Christian Tonnesen

Assignment 1.1

ID: 260847409

Sunday, January 24, 2021 5:20 PM

1.

n	0 .	1	1	0	1	0	1
i	0	-1	-2	-3	-4	-5	-6
b^i	1	1/2	1/4	1/8	1/16	1/32	1/64
nb^i	0	1/2	1/4	0	1/16	0	1/64

32/64 16/64 4/64 1/64 = 53/64 = 0.828125_{10}

2. $8.625 = 8_{10} + 0.625_{10}$

a.

$2\sqrt{8} = 4$, -8, 0 ; $2\sqrt{4} = 2$, -4, 0 ; $2\sqrt{2} = 1$, -2, 0 ; $2\sqrt{1}$, -0, 1

$8_{10} = 1000_2 \longrightarrow 1000.101_2$

0.625 × 2 = 1.250 → 1

0.250 × 2 = 0.50 → 0

0.5 × 2 = 1 → 1

$0.625_{10} = 101_2$

b. 1000.1010_2 → 8_{16} . A_{16}

$8.625_{10} = 1000.1010_2 = 8.A_{16}$

3.

a.

n	2	T	D
En	2	29	13
i	2	1	0
b'	32^2	32^1	32^0

n	0	3	F
En	24	3	15
i	-1	-2	-3
b'	32^{-1}	32^{-2}	32^{-3}

$2989.753387451171875_{10}$

$2989 \le 2^n - 1$

$2990 \le 2^n$

$\log_2(2990) \le n$

$11.5 \le n$

i	11	10	9	8	7	6	5	4	3	2	1	0
b^i	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
S	2048	1024	512	256	128	64	32	16	8	4	2	1
a_i	1	0	1	1	1	0	1	0	1	1	0	1

$2989_{10} = 101110101101_2$

13
928
2048
————
2989

$2048\sqrt{2989} = 1$, -2048, 941 ; $512\sqrt{941} = 1$, -512, 429 ; $256\sqrt{429} = 1$, -256, 173 ; $128\sqrt{173} = 1$, -128, 45

$32\sqrt{45} = 1$, -32, 13 ; $8\sqrt{13} = 1$, -8, 5 ; $4\sqrt{5} = 1$; $1\sqrt{1} = 1$

0.753387451171875 × 2 = 1.50677

0.5067749 × 2 = 1.01355

0.01355 × 2 = 0.0217

0.0217 × 2 = 0.0542

0.0542 × 2 = 0.1084

0.1084 × 2 = 0.2168

0.2168 × 2 = 0.43359

0.43359 × 2 = 0.86719

0.86719 × 2 = 1.73438

0.73438 × 2 = 1.46875

0.46875 × 2 = 0.9375

0.9375 × 2 = 1.875

0.875 × 2 = 1.75

0.75 × 2 = 1.5

0.5 × 2 = 1.0

$0.533874511718 75_{10} = 0.110000001111$

$2989.753387451171875_{10} = 101110101101.110000001111_2$

b. $101110101101.1100000011011110_2$ → B A D . C O D E

$101110101101.1100000011011110_2 = BAD.CODE_{16}$

4.

i	8	7	6	5	4	3	2	1	0
En	2	1	2	3	3	6	6	1	4
b^i	7^8	7^7	7^6	7^5	7^4	7^3	7^2	7^1	7^0
S	11529602	823543	117649	50421	7203	2058	294	7	4

12353145
12588443
12622057
12638864
12646067
12648125
12648419
12648426

$12648430 \le 16^n - 1$

$12648431 \le 16^n$

$\log_{16}(12648431) \le n$

$5.898 \le n$

i	5	4	3	2	1	0
En	16^5	16^4	16^3	16^2	16^1	16^0
b^i	1048576	65536	4096	256	16	1
a_i	12	0	15	15	14	14

$1048576\sqrt{12648430} = 12$, -1048576, 2162670, -2097152, 65518

$4096\sqrt{65518} = 15$, -4096, 24558, -20480, 4078

$256\sqrt{4078} = 15$, -256, 1518, 1280

$16\sqrt{238} = 14$, 16, 78, 64

12646067
12648125
12648419
12648426
12648430_{10}

$256\overline{)4078}$ → 15, −256, 1519, −1280, 238

$16\overline{)238}$ → 14, 16, 78, 64, 14

$1\overline{)14}$ → 14, −14

$212336619_7 = COFFEE_{16}$

5.

