Dual Fixed Point Calculator

ECE 5900 ST: RECONFIGURABLE COMPUTING, FALL 2017

PROFESSOR: DANIEL LLAMOCCA

PROJECT
by
Furzana AbdulRahim; Jing Wu; Zhongda Gan

Introduction

- Dual Fixed Point(DFX) is new compared to Fixed Point(FX) and Floating Point(FP).
- Uses less resources compared to FP.
- Higher precision compared to FX.
- Applied the knowledge acquired in class in designing the DFX calculator

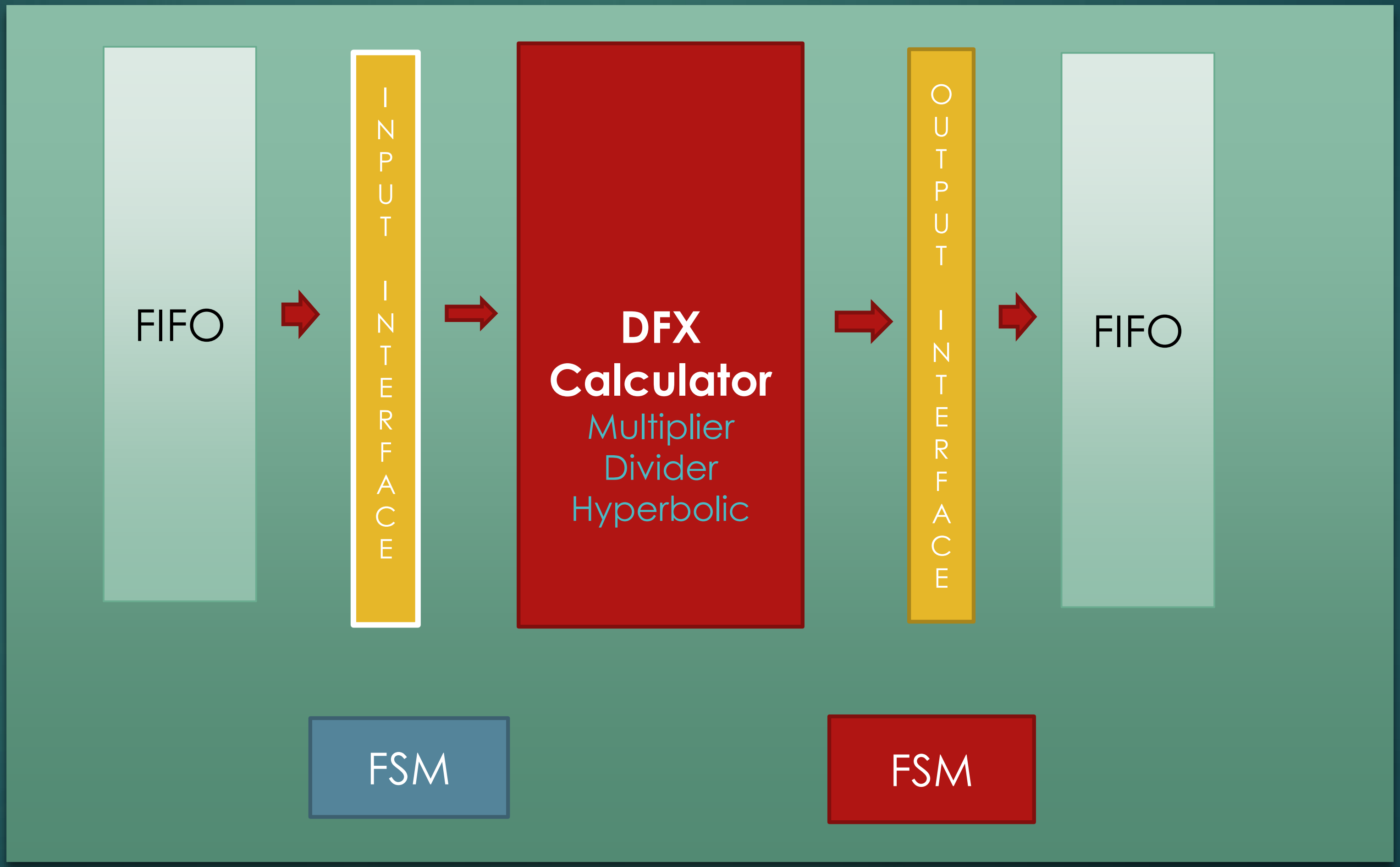
Operations in DFX Calculator

- Multiplication
- Division
- Expanded Hyperbolic

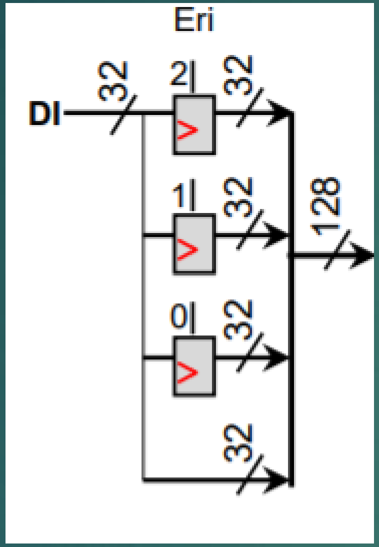
Features

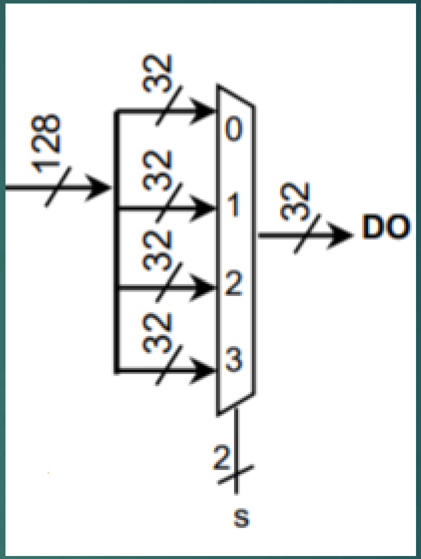
- DMA transaction on AXI full– multiple data writes and reads.
- Pipeline architecture – There is an initial latency and then results are seen continuously.
- Signed 32 bit operand support.

