Experiment 2
Four Bit Divider

Mayur Ware, 19D070070
EE-214, WEL, IIT Bombay

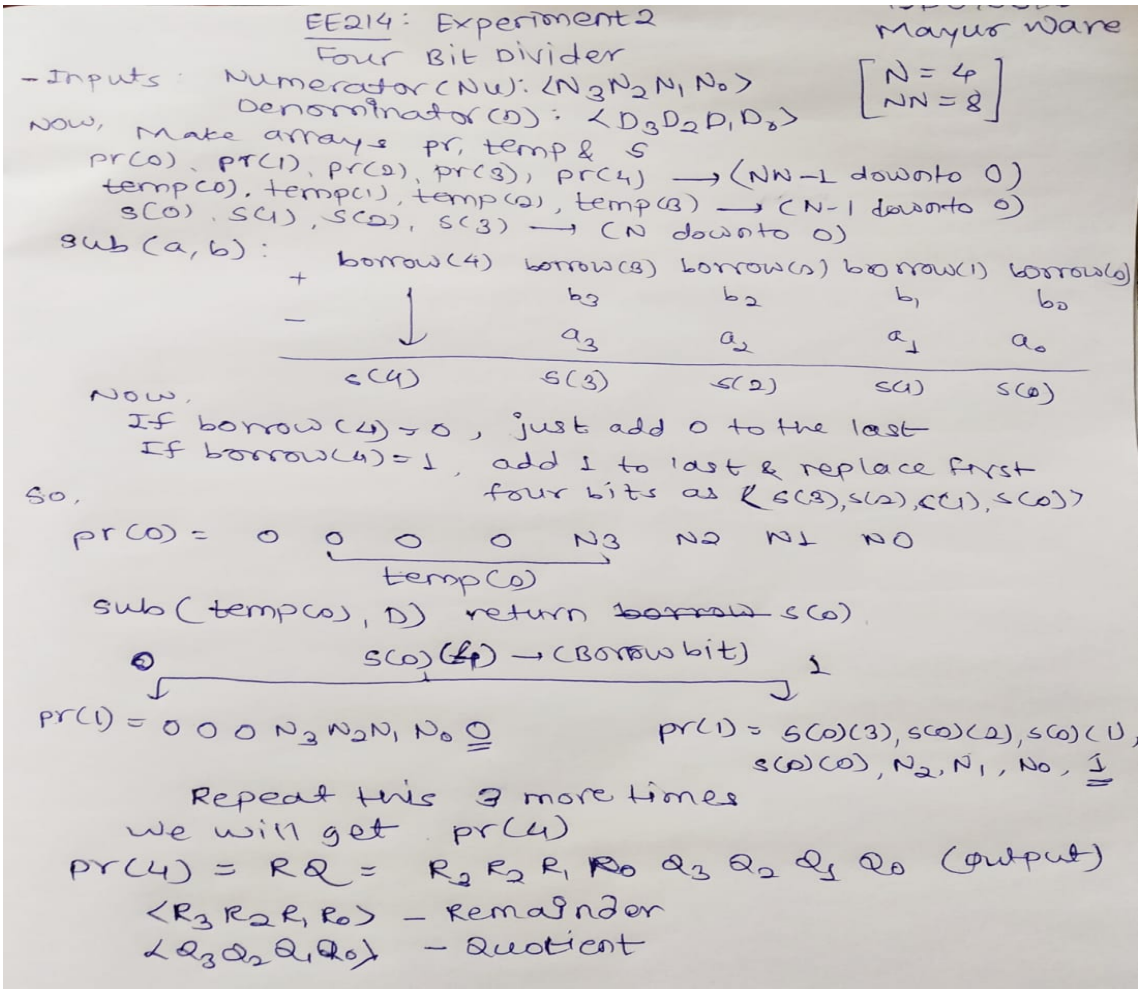
March 11, 2021

Overview of the experiment

- The purpose of the experiment is to design a a block which divides a 4 bit number by another 4 bit number.
- The logic of the experiment was given to us, we only had to implement it using behavioral VHDL. For which I used a function sub which takes two 4 bit numbers as input and gives a 5 bit number as output with the msb being the borrow bit.
  Then, using for and if loops, I implemented the logic.
- The report contains a handmade sheet which describes my approach, some important extracts of VHDL code, the output waveforms and Krypton Observations.

