OSU ECEN 4233 HSCA, Spring 2024

Take-Home Test 2

Instructor: James E. Stine, Jr.

Assigned: Monday, 11/10, 2024
Due **Saturday 11/6, 2024** (midnight)
Handin: `http://canvas.okstate.edu`

Design the following items:

1. Using the Perl CSAM generator, generate a carry-save array multiplier for $n = 16$ and $m = 12$ in Verilog. Modify the SystemVerilog to handle two's complement numbers and test with 256 valid two's complement vectors. Hint: do not shortcut the mathematical process – formulate the equation just as we did in class to understand how to modify the SystemVerilog to handle two's complement arithmetic.

2. Show a dot diagram of a 16-bit Dadda multiplier and its reduction tree. Give a final area and delay analysis of the multiplier along with a final carry-propagate adder. To make things easier to grade, please use a ripple-carry adder for the final adder.

3. Write a program that displays a binary32 and binary64 floating-point number in decimal and hex based on an input. You can use any programming language you like.

4. For the reciprocal function and using piecewise constant approximation, implement the lookup assuming 16 piecewise partitions. What is the accuracy of this approach?

5. Using Goldschmidt's algorithm, show the division of $X = 1.93028319$ and $Y = 1.01902031$ for 16 bits. You can assume the inputs are $[1, 2)$ and that you need to figure out the number of iterations that is needed to use RNE to the given size. Show each iteration and the final answer properly rounded.

6. Peform the same as the previous problem, but compute the square root of $X = 1.01902031$.

Please use your DLD text [1] as well as notes on Canvas to help you get started. You can obtain Perl for your machine at `https://www.perl.org/get.html`. To get the CSAM Perl script to run, you may have to modify the location of the perl engine on the first line. Contact me, if needed, for help.

References

[1] S. Harris and D. Harris, *Digital Design and Computer Architecture, RISC-V Edition*. Elsevier Science, 2021.

①

$$P = X_{n-1} Y_{m-1} 2^{n+m-2} + \sum_{i=0}^{n-2} \sum_{j=0}^{m-2} X_i Y_j 2^{i+j}$$

$n=16 \quad m=12$

$$+ \sum_{i=0}^{n-2} \overline{Y_{m-1} X_i} \, 2^{m-1+i} + \sum_{j=0}^{m-2} \overline{X_{n-1} Y_j} \, 2^{n-1+j}$$

$$+ \left(2^{m-1+0} + 2^{n-1+0}\right) + \left(2^{m-1+(n-2+1)} + 2^{n-1+(m-2+1)}\right)$$

