By: Tom kessous 206018749

Dan Ben Ami 316333079

ECC Encoder & Decoder

Digital Design and Logical Synthesis for Electric Computer Engineering

(36113611)

Course Project

Synthesis

Version 1.0

Version 1.0 27,Dec,2021

Revision Log

Rev	Change	Description	Reason for change	Done By	Date
0.1	Initial document			Tom Kessous	27,Dec,2021
0.2					

Table of Content

LIST OF FIGURES

LIST OF TABLES

1. BLOCKS RTL DESCRIPTION

This module is error correction encoder & decoder, which communicate the CPU via APB bus. Our module can identify up to 2 errors (bits flipped) in a word and correct 1 error in a word. There are 3 operations modes to the system: Encode, Decode and Full channel.

- In the **encode** operation the input data contain uncoded word, the module will encode it (add parity bits) and put the codeword result on the output.
- In the **decode** operation, the input data contain a codeword with up to two errors. The module will decode the data and output the num of errors, If the num of errors was 0 or 1 the module will output the corrected data.
- In the **full channel** operation, the input data will encode in the Encoder module. From there, it will move to the Noise Adder module which add noise to it according to the data from the noise register. Afterwards, the noisy codeword will decoded in the decoder module. The module output the num of errors, and if there are less then two errors the module will also output the corrected data.

Our module designed to be generic in term of data width and APB bus width. The module is written in system Verilog and designed to be fast as possible while consuming low power.

1.1.1 Top level ECC ENC DEC

The top level entity include all the entities descript in the figure below in blue and connect all of them. Moreover it contains the two mux that control the operation mode of the module.

