ECE 4515: Digital Design 2

Project 3: Optimizing for Area

Manav Bhavesh Shah
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, USA
manavbs28@vt.edu

"*I have neither given nor received unauthorized assistance on this assignment.*"
March 18, 2025

~Manav Shah

1. Introduction

This project aims at area optimizations for the Maclaurin Series Solver.
The approach is that we reuse DSP elements for calculations without instantiating multiple DSPs.

2. Design Approach

$$e^x - 1 = \sum_{n=1}^{\infty} \frac{x^n}{n!} = x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} \ldots$$

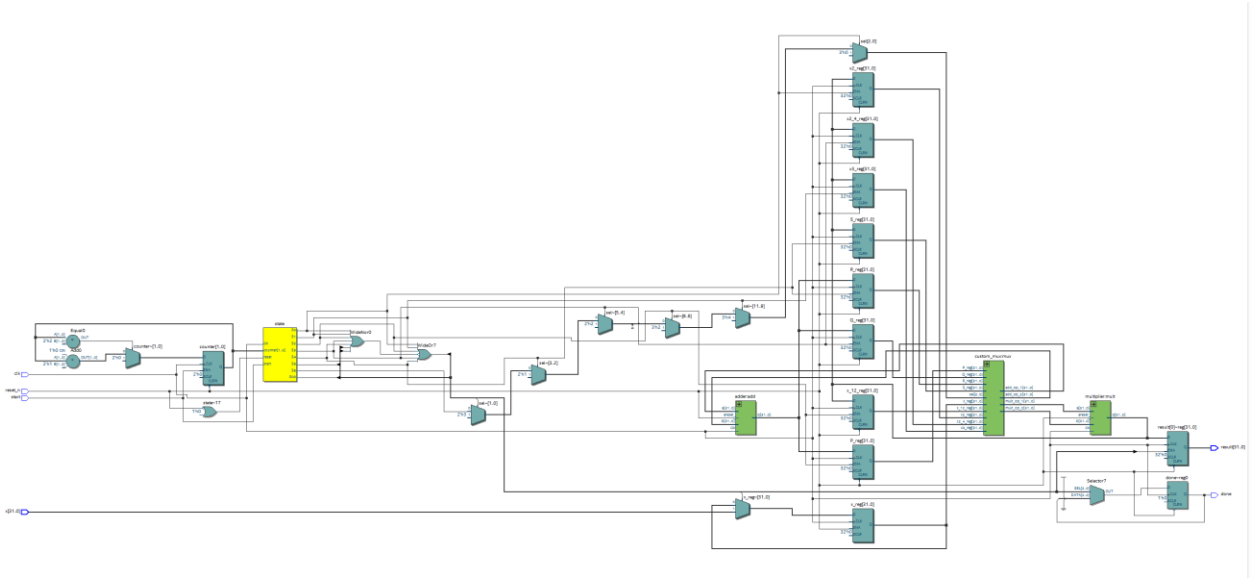
The optimized version of the Maclaurin series expansion is as follows:
$x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} = \frac{x}{24}(24 + 12x + 4x^2 + x^3)$ $=(0.04166668x)(24 + 12x + 4x^2 + x^3)$

For this project, reducing area was the top priority, so I decided to use one multiplier and one adder Floating Point IP. An FSM was made to control the inputs to these Ips and the outputs from them were stored in registers. I made intermediate registers which hold values of calculations. The Mux design decides which registers send their values to the multiplier and adder based upon select input from the state machine.

The table below shows the parallel solving of the Maclaurin series used for this project. The latency of the adder and the multiplier is 2 cycles. This means that every intermediate result is obtained on every third cycle. There is an additional state not shown in the table below which is State00 which stores the input value of x to register named x_reg. This is to prevent the input value from changing and causing miscalculations when the solver is busy.

This solver design produces an answer after 20 cycles of providing the input. (i.e. latency is 20 cycles).