Approach to the experiment

First, I assigned "0000$N_3N_2N_1N_0$" to pr(0).
Then, I used a for loop i in 0 to N-1 which assigns temp(i) as pr(i)(6 downto 3) and subtracts $D_3D_2D_1D_0$ from temp(i) using sub function and stored it's output in s(i) with s(i)(4) being the borrow bit.
I again used a nested for loop i in 7 to 1 which left shifts pr(i)(7 downto 1) and stores it's value in pr(i+1) and also assigns pr(i+1)(0) = s(i)(4).
Then, I used an if loop which checks if s(i)(4) is '1' and then assigns the upper byte of pr(i+1) with s(i)(3 downto 0).
Finally, I assigned RQ the value of pr(4).



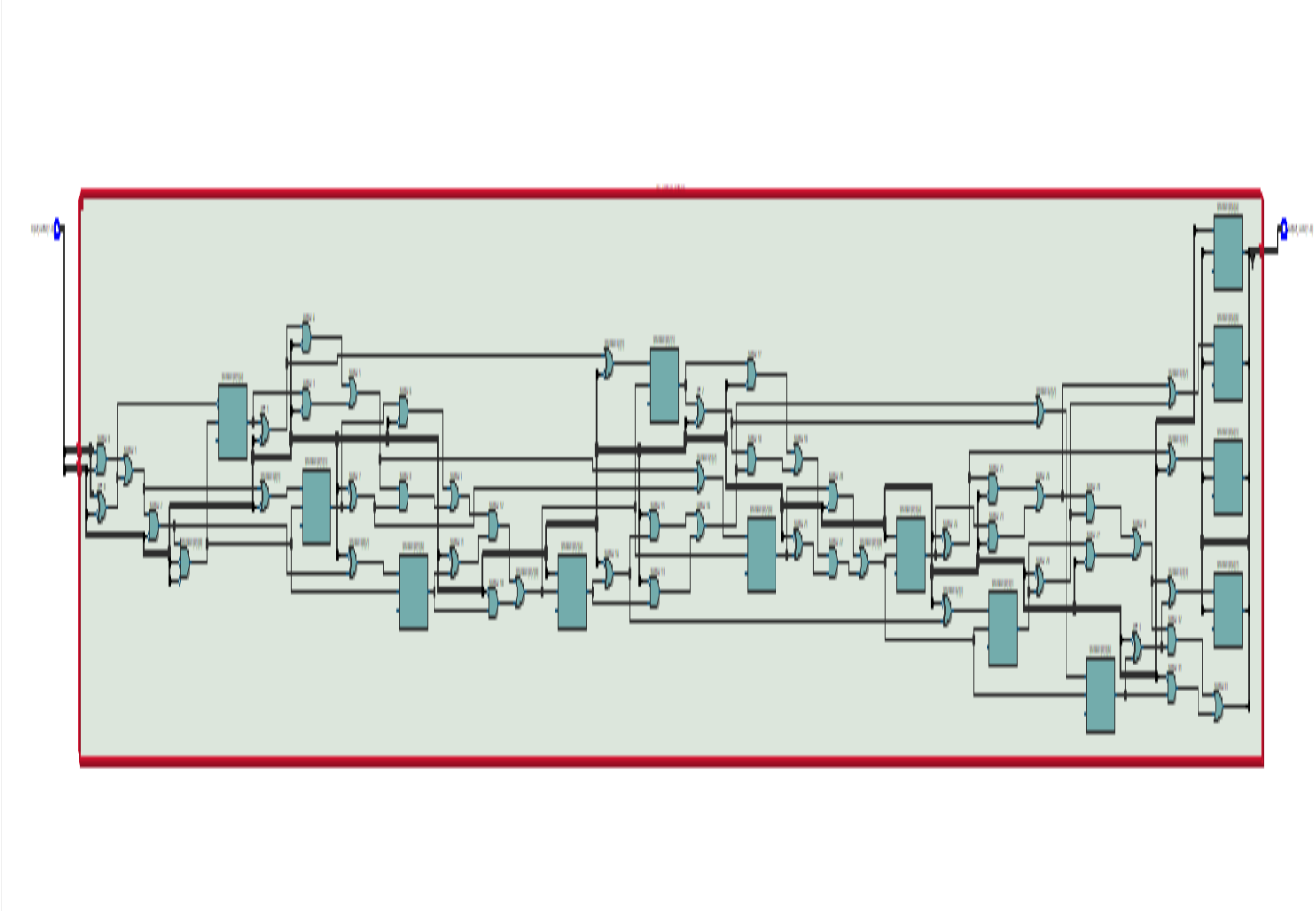
Design document and VHDL code if relevant