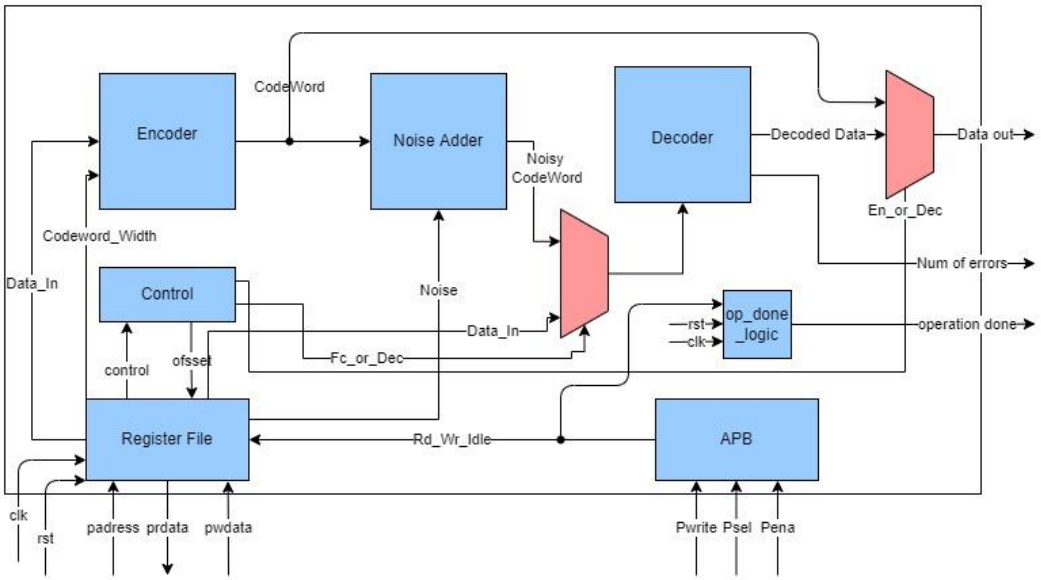

Figure 1: Top level design

Name	Mode	Type	Bounds	Delay	comment
PADDR	Input	Logic	[AMBA_ADDE_WIDTH-1:0]		APB Address bus
PENABLE	Input	Logic			APB Bus Enable/clk
PSEL	Input	Logic			APB Bus Select
PWDATA	Input	Logic	[AMBA_WORD-1:0]		APB Write Data Bus
PWRITE	Input	Logic			APB Bus Write
clk	Input	Logic			system clock
rst	input	Logic			Asynchronous Reset active low
PRDATA	Output	Logic	[AMBA_WORD-1:0]		APB Read Data Bus
Data_out	Output	Logic	[DATA_WIDTH-1:0]	1	Encoded/Decoded data (Valid when operation_done is asserted)
Operation_done	Output	Logic			Indicates an operation is finished. (Pulse , asserts for one cycle)
Num_of_errors	Output	Logic	[1:0]		numer of bit errors after decode operation (valid only after decode/full channel operations)

Table 1: Top level interface.

1.1.2 Encoder

Encoder unit receive a word to encode (Data_In) from the register file unit and encode the word according to DATA_WIDTH parameter and Codeword_Width. The calculation of the parity bits is made by modules encode_H1, encode_H2 and encode_H3 each module compute the parity bits with respect to matrices H1,H2,H3. The output of the encoder unit is CodeWord in size 8,16,32 depend on Codeword_Width.

Name	Mode	bounds	comment
Data_In	In	[DATA_WIDTH-1:0]	Word to encode
CodeWord_Width	In	[1:0]	Size of the encoded word
CodeWord	Out	[DATA_WIDTH-1:0]	The word with parity bits

Table 2: Encoder interface.

1.1.3 Decoder

Decoder unit receive a CodeWord from the mux that choose the CodeWord base on the current task of the system. If the current task is "decode" then the input to the decoder is Data_In from register file, Else the current task of the system is "Full Channel" then the input to the decoder is NoisyCodeWord.

The decoder build from few entities:

- Mat_mult entity which it self build from 3 entities:
    - Mat_mult_1 in case of DATA_WIDTH parameter is 8, responsible to calculate H1 matrix multiply by codeword vector in length of 8 bits.
    - Mat_mult_2 in case of DATA_WIDTH parameter is 16, responsible to calculate H1 and H2 matrix multiply by codeword vector in length of 8 or 16 bits depend on Codeword_Width.
    - Mat_mult_3 in case of DATA_WIDTH parameter is 32, responsible to calculate H1 and H2 and H3 matrix multiply by codeword vector in length of 8 or 16 or 32 bits depend on Codeword_Width.

- Mux_H1 entity output the decoded data and num of errors based on the mat_mult result from H1 matrix.
- Mux_H2 entity output the decoded data and num of errors based on the mat_mult result from H2 matrix.
- Mux_H3 entity output the decoded data and num of errors based on the mat_mult result from H3 matrix.

The decoder can fix one bit flipped (one error in the CodeWord) and recognize two bits flipped (two errors in the CodeWord). There are two outputs from the decoder: the number of errors in the noisy CodeWord, decoded data.

The decoded data from the decoder can be one of the above:

a. the word without the parity bits, in case there are no errors.

b. the fixed word in case of one error.

c. zeros in case of two errors.

Name	Mode	bounds	comment
NoisyCodeWord	In	[DATA_WIDTH-1:0]	CodeWord with errors
Codeword_Width	In	[1:0]	Indicate the Size of the CodeWord
Decoded_Data	Out	[DATA_WIDTH-1:0]	The fixed word
NumOfErrors	out	[1:0]	Number of errors

Table 3: Decoder interface.

1.1.4 Noise Adder

This unit receive Noise from the register file and receive CodeWord from the encoder. Then add (bitwise xor operation) the noise to the CodeWord. The output is the CodeWord with noise namely NoisyCodeWord.

Name	Mode	bounds	comment
noise	In	[DATA_WIDTH -1:0]	Noise from register file
CodeWord	In	[DATA_WIDTH -1:0]	CodeWord from encoder
NoisyCodeWord	Out	[DATA_WIDTH-1:0]	CodeWord with noise

Table 4: Noise Adder interface.

