Time Difference of Arrival (tdoa)

Timestamp Capture IP module

by Joel

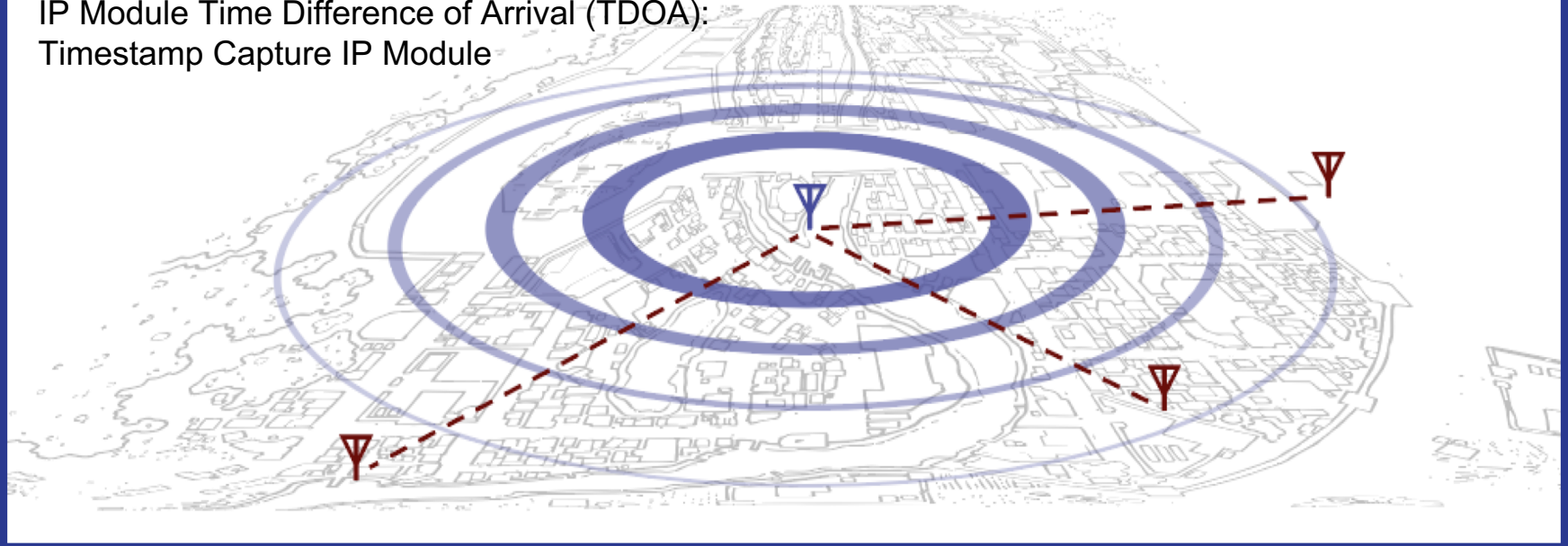
TDOA Overall Scheme

TDOA is based upon a technique called **multilateration**. Multilateration estimates the location of a wireless emitter, and the target position is determined by the time difference between signals arriving at multiple receivers.

