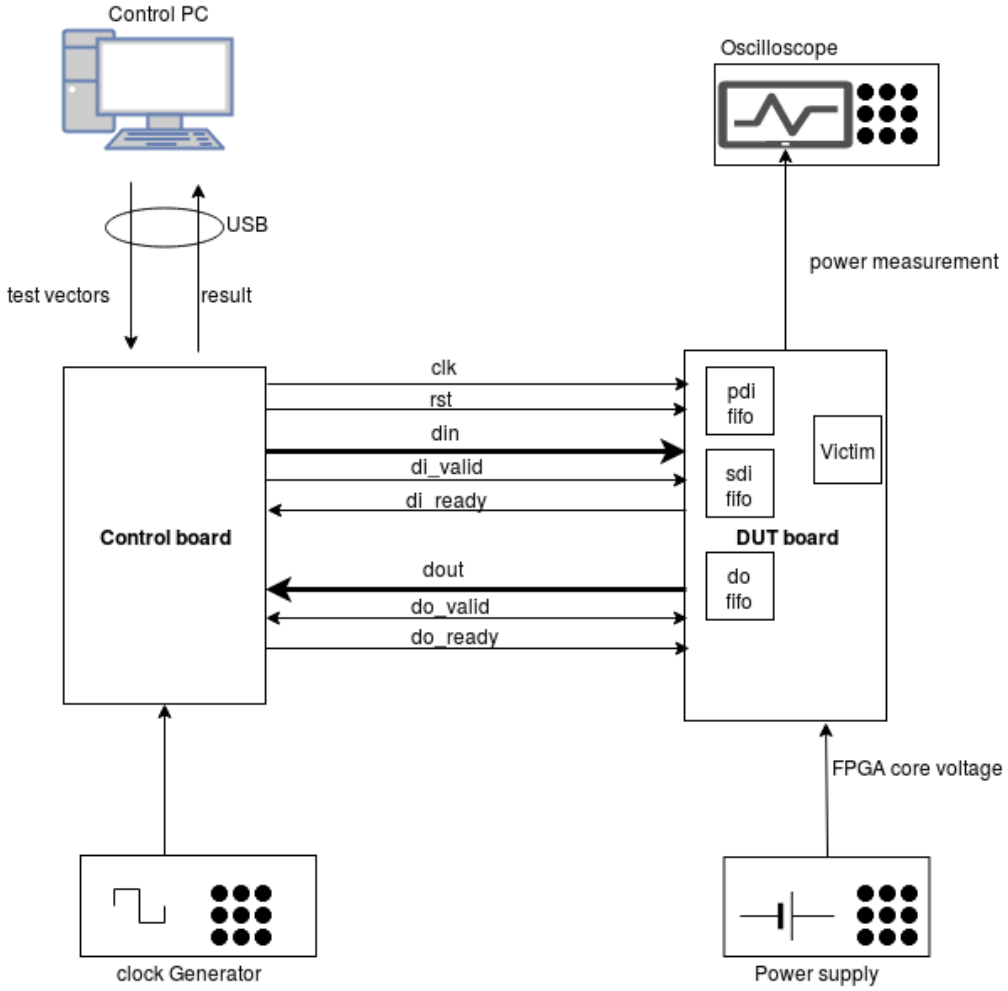
DUT Algorithm Development

This document describes how to interface the DUT (Design Under Test) or victim to the DUT Wrapper. The DUT is the algorithm to be attacked or tested.
The DUT or victim algorithm is user provided. However, the DUT wrapper is included with FOBOS. The DUT Wrapper handels communication to the control board and includes FIFOs to store input data for the DUT along with output FIFO.



Data flow description:

Test vectors are sent form PC one at a time to the control board which stores them breifly in a FIFO. The PC sends a command indicating test vector is complete. This will initiates the process of sending the data from the controller to the DUT through the interface shown in the figure above.
The DUT wrapper then puts data in the correct FIFOs (PDI, SDI and RDI).
Once the DUT wrapper receives the start command from the controller, it de-assersts the reset signal and the DUT will run and use the data in the FIFOs. The output of the DUT is stored in the DO fifo. Once the DO FIFO accumulates EXPECTED_OUTPUT bytes, the DUT wrapper will send this data to the control board which forwards it to the PC.

The DUT Wrapper – DUT interface

The protocol follows a simle AXI stream protocol.
The DUT (victim) is instantiated as follows in the FOBOS_DUT.vhd file

```
victim: entity work.victim(behav)

-- Choices for W and SW are independently any multiple of 4, defined in generics above

--    generic map  (
      G_W         => W, -- pdi and do width (mulltiple of 4)
       G_SW        => SW -- sdi width (multiple of 4)
--    )
port map(
          clk => clk,
          rst => start,  --! The FOBOS_DUT start signal meets requirements for synchronous resets
used in
                       --! CAESAR HW Development Package AEAD
-- data signals
          pdi_data  => pdi_data,
          pdi_valid => pdi_valid,
          pdi_ready => pdi_ready,

   sdi_data => sdi_data,
          sdi_valid => sdi_valid,
          sdi_ready => sdi_ready,
          do_data => result_data,
          do_ready => result_ready,
          do_valid => result_valid

----! if rdi_interface for side-channel protected versions is required, uncomment the rdi interface
--   ,rdi_data => rdi_data,
--   rdi_ready => rdi_ready,
--          rdi_valid => rdi_valid
);
```

The generic W is the PDI and DO width.
The generic SW is the SDI width.

It is highly recommended that the DUT is tested using the sources/dut/fobos_dut_tb.vhd test bench and ensure that the result data in the do port is valid. This testbench needs one test vector to be stored in the file dinFile.txt (see testVectorGeneration in doc/QickStart) .