1.1.5 Control

The control unit generate control signals according to the value in the Control register. There are two control signals, Full channel/decode (namely FC_or_Dec) and encoder/decoder (namely En_or_Dec). The first one control the input data to decoder unit via mux, The second control the output of the system via mux.

Name	Mode	bounds	comment
CTRL	In	[1:0]	Value of control register
FC_or_Dec	out		Control data input of decoder
En_or_Dec	Out		Control system output

Table 5: Control interface.

1.1.6 Register File

Register file have 4 registers. In case of reset occurs we write zeros to all registers. In case of write operation the data is written to the desired register according to the address. In case of read operation we access the desired register based on the address and output its content.

Name	Mode	bounds	comment
clk	In		System clock
rst	In		Asynchronous Reset active low
Rd_Wr_Id	In	[1:0]	Operation select: read/write/idle
offset	In	[AMBA_ADDR_WIDTH-3:0]	Address of the register
Data_to_reg	In	[AMBA_WORD-1:0]	The data to write in the register
Data_out	Out	[AMBA_WORD-1:0]	The read data from register
AMBA_WORD_registers[1:0]	Out	[AMBA_WORD - 1:0]	Data & Noise registers
two_bits_registers[1:0]	Out	[1:0]	Control & CodeWord width registers

Table 6: Register file interface.

1.1.7 APB

APB unit receive pwrite, psel and penable and generate Rd_Wr_Id signal which indicate to the register file which operation to execute. If the Rd_Wr_Id is 0 then read operation will execute, If the Rd_Wr_Id is 1 then write operation will execute and If the Rd_Wr_Id is 2 then no read nor write operation will execute (idle state).

Name	Mode	bounds	comment
pwrite	In		APB protocol signals
psel	In		APB protocol signals
penable	In		APB protocol signals
Rd_Wr_Id	Out	[1:0]	Signal that indicate which operation (write,read or idle)

Table 7: APB interface.

1.1.8 Op_done_logic

This unit responsible to generate the operation done signal. Operation done rise to '1' for 1 cycle after the data_out is valid.

Name	Mode	bounds	comment
clk	In		System clock
rst	In		Asynchronous Reset active low
Rd_Wr_Id	In	[1:0]	Signal that indicate which operation (write,read or idle)
Operation done	Out		Rise to '1' for one cycle after the data_out is valid

Table 8: Op_done_logic interface.

1.1.9 Module Flow Chart

The flow chart below describe the data flow in our system:

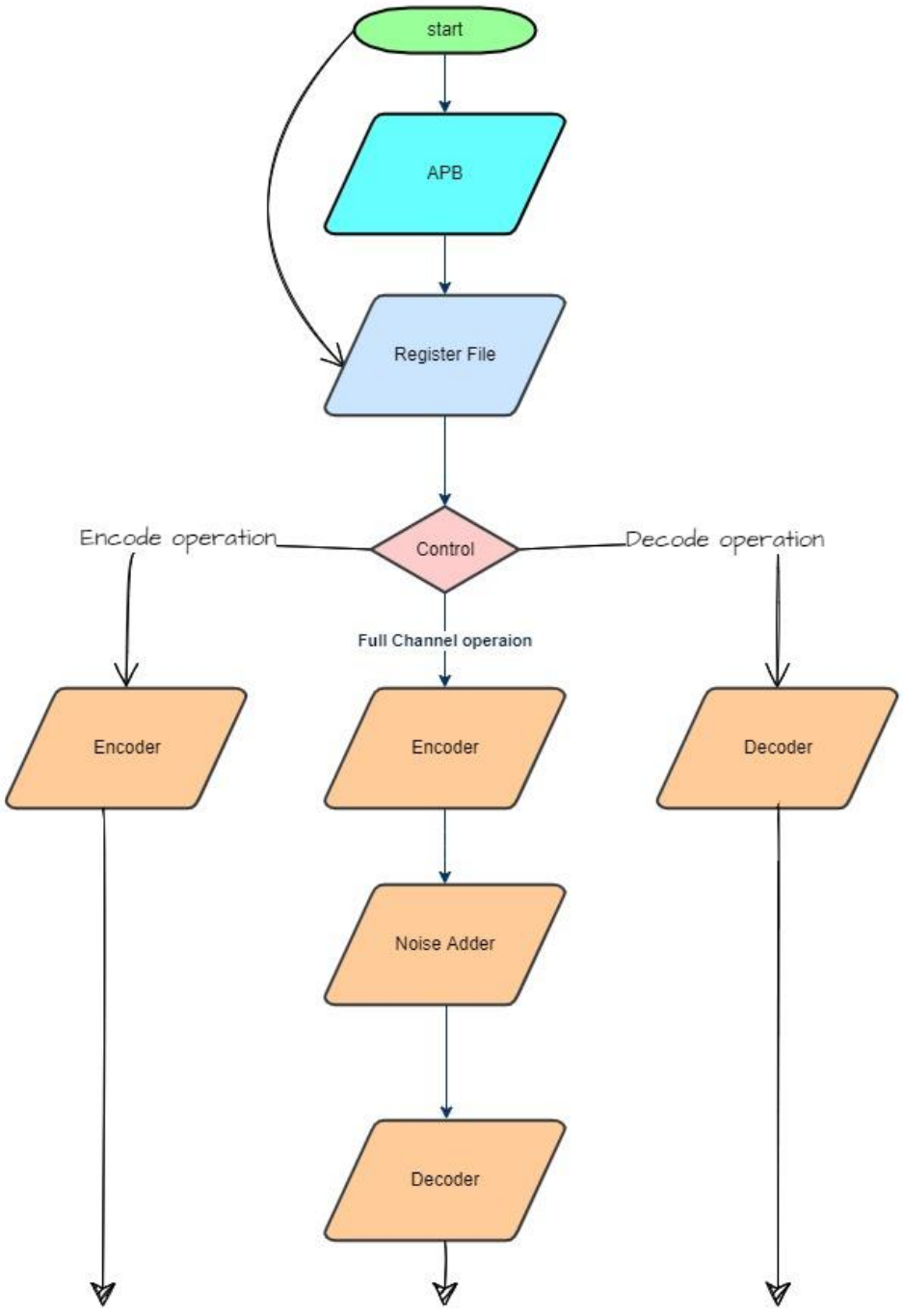

Figure 2: Flow Chart

2. SYNTHESIS

We performed a synthesis to all entities with Design Compiler (DC) program. In addition, we setup a few constraints to the complier in order to stimulate real word conditions and to achieve best performance, area and power.

After completing the synthesis flow, we downloaded the “output” and “report” directory to our own workstation. Then we opened the reports and performed analysis, which their results are described below.

2.1.1 Constraints

We set the following constraints:

Constraint Type	Value
Clock Period	14 ns
Clock frequency	71.4 Mega Hz
Clock uncertainty	Maximum 0.1 ns
Clock transition time	Maximum 0.1 ns
Input delay	Maximum 0.2 ns
Input transition	Maximum 0.1 ns
Output delay	Maximum 0.5 ns
Design area	Smaller than 1,000,000
Wire load model	tsmc18_wl50

Table 9: Constraints.

2.1.1 Area Report

The design area report:

```
****************************************
Report : area
Design : ecc_enc_dec
Version: O-2018.06-SP4
Date   : Sun Dec 26 19:21:06 2021
****************************************

Library(s) Used:

    slow (File: /users/agnon/grad/danbenam/DC/LibraryFiles/db/slow.db)

Number of ports:                          790
Number of nets:                          1838
Number of cells:                         1158
Number of combinational cells:           1073
Number of sequential cells:                69
Number of macros/black boxes:               0
Number of buf/inv:                        414
Number of references:                      35

Combinational area:              23713.905867
Buf/Inv area:                     6060.700873
Noncombinational area:            5691.470413
Macro/Black Box area:                0.000000
Net Interconnect area:          597102.975555

Total cell area:                 29405.376280
Total area:                     626508.351835
1
```

Figure 3: Area report

The area report includes the number of ports, combinational cells, sequential cells etc. it also includes the area of the combinational logic, non-combinational logic, buf/Inv and of course the total area (which includes also the net Interconnect).