Main Entity :

```
entity Div_4Bit is
generic(N : integer:=4;
NN : integer:=8);
port ( Nu: in std_logic_vector(N-1 downto 0);
D: in std_logic_vector(N-1 downto 0);
RQ: out std_logic_vector((NN)-1 downto 0));
end Div_4Bit;
pr(0)(NN-1 downto N) := "0000";
pr(0)(N-1 downto 0) := Nu;

    for i in 0 to N-1 loop
temp(i) := pr(i)(NN-2 downto N-1);
s(i) := sub(temp(i), D);
for n in NN-1 to 1 loop
pr(i+1)(n) := pr(i)(n-1);
end loop;
pr(i+1)(0):= s(i)(N);
if s(i)(N) = '1' then
pr(i+1)(7 downto 4) := s(i)(3 downto 0);
end if;
end loop;
RQ <= pr(N);
```

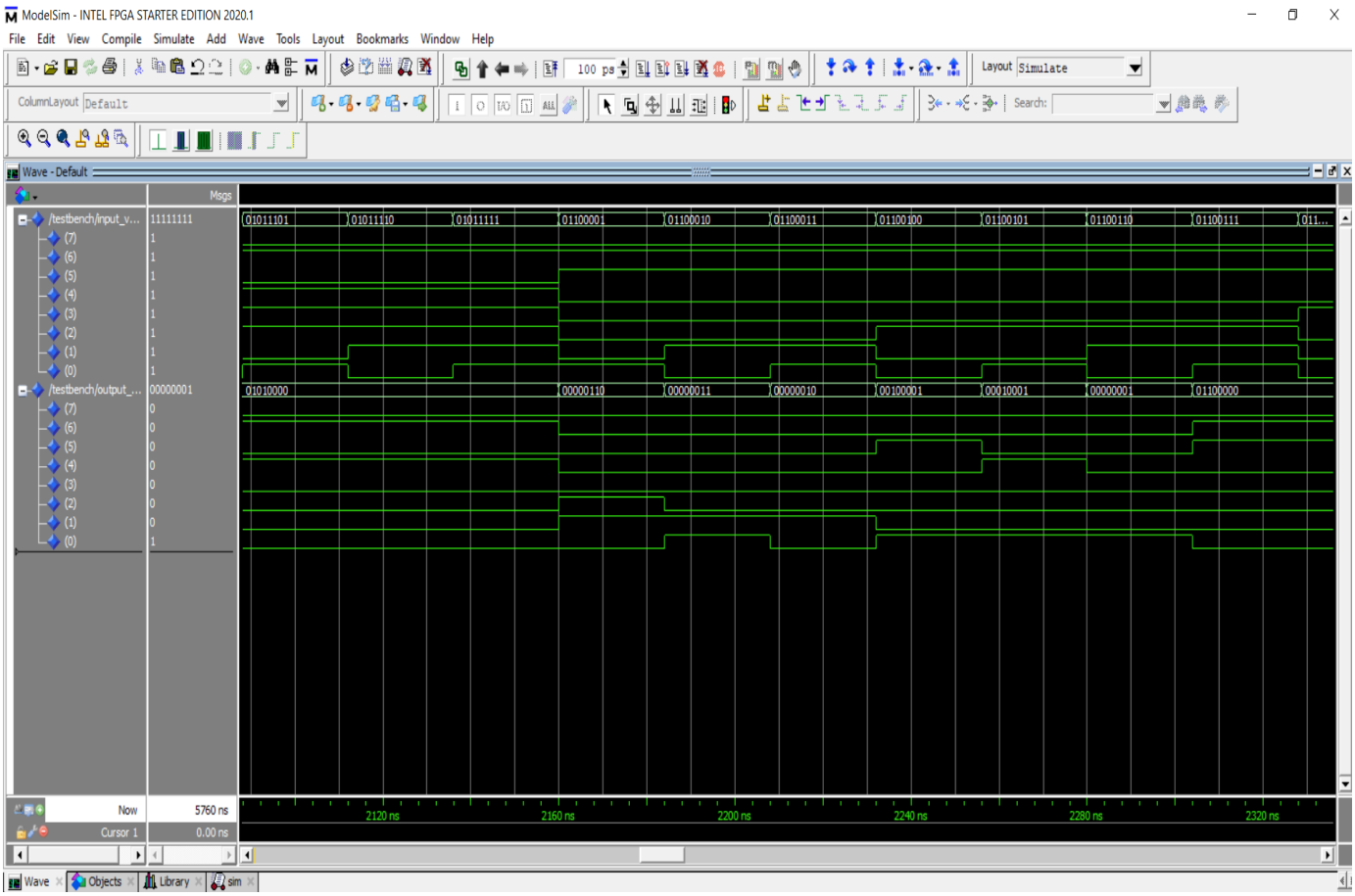
RTL View



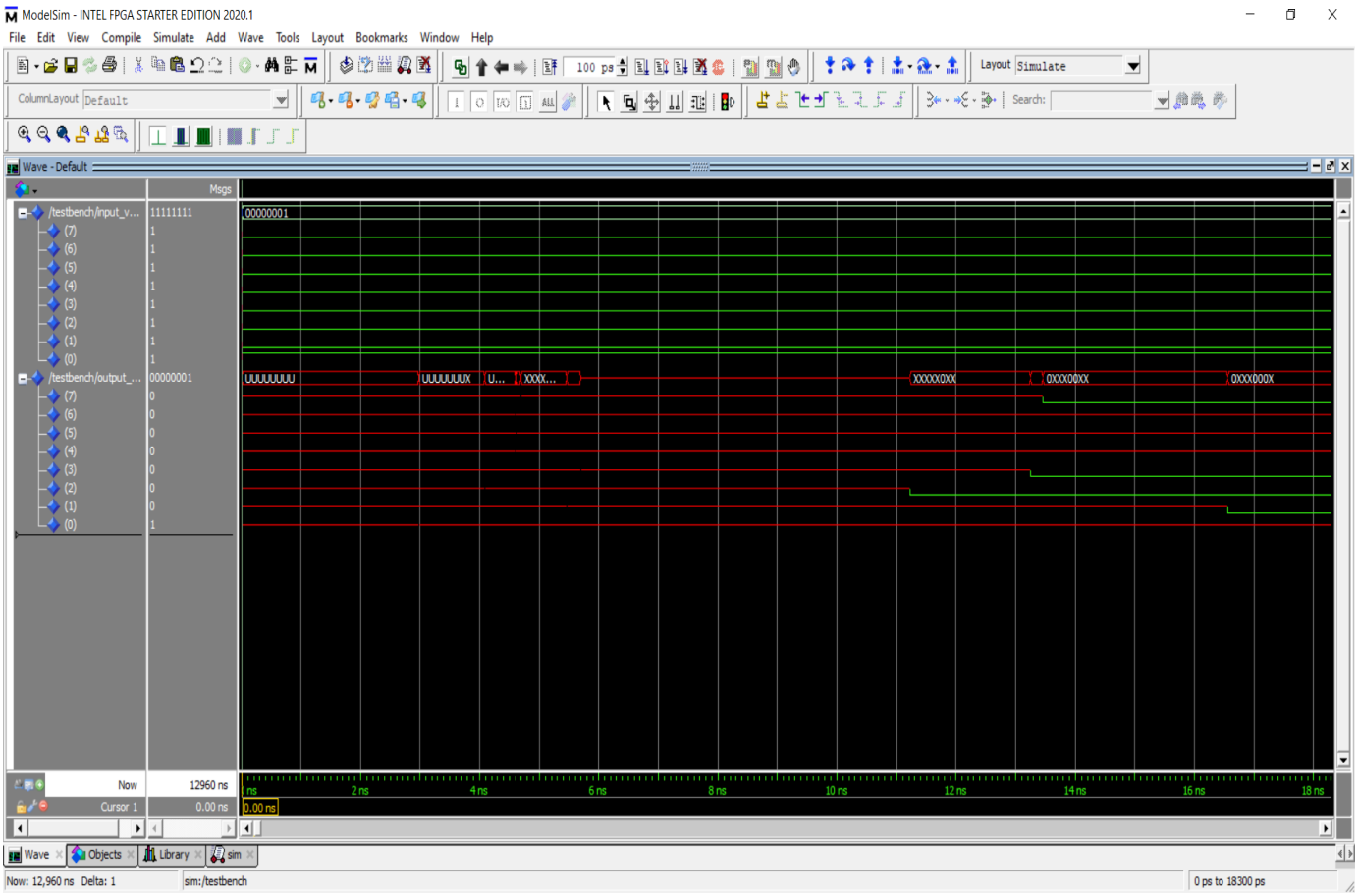
DUT Input/Output Format

```
Nu: in std_logic_vector(N-1 downto 0);          –Numerator
D: in std_logic_vector(N-1 downto 0);          –Denominator
RQ: out std_logic_vector((NN)-1 downto 0);          –Output
01000101      01000000      11111111          –Input Bits, Output Bits, Mask Bits
```