This is the bottom left most cell

This is the row at the bottom

These are the normal cells

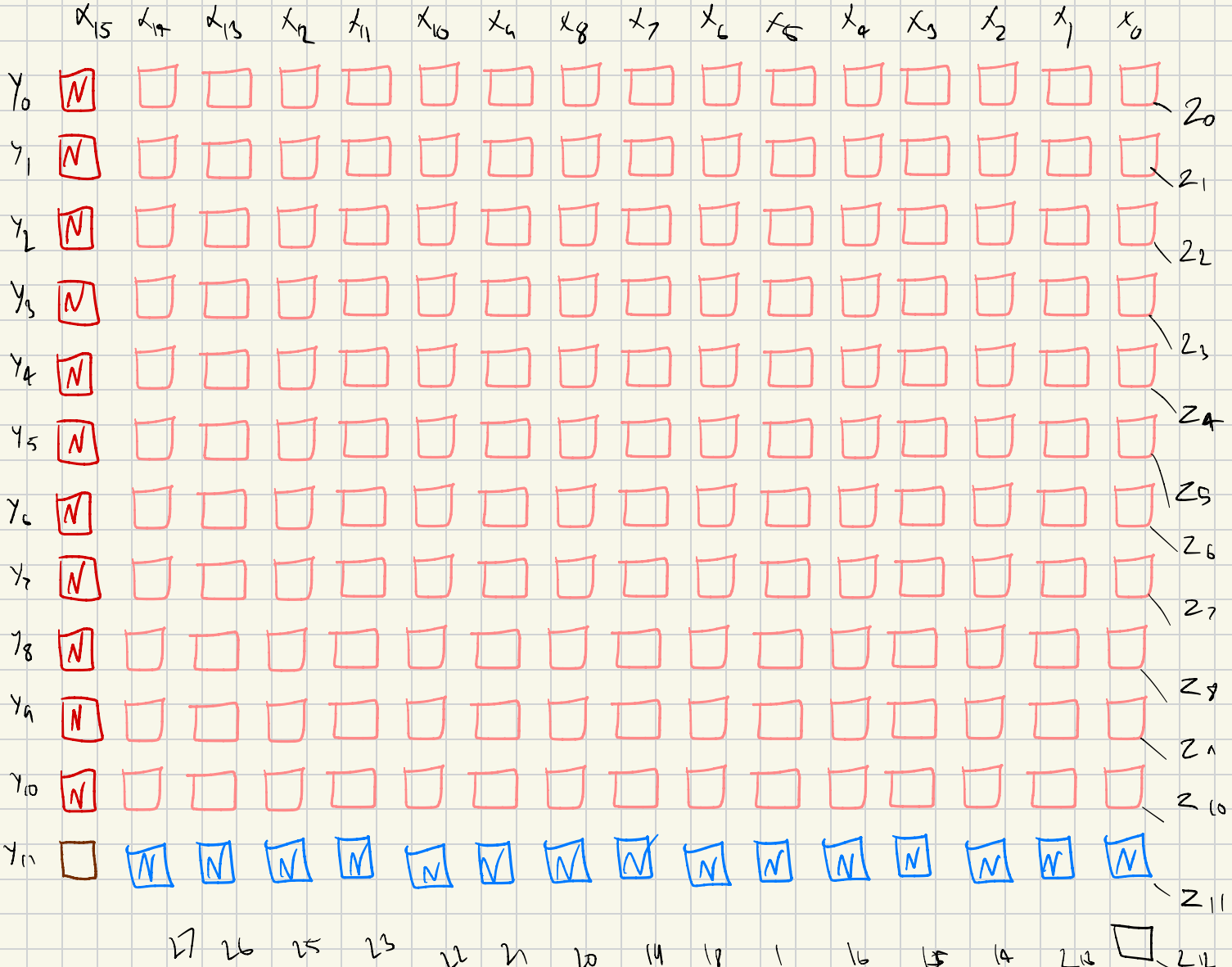
