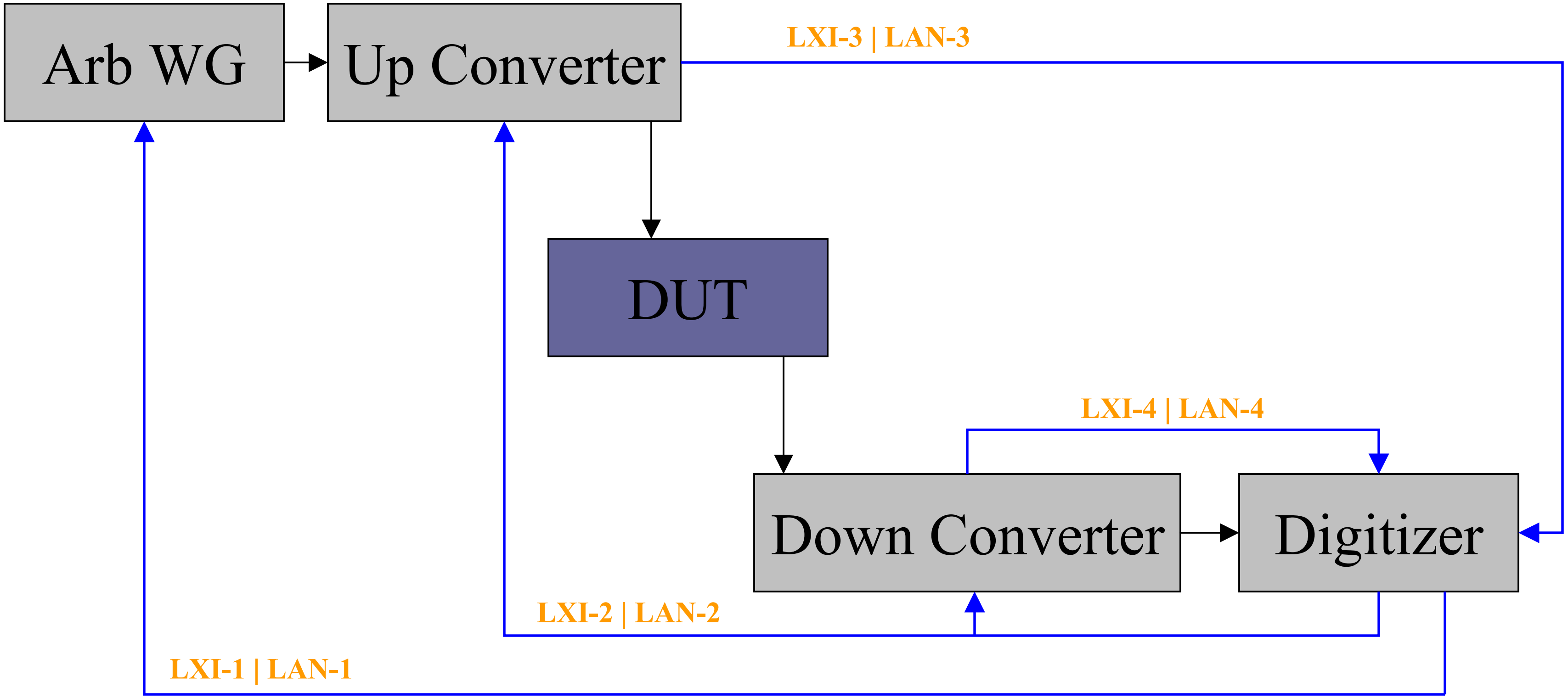
Topics

- Introduction
- LXI Trigger Use Case: Stimulus-Response Measurement
- Trigger Logic Top Level View
- Digitizer Arm-Trigger Model
- Arm Logic
- Trigger Logic
- Event Logic
- Event Sources & Destinations
- Sample Code

LXI Features

- Hardwired Trigger Bus
  - 8 Lanes wide, multi-drop LVDS signaling.
  - Wired-or capable on an as-needed basis
- Ethernet signaling protocol that provides a logical dual to the hardwired trigger bus.
- Very accurate time base synchronization using IEEE 1588
  - Time synchronized triggering
  - Time-stamped data.

