Exercise 1

a) i) $\log(256) = 8 \text{ bits}$

ii) $[-2^{n-1}-1 \,;\, 2^{n-1}]$

$$2^{n-1} = 1024 \Rightarrow n-1 = \log(1024)$$
$$\Rightarrow n-1 = 10$$
$$\Rightarrow n = 11 \text{ bits}$$

iii)
$$\begin{array}{r} 3581 \\ -\ 3325 \\ \hline 0256 \end{array} \Rightarrow 01$$

iv)
$$\begin{array}{r} {\scriptstyle 1\ 1\ \ } \\ 879 \\ +\ 657 \\ \hline 1536 \end{array}$$

$$1024 \cdot 2 > 1536$$

$$\Rightarrow 11 \text{ bits}$$

(b) $\lceil \log(b-a) \rceil$

(c) 2^n

(d)

	Decimal	Binary	Octal	Hex
	139	10001011	213	8B
	185	10111001	271	B9
	58	00111010	72	3A
	201	11001001	311	C9
	56	00111000	70	38
	211	11010011	323	D3
	27	00011011	33	1B
	108	01101100	154	6C

16
32
48
64
80
96
112
128
144
160
176
192
208

$139 = 16 \cdot 8 + 11$

$185 = 16 \cdot 11 + 9$

Exercise 2

(i) $(236)_{10}$

$= (001000110110)_{BCD}$

$(61)_{10}$

$= (01100001)_{BCD}$

$(158)_{10}$

$= (000101011000)_{BCD}$

(ii) $\underbrace{0011}_{3}\ \underbrace{1001}_{9}$ ✓

$\underbrace{0110}_{6}\ \underbrace{1010}_{10}$ ✗

$\underbrace{1110}_{14}$ 0011 ×

(iii) too many bits

(b)

0110
1110 ✓
0101 ×

(c)

$236 = 118 \cdot 2 + 0$
$118 = 59 \cdot 2 + 0$
$59 = 29 \cdot 2 + 1$
$29 = 14 \cdot 2 + 1$
$14 = 7 \cdot 2 + 0$
$7 = 3 \cdot 2 + 1$
$3 = 1 \cdot 2 + 1$
$1 = 0 \cdot 2 + 1$

$(11101100)_2$

11101100

$(10011010)_{Gray}$

$61 = 16 \cdot 3 + 13$

$(00111101)_2$

$(00100011)_{Gray}$

$158 = 16 \cdot 9 + 14$

$(10011110)_2$

$= (11010001)_{Gray}$

Exercise 3

ⓐ Signed $[-2^{n-1}-1, 2^{n-1}]$

Unsigned $[0; 2^n-1]$

Signed is better if app does not need neg. integers.

ⓑ Signed: 0 → max positive → max negative → 0
Unsigned: 0 → max positive → 0

ⓒ

Decimal	Sign and Magn.	2's compl.
-32	10100000	11100000
73	01001001	01001001
-98	11100010	10011110
47	00101111	00101111
-39	10100111	11011001
86	01010110	01010110

d

4-bit SM	8-bit SM	Hex	Dec	8-bit 2's compl.
0110	0000 0110	6	6	0000 0110
1100	1000 0100	84	-4	1111 1100
1111	1000 0111	87	-7	1111 1001
0001	0000 0001	1	1	0000 0001

4-bit 2's	8-bit 2's	Hex	Dec	8-bit SM
1010	1111 1010	FA	-6	1000 0110
0101	0000 0101	5	5	0000 0101
1100	1111 1100	FC	-4	1000 0100
1000	1111 1000	F8	-8	1000 1000

(e) i) 2's complement : non-redundant

ii) 2's complement : addition is the same as subtraction

iii) 2's complement : better numerical range + different overflows

Exercise 4

ⓐ Unsigned

A	Bin(A)	A >> 1	A >> 3	A << 1	A << 3
39	00100111	00010011	00000100	01001110	00111000
192	11000000	01100000	00011000	10000000	00000000
75	01001011	00100101	00001001	10010110	01011000

Signed

A	Bin(A)	A >> 1	A >> 3	A << 1	A << 3
39	00100111	00010011	00000100	01001110	00111000
-38	11011010	11101101	11111011	10110100	11010000
49	00110001	00011000	00000110	01100010	10001000
-106	10010110	11001011	11110010	00101100	10110000

$-38 = -64 + 26$ | $-106 = -128 + 22$

(b) 16 nb de 8 bits
8 nb de 9 bits
4 nb de 10 bits
2 nb de 11 bits
1 nb de 12 bits

16 nb de 8 bits
8 nb de 16 bits
4 nb de 32 bits
2 nb de 64 bits
1 nb de 128 bits

(c) $(6+7)_{10} = (0110 + 0111)_2$

$$
\begin{array}{r}
{\scriptstyle 1\ \ 1\ \ \ \ \ \ \ } \\
0\,1\,1\,0 \\
+\ 0\,1\,1\,1 \\
\hline
1\,1\,0\,1
\end{array}
= (-3)_{10} \quad ⚠
$$

$$(3-8)_{10} = (0011 + 1000)_2$$

$$= (1011)_2 \quad -5$$

$$(-2-10)_{10} = (1110 + 1000)_2$$

$$= (0110)_2 \quad 6 \quad ⚠$$

$$(5-3)_{10} = (0101 + 1101)_2$$

$$= (0010)_2 \quad 2$$

$$(1+5)_{10} = (0001 + 0101)_2$$

$$= (0110)_2 \quad 6$$

(d) $(12)_{10} = (1100)_2$

$(6)_{10} = (0110)_2$

$$
\begin{array}{r}
1100 \\
\cdot\ 0110 \\
\hline
0000 \\
+\ 111000 \\
+\ 110000 \\
\hline
1001000
\end{array}
$$

(e) (i) remove the sign, multiply the magnitude, assign the sign with a XOR(s_1, s_2)

(ii) Convert to SM → multiply → convert back to 2's compl.

Exercise 5

(a) Count the number of 1s and the number of diff. positions.

(b) [i] 7 × (1s)
3 × (0s)

[ii] $\binom{10}{7}$

(c) minimum distance: 2
max distance: 10

(d) 1

(e) 110
001
⇒ HD 3

001
101
⇒ HD 1

8 $\boxed{i}$ 2^n different strings

$(2^n)^2$ couples s.t. (u, v)

we want $u \neq v$ then

$$(2^{2n}) - \underbrace{(2^n)}_{\text{nb of } (u,u)}$$

$\boxed{ii}$ from 0 to n, $(n+1)$ distinct distances.

Exercise 6

(a) $\text{Bias} = 2^{m-1} - 1 = 2^3 - 1$
$= 7$

$E = (0100)_2 = (4)_{10} = (-3)_{10}$

$\left(1 + \frac{1}{2^2}\right) \cdot 2^{-3}$

(b) $(-1) \cdot \left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{16}\right) \cdot 2^5$

(c) $(1) \cdot \left(1 + \frac{1}{2} + \frac{1}{4}\right) \cdot 2^8$

(d) $(1) \cdot \left(1 + \frac{1}{16}\right) \cdot 2^6$

b $\boxed{i}$

$$(2.5)_{10} = (10.1)_2$$

$$= (.01)_2 \cdot 2^1$$

$1 = E - B$

$\Leftrightarrow E = 1 + B$

$= 128 = 1000\ 0000$

$M = 0100 \ldots 000$

$\boxed{ii}$ $(3.75)_{10} = (11.110)_2$

$(.1110) \cdot 2$

$E = 128$

$M = 11100 \ldots$

$\boxed{iii}$ $(-4.25)_{10} = (-100.01)_2$

$(.0001)_2$

$E = 129 \quad S = 1$

$M = 0001$

$\boxed{iv}$ $(-5)_{10} = (-101.00)_2$

$(.0100)_2$

$E = 129 \quad S = 1$

$M = 0100$

$\boxed{v}$ $(6.25)_{10} = (110.01)_2$

$E = 129 \quad S = 0$

$M = 1001$

(c) $(0.75)_{10} = (0.11)_2$

$$E_{single} = -1 + B_{single}$$
$$= -1 + 127$$
$$= 126$$

$$E_{double} = -1 + B_{double}$$
$$= -1 + (2^{10} - 1)$$
$$= 1022$$

Exercise 7

(a) (i) $(1010.1101)_2$

$$= 10 + 0.75 + 0.0625$$
$$= 10.8125$$

(ii) $(10001010.1101)_2$

$$= -128 + 10.8125$$

$= -118 + 0.8125$

$= -117.1875$

iii $(10101001.0001)_2$

$= -128 + 32 + 8 + 1 + 0.0625$

$= -86.9375$

iv $(01010101.1001)_2$

$= 4 + 1 + 16 + 64 + 0.5 + 0.0625$

$= 85.5625$

v $(01011010.1111)_2$

$= 2 + 8 + 16 + 64 + 0.5 + 0.250 + 0.125$

$= 90.875 + 0.0625$ $+ 0.0625$

$= 90.9375$

(b) $\boxed{\text{i}}$ most positive number:

$$2^{n_1} + (1 - 2^{-n_2})$$

$\boxed{\text{ii}}$ 2^{-n_2}

$\boxed{\text{iii}}$ $-2^{n_1 - 1}$

$\boxed{\text{iv}}$ -2^{-n_2}

(c) $(1.5)_{10} = (0001.1000)_2$

$(5.25)_{10} = (1001.0100)_2$

$(-0.125)_{10} = (1111.1110)_2$

$(-3.0)_{10} = (1101.0000)_2$

$(-6.75)_{10} = (1001.0100)_2$

d) i) $(01101.0010)_2$

$E = 3+127 = 130$
$M = 1010010$
$S = 0$

ii) $(11010.1101)_2$

$E = 130$
$S = 1$
$M = 0101101$

iii) $(01111.1111)_2$

$E = 130$
$S = 0$
$M = 1111...$

iv) $(11001.1001)_2 \Rightarrow E = 130$
$S = 1$
$M = 0011001...$

Exercise 8

(a) $\boxed{i}$ max value :

$$E_{max} = (2^8 - 1) - B$$
$$= 2^8 - 2^7$$
$$= 2^7$$

Max mantissa : 111.....

$$= \left(1 + \frac{1}{2} + \frac{1}{4} + \dots\right)$$
$$= 1 \cdot \frac{1-\left(\frac{1}{2}\right)^{24}}{1-\frac{1}{2}}$$
$$= 2\left(1-2^{-24}\right)$$

Value $= 2^{128}\left(2-2^{-23}\right)$

Range $2 \cdot$ Value (symmetric because of SM)

(II) Same calculation with 53 instead of 24.

(ii) resolution: diff min non-zero value repres. and zero

$E_{min} = 0 - 127$.

$M_{min} = (1 + 2^{-23})$ $\quad\quad M_{zero} = 1$

$$V_{min} = (1+2^{-23}) \cdot 2^{-127} - 2^{-127}$$

$$= 2^{-23} \cdot 2^{-127} = \underline{2^{-150}}.$$

$$\begin{array}{r} 0.5 \\ +0.250 \\ +0.125 \\ +0.0625 \\ +0.03125 \end{array}$$

(b) $(2.7)_{10} = (10.1011)_2$

$$E = 1 + (2^{2-1} - 1)$$
$$= 2$$

$(3.9)_{10} = (11.11100)_2$

$E = 1 + 1$ 1111

$= 2$

$(-0.52) = (0.10000)_2$

$M = 0000..$

$E = -1 + 1 = 0$

$(-0.67) = (0.10101)_2$

$M = 0110$

$E = 0$

$$\begin{array}{r} 5 \\ +125 \\ +00 \end{array} \qquad \begin{array}{r} {}^{1} \\ 250 \\ +125 \\ +0325 \\ \hline 4075 \end{array}$$

(c) $(7.2)_{10} = (0111.0011)_2$

$(-6.42)_{10} = -(0110.01101)_2$

$= -(0110.0111)_2$

$= (1001.1001)_2$

$(-3.67)_{10} = -(0011.10101)_2$

$= -(0011.1011)_2$

$= (1100.0101)_2$

$$\begin{array}{r} 250 \\ +0625 \\ \hline 3125 \end{array}$$

$(5.33)_{10} = (0101.01010)_2$

$= (0101.0101)_2$

(d) Resolution $= 2^{-23}$

Range $= (2^8-1) + (1-2^{-23})$

Exercice 9

3125

$$\begin{array}{r} 0.5 \\ + 0.0 \\ .0 \end{array}$$

ⓐ $(14.51)_{10} = (1110.10000)_2$

0.625
0.3125
0.15625

$B = 2^4 - 1 = 15$

$E = 3 + 15 = 18$

$M = 11010000 1$
$S = 0$

$(-21.75)_{10} = (10101.11000)_{10}$

$E = 4 + 15 = 19$
$S = 1$
$M = 0101 11000$

$$\begin{array}{r} 250 \\ + 125 \\ + 0625 \\ \hline 4375 \end{array}$$

$(7.45) = (111.01110 ..)_2$

$M = 110110011$

$E = 2+15 = 17$

$S = 0$

$$\begin{array}{r} 0.5 \\ +0.125 \\ +0.0625 \\ 6875 \\ 3125 \\ \hline 7.1875 \\ +\ 1562 \\ \hline 7.3437 \end{array}$$

$(-2.725) = (10.101100)_2$

+ many more 1s (so it gives 111 when you add 1 to the last bit)

$E = 16$

$S = 1$

$M = 0101110$

$$\boxed{\text{We take the largest exponent } 1011}$$

14.51 $M = 01110000\ldots$

(b) (i) takes less space, better range better resolution

(ii) takes less space

Exercise 10

$$(0.125)_{10} = (0.001)_2$$

$$E = -3$$
$$n = 0000..$$

$$0 \quad \underbrace{01111111}_{E=0} \quad 0000....$$

$$\Rightarrow (0.125)_{10} = (0 \quad 01111100 \quad 000...)$$

$$(0.25)_{10} = (0.0100)_2$$

$$\Rightarrow (0 \quad 01111101 \quad 000...)$$

We take the largest exponent:

0.25	0	01111101	1.00000
0.125	0	01111101	0.10000
	0	01111101	1.10000

280
+ 125

$(-0.375)_{10} = (0.011)_2$

$(0.5)_{10} = (0.1)_2$

0.5	0	01111110	1.0000...
-0.375	1	01111110	0.1100...
Sum	0	01111110	1.11000...
→	0	0111110	0.010000...
→	0	0111100	1.00000...

$(0.625)_{10} = (0.101)_2$
$(0.75)_{10} = (0.110)_2$

0.625	0	01111110	1.0100000
0.75	0	01111110	1.1000000
Sum	0	01111110	10.0110000
	0	01111111	011000

$$
\begin{array}{r}
250 \\
+125 \\
\hline
875
\end{array}
$$

$(0.875)_{10} = (0.111)_2$
$(1.0)_{10} = (1.0)_2$

1	1	0111 1111	1.00000
0.875	1	0111 1111	0.11100
Sum	1	0111 1111	1.111000...

