

LXI Programmatic Interface

Joe Mueller (Agilent)

LXI IVI API

Trigger/Arm

- **Provides uniform control of HW and LAN triggers**
- **Set alarms for use as trigger/arm or elsewhere**

Event

- **Allows arbitrary events in the instrument to be transmitted via arbitrary mechanism (TCP/UDP/HW)**

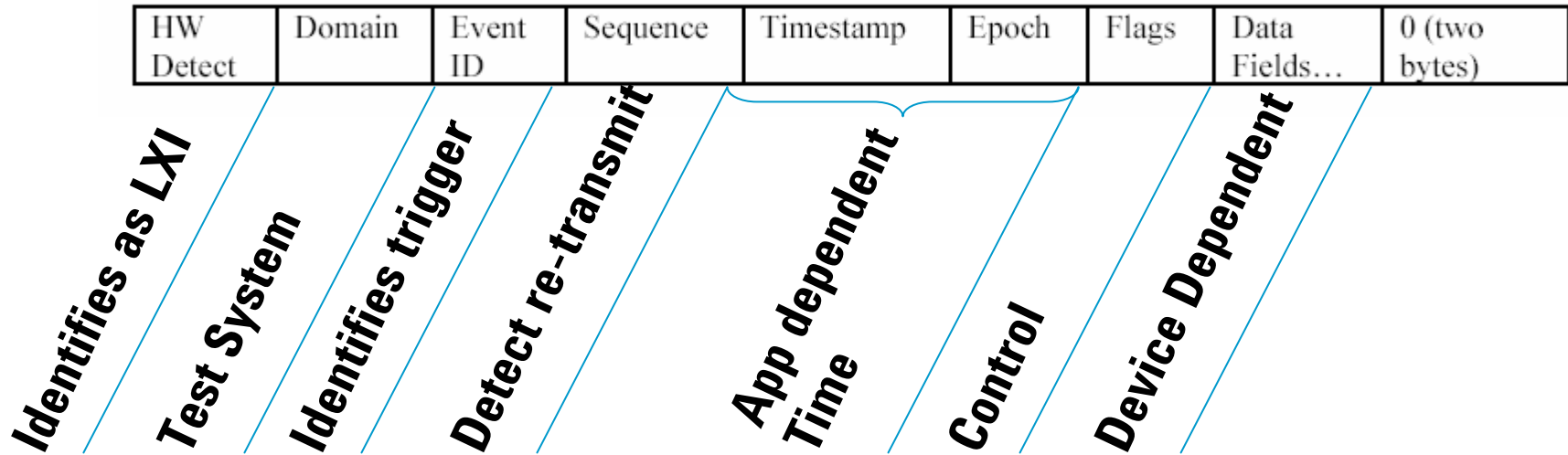
1588 Clock

- **Read time**
- **Verify synchronized**

Event Log

- **Debug**

LXI LAN Events



UDP Event

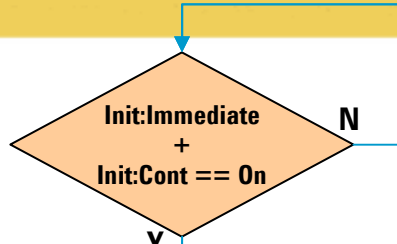
- Multicast trigger (IANA designated multicast address)
- Reasonably reliable multicast mechanism

TCP Event

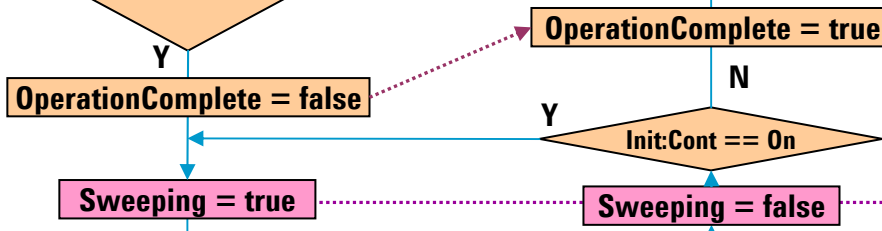
- Point-point
- Reliable

Arm – Trigger Model

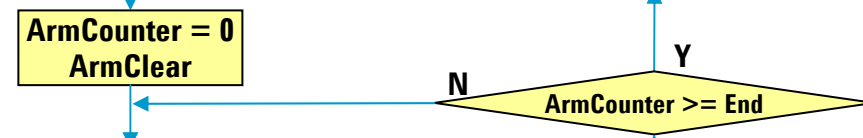
Idle:



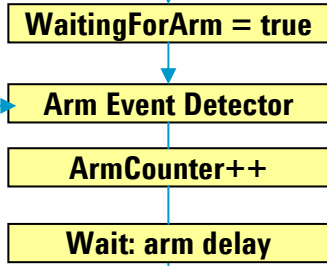
Initiated:



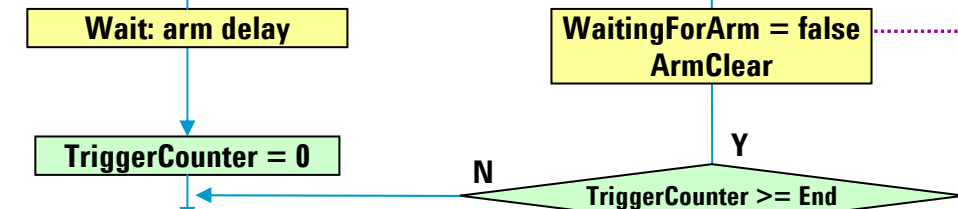
Arm:



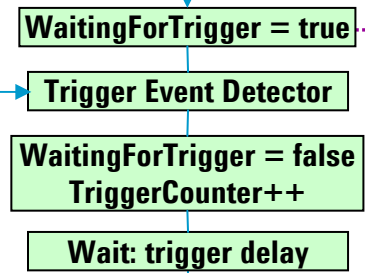
Arm Logic



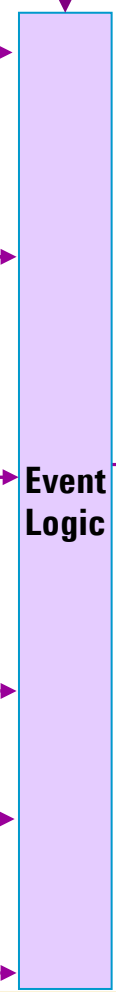
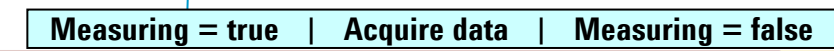
Trigger:



Trigger Logic

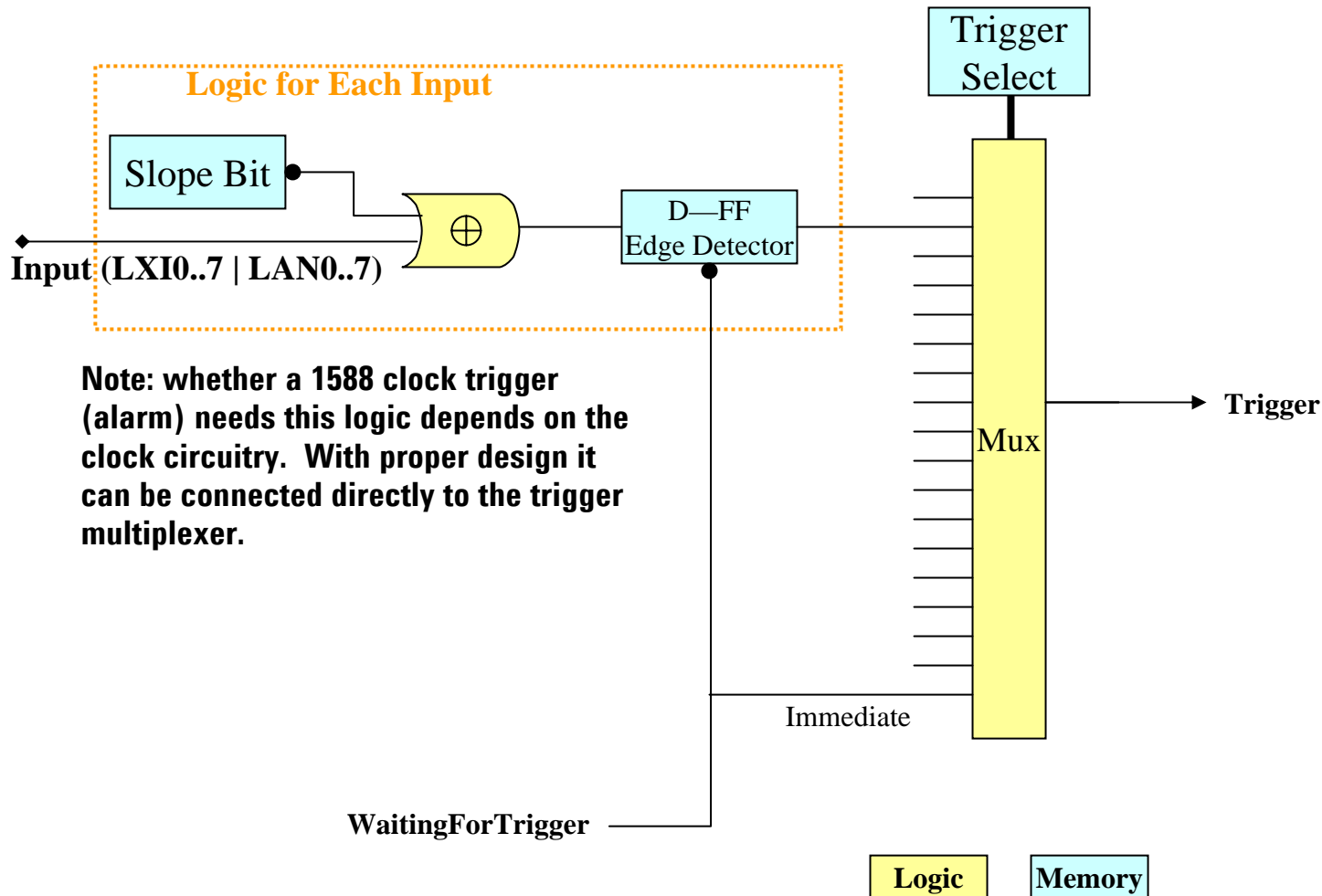


Measure:

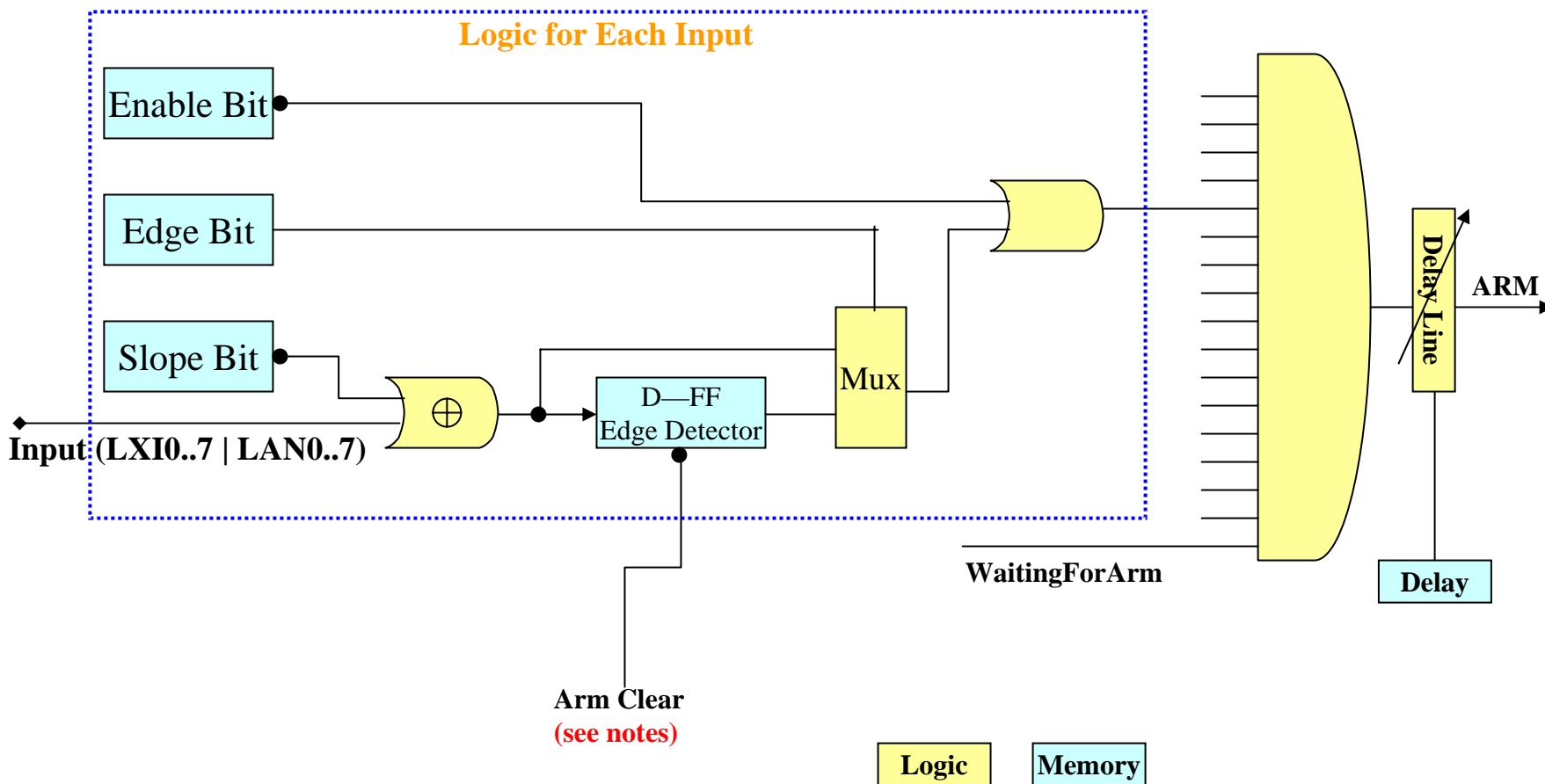


LXI0.7
LAN0.7

Trigger In Logic

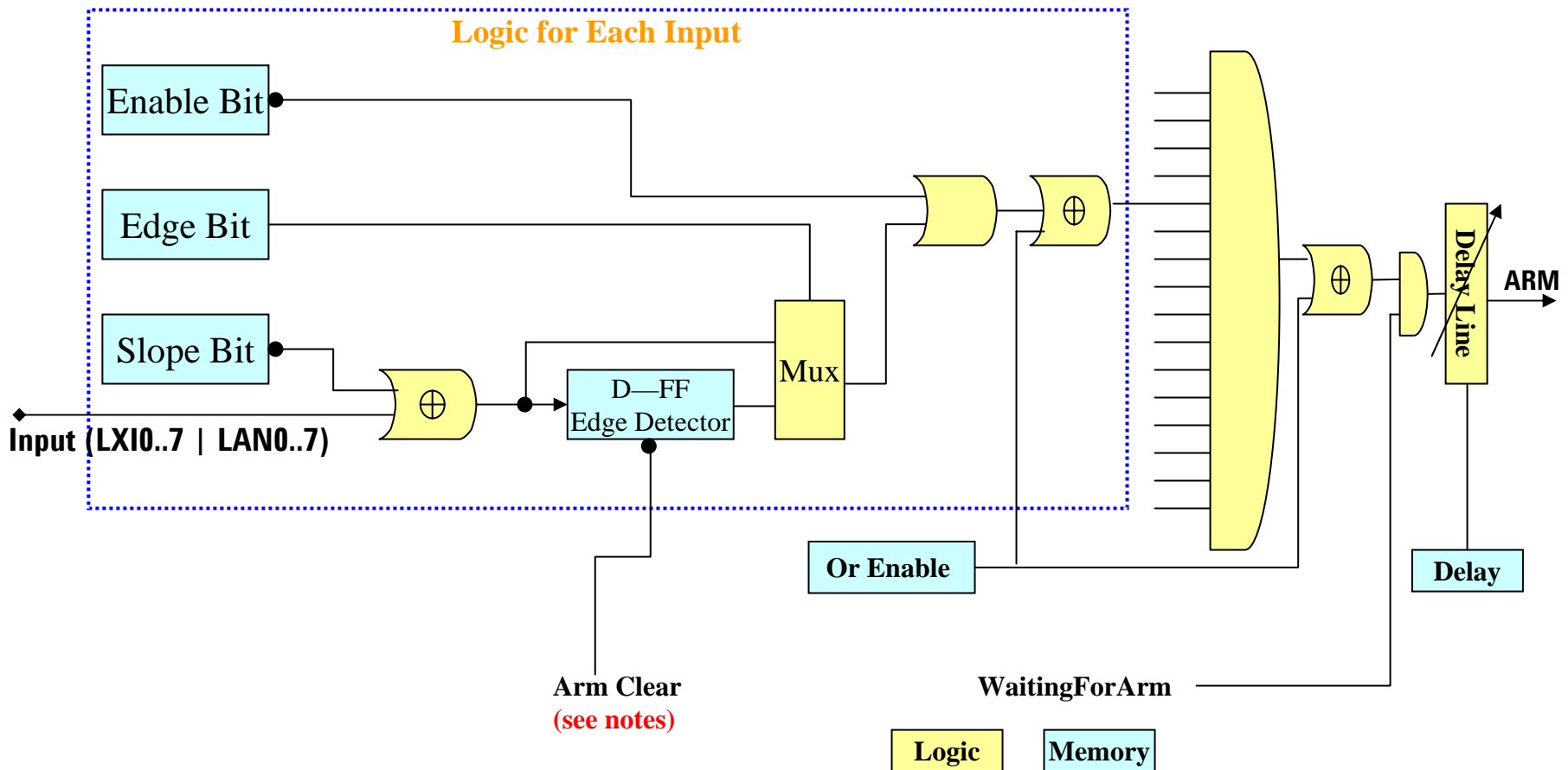


Arm Logic: LXI0..7 & LAN0..7



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

Selectable And & Or Summing



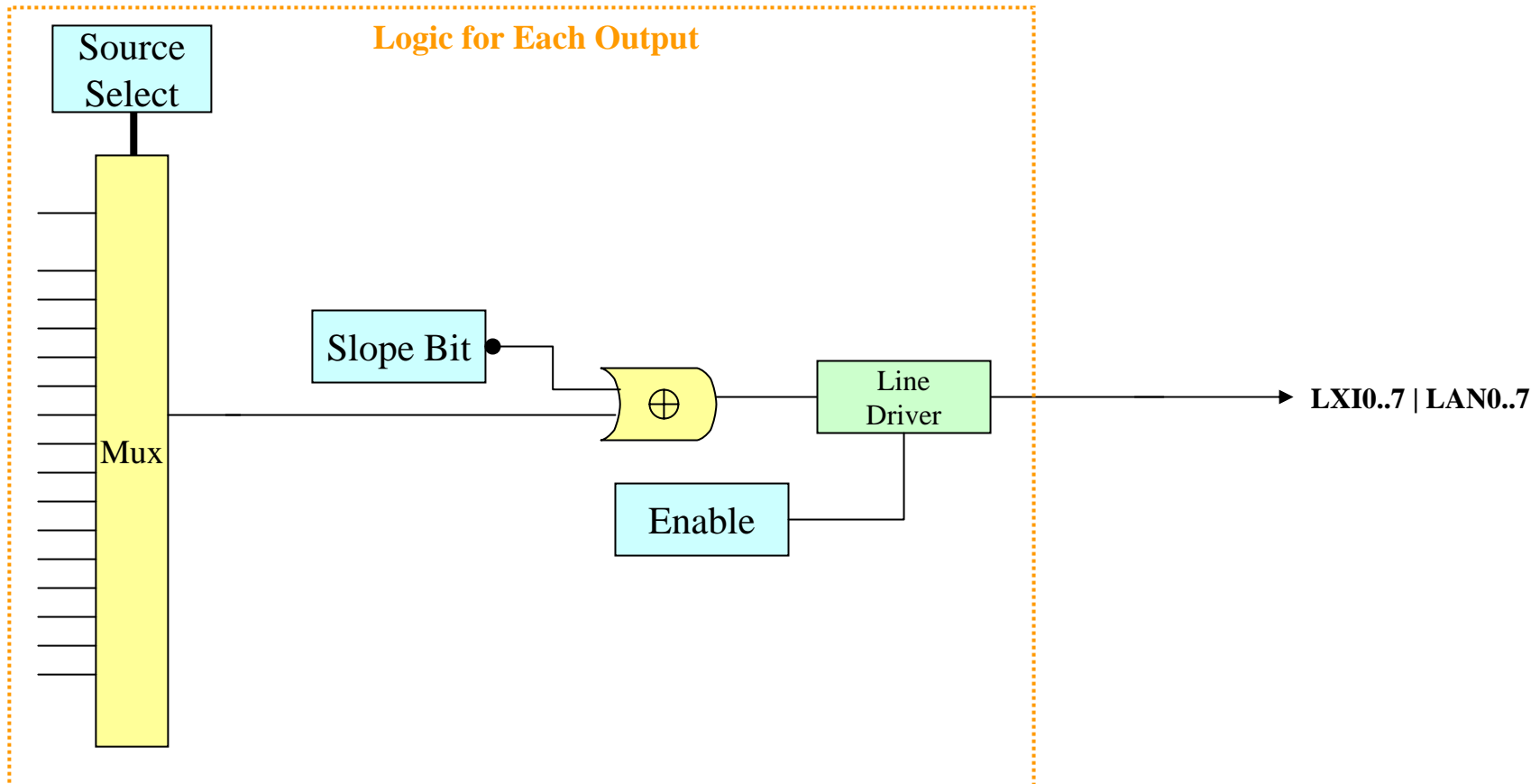
Trigger Subsystem

- **Can accept trigger source signals on LXI0..7 and LAN0..7**
 - Custom trigger sources supported via extensible repeated capabilities
- **Trigger source configuration options**
 - Delay specifies trigger delay
 - Negative value implies pre-trigger acquisition
 - Slope (positive or negative)
 - Filter restricts the senders that will be accepted
 - Details on upcoming slide
- **Trigger in response to Alarm condition (aka “Clock”)**
 - Details in upcoming slide
- **TriggerCount property controls the number of times the trigger has to occur to complete a measurement**

Arm Subsystem

- **Can accept arm source signals on LXI0..7 and LAN0..7**
 - Custom arm sources supported via extensible repeated capabilities
- **Arm source configuration options**
 - Edge detection or level detection
 - Delay to accommodate electrical delay in the signal path
 - Slope (positive or negative)
 - Filter restricts the senders that will be accepted
 - Details on upcoming slide
- **Arm in response to Alarm condition (aka “Clock”)**
 - Details on upcoming slide
- **ArmCount property controls how many times arm-trigger cycle repeats**

Event Out (Trigger Out) Logic



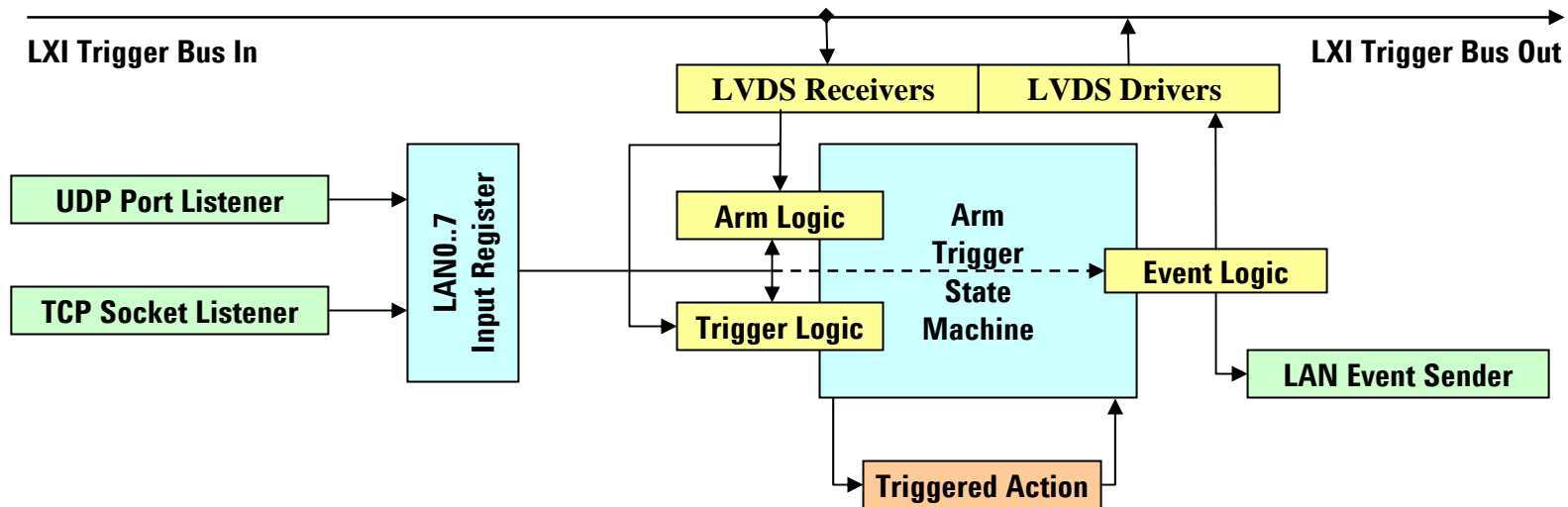
Logic

Memory

Events Subsystem

- **Event *sources* cause an event to be transmitted**
- **Six built-in (reserved) events sources**
 - **OperationComplete**
 - **Measuring**
 - **Settling**
 - **Sweeping**
 - **WaitingForArm**
 - **WaitingForTrigger**
- **Events transmitted on LXI0..7 or LAN0..7 or custom LAN event**
 - Referred to as *destination*
 - TCP broadcast supported