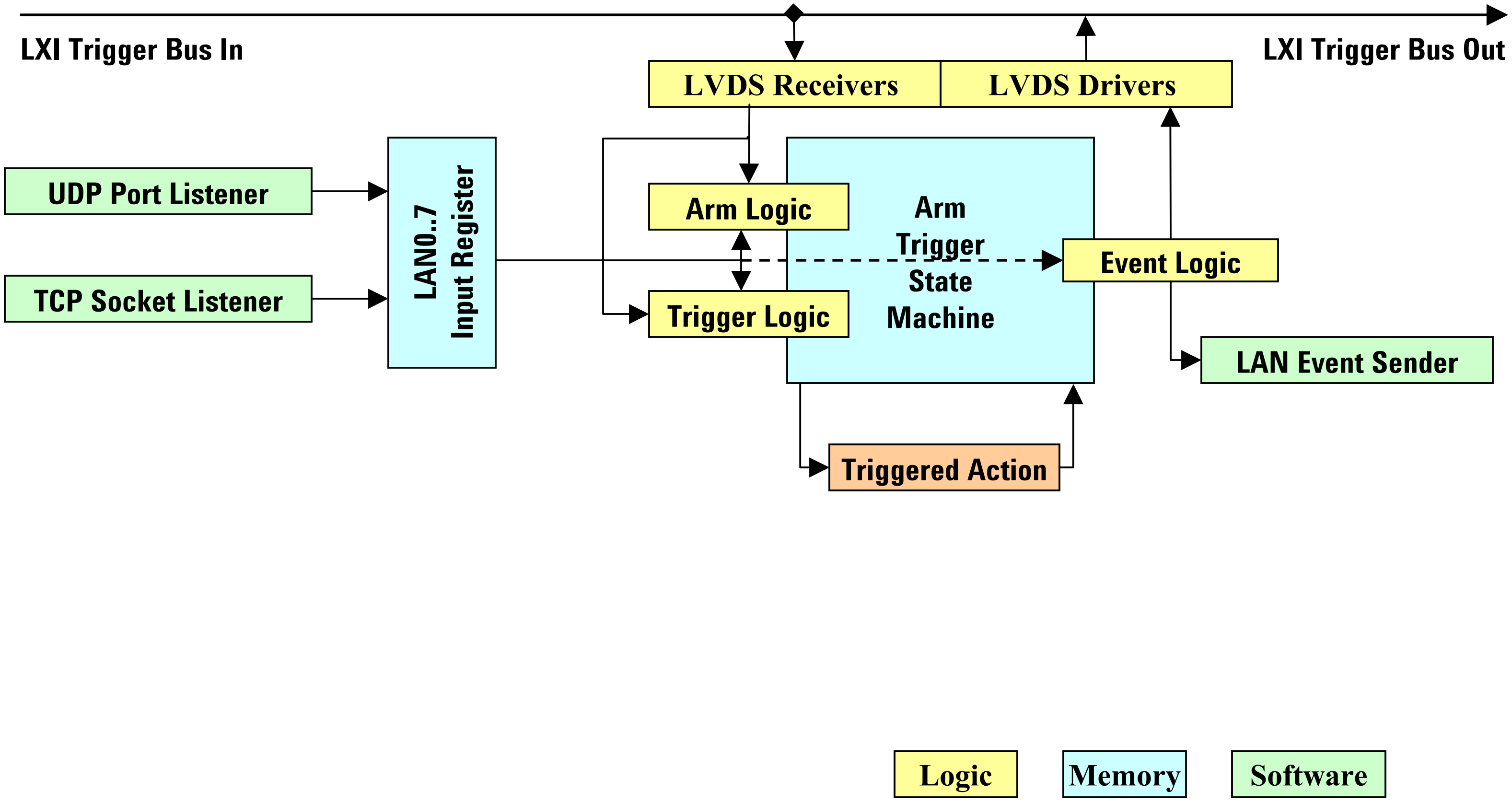
LXI Trigger Use Case: Stimulus Response Test Block Diagram



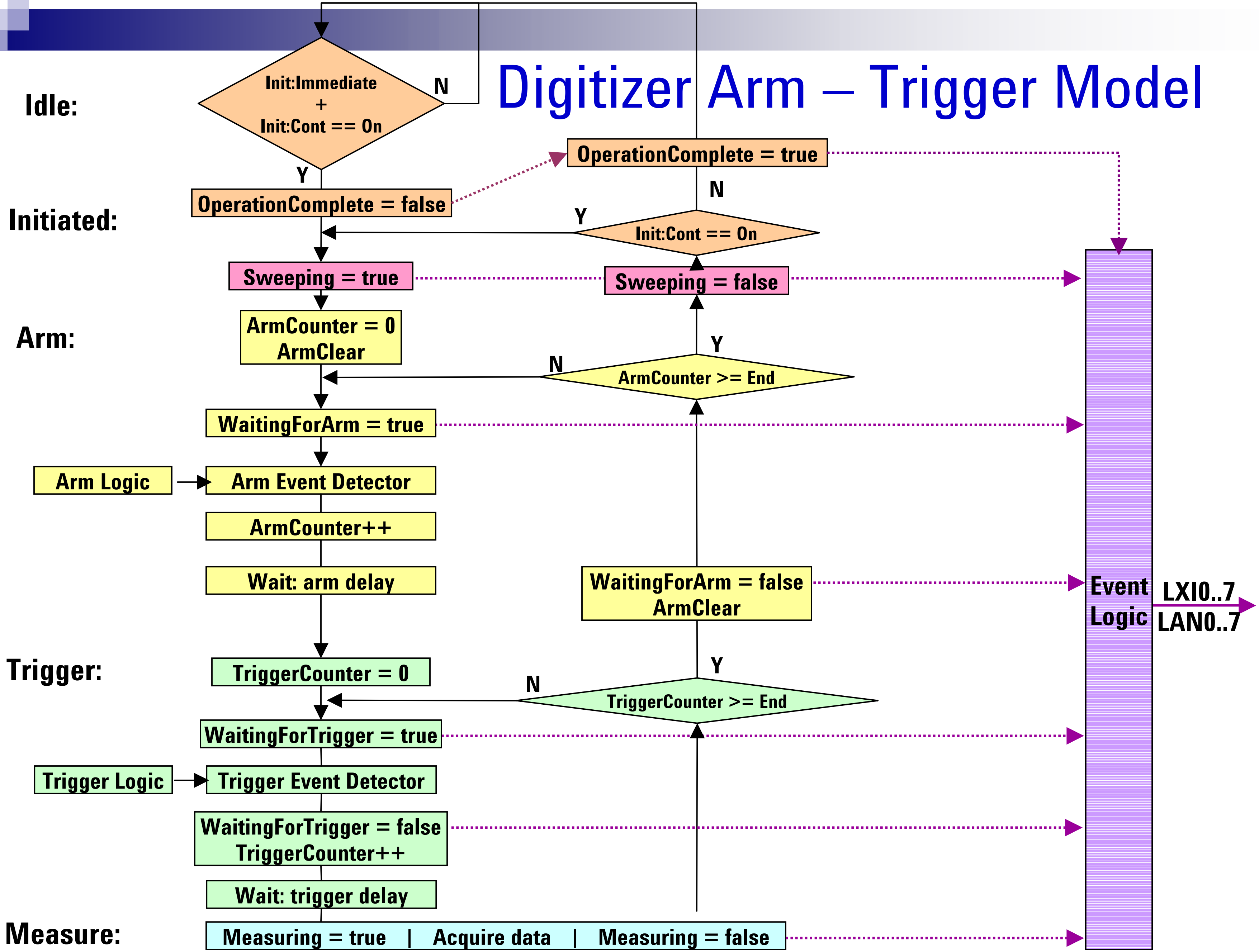
LXI Trigger Use Case: Stimulus Response Test

