PowerDNA Synchronization

1.1 Introduction

The PowerDNA Sync Interface provides two capabilities that are key components of many applications.

- It allows a PowerDNA Cube to be triggered by, or synchronized to, an external event or signal.
- It allows the various I/O layers/boards within a cube to be triggered by and/or synchronized to, a variety of signals within the cube or to external signals brought in directly to an I/O layer.

PowerDNA synchronization is based on two fixed-direction signal connections (*Sync In* and *Sync Out*) which are available on the CPU layer of the Cube as well as on four bidirectional sync signals (*Sync0* through *Sync3*) provided on the primary internal data bus of the Cube and shared by all I/O layers as well as the CPU layer. A block diagram of the system is shown **Figure 1-1**.

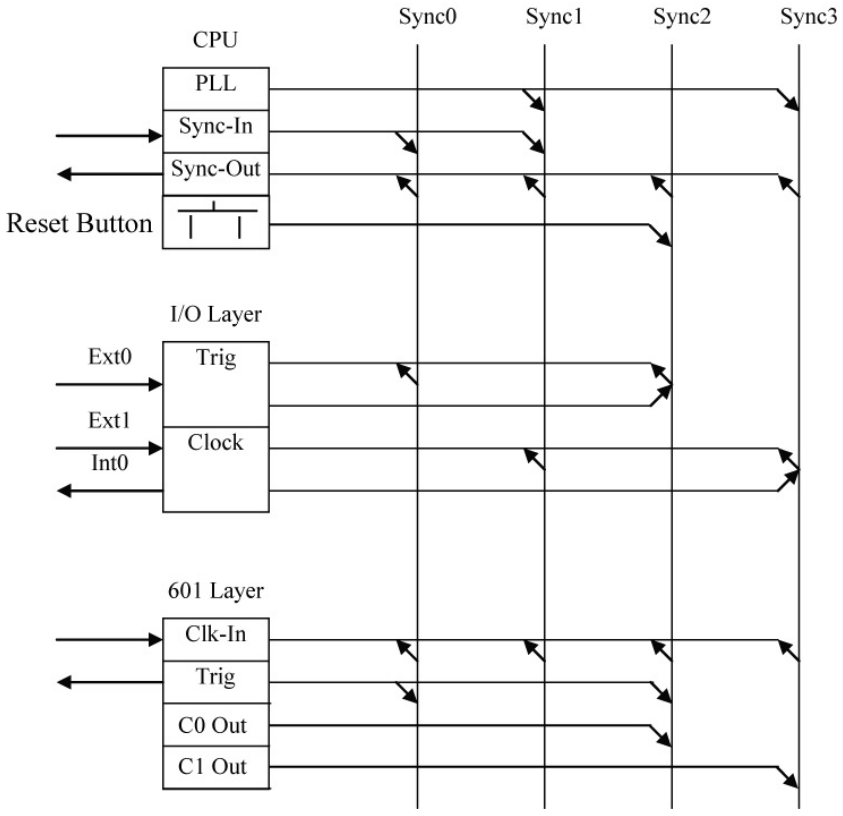

Figure 1-1 Sync Interface Bus Diagram

Note that the 601 Counter/Timer layer is a unique case. The counter timer capabilities of the board make it ideal for generating various timing and synchronization signals. Therefore, it is given more extensive access to the sync bus than standard layers.

FPGA bases of the various layers, combined with the Cube firmware, can be configured to create an almost unlimited set of trigger and synchronization scenarios. Not all of these are supported in the standard released product. However, our standard trigger/synchronization model can satisfy the requirements of virtually all users.

The two-signal external Sync interface and the four-signal internal sync configuration are described in the two following sections, respectively.

1.2 External Sync/Trigger interface

Each PowerPC PowerDNA Cube provides an external Sync connector on the front panel (below the reset button, above the first I/O layer). These Sync interface signals may be monitored or controlled by the logic on the processor board of the Cube, or they may be connected directly to internal Sync signals shared by the internal I/O layer boards. The remainder of this section describes the external Sync interface. Please refer to Section 1.3 for details on the Cube internal sync bus.