This is the n-th left column

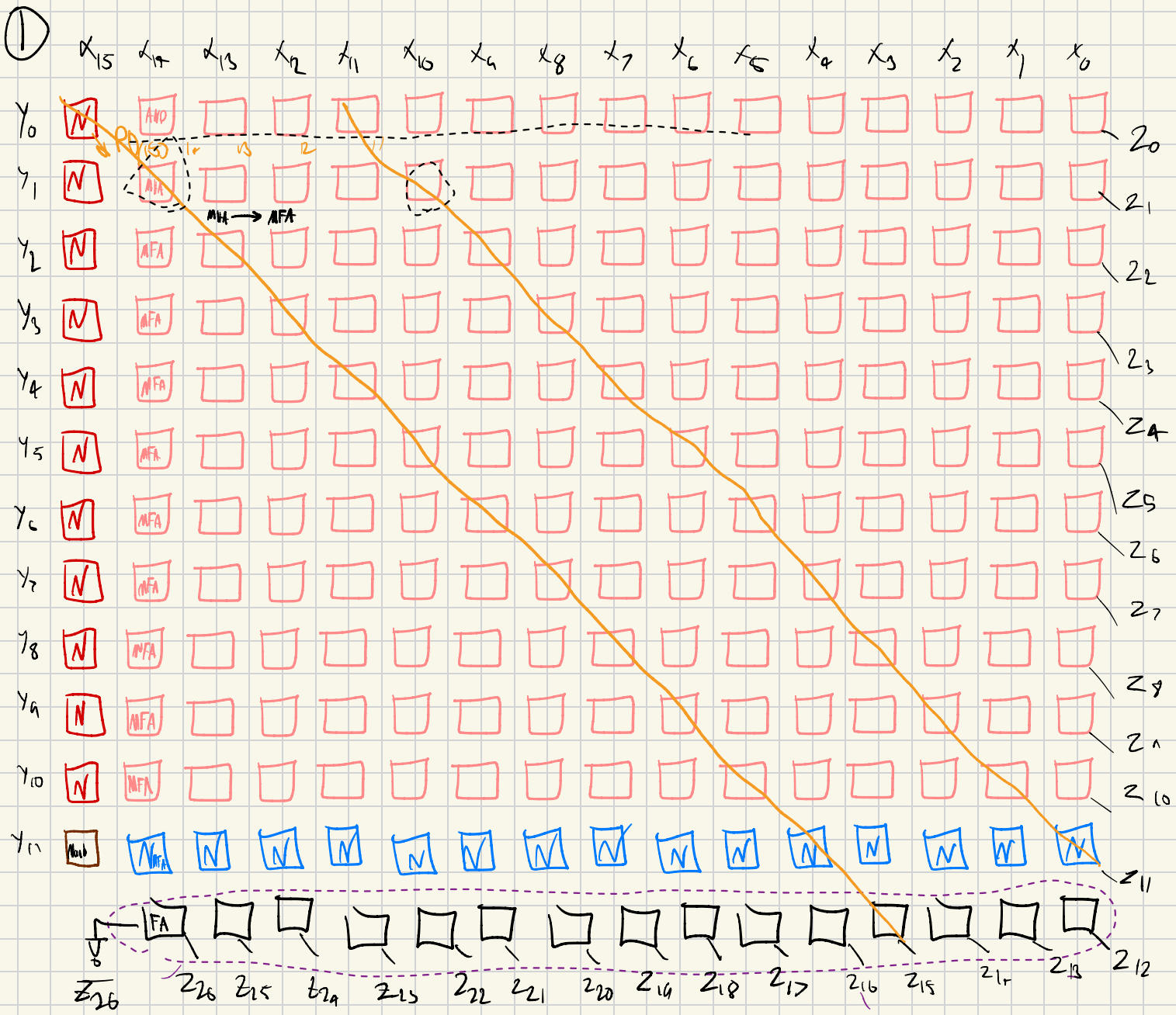
$$= X_{15} Y_{11} 2^{26} + \sum_{i=0}^{14} \sum_{j=0}^{10} X_i Y_j 2^{i+j}$$

