# Listă bibliografică pentru pregătirea examenului la disciplina “Fundamentele calculatoarelor”-an 1 INF-2020/2021


**1. Mircea Vlăduțiu:** “Computer Arithmetic. Algorithms and Hardware Implementations” Springer-Verlag, Heidelgerg, New York, Dordecht, London, 2012, ISBN 978-3-642-18314-0, ISBN 978-3-642-18315-7 (http://www.springer.com/computer/hardware/book/978-3-642-18314-0).

**2. Mircea Vlăduţiu**: „Arhitectura şi organizarea calculatoarelor” Vol.1: Aritmetica sistemelor de calcul (monografie), Editura Politehnica Timişoara, 2008 (274pagini), ISBN 978-973-625-706-3 (general), ISBN 978-973-625-709-4 (vol. 1).

**3. John L. Hennessy, David A. Patterson**: „Computer Architecture. A Quantitative Approach” Morgan Kaufmann Publishers, Inc., Fifth Edition, 2012.

**4. William Stallings**: „Computer Organization and Architecture. Designing for Performance” Prentice Hall, 11th Edition, 2018.

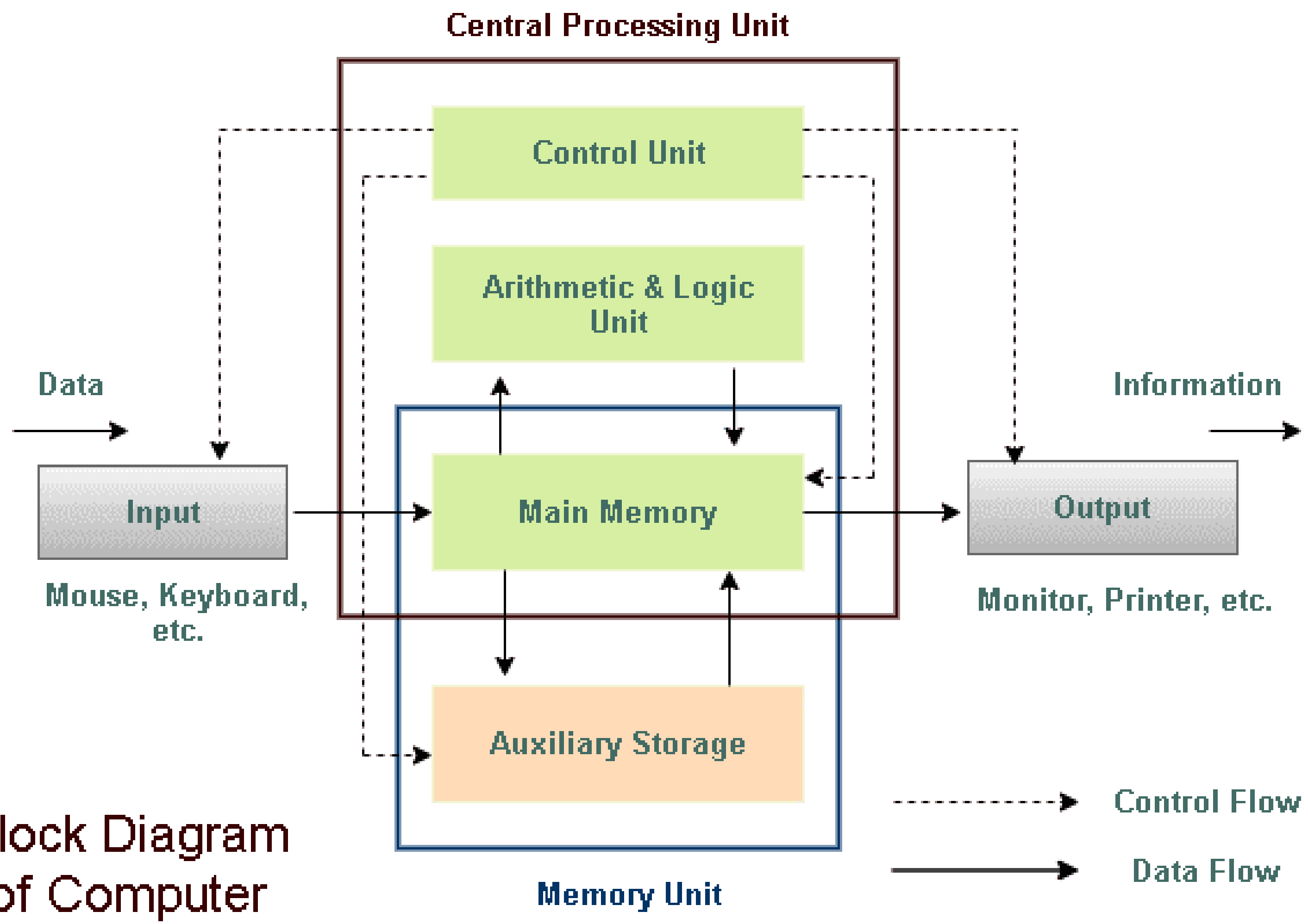
**5**. **David M**. **Harris. Sarah L. Harris**: „Digital Design and Computer Architecture” Morgan Kaufmann Publishers, Inc., Second Edition, 2012.

**6. Alan B. Marcovits**: „Introduction to Logic Desgn” Third Edition, Paperback, 2015.

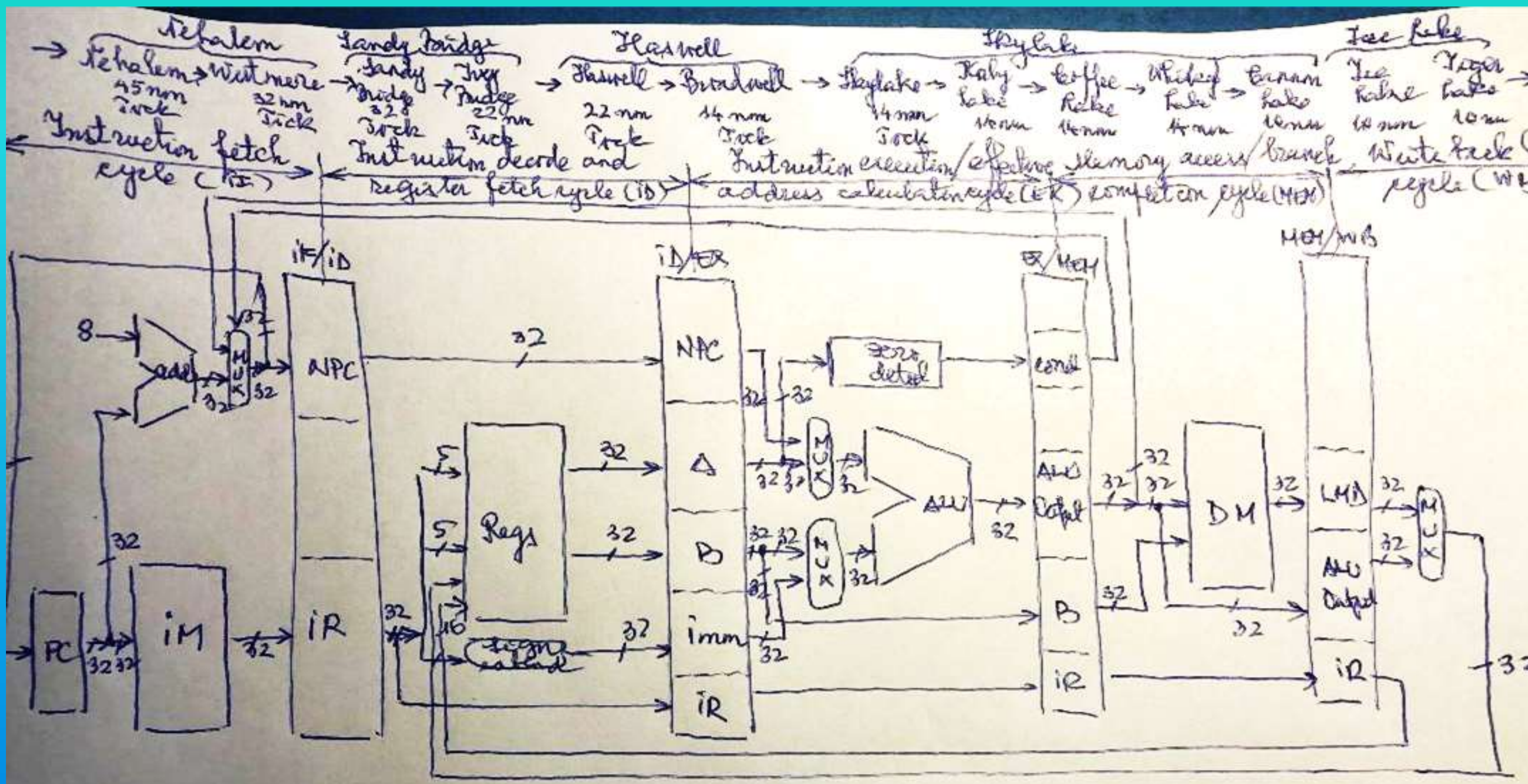
7. **John F. Wakerly**: „Digital Design: Principles and Practices” Fifth Edition, 2018.

**8**. **Tutorialspoint**: „Digital Circuits Design”

**https://www.tutorialspoint.com/digital_circuits/digital_circuits_logic_gates.htm**




# Block Diagram of Computer




PC – Program counter
iM – Instruction memory
iR – Instruction register
NPC – Next program counter

MUX – Multiplexer
Regs – Register file
A, B, imm – Buffer register
Immediate

ALU – Arithmetic/logic unit
DM – Data memory
LMD – Load memory data register

# Information Taxonomy

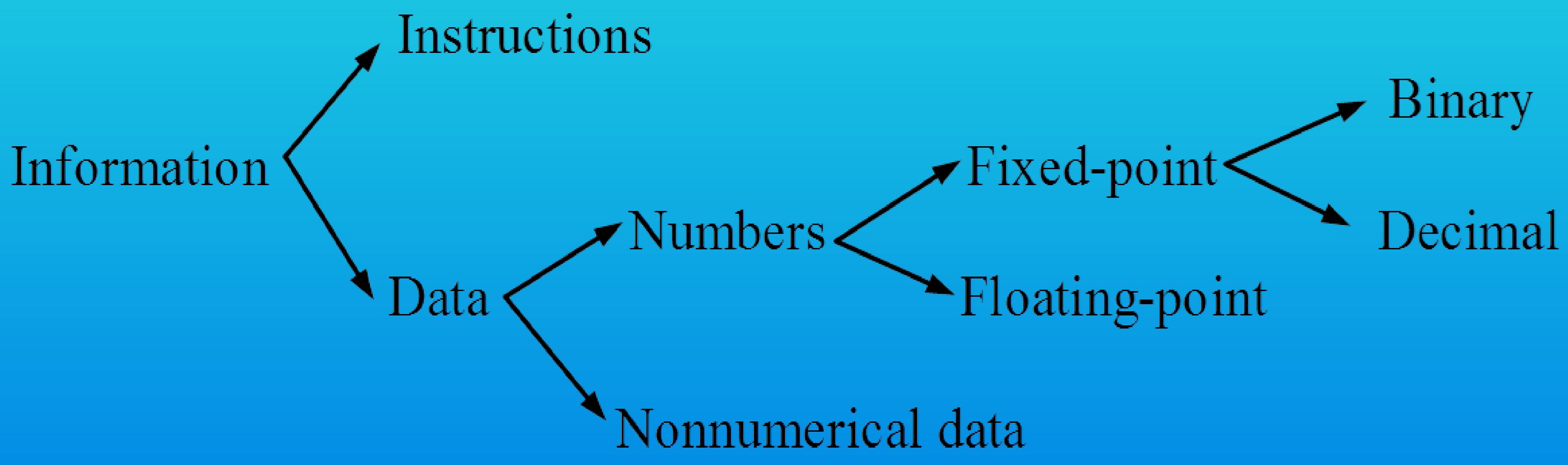



Fig.1.1

integer

$707_{10}$ digital to binary conversion

remainder method

$1011000011_2$

Computer

$1011000011_2$

binary to digital conversion

$707_{10}$

$$707_{10} = 7\times10^2 + 0\times10^1 + 7\times10^0$$

$$
\begin{array}{l|l}
707 & 2 \\
\hline
1 & 353 \mid 2 \\
1 & 176 \mid 2 \\
0 & 88 \mid 2 \\
0 & 44 \mid 2 \\
0 & 22 \mid 2 \\
0 & 11 \mid 2 \\
1 & 5 \mid 2 \\
1 & 2 \mid 2 \\
0 &
\end{array}
$$

$707_{10} = 1011000011_2$

$$1011000011_2 = 1\times2^9 + 0\times2^8 + 1\times2^7 + 1\times2^6 + 1\times2^1 + 1\times2^0 =$$

$$= 512 + 128 + 64 + 2 + 1 = 707_{10}$$

$$N = \sum_{i=0}^{n} x_i \cdot r^i$$

radix

$x_i \in [0,1]$, $i = 0, 1, \dots, n-1$

$\boxed{\text{Fractions}}$

$0{,}046875_{10} = 4\times10^{-2}+6\times10^{-3}+8\times10^{-4}+7\times10^{-5}+5\times1$

$0{,}046875_{10}$

multiplication method $\Downarrow$ digital to binary conversion

$0{,}000011_2$

Computer

$0{,}000011_2$

$\Downarrow$ binary to decimal conversion

$0{,}046875_{10}$

$0{,}046875_{10} = 0{,}000011_2$

$0{,}046875 \times 2$

$0{,}093750 \times 2$

$0{,}187500 \times 2$

$0{,}375000 \times 2$

$0{,}750000 \times 2$

$1{,}500000 \times 2$

$1{,}\boxed{000000}$

$2^{-1} \;\ldots\; 2^{-5}\,2^{-6}$

$$0{,}000011_2 = 0\times2^{-1}+0\times2^{-2}+0\times2^{-3}+0\times2^{-4}+1\times2^{-5}+1\times2^{-6} =$$

$$= \frac{1}{32} + \frac{1}{64} = \frac{3}{64} = 0{,}046875_{10}$$

$N = 707{,}046875_{10} = 1011000011{,}000011_2$

| Decimal digit | Fixed-point decimal codes | | |
|---|---|---|---|
| | BCD | E3 | 2-out-of-5 |
| 0 | 0000 | 0011 | 11000 |
| 1 | 0001 | 0100 | 00011 |
| 2 | 0010 | 0101 | 00101 |
| 3 | 0011 | 0110 | 00110 |
| 4 | 0100 | 0111 | 01001 |
| 5 | 0101 | 1000 | 01010 |
| 6 | 0110 | 1001 | 01100 |
| 7 | 0111 | 1010 | 10001 |
| 8 | 1000 | 1011 | 10010 |
| 9 | 1001 | 1100 | 10100 |

$743_{10}$

digital to binary-coded-decimal (BCD) conversion

$011101000011_{BCD}$

Computer

$011101000011_{BCD}$

binary-coded decimal (BCD) to digital conversion

$743_{10}$

$743_{10} = 0111\ 0100\ 0011_{BCD}$

$743_{10} = 1010\ 0111\ 0110_{E3}$

$743_{10} = 10001\ 01001\ 00110$ 2-out-of-5

$\overbrace{0111}^{7}\ \overbrace{0100}^{4}\ \overbrace{0011}^{3}{}_{BCD} = 743_{10}$

$\underbrace{1010}_{7}\ \underbrace{0111}_{4}\ \underbrace{0110}_{3}{}_{E3} = 743_{10}$

$\underbrace{10001}_{7}\ \underbrace{01001}_{4}\ \underbrace{00110}_{3\ \text{2-out-of-5}} = 743_{10}$

Decimal fixed-point numbers

- Binary-coded-decimal (BCD)
- Excess three (E3)
- Two-out-of-five (2-out-of-5)

| n-1 | n-2 | --- | i | ---- | 1 | 0 |
|---|---|---|---|---|---|---|

computer word, $\underbrace{\quad}_{n \text{ bits}}$

Binary fixed-point numbers → Sign-magnitude (SM)

Binary fixed-point numbers → One's complement (C1)

Binary fixed-point numbers → Two's complement (C2)

Sign-magnitude → integer: sign, magnitude $2^{n-2}$ … $2^{i}$ … $2^{1}$ $2^{0}$

| n-1 | n-2 | … | i | … | 1 | 0 |
|---|---|---|---|---|---|---|

Sign-magnitude → fraction: sign, $2^{-1}$ … $2^{-i}$ $2^{-(n-2)}$ $2^{-(n-1)}$

| n-1 | n-2 | … | i | … | 1 | 0 |
|---|---|---|---|---|---|---|

implicit binary point

| | SM | C1 | C2 |
|---|---|---|---|
| $N_1 = +117_{10}$ | 0 1 1 1 0 1 0 1 | 0 1 1 1 0 1 0 1 | 0 1 1 1 0 1 0 1 |
| $N_2 = -117_{10}$ | 1 1 1 1 0 1 0 1 | 1 0 0 0 1 0 1 0 | 1 0 0 0 1 0 1 1 |

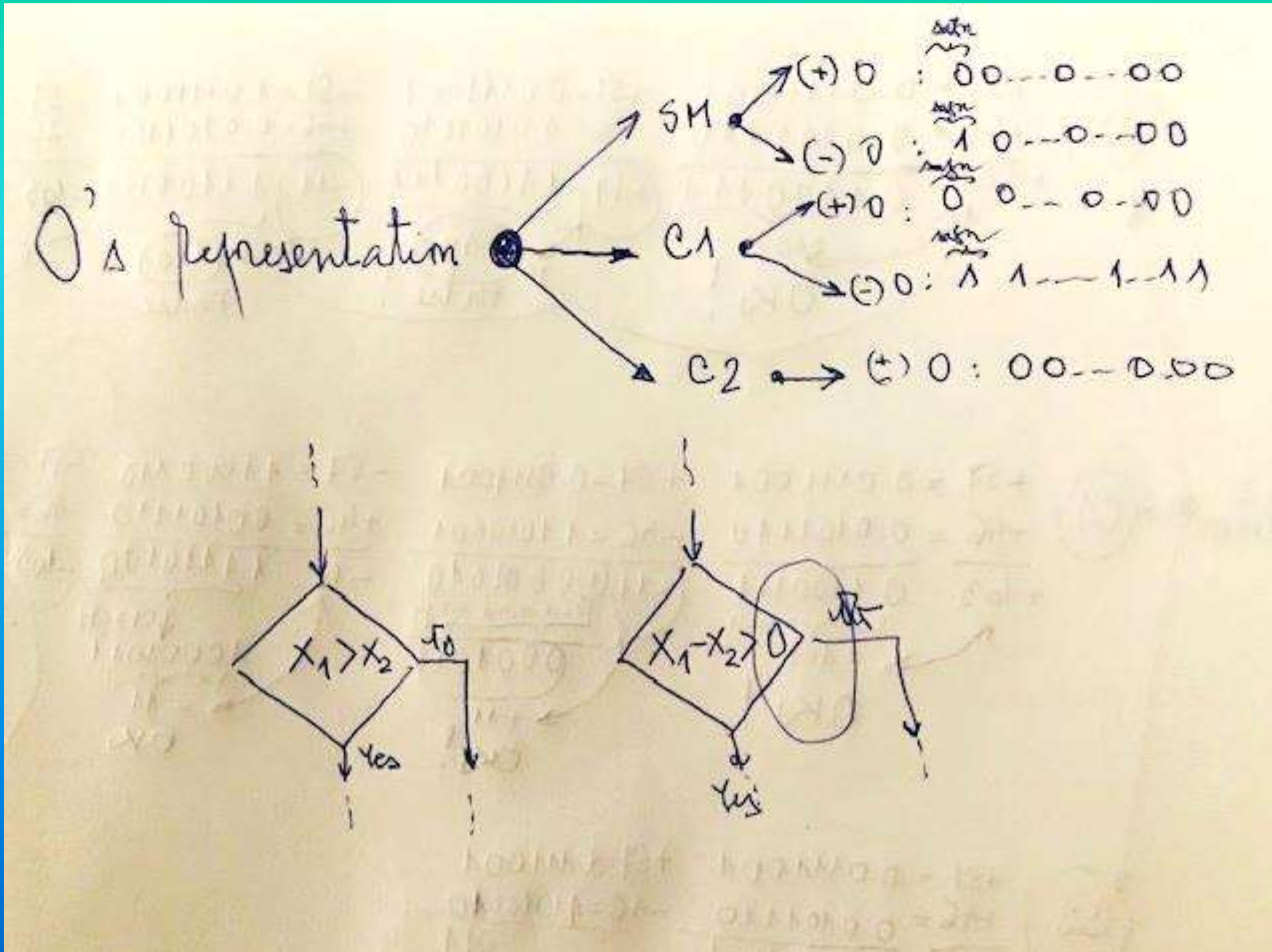
SM weights: sign $2^6$ … $2^1$ $2^0$; C1 weights: sign $2^6$ … $2^1$ $2^0$; C2 weights: sign $2^6$ … $2^1$ $2^0$

no weights

| | SM | C1 | C2 |
|---|---|---|---|
| $N_3 = +0,34575$ | 0 0 1 1 0 1 1 1 0 0 | 0 0 1 0 1 1 | 0 0 1 1 0 1 1 0 0 |
| $N_4 = -0,34575$ | 1 0 1 1 0 1 1 1 0 0 | 1 1 0 1 0 0 1 1 | 1 1 0 1 0 1 0 0 |

SM weights: sign $2^{-1}$ … $2^{-6}$ $2^{-7}$; C1 weights: sign $2^{-1}$ … $2^{-6}$ $2^{-7}$; C2 weights: sign $2^{-1}$ … $2^{-6}$ $2^{-7}$

no weights !


0's Representation
SM
(+) 0 : 00--0--00
(-) 0 : 1 0--0--00
C1
(+) 0 : 0 0--0--00
(-) 0 : 1 1--1--11
C2
(±) 0 : 00--0.00
$x_1 > x_2$
No
Yes
$x_1 - x_2 > 0$
Yes

$$\begin{array}{rl}
 & \overbrace{\phantom{0}}^{\text{sign}}\;\;2^2\;2^1\;2^0 \\
X=+3_{10} = & 0.\,0\;1\;1_{SM} \\
Y=+3_{10} = & 0.\,0\;1\;1_{SM} \quad + \\
\hline
Z = & 0.\,1\;1\;0_{SM} = +6_{10}
\end{array}$$

*a*

$$\begin{array}{rl}
 & \overbrace{\phantom{0}}^{\text{sign}}\;\;2^2\;2^1\;2^0 \\
X=-3_{10} = & 1.\,0\;1\;1_{SM} \\
Y=-3_{10} = & 1.\,0\;1\;1_{SM} \quad + \\
\hline
Z = & \cancel{1}0.\,1\;1\;0_{SM} = +6_{10}\;(!)
\end{array}$$

*b*

$$\begin{array}{rl}
 & \overbrace{\phantom{0}}^{\text{sign}}\;\;2^2\;2^1\;2^0 \\
X=+3_{10} = & 0.\,0\;1\;1_{SM} \\
Y=-3_{10} = & 1.\,0\;1\;1_{SM} \quad + \\
\hline
Z = & 1.\,1\;1\;0_{SM} = -6_{10}\;(!)
\end{array}$$

*c*

$$\begin{array}{rl}
 & \overbrace{\phantom{0}}^{\text{sign}}\;\;2^2\;2^1\;2^0 \\
X=+1_{10} = & 0.\,0\;0\;1_{SM} \\
Y=-6_{10} = & 1.\,1\;1\;0_{SM} \\
\hline
Z = & 1.\,1\;1\;1_{SM} = -7_{10}\;(!)
\end{array}$$

*d*

Amdahl's law: „Make the common case fast"

Decimal

$$\begin{array}{r} 7+ \\ 6 \\ \hline 13 \end{array}$$

1 carry, 3 ← sum

$10^1$ digit

$$\begin{array}{cccc} 0+ & 0+ & 1+ & 1+ \\ 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 10 \end{array}$$

$2^1$

1: carry bit, 0: sum bit

Binary

sign magnitude

SM)

$$\begin{array}{l} X = X_S\, X_M \; + \\ Y = Y_S\, Y_M \\ \hline Z = Z_S\, Z_M \end{array}$$

$$\begin{array}{l} X = 0\, X_M \; + \\ Y = 0\, Y_M \\ \hline Z = 0\, Z_M \end{array}$$

$Z_M = (X_M + Y_M)$

$X_M > Y_M$

$$\begin{array}{l} X = 0\, X_M \; + \\ Y = 1\, Y_M \\ \hline Z = 1\, Z_M \end{array}$$

$Z_M = (X_M + Y_M)$ ← false

$Z_M = (X_M - Y_M)$ ← correct

$X_M < Y_M$

$$\begin{array}{l} X = 1\, X_M \\ Y = 0\, Y_M \\ \hline Z = 1\, Z_M \end{array}$$

→ $Z_M = (X_M + Y_M)$

→ $Z_M = (Y_M - X_M)$

$$\begin{array}{l} X = 1\, X_M \\ Y = 1\, Y_M \\ \hline Z = 10\, Z_M \end{array}$$

$Z_M = (X_M + Y_M)$ false

$Z_M = 1(X_M + Y_M)$ correct

3 cases false result!

time penalty!

