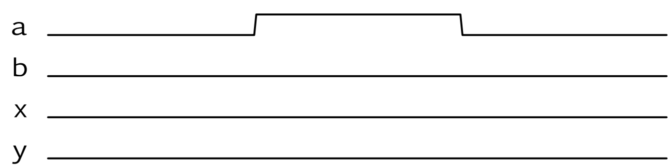
1. (a) $a = \perp$ and $b = \perp$,

Set $x = \perp$ and $y = \perp$, the output is still $x = \perp$ and $y = \perp$; hence, the circuit stays stable.
Set $x = \perp$ and $y = \top$, then the output chnage to $x = \perp$ and $y = \perp$; hence, unstable.
Set $x = \top$ and $y = \perp$, then the output changes to $x = \top$ and $y = \top$; hence, unstable.
Set $x = \top$ and $y = \top$, the output is still $x = \top$ and $y = \top$; hence, the circuit stays stable.

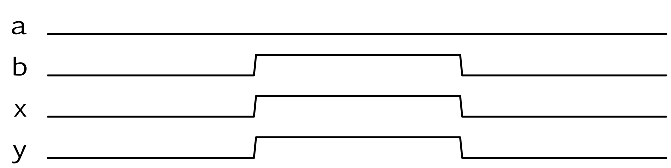
The stable states are $(a, b, x, y) = (\perp, \perp, \perp, \perp)$ or $(a, b, x, y) = (\perp, \perp, \top, \top)$.

(b) For stable state $(a, b, x, y) = (\perp, \perp, \perp, \perp)$, when you apply a pulse,

i. apply pulse to a:

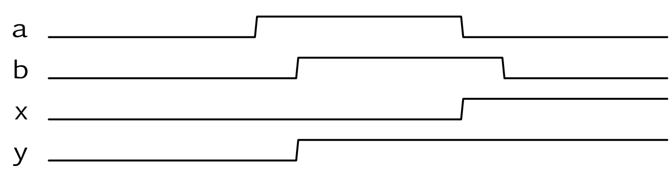


ii. apply pulse to b:

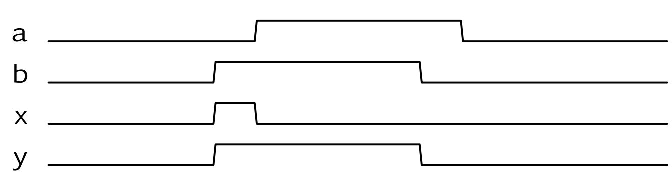


iii. apply pulse to both a and b:

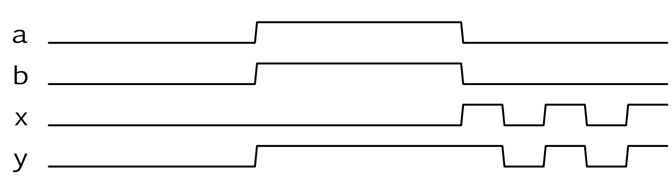
if a happens first:



if b happens first:

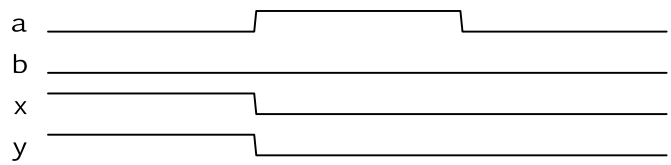


if a and b happen at same time:

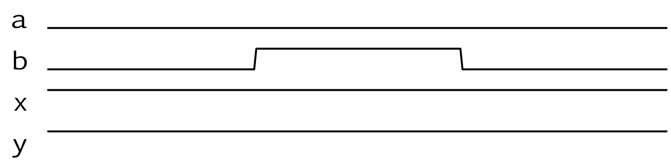


For stable state $(a, b, x, y) = (\perp, \perp, \top, \top)$, when you apply a pulse,

i. apply pulse to *a*:

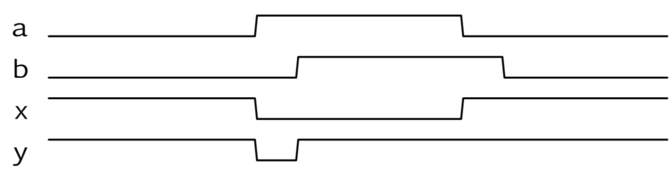


ii. apply pulse to *b*:

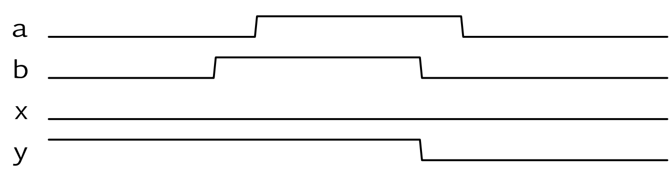


iii. apply pulse to both *a* and *b*:

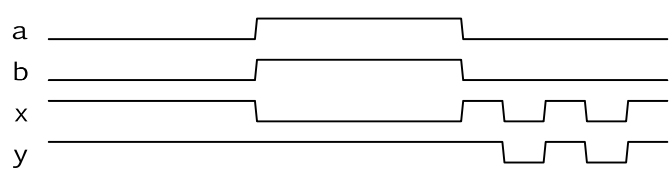
if *a* happens first:



if *b* happens first:



if *a* and *b* happen at same time:



2. We know that the $JKlatchs$

J	K	Q
$\bot$	$\bot$	$\triangleleft Q$
$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\top$
$\top$	$\top$	$-\triangleleft Q$

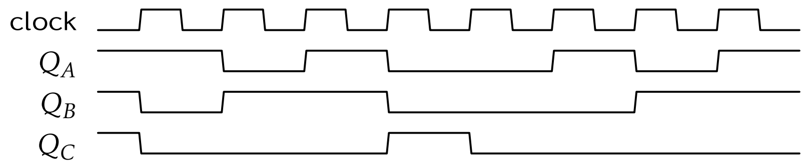
Also, it is clearly to see that:

$$J_A = -\triangleleft Q_C;\ K_A = -\triangleleft Q_C$$

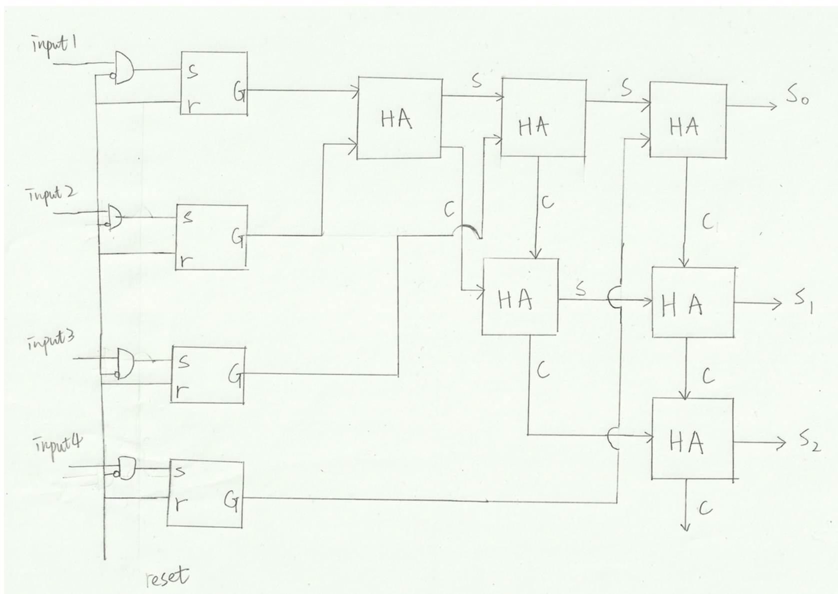
$$J_B = \triangleleft Q_A;\ K_B = \triangleleft Q_A$$

$$J_C = \triangleleft Q_A \wedge \triangleleft Q_B;\ K_C = \triangleleft Q_C$$

from the tables above we can easily come up with the diagram:



3. Here we show the circuit for $n = 4$:



4. (a) convert -42 to IEEE single-precision floating point.

   -42 in binary is -101010,

42	
21	0
10	1
5	0
2	1
1	0
0	1

   which is $-1.01010 \times 2^5 = -1.01010 \times 2^{132-127}$;
   similarly we get 132 in binary is 10000100;
   IEEE single-precision standard: [1 10000100 01010000000000000000000].

   (b) convert 3.14 to IEEE single-precision floating point.