Block Diagram



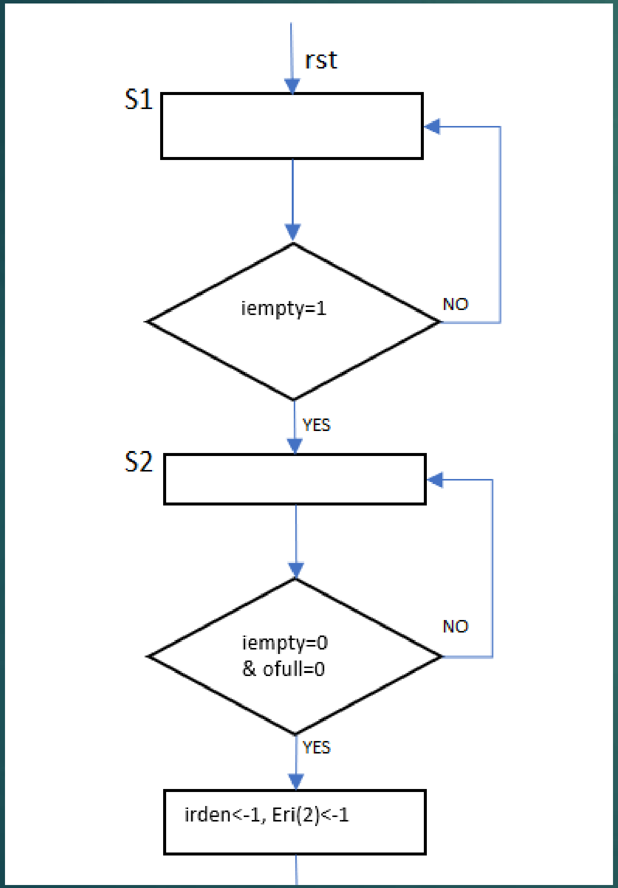
Interface Block

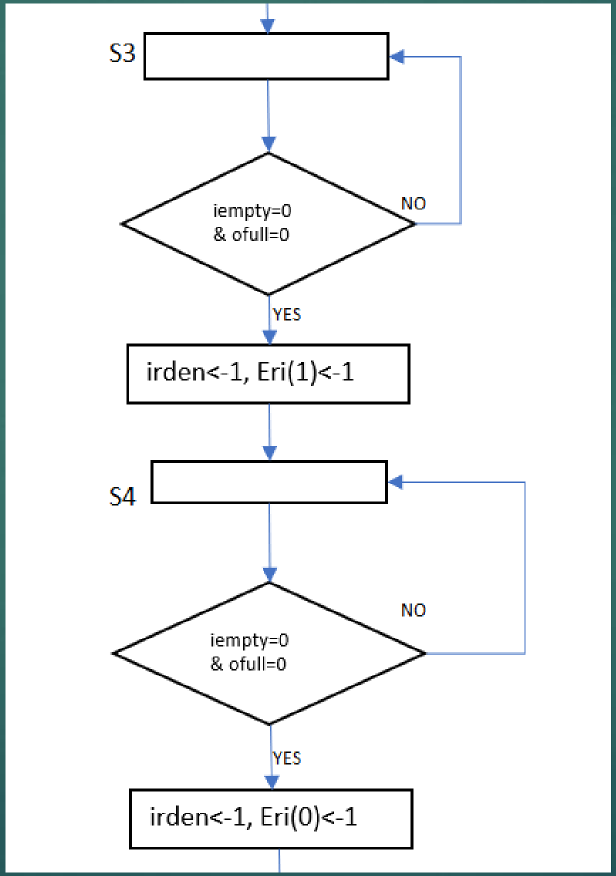


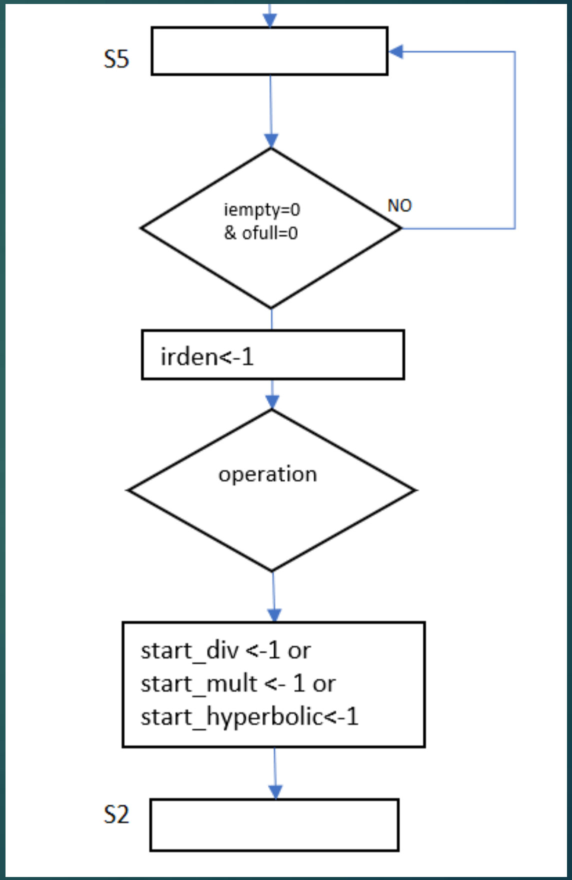


- Due to different number of inputs in operations, we have maintained 4 inputs and outputs for simplicity.