1 | 0 | 1 | ... | 1 |

$z_{n-2}$ $z_0$

sign

overflow emulation

C1

$$X_{C1} = \bar{X} = \begin{cases} \geq 0 \rightarrow & \overset{\text{sign}}{0}\; \overset{2^{n-2}}{x_{n-2}} \ldots \overset{2^{i}}{x_i} \ldots \overset{2^{1}}{x_1}\, \overset{2^{0}}{x_0} \\ & \bar{x}_i = 1 - x_i \quad \text{the 1 complement} \\ \leq 0 \rightarrow & 1\; \bar{x}_{n-2} \ldots \bar{x}_i \ldots \bar{x}_1\, \bar{x}_0 \end{cases}$$

negative numbers

$$X_{C1} = \overset{\text{sign}}{1}\; \overset{2^{n-2}}{1}\; \overset{2^{n-3}}{1} \ldots \overset{2^{i}}{1} \ldots \overset{2^{1}}{1}\; \overset{2^{0}}{1} - \underbrace{x_{n-2}\, x_{n-3} \ldots x_i \ldots x_1\, x_0}_{\text{magnitude } X_M}$$

$$\rightarrow 2^{n-1} + 2^{n-2} + \ldots + 2^{i} + \ldots + 2^{1} + 2^{0} =$$

$$= (2-1)(2^{n-1} + 2^{n-2} + \ldots + 2^{i} + 2^{1} + 2^{0}) = 2^{n} - 1$$

$$\rightarrow X_{C1} = 2^{n} - 1 - X_M$$

positive numbers $X = 0X_M \quad X_{SM} = X_{C1} = X_{C2}$

negative numbers $X_{C1} = \bar{X} = 1\,\bar{x}_{n-2}\,\bar{x}_{n-3}\cdots\bar{x}_i\cdots\bar{x}_1\,\bar{x}_0$

$\bar{x}_i = 1 - x_i$

$$
\begin{array}{ccccccccc}
2^{n-1} & 2^{n-2} & 2^{n-3} & & 2^{i} & & 2^{1} & 2^{0} & \\
1 & 1 & 1 & \cdots & 1 & \cdots & 1 & 1 & - \\
0 & x_{n-2} & x_{n-3} & \cdots & x_i & \cdots & x_1 & x_0 & \\
\hline
\end{array}
$$

$$1\times 2^{n-1} + 1\times 2^{n-2} + 1\times 2^{n-3} + \cdots + 1\times 2^{i} + \cdots + 1\times 2^{1} + 1\times 2^{0} =$$

$$= (2-1)\left(2^{n-1} + 2^{n-2} + 2^{n-3} + \cdots + 2^{i} + \cdots + 2 + 1\right) =$$

$$= 2^n - \cancel{2^{n-1}} + \cancel{2^{n-1}} - \cancel{2^{n-2}} + \cancel{2^{n-2}} - \cancel{2^{n-3}} + \cdots + \cancel{2^{i+1}} - \cancel{2^{i}} + \cdots - \cancel{2^{2}} + \cancel{2^{2}} - 2 + 2 - 1 =$$

$$= 2^n - 1$$

$$X_{C1} = \bar{X} = 2^n - 1 - X_M$$

Addition SM vs C1 (case ±, $X_M \geq Y_M$)

$$\begin{array}{l} X = +3 \; + \\ Y = -2 \\ \hline Z = +1 \end{array} \qquad \begin{array}{lr} X_{SM} = & 0011 \; + \\ Y_{SM} = & 1010 \\ \hline & 1101 \end{array}$$

$-5$ false

$X_M \geq Y_M$

(3) (2)

magnitude comparison!

$$\begin{array}{l} X = 0X_M \\ Y = 1Y_M \\ \hline Z = 0(X_M - Y_M) \end{array}$$

$+(3-2) = +1$

$$\begin{array}{lr} X_{C1} = X_{SM} = & 0011 \; + \\ Y_{C1} = & 1101 \\ \hline & 1\,0000 \end{array}$$

$0$ must be corrected!

must be corrected

$$Z = X_{C1} + Y_{C1} = X_{SM} + 2^n - 1 - Y_{SM} = 2^n + (X_M - Y_M) - 1$$

carry out

$$\begin{array}{r} 0011 \; + \\ 1101 \\ \hline 1\,0000 \; + \\ 1 \\ \hline 0001 \\ +1 \end{array}$$

end around correction

Addition SM vs C1 (case ±, $X_M \leq Y_M$)

$X = +2$

$\underline{Y = -3}$

$Z = -1$

$X_{SM} = 0010 +$

$\underline{Y_{SM} = 1011}$

$\underbrace{1101}_{-5 \text{ false}}$

$X = 0X_M$

$Y_M \geq X_M \quad \underline{Y = 1Y_M}$

$Z \Rightarrow 1(Y_M - X_M)$

$-(3-2) = -1$

$X_{C1} = X_{SM} = 0010 +$

$\underline{Y_{C1} = 1100}$

$1110$

Z in C1

$\Downarrow C1 \to SM$

$\underbrace{1001}_{-1}$ without correction

$Z = X_{C1} + Y_{C1} = X_M + 2^n - 1 - Y_M = \overbrace{2^n - 1 - (Y_M - X_M)}^{\text{C1 of}}$

Addition SM vs C1 (case =)

$X = -3$ $\quad X_{SM} = 1011 +$ $\quad X = 1\,X_M +$

$Y = -2$ $\quad Y_{SM} = 1010$ $\quad Y = 1\,Y_M$

$Z = -5$ $\quad 10101$ $\quad Z = 1\,(X_M + Y_M)$

carry out $\quad$ +5 false $\quad 1\,(3+2) = -5$

$X_{C1} = 1100 +$

$Y_{C1} = 1101$

$1\,1001$

Z in C1 $\quad$ C1→SM

$\underbrace{1110}_{-6}$ must be corrected

$1100$
$1101$
$1\;1001 +$
$1$
$1010$

Z in C1 $\quad$ C1 → SM

$\underbrace{1101}_{-5}$

- $Z = X_{C1} + Y_{C1} = 2^n - 1 - X_M + 2^n - 1 - Y_M =$

$= \underbrace{2^n}_{\text{carry out}} + \underbrace{2^n - 1 - (X_M + Y_M)}_{C_1 \text{ of}} - 1$

$$
\begin{array}{l}
\qquad\qquad\quad \overbrace{\ }^{\text{sign}}\ 2^2\ 2^1\ 2^0 \\
X=+3_{10} = 0.\ 0\ 1\ 1_{SM} = 0.0\ 1\ 1_{C1} \quad + \\
Y=-\ 4_{10} = 1.\ 1\ 0\ 0_{SM} = 1.0\ 1\ 1_{C1} \\
\hline
\qquad\qquad Z=1.1\ 1\ 0_{C1} = 1.001_{SM} = -1
\end{array}
$$

*a*

$$
\begin{array}{l}
\qquad\qquad\quad \overbrace{\ }^{\text{sign}}\ 2^2\ 2^1\ 2^0 \\
X=+4_{10} = 0.\ 1\ 0\ 0_{SM} = 0.1\ 0\ 0_{C1} \quad + \\
Y=-3_{10} = 1.\ 0\ 1\ 1_{SM} = 1.1\ 0\ 0_{C1} \\
\hline
\qquad\qquad 1\ 0.0\ 0\ 0 \qquad \text{end-around carry} \\
\qquad\qquad\qquad\quad 1 \\
\hline
\qquad\qquad Z=0.0\ 0\ 1_{C1} = 0.001_{SM} = +1_{10}
\end{array}
$$

*b*

$$
\begin{array}{l}
\qquad\qquad\quad \overbrace{\ }^{\text{sign}}\ 2^2\ 2^1\ 2^0 \\
X=-3_{10} = 1.\ 0\ 1\ 1_{SM} = 1.\ 1\ 0\ 0_{C1} \quad + \\
Y=-4_{10} = 1.\ 1\ 0\ 0_{SM} = 1.\ 0\ 1\ 1_{C1} \\
\hline
\qquad\qquad 1\ 0.\ 1\ 1\ 1 \qquad \text{end-around carry} \\
\qquad\qquad\qquad\quad 1 \\
\hline
\qquad\qquad Z=1.0\ 0\ 0_{C1} = 1.1\ 1\ 1_{SM} = -7_{10}
\end{array}
$$

*c*

negative numbers $X_{C2} = -X = X_{C1} + 1$ for integers

$$X_{C2} = 2^n - 1 - X_M + 1 = 2^n - X_M$$

$$X_{C2} = -X = X_{C1} + 000\ldots0\ldots01$$ for fractions

$1 \times 2^{-n+1}$

$$X_{C1} = \begin{array}{cccccccc} 2^0 & 2^{-1} & 2^{-2} & & 2^{-i} & & 2^{-n+2} & 2^{-n+1} \\ 1 & 1 & 1 & \ldots & 1 & \ldots & 1 & 1 \\ 0 & x_{n-2} & x_{n-3} & \ldots & x_i & \ldots & x_1 & x_0 \end{array}$$

$$\rightarrow 1\times 2^0 + 1\times 2^{-1} + 1\times 2^{-2} + \cdots + 1\times 2^{-i} + \cdots + 1\times 2^{-n+2} + 1\times 2^{-n+1} =$$

$$= 2^{-n+1}(2-1)\left(2^{n-1} + 2^{n-2} + 2^{n-3} + \cdots + 2^i + \cdots + 2 + 1\right) =$$

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

$$\rightarrow X_{C2} = 2 - \cancel{2^{-n+1}} - X_M + \cancel{2^{-n+1}} =$$

$$= 2 - X_M$$

↑ name of C2

$$X = +3$$
$$\underline{Y = -2}$$
$$Z = +1$$

$$X_{c2} = X_{c1} = X_{SM} = 0011 +$$
$$Y_{c2} = Y_{c1} + 1 = \underline{1110}$$
$$\cancel{1}\,\underbrace{0001}_{+1}$$

carry out

correct!

$$Z = X_{c2} + Y_{c2} = X_M + 2^n - Y_M = \cancel{2^n} + (\underset{3}{X_M} - \underset{2}{Y_M})$$

$$X = +2$$
$$\underline{Y = -3}$$
$$Z = -1$$

$$X_{c2} = X_{c1} = X_{SM} = 0010$$
$$Y_{c2} = Y_{c1} + 1 = \underline{1101}$$
$$1111$$

Z in C1

$$\Downarrow C2 \to C1$$
$$1110$$
$$\Downarrow C1 \to SM$$
$$\underbrace{1001}_{-1}$$

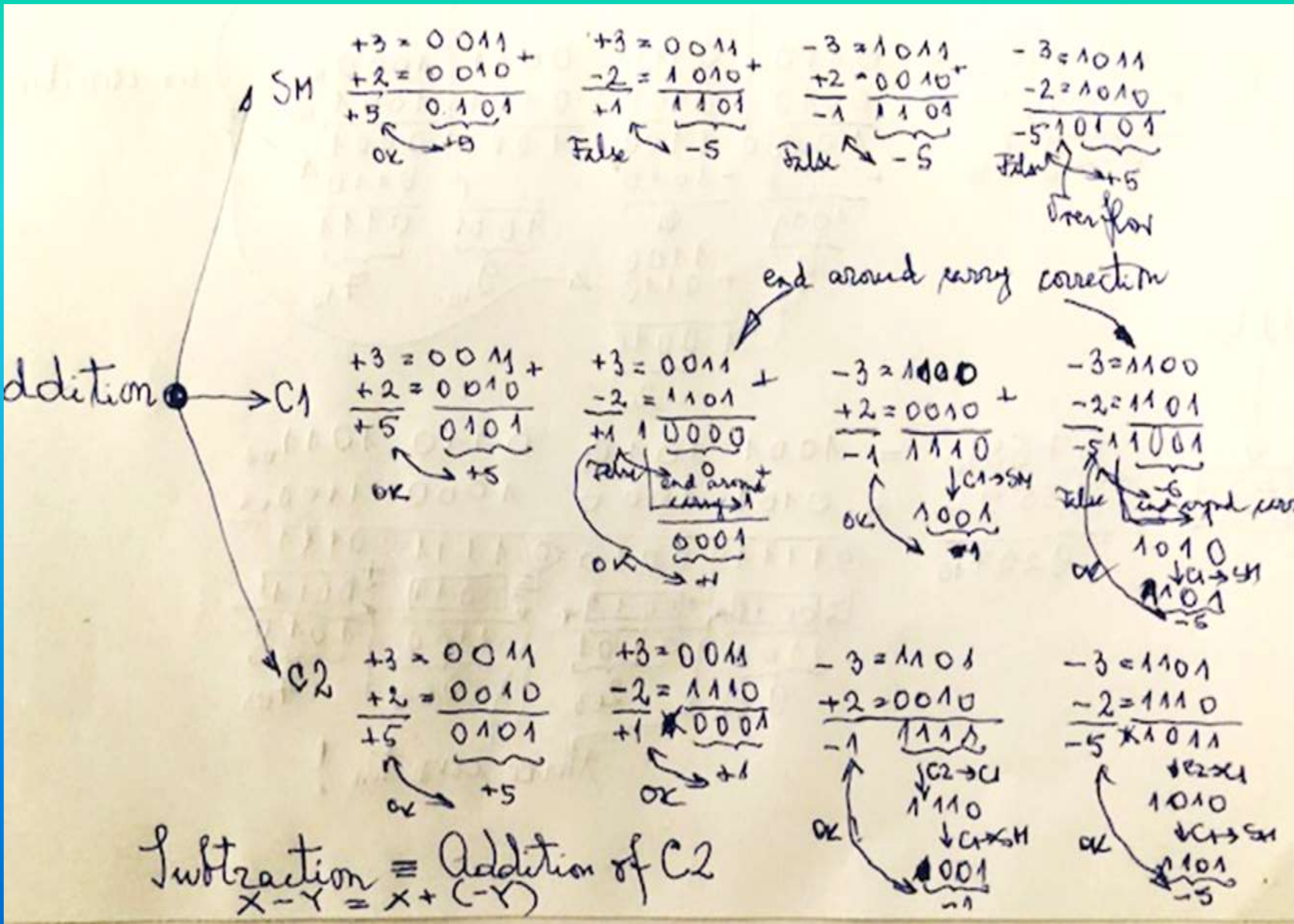
$$Z = X_M + 2^n - Y_M = \overbrace{2^n - (Y_M - X_M)}^{C_2}$$

$$X = -3$$
$$\underline{Y = -2}$$
$$Z = -5$$

$$X_{c2} = X_{c1} + 1 = 1101$$
$$Y_{c2} = Y_{c1} + 1 = \underline{1110}$$
$$\cancel{1}\,1011$$
$$\Downarrow C2 \to C1$$
$$1010$$
$$\Downarrow C1 \to SM$$
$$\underline{1101} \quad -5$$

$$Z = 2^n - X_M + 2^n - Y_M =$$
$$= 2^n + \underbrace{2^n - (X_M + Y_M)}_{C_2}$$

carry out


Addition
SM
+3 = 0011
+2 = 0010
+5 0101
ok +5
+3 = 0011
-2 = 1010
+1 1101
False -5
-3 = 1011
+2 = 0010
-1 1101
False -5
-3 = 1011
-2 = 1010
-5 10101
False +5
Overflow
end around carry correction
C1
+3 = 0011
+2 = 0010
+5 0101
ok +5
+3 = 0011
-2 = 1101
+1 1 0000
False
end around carry → 1
0001
ok +1
-3 = 1100
+2 = 0010
-1 1110
C1 → SM
1001
ok -1
-3 = 1100
-2 = 1101
-5 1 1001
-6
False
end around carry
1010
C1 → SM
1101
-5
ok
C2
+3 = 0011
+2 = 0010
+5 0101
ok +5
+3 = 0011
-2 = 1110
+1 1 0001
ok +1
-3 = 1101
+2 = 0010
-1 1111
C2 → C1
1110
C1 → SM
1001
-1
ok
-3 = 1101
-2 = 1110
-5 1 1011
C2 → C1
1010
C1 → SM
1101
-5
ok
Subtraction ≡ Addition of C2
X − Y = X + (−Y)

| Decimal number | Fixed-point binary codes | | |
|---|---|---|---|
| | SM | C1 | C2 |
| +7 | 0111 | 0111 | 0111 |
| +6 | 0110 | 0110 | 0110 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| +2 | 0010 | 0010 | 0010 |
| +1 | 0001 | 0001 | 0001 |
| (+)0 | 0000 | 0000 | 0000 |
| (-)0 | 1000 | 1111 | |
| -1 | 1001 | 1110 | 1111 |
| -2 | 1010 | 1101 | 1110 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| -6 | 1110 | 1001 | 1010 |
| -7 | 1111 | 1000 | 1001 |
| -8 | - | - | 1000 |

| | | SM | C1 | C2 |
|---|---|---|---|---|
| Comparison | Addition | 3 | 2 | 1 |
| | 0's representation | 2-3 | 2-3 | 1 |
| | Generation | 1 | 2 | 3 |

**Shifting**

Right shifting $\rightarrow$

SM |1|0|0|1|1|0|0|0|

$N = -24_{10}$

C2 |1|1|1|0|1|0|0|0|

$\rightarrow$ 1 position $\frac{N}{2} = 12_{10}$ $\rightarrow$ 2 position $\frac{N}{4} = 6$

|1|0|0|0|1|1|0|0|

|1|1|1|1|0|1|0|0|

left shifting $\leftarrow$

|1|0|0|1|1|0|0|0|

$N = -24_{10}$

|1|1|1|0|1|0|0|0|

$\rightarrow$ 1 position $2 \times N = 48_{10}$ $\rightarrow$ 2 position $4 \times N = 96$

|1|0|1|1|0|0|0|0|

|1|1|0|1|0|0|0|0|

# Six correction

## Case 1

$2^4 = 16 = 10 + 6$

| $10^1$ | $10^0$ | | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|
| | 7 | | | 0 | 1 | 1 | 1 |
| | 9 | | | 1 | 0 | 0 | 1 |
| 1 | $\boxed{6}$ | | 1 | 0 | 0 | 0 | 0 + |
| | | | | | 1 | 1 | 0 |
| | | | | $\boxed{0}$ | $\boxed{1}$ | $\boxed{1}$ | $\boxed{0}$ |

six correction

Decimal BCD

## Case 2

$0 = 10 - 10$

| $10^1$ | $10^0$ | | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|
| | 8 | | 1 | 0 | 0 | 0 |
| | 5 | | 0 | 1 | 0 | 1 |
| 1 | 3 | 0 | 1 | 1 | 0 | 1 − |
| | | | 1 | 0 | 1 | 0 |

$\equiv$

1101 +

0110 six correction

1 $\boxed{0011}$

Decimal

BCD

$10^1 = 10$

$0 < 7_{10} - 10$

$2^4 = 16 = 10 + 6$

**BCD**

$$
\begin{array}{r}
6\,7\,3\,8_{10} \\
+\;2\,5\,5\,9_{10} \\
\hline
9\,2\,9\,7_{10}
\end{array}
\quad
\begin{array}{rrrr}
0110 & 0111 & 0011 & 1000_{BCD} \\
0010 & 0101 & 0101 & 1001_{BCD} \\
\hline
1000 & 1100 & 1000 & 1\,0001 \\
+\;1 & -1010 & 1 & 0110 \\
\hline
1001 & \Downarrow & 1001 & 0111 \\
\underbrace{}_{9_{10}} & 1100 & \underbrace{}_{9_{10}} & \underbrace{}_{7_{10}} \\
 & +\;0110 & & \\
\hline
 & 1\;0010 & & \\
 & \underbrace{}_{2_{10}} & &
\end{array}
$$

six correction

Addition

**E3**

$$
\begin{array}{r}
6738_{10} \\
+\;2559_{10} \\
\hline
9297_{10}
\end{array}
\quad
\begin{array}{rrrr}
= 1001 & 1010 & 0110 & 1011_{E3} \\
= 0101 & 1000 & 1000 & 1100_{E3} \\
\hline
0\;1111 & 1\;0010 & 0\;1111 & 1\;0111 \\
-\;0011 & +\;0011 & -\;0011 & +\;0011 \\
\hline
1100 & 0101 & 1100 & 1010 \\
\underbrace{}_{9_{E3}} & \underbrace{}_{2_{E3}} & \underbrace{}_{9_{E3}} & \underbrace{}_{7_{E3}}
\end{array}
$$

three correction!