- Device Under Test will be stimulated with a pulse waveform at frequencies across its operating range to verify its distortion properties.
- The stimulus will consist of a descending series of steps, each step 2db and 550usec.
- The waveform will be up-converted to microwave frequencies using an up-converter module.
- The receiver will use a down-converter to translate the signal into the range of a digitizer module.
- The up and down converters are capable of frequency step sweep operations (Start Frequency, Stop Frequency, Step Frequency) and can initiate each step based on a trigger signal.
- At each frequency, the digitizer shall acquire the waveform of interest so that it can be returned to the PC for further processing. The digitizer will use an internal signal level trigger to acquire the signal, it shall also wait for the up-converter and down-converter at each step before entering the Waiting-For-Trigger state.
- This is a production line scenario, speed of test is of the essence.

Trigger Logic: Top Level View



Digitizer Arm – Trigger Model



Idle:

Init:Immediate + Init:Cont == On

N

Y

Initiated:

OperationComplete = false

OperationComplete = true

N

Y

Init:Cont == On

Sweeping = true

Sweeping = false

Arm:

ArmCounter = 0
ArmClear

Y

N

ArmCounter >= End

WaitingForArm = true

Arm Logic

Arm Event Detector

ArmCounter++

Wait: arm delay

WaitingForArm = false
ArmClear

Event Logic

LXI0..7

LAN0..7

Trigger:

TriggerCounter = 0

Y

N

TriggerCounter >= End

WaitingForTrigger = true

Trigger Logic

Trigger Event Detector

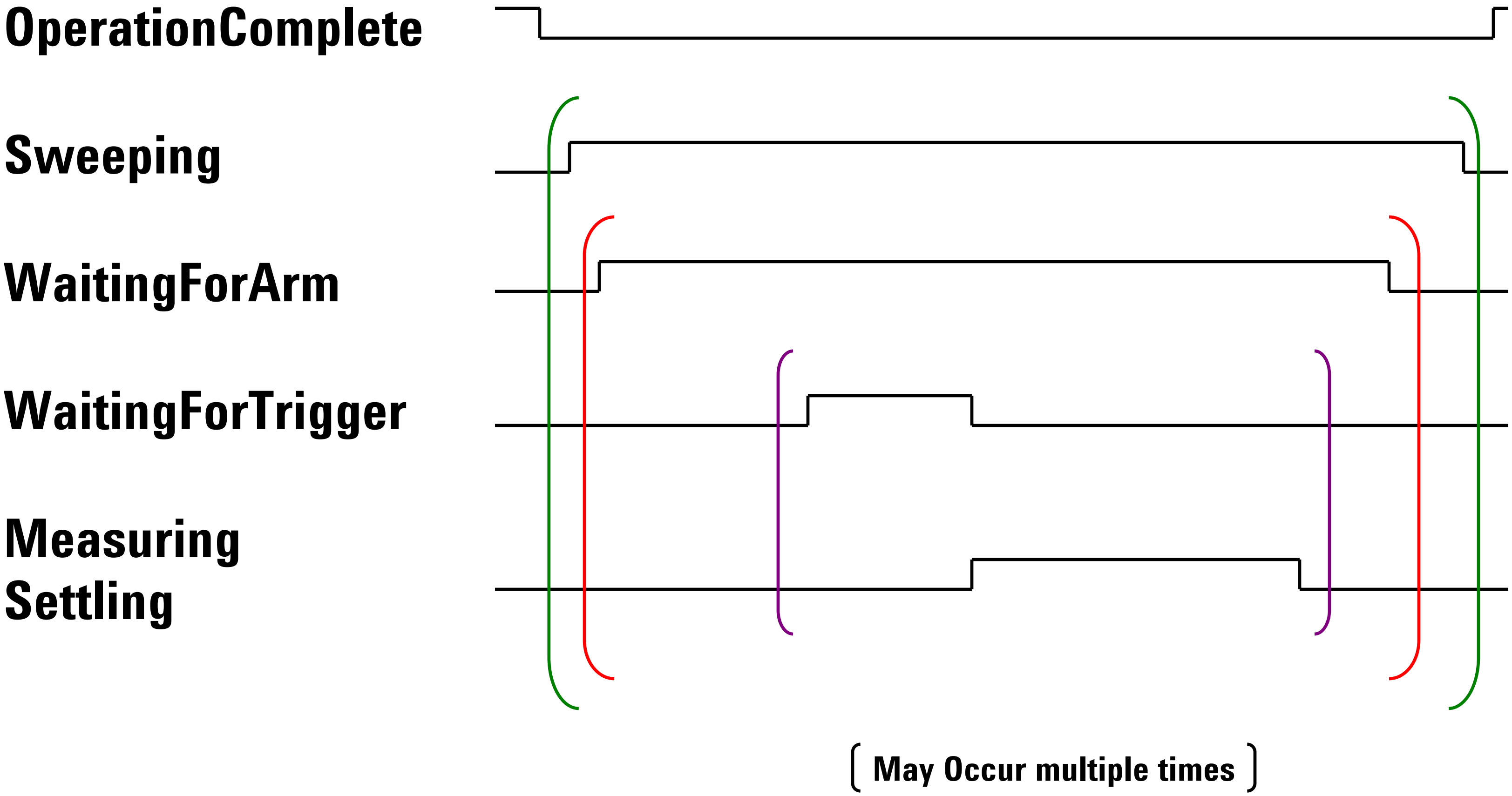
WaitingForTrigger = false
TriggerCounter++

Wait: trigger delay

Measure:

Measuring = true | Acquire data | Measuring = false

Arm-Trigger State Machine Signal Relationships



OperationComplete

Sweeping

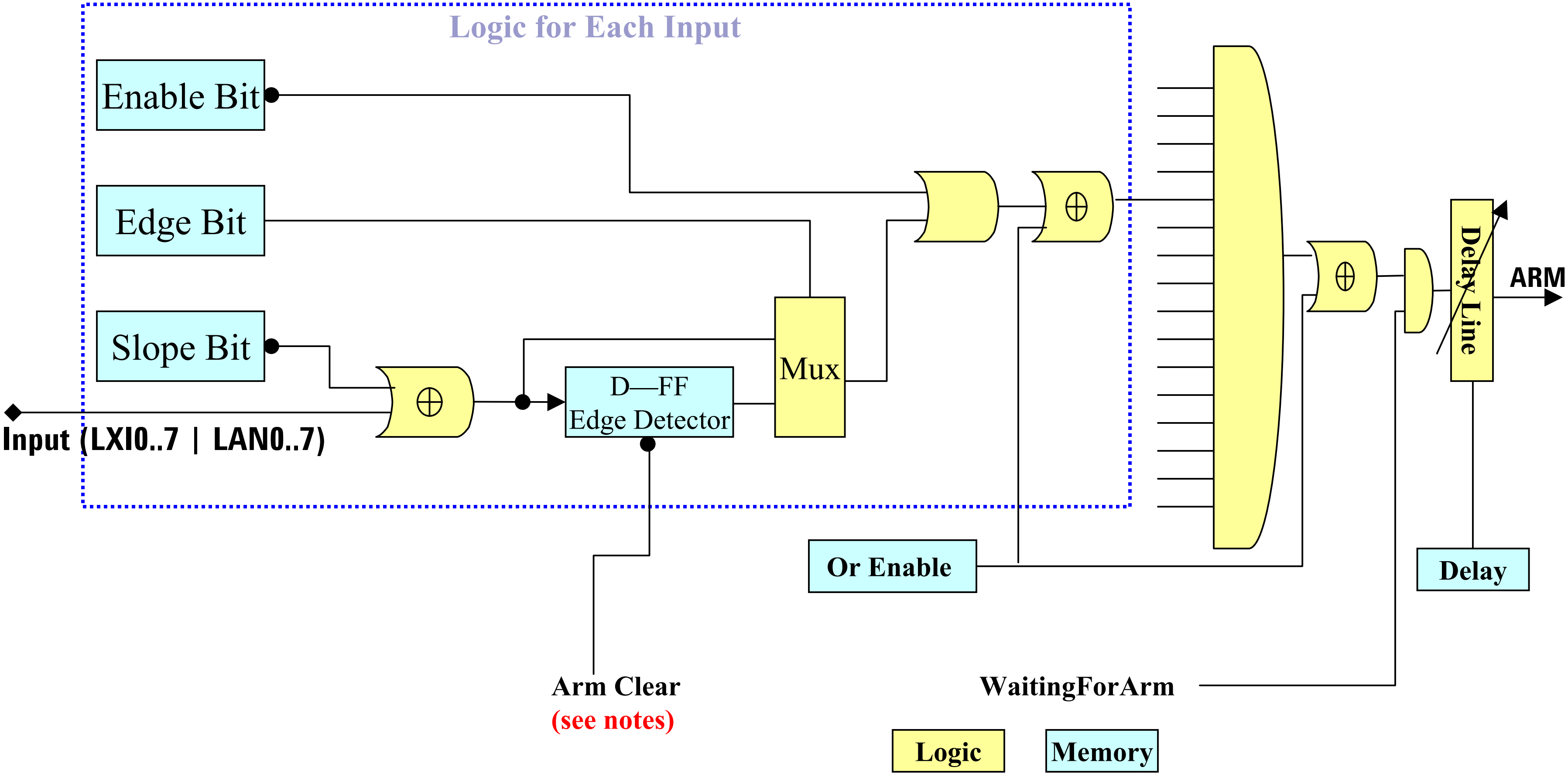
WaitingForArm

WaitingForTrigger

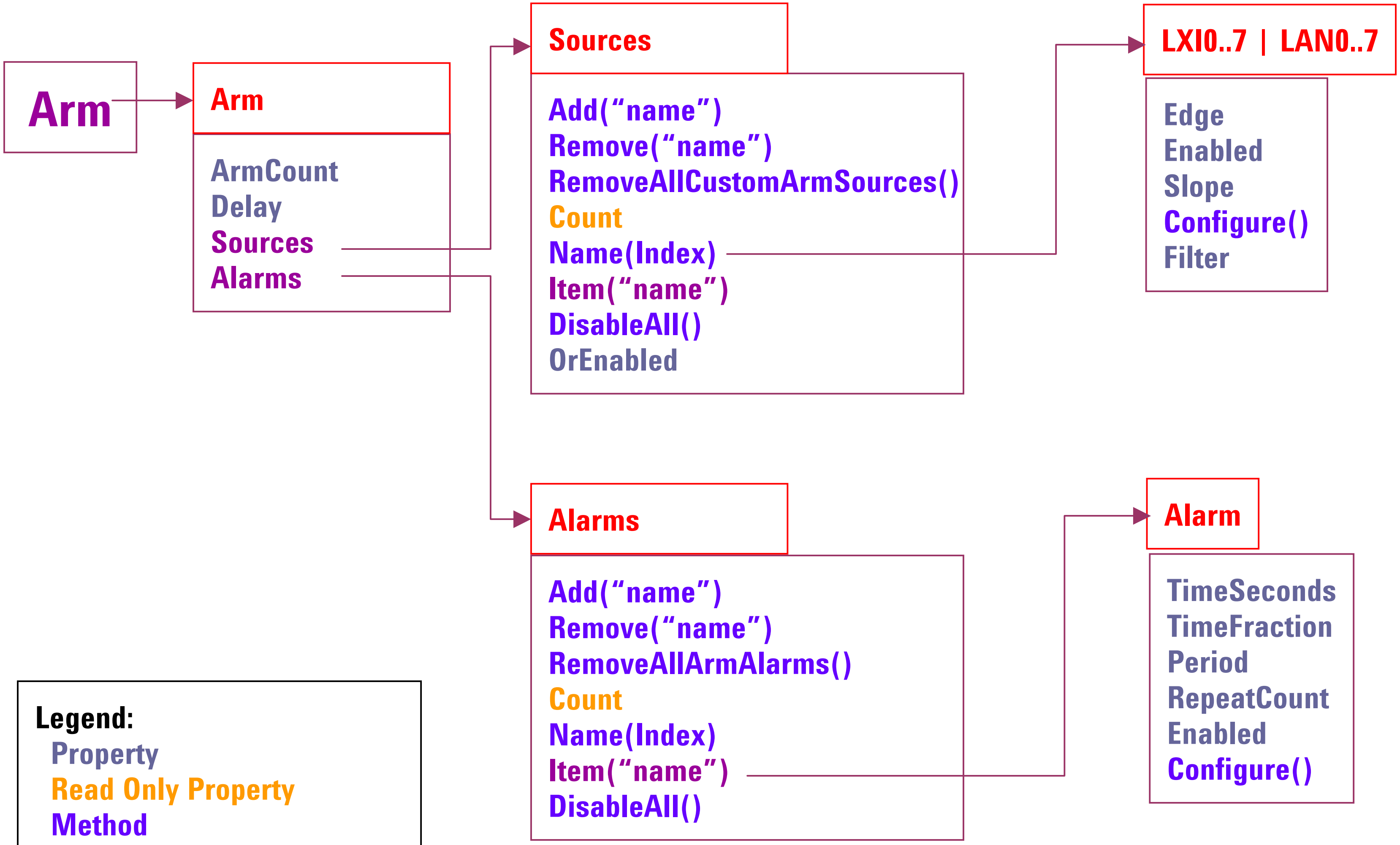
Measuring
Settling

May Occur multiple times

Arm Logic: LXI0..7 & LAN0..7



LXI IVI-COM Arm Interface