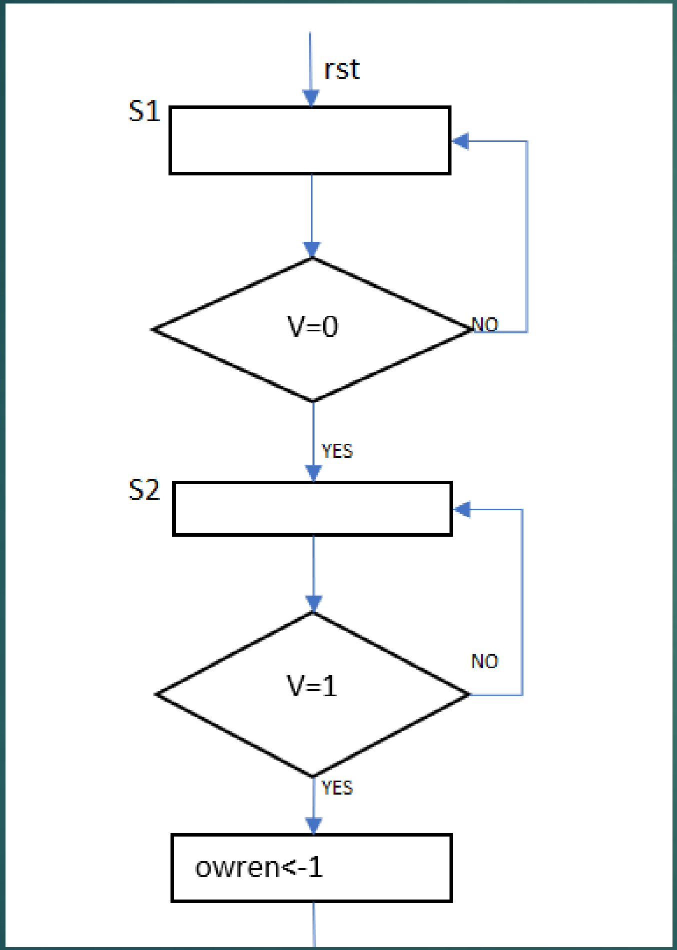
Input State Machine

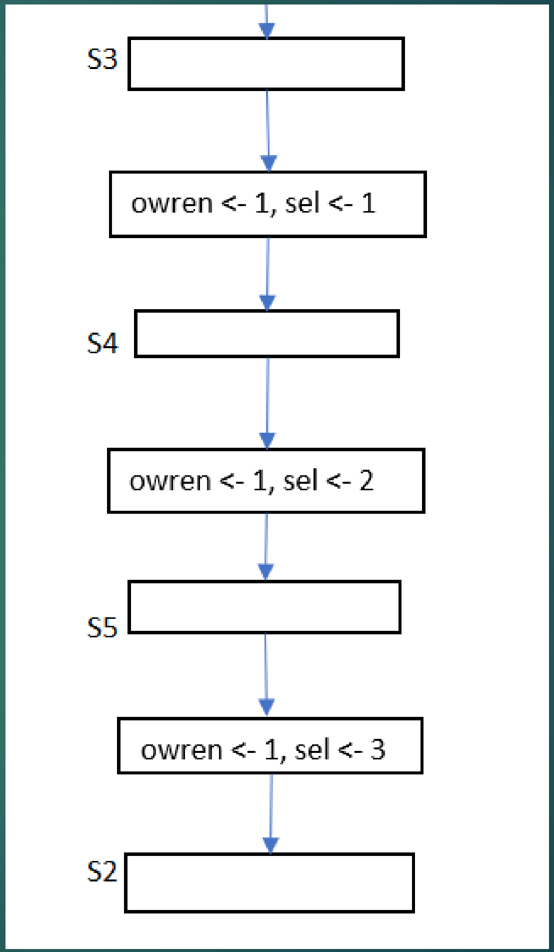




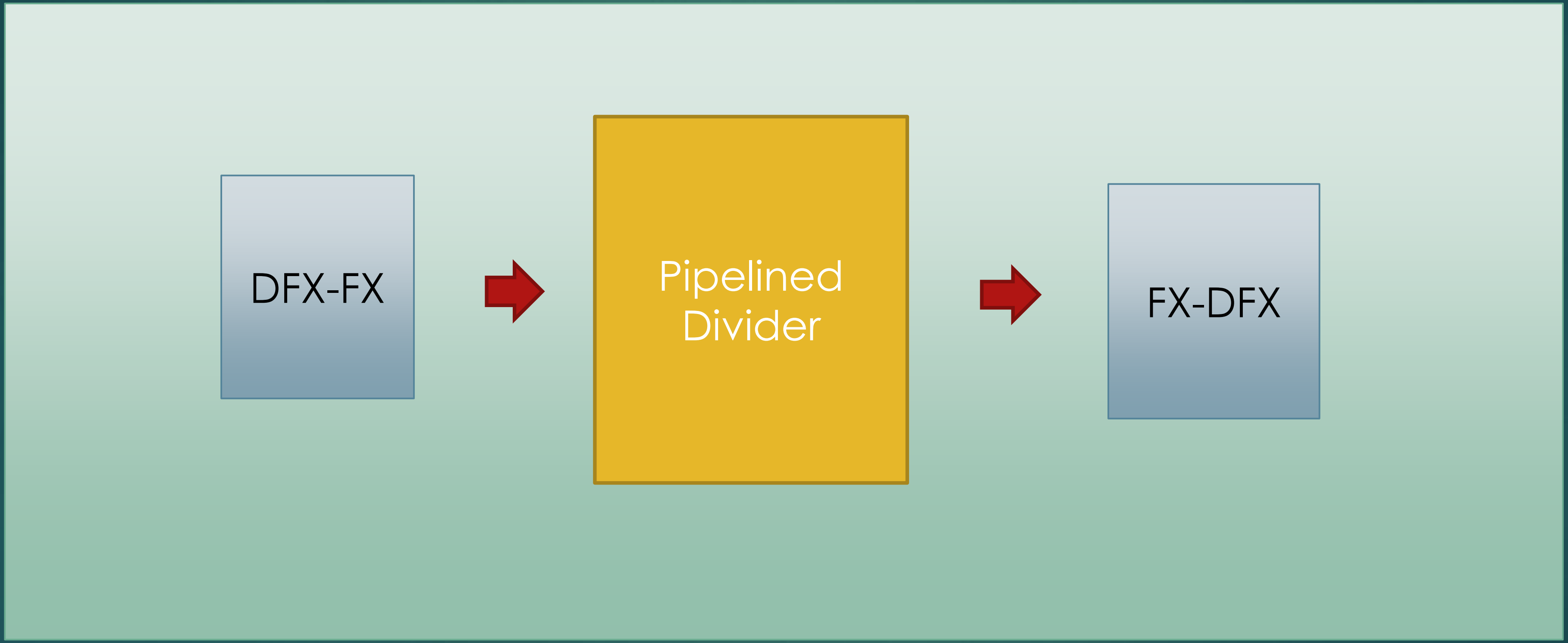


Output State Machine

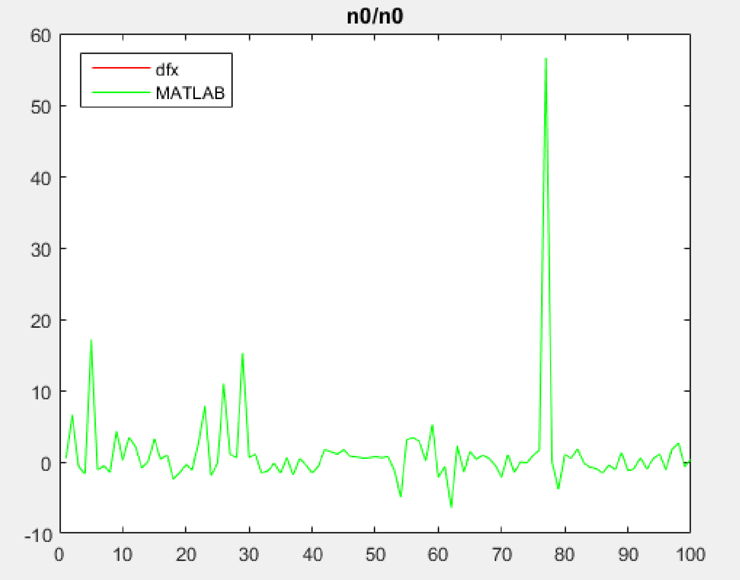


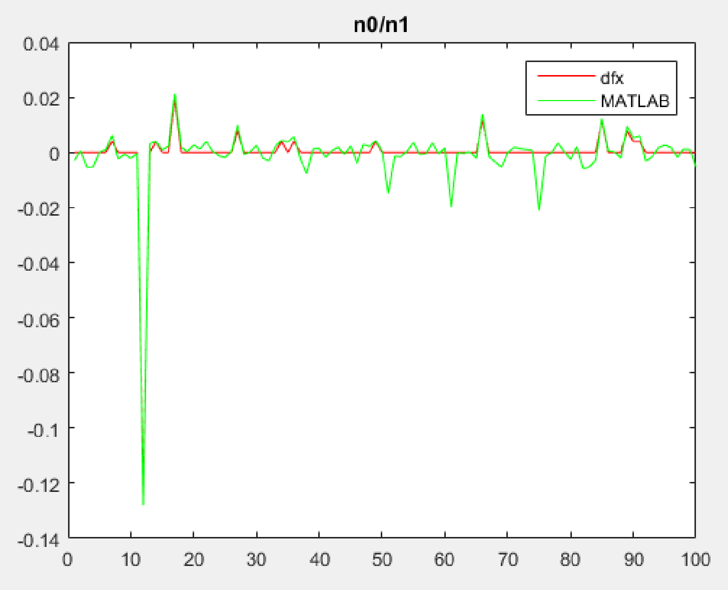


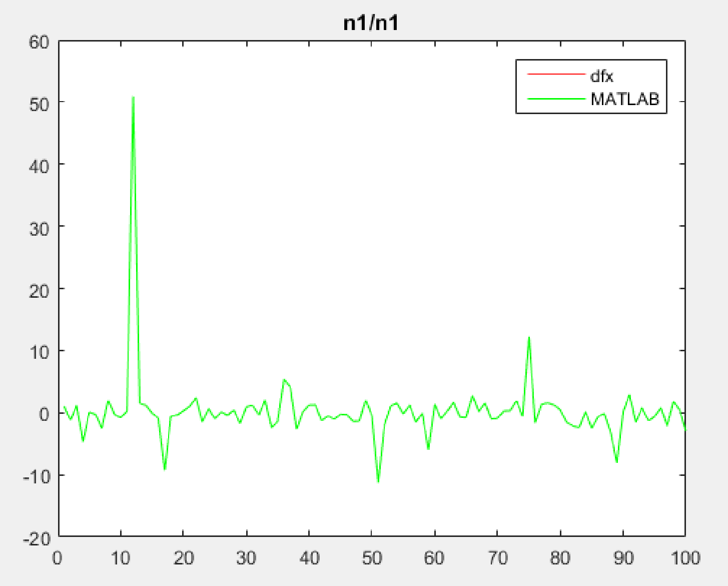
Pipelined Divider

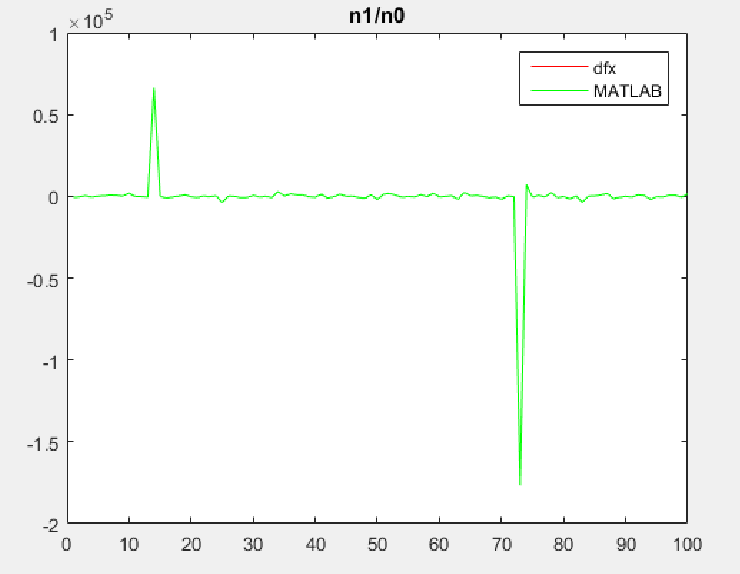


Pipelined Divider









Pipelined Expanded Hyperbolic

For $i \leq 0, i = -M, -m+1, \ldots, 0$. Here we choose M=4

$$\begin{cases} x_{i+1} = x_i - \delta y_i (1 - 2^{i-2}) \\ y_{i+1} = y_i - \delta x_i (1 - 2^{i-2}) \\ z_{i+1} = z_i + \delta tanh^{-1} (1 - 2^{i-2}) \end{cases}$$

For $i > 0, i = 0,1,2, \ldots, N$

$$\begin{cases} x_{i+1} = x_i - \delta y_i 2^{-i} \\ y_{i+1} = y_i - \delta x_i 2^{-i} \\ z_{i+1} = z_i + \delta tanh^{-1} 2^{-i} \end{cases}$$

for i = 4,13, we need to repeat it.

Pipelined Expanded Hyperbolic

In rotation mode:

$$\delta = \begin{cases} +1, if\ z_i \leq 0 \\ -1, if\ z_i > 0 \end{cases}$$

$$\begin{cases} x_{n+1} = A_n(x_{in}\cosh(z_{in}) + y_{in}\sinh(z_{in})) \\ y_{n+1} = A_n(x_{in}\sinh(z_{in}) + y_{in}\cosh(z_{in})) \\ z_{n+1} = 0 \end{cases}$$

