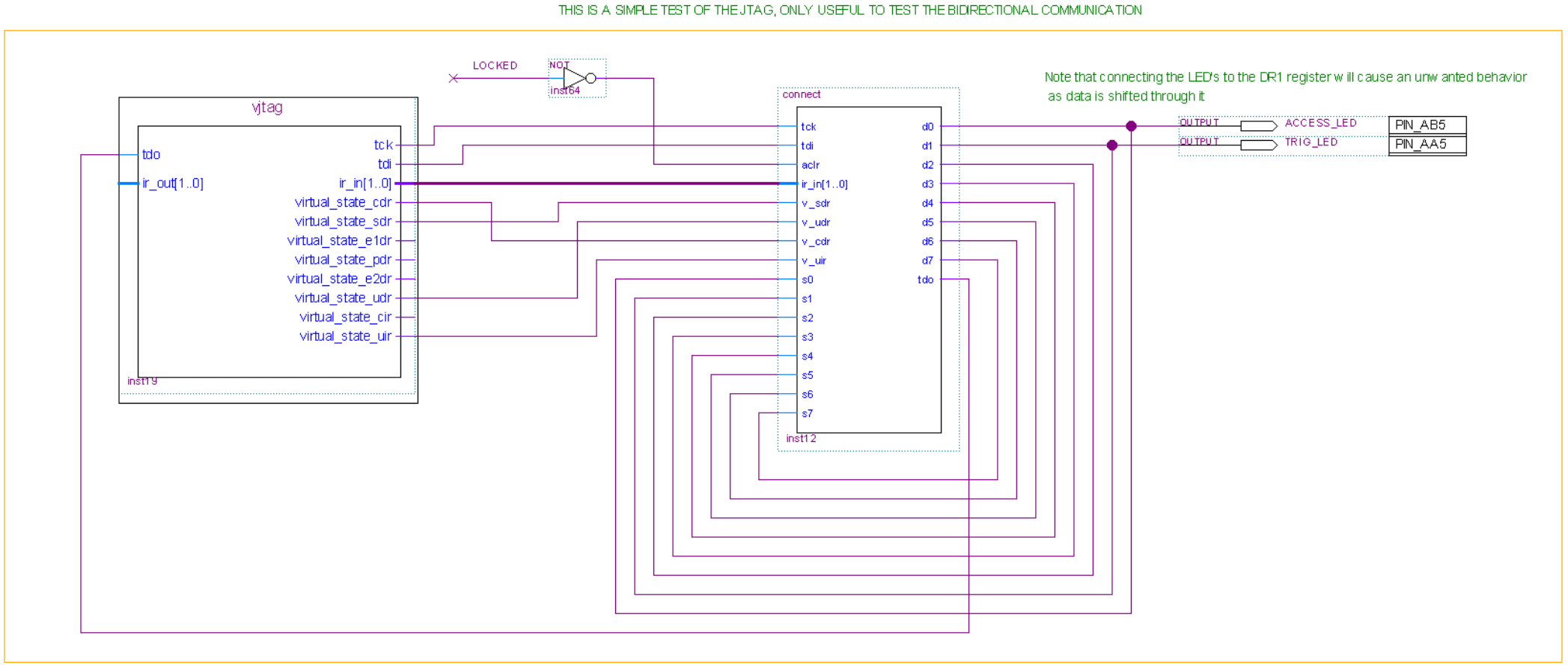
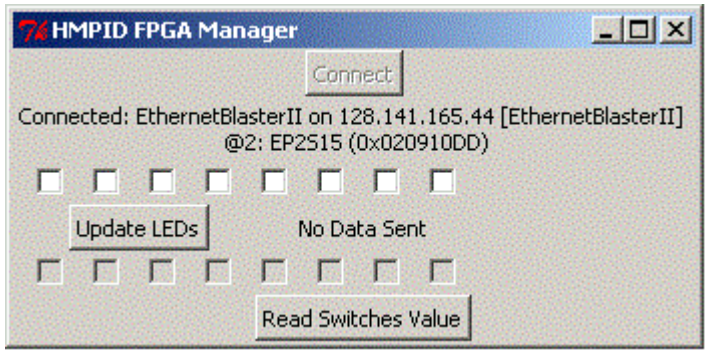
HMPID week meeting - JTAG communication development status



THIS IS A SIMPLE TEST OF THE JTAG, ONLY USEFUL TO TEST THE BIDIRECTIONAL COMMUNICATION. THE PROJECT IS BASED ON THE V26, WAS ONLY ADDED THE INTERFACE TO THE JTAG connected to the LED in our board.