$x_S$ $\quad x_E \quad$ $x_M$

sign $\quad$ exponent $\quad$ mantissa

$$\boxed{X = (-1)^{\overset{\text{sign}}{x_S}} \cdot \underbrace{X_B}_{\text{base}}^{\overset{\text{exponent}}{x_E}} \cdot \underbrace{x_M}_{\text{mantissa}}}$$

Input

normalized floating point number

$\xrightarrow{SM}$ $\overset{\text{sign}}{0}, \overset{\text{magnitude}}{1 \ldots}$ — mantissa

$\xrightarrow{C2}$ $\overset{\text{sign}}{x_S}, x_{M\,msb} \ldots$ — mantissa

Mantissa problem

Computer

floating point number → unnormalized

floating point number → denormalized

normalized

Output floating point number

Error minimum $x_E$ – k bits

$\xrightarrow{SM}$ $-2^{k-1}+1$

$\xrightarrow{C2}$ $-2^{k-1}$

Exponent problem

Bias (Excess)

$\xrightarrow{SM}$ $2^{k-1}-1$

$\xrightarrow{C2}$ $2^{k-1}$

| Exponent bit pattern | Unsigned value | Signed value | |
|---|---|---|---|
| | | Bias = 127 | Bias = 128 |
| 11111111 | 255 | +128 | +127 |
| 11111110 | 254 | +127 | +126 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 10000001 | 129 | +2 | +1 |
| 10000000 | 128 | +1 | 0 |
| 01111111 | 127 | 0 | -1 |
| 01111110 | 126 | -1 | -2 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 00000001 | 1 | -126 | -127 |
| 00000000 | 0 | -127 | -128 |

# IEEE 754 Floating Point Standard

IEEE 754 → 32 bits, 64 bits, 80 bits

**32 bits**

$x_S$   $x_E$

0 1 2 3 4 5 6 7 8 9 … 31

sign; exponent: excess 127 sign-magnitude binary integer; fraction part of sign-magnitude binary significand with hidden integer bit

$$X = (-1)^{x_S} \cdot 2^{x_E - 127} \cdot (1.x_M)$$

hidden bit; significand

where $0 < x_E < 255$

**64 bits**

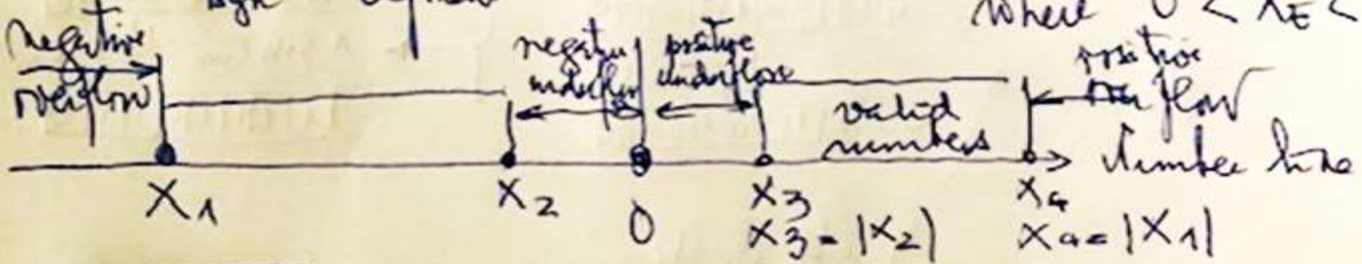
$x_S$   $x_E$

0 1 2 3 4 5 6 7 8 9 10 11 12 … 63

sign; exponent

$$X = (-1)^{x_S} \cdot 2^{x_E - 1023} \cdot (1.x_M)$$

where $0 < x_E < 2047$




| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | … | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|

$$X_1 = (-1)^1 \cdot 2^{254-127} \left(1 + \tfrac{1}{2} + \dots \tfrac{1}{2^{23}}\right) = -2^{127}\left(2 - 2^{-23}\right)$$

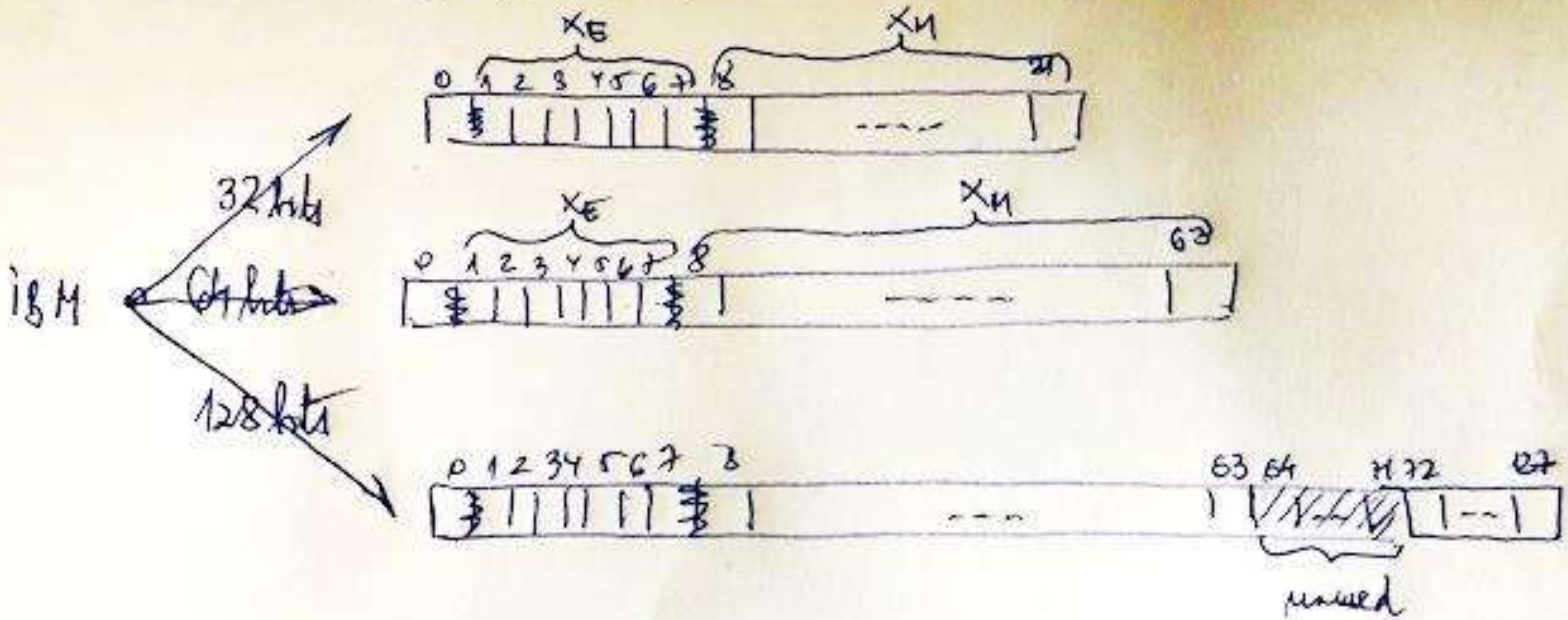
hidden bit

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | … | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|

$$X_2 = (-1) 2^{1-127} (1 + 0 \dots + 0) = -2^{-126}$$

# IBM Floating Point "Standard"

$$X = (-1)^{X_S} \cdot 16^{X_E - 64} \cdot (0.X_M)$$



$X = -794,08984375_{10} = -1100011010,00010111_2$

$X = (-1)^{X_S} \cdot 2^{X_E - 127} \cdot (1.X_M)$

| | |
|---|---|
| 794 | 2 |
| 0 | 397 |
| 1 | 198 |
| 0 | 99 |
| 1 | 49 |
| 1 | 24 |
| 0 | 12 |
| 0 | 6 |
| 0 | 3 |
| 1 | 1 |

0,08984375 × 2
0,17968750 × 2
0,35937500 × 2
0,71875000 × 2
1,43750000 × 2
0,87500000 × 2
1,75000000 × 2
1,50000000 × 2
1,00000000

$X = -1,1000110100001 0111 \times 2^9$

$X_E - 127 = 9 \Rightarrow X_E = 136$

| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

C 4 4 6 8 5 C 0

$X_{32} = C44685C0_{16}$

$X = -794{,}08984375_{10} = -1100011010{,}00010111_2 =$

$= -0{,}0011\,0001\,1010\,0001\,0111 \times 16^3$

$X = (-1)^{x_S} \cdot 16^{x_E - 64} (0.x_M)$ $\quad x_E - 64 = 3 \Rightarrow x_E = 67$

$\underbrace{1100}_{C}\,\underbrace{0011}_{3}\,\underbrace{0011}_{3}\,\underbrace{0001}_{1}\,\underbrace{1010}_{A}\,\underbrace{0001}_{1}\,\underbrace{0111}_{7}\,\underbrace{0000}_{0}$

$X_{IBM} = C331A170_{16}$

$X_{754} = 45ABD070_{16}$

$0\,\underbrace{1000\,1011}_{139 = x_E}\,0101\,0111\,1010\,0000\,1110\,000$

$139 = x_E \Rightarrow x_E - 127 = 139 - 127 = 12$

$X_{754} = (-1)^0 \cdot 2^{12} \cdot 1{,}0101\,0111\,1010\,0001\,111 =$

hidden bit

$= X_{754} = 1010101111010{,}0001111_2 =$

$= +\left(4096 + 1024 + 256 + 64 + 32 + 16 + 8 + 2 + \frac{15}{128}\right) =$

$= +5498{,}1171875$

- Information
  - Intructions
  - Data
    - Numbers
      - Fixed Point Numbers
        - Binary
          - SM (Sign Magnitude) Code
          - C1 (One's Complement) Code
          - C2 (Two's Complement) Code
        - Decimal
          - BCD (Binary Coded Decimal) Code
          - E3 (Three Excess) Code
          - 2 out-of 5 (Two-out of Five) Code
      - Floating Point Numbers
        - Binary
        - Decimal
    - Nonnumerical Data
      - ASCII (American Standard Code for Information Interchange)
      - EBCDIC (Extended Binary Coded Decimal Interchange Code)
      - Unicode Standard

* C2 → SM $\xrightarrow{\text{complement}}$ C1 $\xrightarrow{+1}$ C2

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| sign | magnitude | | | | | | |

$N = -88_{10}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| sign | | | | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| sign | | | | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

James Robertson
$$\begin{cases} X_{C2} = -x_{n-1} \cdot 2^{n-1} + \sum_{i=0}^{n-2} x_i' \cdot 2^i & \text{for } X \text{ integer} \\ X_{C2} = -x_{n-1} \cdot 2^0 + \sum_{i=1}^{n-1} x_{n-i-1}' \times 2^{-i} & \text{for } X \text{ fractional} \end{cases}$$

where the bits $x_i'$ and $x_{n-1-i}'$ coincide with $x_i$ and $x_{n-1-i}$ in case of positive numbers ($x_{n-1} = 0$) and correspond to the bits of C2 code of X in case of negative numbers ($x_{n-1} = 1$)

$N = -88_{10}$ (C2)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

$$\rightarrow X_{C2} = -1 \times 2^7 + \sum_{i=0}^{6} x_i' \cdot 2^i = 1 \times 2^7 + (1 \times 2^5 + 1 \times 2^3) =$$
$$= -128 + 32 + 8 = -88$$

* C2's anomaly

| Decimal number | Fixed-point binary codes | | |
|---|---|---|---|
| | SM | C1 | C2 |
| +7 | 0111 | 0111 | 0111 |
| +6 | 0110 | 0110 | 0110 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| +2 | 0010 | 0010 | 0010 |
| +1 | 0001 | 0001 | 0001 |
| (+)0 | 0000 | 0000 | 0000 |
| (-)0 | 1000 | 1111 | |
| -1 | 1001 | 1110 | 1111 |
| -2 | 1010 | 1101 | 1110 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| -6 | 1110 | 1001 | 1010 |
| -7 | 1111 | 1000 | 1001 |
| -8 | - | - | 1000 |

SM

C2

| | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| $x_7$ | $x_6$ | $x_5$ | $x_4$ | $x_3$ | $x_2$ | $x_1$ | $x_0$ |

$N = -0{,}6875_{10} \rightarrow$ [SM] $\rightarrow$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

$$X_{C2} = -1 \times 2^0 + \sum_{i=1}^{7} x_{7-i} \times 2^{-i} = -1 + (x_6 \times 2^{-1} + x_5 \cdot 2^{-2} + x_4 \times 2^{-3} + x_3 \times 2^{-4} +$$

$$+ x_2 \times 2^{-5} + x_1 \times 2^{-6} + x_0 \times 2^{-7}) = -1 + (2^{-2} + 2^{-4}) =$$

$$= -1 + \frac{1}{4} + \frac{1}{16} = \frac{-16+4+1}{16} = -\frac{11}{16} = -0{,}6875_{10}$$

Sign

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0/1 | | | ... | | | ... | |
| | $x_{n-2}$ | $x_{n-3}$ | | $x_i$ | | $x_1$ | $x_0$ |

SM $\quad X \in [(-2^{n-1}+1), (+2^{n-1}-1)]$

C2 $\quad X \in [(-2^{n-1}), (+2^{n-1}-1)]$

* $n$ bits $\rightarrow 2^n$ unsigned binary numbers

$n$ bits $\rightarrow 10^{\frac{n}{4}}$ unsigned decimal numbers (BCD, E3)

$10^3 = 1000 \cong 1024 = 2^{10}$

$$\left.\begin{array}{l} 3 \ \text{----} \ 10 \\ \frac{n}{4} \ \text{----} \ x \end{array}\right\} \Rightarrow x = \frac{\frac{n}{4} \times 10}{3} = \frac{5}{6} n \cong 0{,}83n$$

$\Rightarrow n$ bits $\rightarrow \approx 2^{0{,}83n}$ unsigned decimal numbers (BCD, E3)

* $n$ bits $\rightarrow 2^n$ unsigned binary numbers

$n$ bits $\rightarrow 10^{\frac{n}{5}}$ unsigned 2-out-of-5 decimal numbers

$$10^3 = 1000 \cong 1024 = 2^{10} \rightarrow \left.\begin{matrix} 3 \text{ --- } 10 \\ \frac{n}{5} \text{ --- } x \end{matrix}\right\} \Rightarrow x = \frac{\frac{n}{5} \times 10}{3} = \frac{2}{3} n \cong 0,66n$$

$\Rightarrow$ $n$ bits $\rightarrow \approx 2^{0,66n}$ unsigned 2-out-of-5 decimal numbers

## Binary floating point numbers

Number notations

- weighted notation: $N = \sum_{i=0}^{n-1} x_i \cdot r^i$ where $0 \le x_i < r$
- scientific notation: $N = M \cdot B^E$

$E$ ← exponent, $B$ ← base, $M$ ← mantissa

SM integer

sign | magnitude

| $n-1$ | $n-2$ | $n-3$ | ... | $i$ | ... | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | $2^{n-2}$ | $2^{n-3}$ | | $2^i$ | | $2^1$ | $2^0$ |

$$X = (-1)^{x_{n-1}} \sum_{i=0}^{n-2} x_i 2^i$$

n bits word

| $n-1$ | $n-2$ | $n-3$ | ... | $i$ | ... | 1 | 0 |
|---|---|---|---|---|---|---|---|

$$n = e + m + 1$$

sign | exponent $X_E$ | mantisse $X_M$

| $n-1$ | $e-1$ | ... | 0 | $m$ | 1 | ... | 0 |
|---|---|---|---|---|---|---|---|
| | $2^{e-1}$ | ... | $2^0$ | $2^{-1}$ | $2^{-2}$ | | $2^{-m+1}$ |

$\underbrace{2^{e-1} \ldots 2^0}_{\text{e bits integer}}$ $\underbrace{2^{-1}\, 2^{-2} \ldots 2^{-m+1}}_{\text{m bits fractional}}$

↑ rising precision ↓ falling range | ↑ rising range ↓ falling precision

$$X = (-1)^{x_{n-1}} \cdot X_M \cdot 2^{X_E}$$

usual

general $2^a$

$a = 1, \cancel{2}, \cancel{3}, 4, \ldots$

IBM

Ex:

$X = 78{,}625_{10} = 1001110{,}101_2 = 1001110{,}101$

$Y = 0{,}171875_{10} = 0{,}001011_2 = 0000000{,}001011$ +

$Z = 78{,}796875_{10}$

$\underbrace{1001110{,}110011}_{78{,}796875}$

$X = 1001110{,}101 = 0{,}10011 10101 \times 2^7$

sign sign

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

$Y = 0{,}001011 = 0{,}1011 \times 2^{-2}$

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Input

Decimal to binary conversion & Normalization

normalized number

unpack

Mantissa Problem

Computer

unnormalized number

denormalized number

Normalization

pack

Binary to decimal

Output conversion

Normalization

Point leftshift → Increasing exponent

Point Rightshift → Decreasing exponent

$X+Y = (X_M + Y_M^{Y_E - X_E}) X^E$ where $X_E \geq Y_E$

mantisse

Floating Point Addition

1. Exponent comparison

Ex. $7 :: (-2) \Rightarrow$

$X_E - Y_E = 7 - (-2) = 9$

2. Rightshift for alignment with $(X_E - Y_E)$ positions

Ex.

$$\begin{array}{r} 0{,}10011 10101 \\ + \; 0{,}000000000 1011 \\ \hline 0{,}10011101 10011 \end{array}$$

3. Addition

4. Normalization

$0{,}10011101{,}0011 \times 2^7$

Normalization rules → Mantissa

SM: $\underbrace{0/1 \; . \; \text{-----}}_{x_E}\;\underbrace{. \; 1 \; \text{---} \; 1}_{x_M}$

for normalization, point leftshift ⇒ increasing exponent or point rightshift & decreasing exponent

C2: $0/1 \; . \; \text{---} \; . \; 1/0 \; \text{---} \; 1$

for normalization, point leftshift ⇒ increasing exponent or point rightshift & decreasing exponent, but WARNING! for negative mantisse rightshift introducing 1s!!

* Observation:
- through normalization operation
$\frac{1}{2} \leq |X_M| < 1$ absolute value

Exponent Problem

$X = +0{,}2_{10} = +0{,}0010001,\ldots_2$

$0{,}2 \times 2$
$0{,}4 \times 2$
$0{,}8 \times 2$
$1{,}6 \times 2$
$0{,}2 \times 2$
$0{,}4 \times 2$
$0{,}8 \times 2$
$1{,}6 \times 2$
$0{,}2$

$X'$: $\boxed{0 . 0 \,|\, 0 \,|\, 1 \,|\, 0 \,|\, 0 \,|\, 1 \,|\, 0}$

$X' = 2^{-3} + 2^{-6} = \frac{9}{64} = 0{,}140625_{10}$

$\Sigma_{error} = 2 - 0{,}140625 = 0{,}059375_{10}$

* from conversion algorithm → truncation errors

Errors → Truncation errors

Errors → Rounding errors → Rounding process ⇒ Time penalty!

* By floating point operation → Partial result 0 mante…

$Z = (-1)^{x_s - 1} \cdot X_M \cdot 2^{X_E}$ → significant erro…

$X_E$: large

$X_M$: very small

because errors

$X_E$ smallest number

S1: $-2^{e-1}+1$

C2: $-2^{e-1}$

C2's anomaly!

$X_S$ $X_E$ $X_M$

| 1 | 1 | 1 | 1 … 1 | 1 | 1 … 1 |

sign, e bits

$X_S$ $X_E$ $X_M$

| 1 | 1 | 0 | 0 … 0 | 0 | … 1 |

e bits

* But, for comparisons reasons (implementing of jumps or branch instructions), we want

0 representation as $\boxed{0/1 \,|\, \overbrace{0 \,|\, \cdots \,|\, 0}^{X_E} \,|\, \overbrace{0 \,|\, \cdots \,|\, 0}^{X_M}}$

and we have
- SM → $\boxed{0/1 \,|\, 1 \,|\, 1 \,|\, \cdots \,|\, 1 \,|\, 1 \,|\, 0 \,|\, \cdots \,|\, 0}$
- → $\boxed{0/1 \,|\, 1 \,|\, 0 \,|\, \cdots \,|\, 0 \,|\, 0 \,|\, 0 \,|\, \cdots \,|\, 0}$

→ Excess (Biased) Representation
- SM → Excess (Bias) $= 2^{e-1} - 1$
- E2 → Excess (Bias) $= 2^{e-1}$

→ Exponent from signed binary integer number $\xrightarrow{\text{through biased representation}}$ unsigned binary number

in excess to ↗ SM

| Exponent bit pattern | Unsigned value | Signed value | |
|---|---|---|---|
| | | Bias = 127 | Bias = 128 |
| 11111111 | 255 | +128 | +127 |
| 11111110 | 254 | +127 | +126 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 10000001 | 129 | +2 | +1 |
| 10000000 | 128 | +1 | 0 |
| 01111111 | 127 | 0 | -1 |
| 01111110 | 126 | -1 | -2 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 00000001 | 1 | -126 | -127 |
| 00000000 | 0 | -127 | -128 |

i.e. with the bias equal to $2^{8-1} - 1 = 127$,

... there are also certain char-

* Observation: for SM mantissas


→ $\underbrace{1.x_M}_{\text{significand}}$

hiding → gain precision

$$1 \le \underbrace{1.x_M}_{\text{significand}} < 2$$

Standards
IEEE
(Institute of Electrical and Electronics Engineers)

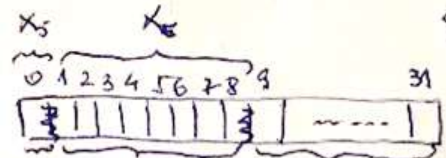
→ IEEE 754 Floating Point Standard
(Zuse, 1985 Kahan, 1987 → 854,
2008, Zuras & Cowlishaw, revision 2019 -
Hough & Cowlishaw)

mainframes

→ IBM Floating Point "Standard"

# IEEE 754 Floating Point Standard

32 bits




sign — exponent excess 127 sign-magnitude binary integer — fraction part of sign-magnitude binary significand with hidden integer bit

$$X = (-1)^{X_S} \cdot 2^{X_E - 127} \cdot (1.X_M)$$

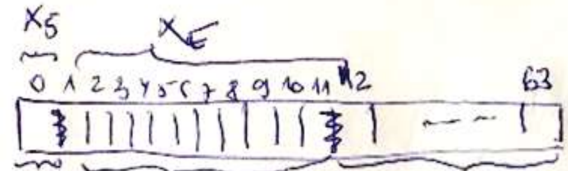
hidden bit

where $0 < X_E < 255$

significand

IEEE 754

64 bits

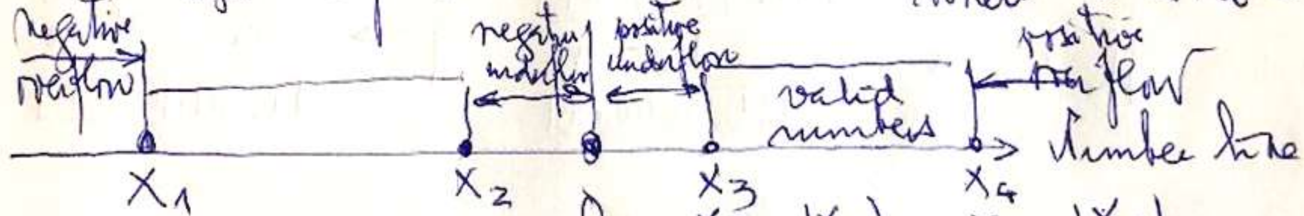



sign — exponent

$$X = (-1)^{X_S} \cdot 2^{X_E - 1023} \cdot (1.X_M)$$

where $0 < X_E < 2047$

80 bits




$X_3 = |X_2|$ $\quad$ $X_4 = |X_1|$

| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | ... | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|