Then convergence range is: $|z_{in}| \leq 9.66581$

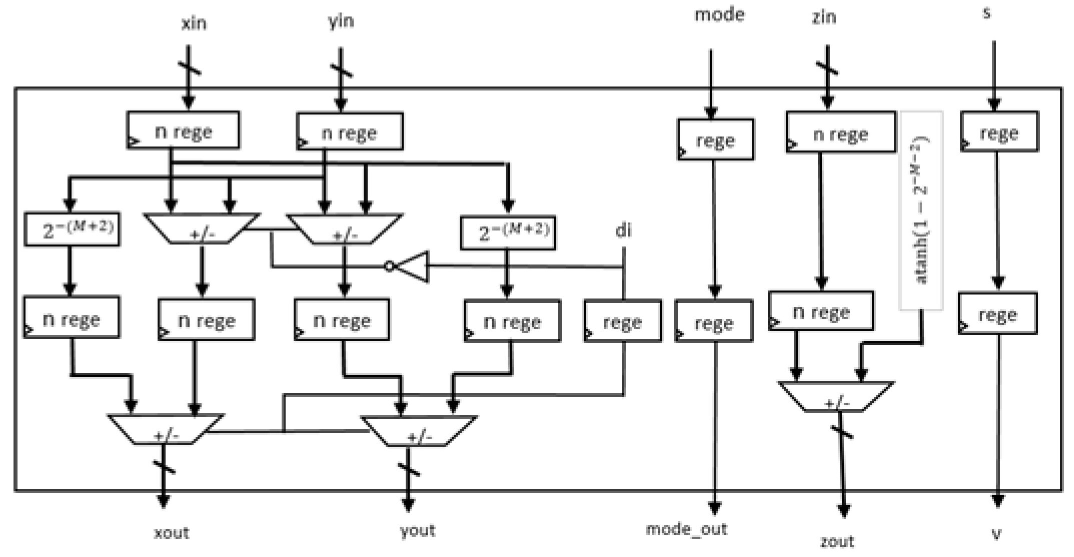
In vector mode:

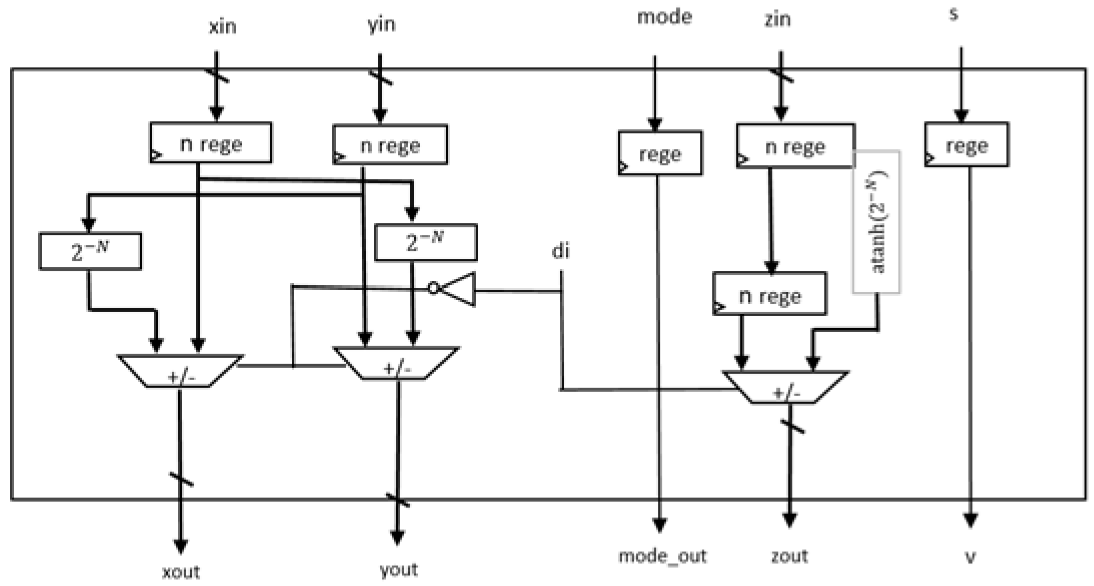
$$\delta = \begin{cases} +1, if\ x_i y_i \geq 0 \\ -1, if\ x_i y_i < 0 \end{cases}$$

$$\begin{cases} x_{n+1} = A_n\sqrt{x_{in}^2 - y_{in}^2} \\ y_{n+1} = 0 \\ z_{n+1} = Z_{in} + tanh^{-1}(y_{in}/x_{in}) \end{cases}$$

