EE518 - VLSI System Design
Course Project

An Area-Efficient and High Throughput Hardware Implementation of Exponent Function

Submitted by,
Monish Nath
Roll No-224102409
May 2, 2023

Contents

List of Figures

List of Tables

1 Objective :

Design an Area-Efficient and High Throughput Hardware Implementation of Exponent Function.

2 Theory :

The exponential function is an important mathematical function used in digital signal processing applications, such as sine, cosine, logarithm, as well as modern computation algorithms, such as Deep Neural Networks (DNNs) , Long Short-Term Memory (LSTM) , Graph Neural Networks , etc. The exponential function calculation can be executed in general purpose Central Processing Units (CPUs), however, for high-speed calculation of the exponential function, the high-speed special solution is desired. Field Programmable Gate Arrays (FPGA) and Application Specific Integrated Circuit (ASIC) are widely used to perform the acceleration of the mathematical functions which include exponential function also.

3 Design Approach :

The proposed method aims to eliminate the need for memory requirements during exponent calculation, so, we do not utilize Taylor series to calculate the exponent value because Taylor series has a high memory requirement. The input x is in the format 1-bit sign, 15-bit integer and 16-bit fraction.

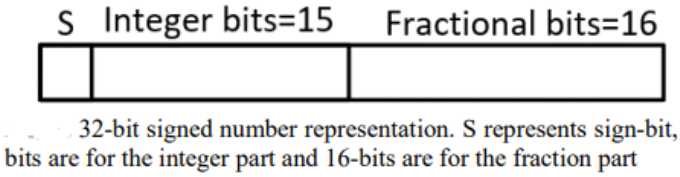

Figure 1: Format of input x - 32b

The exponent function can be written as

$$e^x = 2^{xlog_2(e)} \tag{1}$$

$$2^{xlog_2(e)} = 2^i.2^f \tag{2}$$

where y = $log_2(e)$ and z = x.y = i + f
here i and f are integer and fractional parts respectively. 2^i can be calculated easily through shift operation and for calculation of 2^f we have a separate expression which has been derived from mini-max approach.

$$2^f = a + b.f + c.f^2 + d.f^3 \tag{3}$$

where a=0.99992807, b=0.69326098, c=0.24261112, and d=0.00517166. The value of f should lie between [-0.5,0.5].

