Experiment #4 –
Accelerator and Wrappers

Sana Sari Navaei 810199435
Maryam Jafarabadi 810199549

1.Exponential Engine

1.

```
`include"exponential.sv"
`timescale 1ns/1ns
module exponentialTB();
    reg start = 0, clk = 0, rst = 0;
    reg[15:0] x = 16'b1010011001100110;   //e^0.65 = 1.9155408
    wire done;
    wire[1:0]intpart;
    wire[15:0]fracpart;

    exponential CUT(clk,rst,start,x,done,intpart,fracpart);

    always #10 clk = ~clk;
    initial begin
        #20 rst = 1'd1;
        #50 rst = 1'd0;
        #30 start = 1'd1;
        #3000 x = 16'b0010101110000101;    //e^0.17 = 1.18530485
        #3000 x = 16'b1110000101000111;    //e^0.88 = 2.41089970
        #13000 $stop;
    end
endmodule
```

Fig. 1 Testbench verilog

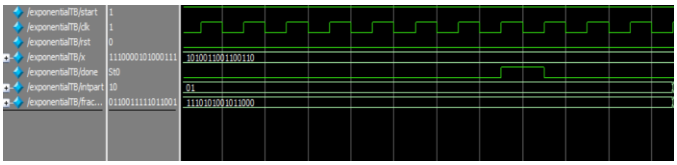

Fig. 2 simulation result

When x is 0.1010011001100110 in binary or 0.65 in decimal , the output is 01.11101001011000 which means 1.915405 but $e^{0.65}$ is 1.9155408 and the difference is 0.000003.

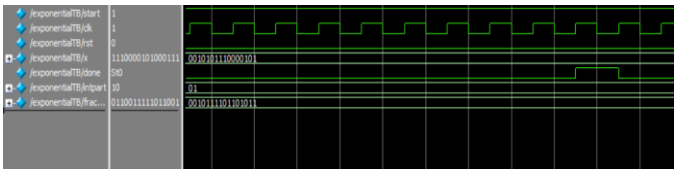

Fig. 3 simulation result

When x is 0. 0010101110000101 in binary or 0.17 in decimal , the output is 01.0010111101101011which means 1.185226 but $e^{0.17}$ is 1.18530485 and the difference is 0.00007885.

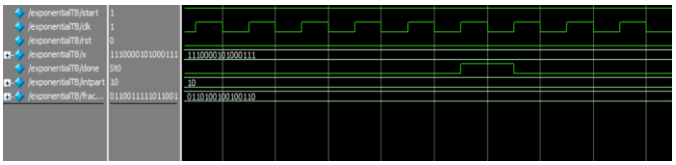

Fig. 4 simulation result

When x is 0. 1110000101000111 in binary or 0.88 in decimal , the output is 10.0110100100100110which means 2.4107360 but $e^{0.88}$ is 2.41089970 and the difference is 0.0001637.

2.

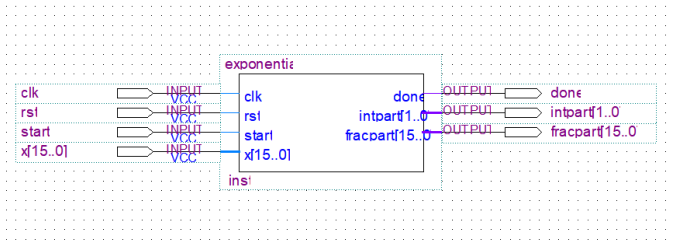

Fig. 5 exponential in Quartus

Flow Status	Successful - Thu Jun 02 23:10:13 2022
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	exponential
Top-level Entity Name	exponential
Family	Cyclone IV E
Device	EP4CE6E22A7
Timing Models	Final
Total logic elements	103 / 6,272 (2 %)
Total registers	61
Total pins	38 / 92 (41 %)
Total virtual pins	0
Total memory bits	0 / 276,480 (0 %)
Embedded Multiplier 9-bit elements	2 / 30 (7 %)
Total PLLs	0 / 2 (0 %)

Fig. 6 synthesis result

3.