$(1.125) = (1.001)_2$
$(-1.25) = (1.010)_2$

1.125	0	0111 1111	1.0010000
-1.25	1	0111 1111	1.0100000

0111 1111 1.111

$\Rightarrow$ 1 0111 1111 0.001000

$\Rightarrow$ 1 0111 1100 00000

(b)

$(1.125)_{10} = (1.001)_2$

$(3.25)_{10} = (11.010)_2$

$$\begin{array}{r} {}^{1}11.010 \\ +\ 1.001 \\ \hline 100.011 \end{array}$$

$$\begin{array}{r} 250 \\ +125 \\ \hline 375 \end{array}$$

$(4.375)_{10} = (0100.0110)_2$

$(-4.375)_{10} = (1011.1010)_2$

$(6.5)_{10} = (110.100)_2$

$$\begin{array}{r} 1011.1010 \\ +\ 0110.1000 \\ \hline 1\,0010.0010 \end{array}$$

$$(2.625)_{10} = (10.101)_2$$

$$(-7.75)_{10} = -(0111.1100)_2 = (1000.0100)_2$$

$$(-0.875)_{10} = -(0.1110)_2 = (1111.0010)_2$$

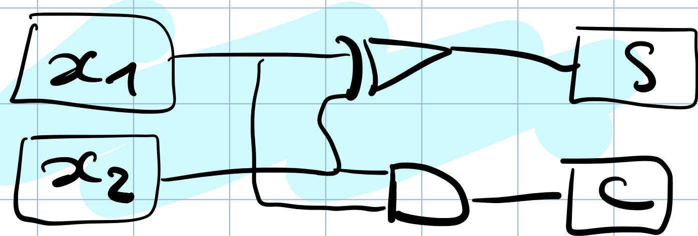
$$(-1.125)_{10} = -(1.0010)_2 = (1110.1110)_2$$

$$(6.25)_{10} = (110.0100)_2$$

$$(-3.875) = -(11.1110)_2 = (1100.0010)_2$$

(half-adder)

x_1	x_2	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



(full-adder)

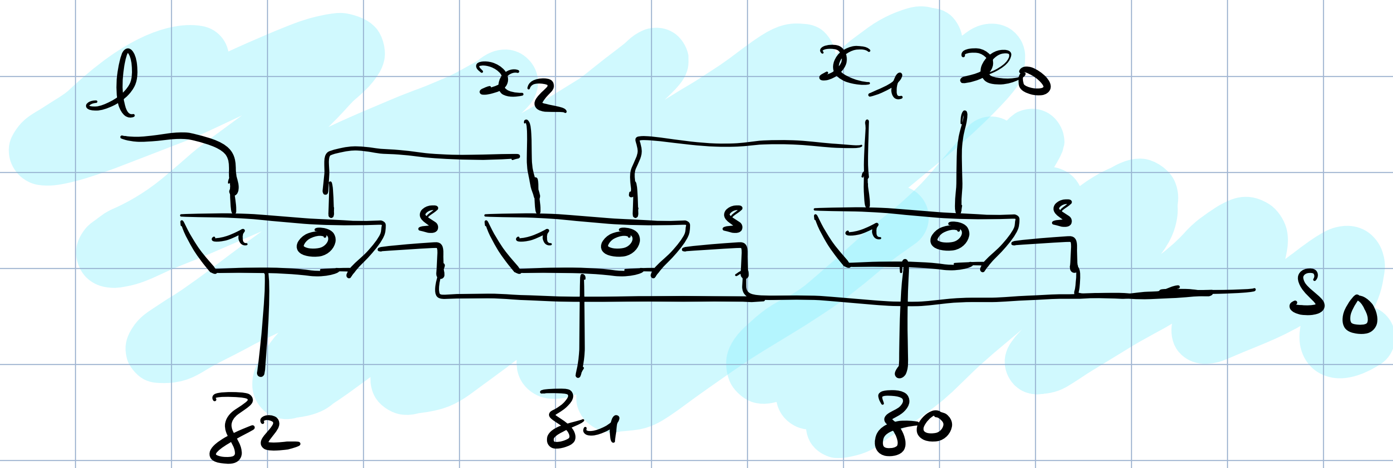
x_1	x_2	c_i	S_o	c_o
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$(s_t, c_1) = HA(x_1, x_2)$
$(s_o, c_2) = HA(s_t, c_i)$

$c_o = c_1$ or c_2

S	OP
0	$x+y$
1	$x+\bar{y}+1$

$\Rightarrow x + (y \text{ XOR } S) + S$


l
x2
x1
x0
1
0
s
s
s
s0
z2
z1
z0

Exercise 1

(a) 2^n

(b) Yes

(c) $\overline{x_1 x_2} = \overline{x_1} + \overline{x_2}$

(d)
$$p_2 = x_1 x_2$$

$$p_1 = (\overline{x_1 x_2})(x_1 + x_2)$$

$$p_3 = p_1 x_3$$

$$p_4 = p_1 + x_3$$

$$f = p_2 + p_3$$
$$g = \overline{p_3}\, p_4$$

Exercise 2

(a) i $\bar{c}(\bar{a}+b)$

ii $a+(\bar{b}c)$

iii $a(\bar{b}+c)+(b)$

(b) $f = abc$

$f = ac+b$

$f = a+\bar{b}\bar{c}$

(c) $a+bc = a+bc$ ✓

$(a+b)(a+\bar{b}) = a$ ✓

Exercise 3

(a) $A = x_3(x_1 + \overline{x_2}) = x_3x_1 + x_3\overline{x_2}$

$B = x_1(\overline{x_2} + \overline{x_3}) = x_1\overline{x_2} + x_1\overline{x_3}$

$C = x_2(x_1 + \overline{x_3}) = x_2x_1 + x_2\overline{x_3}$

$AB + BC + AC$

$(x_3x_1 + x_3\overline{x_2})(x_1\overline{x_2} + x_1\overline{x_3})$

$= x_3x_1\overline{x_2} = AB$

$BC = (x_1\overline{x_2} + x_1\overline{x_3})(x_2x_1 + x_2\overline{x_3})$

$= x_1x_2\overline{x_3}$

$AC = (x_3x_1 + x_3\overline{x_2})(x_2x_1 + x_2\overline{x_3})$

$= x_3x_2x_1$

$$x_1\overline{x_2}x_3 + \overbrace{x_1x_2\overline{x_3} + x_1x_2x_3}^{x_1x_2}$$

$$= x_1\left(\overline{x_2}x_3 + x_2\overline{x_3} + x_2x_3\right)$$

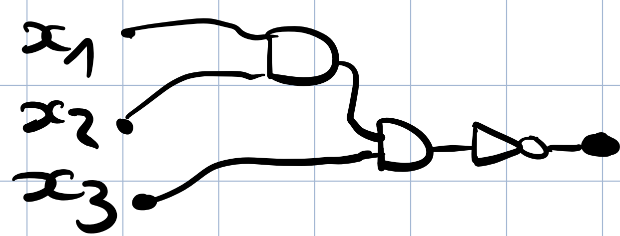
$$= x_1\left(x_2 + x_3\right)$$

b

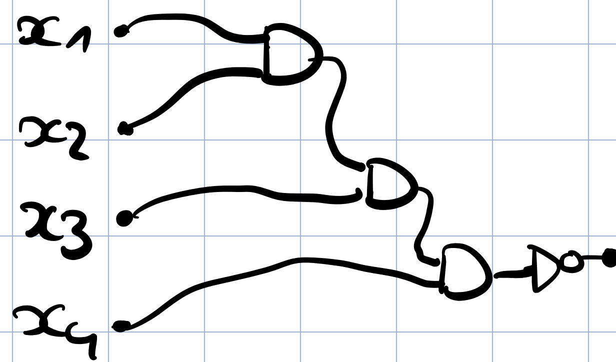
i

x_1	x_2	x_3	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

ii

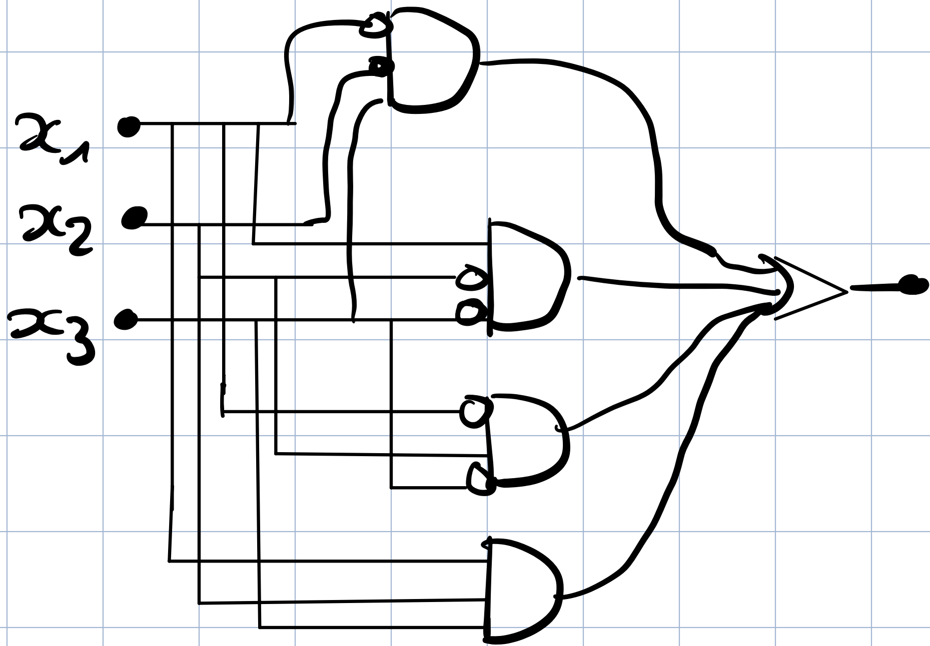


iii



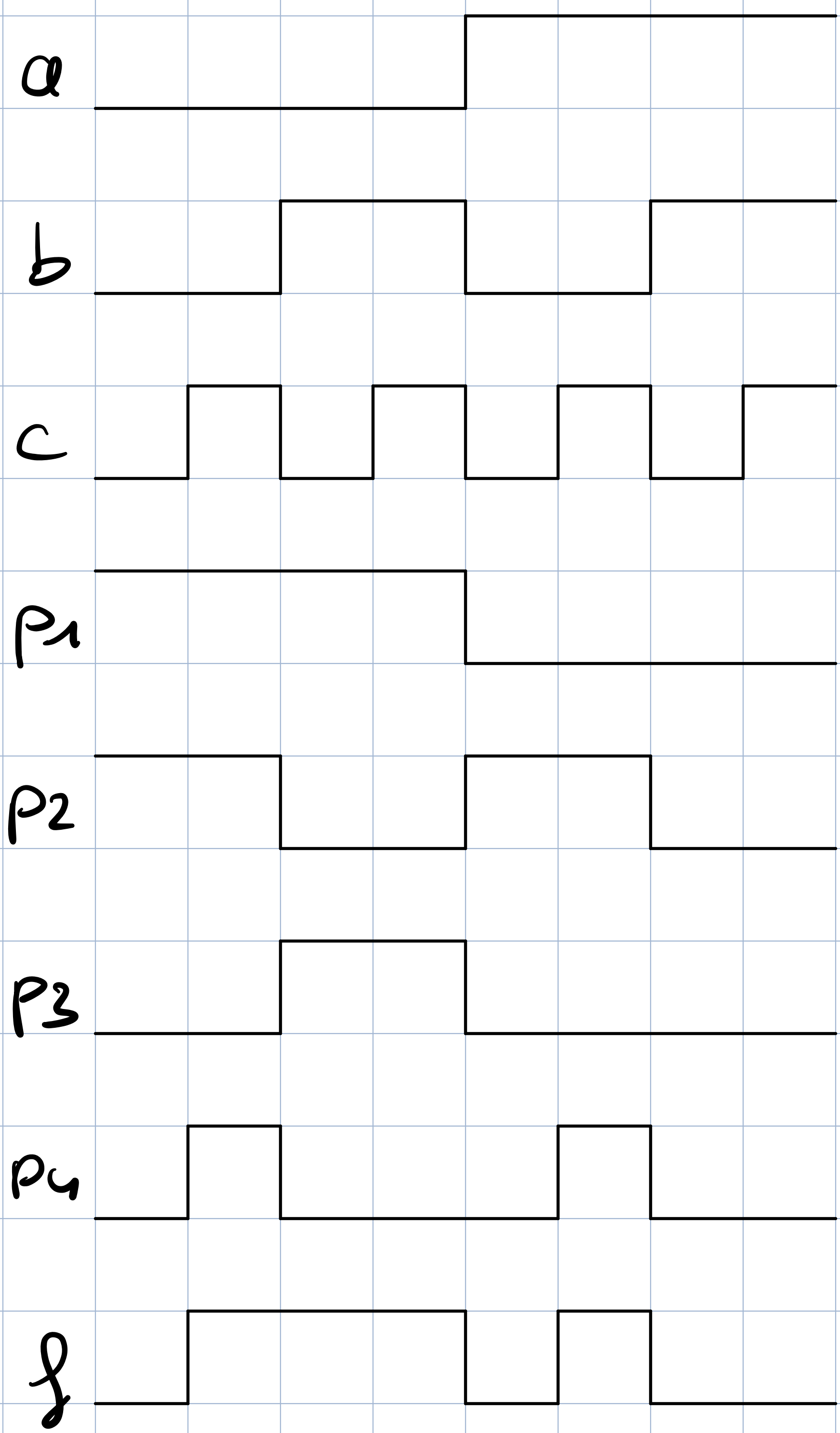
(c)

$\boxed{ii}$

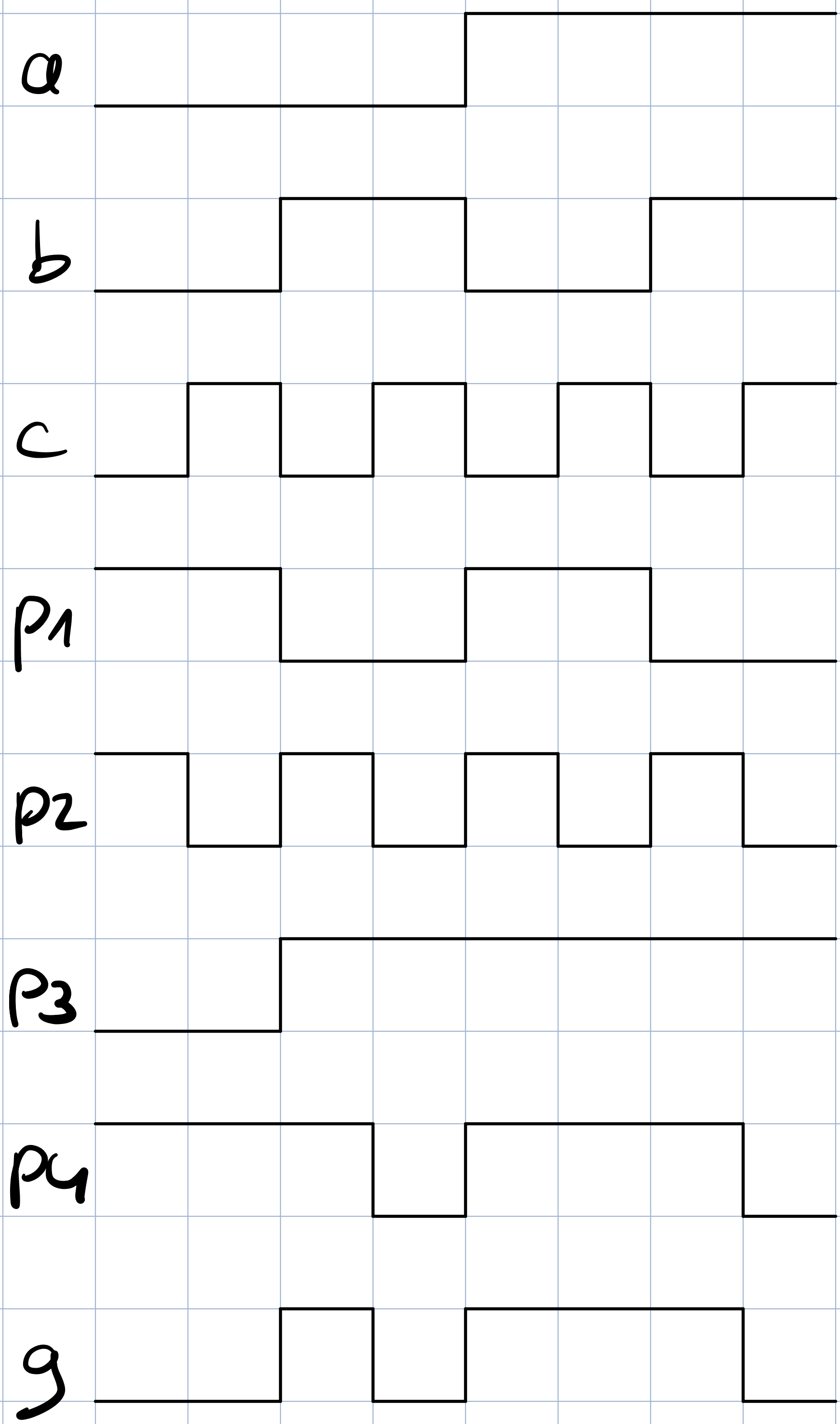


Exercise 4

(a)

