JOINT ICTP-IAEA SCHOOL ON FPGA-BASED SOC

LUCIO J. **MARTÍNEZ GARBINO**

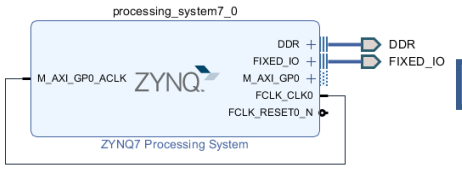
COMISIÓN NACIONAL DE ENERGÍA ATÓMICA
ARGENTINA

OUTLINE

- The School Scope
- Final Project
- New Topics
- Useful IP-COREs

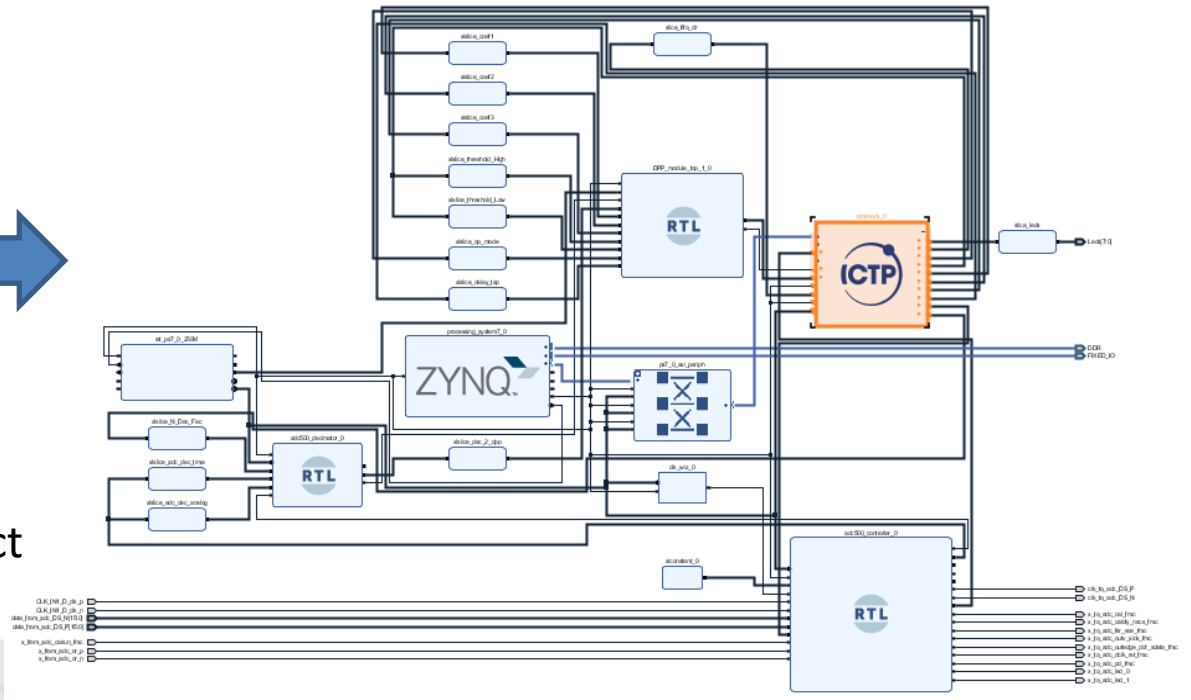
THE SCHOOL SCOPE

A big jump: from the hello world project to a full DAQ System



First Lab: Hello world

Last Lab: integration project



FINAL PROJECT (MY SOLUTION)

PL-SIDE

1- started from adc 500 project

2- modify the comblock adding 7 registers for the DPP configuration and disable the output FIFO

3- add the DPP source files (modified) and put it in the BlockDesign as a component