How the Pieces Fit Together: for a Simple Module



Logic

Memory

Software

Alarm Subsystem

- **Applies to Trigger and Arm subsystems**
- **Provides a signaling mode in addition to trigger bus and LAN events**
- **“Clocking mechanism”**
- **Two timing modes**
 - **Specify an absolute 1588 time**
 - **Specify a period and repeat count**
 - **Repeat count of zero => no repeat (single-shot)**

Primary APIs to Configure Trigger/Event

```
Arm.Sources.Item(<name>).Configure(Bool Edge,  
    Bool Enabled,  
    LxiSyncSourceSlopeEnum Slope);
```

```
Trigger.Sources.Item(<name>).Configure(double Delay,  
    LxiSyncSourceSlopeEnum Slope);
```

```
Events.Item(<name>).Configure(BSTR Source,  
    BSTR DestinationPath,  
    LxiSyncEventEnabledEnum Enabled,  
    LxiSyncSourceSlopeEnum Slope);
```

```
Trigger.Alarms.Item(<name>).Configure(double TimeSeconds, double  
    TimeFraction, double Period, long RepeatCount);
```

Tell: Digitizer to output WaitFor Trigger on LXI1 Arb to trigger when that event arrives

```
digSync.Events ("LXI1").Configure(  
    "WaitForTrigger", "",  
    LxiSyncEventEnabledEnum.LxiSyncEventEnabledOn,  
    LxiSyncSourceSlopeEnum.LxiSyncSourceSlopePositive);  
  
arbSync.Trigger.Sources ("LXI1").Configure(  
    0,  
    LxiSyncSourceSlopeEnum.LxiSyncSourceSlopePositive);  
  
arb.Trigger.Source = "LXI1";    // in instrument specific interface
```

LAN Triggers work similarly

```
digSync.Events ("LAN1").Configure(  
    "WaitingForTrigger", "",  
    LxiSyncEventEnabledEnum.LxiSyncEventEnabledOn,  
    LxiSyncSourceSlopeEnum.LxiSyncSourceSlopePositive);
```

```
arbSync.Trigger.Sources ("LAN1").Configure(  
    0,  
    LxiSyncSourceSlopeEnum.LxiSyncSourceSlopePositive);
```

```
arb.Trigger.Source = "LAN1";
```

Trigger and Arm Source Filtering

- **Specifying a filter restricts the arm sources that will be accepted by a particular receiver**
 - Only allow authorized sources => security mechanism
- ***Accepted* senders are specified using a special syntax**
 - Empty filter means packets from any port on any host are accepted
 - Specifying only the port means that any host communicating on that port can send events

Filter == [host[port]][,Filter]

"192.168.0.1:23"	Port 23 on host at IP 192.168.0.1
"A_SIGGEN1:23, A_SPECAN2:23"	Port 23 on host at DNS A_SIGGEN1 and port 23 on host at DNS A_SPECAN2
"192.168.0.1"	Any port on host at IP 192.168.0.1
":23"	Any host communicating on port 23
":23, A_SPECAN2"	Any host communicating on port 23 and any port on host at DNS A_SPECAN2

IVI Repeated Capabilities

- **LxiSync API makes heavy use of IVI *repeated capabilities***
- **Used extensively in IVI to model collections of things**
 - “Channels” in an oscilloscope
 - “Traces” in a spectrum analyzer
- **LxiSync defines five repeated capabilities**
 - LxiSyncArmAlarm
 - LxiSyncArmSource
 - LxiSyncTriggerAlarm
 - LxiSyncTriggerSource
 - LxiSyncEvent
- **Standard collection interface contains three well-known properties**
 - **Count**: returns number of items in collection
 - **Item**: returns a specific item by name
 - **Name**: returns the name of an item at a specified one-based index

Extensible Repeated Capabilities

- **Drivers must support several required repeated capabilities using reserved words**
 - “LXI0”, “LXI1”, ... “LXI7”
 - “LAN0”, “LAN2”, ... “LAN7”
 - These are registered in the IVI Configuration Store
- **Trigger, Arm, and Event subsystems use extensible repeated capabilities for the “Sources” collection**
 - Allows users to add custom LAN events (or even other custom hardware events)
- **Extensible repeated capabilities contain three well-known methods**
 - **Add**: add a new item to the collection
 - **Remove**: remove a specific item from the collection
 - **RemoveAllXXX**: remove all *custom* items from the collection

LXI 1.0 Rules and Recommendations

- 6.1 Provide IVI driver (C or COM)**
- 6.2 Initialize driver with VISA compatible string**
- 6.3 IVI API should have "Source" as property**
- 6.4 8 LAN Events for Arm/Trigger (inbound) and Events (outbound)**
- 6.5 Time representation (2 64-bit floats)**
 - 6.5.2 All data time-stamped (property names)**
- 6.6 Domain Property**
- 6.7 TCP and UDP for events**
- 6.8 Rec: Event Log**
- 6.9 Rec: API to IDENT light**
- 6.10 – 6.11 Roadmap locking requirements**

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.

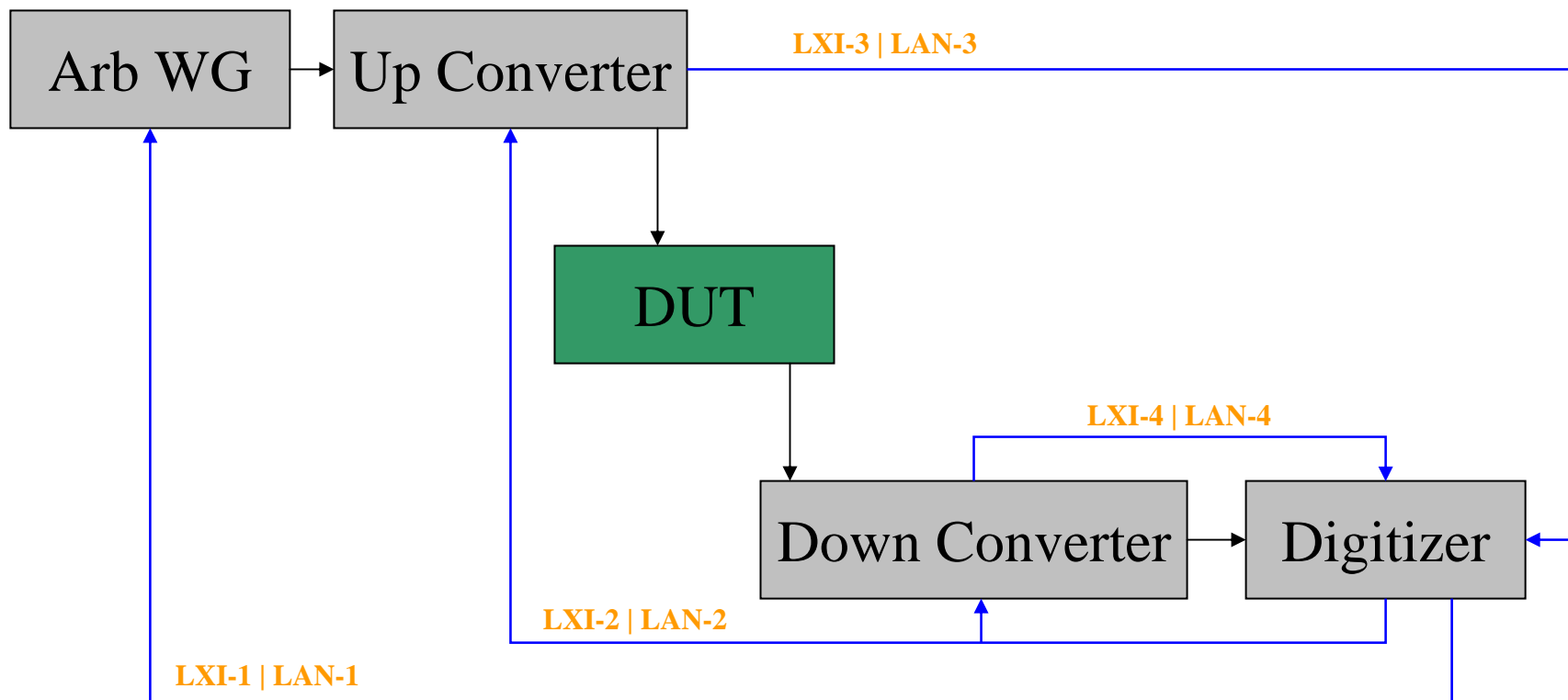
If no name is specified, use Event Name
(LXI0..7 | LAN0..7 | Data |..).