$25062 \leq 2^n - 1$

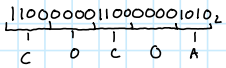
$25906 \leq 2^n$

$\log_2(25063) \leq n$

$17.9 \leq n$

i	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a_i	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	1	0
a_i^{-1}	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1
+1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0

20 bit Signed = 11000000110000001010_2



$11000000110000001010_2 = C0C0A_{16}$

259062/2 = 129,531 + 0
1295 1/2 = 64765 + 1
64765/2 = 2 2 + 1
2 2/2 = 16191 + 0
16191/2 = 095 + 1
095/2 = 4047 + 1
4047/2 = 202 + 1
202 /2 = 1011 + 1
1011/2 = 505 + 1

505/2 = 252 + 1
252/2 = 126 + 0
126/2 = 6 + 0
63/2 = 1 + 1
1/2 = 15 + 1
15/2 = 7 + 1
7/2 = 3 + 1
/2 = 1 + 1
1/2 = 0 + 1

6. $-4.625_{10} = 100.101_2$

Normalized = $1.00101 \cdot 2^2$

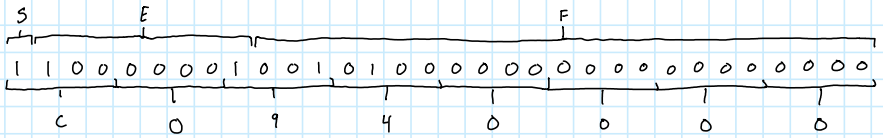
S = 1

L = 2 + 127 = 29

$129_{10} = 10000001_2$

129/2 = 64 + 1
64/2 = 32 + 0
32/2 = 16 + 0
16/2 = 8 + 0
8/2 = 4 + 0
4/2 = 2 + 0
2/2 = 1 + 0
1/2 = 0 + 1

F = 00101 + 18 0's



$-4.625_{10} = C0940000_6$

This representation is accurate as −4.625 does not repeat, which, if it did, would result in precision error when using IEEE-754 standard

Assignment 1.2

Thursday, January 25, 2021 8:36 PM

a_3	a_2	a_1	a_0	S_0	S_1	S_2	S_3	S_4	S_5	S_6
0	0	0	0	1	0	1	1	1	1	0
0	0	0	1	1	1	1	0	1	1	1
0	0	1	0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	1	1	0	1	0	0	1
0	1	0	1	0	0	1	1	0	0	1
0	1	1	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	1	1
1	0	0	0	0	1	1	1	0	0	1

★ Did not include other 8 rows as we don't care about numbers greater than $1000_2 = 8_{10}$

I began by building the Logisim circuit first, then building the truth table. For each number between 0 and 8, there is a submodule of NOR and AND gates to ensure that the buttons that needs to be on are pressed (passed through regular AND) and the ones that need to be off are off (passed through NOR and then connected to the previously mentioned AND). These result in 8 different output streams, which each need to turn on certain segments. However, I needed to do this such that I wouldn't cross wires between any of the submodule outputs. Thus, I assigned each segment its own 8-input OR gate so that if it detected a signal from any of the 8 streams, it would turn on the segment and not cross wires with the other streams. Overall, no such algebra was really involved, more of a brute force to ensure no unwanted signals turned on. This also has the added effect of not producing any segment lights when a binary number past "1000" is turned on with button combos.