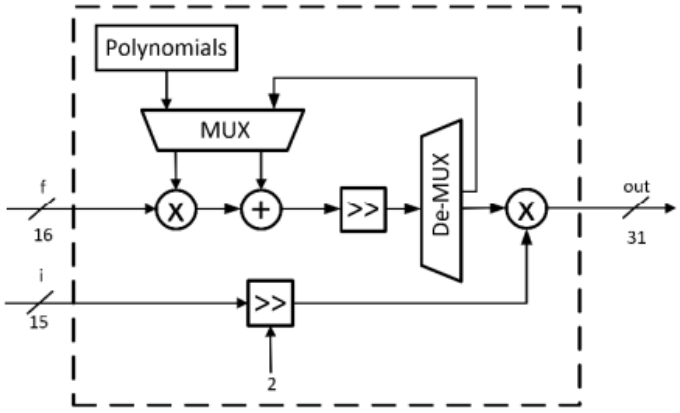

Figure 2: The proposed hardware design

4 Verilog Code :

```
module expfn_pipelined(
    input [31:0] x, //1.15.16 format
    input clock,
    input reset,
    output [46:0] out   //22.25
    );
    parameter signed a = 20'b00001111111111111011,
    b = 20'b00001011000101111001,
    c = 20'b00000111111000011011,
    d = 20'b00000000000101010010;
```

```
    wire [47:0] z;  //16.32
    wire [15:0] i, i1;
    wire signed [15:0] f;
    wire [19:0] two_f, sum; //4.16
    reg [19:0] d_ff;
    reg [1:0] count=0;
    wire signed [19:0] mult, prod_fb, prod_shifted, add;
    wire signed [35:0] prod;
    wire [29:0] two_i;  //21.9

    assign z = x[30:0] * 17'b10111000101010100;
                   //log2(e)=1.01110001010101000111
    assign i = z[46:32] + z[31] ;
    assign f = x[31] ? ~z[31:16] : z[31:16];

    always@(posedge clock or posedge reset)
                           if(reset==1) count <= 'b0;
                           else count <= count + 'b1;
    always@(posedge clock or posedge reset)
                           if(reset==1) d_ff <= 'b0;
                           else d_ff <= sum;

    assign mult = (count == 'd0) ? d : prod_fb;
    assign prod = mult * f;
    assign prod_shifted = prod>>>16;
    assign add = (count[1]) ? (count[0]) ? 'd0 : a
                         : (count[0]) ? b : c;
    assign sum = add + prod_shifted;
    assign prod_fb = (count == 'd3) ? 'b0 : d_ff;
    assign two_f = (count == 'd3) ? d_ff : 'b0;


    assign i1 = x[31] ? 'd9-i : 'd9+i;
    assign two_i = 'b1 << (i1); //-ve x working

    assign out = two_i * two_f[16:0];   //21.9 * 1.16

endmodule
```

5 Test bench ;

```
module expfn_tb(    );

    reg [31:0] x;
    reg clock, reset;
    wire [46:0] out;

    always #50 clock = ~clock;

    expfn_pipelined DUT(x, clock, reset, out);

    initial begin
    clock = 'b1; reset = 'b1; #110 reset = 'b0;
        x = 32'b0_00000000000010_0000000000000000;
    #390 x = 32'b0_00000000000011_0000000000000000;
    #400 x = 32'b0_00000000000000_1000000000000000;
    #400 x = 32'b1_00000000000010_0000000000000000; //-2
    #400 x = 32'b1_00000000000011_0000000000000000; //-3
    end

endmodule
```