In order to test was written a tcl script form.tcl, running the command "quartus_stp -t form.tcl"



Raúl Arteche Díaz (24/Jan/2019)

HMPID week meeting - JTAG communication development status

```
-- LEDs interface with JTAG
-- Based on a verilog code made by harrym 2014

-- INSTRUCTION:
-- 00: bypass
-- 01: read dip-switch
-- 10: update LEDs
-- 11: not used (=bypass)

-- VHDL version for HMPID JTAG Test Raul Arteche 2019

LIBRARY ieee;
USE ieee.std_logic_1164.all;
use IEEE.NUMERIC_STD.ALL;
use ieee.std_logic_unsigned.all;
use IEEE.math_real.all;

    entity connect is
    port(
        tck     : in    std_logic;
        tdi     : in    std_logic;
        aclr    : in    std_logic;
        ir_in : in  std_logic_vector(1 downto 0);
        v_sdr : in  std_logic;
        v_udr : in  std_logic;
        v_cdr : in  std_logic;
        v_uir : in  std_logic;

        s0      : in    std_logic;
        s1      : in    std_logic;
        s2      : in    std_logic;
        s3      : in    std_logic;
        s4      : in    std_logic;
        s5      : in    std_logic;
        s6      : in    std_logic;
        s7      : in    std_logic;

        d0      : out std_logic;
        d1      : out std_logic;
        d2      : out std_logic;
        d3      : out std_logic;
        d4      : out std_logic;
        d5      : out std_logic;
        d6      : out std_logic;
        d7      : out std_logic;
        tdo     : out std_logic
    );
end connect;
```

```
architecture rtl of connect is

    constant BYPASS : std_logic_vector(1 downto 0 ) := "00";
    constant KEY    : std_logic_vector(1 downto 0 ) := "01";
    constant LED    : std_logic_vector(1 downto 0 ) := "10";

    signal DR0      : std_logic_vector(1 downto 0 );
    signal DR1      : std_logic_vector(7 downto 0 );
    signal data_out : std_logic_vector(7 downto 0 ) := "00000000";

begin

tdo <= DR0(0) when (ir_in = BYPASS) else DR1(0);

d0 <=  data_out(0);
d1 <=  data_out(1);
d2 <=  data_out(2);
d3 <=  data_out(3);
d4 <=  data_out(4);
d5 <=  data_out(5);
d6 <=  data_out(6);
d7 <=  data_out(7);
```

```
process (tck)
begin
    if rising_edge (tck) then
        if(aclr = '0') then
            DR0 <= (others => '0');
            DR1 <= (others => '0');
        else
            case ir_in is
                when KEY =>
                    if(v_cdr = '1') then
                        DR1 <= s7 & s6 & s5 & s4 & s3 & s2 & s1 & s0;
                    else
                        if(v_sdr ='1') then
                            DR1 <= tdi & DR1(7 downto 1);
                        end if;
                    end if;

                when LED =>
                    if(v_sdr = '1') then
                        DR1 <= tdi & DR1(7 downto 1);
                    end if;

                when BYPASS =>
                    if(v_sdr = '1') then
                        DR0 <= tdi & DR0(1);
                    end if;

                when others =>
                    if(v_sdr = '1') then
                        DR0 <= tdi & DR0(1);
                    end if;
            end case;
        end if;
    end if;
end process;

process (v_udr)
begin
    if(ir_in = LED) then
        data_out <= DR1;
    end if;
end process;

end rtl;
```

Raúl Arteche Díaz (24/Jan/2019)

HMPID week meeting - JTAG communication development status

```
43  proc open_port {} {
44      global blaster_name
45      global test_device
46      open_device -hardware_name $blaster_name -device_name $test_device
47  }
48
49  proc close_port {} {
50      catch {device_unlock}
51      catch {close_device}
52  }
53
54  proc connect_jtag {} {
55      global blaster_name
56      global test_device
57      global displayConnect
58
59      foreach hardware_name [get_hardware_names] {
60
61          if { [string match "USB-Blaster*" $hardware_name] } {
62              set blaster_name $hardware_name
63          }
64
65          if { [string match "EthernetBlasterII on 128.141.165.44*" $hardware_name] } {
66              set blaster_name $hardware_name
67          }
68      }
69
70      foreach device_name [get_device_names -hardware_name $blaster_name] {
71          if { [string match "@1*" $device_name] } {
72              set test_device $device_name
73          }
74      }
75      set displayConnect "Connected: $hardware_name \n $device_name"
76      .btnConn configure -state disabled
77      .btnSend configure -state active
78      .btnRead configure -state active
79  }
80
81
```

Raúl Arteche Díaz (24/Jan/2019)

Conclusions:

- Is probed the possibility to communicate with the JTAG EthernetBlasterII using a TCL script.

- If the test of the LEDs example work as expected, will probe the bidirectional communication using the JTAG.

- A full VHDL implementation is possible of this example allowing the integration in the new firmware.

- The final data o parameter to monitor using the JTAG need to be discussed in the future.