Arm

Arm

ArmCount
Delay
Sources
Alarms

Sources

Add("name")
Remove("name")
RemoveAllCustomArmSources()
Count
Name(Index)
Item("name")
DisableAll()
OrEnabled

LXI0..7 | LAN0..7

Edge
Enabled
Slope
Configure()
Filter

Alarms

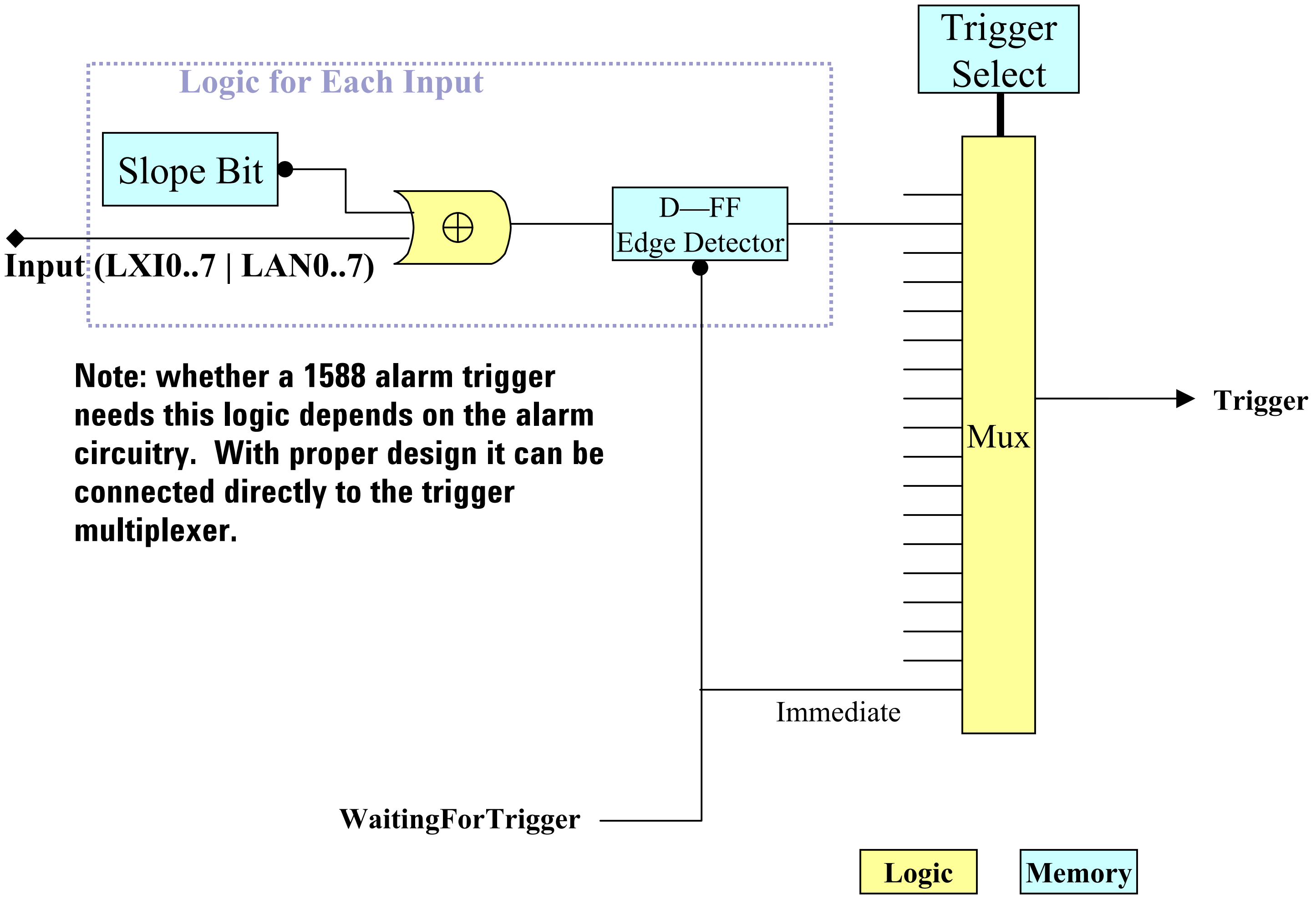
Add("name")
Remove("name")
RemoveAllArmAlarms()
Count
Name(Index)
Item("name")
DisableAll()