6 RTL Schematic :

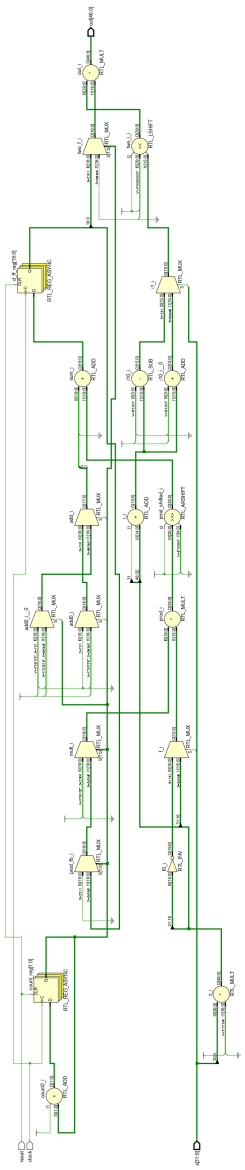

Figure 3: Schematic of the proposed architecture

7 Simulation :

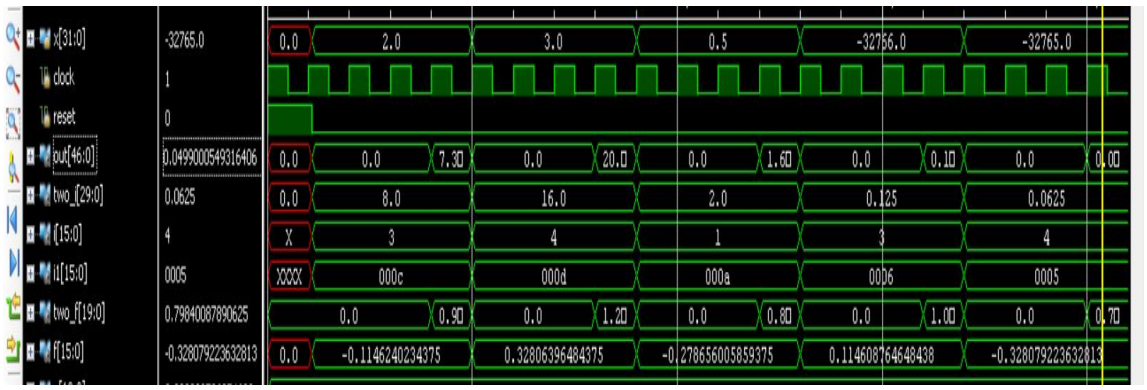

Figure 4: Behavioural simulation waveform

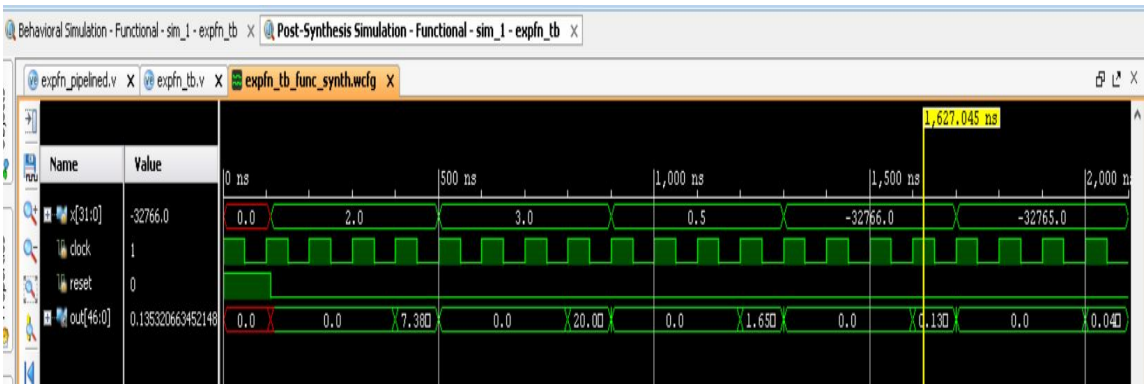

Figure 5: Post-synthesis functional simulation waveform

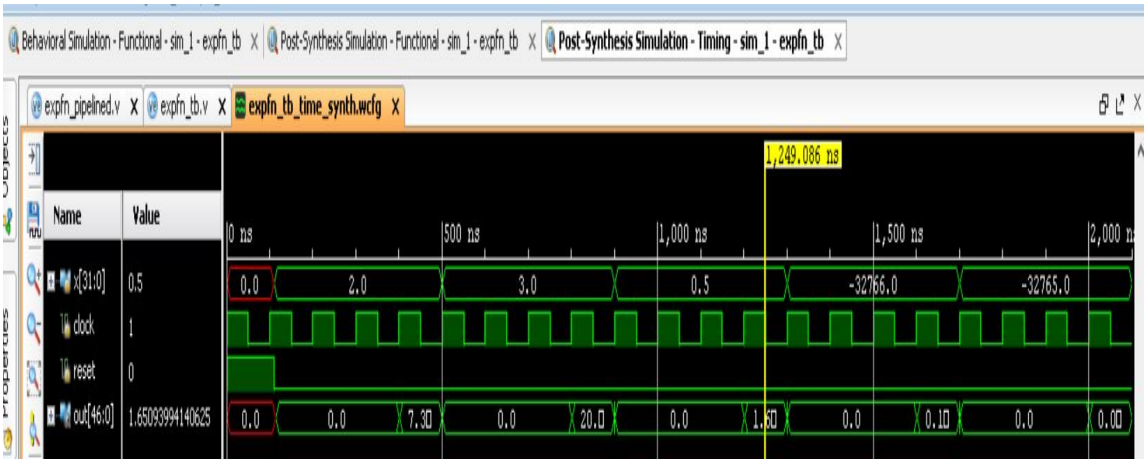

Figure 6: Post-synthesis timing simulation waveform

Note: The representation of the input and output are kept in signed integer format with proper decimal point.

The inputs that are fed are x = 2, 3, 0.5, -2, -3.

The outputs got are 7.3890380859375, 20.05810546875, 1.65093994140625, 0.135320663452148, 0.0499000549316406.

The correct outputs of e^x from actual calculator are 7.389056099, 20.08553692, 1.648721271, 0.1353352832, 0.04978706837.

This shows the relative error is less and accuracy is good.