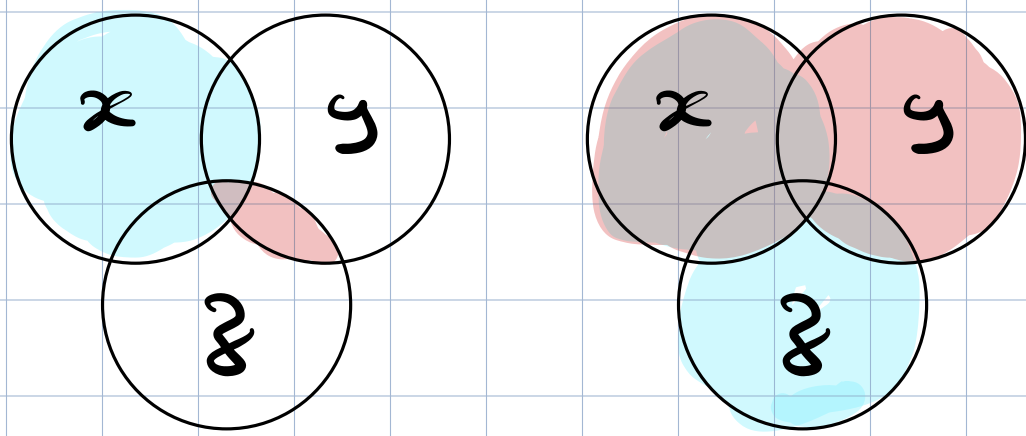


b

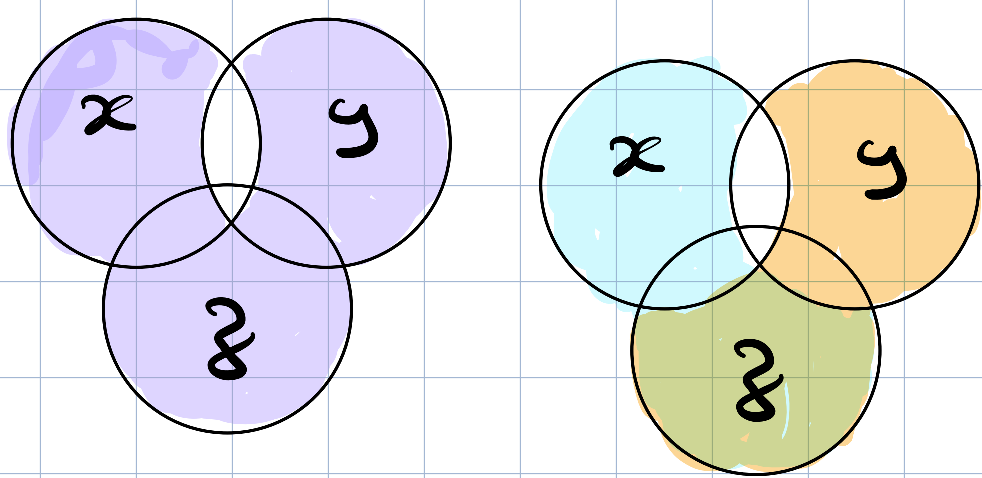


Exercise 5

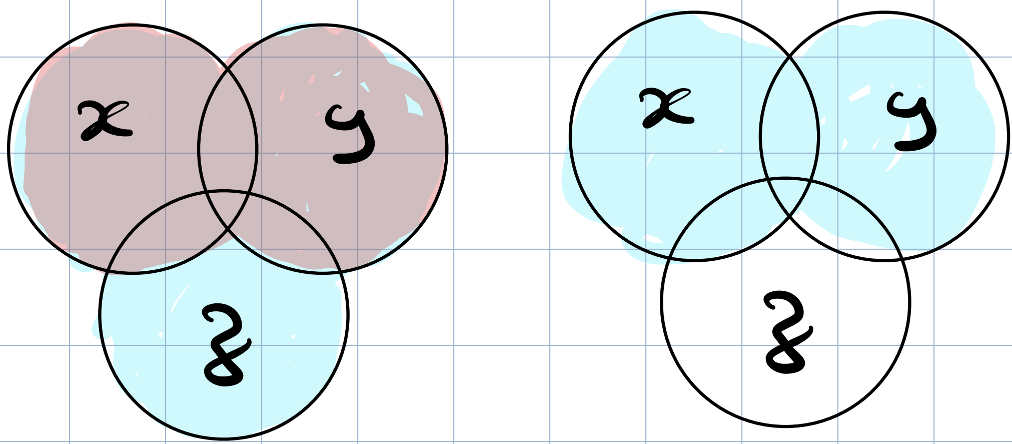
(a)



(b)



(c)



Exercise 7

(a) /

(b) /

(c) time for cin 1 computation:

$x_1 + x_2\ (S_t) : 3ns$

$x_1 + x_2\ (C_t) : 2ns$

$S + cin_0\ (S_0) : 3ns$

$S + cin_0\ (C_0) : 2ns$

$C_0 C_t : 2ns$

$3ns + 2ns + 2ns : \boxed{7ns}$

time for s_1 computation:

$x_1 + x_2\ (St) : 3ns$ ✓

time $cin_a \to cin_b : 2ns + 2ns = \boxed{4ns}$

$$7 + 4 \cdot (\text{ripple\_count} - 1)$$

$$= 7 \cdot 4 \cdot 3 = 12 + 7 = 19\,\text{ns}$$

$$7 + 4(7) = 28 + 7 = \underline{35}\,\text{ns}$$

Exercise 8

ⓐ $\boxed{i}$ $\bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + a\bar{b}c$

$+ ab\bar{c} = f$

$(a+b+\bar{c})(a+\bar{b}+\bar{c})(\bar{a}+b+c)$
$(\bar{a}+\bar{b}+\bar{c}) = f$

$\boxed{ii}$ $\bar{a}\bar{b}c + ab\bar{c} + abc = f$

$(a+b+c)(a+\bar{b}+c)(a+\bar{b}+\bar{c})$
$(\bar{a}+b+c)(\bar{a}+b+\bar{c}) = f$

ⓑ $\boxed{i}$ $f = \underset{m_1}{\bar{a}\bar{b}c} + \underset{m_3}{\bar{a}bc} + \underset{m_2}{\bar{a}b\bar{c}} + \underset{m_7}{abc}$

$$f = \Pi_0 + \Pi_4 + \Pi_5 + \Pi_6$$

$$= (a+b+c)(\bar{a}+b+c)(\bar{a}+b+\bar{c})(\bar{a}+\bar{b}+c)$$

$\boxed{ii}$

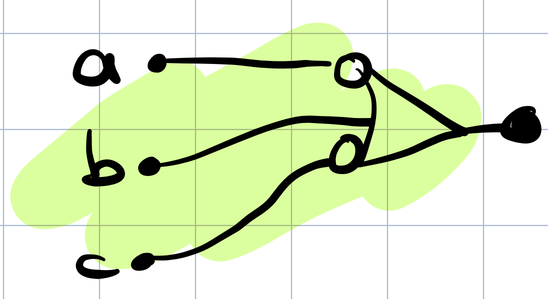
$$f = \underbrace{\bar{a}bc}_{m_3} + \underbrace{\bar{a}b\bar{c}}_{m_2} + \underbrace{\bar{a}\bar{b}c}_{m_1} + \underbrace{\bar{a}\bar{b}\bar{c}}_{m_0}$$

$$+ \underbrace{abc}_{m_7} + \underbrace{ab\bar{c}}_{m_6} + \underbrace{a\bar{b}\bar{c}}_{m_4}$$

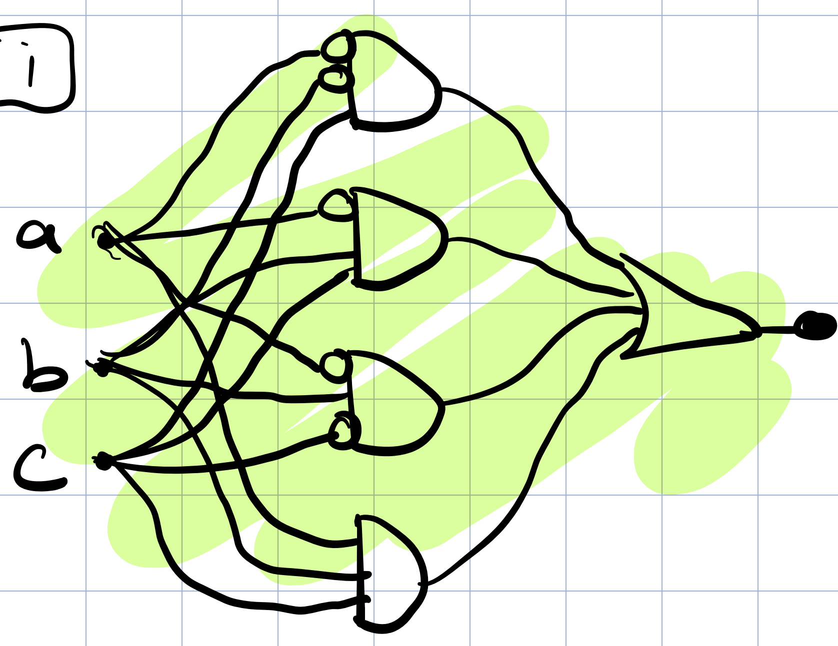
$$f = (\bar{a}+b+\bar{c})$$

c b

$\boxed{ii}$



$\boxed{i}$



Exercise 9

(a)

$$\begin{aligned} &\bar{a}b\bar{c} + \bar{a}\bar{b}\bar{c} && m_2 \quad m_0 \\ &+ abc + \bar{a}bc && m_7 \quad m_3 \\ &+ a\bar{b}c + a\bar{b}\bar{c} && m_5 \quad m_4 \end{aligned}$$

✓

$$\begin{aligned} &\bar{a}b\bar{c} + \bar{a}bc && m_2 \quad m_3 \\ &+ abc + a\bar{b}c && m_7 \quad m_5 \\ &+ a\bar{b}\bar{c} + \bar{a}\bar{b}\bar{c} && m_4 \quad m_0 \end{aligned}$$

(b)

$$\begin{aligned} &\bar{a}bc + \bar{a}\bar{b}c && m_3 \quad m_1 \\ &+ ab\bar{c} && m_6 \\ &+ \bar{a}b\bar{c} + \bar{a}bc && m_2 \quad m_3 \\ &+ a\bar{b}c + a\bar{b}\bar{c} && m_5 \quad m_4 \end{aligned}$$

✓

$$\begin{aligned} &a\bar{b}c + \bar{a}\bar{b}c && m_5 \quad m_1 \\ &+ ab\bar{c} + a\bar{b}\bar{c} && m_6 \quad m_4 \\ &+ ab\bar{c} + \bar{a}b\bar{c} && m_6 \quad m_2 \\ &+ \bar{a}bc && m_3 \end{aligned}$$

(c) $a b \bar{c} + a \bar{b} \bar{c}$ $\quad m_6 \quad m_4$

$+ a b c + \bar{a} b c$ $\quad m_7 \quad m_3$

$+ a \bar{b} \bar{c} + \bar{a} \bar{b} \bar{c}$ $\quad m_4 \quad m_0$

$M_2 + M_1 + M_5$ ✓

(d) $(a+c)(\bar{a}+\bar{b}+c)(\bar{a}+b)$

$= (a+\bar{b}+c)(a+b+c)(\bar{a}+\bar{b}+\bar{c})(\bar{a}+b+c)(\bar{a}+b+\bar{c})$

$M_2 \quad M_0 \quad M_7 \quad M_4 \quad M_5$

$(a+b)(b+c)(\bar{a}+\bar{c})$ ✗

$= (a+b+c)(a+b+\bar{c})(a+b+c)(\bar{a}+b+c)(\bar{a}+b+\bar{c})(\bar{a}+\bar{b}+\bar{c})$

$M_0 \quad M_1 \quad M_0 \quad M_4 \quad M_5 \quad M_7$

Exercise 10

ⓐ $\boxed{i}$

$$f = \bar a \bar b c + \bar a b \bar c + a \bar b \bar c + abc$$

$$f = \overline{\overline{\bar a \bar b c + \bar a b \bar c + a \bar b \bar c + abc}}$$

$$= \overline{(\overline{\bar a \bar b c})(\overline{\bar a b \bar c})(\overline{a \bar b \bar c})(\overline{abc})}$$

$\boxed{ii}$ $$f = (ab + cd) + ((a + \bar b)(\bar c + d))$$

$$= ab + cd + a\bar c + ad + \bar b \bar c + \bar b d$$

$$= \overline{(\overline{ab})(\overline{cd})(\overline{a\bar c})(\overline{ad})(\overline{\bar b \bar c})(\overline{\bar b d})}$$

$\boxed{iii}$ $$f = \overline{(\overline{\bar a bc})(\overline{a \bar b \bar c})(\overline{ab\bar c})(\overline{abc})}$$

(b) i)

$$f = (a+b+c)(a+\bar{b}+\bar{c})(\bar{a}+b+\bar{c})(\bar{a}+\bar{b}+c)$$

$$= \overline{\overline{(a+b+c)} + \overline{(a+\bar{b}+\bar{c})} + \overline{(\bar{a}+b+\bar{c})} + \overline{(\bar{a}+\bar{b}+c)}}$$

ii)

$$f = \left(\overline{a\bar{b}} + \overline{\overline{cd}}\right) + \left((a+\bar{b})(\bar{c}+d)\right)$$

$$= \overline{\overline{(\bar{a}+\bar{b}) + (\bar{c}+\bar{d}) + \overline{\overline{(a+\bar{b})} + \overline{(\bar{c}+d)}}}}$$

$$= (\bar{a}+\bar{b}) + (\bar{c}+\bar{d}) + \overline{\overline{(a+\bar{b})} + \overline{(\bar{c}+d)}}$$

iii)

$$f = \underbrace{\bar{a}bc}_{m_2} + \underbrace{a\bar{b}\bar{c}}_{m_4} + \underbrace{ab\bar{c}}_{m_6} + \underbrace{abc}_{m_7}$$

$$= \underbrace{(a+b+c)}_{M_0}\underbrace{(a+b+\bar{c})}_{M_1}\underbrace{(a+\bar{b}+\bar{c})}_{M_3}\underbrace{(\bar{a}+b+\bar{c})}_{M_5}$$

$$= \overline{\overline{(a+b+c)} + \overline{(a+b+\bar{c})}(a+\bar{b}+\bar{c})(\bar{a}+b+\bar{c})}$$

Exercise 12

(a)

	w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

$$a = (w+x+y+\bar{z})(w+\bar{x}+y+z)$$

$$= w + y + \bar{z}\bar{x} + xz$$

$$b = (w+\bar{x}+y+\bar{z})(w+\bar{x}+\bar{y}+z)$$

$$= w + \bar{x} + yz + \bar{z}\,\bar{y}$$

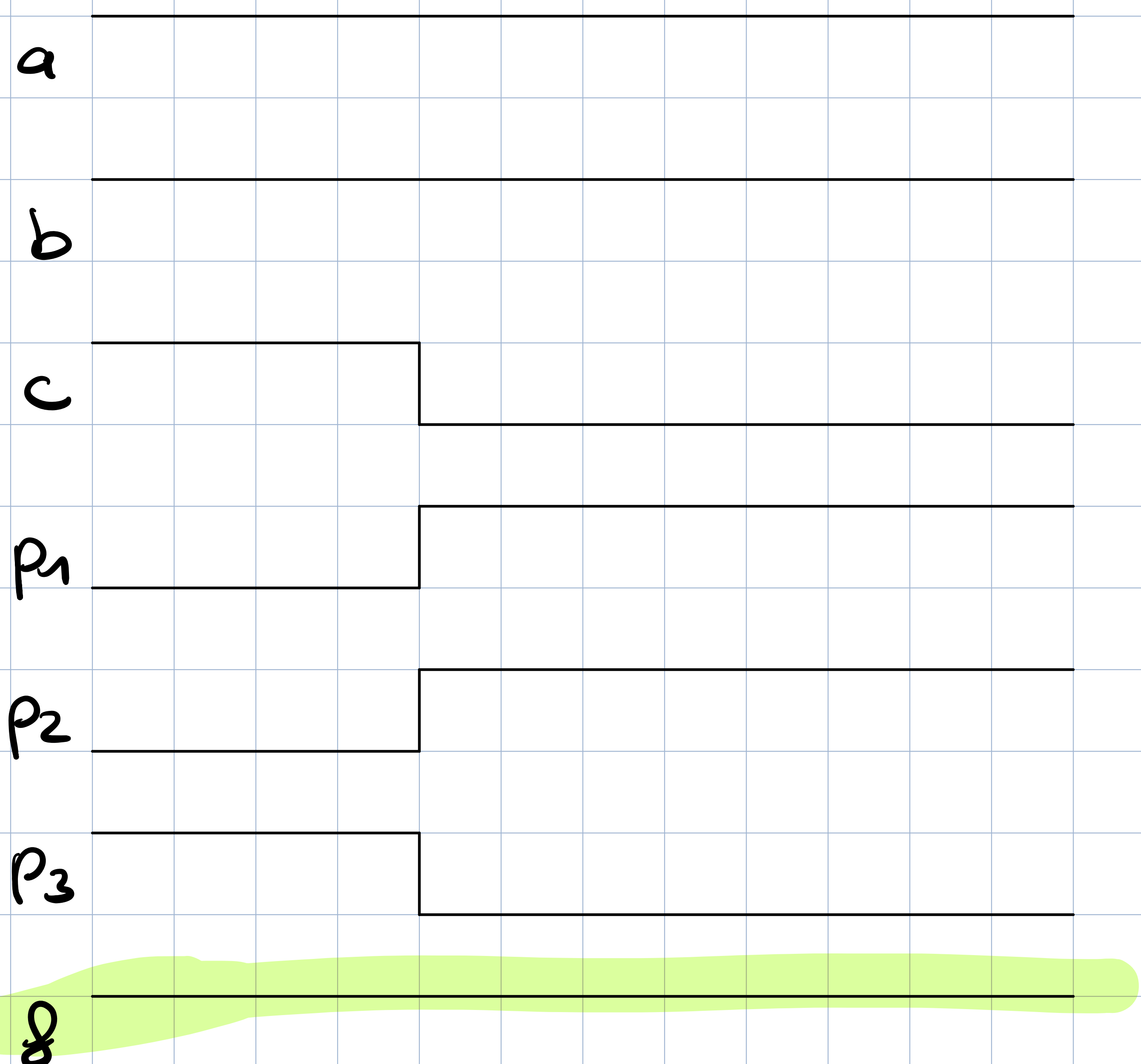
yz \ wx	10	11	01	00
10	0	0	1	1
11	0	0	1	1
01	1	0	1	0
00	1	0	0	1