Alarm

TimeSeconds
TimeFraction
Period
RepeatCount
Enabled
Configure()

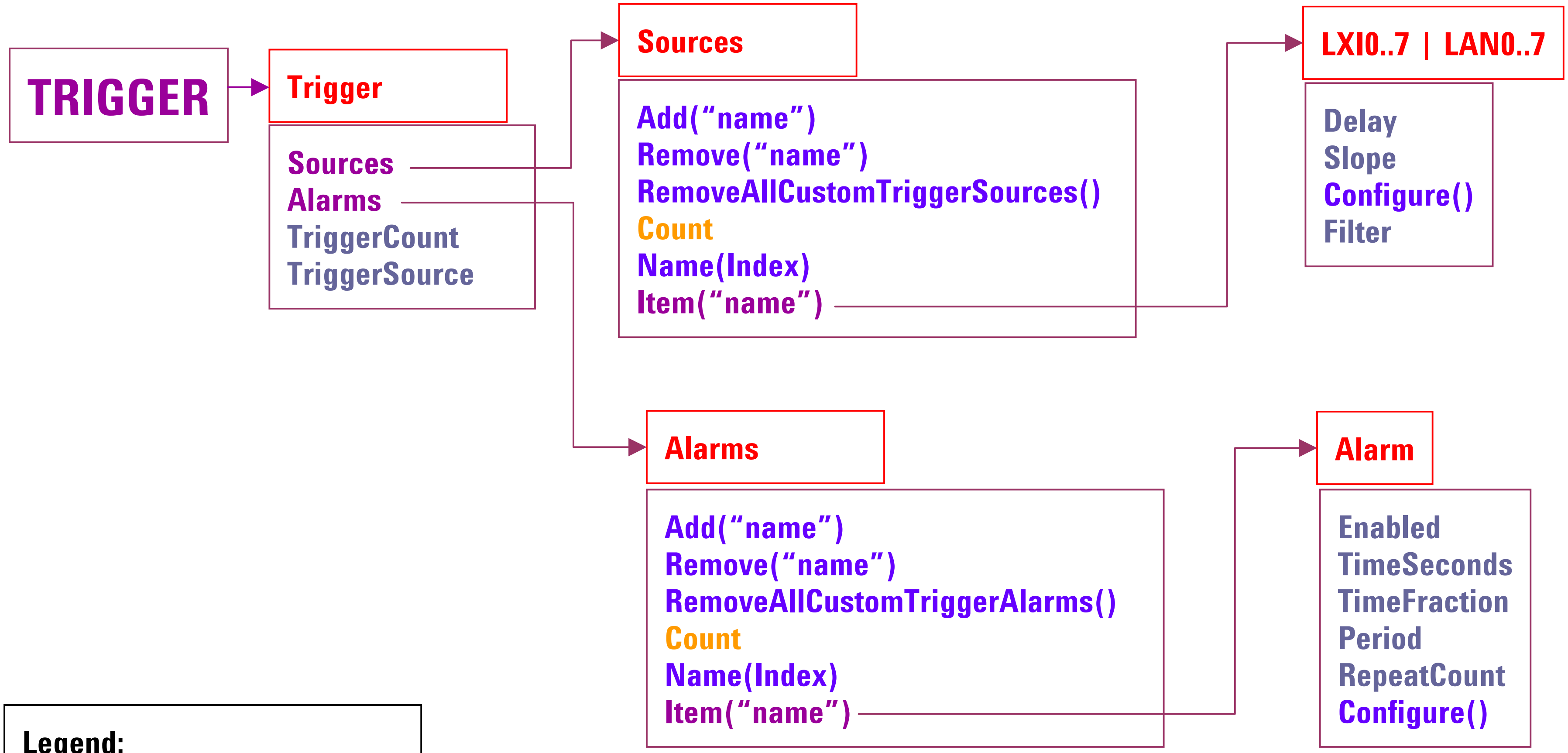
Legend:
- Property
- Read Only Property
- Method
- Interface Pointer (Property)
- Interface

Trigger In Logic



Note: whether a 1588 alarm trigger needs this logic depends on the alarm circuitry. With proper design it can be connected directly to the trigger multiplexer.

LXI IVI-COM Trigger Interface



TRIGGER

Trigger

Sources
Alarms
TriggerCount
TriggerSource

Sources

Add(“name”)
Remove(“name”)
RemoveAllCustomTriggerSources()
Count
Name(Index)
Item(“name”)

