Pulse-Per-Second (PPS) Signals and the OpenCPI Time Server

Overview

Every standard OpenCPI FPGA platform has an instance of the TimeServer IP within the infrastructure. In the absence of any PPS source, this logic freewheels to maintain a local timebase on the FPGA. In the presence of a PPS signal, the TimeServer will detect a valid signal and use it for phase lock.

Signals

The table below lists the signals that are central to the TimeServer behavior.

Signal Name	Function
`sys0_clk`	Low-Jitter on board XO or TCXO (typical 200 MHz, +/- 50 ppm or better)
`ppsOut`	PPS Phase-Aligned Output (active with and without ppsExtIn)
`ppsExtIn`	PPS Input (< 1000 ppm sampling window) (typically from GPS)

Platform-Specific Pinouts

Depending on the platform, the signals above are brought on and off the card on different pins and connectors. Where possible, we have chosen single-ended SMA connectors for PPS in and out (for example, Xilinx ML605). On other platforms (for example Altera S4GX) there are no available SMAs; so we have assigned these pins to proprietary pins suitable for the particular etch (on the alst4 platform, we use the transition board for the Altera-Proprietary HSMC board). In all cases, the user should consult both the top-level pinout constraint file (UCF or QSF) as well as the schematic. Users can move the IO to any suitable location for their platform.

System Behavior

When two or more boards are used, the PPS may be configured in a star or daisy-chain configuration.

In a star configuration, a single PPS signal is distributed equally, possibly through a distribution amplifier, to all the FPGA platforms. At the cost of PPS distribution, this approach potentially has the lowest time jitter and uncertainty; as a single source feeds all platforms equally.

In a daisy-chain configuration, the PPS signal is fed to one platform, and its output fed to the next, and so on. This configuration may be easy to implement, but despite the long-term zero-phase position of the ppsOut signal; there can be jitter and error propagation added from one board to the next.