8 Synthesis Reports :

The values have been found after synthesis of the corresponding designs. I have done the experiment on Vivado 2014.1. The FPGA board selected is Artix-7. The LUTs and Flops have been found from the utilization report. The delay has been found from the timing report and the power has been found from the power report.

We have added proper constraints and the synthesis results are shown below.

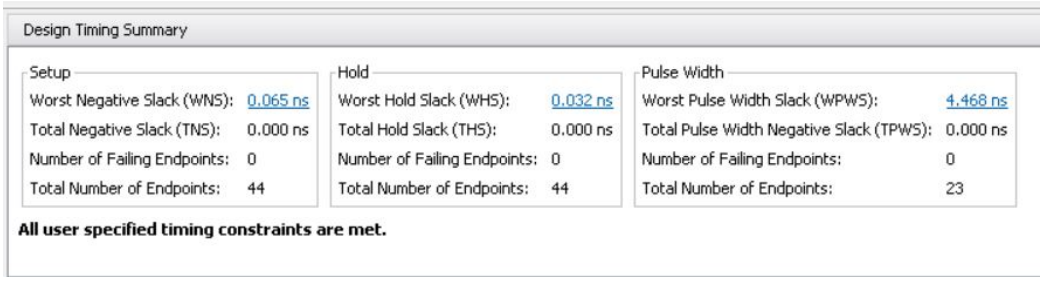

Figure 7: Timing summary

Site Type	Used	Locked	Available	Util%
Slice LUTs*	356	0	134600	0.26
LUT as Logic	356	0	134600	0.26
LUT as Memory	0	0	46200	0.00
Slice Registers	22	0	269200	<0.01
Register as Flip Flop	22	0	269200	<0.01
Register as Latch	0	0	269200	0.00
F7 Muxes	0	0	67300	0.00
F8 Muxes	0	0	33650	0.00

Figure 8: Utilization report

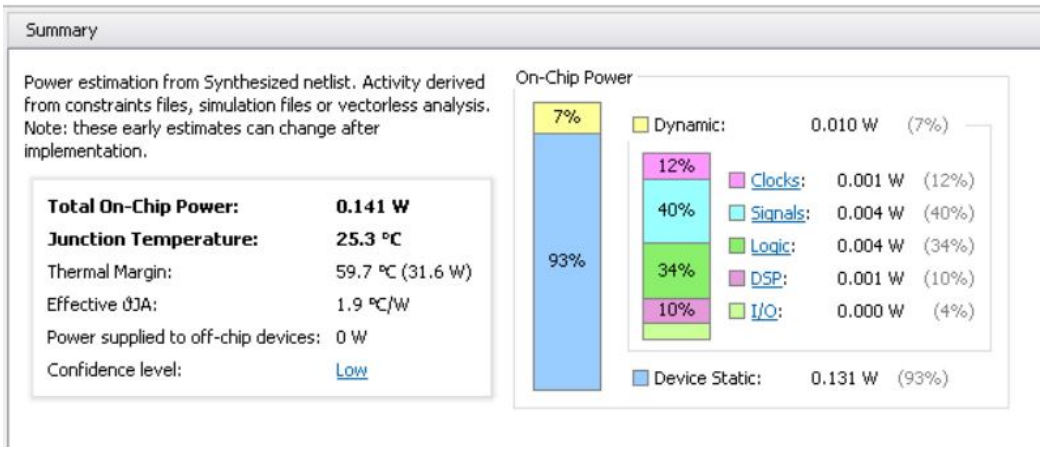

Figure 9: Power report

9 Conclusions :

- We have designed the circuit as per the requirement.
- The functionality of our design have been verified . The functionality are showing as expected.
- The different parameters of the design such as LUTs, delay and power have been calculated from the synthesis and tabulated.

10 Reference Paper:

M. A. Hussain, S. -W. Lin and T. -H. Tsai, "An Area-Efficient and High Throughput Hardware Implementation of Exponent Function," 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 2022, pp. 3369-3372, doi: 10.1109/ISCAS48785.2022.9937238.