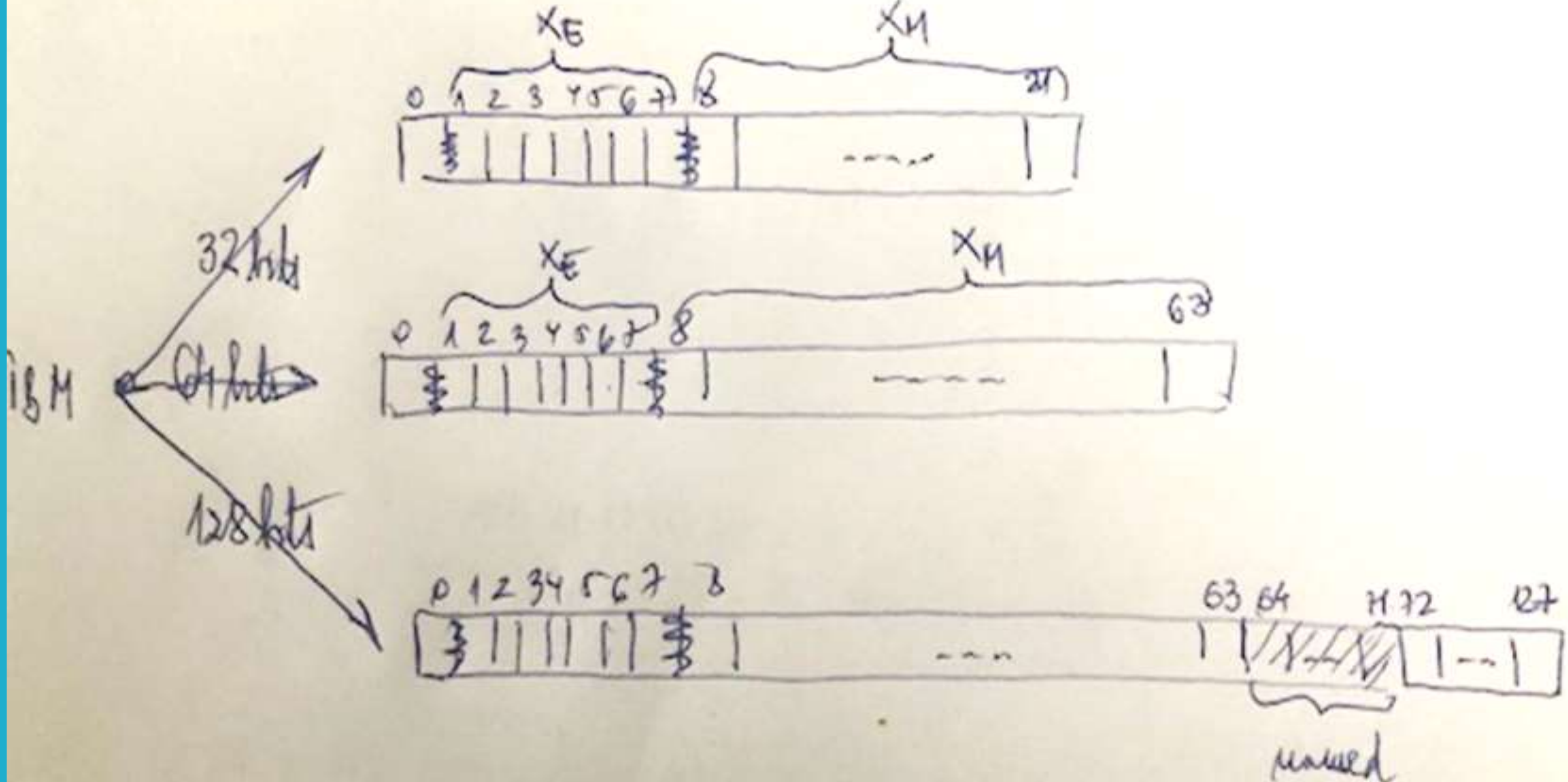
$$X_1 = (-1)^1 \cdot 2^{254-127} \underbrace{(1}_{\text{hidden bit}} + \frac{1}{2} + \dots \frac{1}{2^{23}}) = -2^{127}(2 - 2^{-23})$$

| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ... | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

$$X_2 = (-1) 2^{1-127} (1 + 0 \dots + 0) = $$
$$= -2^{-126}$$

# IBM Floating Point „Standard"

$$X = (-1)^{X_S} \cdot 16^{X_E - 64} \cdot (0.X_M)$$



$$X = -794{,}08984375_{10} = -1100011010{,}00010111_2$$

$$X = (-1)^{x_S} \cdot 2^{x_E - 127} \cdot (1.x_M)$$

| | | |
|---|---|---|
| 794 | 2 | |
| 0 | 397 | 2 |
| 1 | 198 | 2 |
| 0 | 99 | 2 |
| 1 | 49 | 2 |
| 1 | 24 | 2 |
| 0 | 12 | 2 |
| 0 | 6 | 2 |
| 0 | 3 | 2 |
| 1 | 1 | |

$0{,}08984375 \times 2$
$0{,}17968750 \times 2$
$0{,}35937500 \times 2$
$0{,}71875000 \times 2$
$1{,}43750000 \times 2$
$0{,}87500000 \times 2$
$1{,}75000000 \times 2$
$1{,}50000000 \times 2$
$1{,}00000000$

$$X = -1{,}100011010\,00010111 \times 2^9$$

$$X_E - 127 = 9 \Rightarrow X_E = 136$$

$$\underbrace{1\,1\,0\,0}_{C}\;\underbrace{0\,1\,0\,0}_{4}\;\underbrace{0\,1\,0\,0}_{4}\;\underbrace{0\,1\,1\,0}_{6}\;\underbrace{1\,0\,0\,0}_{8}\;\underbrace{0\,1\,0\,1}_{5}\;\underbrace{1\,1\,0\,0}_{C}\;\underbrace{0\,0\,0\,0}_{0}$$

$$X_{754} = C44685C0_{16}$$

$$X = -794{,}0898437 5_{10} = -1100011010{,}00010111_2 =$$

$$= -0{,}001100011010000010111 \times 16^3$$

$$X = (-1)^{X_S} \cdot 16^{X_E - 64} (0.X_M) \qquad X_E - 64 = 3 \Rightarrow X_E = 67$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1100 | 0011 | 0011 | 0001 | 1010 | 0001 | 0111 | 0000 |
| C | 3 | 3 | 1 | A | 1 | 7 | 0 |

$$X_{IBM} = C331A170_{16}$$

$$X_{754} = 45ABD070_{16}$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0100 | 0101 | 1010 | 1011 | 1101 | 0000 | 1111 | 0000 |

$$139 = X_E \Rightarrow X_E - 127 = 139 - 127 = 12$$

$$X_{754} = (-1)^0 \cdot 2^{12} \cdot 1{,}01010111101000011111 =$$

hidden bit

$$= \cancel{2^{12}\cdots} X_{754} = 1010101111010{,}0001111_2 =$$

$$= +\left(4096 + 1024 + 256 + 64 + 32 + 16 + 8 + 2 + \frac{15}{128}\right) =$$

$$= +5498{,}\cancel{1171875}\ 1171875_{10}$$

$X_{IBM} = 454BD0F0_{16}$

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

$69 = X_E \Rightarrow X_E - 64 = 5$

$X_{IBM} = (-1)^0 \cdot 16^5 \cdot 0,1010\,1011\,1101\,0000\,1111 =$

$= +1010\,1011\,1101\,0000\,1111 =$

$= +(524\,288 + 131\,072 + 32\,768 + 8192 + 4096 + 2048 + 1024 +$

$+ 256 + 8 + 4 + 2 + 1) = +703759_{10}$

# Exceptions

$x_S$ $x_E$ (255) (2047) $x_M \neq 0$ Not a number (NaN)

→ divide by zero
→ square root of a negative number

$x_S$ $x_E$ (255) (2047) $x_M$

$-\infty$ $+\infty$

negative overflow | valid numbers | negative underflow | positive underflow | valid numbers | positive overflow

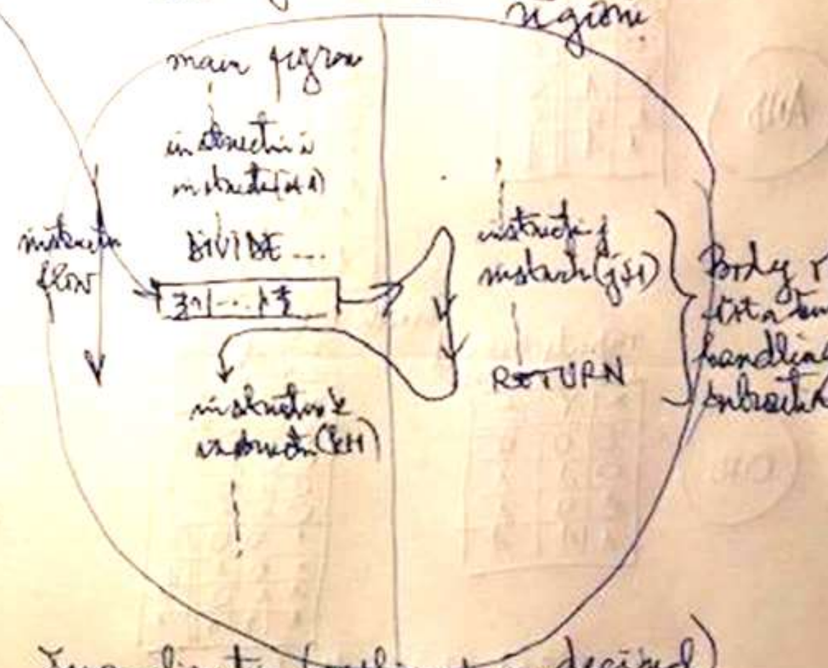
$x_1$ $x_2$ 0 $x_3$ $x_4$

$x_S$ $x_E$ (0) $x_M$ Denormalized number

$$X_{den} = (-1)^{x_S} \cdot 2^{-126} (0.x_M)$$

$x_S$ $x_E$ (0) $x_M$

$$X = (-1)^{x_S} 0$$




Denormalization (explained in decimal)

p = 4 normalized

$X_{min} = 0{,}1234$

$\frac{X_{min}}{10} = 0{,}0123$

$\frac{X_{min}}{100} = 0{,}0012$

$\frac{X_{min}}{1000} = 0{,}0001$

$\frac{X_{min}}{10000} = 0$

# Logic functions

**AND**

Truth table (inputs, output)

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

(inputs, output)

| w | x | y | z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Symbol

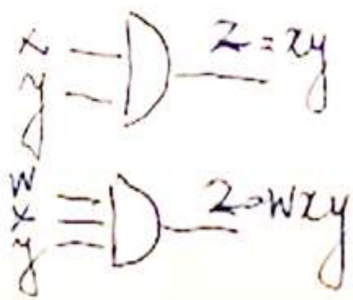
logic product

$Z = x \cdot y = xy =$

$= x$ and $y$

$Z = wxy$

Logic gate




**OR**

Truth table (inputs, output)

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

(inputs, output)

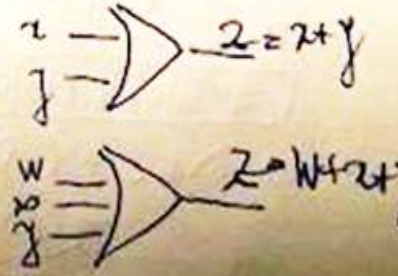
| w | x | y | z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Symbol

or logic sum

$Z = x + y = x$ or $y$

$Z = w + x + y$




**NOT**

Input output Truth table

| x | z |
|---|---|
| 0 | 1 |
| 1 | 0 |

Symbol

$Z = \overline{X}$

(complement)

## NAND

input / output

| x | y | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

input / output

| w | x | y | z |
|---|---|---|---|
| 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 |

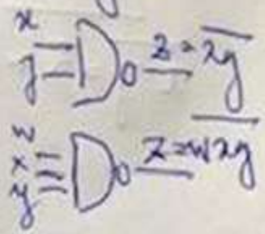
Truth table

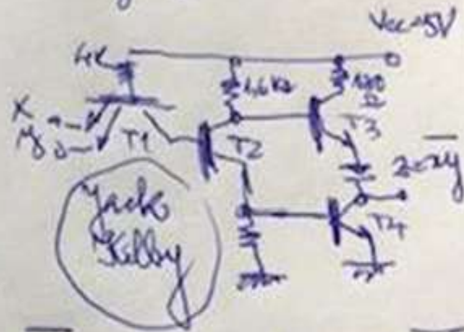
Symbol

$Z = \overline{x \cdot y} = \overline{xy} = x \text{ nand } y$

$Z = \overline{wxy}$

Logic gate




## NOR

input / output

| x | y | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

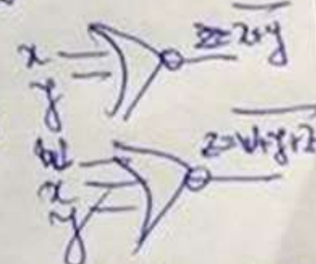
input / output

| W | X | Y | Z |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Symbol

$Z = \overline{x+y} = x \text{ nor } y$

$Z = \overline{w+y+x} = w \text{ nor } y \text{ nor } x$




## EXCLUSIVE OR

input / output

Truth table

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

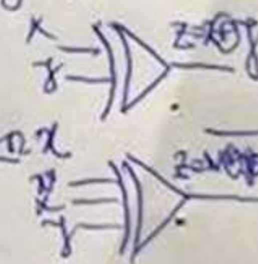
inputs / output

| w | x | y | z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Symbol

$Z = x \oplus y = x \text{ ex-or } y$

$Z = w \oplus y \oplus z = w \text{ ex-or } y \text{ ex-or } x$



Truth table — Symbol — Logic Gate

EXCLUSIVE NOR COINCIDENCE

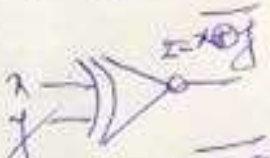
inputs / output

| $x$ | $y$ | $z$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

inputs / output

| $w$ | $x$ | $y$ | $z$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

$Z = \overline{x \oplus y} = x$ XNOR $y$




$Z = \overline{w \oplus x \oplus y} =$
$= w$ XNOR $x$ XNOR $y$


## Laws and Postulates of Boole's Algebra

| AND | OR | |
|---|---|---|
| $x \cdot 0 = 0$ | $x + 0 = x$ | 0's laws |
| $x \cdot 1 = x$ | $x + 1 = 1$ | 1's laws |
| $x \cdot \overline{x} = 0$ | $x + \overline{x} = 1$ | Negation laws |
| $\overline{\overline{x}} = x$ | | Double negation law |
| $x \cdot x = x$ | $x + x = x$ | Idempotence laws |
| $xy = yx$ | $x + y = y + x$ | Complementarity laws |

| AND | OR | |
|---|---|---|
| $x(yz) = (xy)z = (xz)y$ | $x + (y+z) = (x+y) + z = (x+z) + y$ | Associativity laws |
| $x \cdot (y+z) = xy + xz$ | $x + yz = (x+y)(x+z)$ | Distributivity laws |
| $x \cdot (x+y) = x$ | $x + xy = x$ | Absorption laws |
| $x \cdot (\overline{x}+y) = xy$ | $x + \overline{x}y = x + y$ | Laws of common identities |

AND OR

$\overline{x \cdot y} = \overline{x} + \overline{y}$ $\overline{x + y} = \overline{x} \cdot \overline{y}$ De Morgan's laws

Proofs

Distributivity OR law

| Input | | | | M1 | | | M2 |
|---|---|---|---|---|---|---|---|
| $x$ | $y$ | $z$ | $yz$ | $x+yz$ | $x+y$ | $x+z$ | $(x+y)\cdot(x+z)$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$\equiv$

qed

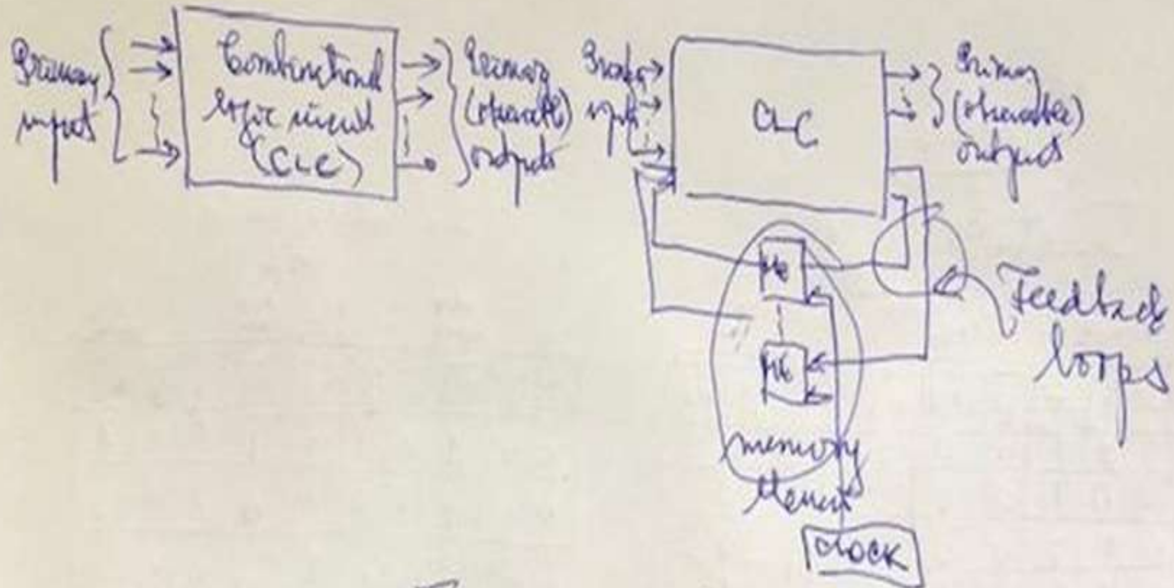
De Morgan's laws

| | | M1 | | | M2 | M1 | | | M2 |
|---|---|---|---|---|---|---|---|---|---|
| $x$ | $y$ | $\overline{x \cdot y}$ | $\overline{x}$ | $\overline{y}$ | $\overline{x}+\overline{y}$ | $\overline{x+y}$ | $\overline{x}$ | $\overline{y}$ | $\overline{x}\cdot\overline{y}$ |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$\equiv$ $\equiv$

qed qed

# Combinational logic circuits vs Sequential logic circuit



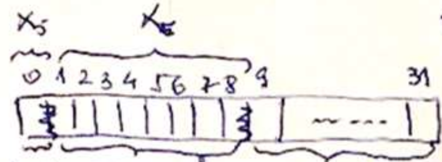

## Design of Combinational logic circuits

Problem statement → Truth table constructed → Logic equations written → Logic equations minimization → Logic circuits drawn → Logic circuits built

# IEEE 754 Floating Point Standard

IEEE 754

32 bits



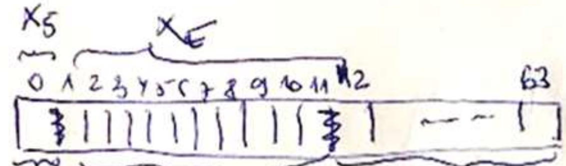

$$X = (-1)^{X_S} \cdot 2^{X_E-127} \cdot (1.X_M)$$

hidden bit

significand

where $0 < X_E < 255$

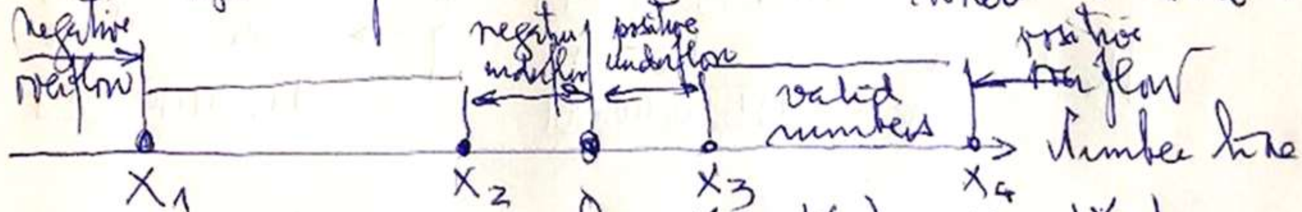
64 bits




$$X = (-1)^{X_S} \cdot 2^{X_E-1023} \cdot (1.X_M)$$

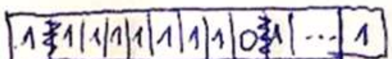
where $0 < X_E < 2047$

80 bits

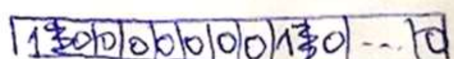



$X_3 = |X_2|$

$X_4 = |X_1|$



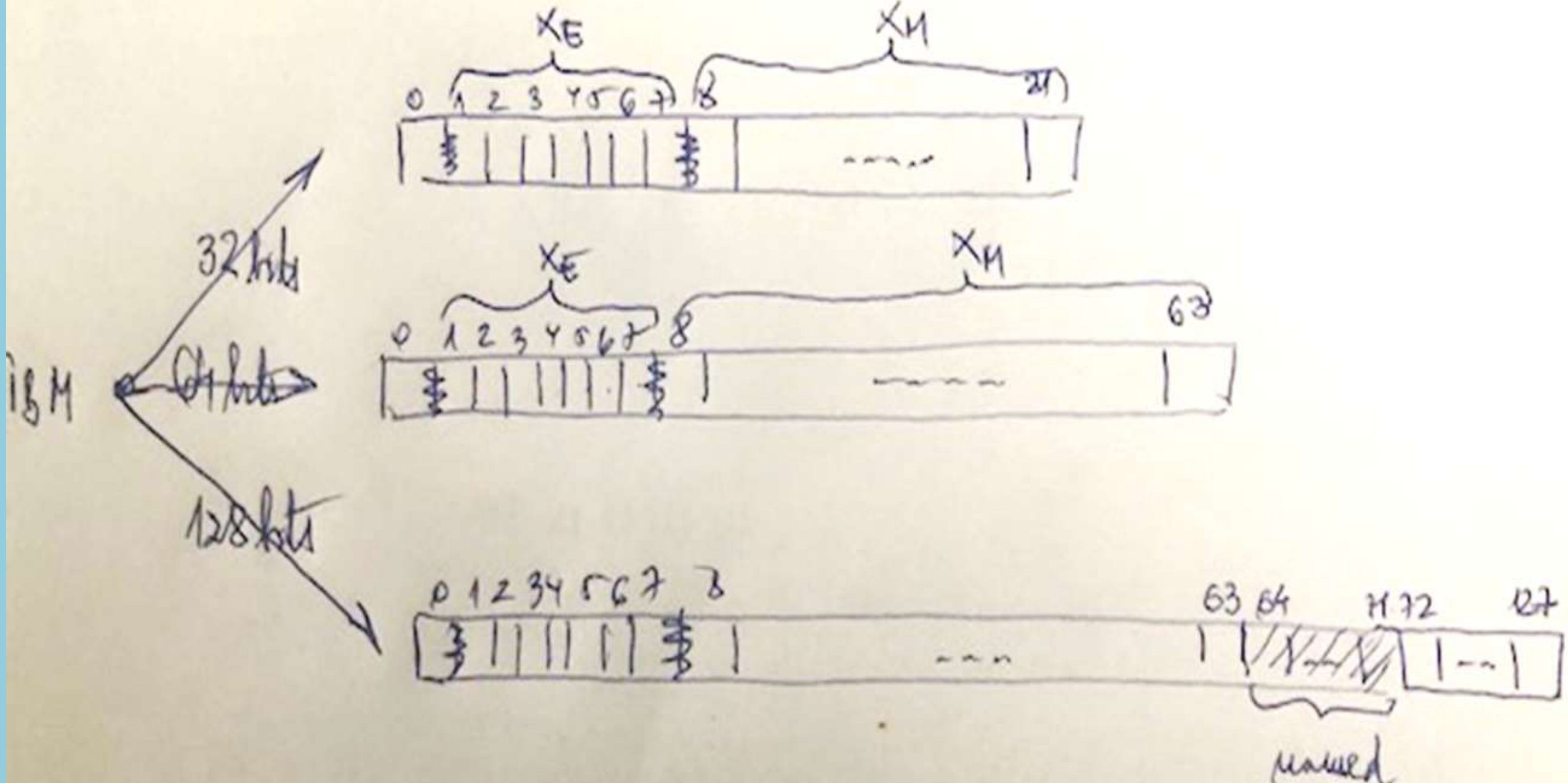

$$X_1 = (-1)^1 \cdot 2^{254-127} \underbrace{(1+\tfrac{1}{2}+\dots\tfrac{1}{2^{23}})}_{\text{hidden bit}} \approx -2^{127}(2-2^{-23})$$




$$X_2 = (-1)\, 2^{1-127}(1+0\dots+0) = -2^{-126}$$

# IBM Floating Point "Standard"

$$X = (-1)^{X_S} \cdot 16^{X_E - 64} \cdot (0.X_M)$$



$$X = -794{,}08984375_{10} = -1100011010{,}00010111_2$$

$$X = (-1)^{X_S} \cdot 2^{X_E - 127} (1.X_M)$$

$$
\begin{array}{l|l}
794 & 2 \\
0 & 397 \mid 2 \\
1 & 198 \mid 2 \\
0 & 99 \mid 2 \\
1 & 49 \mid 2 \\
1 & 24 \mid 2 \\
0 & 12 \mid 2 \\
0 & 6 \mid 2 \\
0 & 3 \mid 2 \\
1 & 1
\end{array}
$$