LXI0..7 | LAN0..7

Delay
Slope
Configure()
Filter

Alarms

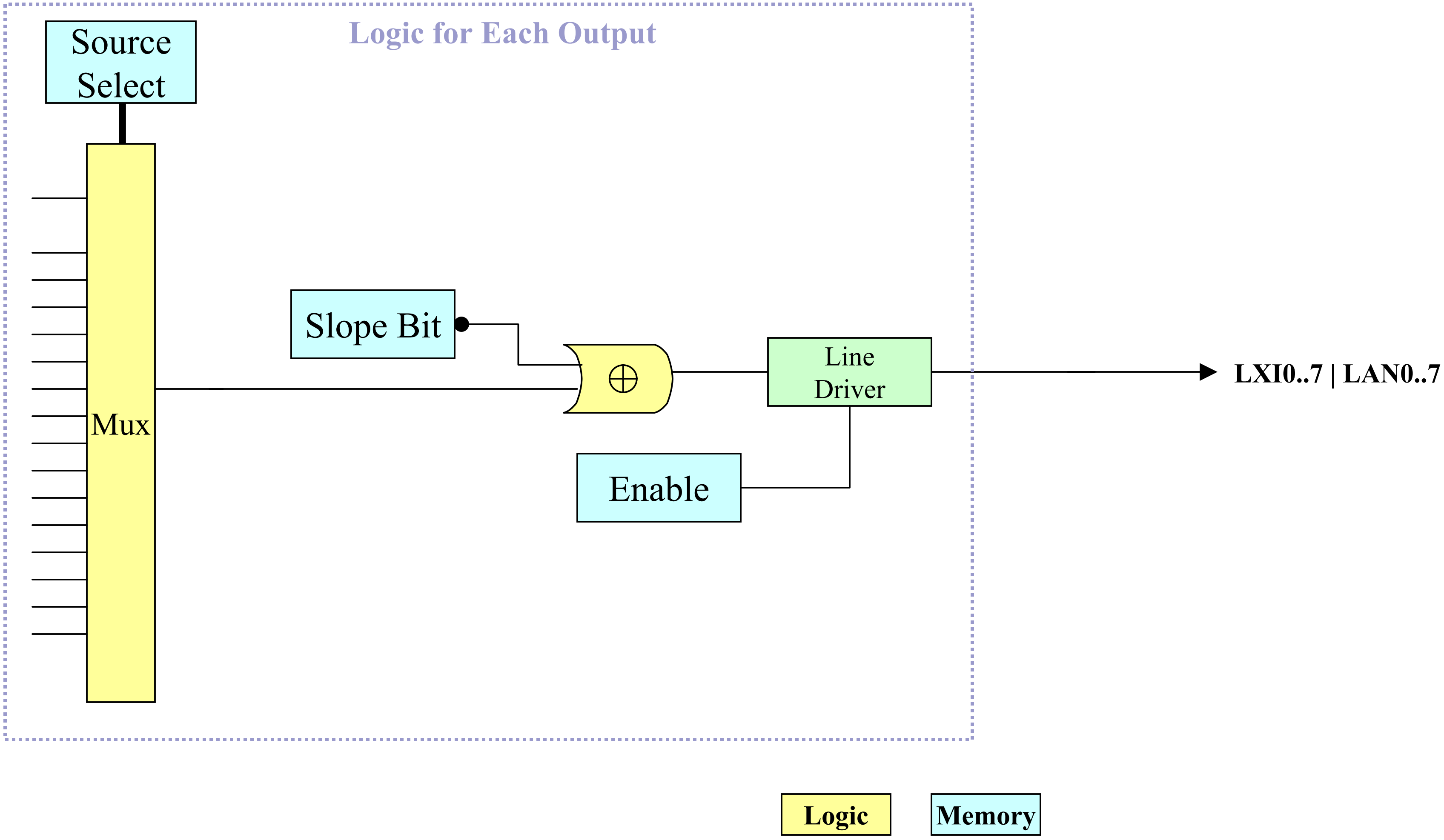
Add(“name”)
Remove(“name”)
RemoveAllCustomTriggerAlarms()
Count
Name(Index)
Item(“name”)

Alarm

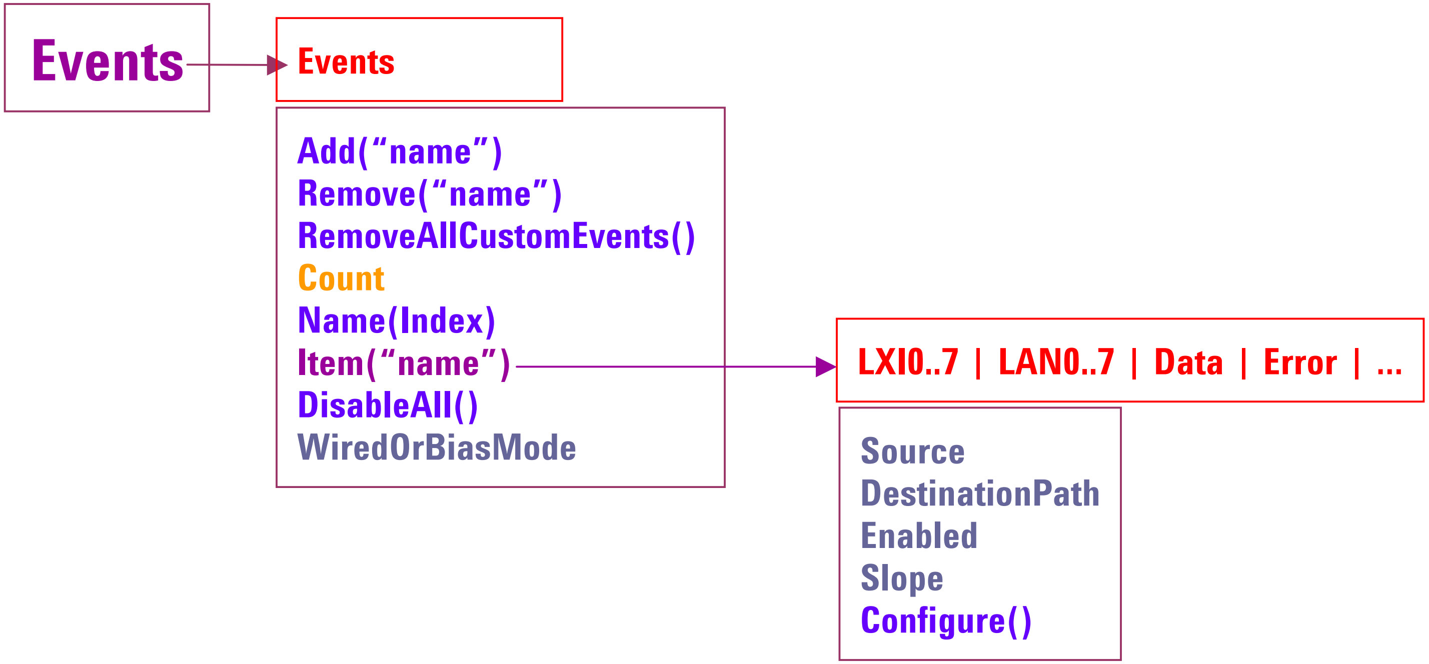
Enabled
TimeSeconds
TimeFraction
Period
RepeatCount
Configure()