The Hierarchical area report:

Hierarchical area distribution

Hierarchical cell	Global cell area Absolute Total	Percent Total	Local cell area Combi-national	Noncombi-national	Black-boxes	Design
ecc_enc_dec	29405.3763	100.0	4031.5968	0.0000	0.0000	ecc_enc_dec
U_apb	23.2848	0.1	23.2848	0.0000	0.0000	apb
U_control_unit	56.5488	0.2	56.5488	0.0000	0.0000	control_unit
U_decoder	10035.7489	34.1	1397.0880	0.0000	0.0000	decoder_DATA_WIDTH32
U_decoder/H_1	442.4112	1.5	442.4112	0.0000	0.0000	mux_H1
U_decoder/U_0	3585.8593	12.2	0.0000	0.0000	0.0000	mat_mult_DATA_WIDTH32
U_decoder/U_0/genblk1.mat_mult_3	3585.8593	12.2	3585.8593	0.0000	0.0000	mat_mult_3_DATA_WIDTH32
U_decoder/genblk1.H_2	1293.9696	4.4	1293.9696	0.0000	0.0000	mux_H2
U_decoder/genblk1.H_3	3316.4208	11.3	3316.4208	0.0000	0.0000	mux_H3
U_encoder	4397.5009	15.0	1297.2960	0.0000	0.0000	encoder_DATA_WIDTH32
U_encoder/genblk1.U_encode_H1	159.6672	0.5	159.6672	0.0000	0.0000	encode_H1_DATA_WIDTH32
U_encoder/genblk1.U_encode_H2	681.9120	2.3	681.9120	0.0000	0.0000	encode_H2_DATA_WIDTH32
U_encoder/genblk1.U_encode_H3	2258.6256	7.7	2258.6256	0.0000	0.0000	encode_H3_DATA_WIDTH32
U_noise_adder	1200.8304	4.1	1200.8304	0.0000	0.0000	noise_adder_DATA_WIDTH32
U_op_done_logic	96.4656	0.3	26.6112	69.8544	0.0000	op_done_logic
U_register_file	9563.4001	32.5	3941.7841	5621.6160	0.0000	register_file_AMBA_WORD32
Total			23713.9059	5691.4704	0.0000	

1

Figure 4: Hierarchical area report

The Hierarchical area report includes the area of the combinational logic, non-combinational logic and of course the total area of each module in the ecc_enc_dec (top).

As seen above, the register file is one of the biggest modules, that is one of the reasons we decided to minimize the use of registers and flip-flops in our design (only the register_file unit and the op_done_logic unit contain a non-combinational logic).

As seen above, most of the area is taking by the net Interconnect (597102 out of 626508), so if we had more time to do the project, we could save more area by trying to reduce the amount of wires and ports in the design.

It also can be seen that the decoder unit is taking 34% of the whole design (without the net Interconnect), so if we had more time to do the project, we would try to reduce the combinational logic in it.

2.1.2 Timing Report

As we can see from the report below the longest path is from the register file to data out. We designed our module such that the output is available one clock after writing the input data to the register file, In other words, we knew that the longest path will be from input (register file) to output (data out) of the module. Although the longest path is from input to output we can achieve relative high clock frequency of 71.4 mega Hz, this is the highest clock frequency that our module can operate. we tried to decrease the clock period until violation (slack = 0). If we would like to increase performance we could pipelined our module in to smaller blocks, each block perform different operation and all blocks can run simultaneously. In that type of design the longest path will be shorter and therefore the clock frequency will be higher but the output of the module will be available after a few cycles of the clock. Moreover, power and area will be bigger due to registers needed between each block. We can see from the report that we have no timing violations.