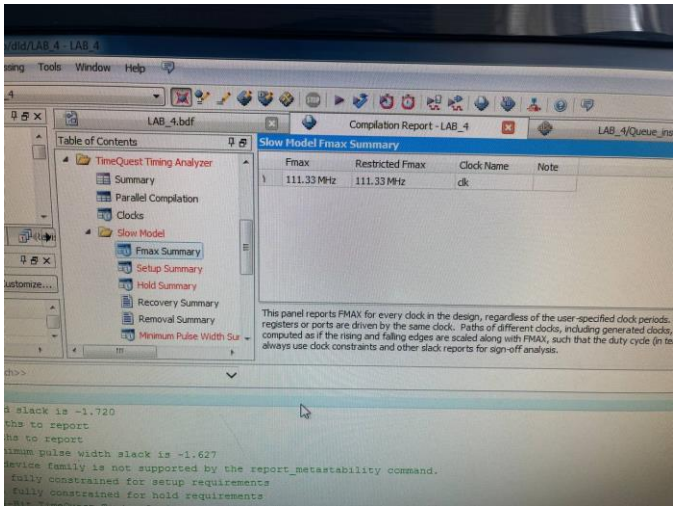

Fig. 7 maximum frequency of accelerator

2.2 The Wrapper controller

1.

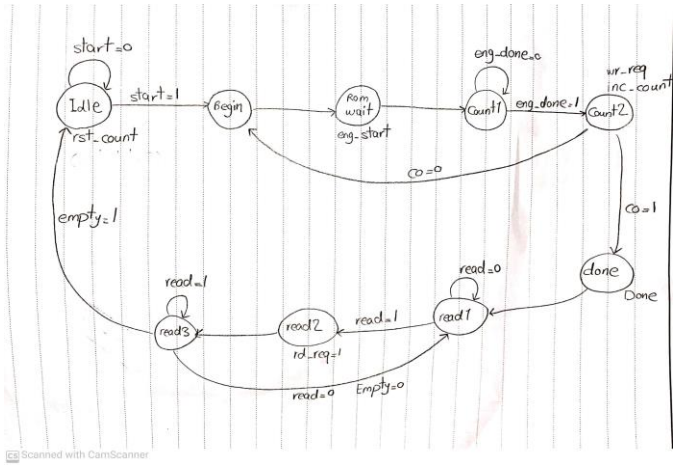

Fig. 8 state diagram of the controller

```
module MyController(input Empty,start,clk,rst,eng_done,read,output reg eng_start,done,rd_req,wr_req, output[2:0] count_out);
    reg[3:0]ps,ns;
    reg rst_count,inc_count;
    wire co;

    parameter [3:0] Idle = 4'd0, Beginn = 4'd1, Rom_wait= 4'd2, Count1 = 4'd3, Count2 = 4'd4,
            Done = 4'd5, Read1 = 4'd6, Read2 = 4'd7,Read3= 4'd8;

    Counter8bit counterr(inc_count,rst_count,clk, rst,count_out,co);

    always@(ps,co,start,eng_done,Empty,read)begin
            {rst_count,eng_start,inc_count,wr_req,done,rd_req}=6'b0;
        case(ps)
            Idle:begin ns = start?Beginn:Idle;rst_count = 1'd1;end
            Beginn:begin ns = Rom_wait;end
            Rom_wait:begin ns = Count1;eng_start = 1'd1;end
            Count1:begin ns = eng_done?Count2:Count1;end
            Count2:begin ns = co?Done:Beginn;wr_req = 1'd1;inc_count=1'd1;end
            Done: begin ns = Read1;done = 1'd1;end
            Read1: begin ns = read?Read2:Read1;end
            Read2: begin ns = Read3; rd_req=1'd1;end
            Read3: begin ns = Empty?Idle:read? Read3:Read1;end
            default ns = Idle;
        endcase
    end

    always@(posedge clk, posedge rst)begin
        if (rst) ps <= Idle;
        else ps <= ns;
    end
endmodule
```

Fig. 9 Controller verilog

2.