[optional item]

END

LXI Trigger Use Case: Stimulus Response Test

Block Diagram



Arm-Trigger State Machine Signal Relationships

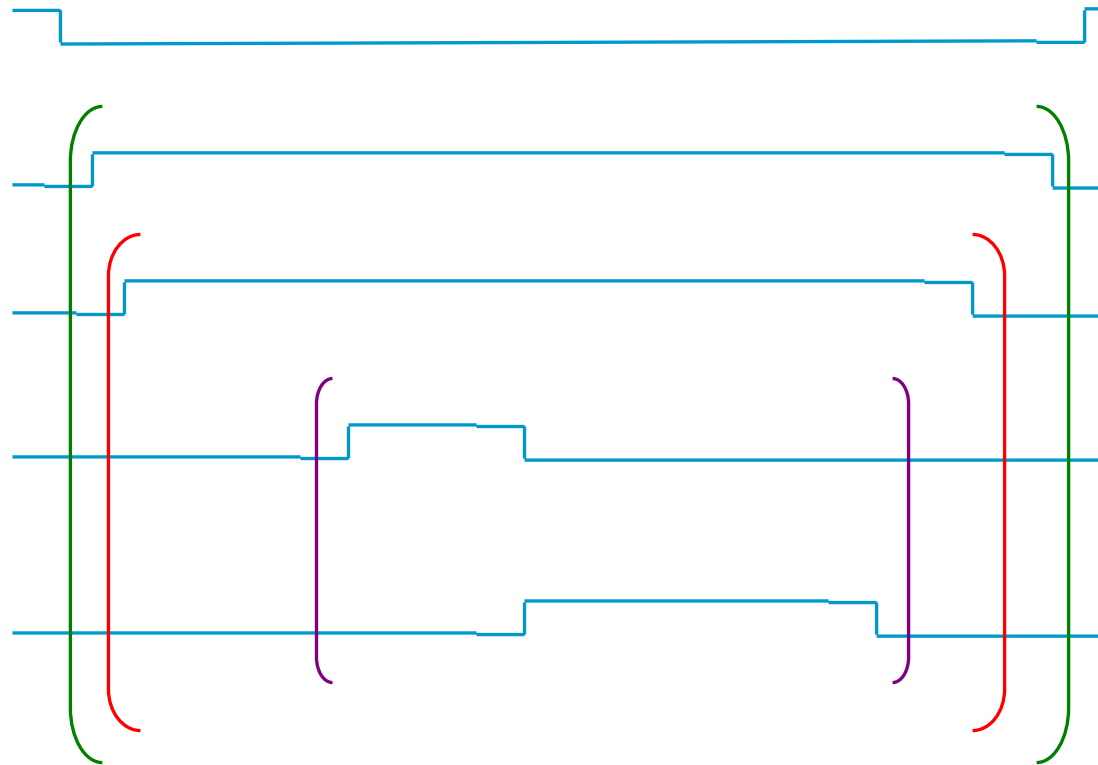
OperationComplete

Sweeping

WaitingForArm

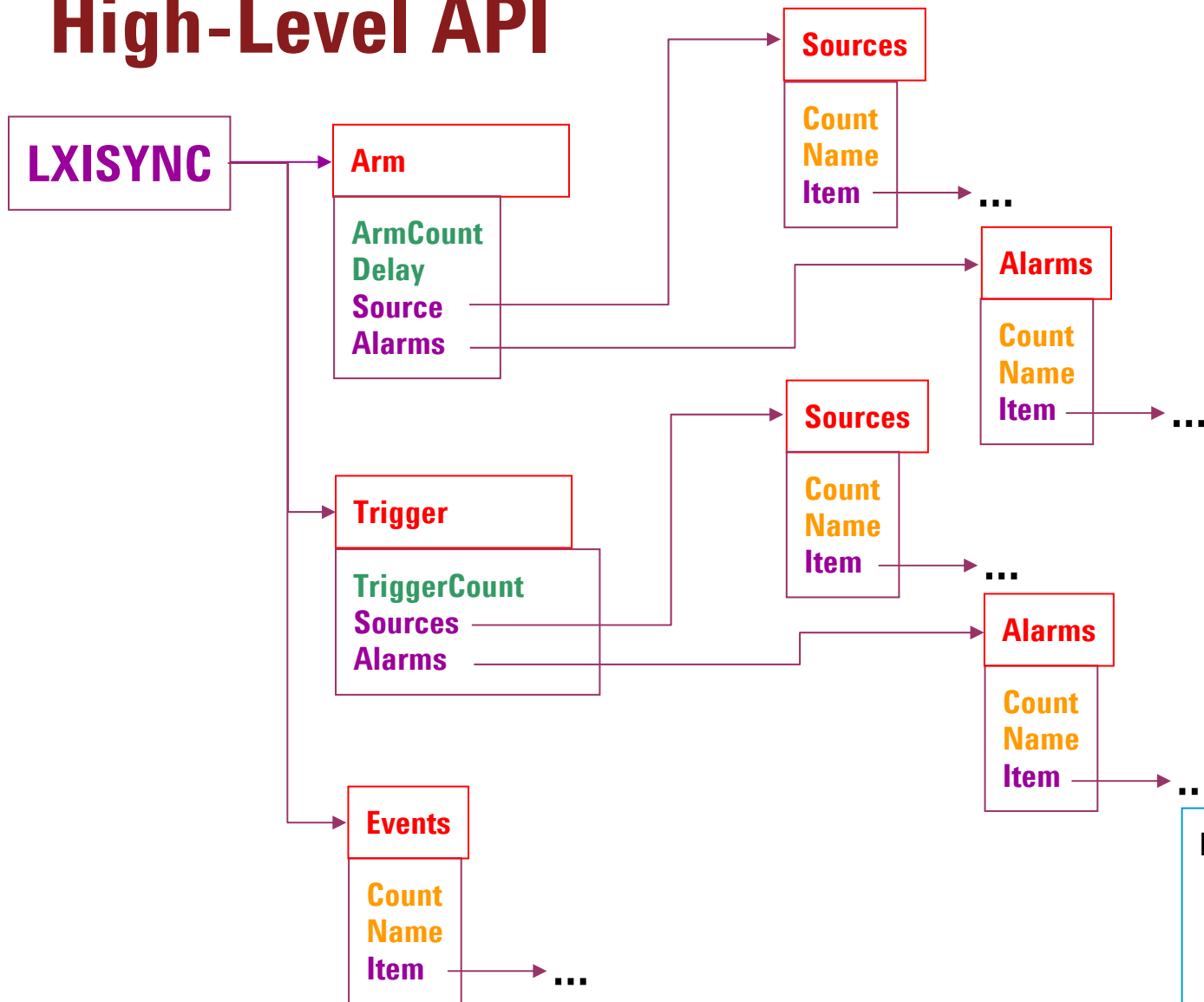
WaitingForTrigger

**Measuring
Settling**



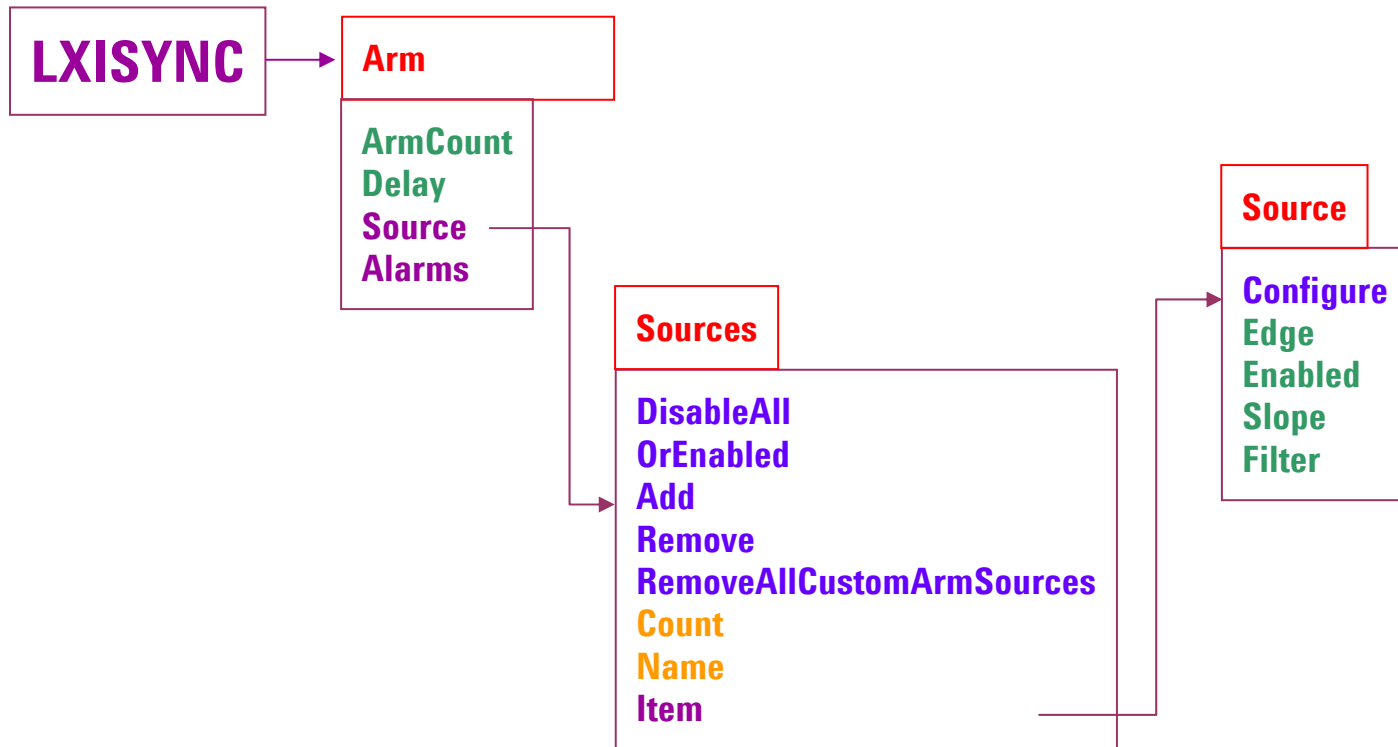
[May Occur multiple times]