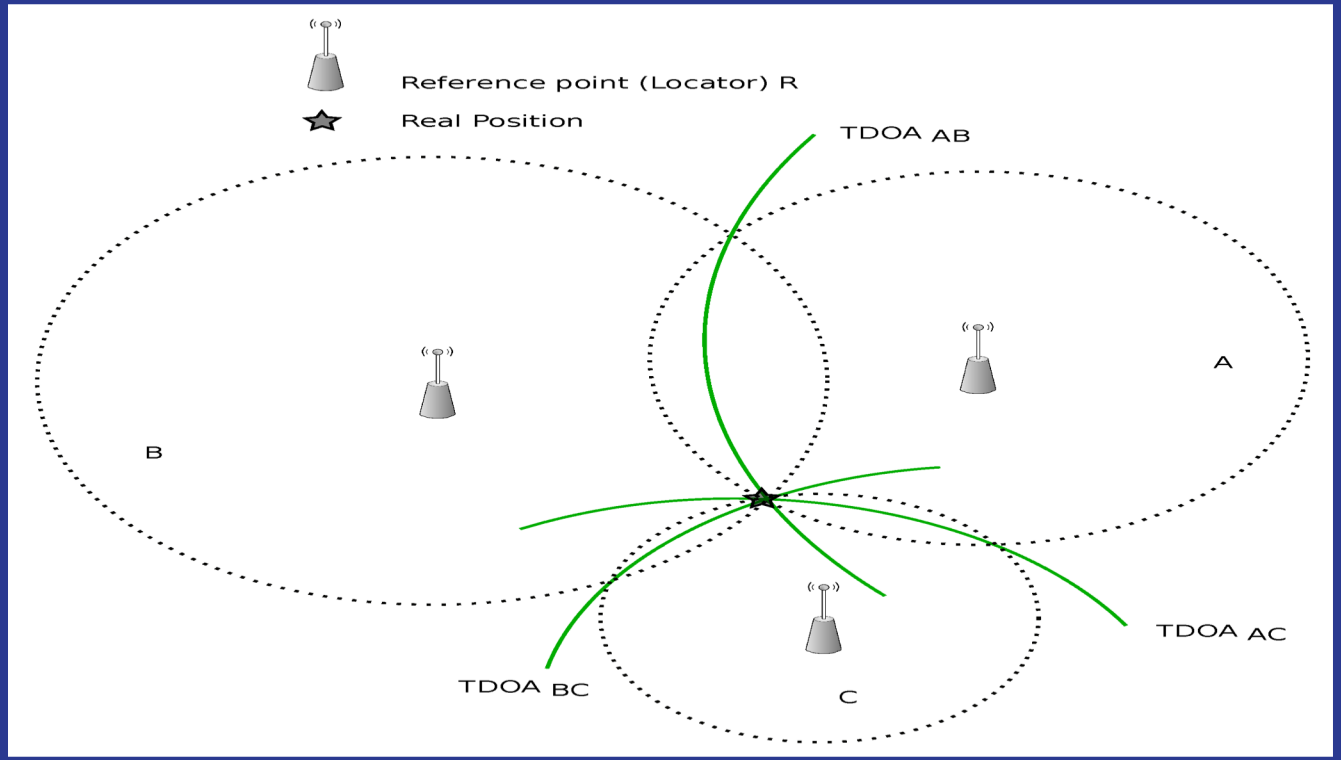
Once these wireless emitted signals are captured and time-stamped via the Pico-Zed receivers they are then placed into a software based TDOA algorithm to determine the location of the emitter target.

TDOA Illustration

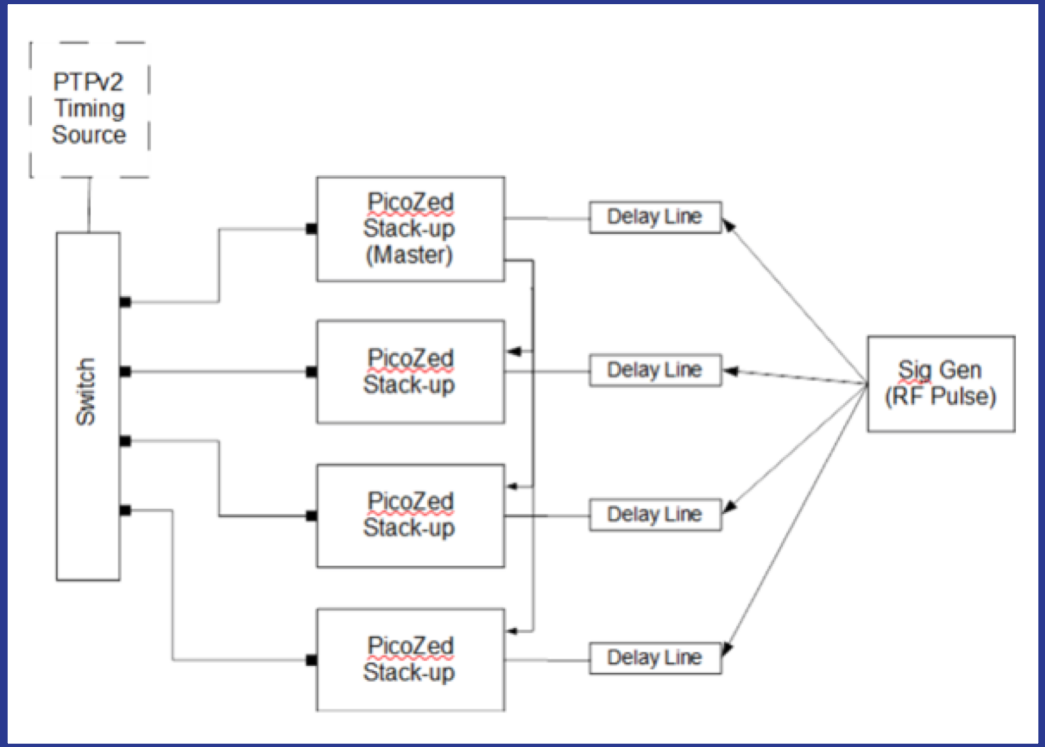
Time Difference of Arrival (TDOA): Timestamp Capture IP Module Time Difference of Arrival (TDOA): Timestamp Capture IP Module