$$+ \sum_{i=0}^{14} \overline{Y_{11} X_i} \, 2^{11+i} + \sum_{j=0}^{10} \overline{X_{15} Y_j} \, 2^{15+j}$$

$$+ \left(2^{15} + 2^{11}\right) + 2^{27}$$

I know





This is the bottom left most cell

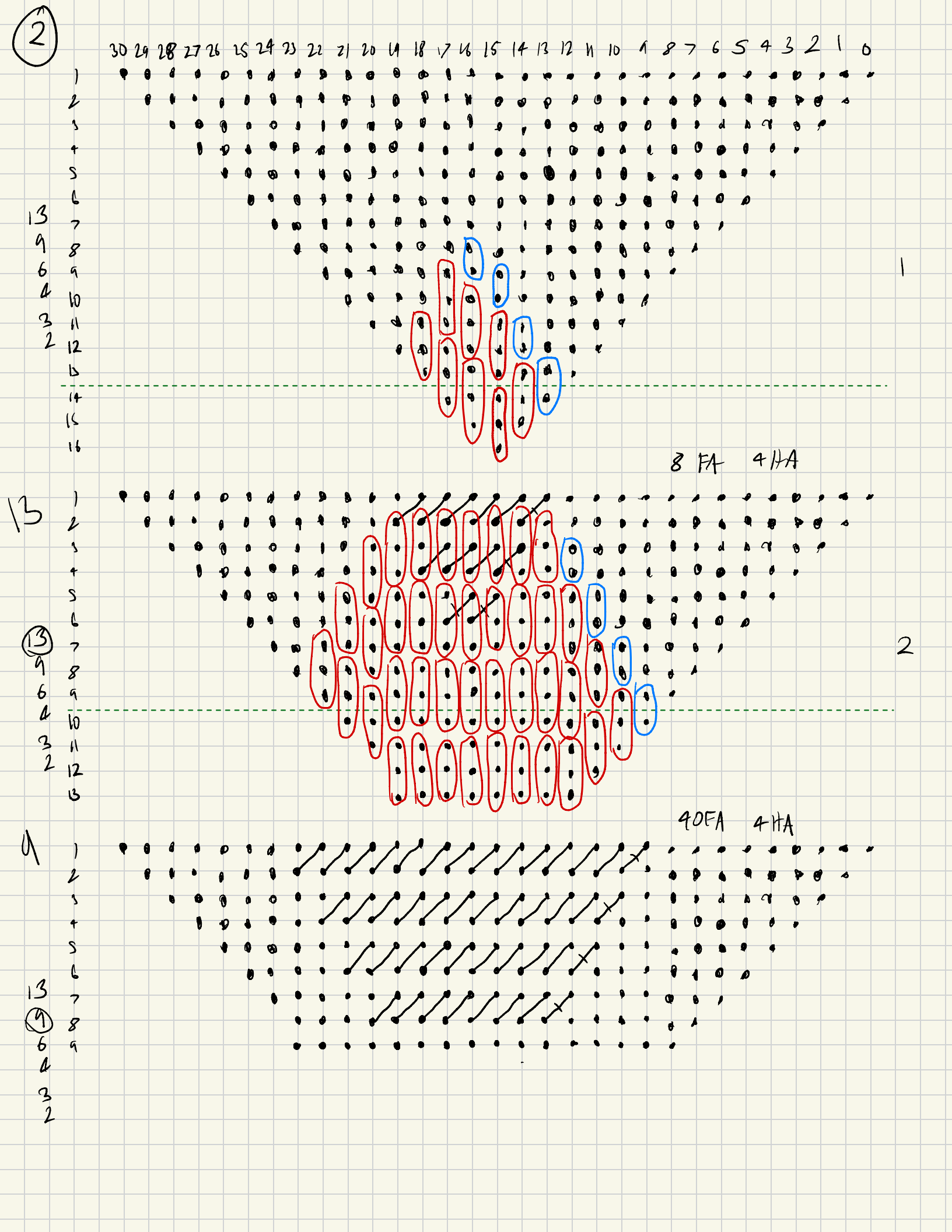
This is the row at the bottom

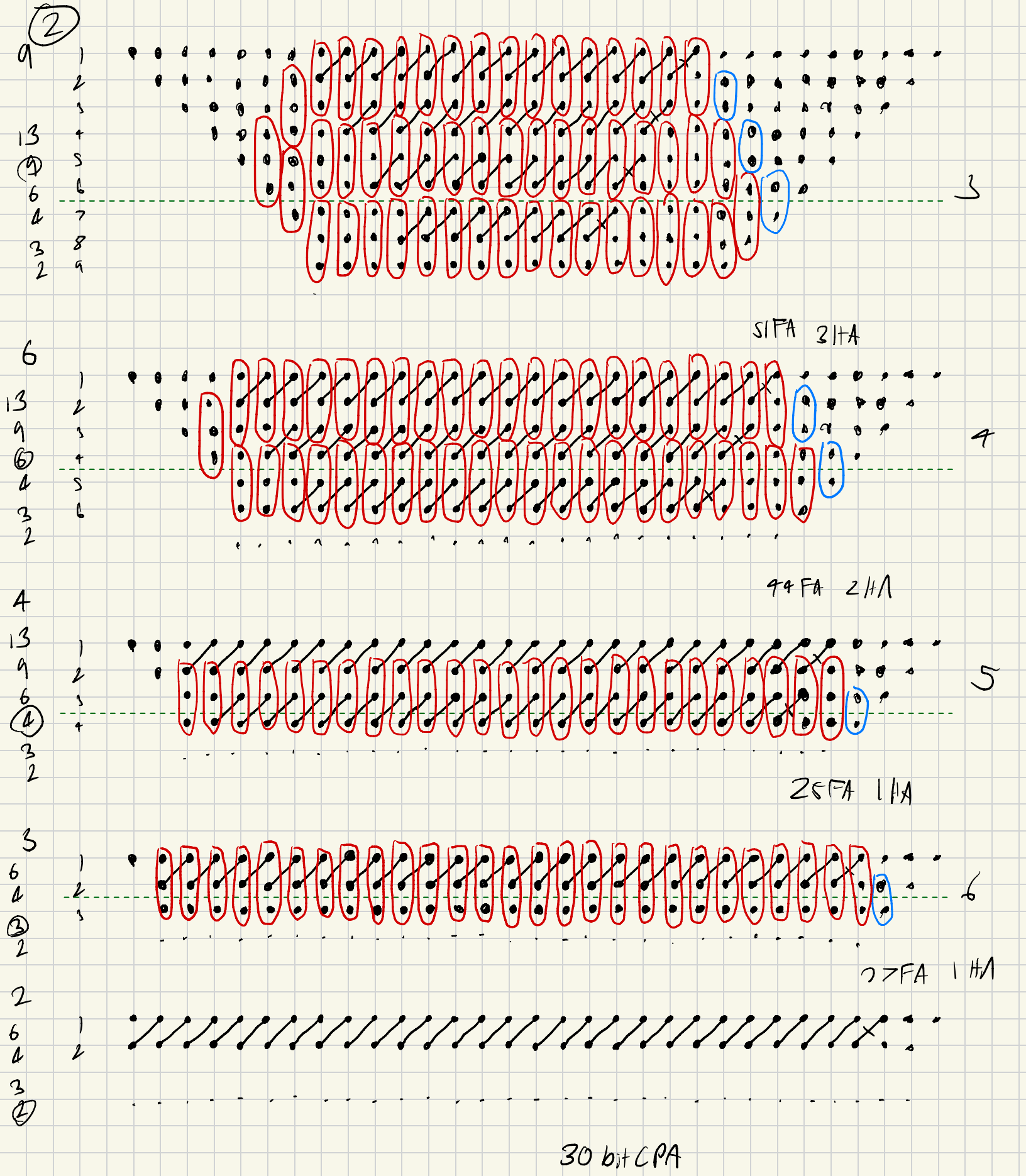
These are the normal cells

This is the And at the left column

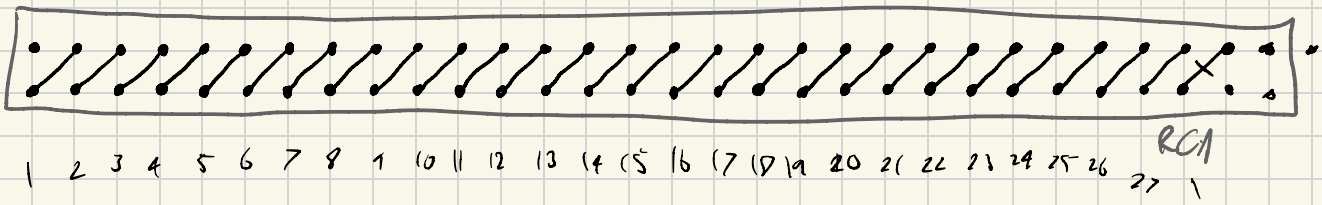
$$= x_{15} y_{11} 2^{26} + \sum_{i=0}^{14} \sum_{j=0}^{10} x_i y_j 2^{i+j} + \sum_{i=0}^{14} \overline{y_{11} x_i}\, 2^{11+i} + \sum_{j=0}^{10} \overline{x_{15} y_j}\, 2^{15+j} + \left(2^{15} + 2^{11}\right) + 2^{27}$$