	Multiplier input	Multiplier input	Mult output	Adder Input 1	Adder input 2	Add output
State 0	x	x	x^2			
State 1	x^2	x	x^3			
State 2	12	x	12x	24	x^3	P
State 3	4	x^2	4x^2	P	12x	Q
State 4	x	0.0416	S	Q	4x^2	R
State 5	S	R	Result			

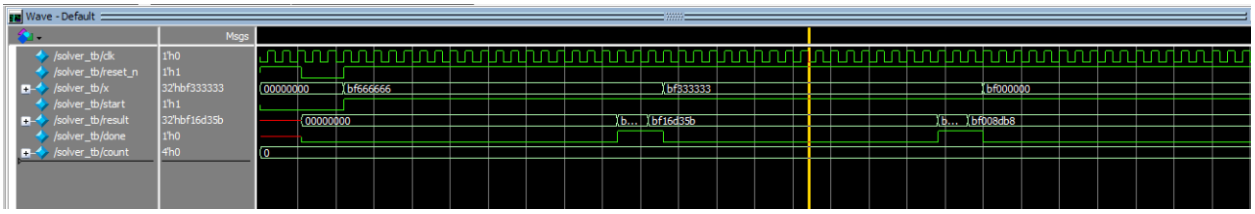


RTL view of the Solver

3. Latency

The latency for the multiplier and adder is 2 cycles. This means that a result is available at the third cycle for every multiplication and addition operation. Given the number of calculations the design makes to generate a result, we expect an output at every 20th cycle.

The figure below shows that the result is ready after 20 cycles of the input being provided.



Waveform showing 20 cycle latency of the solver.

The maximum operating frequency for this design is found to be 91.77Mhz.
Therefore, clock period = 1/91.77Mhz = **10.9ns**
Instruction Cycle Time = latency * clock period = 20 * 10.9ns = **218ns.**

Since the design is serial non pipelined, this instruction cycle time is indicative of the throughput.

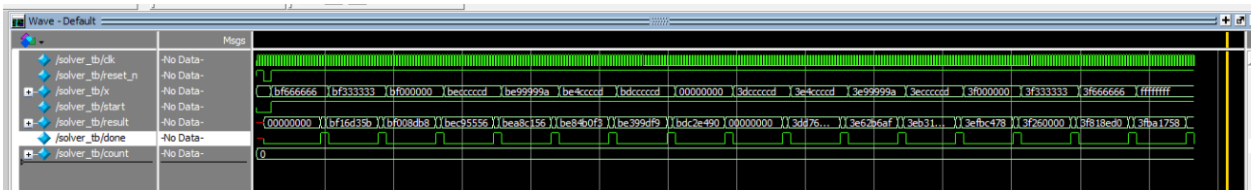
4. Results

For designing the testbench of this project, I used the following approach:

```
// Apply reset
#55 reset_n = 0;
#55 begin reset_n = 1;
end
x = 32'hbf666666;
start = 1;


wait (done);
@(negedge done);
x = 32'hbf333333;


wait(done);
 @(negedge done);
x = 32'hbf000000;

wait(done);
 @(negedge done);
x = 32'hbeccccd;

wait(done);
 @(negedge done);
x = 32'hbe99999a;
```

After the reset has been applied, I apply the start signal along with the first test value of x input. I wait till the done signal goes high and then at the negedge of the done signal I apply my next input x test vector. The same thing is repeated for all the input testcases. Testing shows that the results generated are correct and are obtained at the right latency for every testcase. The following table shows the expected results for x inputs and the waveform which follows shows the testbench output which validates the correctness of the design.

Maclaurin Input (x)	Expected Values	Hex values
-0.9	-0.5891625	bf16d35b
-0.7	-0.5021625	bf008de6
-0.5	-0.393229167	bec95555
-0.3	-0.2591625	be84b0f2
-0.1	-0.0951625	bdc2e48f
0	0	0
0.1	0.105170833	3dd763ce
0.3	0.3498375	3eb31de7
0.5	0.6484375	3f260000
0.7	1.012170833	3f818ed0
0.9	1.4538375	3fba1759



Flow Summary

<<Filter>>

Flow Status	Successful - Tue Mar 18 20:04:39 2025
Quartus Prime Version	23.1std.1 Build 993 05...4/2024 SC Lite Edition
Revision Name	top
Top-level Entity Name	solver
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	517 / 32,070 (2 %)
Total registers	494
Total pins	68 / 457 (15 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	1 / 87 (1 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Slow 1100mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	91.77 MHz	91.77 MHz	clk	

Area Calculation:

*Area = (75 * DSP) + (ALM) + (0.001 * BLOCK_MEMORY_BITS)*
*= (75 * 1) + (517) + (0.001*0) = 592*

5. Conclusion

This project showed how I can reuse IP blocks with control FSM to reduce area consumption / resource utilization to achieve the same results at a reduced maximum frequency of operation. An improvement possible to the current design would be to further optimize the equation itself by taking x values common which would further reduce the area.