TDOA Illustration



TDOA Project Overview



Tasking

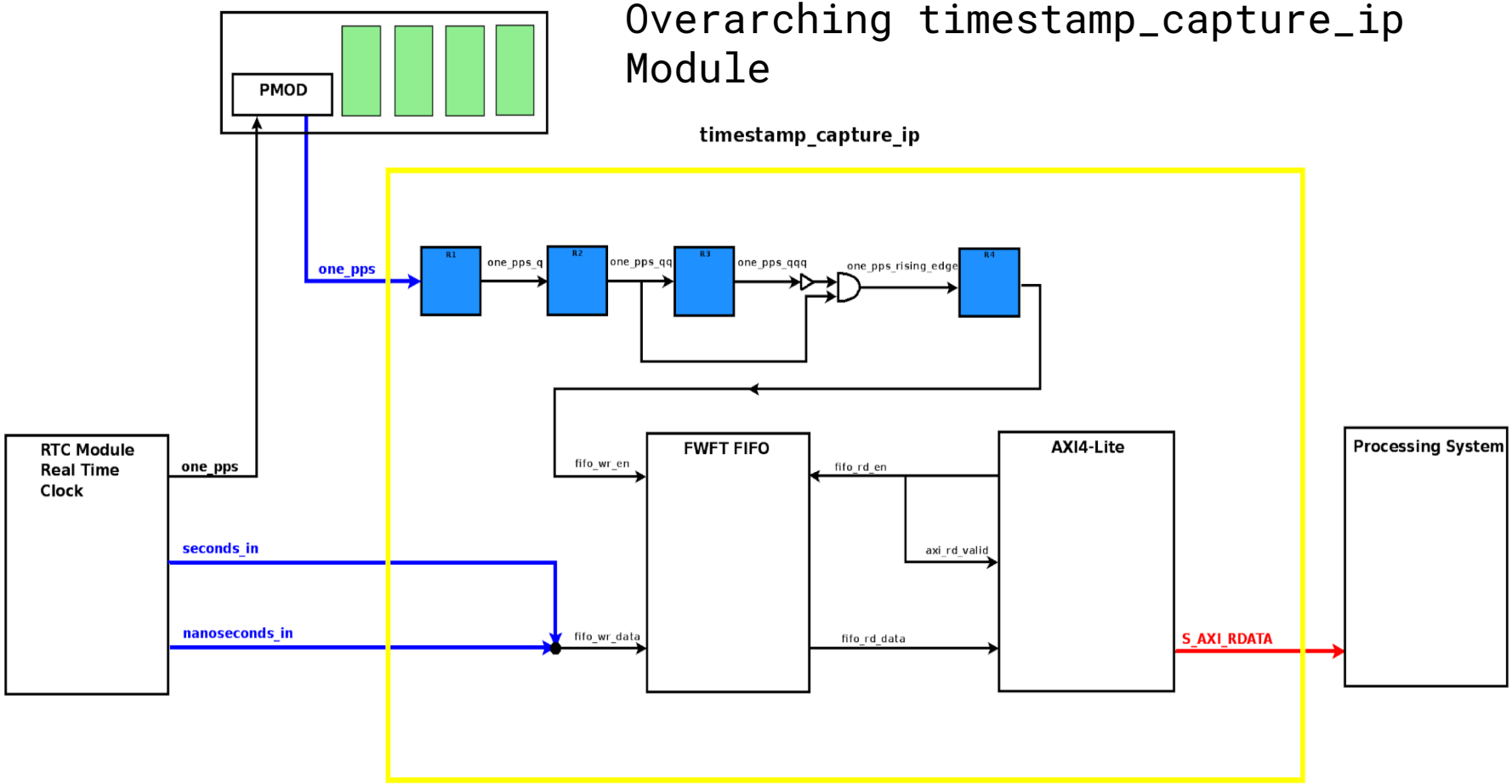
- Develop an AXI-Lite Firmware Intellectual Property (IP) core that can capture PTPv2 timestamps and be configured as a master or a slave.
- When configured as a master, the IP core should generate its own 1 Pulse Per Second (1PPS) digital output with a 1us pulse width.
- When configured as a master or a slave, the IP core should input the 1PPS signal and write the current timestamp (seconds and nanoseconds) to a FIFO. The FIFO should be readable from software.