0,08984375 × 2
0,17968750 × 2
0,35937500 × 2
0,71875000 × 2
1,43750000 × 2
0,87500000 × 2
1,75000000 × 2
1,50000000 × 2
1,00000000

$$X = -1{,}1000\,11\,010\,00010\,111 \times 2^9$$

$$X_E - 127 = 9 \Rightarrow X_E = 136$$

$$\underbrace{1\,|\,1\,0\,0}_{C}\;\underbrace{0\,1\,0\,0}_{4}\;\underbrace{0\,|\,1\,0\,0}_{4}\;\underbrace{0\,1\,1\,0}_{6}\;\underbrace{1\,0\,0\,0}_{8}\;\underbrace{0\,1\,0\,1}_{5}\;\underbrace{1\,1\,0\,0}_{C}\;\underbrace{0\,0\,0\,0}_{0}$$

$$X = C44685C0_{16}$$

$X = -794{,}08984375_{10} = -1100011010{,}00010111_2 =$

$= -0{,}001100011010000010111 \times 16^3$

$X = (-1)^{X_S} \cdot 16^{X_E - 64} (0{,}X_M) \qquad X_E - 64 = 3 \Rightarrow X_E = 67$

| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

$\underbrace{1100}_{C}\ \underbrace{0011}_{3}\ \underbrace{0011}_{3}\ \underbrace{0001}_{1}\ \underbrace{1010}_{A}\ \underbrace{0001}_{1}\ \underbrace{0111}_{7}\ \underbrace{0000}_{0}$

$X_{IBM} = C331A170_{16}$

$X_{754} = 45ABD070_{16}$

| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

$139 = X_E \Rightarrow X_E - 127 = 139 - 127 = 12$

$X_{754} = (-1)^0 \cdot 2^{12} \cdot 1{,}0101011110100001111 =$

hidden bit

$= X_{754} = 1010101111010{,}0001111_2 =$

$= +\left(4096 + 1024 + 256 + 64 + 32 + 16 + 8 + 2 + \frac{15}{128}\right) =$

$= +5498{,}1171875$

$X_{IBM} = 454BD0F0_{16}$

$$\boxed{0}\,\boxed{1\,0\,0\,0\,1\,0\,1}\,\boxed{1\,0\,1\,0\,1\,0\,1\,1\,1\,1\,0\,1\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0}$$

$69 = x_E \Rightarrow x_E - 64 = 5$

$$X_{IBM} = (-1)^0 \cdot 16^5 \cdot 0{,}1010\,1011\,1101\,0000\,1111 =$$

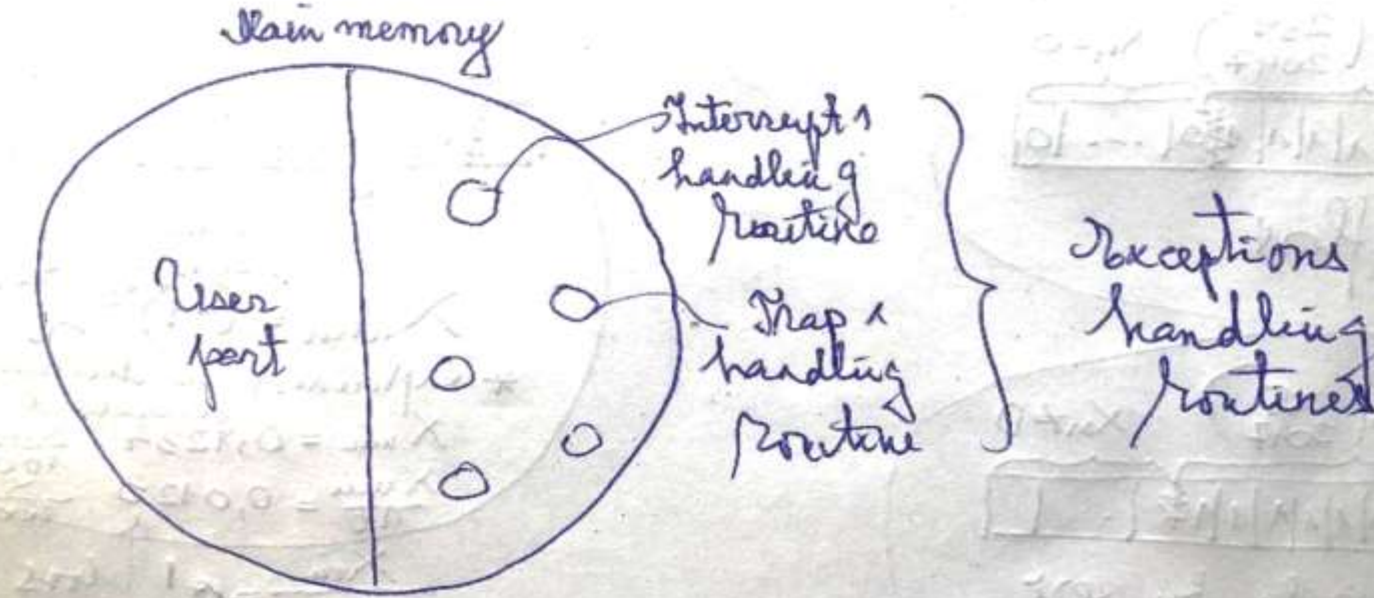
$$= +10101011110100001111 =$$

$$= +(524\,288 + 131\,072 + 32\,768 + 8192 + 4096 + 2048 + 1024 +$$

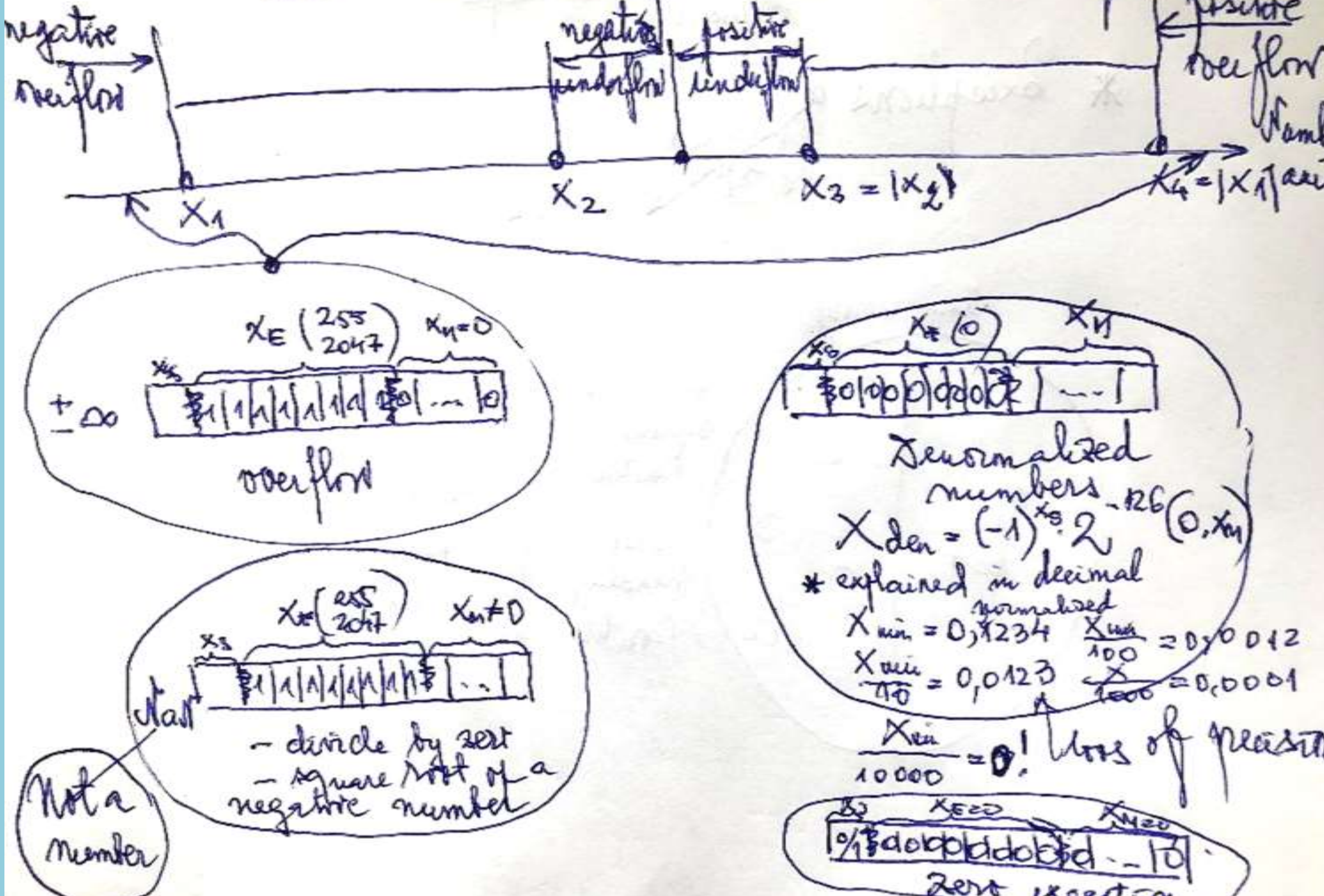
$$+ 256 + 8 + 4 + 2 + 1) = +703759_{10}$$

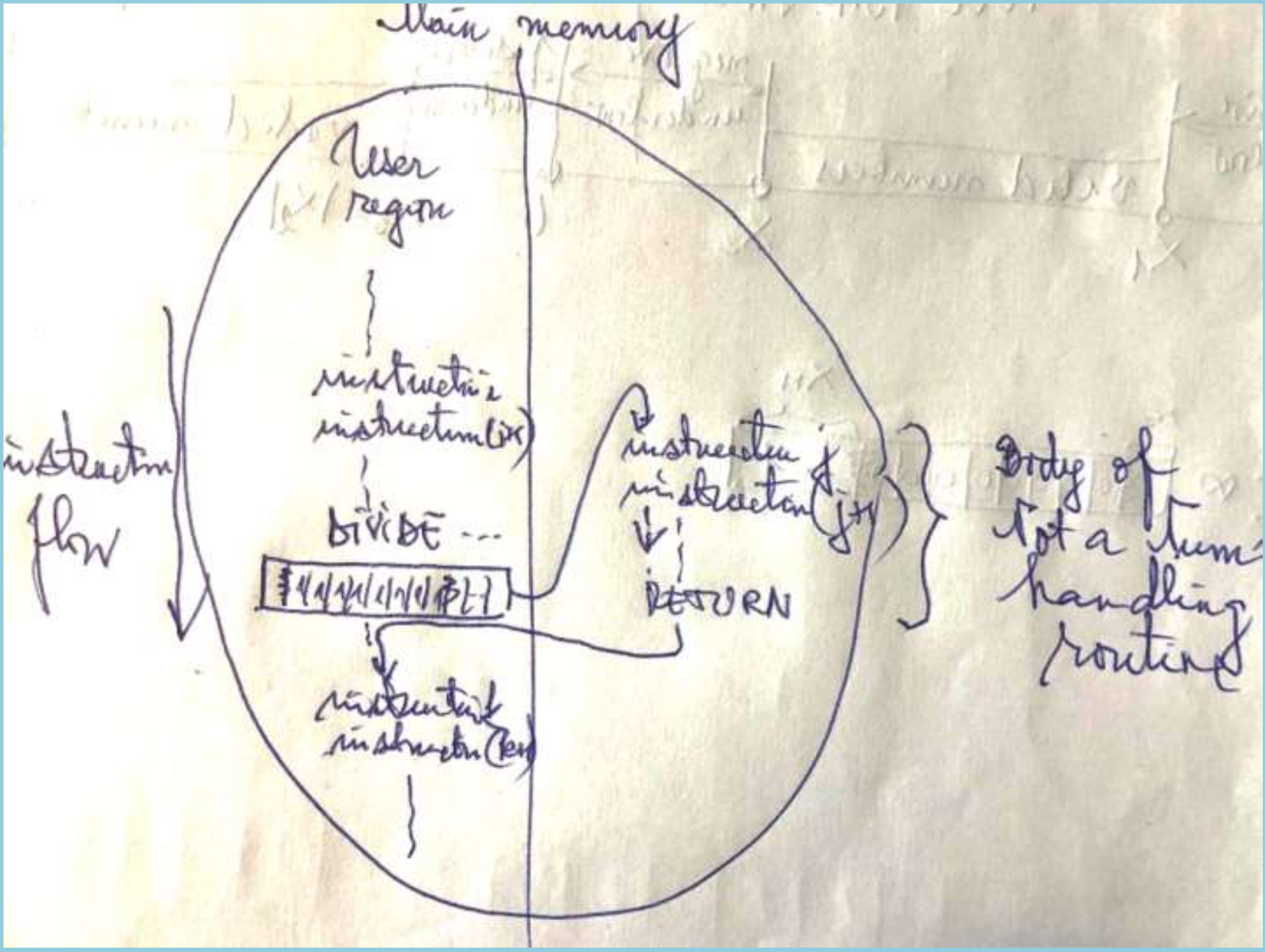
* Exceptions $\xrightarrow{\text{produced by hardware}}$ Interrupts

* Exceptions $\xrightarrow{\text{produced by software}}$ Traps

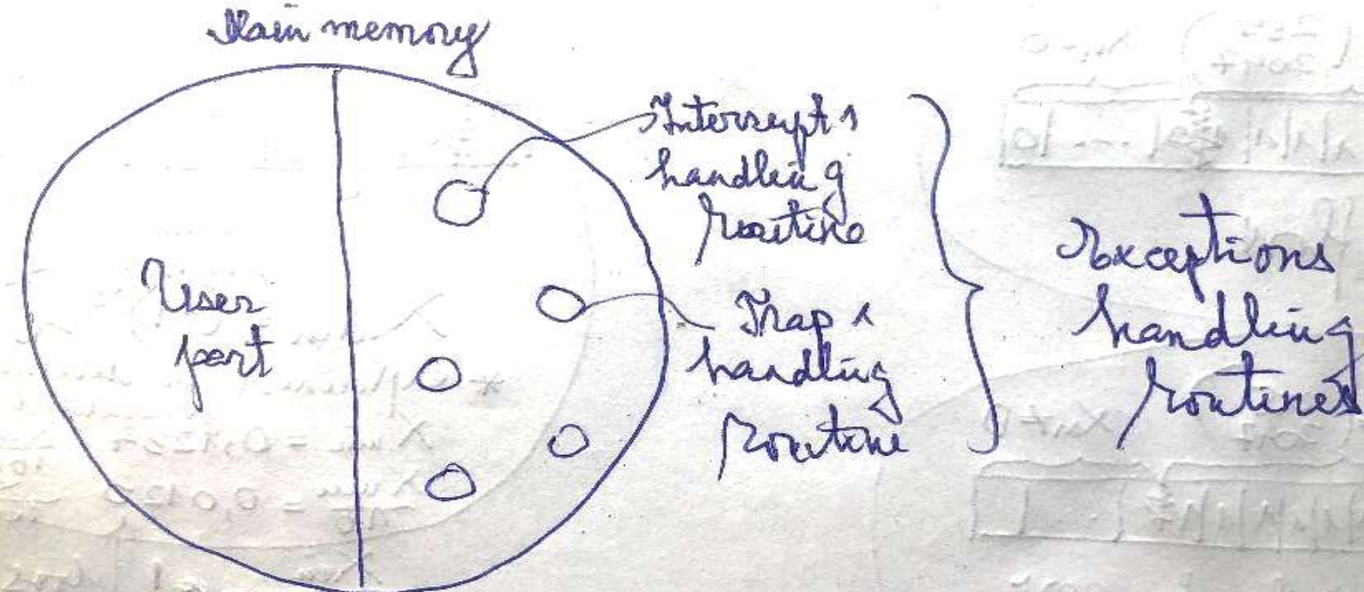


# IEEE 754 Floating Point Standard Exceptions

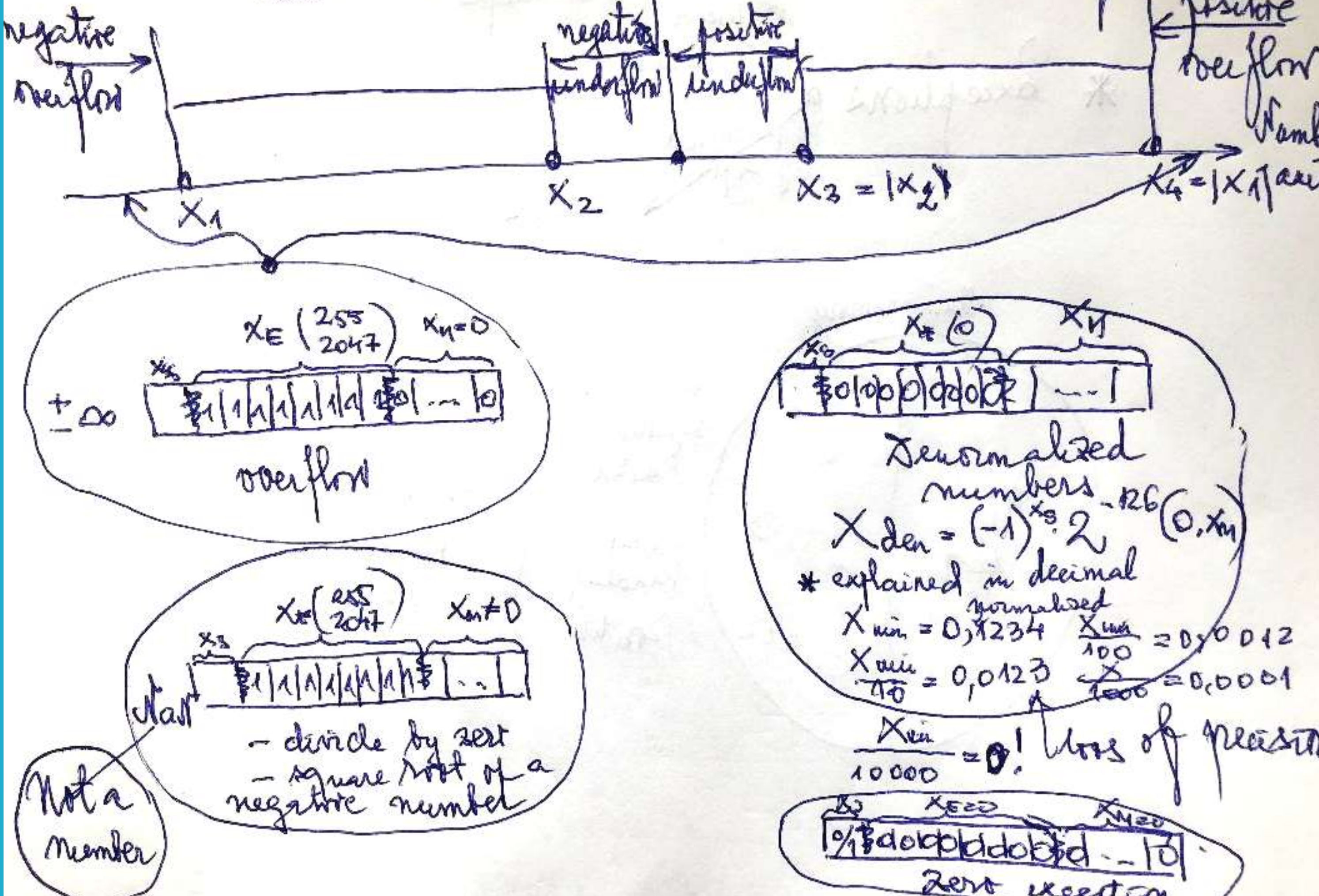


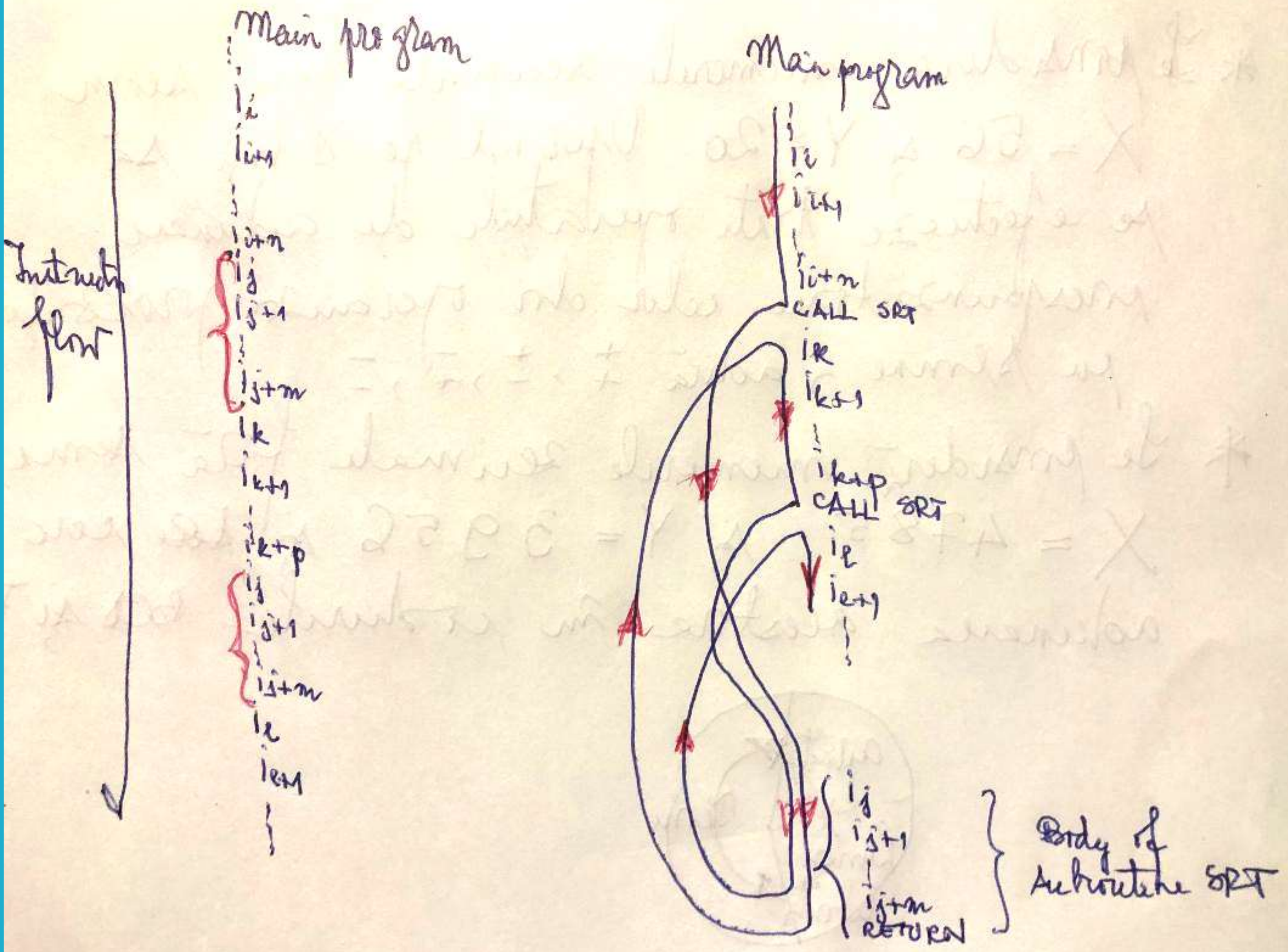

Main memory
User region
instruction
instruction (i+)
instruction flow
DIVIDE ...
RETURN
instruction
instruction (k+)
Body of fault handling routine