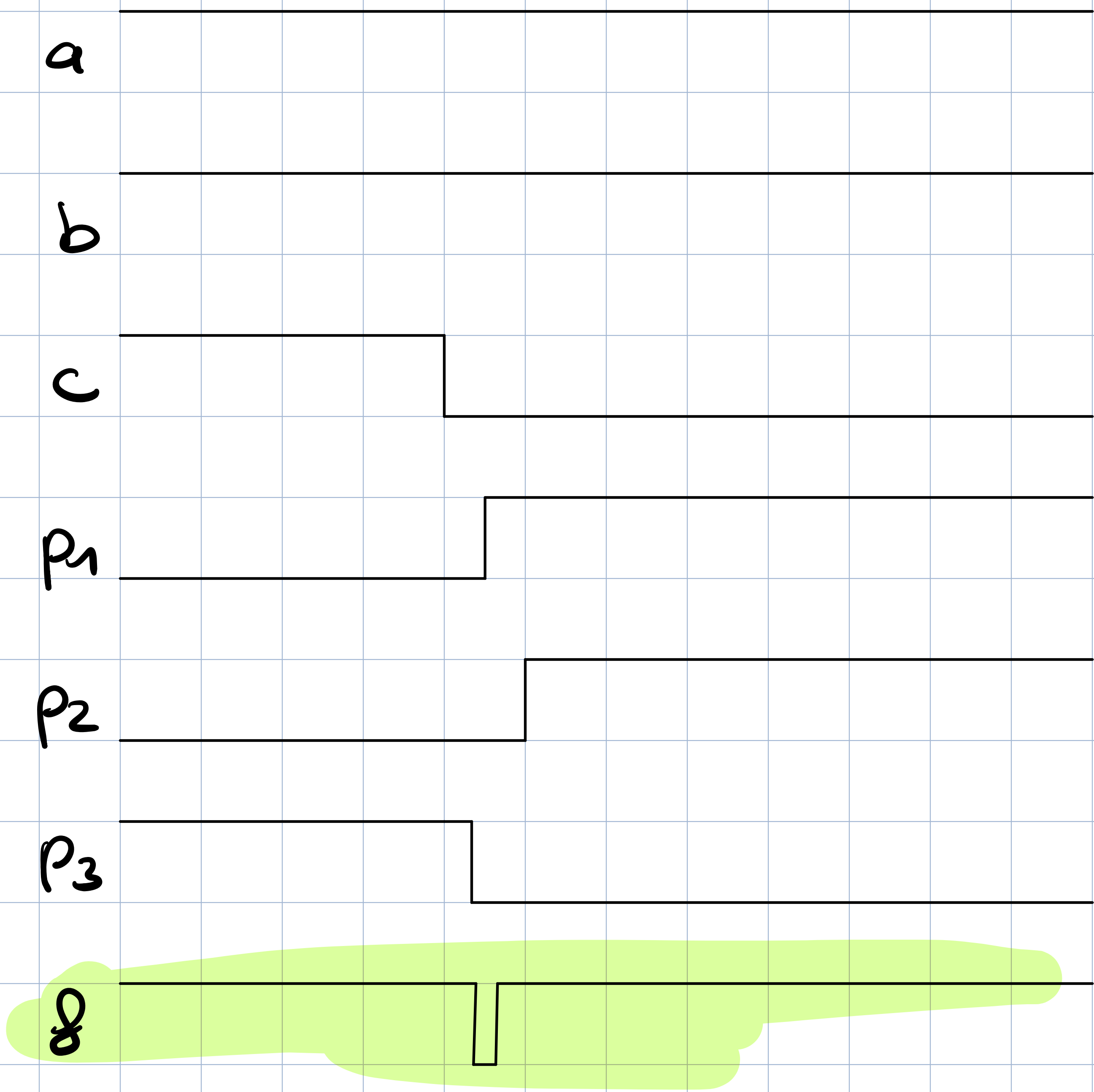
$$w\bar{x}\bar{y} + \bar{w}y + \bar{w}xz + \bar{w}\bar{x}\bar{z}$$

Exercise 13

(a) (i) $f = a\bar{c} + cb$

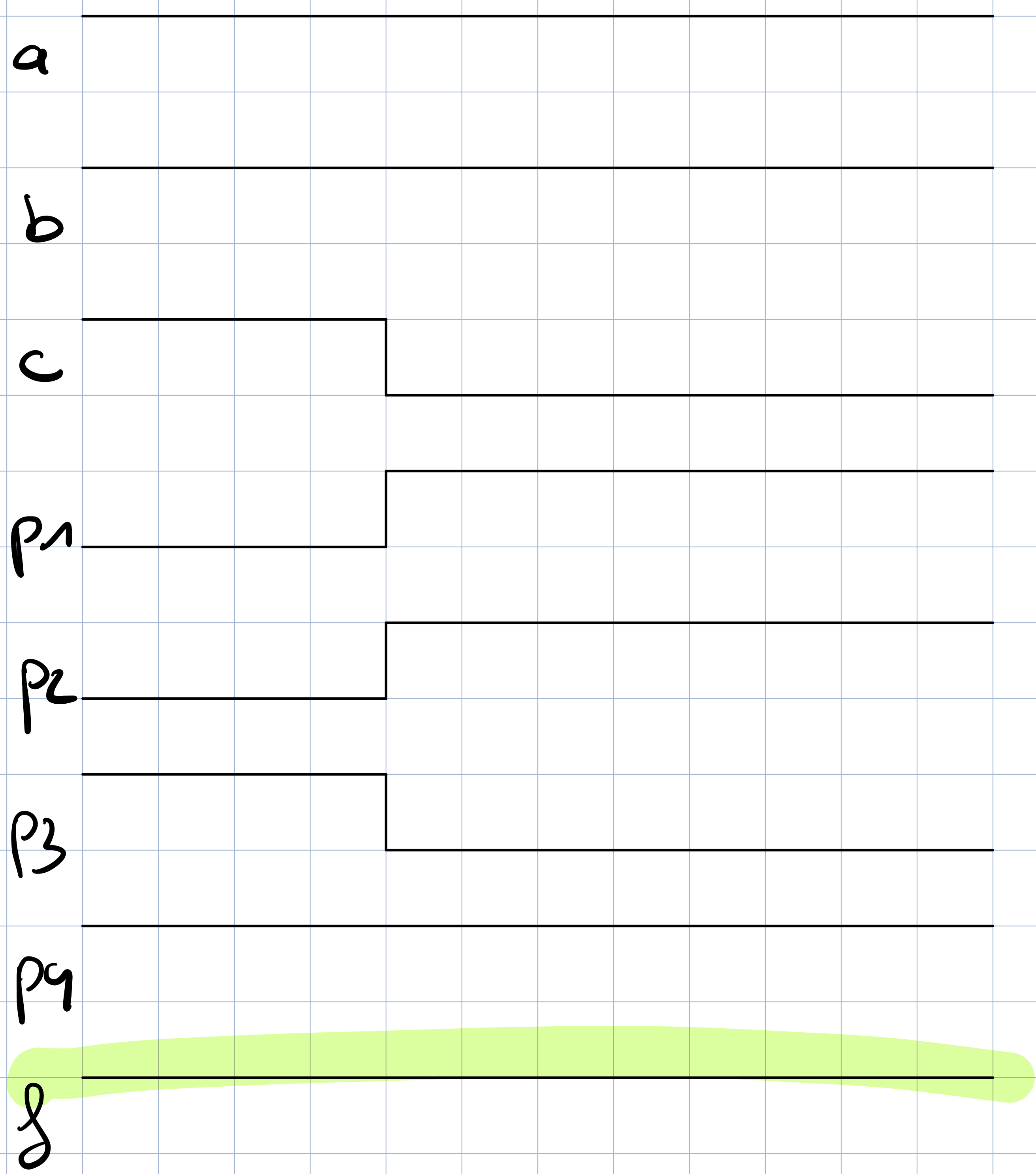
(ii)




a
b
c
p_1
p_2
p_3
g

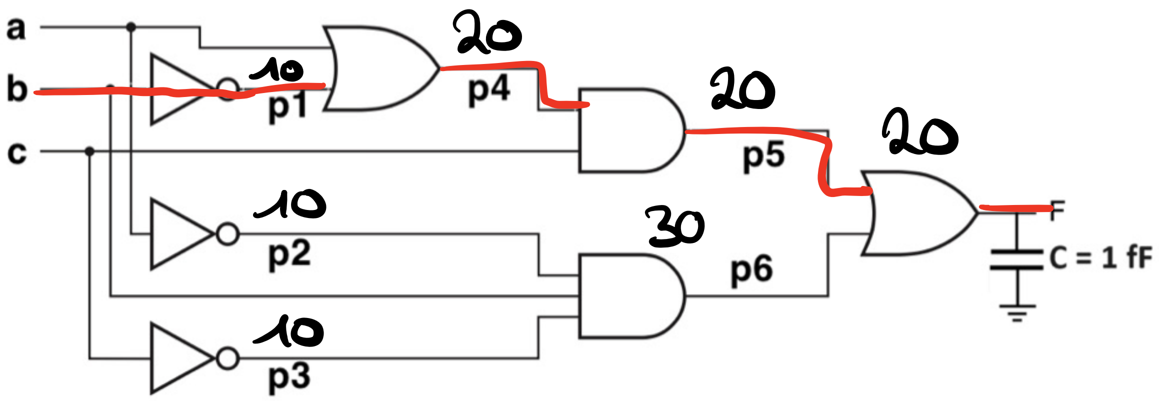
(b) (i)

$$f = a\bar{c} + bc + ab$$



Exercise 14

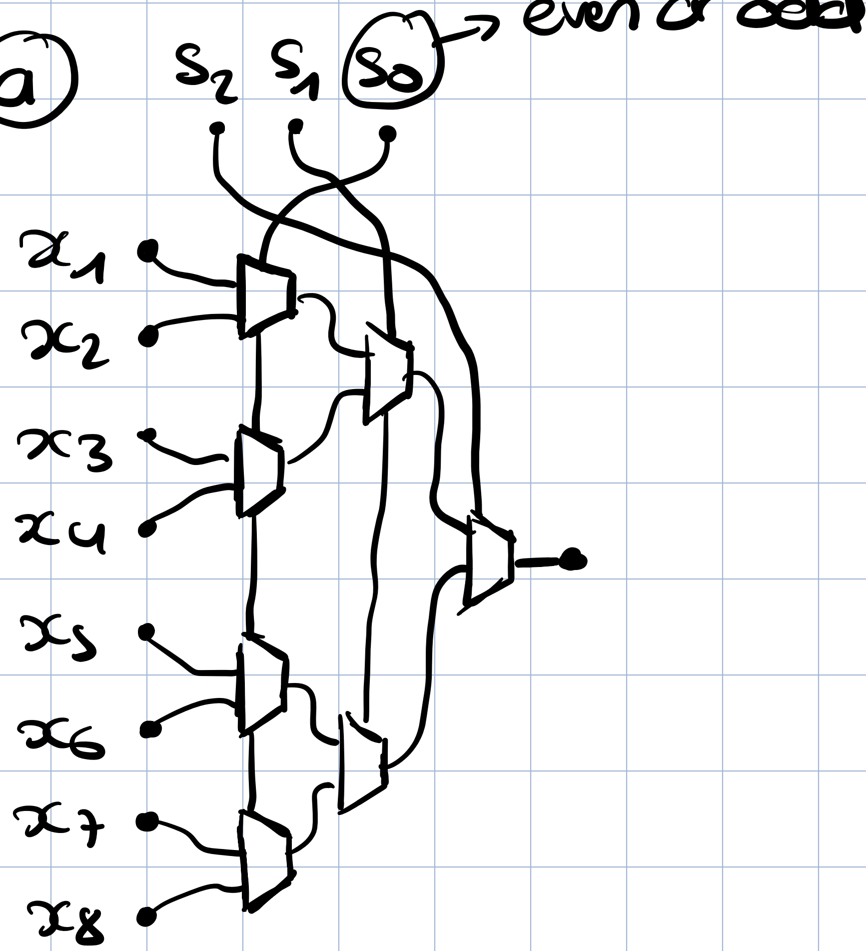
(a)



(b) $10 + 20 + 20 + 20 = \underline{70\,ns}$

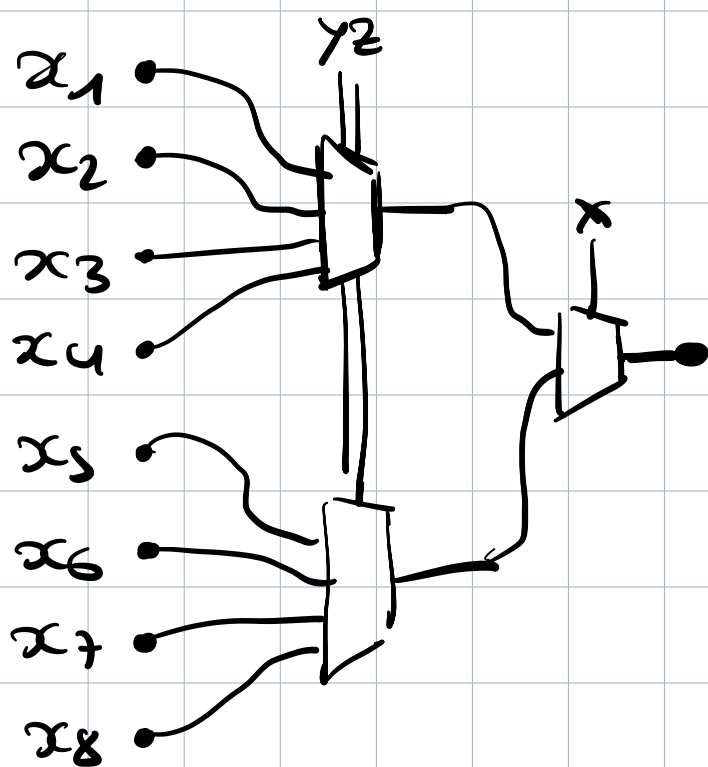
Exerzise 15

ⓐ



s_2
s_1
s_0
→ even or odd
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8

ⓑ



yz
x
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8

Exercice 16

$$fC(V_{DD})^2 = P_{DISS}$$

ⓐ $\begin{cases} f = 320 \cdot 10^6 \ Hz \\ V_{DD} = 1.05 V \\ C = 100 fF = 10^{-13} F \end{cases}$