High-Level API



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

Arm Sources

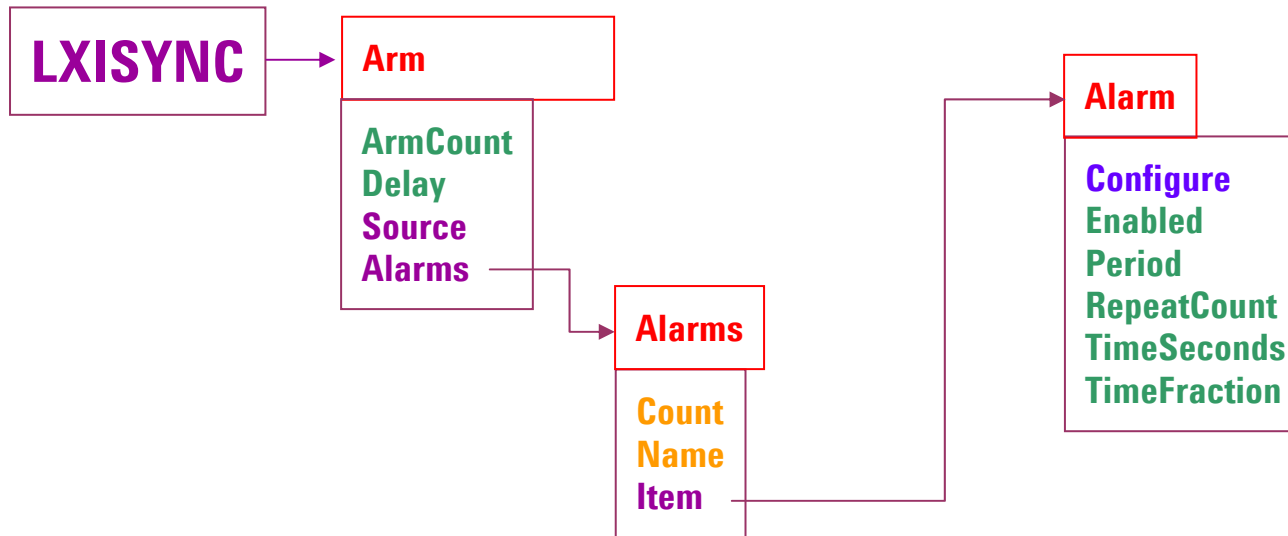


Legend:

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

Page 26

Arm Alarms

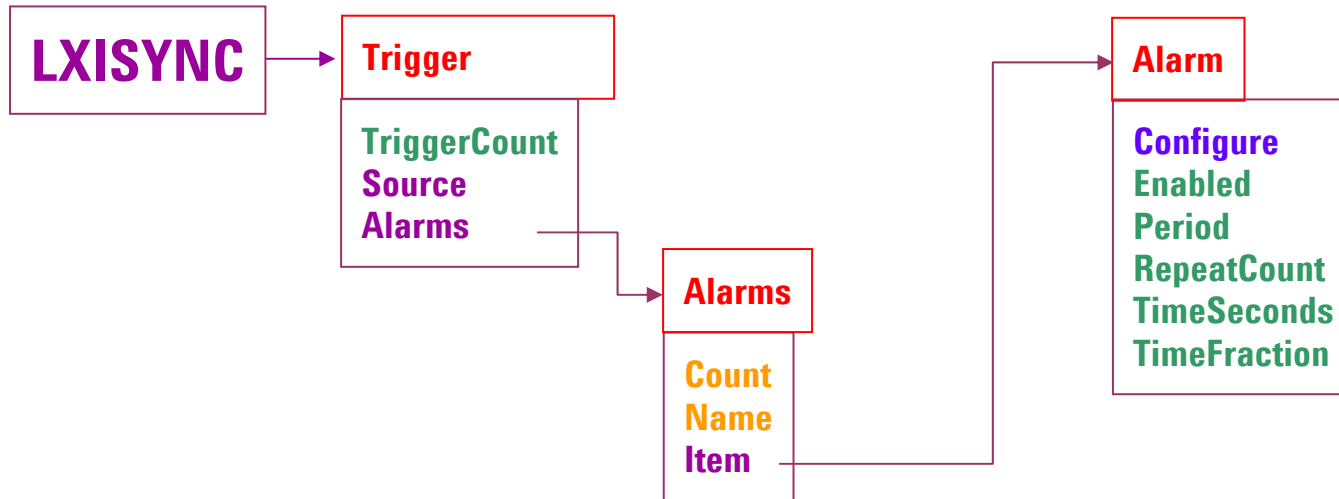


Legend:

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

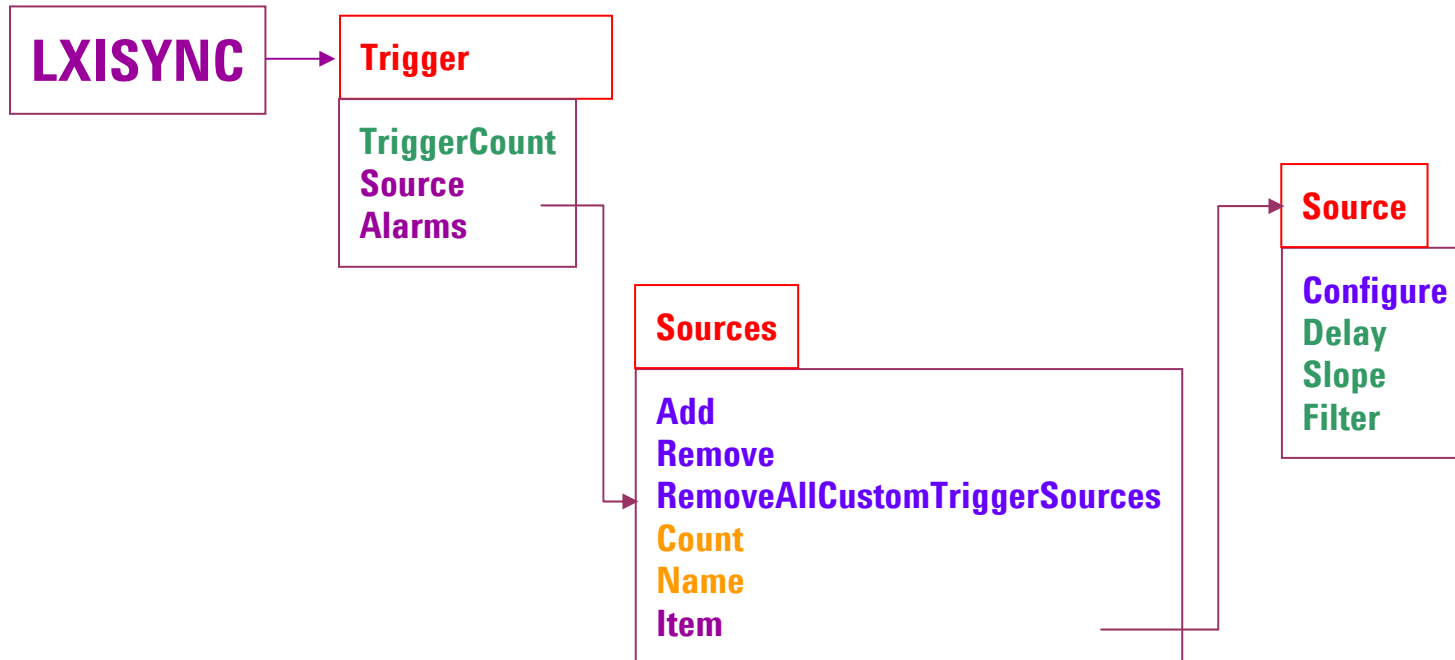
Page 27

Trigger Alarms



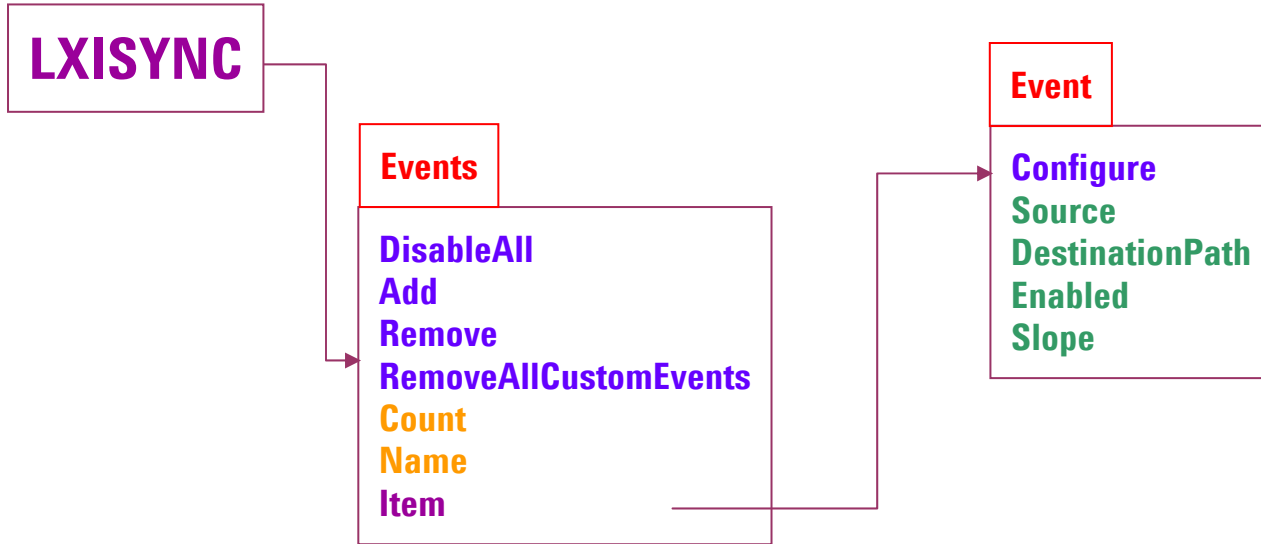
Legend:
 Property
 Read Only Property
 Method
 Interface Pointer (Property)
 Interface

Trigger Sources



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

Events

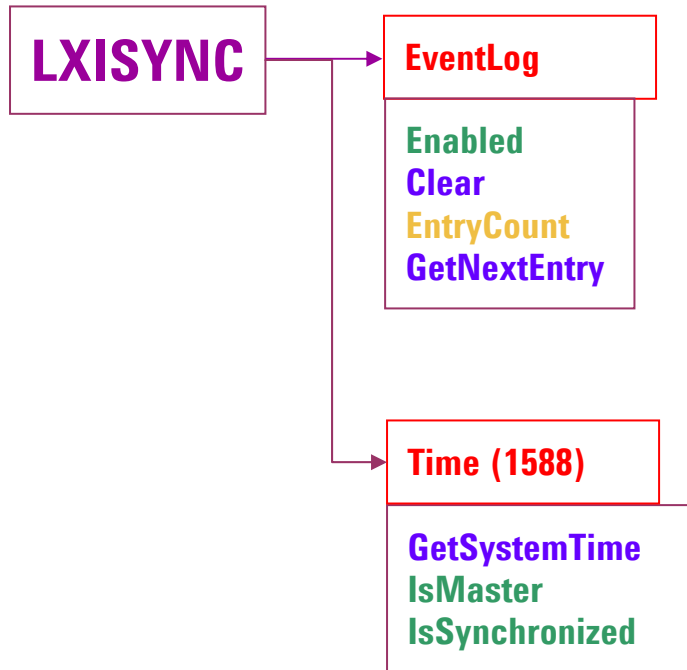


Legend:

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

Page 30

High-Level API (cont.)

**Legend:**

Property

Read Only Property

Method

Interface Pointer (Property)

Interface

Page 31

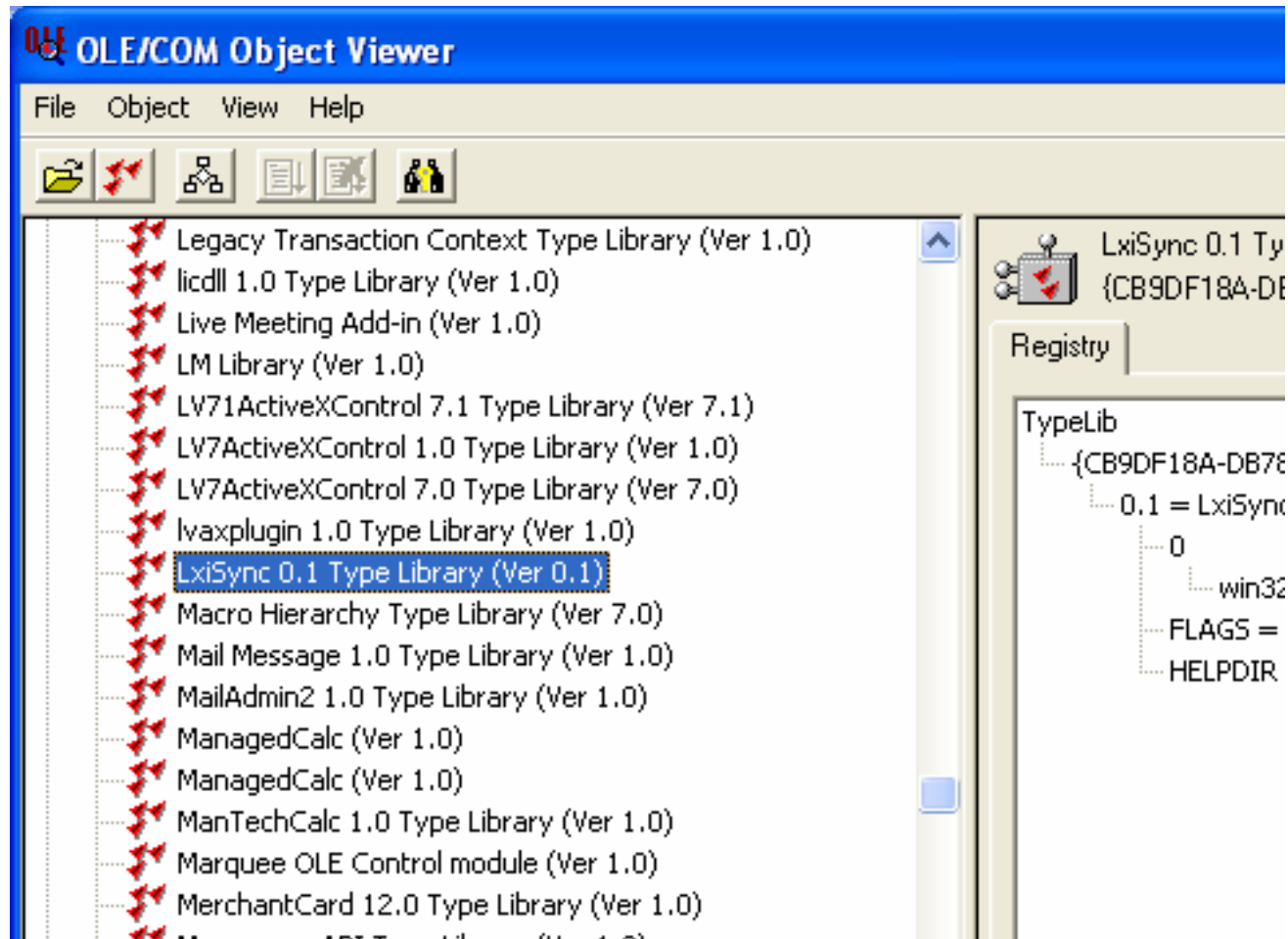
LxiSync and Nimbus

The screenshot displays the LxiSync - Nimbus application window. The interface includes a menu bar (File, Edit, View, Metadata, Project, Tools, Help), a toolbar, and a Project Explorer on the left. The Project Explorer shows a tree view of the project structure, with the 'Configure' method selected under the 'ILxiSyncArmAlarm' class. The main area shows the configuration for the 'Configure' method, including its return type (Void) and a list of parameters.