The external Sync interface provides four connections. The Sync interface pins share a common ground, but are fully isolated from the Cube itself.

- Sync In
- Sync Out
- +5 VDC (up to 10 mA)
- Ground

Sync In is a dedicated input and may be used as a trigger source for the layer or to provide an external clock source to the cube. As a trigger, it supports the following modes.

- **Trigger Mode –** Start an application on a rising or falling edge (software selectable)
- **Trigger/Stop Mode –** Start an application on a rising (or falling) edge, Stop the application on the next rising (or falling) edge.
- **Gate High Mode –** Run the application while *Sync In* is High, Stop when *Sync In* is Low
- **Gate Low Mode –** Run the application while *Sync In* is Low, Stop when *Sync In* is High
- **Direct Layer Mode –** The *Sync In* terminal does not have a direct “Cube wide” function, but is connected directly to a *Sync* pin on one of the I/O Layers.

Sync Out is a dedicated output that may be configured to output any of the following:

- **Sync Buffer Mode –** The Sync Out signal is simply a buffered version of Sync In
- **Ext Clock Mode –** The internal clock of the cube is brought out to the Sync out connection and may be used to synchronize clocks across cubes or throughout an application.
- **Direct Layer Mode –** The Sync Out signal is controlled by one of the I/O layers within the Cube.

There are two Sync cables available, the DNA-CBL-SYNC-RJ and the DNA-CBL-SYNC-30. The DNA-CBL-SYNC-RJ provides a Sync connector on one end and an RJ-45 connector on the other. This cable is used to connect external signals to the cube. Typically, the DNA-CBL-SYNC-RJ is plugged into the cube and also into the DNA-STP-SYNC panel. The DNA-STP-SYNC provides three sets of connections as shown below.

- The board provides a screw terminal connection for each of the four Sync signals.
- The board provides 6 parallel RJ-45 connectors. All four SYNC signals are connected in parallel as well. These parallel connections allow the user to easily connect the identical external trigger signals to multiple cubes.
- The board provides a seventh RJ-45 connector with its Sync Out pin connected to the Sync In terminals of the four paralleled connectors. This will allow the Sync output of a single Cube to control other Cube Sync inputs without injecting the additive delays of multiple daisy-chained DNA-CBL-SYNC-30 connections.

A block diagram of the DNA-STP-SYNC is shown in **Figure 1-2** on page 3.

The DNA-CBL-SYNC-30 cable is a 30-inch cable that simply crosses the Sync In and Sync Out connections. This connects the Sync Out of one Cube to the Sync In of the next, allowing the second cube to be slaved to the first.