Hurdles of Design

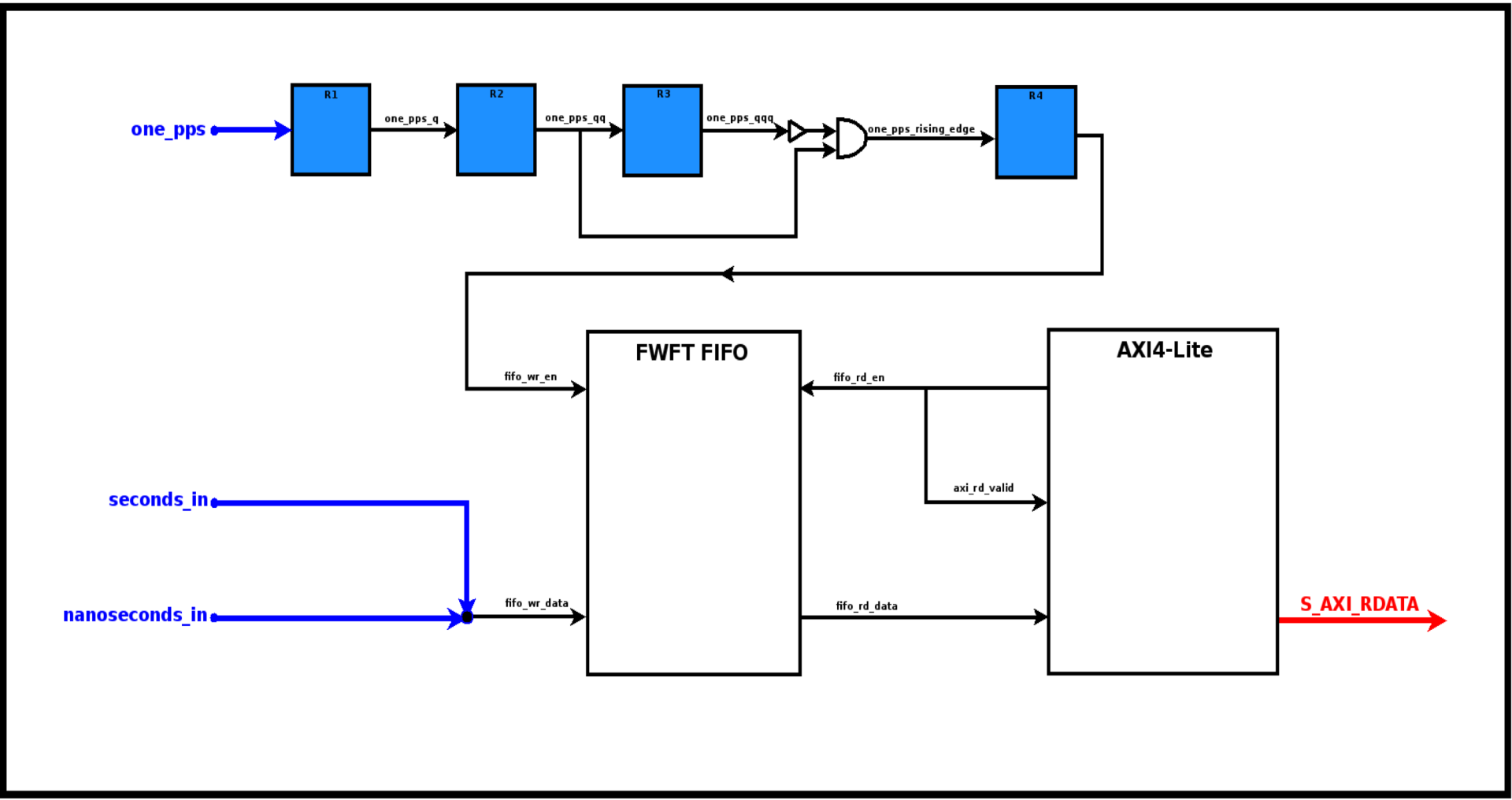
- Learning VHDL, and the Vivado ecosystem.
- Understanding individual design components such as the FWFT FIFO (First-Word fall through First-In-First-Out) component, and the AXI4-Lite (Advanced eXtensible Interface protocol) component.