$$P_1 = 320 \cdot 10^6 \cdot 100 \cdot 10^{-13} \cdot (1.05)^2$$

ⓑ $\nearrow$

ⓒ $\nearrow$

ⓓ $\nearrow$

Exercise 17

(a) delay p_1 : $60ns + 0ns$

delay p_2 : $20ns + 0ns$

delay p_3 : $40ns + 0ns$

delay p_4 : $30ns + \overbrace{max(p_1, p_2)}^{60ns}$

delay p_5 : $60ns + 0ns$

delay p_6 : $60ns + max(p_1, p_3)$

delay p_7 : $30ns + max(p_2)$

delay p_8 : $20ns + max(p_4, p_5, p_6)$

delay p_9 : $20ns + max(p_5, p_6, p_7)$

worst-case path : $a \rightarrow p_1 \rightarrow p_6 \rightarrow f$

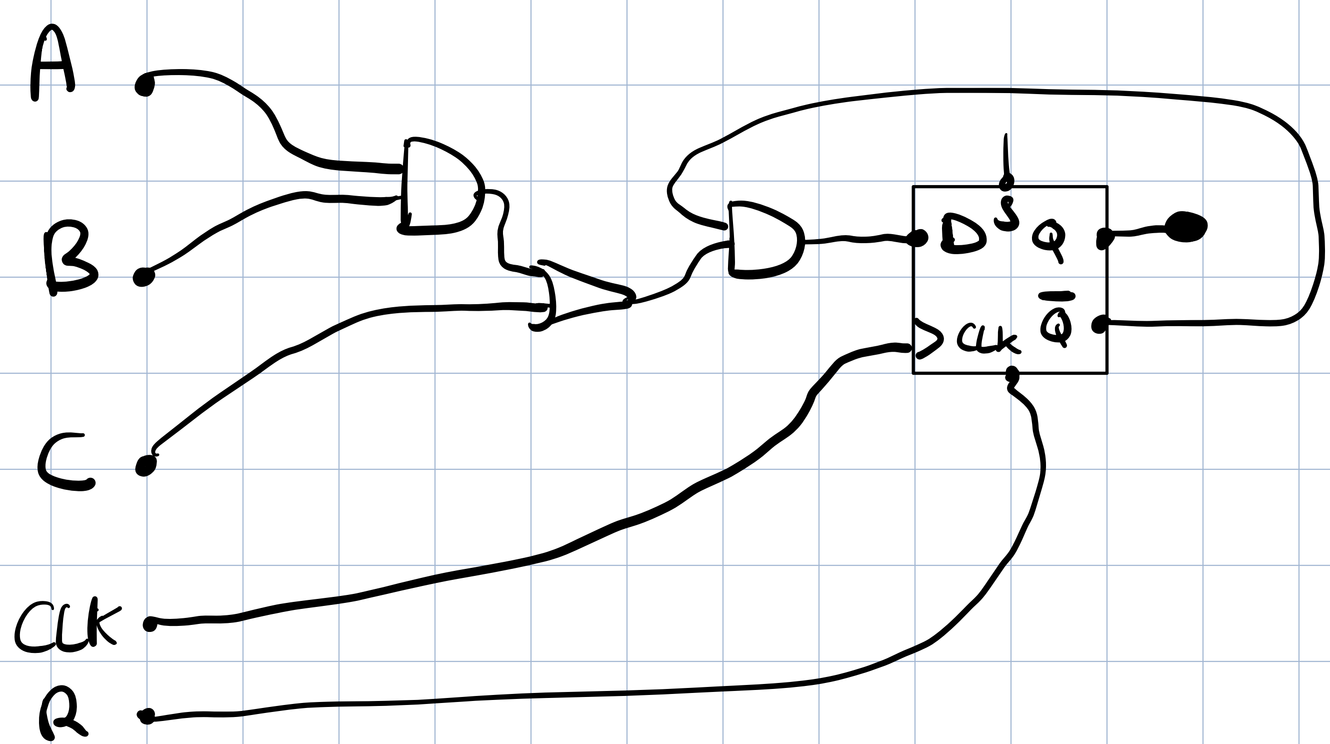
Exercise 18, 19, 20, 21, 22, 23, 24

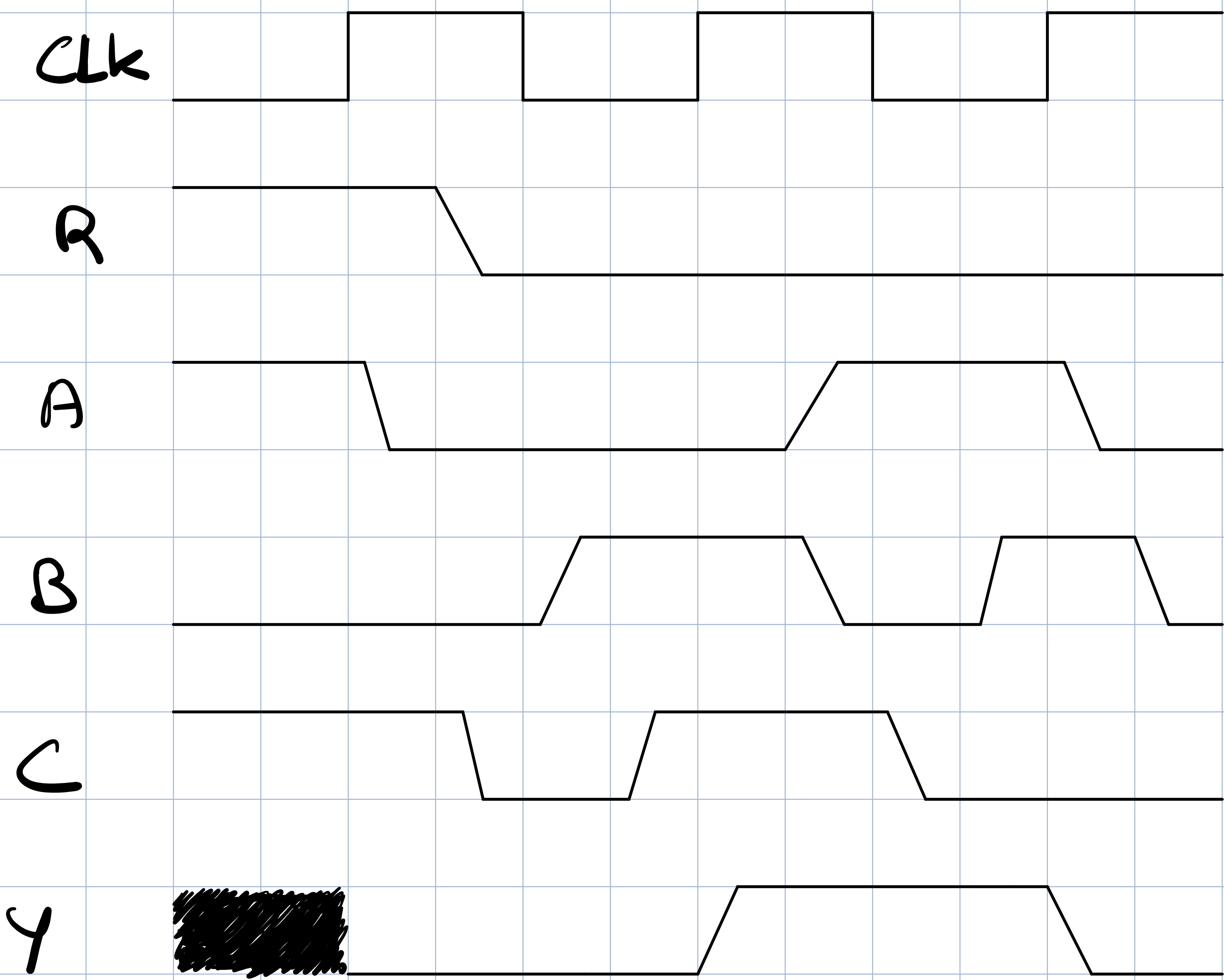
// Verilog

Exercise 25

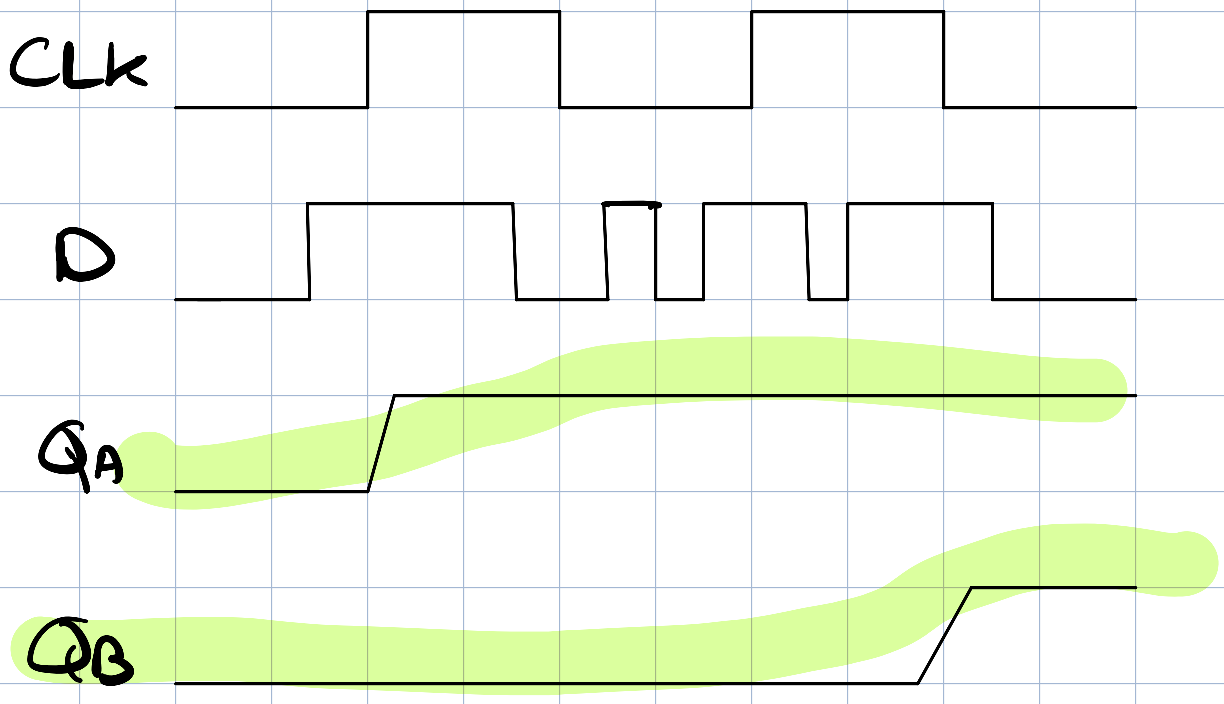
(a)

$$y(t) = \bar{y}(AB + C)$$




Clk
R
A
B
C
Y

Exercise 27



Exercise 28

(1) $X_0 X_1 X_2 X_3 = \begin{pmatrix} 1000 \\ 0100 \\ 1100 \\ 0010 \\ \vdots \end{pmatrix}$

$Y_0 Y_1 Y_2 Y_3 = \begin{pmatrix} 1000 \\ 0100 \\ \vdots \end{pmatrix}$

(2) (a) 10 possible states: $\{0, 1, 2, 3, \dots, 9\}$

(b) $10 \cdot 10$ possible states: $\{0, \dots, 99\}$

(3) to count from 0 to 99.

(4) // Verilog

Exercise 27

(a)

2^3-1

Q_2	Q_1	Q_0
0	0	1
0	1	0
1	0	0
0	1	1
1	1	0
1	1	1
1	0	1

$Q_1 = \overline{Q_0} \text{ XOR } \overline{Q_2}$

$Q_0 = \overline{Q_2}$

$Q_2 = \overline{Q_1}$

(b) multiplexers

Exercise 30

(a)

```
module up_down_counter (

        input reset, load, clk,
        input [0:4] load_val,
        input up,
        output reg [0:4] out
);


    always @(posedge clk) begin

        if (reset) out = 0;
        else if (load) out = load_val;
        else out = up ? out++ : out--;
    }

endmodule
```

```
module up_down_counter_tb;

    reg load = 1;
    reg reset = 0;
    reg clk = 0;
    reg [3:0] load_val = 4'b0000;
    wire [3:0] out;

    up_down_counter udc (
            .load(load),
            .reset(reset),
            .clk (clk),
            .load_val(load_val),
            .out(out)
    );

    always begin
        #5;
        clk = ~clk
    end
```

```
initial begin

  #20; wait for 4 clock cycles

  load = 0;

  #10; wait for a clock cycle

  if(out != 0001) begin
     $error("something went wrong");
  end else $display("ok! [%b]", out);
  #10;
  if(out != 0010) begin
     $error("something went wrong");
  else
   ;
   ;
   ;
   ;
```

Exercise 31

a) $\log(4096) = 12$

b) // Verilog

Exercise 32

(a) $\lceil \log(k) \rceil$ bits

(b) // Verilog

Exercise 33

$Q_0^* = b$

$Q_1^* = Q_0$

$Q_2^* = Q_1$

$x^* = (!Q_2)Q_1$

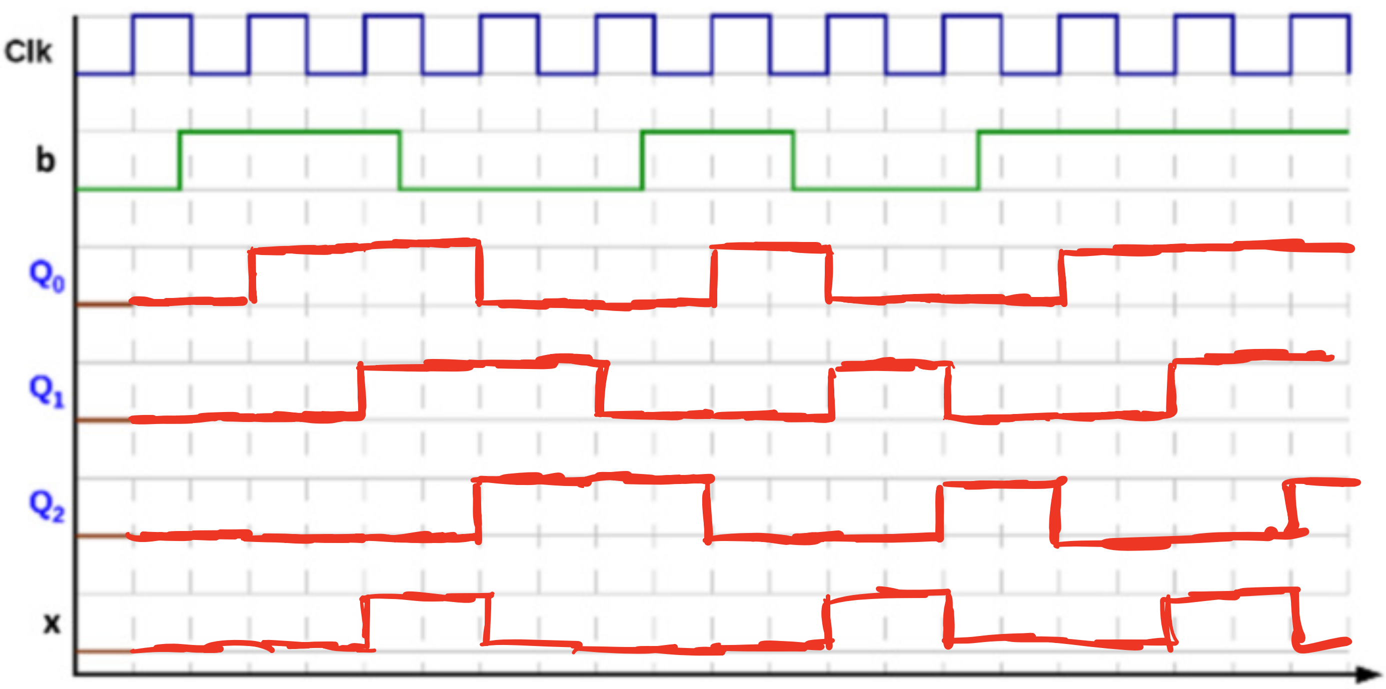


Figure 70: Waveform Template

→ we add several flip-flops to avoid metastability issues

→ it is a rising edge detector

Exercise 34

① $Y_1 = (\overline{y_1} w) + (y_2 w) = w(\overline{y_1} + y_2)$

$Y_2 = (y_1 w) + (y_2 w) = w(y_1 + y_2)$

$z = Y_1 Y_1$

②

		$w=0$	$w=1$	
y_1	y_2	$Y_1 Y_2$	$Y_1 Y_2$	z
0	0	00	10	0
0	1	00	11	0
1	0	00	01	0
1	1	00	11	1