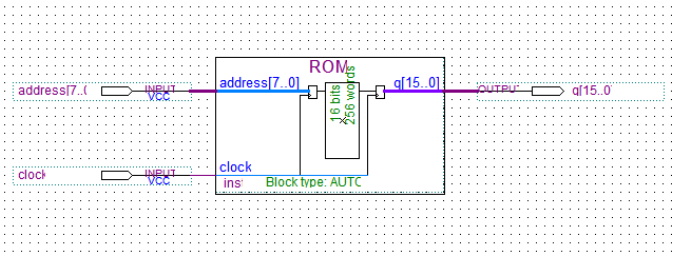

Fig. 10 ROM IP

```
WIDTH=16;
DEPTH=5;

ADDRESS_RADIX=UNS;
DATA_RADIX=UNS;

CONTENT BEGIN
        0  :  10000;
        1  :  5432;
        2  :  756;
        3  :  25000;
        4  :  6666;
END;
```

Fig. 11 mif file

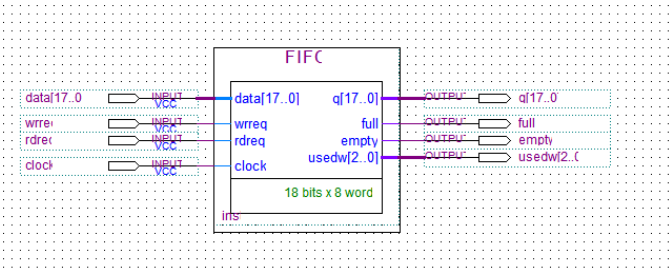

Fig. 12 FIFO IP

3.

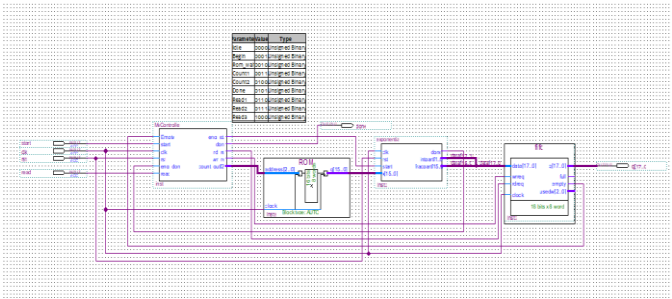

Fig. 13 Wrapper in Quartus

4.

```
`include"LAB_4.vo"
`timescale 1ns/1ns

module expTB();
    reg start = 0, read = 0, clk = 0, rst = 0;

    wire done;
    wire [17:0] q;

    LAB_4 CUT(done,clk,rst,start,read,q);

    always #10 clk = ~clk;

    initial begin
        #6 rst = 1;
        #6 rst = 0;
        #20 start = 1;
        #20 start = 0;
        while (~done) #10;
        #70 read = 1;
        #70 read = 0;
        #70 read = 1;
        #70 read = 0;
        #70 read = 1;
        #70 read = 0;
        #70 read = 1;
        #70 read = 0;
        #70 read = 1;
        #70 read = 0;
        #1000 $stop;
    end
endmodule
```

Fig. 14 Testbench verilog

5.

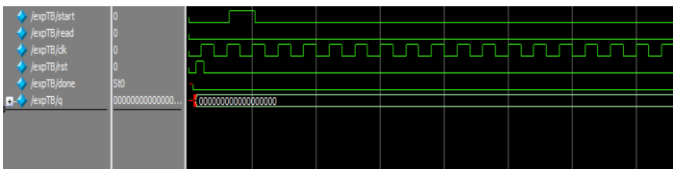

Fig. 15 a complete pulse on signal “start”

6.