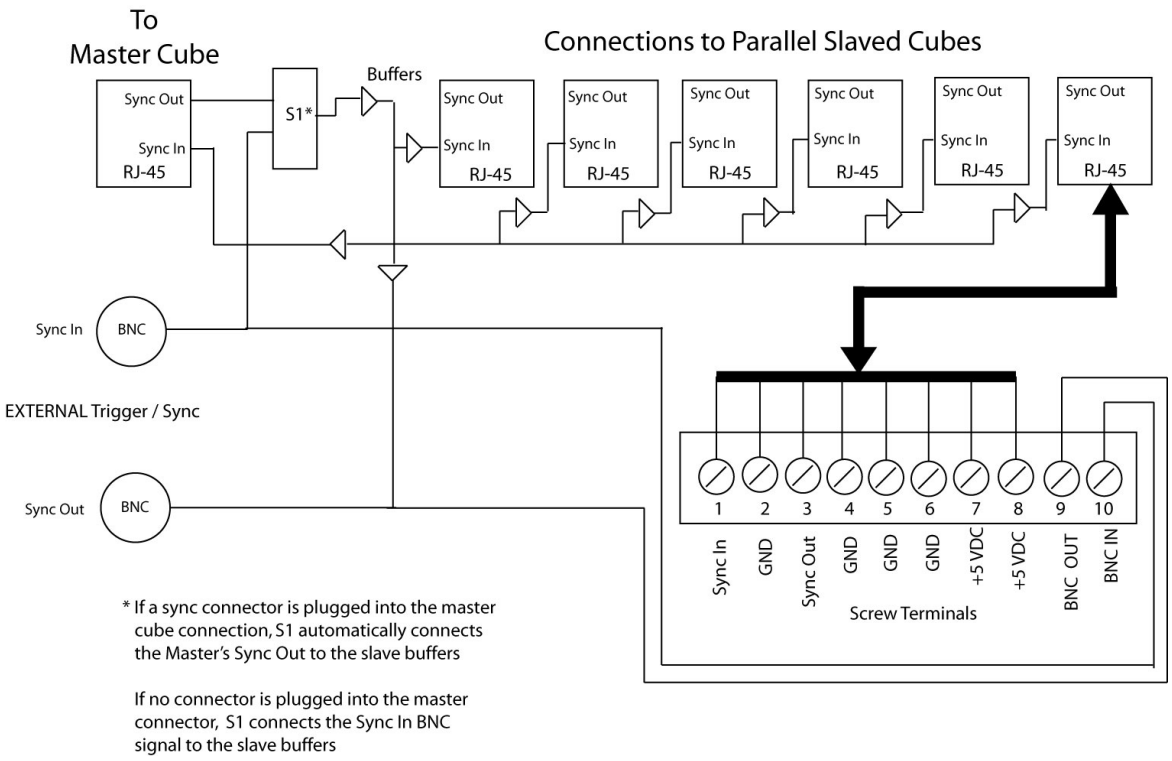

Figure 1-2. DNA-STP-SYNC Block Diagram

1.3 Internal Sync Connections

Four Sync signals on the internal I/O interconnect bus are brought to each layer.

These four lines are designated as “Sync0” through “Sync3”. The diagram below shows the configuration of the four internal Sync signals and also the pinout on the I/O layers.

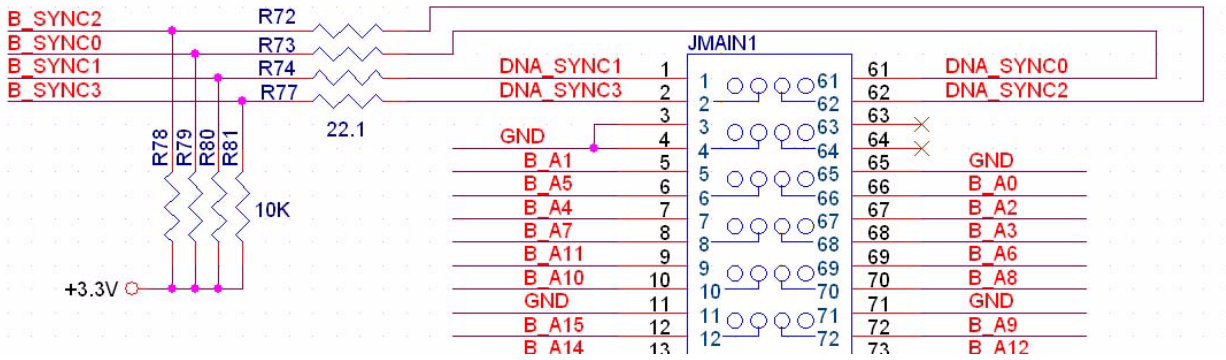

Figure 1-3. Schematic of Internal Sync Connections

As you can see, each line is pulled up with 10k resistor. In the maximum PPC-8 cube, the total resistance is 1430 Ohms with a termination current of 2.3mA. These parameters prevent synchronization lines from bouncing and also ensure that proper drive is available from every layer.

The four synchronization lines have identical functionality and any of the synchronization signals can be routed to any one of the synchronization lines. These capabilities allow great flexibility of synchronization interface configuration. However, they make the synchronization model very complex.

To simplify the synchronization interface model, UEI has standardized on the following conventions.