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : ecc_enc_dec
Version: O-2018.06-SP4
Date   : Sun Dec 26 19:21:06 2021
****************************************

Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

  Startpoint: U_register_file/two_bits_registers_reg[0][1]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: data_out[4]
            (output port clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  ecc_enc_dec        tsmc18_wl50           slow

  Point                                                   Incr       Path
  ---------------------------------------------------------------------------
  clock clk (rise edge)                                   0.00       0.00
  clock network delay (ideal)                             0.00       0.00
  U_register_file/two_bits_registers_reg[0][1]/CK (DFFRX4)
                                                          0.00       0.00 r
  U_register_file/two_bits_registers_reg[0][1]/Q (DFFRX4)
                                                          0.90       0.90 r
  U_register_file/two_bits_registers[0][1] (register_file_AMBA_WORD32)
                                                          0.00       0.90 r
  U_control_unit/CTRL[1] (control_unit)                   0.00       0.90 r
  U_control_unit/U4/Y (INVX8)                             0.20       1.10 f
  U_control_unit/U5/Y (AND2X4)                            0.31       1.41 f
  U_control_unit/FC_or_Dec (control_unit)                 0.00       1.41 f
  U240/Y (CLKINVX8)                                       0.11       1.53 r
  U237/Y (INVX8)                                          0.15       1.68 f
  U158/Y (CLKINVX3)                                       0.37       2.04 r
  U132/Y (INVX20)                                         0.34       2.38 f
  U244/Y (INVXL)                                          1.22       3.60 r
```

Figure 5: Timing report

```
U259/Y (AOI22X4)                                          0.31       3.91 f
U164/Y (CLKINVX4)                                         0.45       4.36 r
U_decoder/NoisyCodeWord[31] (decoder_DATA_WIDTH32)        0.00       4.36 r
U_decoder/U_0/NoisyCodeWord[31] (mat_mult_DATA_WIDTH32)
                                                          0.00       4.36 r
U_decoder/U_0/genblk1.mat_mult_3/NoisyCodeWord[31] (mat_mult_3_DATA_WIDTH32)
                                                          0.00       4.36 r
U_decoder/U_0/genblk1.mat_mult_3/U46/Y (XOR2X4)           0.55       4.91 r
U_decoder/U_0/genblk1.mat_mult_3/U44/Y (XNOR2X4)          0.51       5.42 r
U_decoder/U_0/genblk1.mat_mult_3/U37/Y (XOR2X4)           0.48       5.90 r
U_decoder/U_0/genblk1.mat_mult_3/U22/Y (XOR2X4)           0.50       6.40 r
U_decoder/U_0/genblk1.mat_mult_3/U21/Y (XOR2X4)           0.38       6.78 r
U_decoder/U_0/genblk1.mat_mult_3/U20/Y (XOR2X1)           0.69       7.48 r
U_decoder/U_0/genblk1.mat_mult_3/U14/Y (XOR2X4)           0.68       8.15 r
U_decoder/U_0/genblk1.mat_mult_3/U3/Y (OAI22X4)           0.25       8.40 f
U_decoder/U_0/genblk1.mat_mult_3/column[4] (mat_mult_3_DATA_WIDTH32)
                                                          0.00       8.40 f
U_decoder/U_0/column[4] (mat_mult_DATA_WIDTH32)           0.00       8.40 f
U_decoder/U27/Y (BUFX20)                                  0.47       8.87 f
U_decoder/genblk1.H_3/column[4] (mux_H3)                  0.00       8.87 f
U_decoder/genblk1.H_3/U49/Y (NOR2X2)                      0.52       9.39 r
U_decoder/genblk1.H_3/U5/Y (BUFX20)                       0.26       9.65 r
U_decoder/genblk1.H_3/U156/Y (NOR4BXL)                    1.44      11.08 r
U_decoder/genblk1.H_3/U35/Y (AOI2BB1X2)                   0.41      11.49 f
U_decoder/genblk1.H_3/U34/Y (INVX1)                       0.54      12.03 r
U_decoder/genblk1.H_3/U68/Y (AOI32X4)                     0.40      12.43 f
U_decoder/genblk1.H_3/U16/Y (CLKINVX3)                    0.20      12.63 r
U_decoder/genblk1.H_3/Decoded_Data[4] (mux_H3)            0.00      12.63 r
U_decoder/U11/Y (AOI22X4)                                 0.22      12.85 f
U_decoder/U30/Y (INVX8)                                   0.19      13.04 r
U_decoder/Decoded_Data[4] (decoder_DATA_WIDTH32)          0.00      13.04 r
U38/Y (AOI22X4)                                           0.20      13.23 f
U257/Y (INVX8)                                            0.16      13.40 r
data_out[4] (out)                                         0.00      13.40 r
data arrival time                                                   13.40