* Exceptions
  - produced by hardware → Interrupts
  - produced by software → Traps

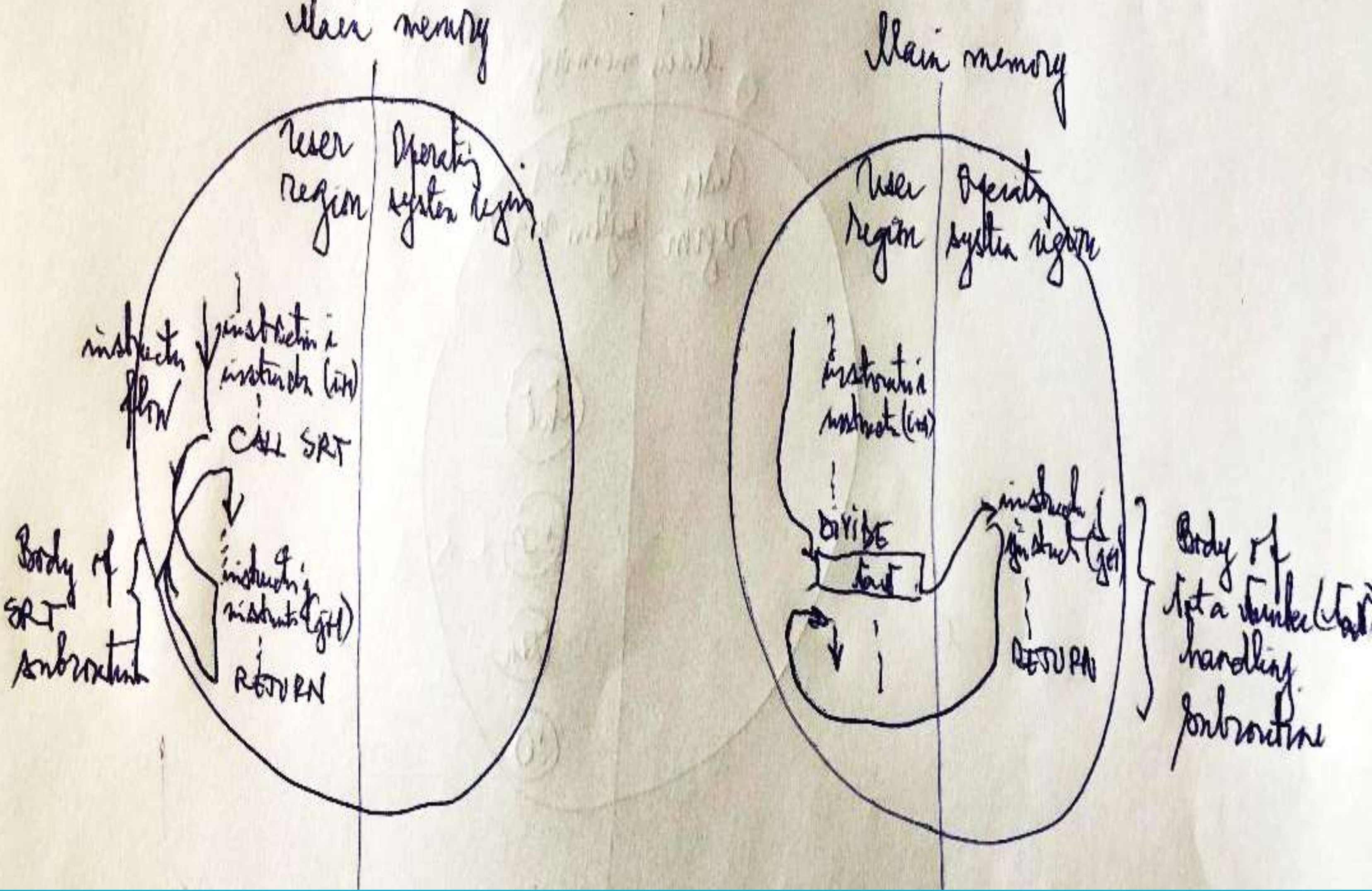


# IEEE 754 Floating Point Standard Exceptions

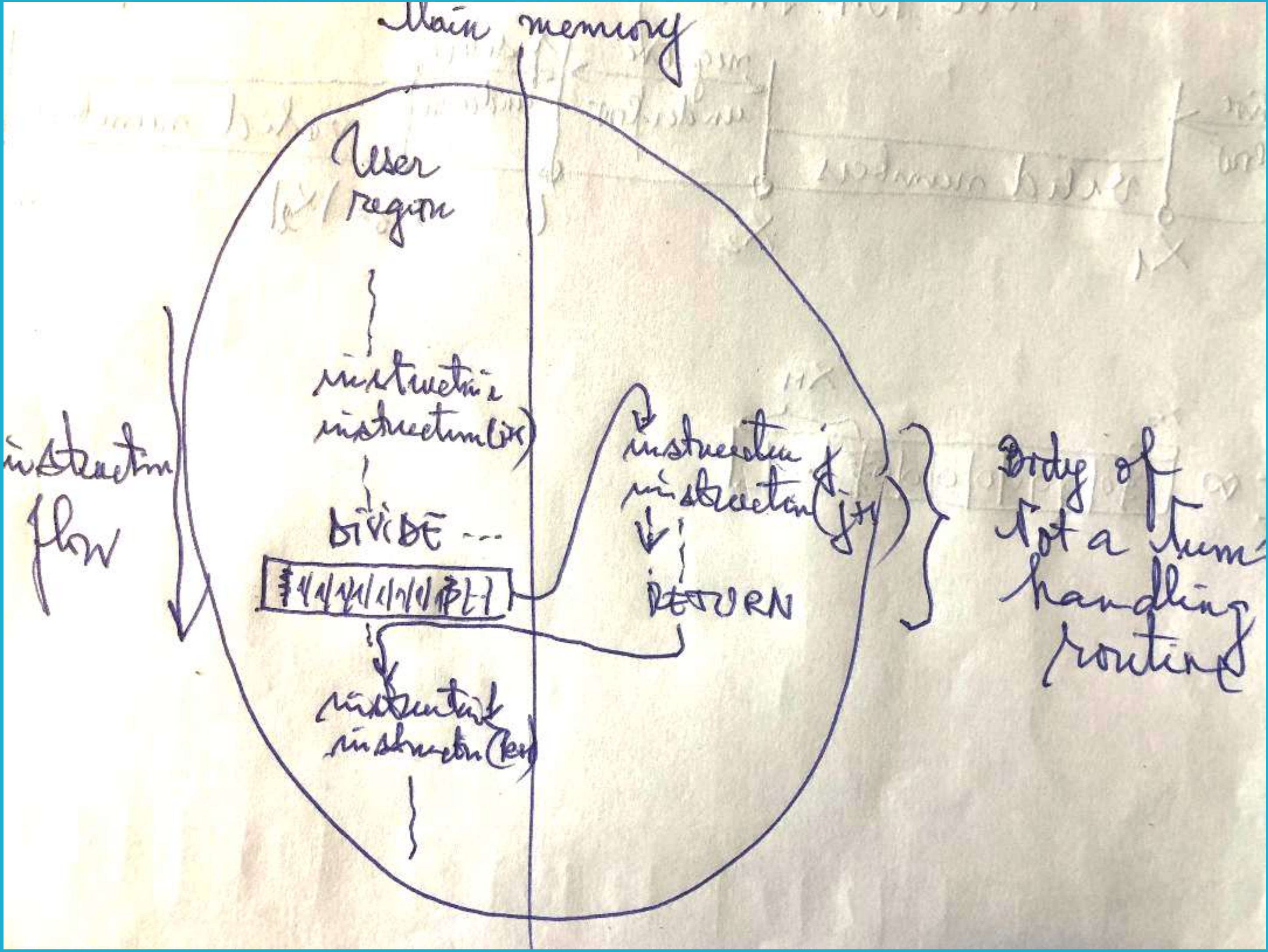


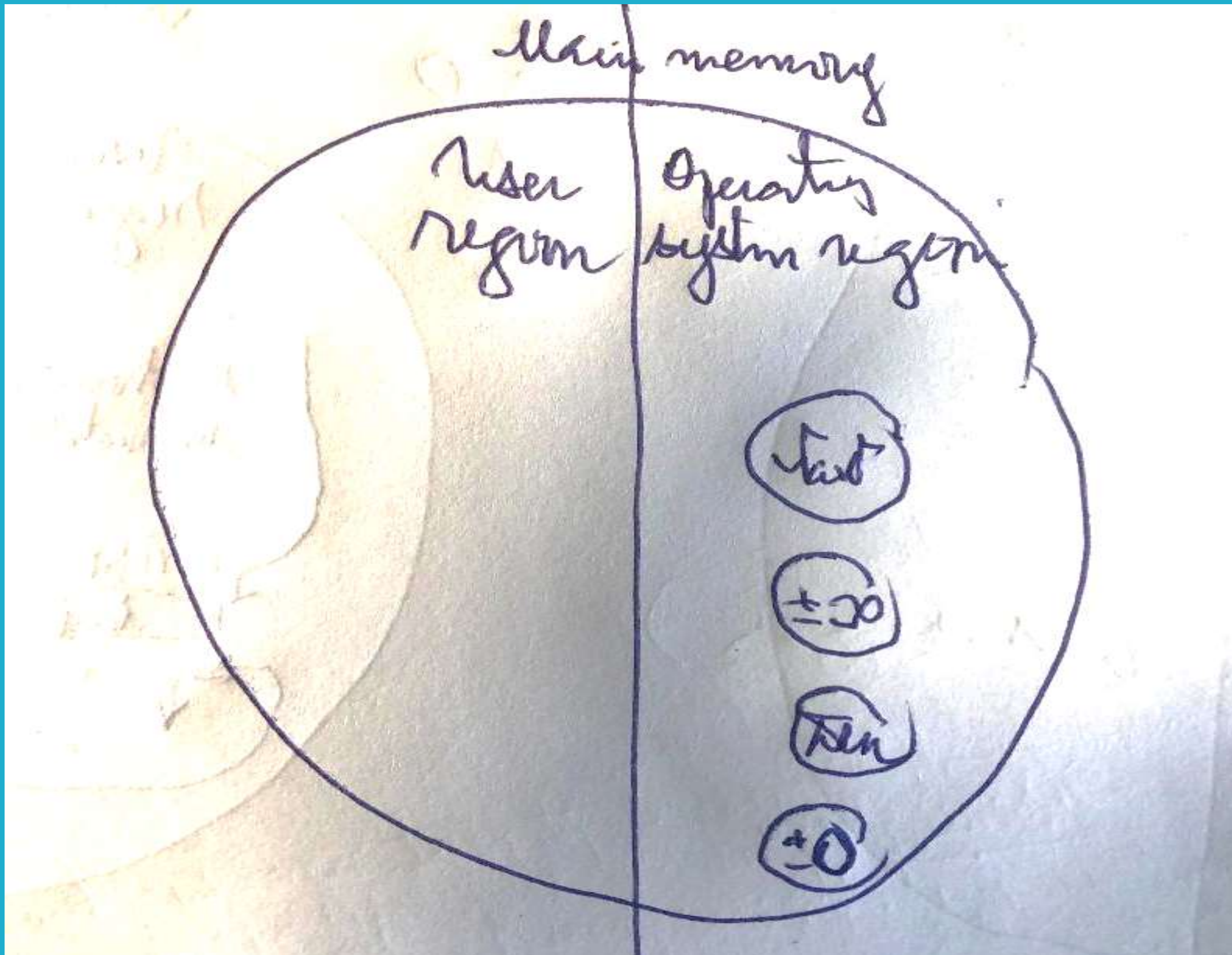


Main program
Instruction flow
i_i
i_{i+1}
i_{i+n}
i_j
i_{j+1}
i_{j+m}
i_k
i_{k+1}
i_{k+p}
i_j
i_{j+1}
i_{j+m}
i_l
i_{l+1}
Main program
i_i
i_{i+1}
i_{i+n}
CALL SRT
i_k
i_{k+1}
i_{k+p}
CALL SRT
i_l
i_{l+1}
i_j
i_{j+1}
i_{j+m}
RETURN
Body of subroutine SRT



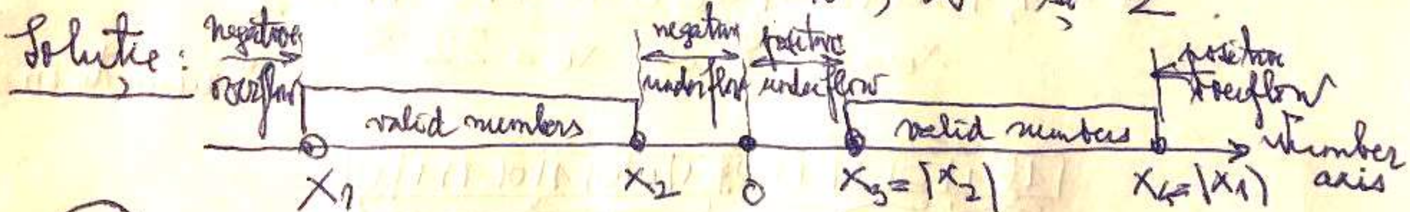
Main memory
User region
Operating system region
instruction flow
instruction i
instruction (i+1)
CALL SRT
Body of SRT subroutine
instruction j
instruction (j+1)
RETURN
Main memory
User region
Operating system region
instruction i
instruction (i+1)
DIVIDE
fault
instruction j
instruction (j+1)
RETURN
Body of fault handling subroutine


Main memory
User region
instruction i
instruction (i+1)
DIVIDE ...
instruction flow
RETURN
Body of
handling
routine


Main memory
user region
Operating system region
±∞
±0

**Subiect**: Se consideră un format ipotetic de virgulă flotantă pe 14 biți, cu 5 biți alocați exponentului și 8 biți alocați mantisei. Adaptând normele specifice standardului IEEE 754 la formatul dat, să se determine câmpul valoric al numerelor de reprezentare validă și să se prezinte în secvențe de semne hexazecimale reprezentările numerelor zecimale $-203,0888671875_{10}$, $2^{-15}$ și $2^{12}$.

**Soluție**:




$X_1$

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$$X_1 = (-1)^1 \cdot 2^{30-15} \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{2^8}\right) =$$

$$= -2^{15} \cdot \frac{511}{2^8} \cong -2^{16} \cong -10^{4,8}$$

$10 \dots 3$
$16 \dots x$

$x = 4,8$

$X_2$

| 1 | 0 0 0 0 1 | 0 0 0 0 0 0 0 0 0 0 |
|---|---|---|

$$X_2 = (-1)^1 \cdot 2^{1-15}(1+0+\dots+0) = -2^{-14} = -10^{4,2}$$

$$\begin{array}{l} 10 - 3 \\ 14 - x \\ \hline x = 4{,}2 \end{array}$$

2a $X^* = -203{,}08886 71875_{10} = -11001011{,}0001011011_2 =$

$= -1{,}1001011 \underbrace{0001011011}_{\text{truncation error}} \times 2^7$

$x_E - 15 = 7 \Rightarrow x_E = 22$

| $\underbrace{1 \; 1}_{3}$ | $\underbrace{0 \; 1 \; 1 \; 0}_{6}$ | $\underbrace{1 \; 0 \; 0 \; 1}_{9}$ | $\underbrace{0 \; 1 \; 1 \; 0}_{6}$ |
|---|---|---|---|

$\Rightarrow X^* = 3696_{H}$

$$X^{**} = 2^{-15} = \underbrace{2^{-14}}_{X_3 = X_{E\min}} \times 2^{-1} = 2^{-14}\cdot\Big(\underbrace{0}_{\text{"hidden"}} + \frac{1}{2} + 0 + \dots + 0\Big)$$

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$\underbrace{0}_{0}\ \underbrace{0\,0\,0\,0\,1}_{1}\ \underbrace{1\,0\,0\,0}_{8}\ \underbrace{0\,0\,0\,0}_{0}$

$X^{**} = 0180_H$ (denormalized number)

$$X^{***} = 2^{12} = \overset{2^{12}}{1}\overset{2^{11}}{0}00000000\overset{2^{1}}{0}\overset{2^{0}}{0} = (1 + 0 + \dots + 0) \times 2^{12}$$

$X_E - 15 = 12 \Rightarrow X_E = 27$

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$\underbrace{0\,1}_{1}\ \underbrace{1\,0\,1\,1}_{B}\ \underbrace{0\,0\,0\,0}_{0}\ \underbrace{0\,0\,0\,0}_{0}$

$X^{***} = 1B00_H$

* Se consideră numerele zecimale fără semn X = 56 și Y = 20. Operând pe 8 biți să se efectueze toate operațiile de adunare corespunzătoare celor doi operanzi prevăzuți cu semne, adică $\pm$ : $+/+$, $+/-$, $-/+$, $-/-$.

* Se consideră numerele zecimale fără semn X = 4785 și Y = 3956 și se cere adunarea acestora în codurile BCD și E3.

I. The Representaton of Numbers in Computing Systems

II. Boolean Algebra

* History - George Boole (1815-1864), Edward Huntington (1874-1952)
* variables, literals

Claude Shannon (1916-2001) - "the father of information theory", Augustus De Morgan (1806-1871)

## Logic Functions

Truth table

AND

| Inputs | | Output |
|---|---|---|
| x | y | Z |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

for n input variables → $2^n$ lines

| Inputs | | | Output |
|---|---|---|---|
| w | x | y | Z |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

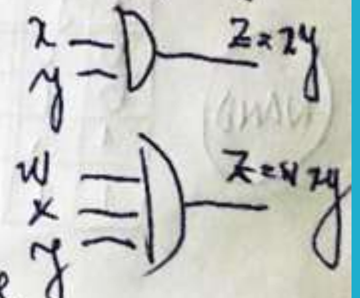
Symbol

logic product

$Z = x \cdot y = xy =$
$= x$ and $y$
$(Z = x \wedge y)$
$Z = xy$

* 0's dominance
* conjunction

Logic gates



Truth table | Symbol | Logic gate

**OR**

| Inputs | | Output |
|---|---|---|
| x | y | z |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

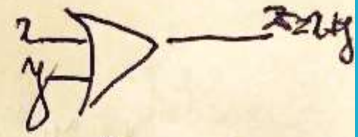
| Inputs | | | Output |
|---|---|---|---|
| w | x | y | z |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

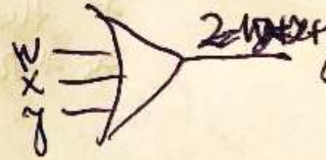
Logic sum

$Z = x + y = x$ OR $y$

$(Z = x \vee y)$

$Z = w + y + z$



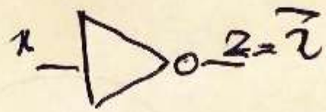

* 1's dominance
* disjunction

**NOT**

| Input | Output |
|---|---|
| x | z |
| 0 | 1 |
| 1 | 0 |

complement

$Z = \overline{x}$



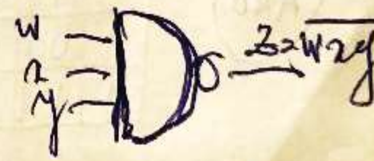

**NAND**

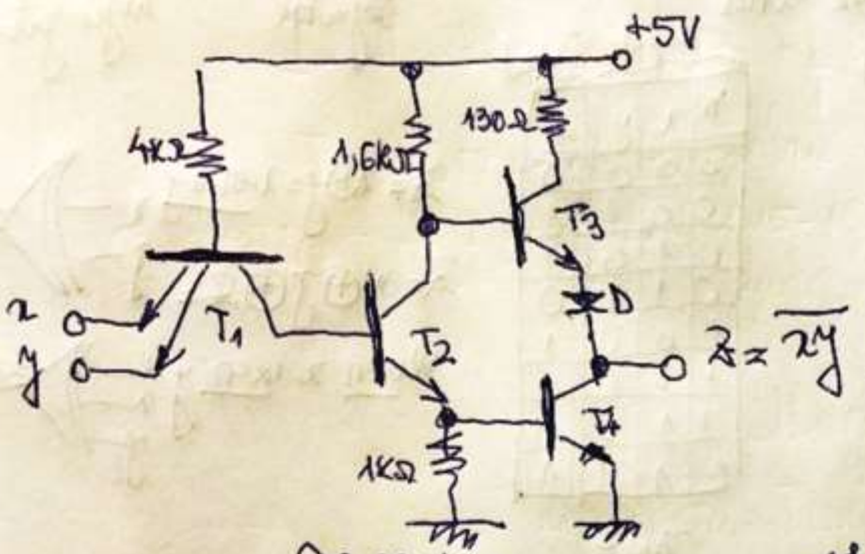
| Inputs | | Output |
|---|---|---|
| x | y | z |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| Inputs | | | Output |
|---|---|---|---|
| x | y | z | z |
| 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 |

$Z = \overline{x \cdot y} = \overline{xy} = x$ nand $y$

$Z = \overline{wxy}$




Jack Kilby – Texas Instruments

| | Voltage |
|---|---|
| 0 | L (Low) |
| 1 | H (High) |

Truth tables

NOR

| x | y | z |
|---|---|---|
| | Inputs | Output |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

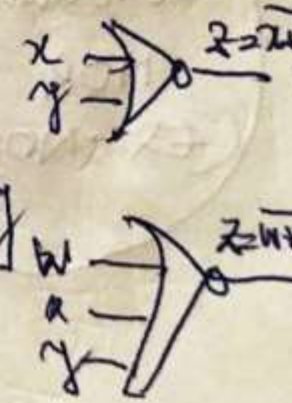
| w | q | y | z |
|---|---|---|---|
| | Inputs | | Output |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

Symbol

Logic gate

$z = \overline{x+y} = x \ \underline{\text{NOR}} \ y$

$z = \overline{w+q+y} = w \ \underline{\text{NOR}} \ q \ \underline{\text{NOR}} \ y$



Truth tables | Symbol | Logic gates

## Exclusive OR (EX-OR)

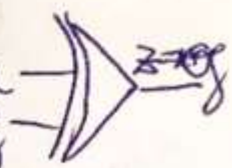
Inputs / Output

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Inputs / Output

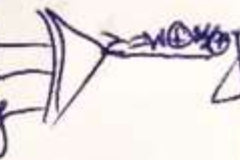
| w | x | y | z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

$Z = x \oplus y =$ x ex-or y




$Z = w \oplus x \oplus y =$

w ex-or x ex-or y




## Exclusive NOR (EX-NOR)

Inputs / Output

| x | y | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

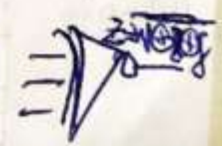
Inputs

| w | x | y | z |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

$Z = \overline{x \oplus y} =$ x ex-nor y


$Z = \overline{w \oplus x \oplus y} =$

= w ex-nor x ex-nor y



# Laws and Postulates of Boole's Algebra

| AND | OR | |
|---|---|---|
| $x \cdot 0 = 0$ | $x + 0 = x$ | 0's laws |
| $x \cdot 1 = x$ | $x + 1 = 1$ | 1's laws |
| $x \cdot \bar{x} = 0$ | $x + \bar{x} = 1$ | Negation laws |

$$\bar{\bar{x}} = x \quad \text{Double negation law}$$

| AND | OR | |
|---|---|---|
| $x \cdot x = x$ | $x + x = x$ | Idempotence laws |
| $x \cdot y = y \cdot x$ | $x + y = y + x$ | Commutativity laws |
| $x \cdot (y z) = (x y) z = (x z) y$ | $x + (y + z) = (x + y) + z = (x + z) + y$ | Associativity laws |
| $x \cdot (y + z) = xy + xz$ | $x + yz = (x + y)(x + z)$ | Distributivity laws |