Legend:
Property
Read Only Property
Method
Interface Pointer (Property)
Interface

Event Out (Trigger Out) Logic



LXI IVI-COM Synchronization Event Interface



Events

Events

Add("name")
Remove("name")
RemoveAllCustomEvents()
Count
Name(Index)
Item("name")
DisableAll()
WiredOrBiasMode

LXI0..7 | LAN0..7 | Data | Error | ...

Source
DestinationPath
Enabled
Slope
Configure()

Legend:

Property
Read Only Property
Method
Interface Pointer (Property)
Interface

Event

Sources

- 1588 Clock (see notes)
- Calibration
- Settling (red = required)
- Ranging
- Sweeping
- Measuring
- Waiting For Arm
- Waiting For Trigger
- Correcting
- Operation Complete
- Error
- Data
- LXI0..7 (see notes)
- LAN0..7 (see notes)

Destinations

- LXI Trigger Line 0..7
- LAN Trigger 0..7 on host:port
- Error on host:port
- Data on host:port

- Destination Path Syntax:
  [host[:port]][/name][, Destination Path]

  If no host is specified, use local module.
  If host is 'ALL', send a UDP broadcast.
  If no port is specified, use IANA default.
  If no name is specified, use Event Name (LXI0..7 | LAN0..7 | Data |..).
  [optional item]

Sample Code: Hardwired Trigger Bus Setup

```
// tell the digitizer to output WaitingForTrigger on LXI1—used to trigger Arb
Digitizer.Events.Item(“LXI1”).Configure(WaitingForTrigger,””,On,Positive);
Arb.Trigger.Item(“LXI1”).Configure(0,Positive);
Arb.Trigger.TriggerSource = “LXI1”;

// tell digitizer to output WaitingForArm falling edge as trigger for up and down converters
Digitizer.Events.Item(“LXI2”).Configure(WaitingForArm,””,On,Negative);
UpConverter.Trigger.Item(“LXI2”).Configure(0,Positive);
UpConverter.Trigger.TriggerSource = “LXI2”;
DownConverter.Trigger.Item(“LXI2”).Configure(0,Positive);
DownConverter.Trigger.TriggerSource = “LXI2”;

// tell up-converter to output Settling on LXI3 for use by digitizer
UpConverter.Events.Item(“LXI3”).Configure(Settling,””,On,Negative);
Digitizer.Arm.Sources.Item(“LXI3”).Configure(true,true,Positive);

// tell down-converter to output Settling on LXI4 for use by digitizer
DownConverter. Events.Item(“LXI4”).Configure(Settling,””,On,Negative);
Digitizer.Arm.Sources.Item(“LXI4”).Configure(true,true,Positive);
```

Sample Code: LAN Trigger Setup

```
// tell the digitizer to output WaitingForTrigger on LAN1—used to trigger Arb
Digitizer.Events.Item(“LAN1”).Configure(WaitingForTrigger,””,On,Positive);
Arb.Trigger.Item(“LAN1”).Configure(0,Positive);
Arb.Trigger.TriggerSource = “LAN1”;

// tell digitizer to output WaitingForArm falling edge as trigger for up and down converters
Digitizer.Events.Item(“LAN2”).Configure(WaitingForArm,””,On,Negative);
UpConverter.Trigger.Item(“LAN2”).Configure(0,Positive);
UpConverter.Trigger.TriggerSource = “LAN2”;
DownConverter.Trigger.Item(“LAN2”).Configure(0,Positive);
DownConverter.Trigger.TriggerSource = “LAN2”;

// tell up-converter to output Settling on LAN3 for use by digitizer
UpConverter.Events.Item(“LAN3”).Configure(Settling,””,On,Negative);
Digitizer.Arm.Sources.Item(“LAN3”).Configure(true,true,Positive);

// tell down-converter to output Settling on LAN4 for use by digitizer
DownConverter. Events.Item(“LAN4”).Configure(Settling,””,On,Negative);
Digitizer.Arm.Sources.Item(“LAN4”).Configure(true,true,Positive);
```

Do the Measurement

```
// load signal file into arb and get it ready to go
…
// setup rest of digitizer parameters (acquisition length, SignalLevel trigger, etc.)
…
// start digitizer
Digitizer.Arm.ArmCount = 101;  // acquire 101 signal packets
Digitizer.Initiate();     // digitizer will now wait until it sees Settled from the up and down
converters.

// setup frequencies for up and down converter
UpConverter.Frequency.Sweep.Configure(2e9,3e9,10e6);
DownConverter.Frequency.Sweep.Configure(2e9,3e9,10e6);

// as soon as both converters settle, the digitizer will trigger the arb and take the first
measurement
// loop and read back data
double [,] data = new double[101,10000];
for( int index = 0; index <=100; index++) data[index] = Digitizer.Trace.Fetch();

// Done
```

Summary

- A key element in LXI design is the flexible dual nature of the Hardwired Trigger Bus and the LAN triggers.
- Some Rules of Thumb:
  - □ Use Hardwired Trigger Bus when timing requirements are in the few nanoseconds or less range.
  - □ Use 1588 triggers when hardware is widely dispersed or the the timing requirements are in the few tens of nanoseconds range.
  - □ Use LAN triggers when the timing requirements can tolerate latencies of tens to a few hundred microseconds.