- The 4:1 aspect ratio of the FIFO component.
- Time Domain synchronization registers.
- Rising-Edge detect of the 1PPS signal

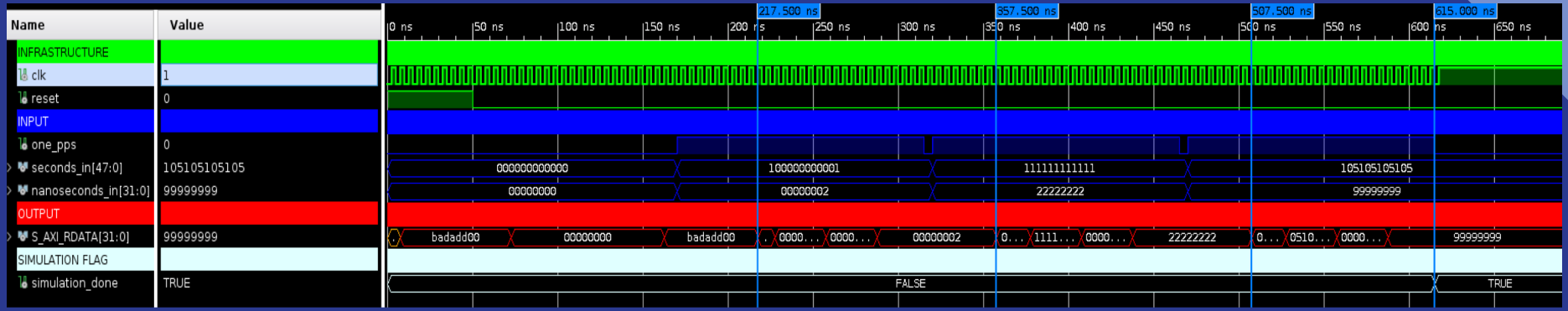
Overarching timestamp_capture_ip Module



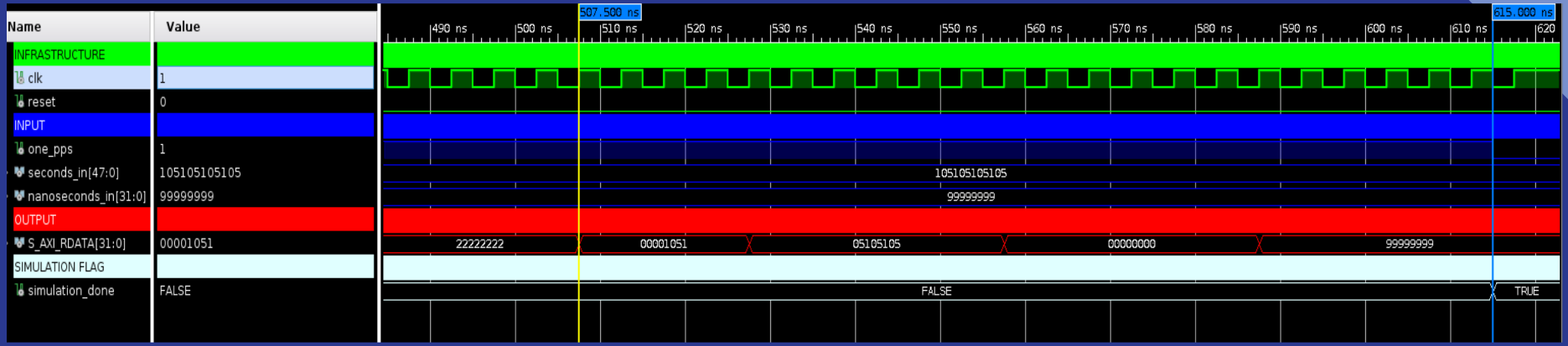
Timestamp_capture_ip



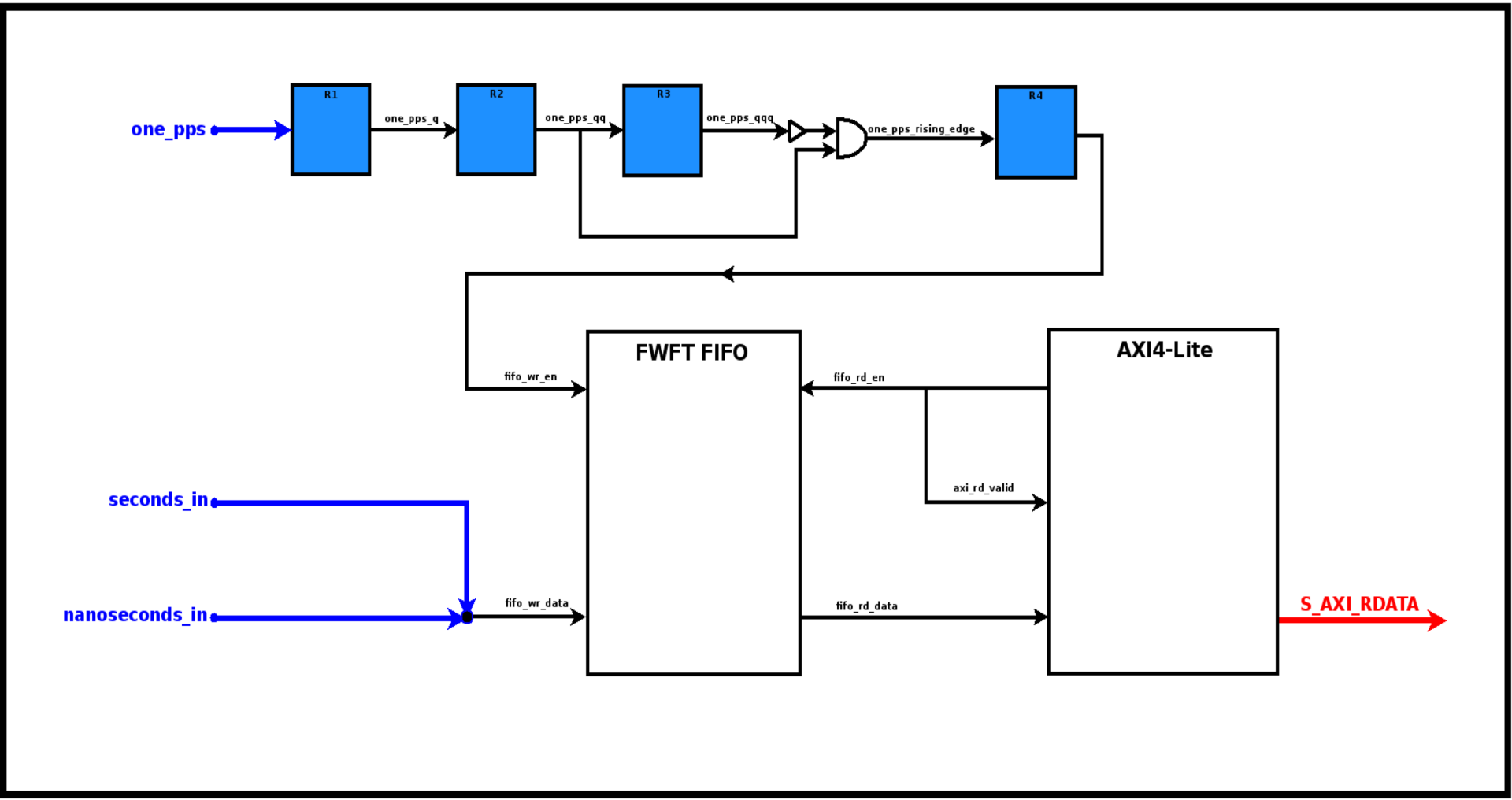
I/O Test-Bench