Add ons to bits


2
30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
13 9 6 4 3 2
1
8 FA 4 HA
13
2
13 9 6 4 3 2
4
40FA 4 HA
13 9 6 4 3 2


2
3
SIFA 31HA
4
99FA 2HA
5
25FA 1HA
6
27FA 1HA
30 bit CPA

(2)



HA (2,2) = 4 gates
3Δ delay (Sum)

FA [3,2] = 9 gates
6Δ delay (Sum)

RCA 9n gates
2n+4 delay

FA	HA
8	9
40	4
51	3
44	2
25	1
27	1
195	15

Overall Area:

195(9 gates) + 15(4 gates) + 9(30) gates + 1 gate

Area = 2086 gates

Overall Delay:

6 layers

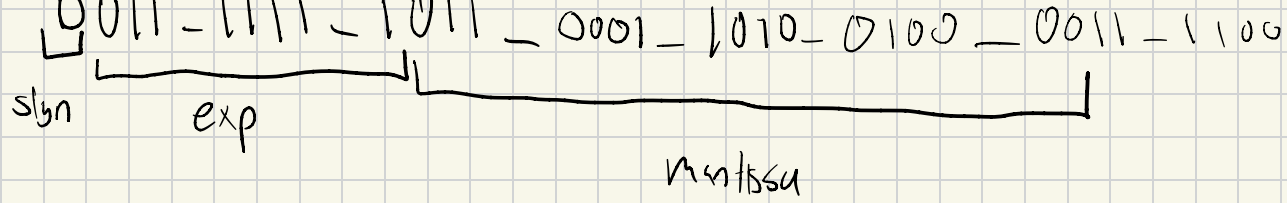
∴

6(6Δ) + 2(30)+4 + 1Δ

Delay = 101Δ

(3) Python Program

Done in Prob3 folder.