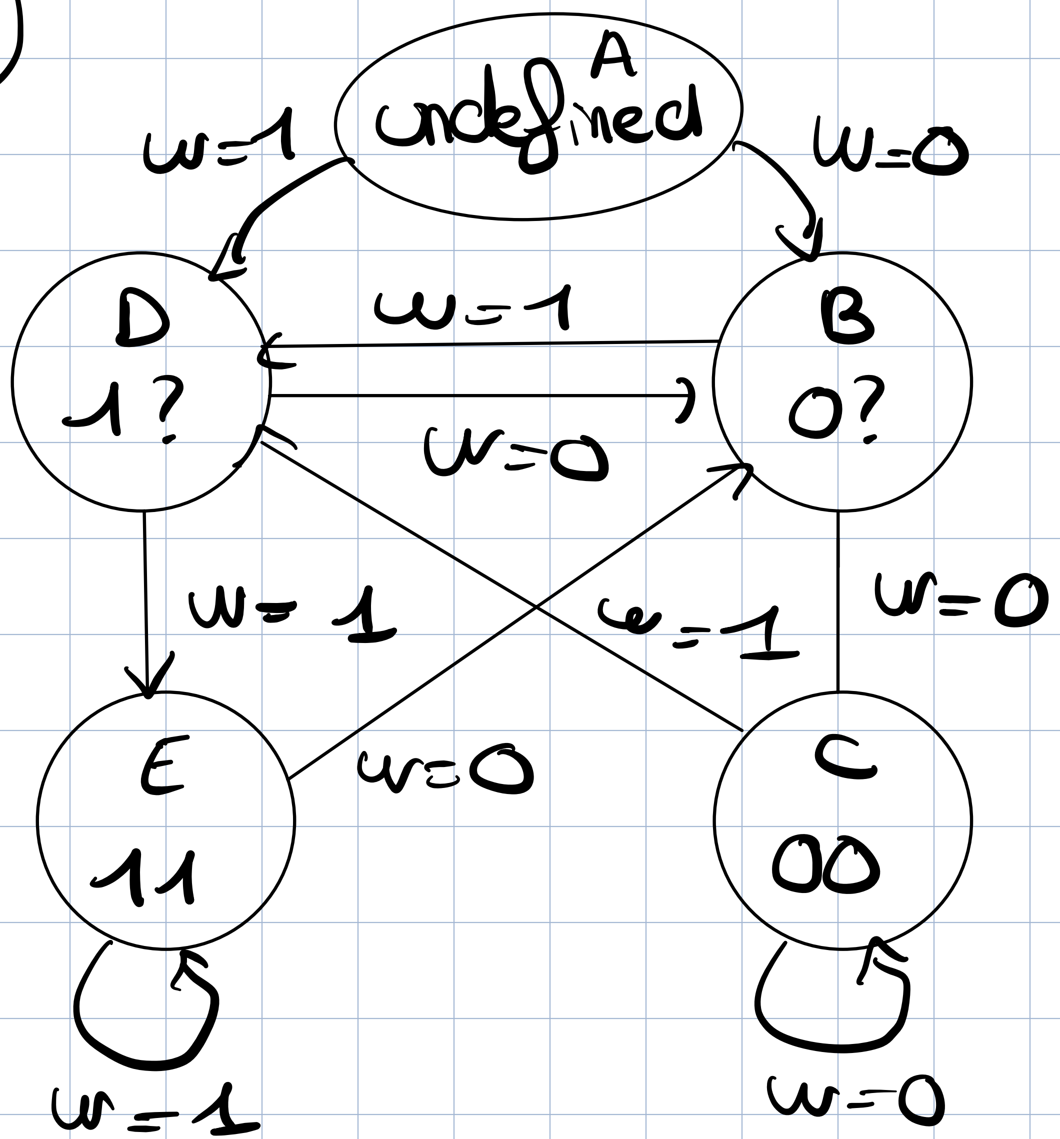
3 $00 \to 10 \to 01 \to 11$

sequence detector

Exercice 35

① S

②



3

$Y_2 Y_1 Y_0$		Next state $w=0$		$w=1$		output Z
000	A	001	B	011	D	0
001	B	010	C	011	D	0
010	C	010	C	011	D	1
011	D	001	B	100	E	0
100	E	001	B	100	E	1

$$Y_2^* = \left(\overline{Y_2} Y_1 Y_0 + Y_2 \overline{Y_1}\,\overline{Y_0}\right) W$$

$$Y_1^* = \overline{Y_2}\left(\overline{Y_0} + \overline{Y_1} Y_0\right) W$$

$$+ \overline{Y_2}\left(Y_1 \text{ XOR } Y_0\right) \overline{W}$$

$$Y_0^* = W\left(Y_1 \text{ XOR } Y_0 + \overline{Y_2}\,\overline{Y_1}\,\overline{Y_0}\right)$$

$$+ \overline{W}\left(Y_1 \text{ XOR } Y_0\right)$$

$$z = \overline{y_2} y_1 \overline{y_0} + y_2 \overline{y_1} \overline{y_0}$$

$y_4 y_3 y_2 y_1 y_0$		next state $w=0$		next state $w=1$		output z
00001	A	00010	B	01000	D	0
00010	B	00100	C	01000	D	0
00100	C	00100	C	01000	D	1
01000	D	00010	B	10000	E	0
10000	E	00010	B	10000	E	1

1 ✓

2
$$y_4^* = w(y_4 + y_3)$$
$$y_3^* = w(y_1 + y_2 + y_3)$$
$$y_2^* = \overline{w}(y_3 + y_4)$$
$$y_1^* = \overline{w}(y_3 + y_4)$$
$$y_0^* = 0$$

3 a ↗ more flip-flops

b ↘ less

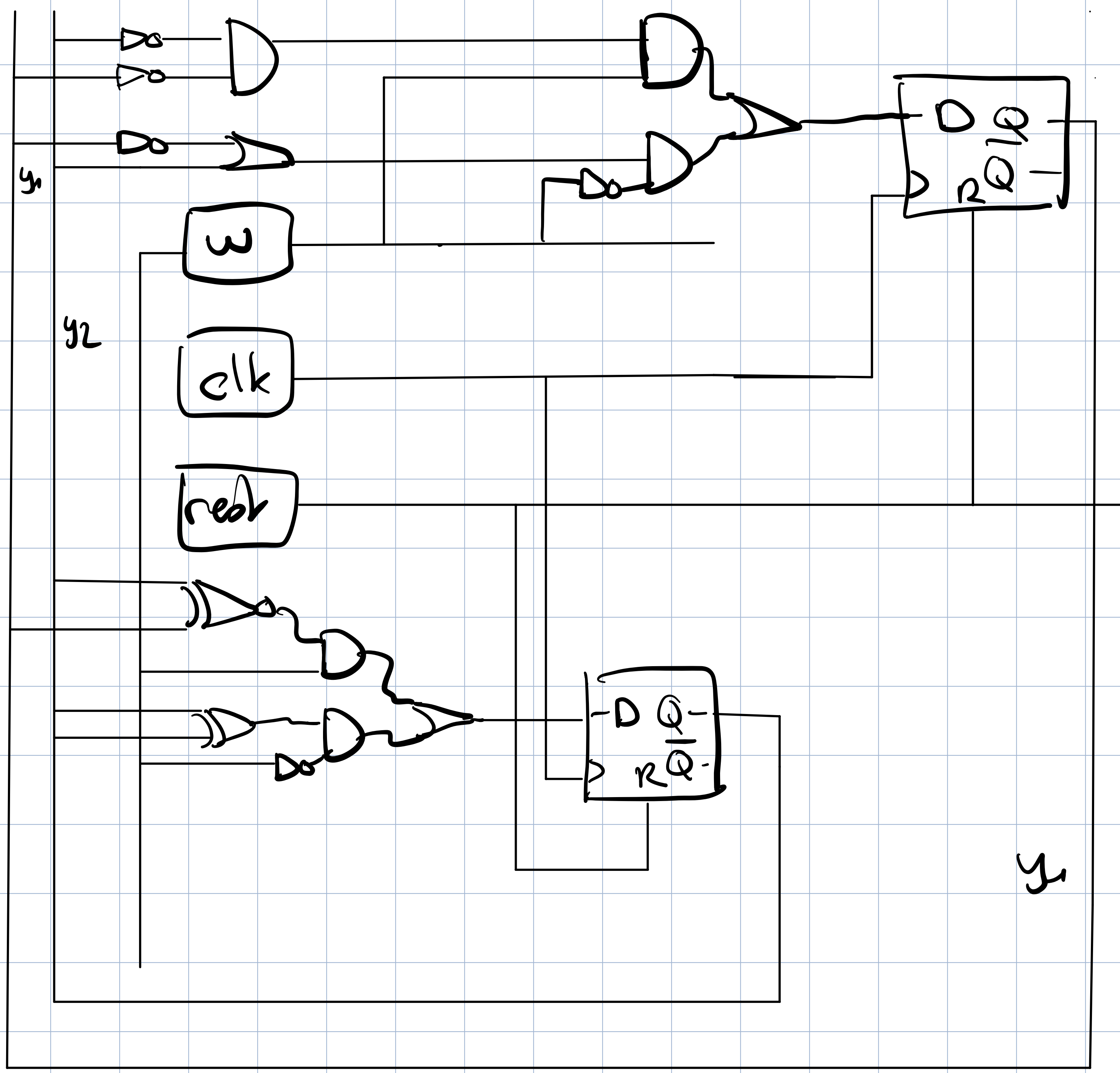
Exercise 36

① $Y_2 = w(\bar{y}_2 \bar{y}_1)$
$\quad + \bar{w}(y_2 + \bar{y}_1)$

$Y_1 = w(\overline{y_1 \oplus y_2})$
$\quad + \bar{w}(y_1 \oplus y_2)$

$z = y_2 y_1$

② Moore → only depends on the previous state


y1
y2
w
clk
reset
D
Q
R
y1

Exercise 37

(1)

inputs: $\{1, 2, 5\}$

outputs: $\{0, 1\}$

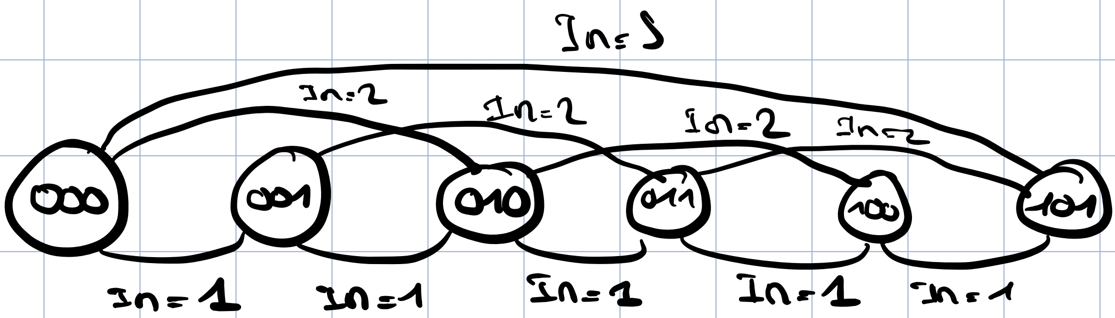
internal states: $\{0, 1, 2, 3, 4, 5, 6, 7\}$

(2) inputs: $\{1, 2, 5\}$

outputs: $\{0, 1\}$

internal states: $\{000, 001, \dots, 111\}$

(3)



etc.

(4)

	Inputs			Output Input		
States	1	2	S	1	2	S
0	1	4	S	0	0	0
1	2	3	6	0	0	0
2	3	4	7	0	0	0
3	3	S	0	0	0	1
4	S	6	1	0	0	1
S	6	0	2	0	1	1
6	7	0	3	0	1	1
7	0	1	4	1	1	1

Inputs

States	1	2	5	Output
0	1	4	5	0
1	2	3	6	0
2	3	4	7	0
3	3	5	0	0
4	5	6	1	0
5	6	0	2	0
6	7	0	3	0
7	8	9	12	0
8				1
9				1
10			7	1
11	12	5		1
12				1

Exercise 38

(a) $t_{setup} + t_{comb,max} + t_{cq,max}$ has to fit within one clock cycle.

$0.6ns + (1+0.2)\cdot 2ns + 1ns$

$= 0.6 + 2.4 + 1 = 4ns$ (Q_1 to D_2)

$0.6ns + (1+0.1)ns + 1ns$

$= 2.7ns$ (Q_2 to D_3)

$0.6ns + (0)ns + 1ns = 1.6ns$
(Q_3 to D_1)

$0.6ns + (1.2)ns + 1ns = 2.8ns$
(Q_2 to D_2)

$$0.6\text{ns} + (2.4)\text{ns} + 1\text{ns} = 4\text{ns}$$
(Q_3 to D_2)

$$f_{max} = \frac{1}{4\text{ns}}$$

$$t_{hold} \leq t_{comb_{min}} + t_{cq,min}$$

$Q1/D_2 \Rightarrow 3.2\text{ns}$ ✓
$Q2/D_3 \Rightarrow 1.9\text{ns}$ ✓
$Q3/D_1 \Rightarrow 0.8\text{ns}$ ✓
$Q2/D_2 \Rightarrow 2\text{ns}$ ✓
$Q3/D_2 \Rightarrow 3.2\text{ns}$ ✓

$- t_{setup}$
$- \Delta(t_{comb_{max}}, t_{comb_{min}})$

(b)

Q_1 to D_2

$0.6ns + 2.4ns + 1 - 0.7ns = 3.3ns$

Q_2 to D_3

$0.6ns + 1.1ns + 1 + 0.7ns = 3.4ns$

Q_3 to D_2

$0.6ns + 2.4ns + 1 - 0.7ns = 3.3ns$

Q_2 to D_2

$0.6ns + 1.1ns + 1 = 2.7ns$

Q_3 to D_1

$0.6ns + 1ns = 1.6ns$

$$f_{max} = \frac{1}{3.4ns}$$

$$t_{comb,min} + t_{cq,min} \geq t_{hold} + skew$$

(it has to stay longer)

$Q1/D2$: $3.2ns - 0.7ns = 2.5ns$

$Q2/D3$: $1.9ns + 0.7ns = 2.6ns$

Q_3/D_1: $0.8ns$

Q_2/D_2: $2ns$

$Q3/D_2$: $3.2ns - 0.7ns = 2.5ns$

(c)

Q_1/D_2: $3.2ns$

Q_2/D_3: $1.9ns - 0.5ns = 1.4ns$

Q_3/D_1: $0.8ns - (1 - 0.5ns) = 0.3ns$

Q_2/D_2: $2ns$

Q_3/D_2: $3.2ns + 0.5ns = 3.7ns$

f_{max} :

a

Q_1 / D_2 : $4ns + 1ns = 5ns$

$f_{max} = \boxed{\dfrac{1}{5ns}}$

Exercise 39

a

State	In: 0	In: 1	out
IDLE	IDLE	A0	0
A0	IDLE	A1	0
A1	IDLE	A2	0
A2	IDLE	A3	0
A3	IDLE	A4	0
A4	IDLE	A5	0
A5	IDLE	A6	0
A6	IDLE	A7	0
A7	IDLE	A8	0
A8	IDLE	A9	0
A9	IDLE	B	0
B	IDLE	B	1

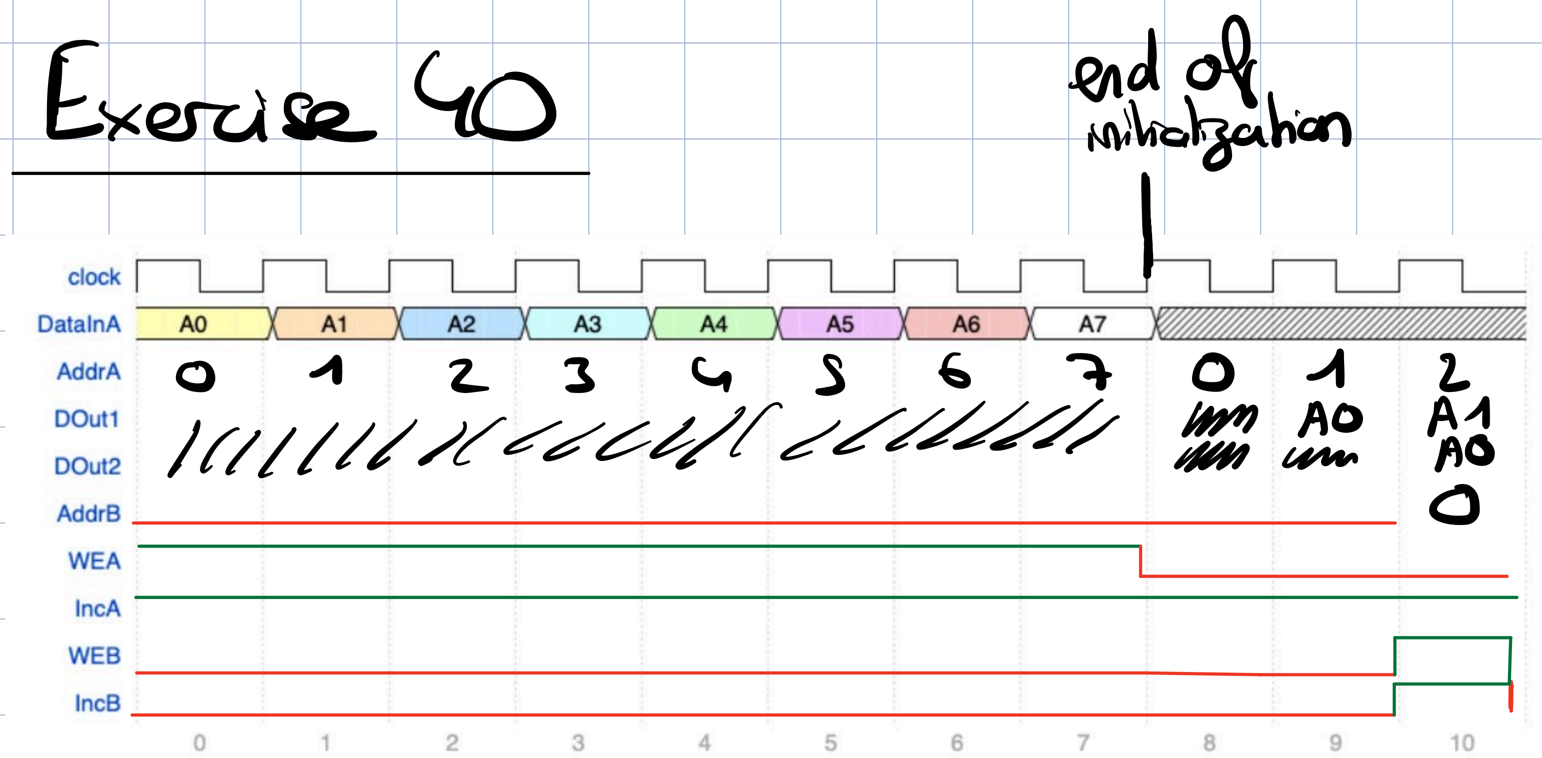
State	Counter	In	NextS	NextC	out
IDLE	x	1	A	0	0
A	0	1	A	1	0
A	1	1	A	2	0
A	2	1	A	3	0
A	3	1	A	4	0
A	4	1	A	5	0
A	5	1	A	6	0
A	6	1	A	7	0
A	7	1	A	8	0
A	8	1	A	9	0
A	x	0	IDLE	x	0
A	9	1	B	9	1
B	9	x	IDLE	9	0
IDLE	x	0	IDLE	x	0