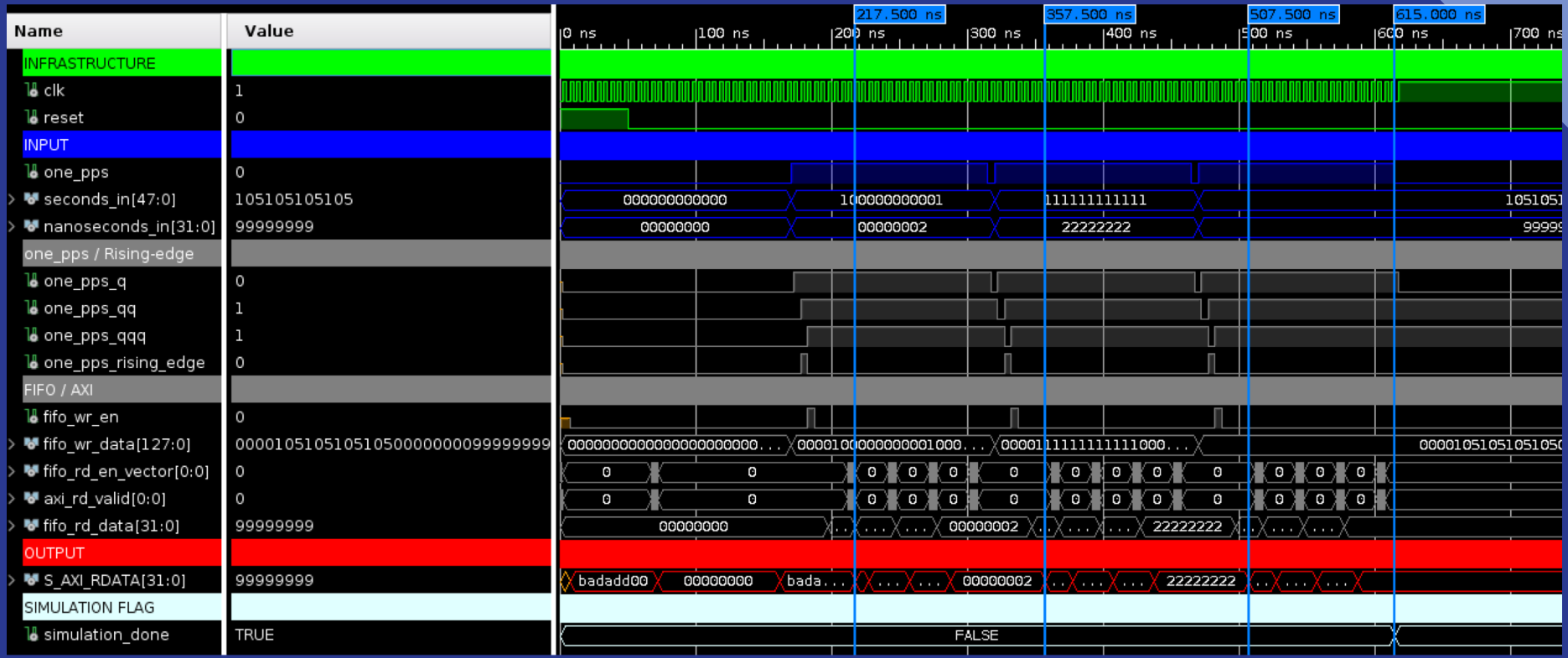
Time-stamp samples Test-Bench



Timestamp_capture_ip



Overall Test-Bench



Notes

- The 4:1 aspect ratio of the FIFO.
- The ‘0’ padding of the 32 bit data words.

Next Task

- Add IP to repo and pico_zed_fmc2 repo as a submodule.
- Build IP with ‘make’.
- Build the picozed_fmc2 with ‘make’.
- Open the picozed_fmc2 project in Vivado, open the block design.
- Add a user IP repository in the Vivado IP Catalog that points to the IP.
- Instantiate the IP in the Vivado block design.
- Connect the IP to the Processor System (Zynq) in the block design.
- Connect the IP to the Real-Time Clock in the block design.

End Presentation