4- finally modify the data stream connections: ADC-> DECIMATOR-> DPP-> FIFO_in

PS-SIDE

SDK: I used the same project that I used in the UDMA lab

PC-SIDE

I wrote a little UDMA script to automate DPP configuration and data acquisition

```
--FIFO_wr_en <= photon_arrival;  --TODO: Delay the "arrival_flag" the latency of the main FIR (138)
FIFO_wr_en <=  photon_arrival_delayed ;

private: block
  type TSTATE  is( S_WAIT_PHOTON , S_DELAYING );
  signal st_now,st_next : TSTATE ;
  signal cnt_now ,cnt_next  : unsigned(delay_tap_i'range);
begin
    demora_sync: process(clk_i)
    begin
        if rising_edge(clk_i) then
          if rst_i = '1' then
                st_now <= S_WAIT_PHOTON ;
            else
                st_now <= st_next ;
                cnt_now <=cnt_next ;
            end if;
        end if;
    end process demora_sync;

    demora_comb: process(st_now,cnt_now,photon_arrival )
    begin
      st_next <= st_now ; -- si no pasa nada se mantiene
      cnt_next <=cnt_now + to_unsigned(1,delay_tap_i'length) ;-- por defecto
      photon_arrival_delayed_aux <= '0'; -- salidas de un solo pulso de reloj de duracion
      CASE st_now IS
          WHEN S_WAIT_PHOTON=>
            if photon_arrival= '1' then
              cnt_next<= (others => '0'); -- reseteo el contador
              st_next <= S_DELAYING ;      -- voy a edsprar que pase la cantidad d ecilcos especificada
            end if;
          WHEN S_DELAYING    =>
            -- aca se podria usar look_ahead y evitar un pocod e aritmetica
            if cnt_now = unsigned(delay_tap_i)-to_unsigned(1,delay_tap_i'length) then
              -- genero pulsod e escritura de FIFO
              photon_arrival_delayed_aux <= '1';
              st_next <= S_WAIT_PHOTON ; -- vuelvo a esperar el siguiente pulso
            end if;
          WHEN OTHERS        =>
            st_next <= S_WAIT_PHOTON ;
      END CASE;
    end process demora_comb;
    photon_arrival_delayed <= photon_arrival when delay_tap_i=std_logic_vector(to_unsigned(0,delay_tap_i'length))
end block;
```

FSM for requested DPP Improvement

this FSM delays the pulse of the arrival_flag signal the number of clock cycles indicated in delay_tap_i

A basic UDMA script to automate DPP configuration and data acquisition

A small index to know the mapping of registers to component parameters or function

```
#-------------------------------------------------------
# Register    function        bits   Peripheral
#-------------------------------------------------------
#   0         Leds             7:0
#   1         Clear FIFO         0   ComBlock
#   2         op_mode          1:0   DPP         [REF 1]
#   3         threshold[H,L]  31:0   DPP
#   4         coef_1           4:0   DPP
#   5         coef_2           4:0   DPP
#   6         coef_3           4:0   DPP
#   7         delay_tap       15:0   DPP
#   8         control_reg     31:0   adc500_controller
#   9         N dec_factor    22:0   adc500_decimator

#      [REF 1]
# 0: Amplitude measurement
# 1: ADC debugging
# 2: FIR debugging
# 3: derivative FIR output


#cd /home/mlab21/.local/bin/
#./udma
# run_script ~/script_udma.txt
```

```
#---------------------------------------
# conection
#---------------------------------------
connect -s 192.168.0.5:5900

#---------------------------------------
# All leds On
x_write_reg 0 255

#---------------------------------------
# Fifo clean
x_write_reg 1 1
x_write_reg 1 0

#---------------------------------------
# DPP Operation mode
x_write_reg 2 0

#---------------------------------------
# DPP threshold
x_write_reg 3  1023

#---------------------------------------
# DPP Coefs
x_write_reg 4   1
x_write_reg 5   3
x_write_reg 6   5

#---------------------------------------
# DPP Tap Delay (time to peak)
x_write_reg 7   128

#---------------------------------------
#  Decimator Factor
x_write_reg 9   10

#---------------------------------------
# Read FIFO
x_read_fifo -r d -f salida.txt 4096
```

NEW TOPICS

GIT + TCL + VIVADO

A useful things that I learned in this school is the use of tools like GIT and TCL and their interaction with VIVADO

Future Work

Free Open Source Software for FPGA design

USEFUL IP-CORES

Comblock

-systems configuration

-interface or abstraction layer

-perform rapid tests

Thanks for your attention!