* Commutativity and associativity can be extended to an arbitrary, but finite number of terms, regardless to their order.

| AND | OR | |
|---|---|---|
| $x \cdot (x + y) = x$ | $x + xy = x$ | Absorption laws |
| $x \cdot (\bar{x} + y) = xy$ | $x + \bar{x} y = x + y$ | Laws of common identities |
| $\overline{x \cdot y} = \bar{x} + \bar{y}$ | $\overline{x + y} = \bar{x} \cdot \bar{y}$ | De Morgan's laws |

* De Morgan's laws can be generalised to an arbitrary number of terms

$$\overline{x_1 \cdot x_2 \cdot \ldots \cdot x_n} = \bar{x}_1 + \bar{x}_2 + \ldots + \bar{x}_n$$

$$\overline{x_1 + x_2 + \ldots + x_n} = \bar{x}_1 \cdot \bar{x}_2 \cdot \ldots \cdot \bar{x}_n$$

# Laws and Postulats of Boole's Algebra

| AND | OR | |
|---|---|---|
| $x \cdot 0 = 0$ | $x + 0 = x$ | 0's laws |
| $x \cdot 1 = x$ | $x + 1 = 1$ | 1's laws |
| $x \cdot \bar{x} = 0$ | $x + \bar{x} = 1$ | Negation laws |
| $\bar{\bar{x}} = x$ | | Double negation law |
| $x \cdot x = x$ | $x + x = x$ | Idempotence laws |
| $x \cdot y = y \cdot x$ | $x + y = y + x$ | Commutativity laws |
| $x \cdot (y z) = (x y) z = (x z) y$ | $x + (y + z) = (x + y) + z = (x + z) + y$ | Associativity laws |
| $x \cdot (y + z) = xy + xz$ | $x + yz = (x + y)(x + z)$ | Distributivity laws |

* Commutativity and associativity can be extended to an arbitrary, but finity number of terms, regardless to their order.

| AND | OR | |
|---|---|---|
| $x \cdot (x + y) = x$ | $x + xy = x$ | Absorption laws |
| $x \cdot (\bar{x} + y) = xy$ | $x + \bar{x} y = x + y$ | Laws of common identities |
| $\overline{x \cdot y} = \bar{x} + \bar{y}$ | $\overline{x + y} = \bar{x} \cdot \bar{y}$ | De Morgan's laws |

* De Morgan's laws can be generalized to an arbitrary number of terms

$$\overline{x_1 \cdot x_2 \cdots x_n} = \bar{x}_1 + \bar{x}_2 + \dots + \bar{x}_n$$

$$\overline{x_1 + x_2 + \dots + x_n} = \bar{x}_1 \cdot \bar{x}_2 \cdots \bar{x}_n$$

# Proofs

## Distributivity OR law

| Inputs | | | | M1 | | | M2 |
|---|---|---|---|---|---|---|---|
| $x$ | $y$ | $z$ | $y \cdot z$ | $x+y \cdot z$ | $x+y$ | $x+z$ | $(x+y)(x+z)$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$\equiv$

q.e.d

## De Morgan's laws

| Inputs | | M1 | | | M2 | M1 | | | M2 |
|---|---|---|---|---|---|---|---|---|---|
| $x$ | $y$ | $\overline{x \cdot y}$ | $\bar{x}$ | $\bar{y}$ | $\bar{x}+\bar{y}$ | $\overline{x+y}$ | $\bar{x}$ | $\bar{y}$ | $\bar{x} \cdot \bar{y}$ |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

$\equiv$ $\equiv$

q.e.d q.e.d

① $F(x,y) = \overline{x+y}\cdot(\bar{x}+\bar{y}) = \bar{x}\cdot\bar{y}\,(\bar{x}+\bar{y}) = \bar{x}\bar{y}+\bar{x}\bar{y} = \bar{x}\bar{y}$

② $F(x,y,z) = xyz + \bar{x}y + xy\bar{z}$

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

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

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

$= y(1+\bar{x}) = y$

Inputs

| $x$ | $y$ | $z$ | $xyz$ | $\bar{x}y$ | $xy\bar{z}$ | $F(x,y,z)$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 |

=

$F(A,B,C,D) = (B\bar{C}+\bar{A}D)(A\bar{B}+C\bar{D}) = \underbrace{A\bar{B}B\bar{C}}_{0} + \underbrace{A\bar{A}\bar{B}D}_{0} +$

$+ \underbrace{BC\bar{C}\bar{D}}_{0} + \bar{A}\,\underbrace{CD\bar{D}}_{0} = 0$

(4) $F(A,B,C) = \bar{A}\bar{C} + ABC + A\bar{C} = \bar{C}(\underbrace{\bar{A}+A}) + ABC =$

$= \bar{C} + ABC = (\underbrace{\bar{C}+C})(\bar{C}+AB) = \bar{C} + AB$

| A | B | C | $\bar{A}\bar{C}$ | $ABC$ | $A\bar{C}$ | $F(A,B,C)$ | $\bar{C}$ | $AB$ | $F'(A,B,C)$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

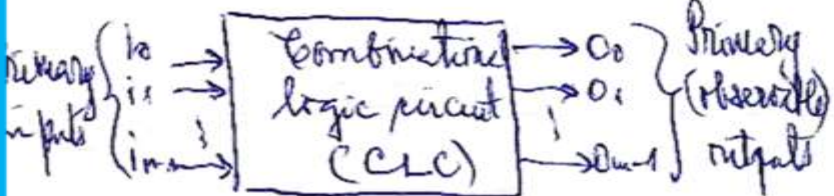
(5) $F(A,B,C,D) = \bar{A}B(\bar{D} + \bar{C}D) + B(A + \bar{A}CD) =$

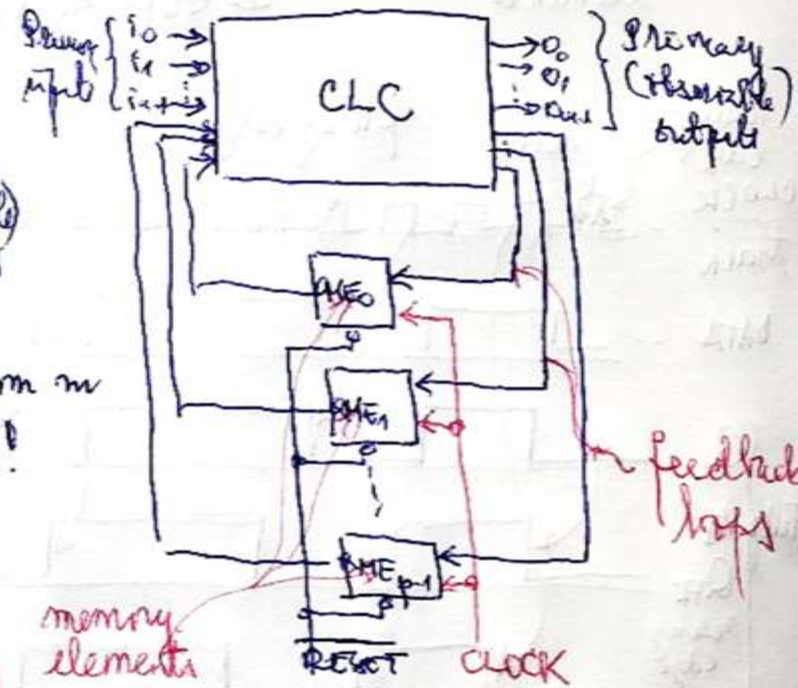
$= \bar{A}B(\bar{D}+\bar{C}) + B(A+CD) = B(\bar{A}\bar{D} + \bar{A}\bar{C} + A + CD =$

$= B(A + \bar{D} + A + \bar{C} + CD) = B(A + \underbrace{\bar{D} + \bar{C} + D}_{1}) =$

$= B(\underbrace{A + \bar{C} + 1}_{1}) = B$

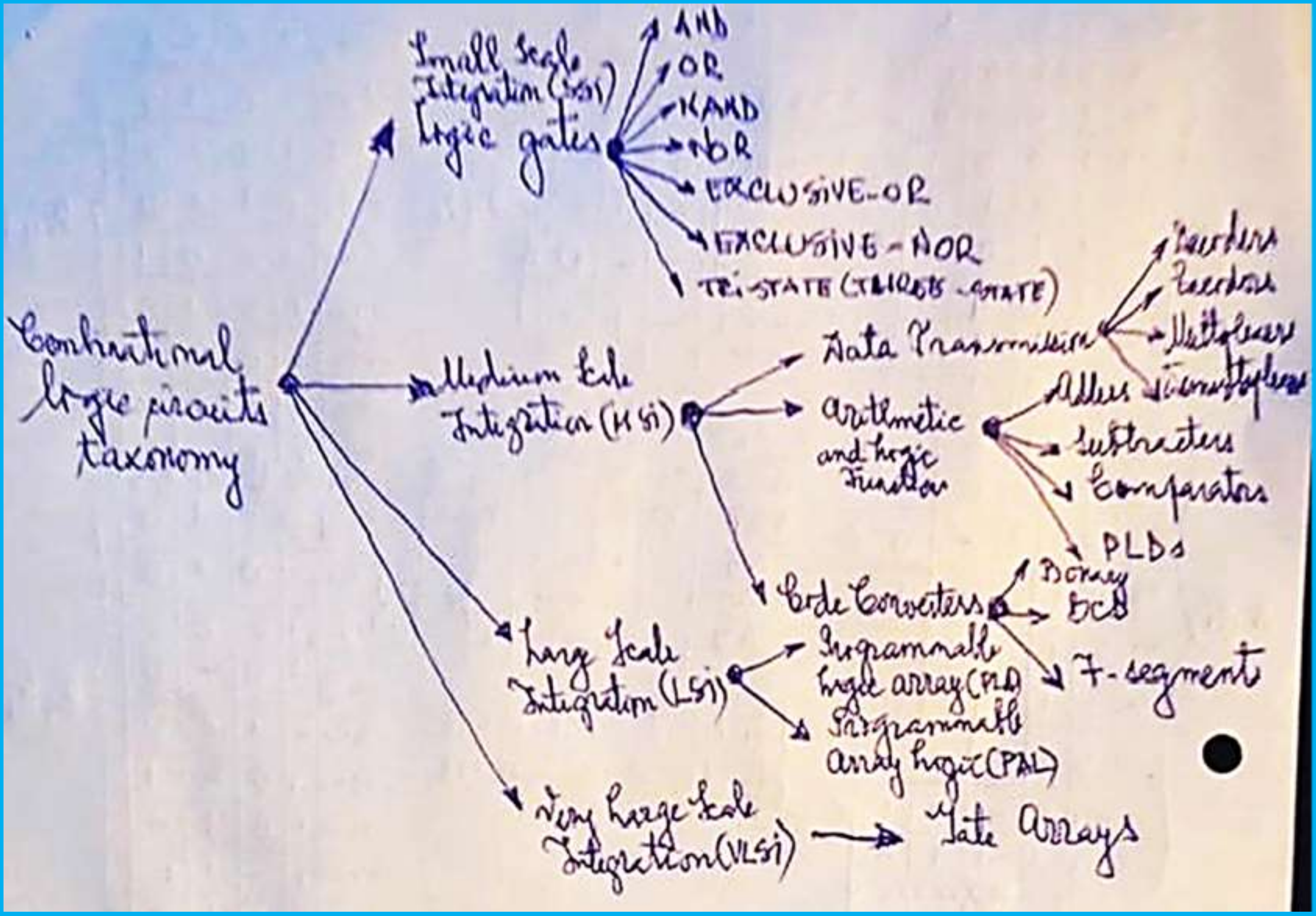
# Combinational vs sequential logic circuits

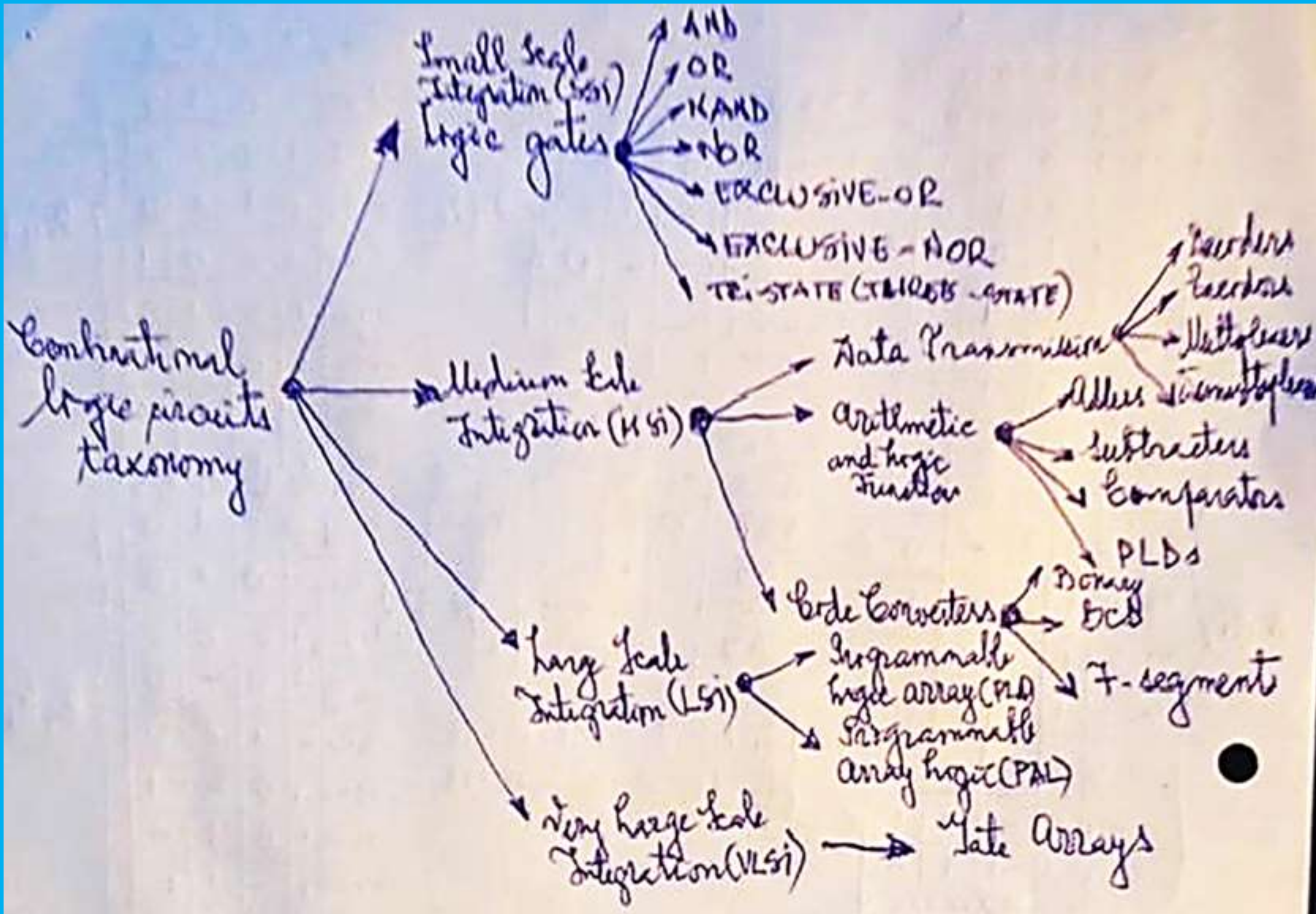



* n in general different from m
* no memory elements!
* no feedback loops

* Sequential logic circuit
  - → Synchronous (with CLOCK)
  - → Asynchronous (without CLOCK)


Combinational logic circuits taxonomy
Small Scale Integration (SSI) logic gates
AND
OR
NAND
NOR
EXCLUSIVE-OR
EXCLUSIVE-NOR
TRI-STATE (THREE-STATE)
Medium Scale Integration (MSI)
Data Transmission
Decoders
Encoders
Multiplexers
Demultiplexers
Arithmetic and logic Function
Adders
Subtracters
Comparators
PLDs
Code Converters
Binary
BCD
7-segment
Large Scale Integration (LSI)
Programmable logic array (PLA)
Programmable array logic (PAL)
Very Large Scale Integration (VLSI)
Gate Arrays


Combinational logic circuits taxonomy
Small Scale Integration (SSI) logic gates
AND
OR
NAND
NOR
EXCLUSIVE-OR
EXCLUSIVE-NOR
TRI-STATE (THREE-STATE)
Medium Scale Integration (MSI)
Data Transmission
Decoders
Encoders
Multiplexers
Demultiplexers
Arithmetic and logic function
Adders
Subtracters
Comparators
PLDs
Code Converters
Binary
BCD
7-segment
Large Scale Integration (LSI)
Programmable logic array (PLA)
Programmable array logic (PAL)
Very Large Scale Integration (VLSI)
Gate Arrays

* Product term

$x\bar{y}, x\bar{y}z, \ldots$

* Minterm

For a Boolean function of n variables $x_1, x_2, \ldots, x_n$, a product term in which each ~~variables~~ of the n variables appears $\boxed{\text{once}}$ (in either its complemented or uncomplemented form) is called a $\boxed{\text{minterm}}$

* Canonical disjunctive normal form (CDNF) or minterm canonical form

* Sum of Products (SOP)

* Sum term

$x+\bar{y}, x+\bar{y}+z, \ldots$

* Maxterm

For a Boolean function of n variables $x_1, x_2, \ldots, x_n$, a sum term in which each of the n variables appears $\boxed{\text{once}}$ (in either its complemented or uncomplemented form) is called a $\boxed{\text{maxterm}}$

* Canonical conjunctive normal form (CCNF) or maxterm canonical form

* Products of Sum (POS)

# Design Steps of Combinational Logic Circuits

Problem statement → Truth table construct → Logic equations written → Logic equations minimization → Logic circuits drawn → Logic circuits built

$$\begin{cases} X = +57_{10} = 00111001_{2\text{-SM}} = 00111001_{2\text{-C1}} = 00111001_{2\text{-C2}} \\ Y = -42_{10} = 10101010_{2\text{-SM}} = 11010101_{2\text{-C1}} = 11010110_{2\text{-C2}} \\ Z = +15_{10} \end{cases}$$

$11100011_{2\text{-SM}} = -99_{10}$ False

$①00001110$, end-around carry $+1$ → $\underbrace{00001111}_{+15\text{ - OK}}$

$\times\underbrace{00001111}_{+15\text{ - OK}}$

$x_i + y_i + r_i \to r_{i+1}, z_i$

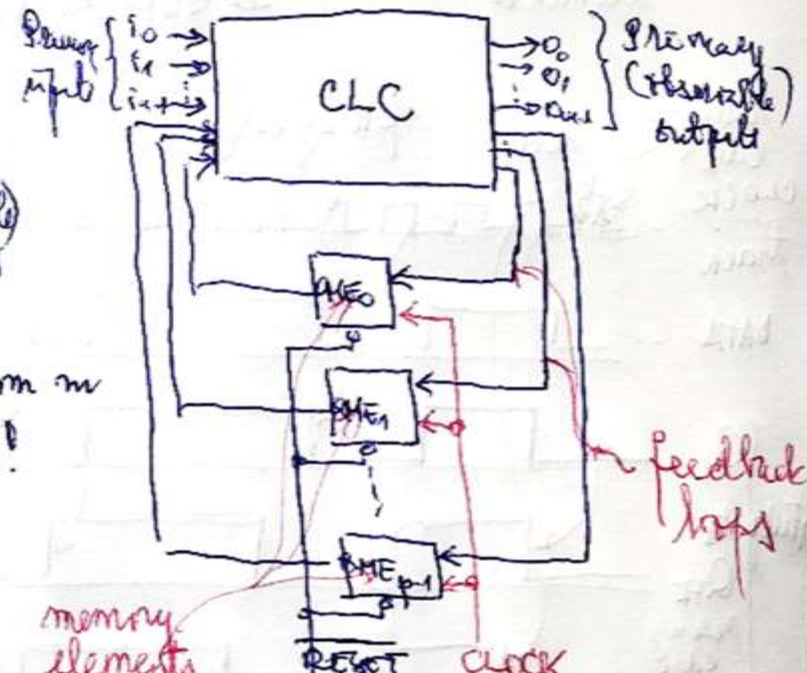
| $x_i$ | $y_i$ | $r_i$ | $r_{i+1}$ | $z_i$ | Minterms | Maxterms |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $\bar{x}_i\bar{y}_i\bar{r}_i$ | $x_i+y_i+r_i$ |
| 0 | 0 | 1 | 0 | 1 | $\bar{x}_i\bar{y}_i r_i$ | $x_i+y_i+\bar{r}_i$ |
| 0 | 1 | 0 | 0 | 1 | ⋮ | |
| 0 | 1 | 1 | 1 | 0 | | |
| 1 | 0 | 0 | 0 | 1 | | |
| 1 | 0 | 1 | 1 | 0 | | |
| 1 | 1 | 0 | 1 | 0 | $x_i y_i \bar{r}_i$ | $\bar{x}_i+\bar{y}_i+r_i$ |
| 1 | 1 | 1 | 1 | 1 | $x_i y_i r_i$ | $\bar{x}_i+\bar{y}_i+\bar{r}_i$ |

$$\text{SOP}\begin{cases} r_{i+1} = \bar{x}_i y_i r_i + x_i\bar{y}_i r_i + x_i y_i \bar{r}_i + x_i y_i r_i \\ z_i = \bar{x}_i\bar{y}_i r_i + \bar{x}_i y_i \bar{r}_i + x_i\bar{y}_i\bar{r}_i + x_i y_i r_i \end{cases}$$