(4)

Assuming $[1, 2)$

$2^m = 16 \Rightarrow m = 4$

Subintervals = partitions

$2^{-m} = 2^{-4} = 0.0625$

$$\left(\frac{f(a) + f(b)}{2}\right)$$

$$\text{Output} = \frac{\left(\frac{1}{x_a} + \frac{1}{x_b}\right)}{2}$$

U0.4

input	modified input	input (bin)	Output	Output U1.15 (hex)
0	1	0000	0.970586235204	7C3C
1	1.0625	0001	0.915032679739	751F
2	1.125	0010	0.865447078023	6EC8
3	1.1875	0011	0.821052631579	6918
4	1.25	0100	0.780952380952	63F6
5	1.3125	0101	0.744589744589	5F4E
6	1.375	0110	0.711462450593	5B11
7	1.4375	0111	0.681154942029	5750
8	1.5	1000	0.653333333333	5340
9	1.5625	1001	0.627642307692	5058
10	1.625	1010	0.603988603989	4D4F
11	1.6875	1011	0.582010582011	4A7F
12	1.75	1100	0.561576754408	47E1
13	1.8125	1101	0.542527573532	4571
14	1.875	1110	0.524781121746	4324
15	1.9375	1111	0.508064516129	4108

For 1 to 2 error would be $\varepsilon = 1 - \frac{1+\frac{1}{2}}{2} = 0.25$, 25%

but the highest error between each of the values

would be $a = 1.875$ $b = 1.9375$, where the

error is $1 - \frac{\left(\frac{1}{1.875} + \frac{1}{1.9375}\right)}{2} = 0.47527$ or 47.53%

(5)

Initial Approximation = 0.75

1.93028319	N(X)		IA(K_0)	0.75	
1.01902031	D(Y)				
	q*K	r*K	2-D*Xi (K)	Error	bit
0	1.447712393	0.764265233	1.235734768	0.446541501	1.16313383
1	1.788988537	0.944429119	1.055570881	0.105265357	3.247897374
2	1.888404205	0.996911877	1.003088123	0.005849689	7.417424462
3	1.894235829	0.999990463	1.000009537	1.80646E-05	15.75647864
4	1.894253894	1	1	1.72273E-10	32.43458549
5	1.894253894	1	1	2.22045E-16	52
6	1.894253894	1	1	2.22045E-16	52
7	1.894253894	1	1	2.22045E-16	52
8	1.894253894	1	1	2.22045E-16	52

iteration	$q_i = q_{i-1} \cdot k_{i-1}$	$r_i = r_{i-1} \cdot k_{i-1}$	k_i	$\text{abs}(q_i - (\frac{N}{D}))$	$-\log_2(\text{error})$

$$\text{Approx} = \frac{N}{D}\left(\frac{k_0}{k_2} \cdots \frac{k_n}{k_n}\right) \qquad (2 - D x_i)$$

* $x_i = q_i$

* Used Excel File is in probS folder *

If my initial approximation is 0.75, it will take 4 iterations for the use of RNE to give a 16 bit accuracy.

1.894253894

Sign = 0

exp = 0 + bias(15) = 15 = 0111

0.894253894 => 1110_0100_1111 (by multiplying by 2)

mantissa (has to be 10 bits only)

11_1001_0011

+1

11_1001_0100 need to round so add 1

0_0111_1110010100 or 1.89453125

(6)

Doesn't factor into this problem

[redacted]			IA(K_0)	0.75	
1.01902031	D(Y)				
i	q*K	q*K*D	(3-D*xi^2)/2	Error	bit
0	0.75	0.764265233	1.213400538	0.245200126	2.027968375
1	0.910050403	0.927359844	1.0780279	0.082105514	3.606377071
2	0.981059725	0.999719785	1.009607591	0.009745573	6.681037232
3	0.990485346	1.009324684	1.000139346	0.000140674	12.79535354
4	0.990623366	1.009465329	1.000000029	2.94042E-08	25.01940301
5	0.990623394	1.009465358	1	1.11022E-15	49.67807191
6	0.990623394	1.009465358	1	0	#NUM!
7	0.990623394	1.009465358	1	0	#NUM!
8	0.990623394	1.009465358	1	0	#NUM!

Iteration	$q_i = q_{i-1} \cdot k_{i-1}$ Reciprocal Value $\frac{1}{\sqrt{x}}$	$q_i \cdot D$ (Actual value $\sqrt{x}$)	$k_i = \frac{(3 - D \cdot x_i^2)}{2}$ Approximation	$q_i - \sqrt{D}$ Error	Bits $-\log_2(\text{Error})$

$\frac{1}{\sqrt{x}} = \text{Approx} = \frac{(3 - D \cdot x_i^2)}{2}$ * $x_i = q_i$ *

* In Prob 6 folder *

If my initial approximation is 0.75, it will take 4 iterations for RNE to get a 16-bit accuracy.

0.990623366

Sign = 0

exp = 1 + bias(15) = 14 = 01110

0.990623366 => 1111 1101 1001 because 7 > 5

11-1110-1100 Rounding

+1

11-1110-1101

0-01110-1111101101