Yash Malviya
2016CS50403
Hardik Khichi
2016CS50404

COL 215 – Mini Project

CRC Computation

Note: The problem statement for mini project was changed after submission of lab report 3. In accordance to that we had to make some changes to our implementation starting from the scratch. So, many points in this report may not match with the previously submitted reports.

1)Specifications:

The user will enter n (=10) 16-bit binary numbers using the switches on the board. One input is registered after the user presses button no.1. These numbers will be stored one by one in a B-ram cell that can store 10 16-bit vectors.

If the user presses button no.3, a number form B-ram cell will be input to the CRC calculation entity, which will further display the hex value of the calculated checksum.

If the user presses button no.4, a new number will be input to the CRC calculation entity but its 7^{th}-bit will be inverted. Similarly, if user presses button no.5, the number's 12^{th}-bit will be inverted. Finally, its checksum is displayed.

If the user presses button no.2 (RESET Button), all the numbers in the B-ram cell will be removed. The SSD will start displaying the seed value (i.e. ffff).

2)Description of the major subsystems/components and interconnecting signals:

The project contains four major components/entities:

1) B-RAM component

addr = To give location to input/output number in B-ram cell.

clk = Clock signal for B-ram cell.

din(16 bit) = Signal for number input.

dout(16 bit) = Signal for number output.

en = Enable(1)/Disable(0) B-ram cell.

we = Write enable.

2)CRC Calculation unit

clk= Clock.
pb1 = To load the 16-bit inputs in a B-ram (for n = 10).
pb2 = To reset the CRC calculation, reset the checksum to the seed value.
pb3 = To start the CRC computation with no error introduced.
pb4 = To start the CRC computation with error at the 7th bit.
pb5 = To start the CRC computation with error at the 12th bit.

inp_bram_data(16 bit) = Input from input B-ram cell.

crc_bram_data(16 bit) = Input from CRC B-ram cell.

data_inp(16 bit) = Input from switches.

crc_disp(16 bit) = Calculated CRC to be displayed on SSD from CRC Bram.

wr_crc(16 bit) = Input for input B-ram cell.

wr_inp(16 bit) = Input for CRC B-ram cell.

done = Shows if calculation is complete.

resetting = Shows when reset(pb2) is pushed.

started = Shows when CRC calculation starts(pb3/pb4/pb5 is pressed).

reading = Shows wait time of 3 cycles in reading from B-ram.

we_inp = Write enable for input B-ram.

we_crc = Write enable for CRC B-ram.

addr_inp = Address for input B-ram.

addr_crc = Address for CRC B-ram.

3) 7 Segment Display

Copied from the previous labs.

4) Port Map Unit

switches = Input from switches.

pb(5 bit) = Input from 5 buttons.

clk = Clock.

w_addr_inp(4 bit) = Shows write location in input B-ram.

r_addr_inp = Shows read location in input B-ram.

w_addr_crc = Shows write address in CRC B-ram.

done = Shows if calculation is complete.

rsting = Shows when reset(pb2) is pushed.

started = Shows when CRC calculation starts(pb3/pb4/pb5 is pressed).

reading = Shows wait time of 3 cycles in reading from B-ram.

anode = For SSD

cathode = For SSD

3) Validation

1) To check the correctness of our calculated checksum, we input random 16 bit values on fpga and matched its checksum values with online 16bit CRC-CCITT calculators.
2) We checked the proper working of B-ram cells and CRC calculations using vivado simulation. We noted read delay is 3 clock cycles and write can be successfully done in 1 clock cycle. For bigger BRAM cells write enable is divided for each four bit block.
3) As SSD is taken from previous labs its validation is not required.