Then convergence range is: $|tanh^{-1}(y_{in}/x_{in})| \leq 9.66581$

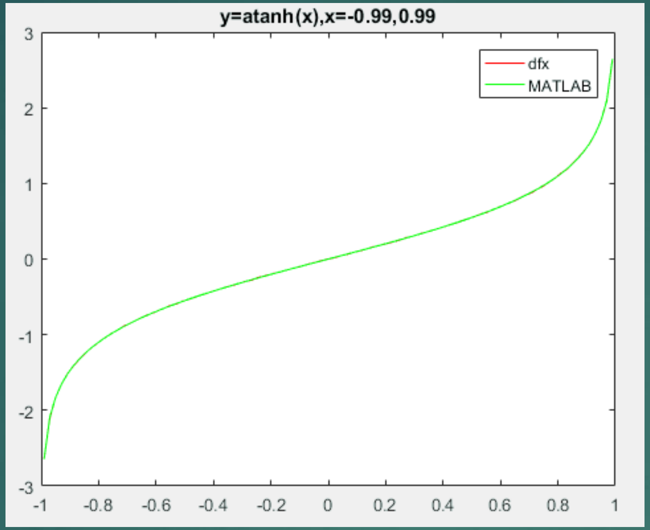
Pipelined Expanded Hyperbolic Status

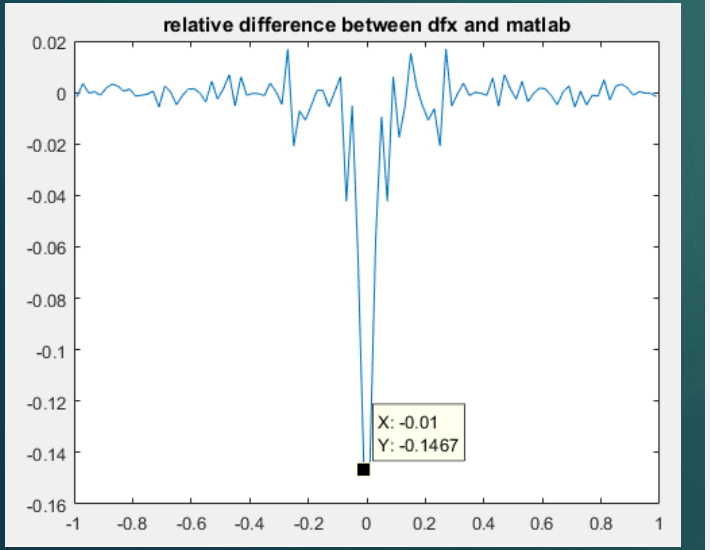


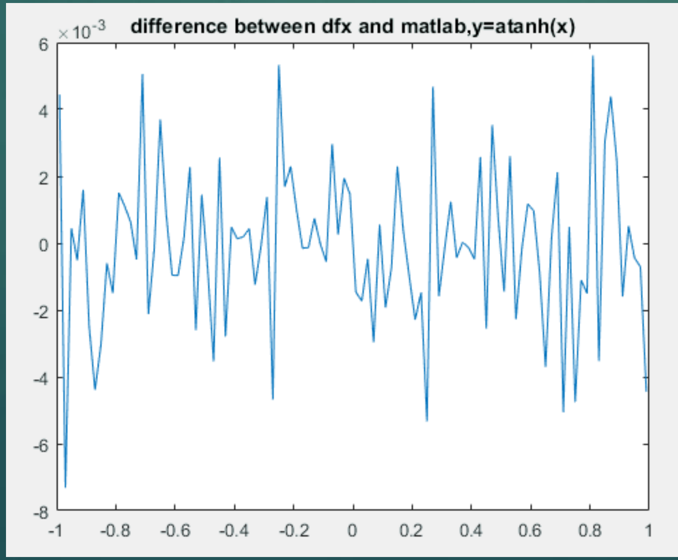


Pipelined Expanded Hyperbolic

input:
xin = 1
yin = x
zin = 0



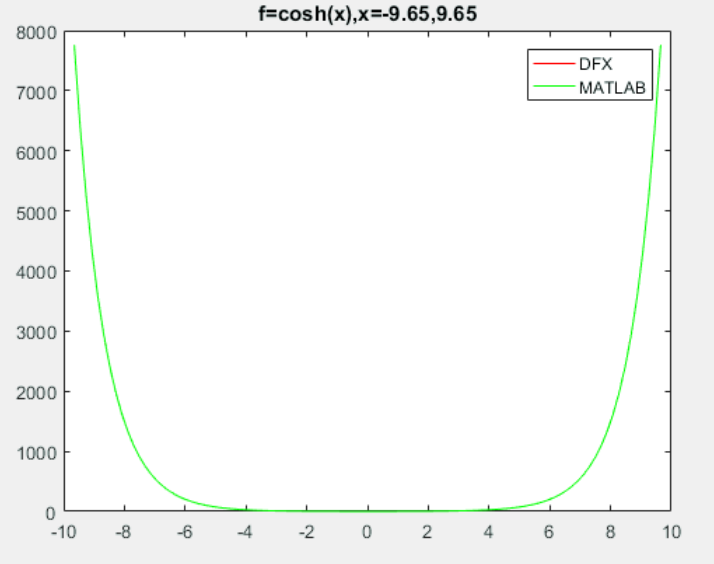


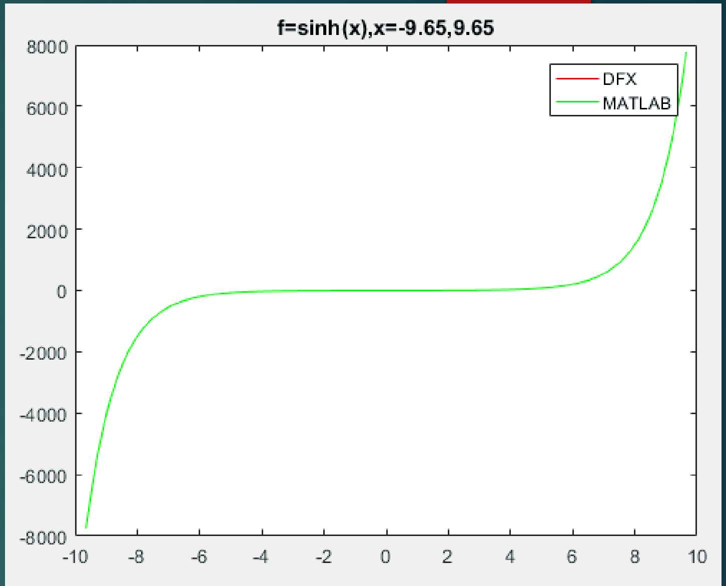


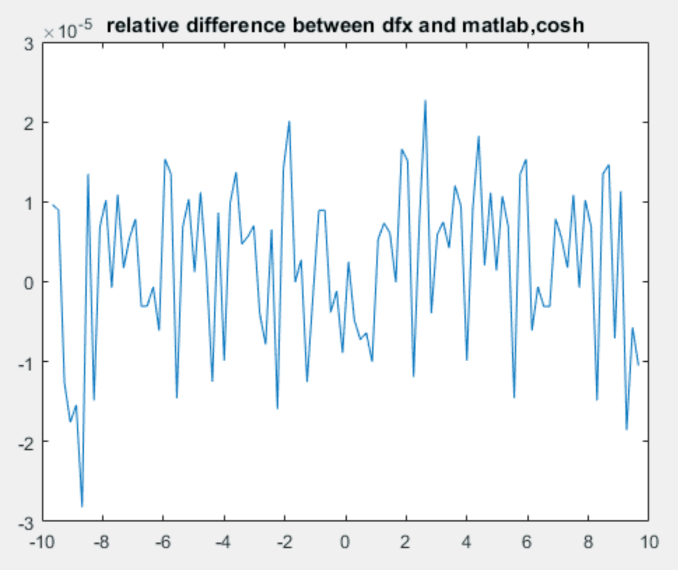
	sigma	xin	yin	zin
vector	1	1	0.01	0
	-1	0.990156	-0.97438	2.422094
	-1	0.04623	-0.01516	0.350526
	1	0.032017	0.02818	-1.36647
	1	0.007359	0.000165	-0.01244
	-1	0.007236	-0.00535	0.960513
	-1	0.004558	-0.00174	0.411207
	-1	0.004124	-0.0006	0.155794
	-1	0.00405	-8.2E-05	0.030137
	1	0.004044	0.000172	-0.03244
	1	0.004039	4.52E-05	-0.00118
	-1	0.004038	-1.8E-05	0.014441
	1	0.004038	1.36E-05	0.006629
	-1	0.004038	-2.2E-06	0.010535
	1	0.004038	5.73E-06	0.008582
	1	0.004038	1.78E-06	0.009559
	-1	0.004038	-1.9E-07	0.010047
	1	0.004038	7.98E-07	0.009803
	1	0.004038	3.05E-07	0.009925
	1	0.004038	5.89E-08	0.009986
	-1	0.004038	-6.4E-08	0.010016
	-1	0.004038	-2.7E-09	0.010001