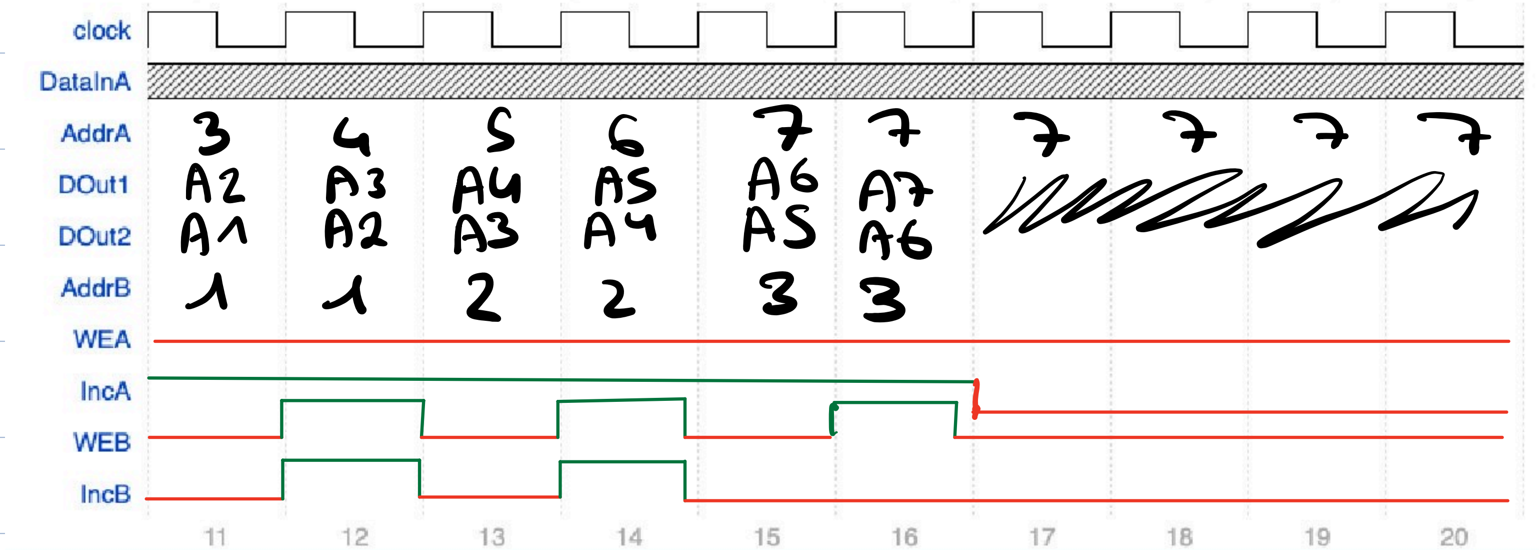
```
module ten_counter (

        input clk, reset, in,
);

reg IDLE_STATE = 2'b00;        | reg counter
reg A_STATE = 2'b01;           |      = 0;
reg B_STATE = 2'b10;           |
reg current_state = 2'b00;     |
reg next_state = 2'b00;        |

always @* begin

 if (in) begin

 if (current_state == IDLE_STATE) begin
       next_state <= A-STATE;
       counter <= 0;
  end else if (current_state == B_STATE) begin
       next_state <= IDLE_STATE;
  end else if (current_state == A STATE)
```

```
                    next = B
end else begin
   if (current_state == A_STATE) begin
      next_state <= IDLE_STATE;
   end else if (current_state == B_STATE) begin
      next_state <= IDLE_STATE
   end
end


always @(posedge clock) begin

      if (reset) begin
        current_state <= IDLE_STATE;
        counter <= 0;
      end else begin
        current_state <= next_state;
        if (A_STATE) counter <= counter+1;
      end

end

always @* begin
      out = current_state == B_STATE
```

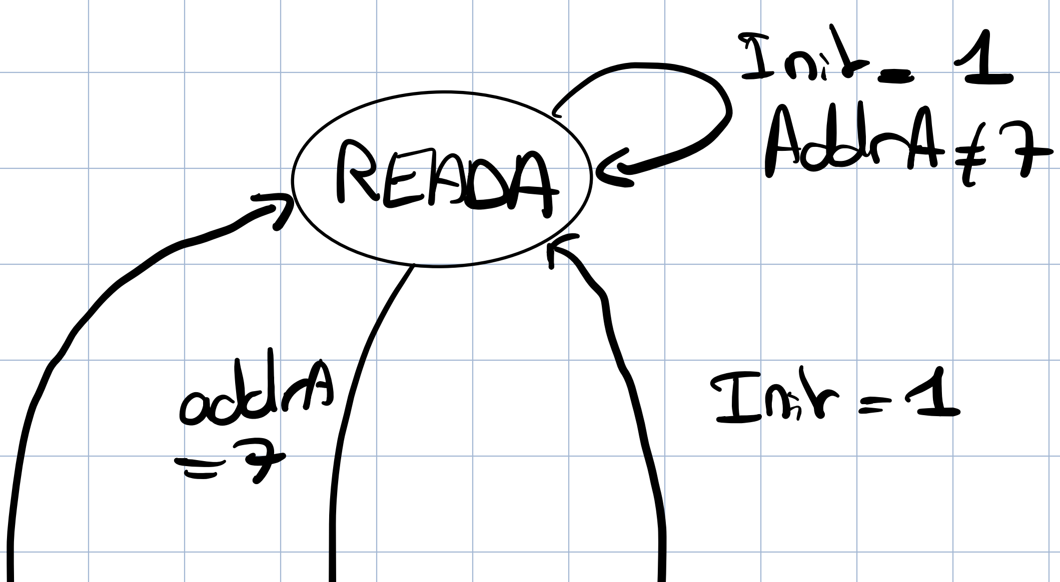
en 1

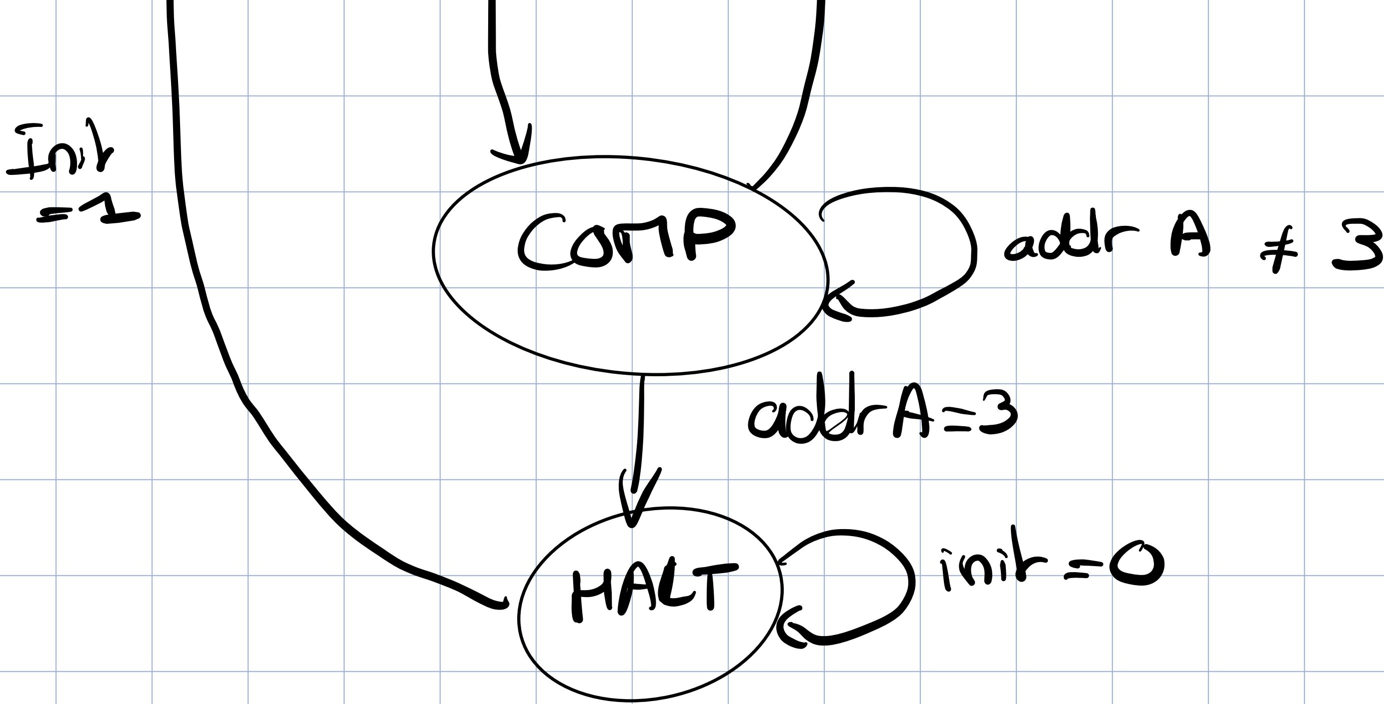
Exercise 40



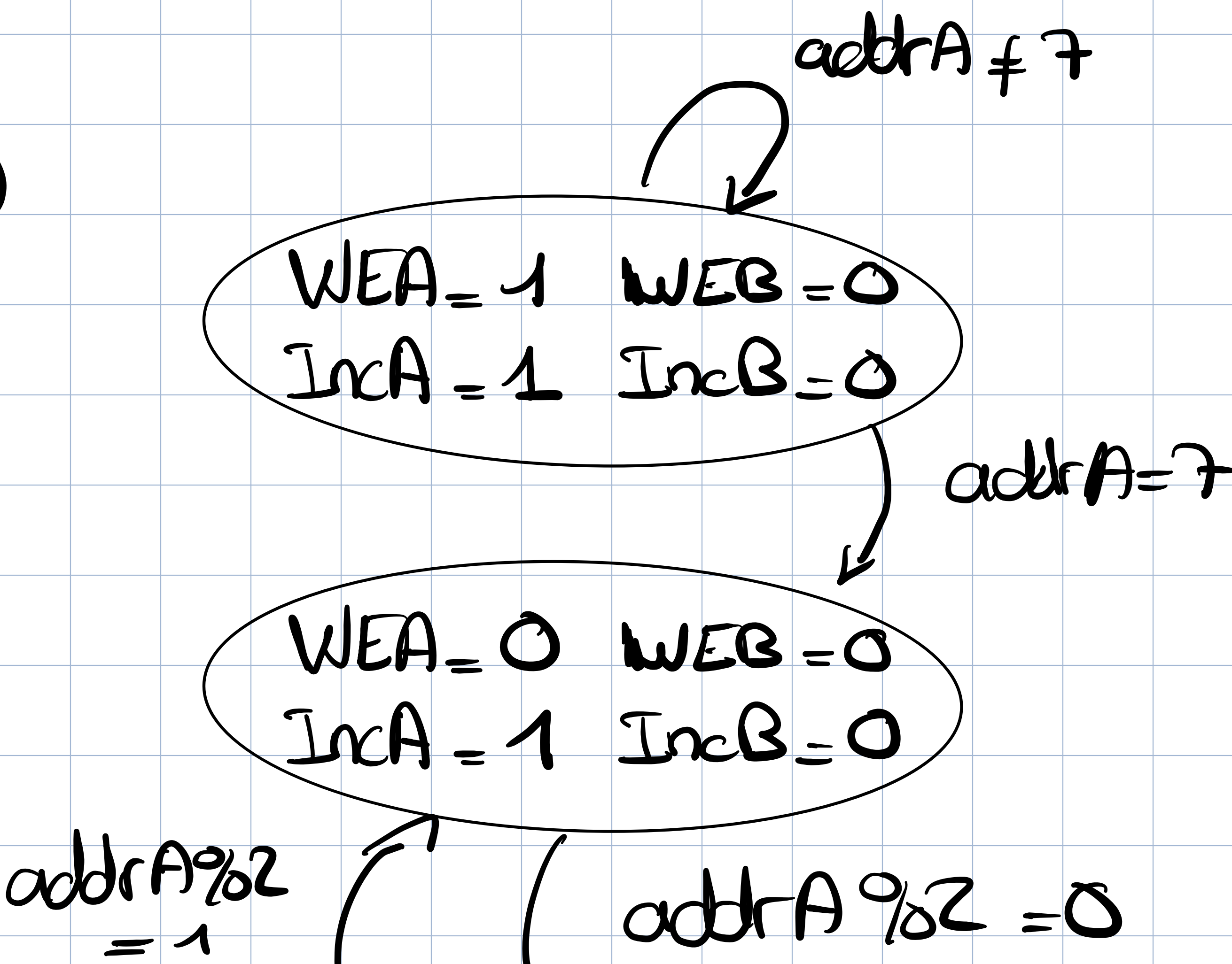


(b)




Init =1
COMP
addr A ≠ 3
addrA=3
HALT
init=0

(c)


addrA ≠ 7
WEA=1 WEB=0
IncA=1 IncB=0
addrA=7
WEA=0 WEB=0
IncA=1 IncB=0
addrA%2 =1
addrA%2 =0

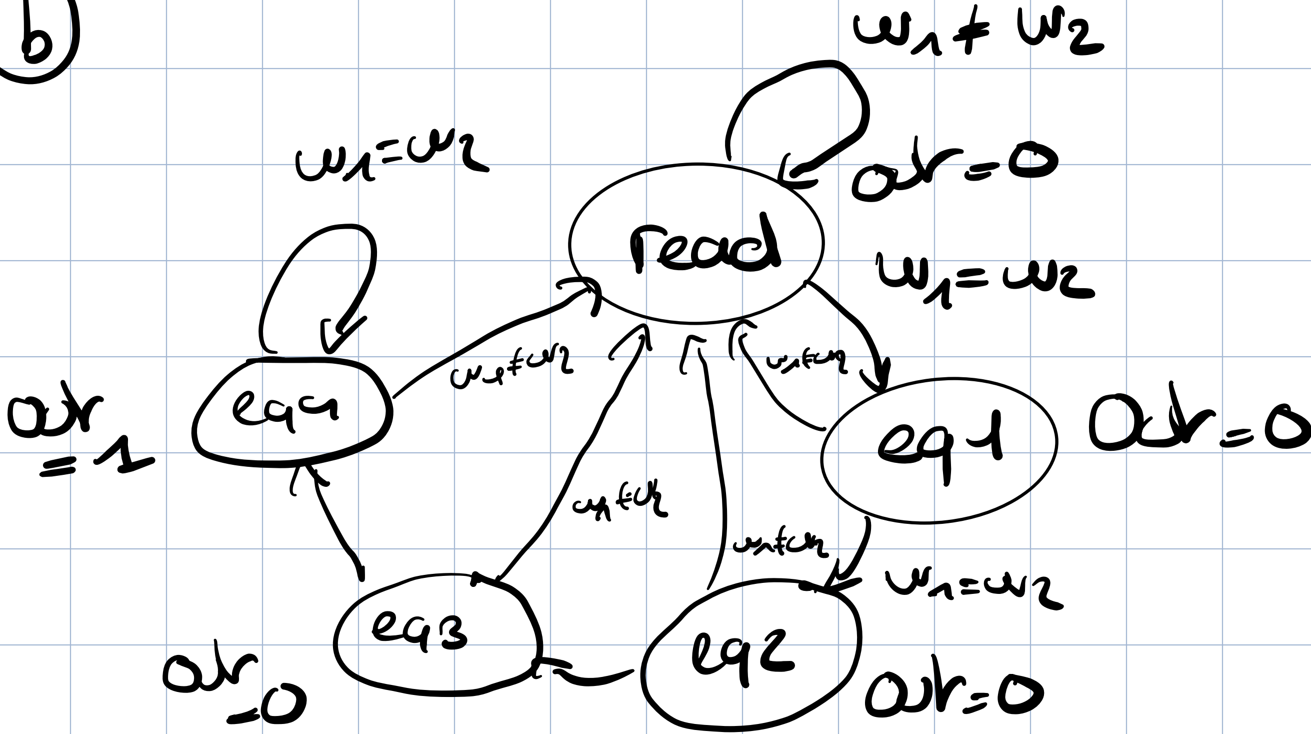
WEA = 0 WEB = 1
IncA = 1 IncB = 1

Current				Next			Outs			
St	CntA	CntB	Init	St	CntA	CntB	IncA	IncB	WEA	WeB
R	0	0	0	R	1	0	1	0	1	0
R	1	0	0	R	2	0	1	0	1	0
R	2	0	0	R	3	0	1	0	1	0
R	3	0	0	R	4	0	1	0	1	0
R	4	0	0	R	5	0	1	0	1	0
R	5	0	0	R	6	0	1	0	1	0
R	6	0	0	R	7	0	1	0	1	0
R	7	0	0	C	0	0	1	0	1	0
C	0	0	0	C	1	0	1	0	0	0
C	1	0	0	C	2	0	1	0	0	0
C	2	0	0	C	3	0	1	1	0	1
C	3	1	0	C	4	1	1	0	0	0
C	4	1	0	C	5	1	1	1	0	1
C	5	2	0	C	6	2	1	0	0	0
C	6	2	0	C	7	2	1	1	0	1
C	7	3	0	C	0	3	1	0	0	0
C	0	3	0	H	0	3	0	1	0	1
H	0	0	0	H	0	0	0	0	0	0