- Sync0 – dedicated trigger input
- Sync1 – dedicated input clock or system timebase clock
- Sync2 – inter-layer triggering
- Sync3 – inter-layer clocking

This line assignment addresses virtually all anticipated synchronization requirements.

The logic on the CPU board allows either of the external Sync connections provided at the external Sync connector to be mapped to any of these four internal sync signals. In most applications, the master Sync input from the CPU board is connected either to the Sync0 or Sync1 terminals. If an I/O layer is being used as a master system trigger, however, it is expected that the external Sync Out connection would be mapped to either Sync2 or Sync3.

1.3.1 Layer triggering and clocking

A layer can be triggered using the following sources:

- Firmware executing DaqBIOS Start command
- EXT0 line
- Sync0 line
- Sync2 line

A layer can be clocked using the following sources:

- Internally
- EXT1 line (or EXT0 in a single-line layer)
- Sync1 line
- Sync3 line

A layer can feed its trigger signal to Sync2 line. A layer can feed its clock signal to the Sync3 line.

The Sync-Out line on the CPU layer can either output Sync[0..3] lines or be used for alarm notification.

1.4 Use Application Cases

Use Case 1 – Starting/Stopping Multiple Layers at the Same Time (No External Trigger)

This is a very common operation in ACB and DMap modes, in which the software issues a command to begin or end an application, but multiple layers within the cube need to be synchronized. If the layers involved are in software triggering mode, the firmware proceeds as follows:

- The start sequence for all layers involved is stored (performed by prog_...() functions in the device driver)
- All I/O layer timestamp counters are reset and synchronized with the timestamp counter on the CPU layer (this is required to align data relative to timestamps)
- The start sequence is executed (normally it is a single write to LCR register of each layer involved)

Layers can be clocked internally or externally in this case.

Use Case 2 – External Trigger (via Sync Connector)

An external trigger drives Sync-In. Sync-In is sampled by the CPU layer, which then drives the internal Sync0 line. Installed I/O layers use the Sync0 line as a trigger.

Use Case 3 – External Trigger (through an I/O Layer)

An I/O layer can be used to trigger one or more of the other layers in the cube. This trigger may be based directly upon an external trigger, or based upon its own trigger or terminal count. In this case, the master layer (which provides the sync signal) drives the Sync2 line. Other layers are triggered by this Sync2 signal. Clock configuration defines what signal (software, internal, external, sync bus) will be used as a layer clock.[1]

1.Note that this changes the definition of DQ_LN_CxCKSRCx bits. The bit combinations used are: 0 - software, 1 - internal, 2 - external, and 3 - sync interface.

Use Case 4 – External Clock

An external clock can be either fed into the Sync-In input on the CPU layer or the CPU-layer PLL output may be routed to SYNC-Out and then back to SYNC-In. The clock configuration defines what signal (software, internal, external, sync bus) will be used as a layer clock.[1]

Use Case 5 – Master-Slave Clocking

In this case, one layer produces a clock signal and places it on the Sync3 line. Other layer(s) in the cube then uses it as a clock. The clock source can be a standard (analog/digital) layer as well as a counter-timer (CT) layer.

Use Case 6 – Synchronous Buffered Input and Output

In this case, an analog output layer feeds its clock to the Sync3 line. An analog input layer then uses this clock signal to synchronize its A/D sampling to the D/A layer clock. This allows the PowerDNA to be used in stimulus/response applications.

Use Case 7 – Sequenced Acquisition (Based on the DNA-CT-601 Counter/Timer Layer)

Sequenced acquisition can be accomplished by feeding the Sync-In trigger or clock signal into one of the counter-timers on CT-601 layers (via Sync0 and Sync1 lines) and then back out to other I/O layers (using Sync2 and Sync3 lines). This allows the PowerDNA Cube to acquire data, or output waveforms, based on a predefined sequence in the CT-601 FIFO.

1.5 Synchro-nization API

TBD

1. Note that this changes the definition of DQ_LN_TRIGEDGEx bits. The bit combinations used are: 0 - software, 1 - internal, 2 - external, and 3 - sync interface.