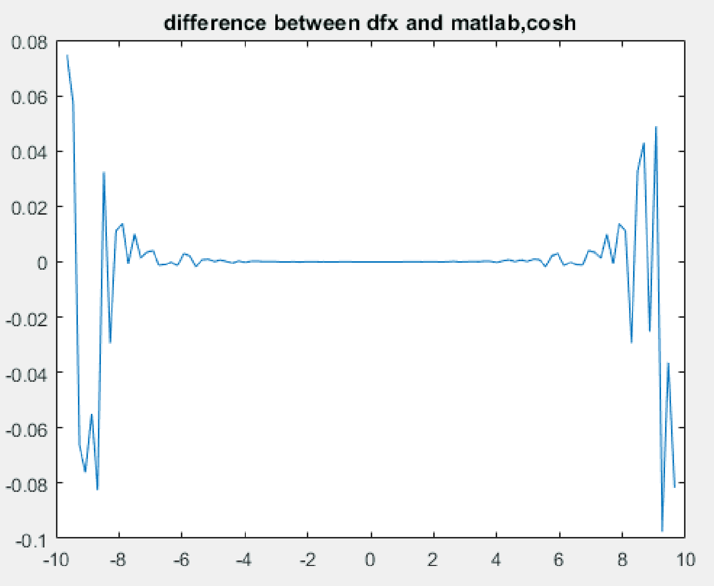
Pipelined Expanded Hyperbolic

input:
xin = 1/an
yin = 0
zin = x



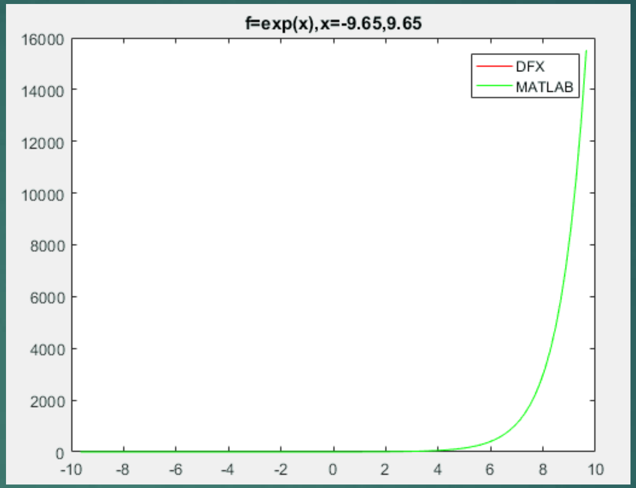


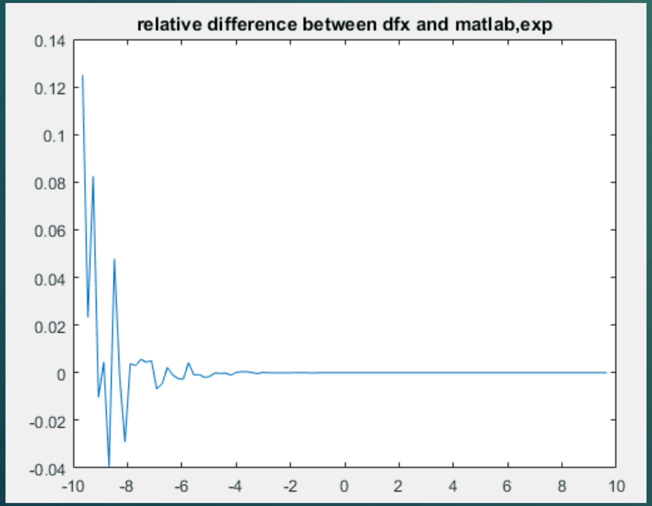


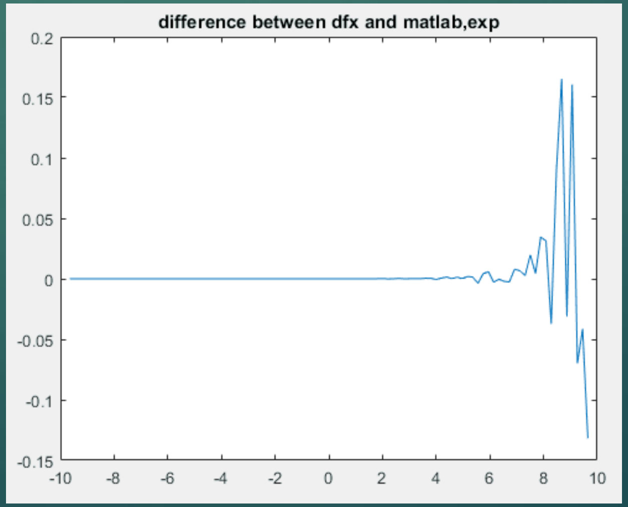


Pipelined Expanded Hyperbolic

input:
xin = 1/an
yin = 1/an
zin = x





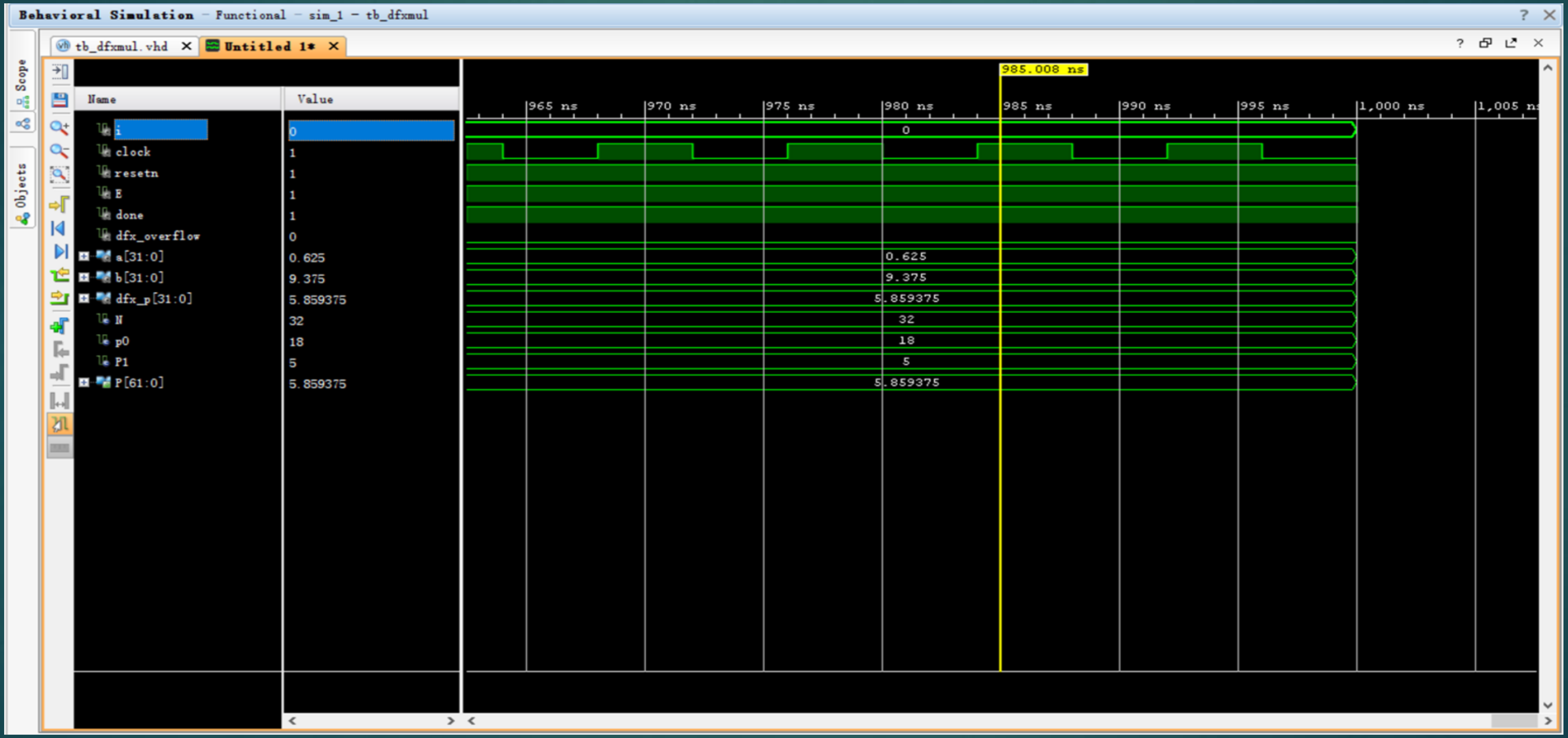


Pipelined Multiplier

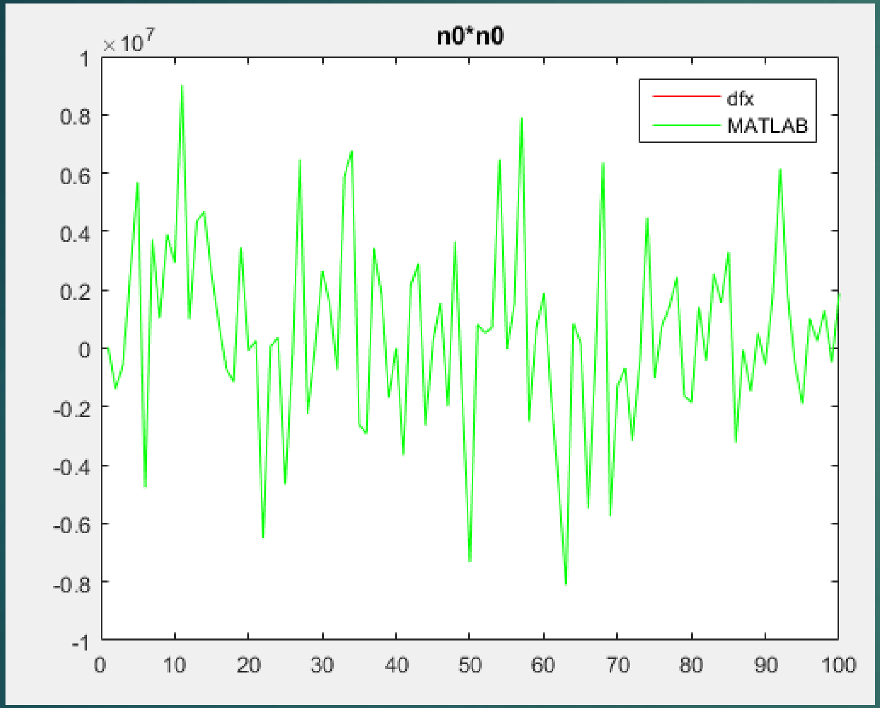
Arithmetic

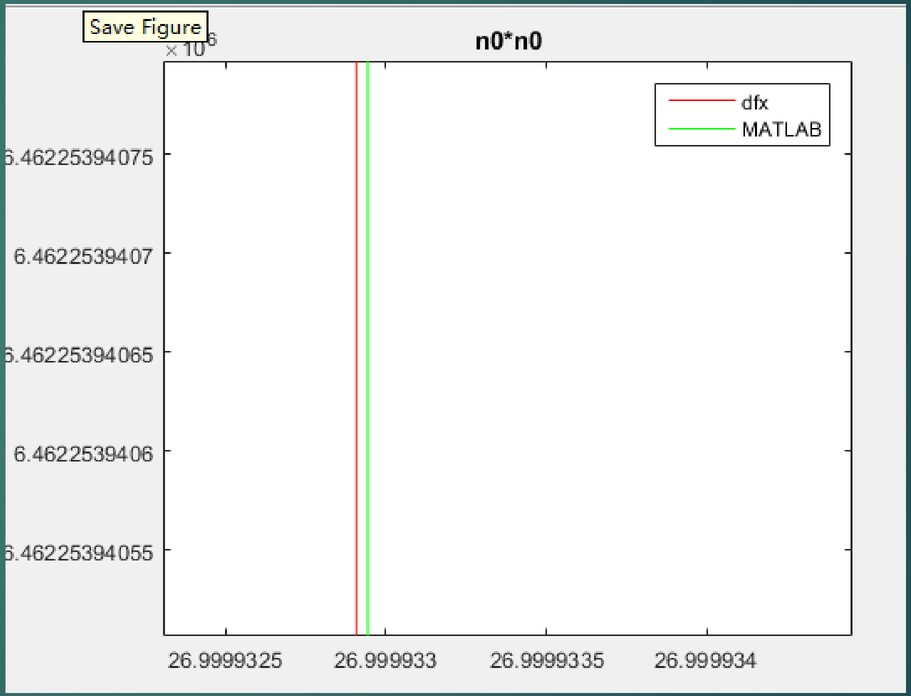
1. Signed number to unsigned number.
2. Two DFX inputs with N bits.
3. After multiplication, the result is 2N-2 bits.
4. Three situations: $n0 \times n0$, $n0 \times n1$, $n1 \times n1$ which has 2p0, p0+p1, 2p1 fractional bits respectively.
5. Use range detector to determine if the result is n0 or n1.
6. Unsigned to signed.

Pipelined Multiplier



Pipelined Multiplier





Project Management

Effort

- Furzana : Divider, FSM
- Jing : Expanded Hyperbolic, MATLAB scripts
- Zhongda : Multiplier

Conclusion

- Apply the knowledge of DFX arithmetic in coding.
- Apply the knowledge of different interfaces between FIFO and IP.

Future Work

- Increase the operations supported such as power and exponent.
- Test more dual fixed format to find out the most suitable one.