0.14	
0.28	0
0.56	0
1.12	1
0.24	0
0.48	0
0.96	0
1.92	1
1.84	1
1.68	1
1.36	1
0.72	0
1.44	1
0.88	0
1.76	1
1.52	1
1.04	1
0.08	0
0.16	0
0.32	0
0.64	0
1.28	1

3 in binary is 11, and 0.14 in binary is $0.0\,\overline{01000111101011100001}$
3.14 in binary is $11.0\,\overline{01000111101011100001} = 1.10\,\overline{01000111101011100001} \times 2^1$;
normalize $314_2 = 1.10\,\overline{01000111101011100001} \times 2^{128-127}$;
128 in binary is 10000000;
IEEE single-precision standard: [0 10000000 10010001111010111000010].

5. convert [0 10000011 01011010110101101011010] from IEEE single-precision floating-point to base ten.

   from the first bit 0 we can see that the number is positive;

   the next 8 bits 10000011 stand for exponent which is 131, then the exponent is $131 - 127 = 4$;

   then the next stand for significant, which is 1.35483860

   the number is $1.35483860 \times 2^4 = 21.677$ with 5 significant digits.

6. (a) In order to calculated $-\frac{5}{6}$, we first compute $\frac{1}{3}$, then $\frac{5}{3}$, then $\frac{5}{6}$ and finally $-\frac{5}{6}$

   compute $\frac{1}{3}$ is $1 \div 11 = 01'1$:

$$
\begin{array}{rrrr}
 & & & 1' \\
- & & 1 & 1 \\
\hline
 & & 1' & \\
- & 1 & 1 & \\
\hline
1' & 0 & &
\end{array}
$$

compute then $\frac{5}{3}$ is $01'1 \times 101 = 10'111$:

$$\begin{array}{r} 0\ 1'\ 1 \\ +\ 01'\ \ 1\ \ 1\ \ \ \ \ \ \\ \hline 10'\ \ 1\ \ 1\ 1 \end{array}$$

compute then $\frac{5}{6}$ by shifting the decimal point:

$$10'11.1$$

compute $\frac{-5}{6}$ by 2's complement:

$$01'00.1$$

(b) We know that 7 is binary is 111; hence, int binary quote notation is 111 as well.

(c) compute $01'00.1 - 111 = 10'01101.1$:

$$\begin{array}{r} -\ \ \ 01'\ 0\ 0.\ 1 \\ 0'1\ 1\ 1\ \ \ \ \ \ \\ \hline 10'0\ 11\ 0\ 1.\ 1 \end{array}$$

(d) generate from the quote notation paper we have the formula:
let

$$x = 10$$

$$y = 01101.1$$

$$a = 1$$

$$w = sequence\ of\ a's\ the\ same\ length\ as\ x$$

$$z = digit\ 1\ followed\ by\ a\ sequence\ of\ zeros\ of\ the\ same\ length\ as\ y$$

then

$$(10'01101.1)_{10} = y - \frac{xz}{w} = 01101.1 - \frac{10 \times 100000}{11}$$

$$= 13\frac{1}{2} - \frac{64}{3}$$

$$= \frac{81}{6} - \frac{128}{3} = -\frac{47}{6} = -7.8\bar{3}$$

7. First convert decimal ASII to binary, and we have:
d in ASCII code is 100 in decimal, then 1100100 in binary;
o in ASCII code is 111 in decimal, then 1101111 in binary;
space in ASCII code is 32 in decimal,then 0100000 in binary;
i in ASCII code is 105 in decimal, then 1101001 in binary;
t in ASCII code is 116 in decimal,then 1110100 in binary;
Then using Hamming code to encode we have,

symbol	p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7
b	1	1	1	1	1	0	0	1	1	0	0

o	1	0	1	0	1	0	1	1	1	1	1
space	1	0	0	1	1	0	0	0	0	0	0
i	0	1	1	0	1	0	1	1	0	0	1
t	1	0	1	0	1	1	0	1	1	0	0

If the third bit of third character is flipped. Then we will get 10111000000 for the third character.

We know that :

$$c_1 = p_1 d_1 d_2 d_4 d_5 d_7$$
$$c_2 = p_2 d_1 d_3 d_4 d_6 d_7$$
$$c_3 = p_3 d_2 d_3 d_4$$
$$c_4 = p_4 d_5 d_6 d_7$$

Then, $c_1 = 1$, $c_2 = 1$, $c_3 = 0$ and $c_4 = 0$. Hence, we get 0011, then we know the third bit is flipped.