Name	Type	Direction
TimeSeconds	Double	In
TimeFraction	Double	In
Period	Double	In
RepeatCount	Int32	In
Enabled	Boolean	In

Additional controls include 'Add...', 'Remove', 'Edit...', 'Up', and 'Down' buttons, and a checked 'Initialize required' checkbox. The right side of the window features an Editors panel with options like Metadata, SCPI, Implementation,IVI-C Wrapper, Models, and Help. The bottom of the window has an Output window and a Task List checkbox.

LxiSyncTypeLib in Oleview



LxiSyncTypeLib in Oleview (cont.)

The screenshot shows the ITypeLib Viewer application. The left pane displays a tree view of the LxiSyncLib (LxiSync 0.1 Type Library) with the following structure:

- Enums
 - enum typedef enum LxiSyncSourceSlopeEnum
 - enum typedef enum LxiSyncEventEnabledEnum
 - enum typedef enum LxiSyncErrorCodesEnum
- Structs
- Modules
- Interfaces
 - interface ILxiSync
 - interface ILxiSyncArm
 - interface ILxiSyncArmSources
 - interface ILxiSyncArmSource
 - interface ILxiSyncArmAlarms
 - interface ILxiSyncArmAlarm
 - interface ILxiSyncEventLog
 - interface ILxiSyncEvents
 - interface ILxiSyncEvent
 - interface ILxiSyncTime
 - interface ILxiSyncTrigger
 - interface ILxiSyncTriggerSources
 - interface ILxiSyncTriggerSource
 - interface ILxiSyncTriggerAlarms
 - interface ILxiSyncTriggerAlarm
- Dispinterfaces
- CoClasses
- Typedefs
- Unions

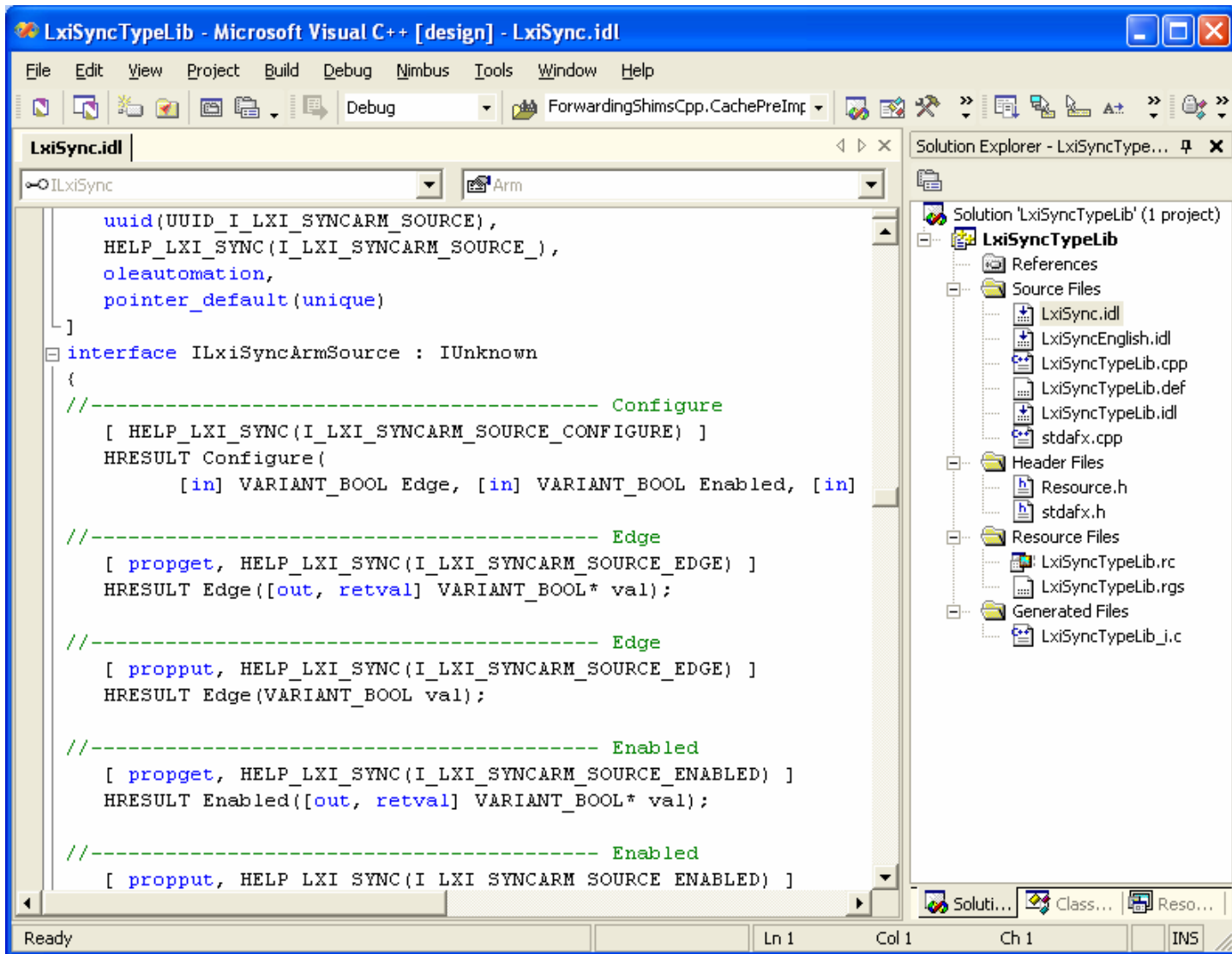
The right pane displays the generated .IDL file code:

```
// Generated .IDL file (by the OLE/COM Object Viewer)
//
// typelib filename: LxiSyncTypeLib.dll

[
  uuid(CB9DF18A-DB78-4CF9-B417-E4093B45F8AE),
  version(0.1),
  helpstring("LxiSync 0.1 Type Library"),
  helpfile("LxiSync.chm"),
  helpcontext(00000000),
  custom(DE77BA64-517C-11D1-A2DA-0000F8773CE9, 100663657),
  custom(DE77BA63-517C-11D1-A2DA-0000F8773CE9,
1124924539),
  custom(DE77BA65-517C-11D1-A2DA-0000F8773CE9, Created by
MIDL version 6.00.0361 at Wed Aug 24 16:02:18 2005
)
]
library LxiSyncLib
{
  // TLib :          // TLib : OLE Automation : {00020430-
0000-0000-C000-00000000000046}
  importlib("stdole2.tlb");

  // Forward declare all types defined in this typelib
  interface ILxiSync;
  interface ILxiSyncArm;
  interface ILxiSyncArmSources;
  interface ILxiSyncArmSource;
  interface ILxiSyncArmAlarms;
  interface ILxiSyncArmAlarm;
  interface ILxiSyncEventLog;
  interface ILxiSyncEvents;
  interface ILxiSyncEvent;
}
```

LxiSync Visual Studio Project



Arm digitizer when both upconverter and downconverter done settling

```
// tell up-converter to output Settling on LXI3 for use by digitizer
upSync.Events.get_Item("LXI3").Configure("Settling", "",
    LxiSyncEventEnabledEnum.LxiSyncEventEnabledOn,
    LxiSyncSourceSlopeEnum.LxiSyncSourceSlopeNegative);
```

```
digSync.Arm.Sources.get_Item("LXI3").Configure(true, true,
    LxiSyncSourceSlopeEnum.LxiSyncSourceSlopePositive);
```

```
// tell down-converter to output Settling on LXI4 for use by digitizer
downSync.Events.get_Item("LXI4").Configure("Settling", "",
    LxiSyncEventEnabledEnum.LxiSyncEventEnabledOn,
    LxiSyncSourceSlopeEnum.LxiSyncSourceSlopeNegative);
```

```
digSync.Arm.Sources.get_Item("LXI4").Configure(true, true,
    LxiSyncSourceSlopeEnum.LxiSyncSourceSlopePositive);
```

Relationship to IVI Classes

- **Add LXI trigger sources to existing trigger source**
 - LXI Sync API used to configure sources
 - Existing classes use either strings or enums – initially these will require custom API's to unify triggering.
 - LXI Sync will recommend how to handle for consistency
- **Trigger Arm**
 - Arm functionality not in existing classes
 - When enabled, becomes an additional precondition on instrument classes
 - Applications that don't touch Arm methods will work the same
- **Events, Logging, Alarm setup, Time access**
 - Independent of existing IVI Class interfaces

LXI Sync Specification Status circa 8/19

- **Phase 1 complete including**
 - **All APIs specified and documented**
 - **COM and IVI-C hierarchy documented**
 - **Error values documented**
 - **Substantial review completed**
 - **Type Library (currently coding against this in lieu of prototypes)**