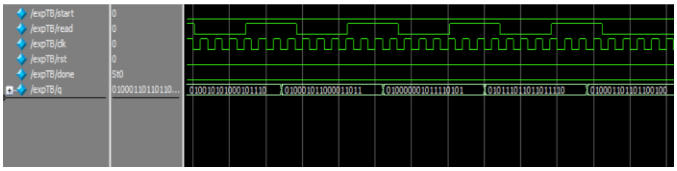

Fig. 16 simulation result

Input 1:

10000 in decimal → 0010011100010000 in binary

0. 0010011100010000 in binary → 0.15258789 in decimal

$e^{0.15258789}$ = 1.1648448

our result in waveform → 01.0010101000101110 in binary and 1.1647644 in decimal. The difference is 0.0000804.

Input 2:

5432 in decimal → 0001010100111000 in binary

0. 0001010100111000 in binary → 0.0828857 in decimal

$e^{0.0828857}$ = 1.0864176

our result in waveform → 01.0001011000011011 in binary and 1.0863494 in decimal. The difference is 0.0000682.

Input 3:

756 in decimal → 0000001011110100 in binary

0.0000001011110100 in binary → 0.0115356 in decimal

$e^{0.0115356}$ = 1.0116023

our result in waveform → 01.0000001011110101 in binary and 1.0115509 in decimal. The difference is 0.0000153.

Input 4:

25000 in decimal → 0110000110101000 in binary

0.0110000110101000 in binary → 0.3814697 in decimal

$e^{0.3814697}$ = 1.4651603

our result in waveform → 01.0111011011011110 in binary and 1.4643249 in decimal. The difference is 0.0008354.

Input 5:

6666 in decimal → 0001101000001010 in binary

0. 0001101000001010 in binary → 0.1017150 in decimal

$e^{0.1017150}$ = 1.1070679

our result in waveform → 01.0001101101100100 in binary and 1.1069946 in decimal. The difference is 0.0000733.

3.Implementing Accelerator on FPGA

1.

Flow Status	Successful - Fri Jun 03 02:13:55 2022
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	LAB_4
Top-level Entity Name	LAB_4
Family	Cyclone IV GX
Device	EP4CGX15BF14A7
Timing Models	Final
Total logic elements	478 / 14,400 (3 %)
Total registers	84
Total pins	23 / 81 (28 %)
Total virtual pins	0
Total memory bits	272 / 552,960 (< 1 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)

Fig. 17 synthesis report

3.

```
`include"Hexdisplay.v"
`include"LAB_4.vo"
module CA4(input clk,rst,start,read,output[6:0]displ1,displ2,displ3,displ4,output done);
    wire[17:0]q;
    Hexdisplay HEX1(q[17:14],displ1);
    Hexdisplay HEX2(q[13:10],displ2);
    Hexdisplay HEX3(q[9:6],displ3);
    Hexdisplay HEX4(q[5:2],displ4);
    LAB_4 CUT(done,clk,rst,start,read,q);
endmodule
```

Fig. 18 converter for 7-segment display

4.

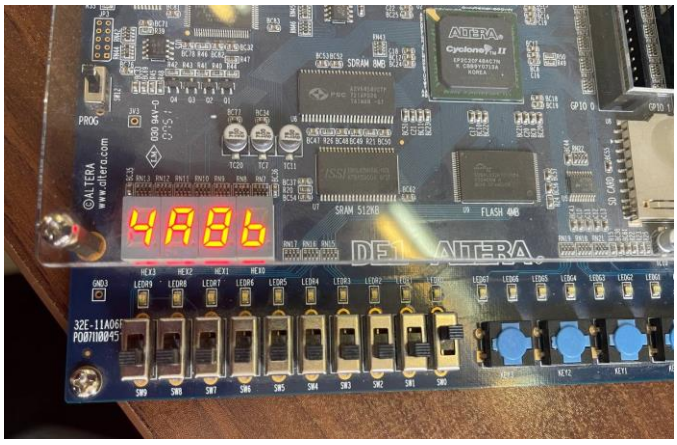

Fig. 19 Exp-out

Input 1:

10000 in decimal → 0010011100010000 in binary

0. 0010011100010000 in binary → 0.15258789 in decimal

$e^{0.15258789}$ = 1.1648448

HEX : 4a8b → Binary : 0100 1010 1000 1011

Decimal with two bit integer : 1.16473388

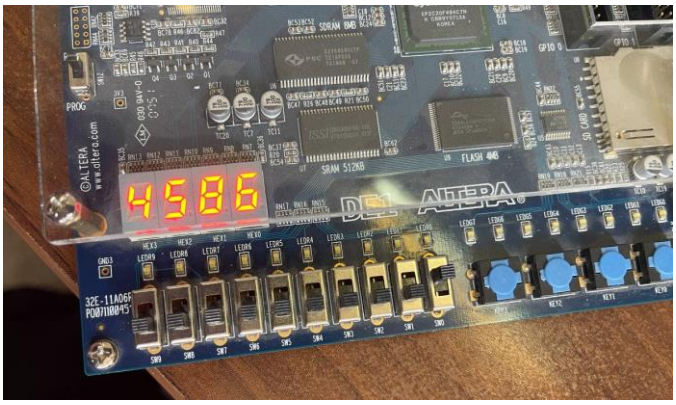

Fig. 20 Exp-out

Input 2:

5432 in decimal → 0001010100111000 in binary

0. 0001010100111000 in binary → 0.0828857 in decimal

$e^{0.0828857}$ = 1.0864176

HEX : 4586 → Binary : 0100 0101 1000 0110

Decimal with two bit integer : 1.08630371

Fig. 21 Exp-out

Input 3:

756 in decimal → 0000001011110100 in binary

0.0000001011110100 in binary → 0.0115356 in decimal

$e^{0.0115356}$ = 1.0116023

HEX : 406d → Binary : 0100 0000 0110 1101

Decimal with two bit integer : 1.01153564

Fig. 22 Exp-out

Input 4:

25000 in decimal → 0110000110101000 in binary

0.0110000110101000 in binary → 0.3814697 in decimal

$e^{0.3814697}$ = 1.4651603

HEX : 5db7 → Binary : 0101 1101 1011 0111

Decimal with two bit integer : 1.46429443

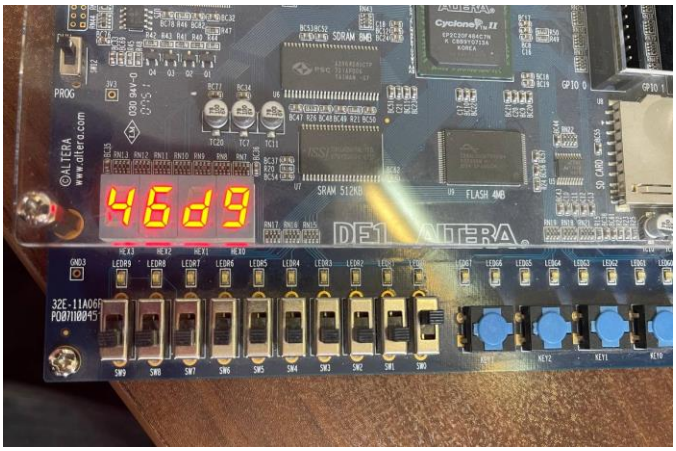

Fig. 23 Exp-out

Input 5:

6666 in decimal → 0001101000001010 in binary

0. 0001101000001010 in binary → 0.1017150 in decimal

$e^{0.1017150}$ = 1.1070679

HEX : 46d9 → Binary : 0100 0110 1101 1001

Decimal with two bit integer : 1.10699462