RTL Simulation



Gate-level Simulation



Krypton board

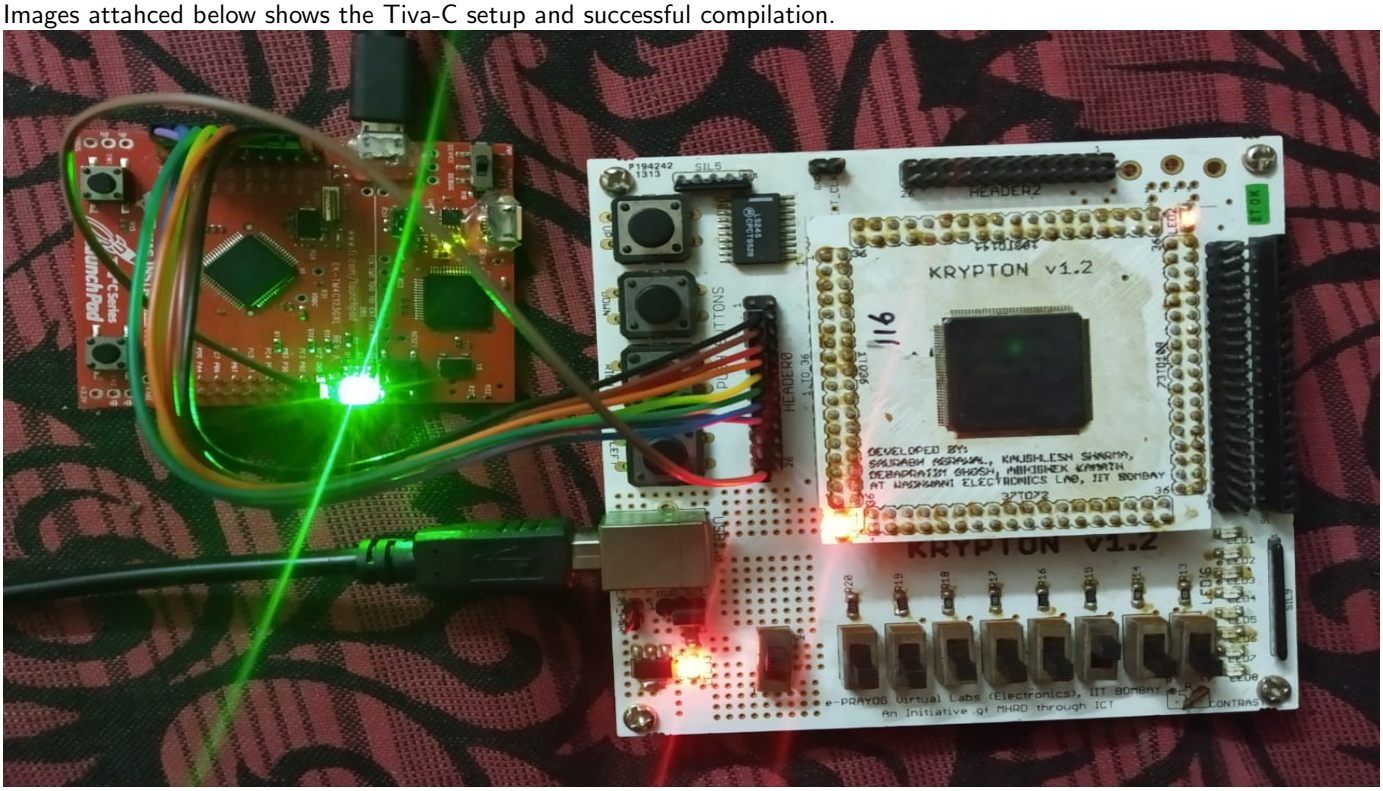
—Nil—

Observations

—Nil–

Tiva C

After generating all 16 testcases in in.txt, Tiva-C compiled outputs in output.txt.
Images attahced below shows the Tiva-C setup and successful compilation.



.

```
C:\Windows\system32\cmd.exe
#------ Command -  476 : RUNTEST 1 MSEC ------#


#------ Command -  477 : SDR 8 TDI(FE) 8 TDO(11) MASK(FF) ------#

Successfully entered the input..
Sampling out data..
FF
Output Comparison :  Success

#------ Command -  478 : RUNTEST 1 MSEC ------#


#------ Command -  479 : SDR 8 TDI(FF) 8 TDO(01) MASK(FF) ------#

Successfully entered the input..
Sampling out data..
FF
Output Comparison :  Success

#------ Command -  480 : RUNTEST 1 MSEC ------#

Sampling out data..
FF
Output Comparison :  Success
OK. All Test Cases Passed.
Transaction Complete.

E:\4. SPRING 2021\EE214\EXP 2\Div_4Bit>
```

References

Reference code and block diagrams given in Experiment 2 document.
Scan_Chain.pdf