Exercise 41

(a) S

(b)



(c)

	Next State		
State	$w_1 = w_2$	$w_1 \neq w_2$	Output
R	E_1	R	0
E1	E_2	R	0
E2	E_3	R	0
E3	E4	R	0
E4	E4	R	1

(d) module two_input_seq_detector (
input clk, reset, w1, w2,
output out
);

```
reg [2:0] next_state;
reg [2:0] current_state;
parameter [2:0] R = 3'b000;
parameter [2:0] E1 = 3'b001;
parameter [2:0] E2 = 3'b010;
parameter [2:0] E3 = 3'b011;
parameter [2:0] E4 = 3'b100;

always @* begin
    next_state = current_state;
    if(w1 == w2) begin

        case (current_state)
            R: next_state = E1;
            E1: next_state = E2;
            E2: next_state = E3;
```

```
            E3: next_state = E4;
            E4: next_state = E4;
            default: next_state = E4;
        endcase
    end else begin
        next_state = R;
    end
end

always @(posedge clk or posedge reset) begin

    if (reset) current_state <= R;
    else current_state <= next_state;
end

always @* begin
    out = current_state == E4 ? 1 : 0;
end

endmodule
```

Exercise 42

(a)

```
module bit_counter(
        input clk,
        input reset,
        input [7:0] load_data,
        input load,
        input enable,
        output reg [3:0] count,
        output reg done
);
  reg currently_reading_pos;
  reg [7:0] data;
  reg [1:0] next_state;
  reg [1:0] current_state;

  parameter S1 = 2'b00, S2 = 2'b01,
            S3 = 2'b11;
```

```
always @* begin

    if(current_state == S1) begin
        if(load) begin
            data = load_data;
        end
        if(enable) begin
            next_state = S2;
        end
    end else if(current_state == S2) begin
        if(currently_reading_pos == 7) begin
            next_state = S3
        end
    end
end
```

```
always @ (posedge clock or posedge Reset)
                                    begin
    if(Reset) begin
        currently_reading_pos <= 0;
        count <= 0;
    end else if (current_state == S2) begin
        count <= data[currently_reading_
                  _pos] == 1 ? count + 1 : count;

        currently_reading_pos <= currently_rea-
    end                          ding_pos + 1;
end


always @(posedge clock or
             posedge Reset) begin

    if (Reset) begin
        current_state <= S1;
    end else begin
        current_state <= next_state;
```

```
        end
end

always @* begin
    done = current_state == S3 ? 1 : 0;
end
```

Exercise 43

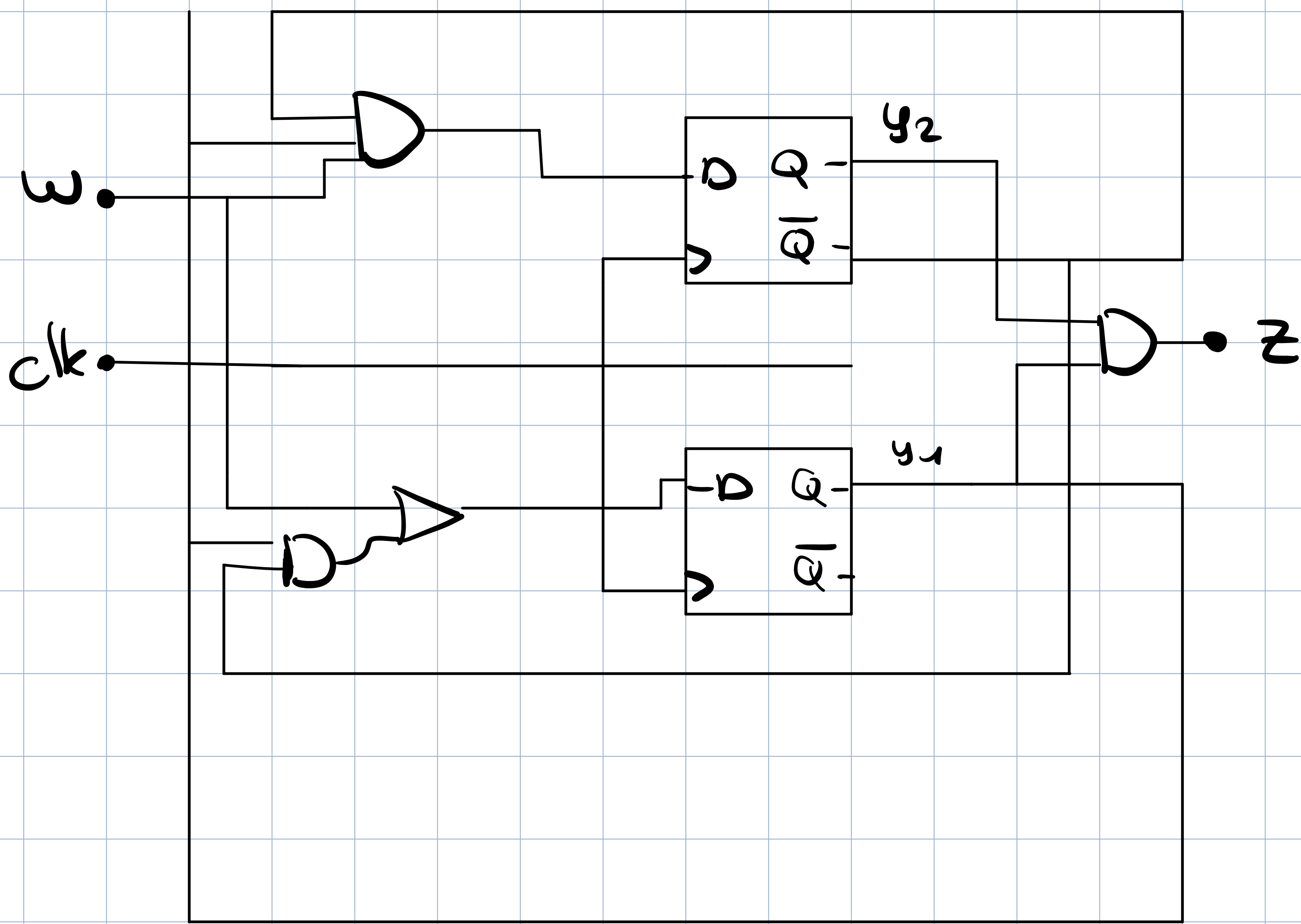
(a)

State y_2y_1	Input $w=0$	$w=1$	Output
00 A	00 A	01 B	0
01 B	01 B	11 C	0
11 C	00 A	01 B	1

$$Y_2^* = w(\overline{y_2}\, y_1)$$

$$Y_1^* = w + \overline{y_2}\, y_1$$

$$z = y_2 y_1$$



$Q_2 \rightarrow D_2$

$t_{setup} + t_{cq,max} + t_{comb,max} > t_{cycle}$

$100 + 200 + 400 = 700ps$

$Q_1 \rightarrow D_1$

$100 + 200 + 300 + 400 = 1000ps$

$$f_{max} = \frac{1}{1000}$$

Exercise 2

(a) (1) $\overset{rd}{t_0 : 5}$ $\overset{r_1}{t_1 : 6}$ $\overset{r_2}{t_2 : 7}$

R-type

| 0x00 | 0111 | 0110 | 0x0 | 0101 | 0x33

$\rightarrow$ | 0000000 | 00111 | 00110 | 000 | 00101 | 0110011

$t_0 = 15 = F$

(2) $t_4 = 1$

$t_4 = x_{29}$ $s_0 = x_8$ $s_3 = x_{19}$

0000000 | 10011 | 01000 | 010 | 11101 | 0110011

(3) $t_2 = x_7$ $t_5 = x_{30}$

$\boxed{t_2 = 0}$

0000000 | 11110 | 00111 | 011 | 00111 | 0110011

(4) $t_0 = x_8$ $t_1 = x_6$

000000010000 | 00110 | 000 | 00101 | 010011

$t_0 = 0xD0$

(5) slli $s_2 = x_{18}$

0000000011 | 10010 | 001 | 10010 | 010011

$s_2 = 1\cancel{0}0000111101010000$

6 $t_6 = x_{31}$

$0x00012300$

$\underbrace{\overset{1 hex}{4} \cdot 3}_{3 hex} = 12$

$0x12300000$

$0x00012300 | 11111 |$ 0110111

Exercise 3

101
101

(a)

00010010000110100 0

$$t_0' = t_0 \oplus t_2$$
$$t_2' = (t_0 \oplus t_2) \oplus t_2 = t_0$$
$$t_0'' = t_0 \oplus t_2 \oplus t_0$$
$$= \underline{t_2}$$

bne r1, r1, 13 $\equiv$

$\Rightarrow$ bne r1, r2, 12

bne $r_1, r_2,$ imm
$= \boxed{42}$

Exercise 5

① $t_0 = x_5$ $t_1 = x_6$

$$\begin{array}{r} 32 \\ + \ 8 \\ + \ 2 \end{array}$$

40

42

imm << 1

$\times$

4 1

$\underline{101010\cancel{0}}$

B | 0000001 | 00110 | 00101 | 000 | 01010 |

110,0011

② bltu

$s_1 : x_9$ $s_2 : x_{18}$

4 1

$-4 : 111111111110\cancel{0}$

B | 1111111 | 10010 | 01001 | 110 | 1110 | 1 |

110,0011

③ lw $t_2 : x_7$

$s_3 : x_{19}$

B| 0000000 1100 | 10011 | 010 | 00111 | 0000011

④ $t_0 : x_5$ sb

$s_6 : x_{22}$

imm : 4 100

S| 0000000 | 00101 | 10110 | 000 | 00100 |

0100011

⑤ $t_0 : x_5$ 0x 12345

0

0001, 0010, 0011, 0100, 0101

5 | 0,0110100010,0,001010 | 00101 |
0110001 1

6 jalr t_1, 16(t_2)

t_1 : rd : x_6

t_2 : rs_1 : x_7

000000010000 | 00111 | 000 | 00110 |
1100111

Simon Lefort
Il y a 4 minutes

Exercise 10

Hello, yes, you should try to understand what the program does so you can do the computation by yourselves without executing assembly on paper.

Here is how I did it (maybe not the best way but it worked for the few exercises we have)

first, let's start with the `start` part:

- hum, t0, t1, r3 get initialized with the value 0, they are probably some counters (we don't know what they will count yet)
- then t5 takes the value s0... from the statement... all right, it is the starting address for the matrix. it will probably be a pointer towards the current memory address that we will increment when we want to read a new address

ok, now the `loop` part.

- `lub` says we want to load a unsigned byte (8 bits) to a register... why would we want to do so? ok the statement says we are storing a bunch of 8 bits unsigned number in our memory. so that's probably the first element of our matrix!
- then... t2 = t2 + s3. ok, we don't know why they do this yet
- then they are comparing t2 and t3... and jumping to skip if... t2 < t3 is 1 which means that t2 is indeed less than t3 (signed comparison!). same thing, no idea why they would want to this yet but anyway let's keep reading

I personnally then went to the `skip` part

- so if `t2 < t3` then we store at address `t5` (the current address) to the value of `t2` but wait what is `t2` now? it is `t2 + s3` right and what is `s3` again? ok, the unsigned number they said to be chosen arbitrarily. So basically... if the current element + the arbitrary number is less than the counter t3 (we don't know what it is yet)... we set current element to itself + s3. Right.
- then, we increase t5 which was the pointer to the address (so basically we set the address to the next address)
- then we increase t0 by 1... right
- then we are comparing t0 with s1.. which is the number of rows! ahah, we found it, now we know that t0 is the current index of the row, and t1 is probably the current index of column!
- that's confirmed by the rest of the `skip` part, we see that if we arrive at the end of the rows, we are resetting the counter for the rows and increasing the counter for the column
- and then loop again to read the next element and process

now we know what is `t0` (idx of row), `t1` (idx for col), `t2` (current element + arbitrary number), `t4` (the result of current element + abitrary number < t3) (even though we don't know what is `t3` yet)

and we also know that when (the result of current element + abitrary number < t3), we are saving as the current element `itself + arbitrary number`.

So what is `t3`? well, let's see what's after bne

`s4 = t0 + 0`... `s5 = t1 + 0`... oh! so `s4` and `s5` are row and colum indices!

and `t3 = t2 + 0` oh so if `t2 > t3` then... `t3 = t2` ? that really seems like an algorithm to find an a max!

now we know what are all the registers used in our program

(i) S_4 = # row
S_5 = # col

$12+78+235 = 125$
(neg)

$S_4 = 3$
$S_5 = 1$

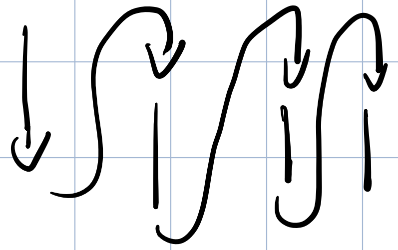
(ii) 12

(iii) all elements get +P (because we will not have any negative value

(iv) wheher we want the absolute max or just max

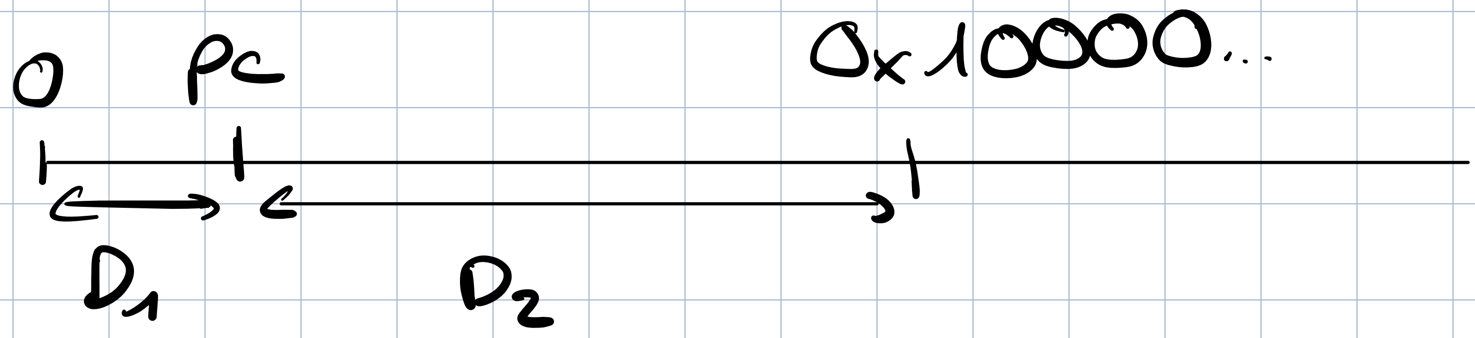
(b) see ED.

(c) 0x1000 0000 ⇒ 12
78
35



le programme est codé par line comme ça

11
34
⋮



auipc x_8 0x10000...

$\Rightarrow x_8 = D_1 +$ 0x10000...

add x_8, zero, $-D1$

$\Rightarrow x_8 =$ 0x10000...

(e)

(i) 8b+8b can lead to 9 bits

(ii) le save $\Rightarrow$ 32 bits to 8 bits.

```
srl  t6, t2, 8
beq  t6, zero, continue
addi t2, zero, 0xFF
continue:
```

Exercise 11

```
# will be my current pointer
add t0, zero, s0
# will be my counter of A[i] > B[i]
add s1, zero, zero

loop:

  lw   t2, 0(t0)
  beq  t2, zero, finish
  add  t3, zero, t2
  srai t3, t3, 16      ) clean up B
  slai t3, t3, 16
  srai t2, t2, 16      ) A
  slr  t4, t3, t2
  add  s1, s1, t4
  addi t0, t0, 4
  j loop
```