clock clk (rise edge)                                    14.00      14.00
clock network delay (ideal)                               0.00      14.00
clock uncertainty                                        -0.10      13.90
output external delay                                    -0.50      13.40
data required time                                                  13.40
-------------------------------------------------------------------------
data required time                                                  13.40
data arrival time                                                  -13.40
-------------------------------------------------------------------------
slack (MET)                                                         0.00
```

Clock settings:

```
****************************************
Report : clocks
Design : ecc_enc_dec
Version: O-2018.06-SP4
Date   : Sun Dec 26 19:21:06 2021
****************************************

Attributes:
    d - dont_touch_network
    f - fix_hold
    p - propagated_clock
    G - generated_clock
    g - lib_generated_clock

Clock          Period   Waveform            Attrs     Sources
------------------------------------------------------------------------------
clk             14.00   {0 7}               d         {clk}
------------------------------------------------------------------------------

****************************************
Report : clock_skew
Design : ecc_enc_dec
Version: O-2018.06-SP4
Date   : Sun Dec 26 19:21:06 2021
****************************************

                 Rise      Fall  Min Rise  Min fall        Uncertainty
Object           Delay     Delay     Delay     Delay     Plus      Minus
------------------------------------------------------------------------------
clk                 -         -         -         -         -       0.10

                 Max Transition      Min Transition
Object           Rise      Fall      Rise      Fall
------------------------------------------------------
clk              0.10      0.10      0.10      0.10
1
```

Figure 6: Clock settings

2.1.1 Power Report

In the report below we can see power consumption of our module. In order to improve the power we could write the register in that way that the compiler can optimize them with clock gating. Since we have filp flop just in the register files, the power that could be saved not that high.

```
Design : ecc_enc_dec
Version: O-2018.06-SP4
Date   : Sun Dec 26 19:21:06 2021
****************************************


Library(s) Used:

    slow (File: /users/agnon/grad/danbenam/DC/LibraryFiles/db/slow.db)


Operating Conditions: slow   Library: slow
Wire Load Model Mode: top

Design        Wire Load Model            Library
------------------------------------------------
ecc_enc_dec             tsmc18_wl50       slow


Global Operating Voltage = 1.62
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  = 405.1429 uW   (63%)
  Net Switching Power  = 242.8254 uW   (37%)
                         ---------
Total Dynamic Power    = 647.9683 uW  (100%)

Cell Leakage Power     =   1.5104 uW

                  Internal         Switching           Leakage            Total
Power Group       Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000              0.0000            0.0000  (   0.00%)
memory             0.0000           0.0000              0.0000            0.0000  (   0.00%)
black_box          0.0000           0.0000              0.0000            0.0000  (   0.00%)
clock_network      0.0000           0.0000              0.0000            0.0000  (   0.00%)
register           0.3725        6.8554e-03          2.1817e+05           0.3796  (  58.44%)
sequential         0.0000           0.0000              0.0000            0.0000  (   0.00%)
combinational   3.2662e-02          0.2360          1.2922e+06            0.2699  (  41.56%)
--------------------------------------------------------------------------------------------------
Total              0.4051 mW        0.2428 mW       1.5104e+06 pW         0.6495 mW
1
```

Figure 7: Power report

3. APPENDIX

3.1 Terminology

LSB - Least Significant Bit

TBR - To Be Reviewed

TBD - To Be Defined

3.2 References