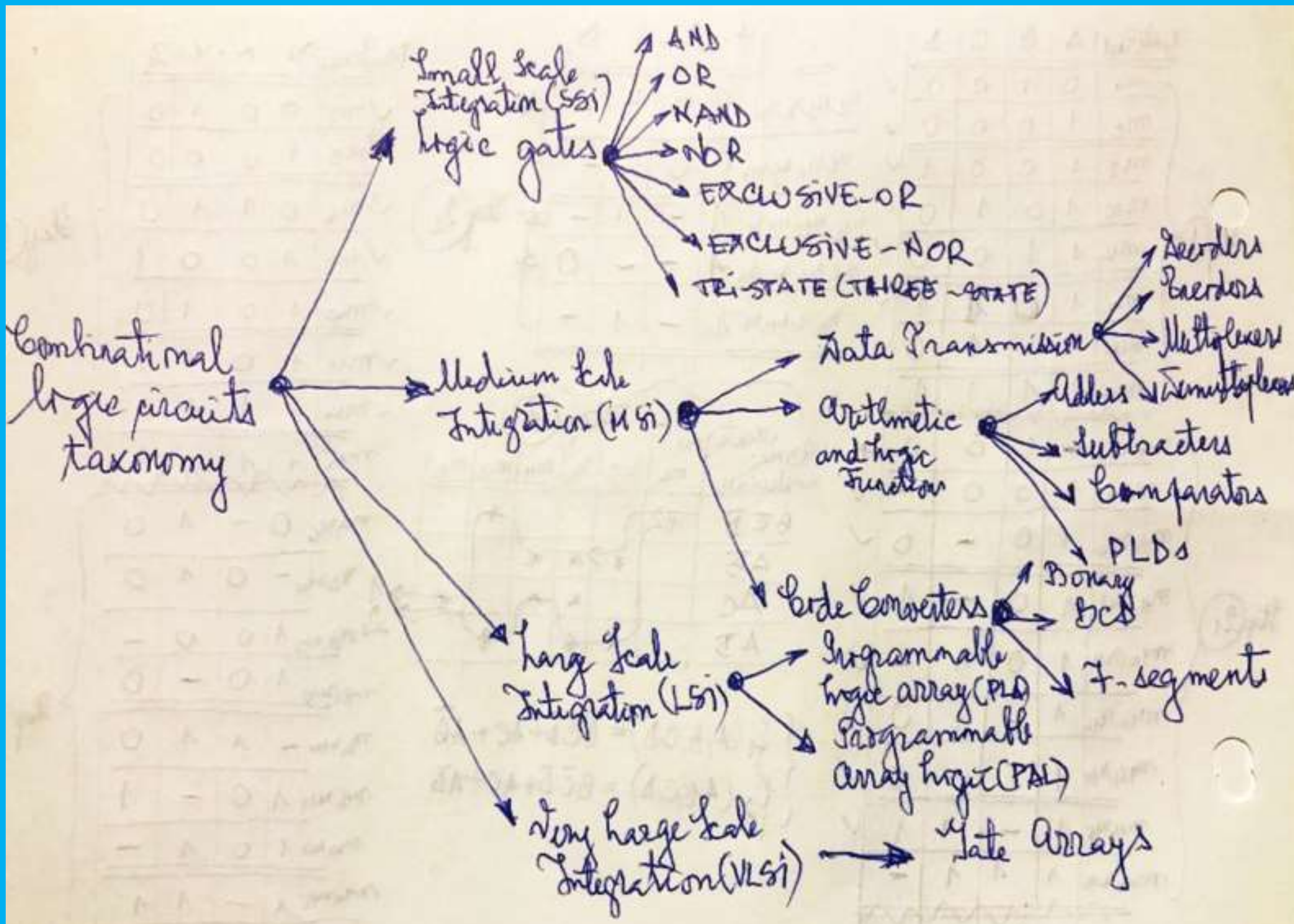
$$\text{POS}\begin{cases} r_{i+1} = (x_i+y_i+r_i)(x_i+y_i+\bar{r}_i)(x_i+\bar{y}_i+r_i)(\bar{x}_i+y_i+r_i) \\ z_i = (x_i+y_i+r_i)(x_i+\bar{y}_i+\bar{r}_i)(\bar{x}_i+y_i+\bar{r}_i)(\bar{x}_i+\bar{y}_i+r_i) \end{cases}$$

# Combinational vs sequential logic circuits


* n in general different from m
* no memory elements!
* no feedback loops




* Sequential logic circuit
  - Synchronous (with CLOCK)
  - Asynchronous (without CLOCK)


Combinational logic circuits taxonomy
Small Scale Integration (SSI) logic gates
AND
OR
NAND
NOR
EXCLUSIVE-OR
EXCLUSIVE-NOR
TRI-STATE (THREE-STATE)
Medium Scale Integration (MSI)
Data Transmission
Decoders
Encoders
Multiplexers
Demultiplexers
Arithmetic and logic Function
Adders
Subtracters
Comparators
PLDs
Code Converters
Binary
BCD
7-segment
Large Scale Integration (LSI)
Programmable logic array (PLA)
Programmable array logic (PAL)
Very Large Scale Integration (VLSI)
Gate Arrays

* Product term

$x\bar{y}, x\bar{y}z, \ldots$

* Minterm

For a Boolean function of n variables $x_1, x_2, \ldots, x_n$, a product term in which each ~~variables~~ of the n variables appears $\boxed{\text{once}}$ (in either its complemented or uncomplemented form) is called a $\boxed{\text{minterm}}$

* Canonical disjunctive normal form (CDNF) or minterm canonical form

* Sum of Products (SOP)

* Sum term

$x+\bar{y}, x+\bar{y}+z, \ldots$

* Maxterm

For a Boolean function of n variables $x_1, x_2, \ldots, x_n$, a sum term in which each of the n variables appears $\boxed{\text{once}}$ (in either its complemented or uncomplemented form) is called a $\boxed{\text{maxterm}}$

* Canonical conjunctive normal form (CCNF) or maxterm canonical form

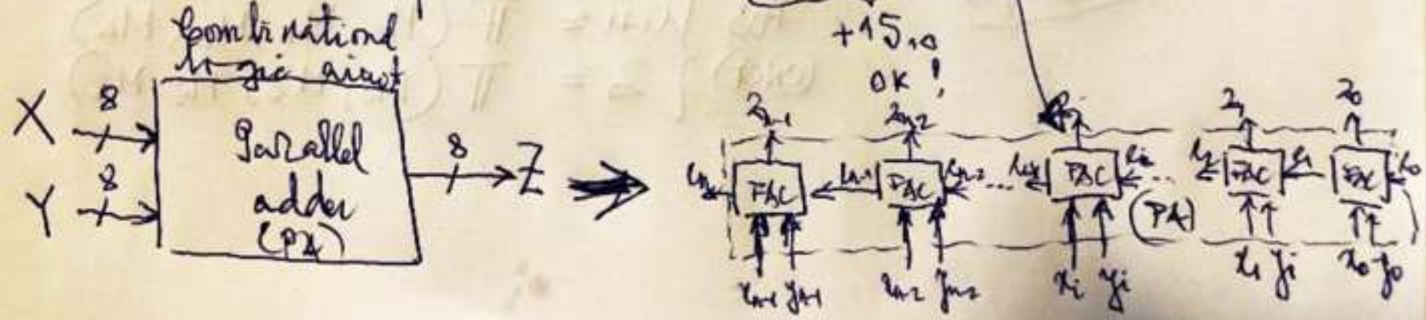
* Products of Sums (POS)

Problem statement → Truth table construct → Logic equation(s) written → Logic equation(s) minimization → Logic circuit(s) drawn → Logic circuit(s) built

$$
\begin{array}{rll}
 & X = +57_{10} = 00111001_{2\text{-SM}} = 00111001_{2\text{-C1}} = 00111001_{2\text{-C2}} \\
+ & Y = -42_{10} = 10101010_{2\text{-SM}} = 11010101_{2\text{-C1}} = 11010110_{2\text{-C2}} \\
\hline
 & Z = +15_{10} \quad \underbrace{11100011_{2\text{-SM}}}_{\cancel{-99_{10}}\ \text{false!}} \quad 1\,00001110_{2\text{-C1}} \quad \cancel{1}\,\underbrace{00001111_{2\text{-C2}}}_{+15_{10}\ \text{ok!}}
\end{array}
$$

$$
\begin{array}{r}
00001110 \\
+\quad 1 \\
\hline
\underbrace{00001111}_{+15_{10}\ \text{OK!}}
\end{array}
$$

end-around carry



Problem statement → $\boxed{\text{Truth table constructed}}$ → logic equation(s) written → logic equation(s) minimized → logic circuit(s) drawn → logic circuit(s) built

| Inputs | | | Outputs | |
|---|---|---|---|---|
| $x_i$ | $y_i$ | $c_i$ | $c_{i+1}$ | $z_i$ |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

| Minterms | Maxterms |
|---|---|
| $\bar{x}_i \bar{y}_i \bar{c}_i$ — $m_0$ | $x_i + y_i + c_i$ — $M_0$ |
| $\bar{x}_i \bar{y}_i c_i$ — $m_1$ | $x_i + y_i + \bar{c}_i$ — $M_1$ |
| $\bar{x}_i y_i \bar{c}_i$ — $m_2$ | $x_i + \bar{y}_i + c_i$ — $M_2$ |
| $\bar{x}_i y_i c_i$ — $m_3$ | $x_i + \bar{y}_i + \bar{c}_i$ — $M_3$ |
| $x_i \bar{y}_i \bar{c}_i$ — $m_4$ | $\bar{x}_i + y_i + c_i$ — $M_4$ |
| $x_i \bar{y}_i c_i$ — $m_5$ | $\bar{x}_i + y_i + \bar{c}_i$ — $M_5$ |
| $x_i y_i \bar{c}_i$ — $m_6$ | $\bar{x}_i + \bar{y}_i + c_i$ — $M_6$ |
| $x_i y_i c_i$ — $m_7$ | $\bar{x}_i + \bar{y}_i + \bar{c}_i$ — $M_7$ |

$x_i + y_i + c_i \rightarrow c_{i+1}, \; z_i$

$\boxed{\text{plus!!}}$

arithmetic!

Problem statement → Truth table construct → $\boxed{\text{Logic equation(s) written}}$ → Logic equation(s) minimization → Logic circuit(s) drawn → Logic circuit(s) built

| Inputs | | | Outputs | |
|---|---|---|---|---|
| $x_i$ | $y_i$ | $c_i$ | $c_{i+1}$ | $z_i$ |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

SOP (CNDF)
$$\begin{cases} c_{i+1} = x_i y_i c_i + x_i \bar{y}_i c_i + x_i y_i \bar{c}_i + x_i y_i c_i \\ z_i = \bar{x}_i \bar{y}_i c_i + \bar{x}_i y_i \bar{c}_i + x_i \bar{y}_i \bar{c}_i + x_i y_i c_i \end{cases}$$

POS (CNCF)
$$\begin{cases} c_{i+1} = (x_i+y_i+c_i)(x_i+\bar{y}_i+c_i)(x_i+y_i+\bar{c}_i)(\bar{x}_i+y_i+c_i) \\ z_i = (x_i+y_i+c_i)(x_i+\bar{y}_i+\bar{c}_i)(\bar{x}_i+y_i+\bar{c}_i)(\bar{x}_i+\bar{y}_i+c_i) \end{cases}$$

SOP (CNDF)
$$\begin{cases} c_{i+1} = \sum(m_3, m_5, m_6, m_7) \\ z_i = \sum(m_1, m_2, m_4, m_7) \end{cases}$$

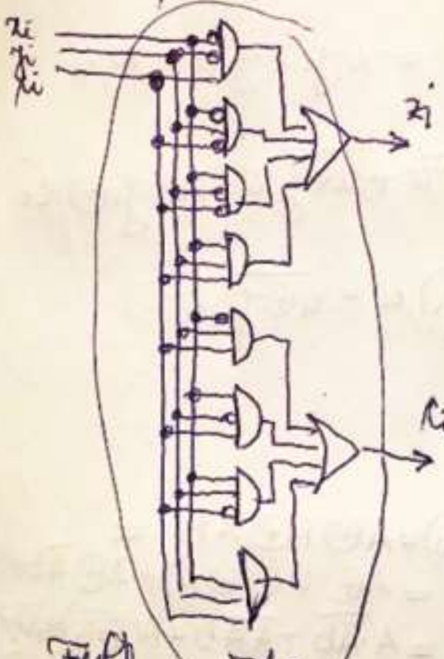
POS (CNCF)
$$\begin{cases} c_{i+1} = \prod(M_0, M_1, M_2, M_4) \\ z_i = \prod(M_0, M_3, M_5, M_6) \end{cases}$$

$$c_{i+1} = (x_i+y_i+c_i)(x_i+y_i+\bar{c}_i)(x_i+\bar{y}_i+c_i)(\bar{x}_i+y_i+c_i) = (x_i + \cancel{x_i y_i} + \cancel{x_i c_i} + \cancel{x_i y_i} + y_i + \cancel{y_i c_i} + \cancel{x_i \bar{c}_i} + y_i \bar{c}_i + c_i \bar{c}_i)(\underset{0}{x_i \bar{x}_i} + \bar{x}_i \bar{y}_i + \cancel{\bar{x}_i c_i} + x_i y_i + \underset{0}{y_i \bar{y}_i} + \cancel{y_i c_i} + \cancel{x_i c_i} + \cancel{\bar{y}_i c_i} + c_i) = (x_i+y_i)(\bar{x}_i \bar{y}_i + x_i y_i + c_i) =$$
$$= \underset{0}{x_i \bar{x}_i} y_i + \bar{x}_i \underset{0}{y_i \bar{y}_i} + x_i y_i + x_i y_i + x_i c_i + y_i c_i = x_i y_i + x_i c_i + y_i c_i$$

Problem statement → truth table construct → logic equation(s) written → $\boxed{\text{logic equation(s) minimization}}$ → logic circuit(s) drawn → logic circuit(s) built

SOP (CNDF)

$$\begin{cases} z_i = \bar{x}_i \bar{y}_i c_i + \bar{x}_i y_i \bar{c}_i + x_i \bar{y}_i \bar{c}_i + x_i y_i c_i \\ c_{i+1} = \bar{x}_i y_i c_i + x_i \bar{y}_i c_i + x_i y_i \bar{c}_i + x_i y_i c_i \end{cases}$$




Full adder cell

criteria used for minimization
- minimization of the number of variables
- minimization of the number of terms
- global minimization of the number of variables and the terms, so that their sum becomes minimal

methods for logic equation(s) minimization
- using algebraic methods based on the laws and the postulats of Boole's algebra
- using graphical method of Karnaugh (Karnaugh maps)
- using tabular method of Quine - McCluskey

Ex 1) $c_{i+1} = \underbrace{\bar x_i y_i c_i + x_i \bar y_i c_i + x_i y_i \bar c_i + x_i y_i c_i}_{\text{SOP(CNDF)}} + x_i y_i c_i + x_i y_i c_i =$

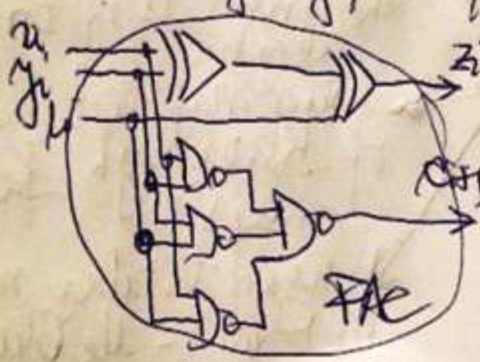
$= y_i c_i(\underbrace{\bar x_i + x_i}_{1}) + x_i c_i(\underbrace{\bar y_i + y_i}_{1}) + x_i y_i(\underbrace{\bar c_i + c_i}_{1}) = x_i y_i + y_i c_i + x_i c_i$

Ex 2) $z_i = \bar x_i \bar y_i c_i + \bar x_i y_i \bar c_i + x_i \bar y_i \bar c_i + x_i y_i c_i = (x_i \oplus y_i)\bar c_i + \overline{x_i \oplus y_i}\, c_i = x_i \oplus y_i \oplus c_i$

$\underbrace{(\bar x_i y_i + x_i \bar y_i)}_{x_i \oplus y_i}\bar c_i$

$(\bar x_i \bar y_i + x_i y_i) c_i = \overline{x_i \oplus y_i}\, c_i$

$c_{i+1} = \overline{\overline{x_i y_i + y_i c_i + x_i c_i}} = \overline{\overline{x_i y_i} \cdot \overline{y_i c_i} \cdot \overline{x_i c_i}}$



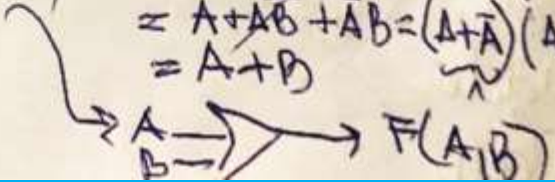

Ex. 3) $F(A,B) = A \oplus B \oplus AB =$

$= A \oplus B(1 \oplus A) = A \oplus \bar A B =$

$= A \cdot \overline{\bar A B} + \bar A \cdot \bar A B = A(\bar{\bar A} + \bar B) + \bar A B =$

$= A + A\bar B + \bar A B = (\underbrace{A + \bar A}_{1})(A+B)$

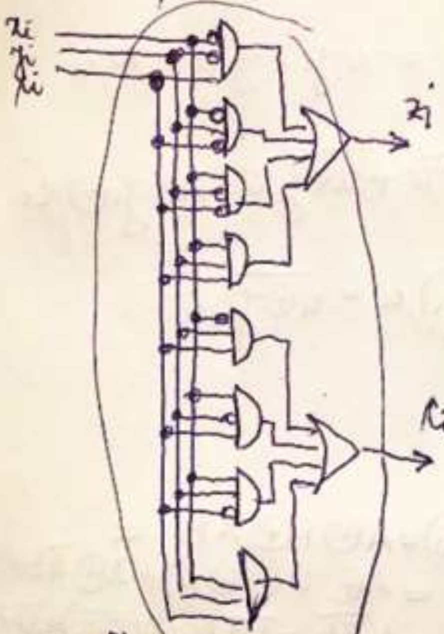
$= A + B$

Problem statement → truth table construct → logic equation(s) written → $\boxed{\text{logic equation(s) minimization}}$ → logic circuit(s) drawn → logic circuit(s) built

SOP (CNDF)

$$\begin{cases} z_i = \bar{x}_i \bar{y}_i c_i + \bar{x}_i y_i \bar{c}_i + x_i \bar{y}_i \bar{c}_i + x_i y_i c_i \\ c_{i+1} = \bar{x}_i y_i c_i + x_i \bar{y}_i c_i + x_i y_i \bar{c}_i + x_i y_i c_i \end{cases}$$




Full adder cell

FAC

criteria used for minimization

- minimization of the number of variables
- minimization of the number of terms
- global minimization of the number of variables and the terms, so that their sum becomes minimal

methods for logic equation(s) minimization

- using algebraic methods based on the laws and the postulats of Boole's algebra
- using graphical method of Karnaugh (Karnaugh maps)
- using tabular method of Quine-McCluskey

Ex. 1

$$c_{i+1} = \bar{x}_i y_i c_i + x_i \bar{y}_i c_i + x_i y_i \bar{c}_i + x_i y_i c_i + x_i y_i c_i + x_i y_i c_i =$$

SOP(CNDF)

$$= y_i c_i(\underbrace{\bar{x}_i + x_i}_{1}) + x_i c_i(\underbrace{\bar{y}_i + y_i}_{1}) + x_i y_i(\underbrace{\bar{c}_i + c_i}_{1}) = x_i y_i + y_i c_i + x_i c_i$$

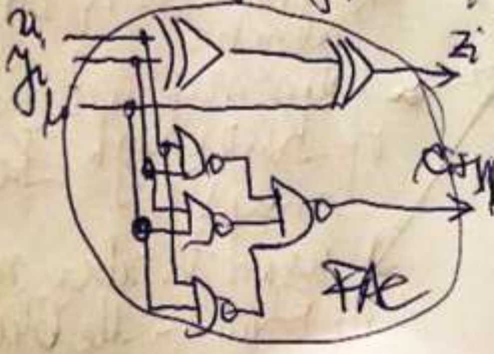
Ex. 2

$$z_i = \bar{x}_i \bar{y}_i c_i + \bar{x}_i y_i \bar{c}_i + x_i \bar{y}_i \bar{c}_i + x_i y_i c_i = (x_i \oplus y_i)\bar{c}_i + \overline{x_i \oplus y_i}\, c_i = x_i \oplus y_i \oplus c_i$$

$$(\underbrace{\bar{x}_i y_i + x_i \bar{y}_i}_{x_i \oplus y_i})\bar{c}_i$$

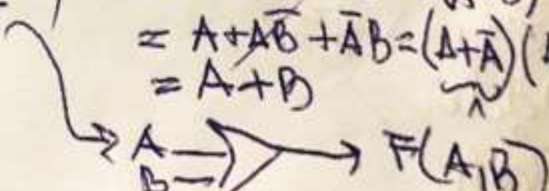
$$(\bar{x}_i \bar{y}_i + x_i y_i) c_i = \overline{x_i \oplus y_i}\, c_i$$

$$c_{i+1} = \overline{\overline{x_i y_i + y_i c_i + x_i c_i}} = \overline{\overline{x_i y_i} \cdot \overline{y_i c_i} \cdot \overline{x_i c_i}}$$




Ex. 3

$$F(A,B) = A \oplus B \oplus AB =$$
$$= A \oplus B(1 \oplus A) = A \oplus \bar{A}B =$$
$$= A \cdot \overline{\bar{A}B} + \bar{A} \cdot \bar{A}B = A(\bar{\bar{A}} + \bar{B}) + \bar{A}B =$$
$$= A + A\bar{B} + \bar{A}B = (\underbrace{A + \bar{A}}_{1})(A+B)$$
$$= A + B$$

# Karnaugh maps

| $x_i \backslash y_i$ | 0 | 1 |
|---|---|---|
| 0 | $m_0$ | $m_1$ |
| 1 | $m_2$ | $m_3$ |

| $x_i$ | $y_i$ | $m_i$ | $M_i$ |
|---|---|---|---|
| 0 | 0 | $\bar{x}_i \bar{y}_i$ | $x_i + y_i$ |
| 0 | 1 | $\bar{x}_i y_i$ | $x_i + \bar{y}_i$ |
| 1 | 0 | $x_i \bar{y}_i$ | $\bar{x}_i + y_i$ |
| 1 | 1 | $x_i y_i$ | $\bar{x}_i + \bar{y}_i$ |

$$\boxed{A + \bar{A} = 1}$$

| $x_i \backslash y_i z_i$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 1 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |

| $w_i x_i \backslash y_i z_i$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 01 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |
| 11 | $m_{12}$ | $m_{13}$ | $m_{15}$ | $m_{14}$ |
| 10 | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ |

| $AB \backslash CDE$ | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |
|---|---|---|---|---|---|---|---|---|
| 00 | $m_0$ | $m_1$ | $m_3$ | $m_2$ | $m_6$ | $m_7$ | $m_5$ | $m_4$ |
| 01 | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ | $m_{14}$ | $m_{15}$ | $m_{13}$ | $m_{12}$ |
| 11 | $m_{24}$ | $m_{25}$ | $m_{27}$ | $m_{26}$ | $m_{30}$ | $m_{31}$ | $m_{29}$ | $m_{28}$ |
| 10 | $m_{16}$ | $m_{17}$ | $m_{19}$ | $m_{18}$ | $m_{22}$ | $m_{23}$ | $m_{21}$ | $m_{20}$ |

$F(A,B,C,D) = \sum(m5, m7, m13, m15) = BD$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | 1 | 1 | |
| 11 | | 1 | 1 | |
| 10 | | | | |

$F(A,B,C,D) = \sum(m5, m6, m7, m13, m14, m15) = BC + BD$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | 1 | 1 | 1 |
| 11 | | 1 | 1 | 1 |
| 10 | | | | |

BC

BD

$F(A,B,C,D) = \sum(m3, m5, m6, m7, m13, m14, m15) = \bar{A}CD + BC + BD$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | |
| 01 | | 1 | 1 | 1 |
| 11 | | 1 | 1 | 1 |
| 10 | | | | |

$\bar{A}CD$

BC

BD

$$F(A,B,C,D) = \sum(m_8, m_{10}, m_{12}, m_{14}) = A\overline{D}$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | 1 | | | 1 |
| 10 | 1 | | | 1 |

$$F(A,B,C,D) = \sum(m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}) = \overline{A}B + A\overline{B} = A\oplus$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | 1 | 1 | 1 | 1 |
| 11 | | | | |
| 10 | 1 | 1 | 1 | 1 |

$$F(A,B,C,D) = \sum(m_0, m_2, m_7, m_8, m_{10}, m_{15}) = \overline{B}\,\overline{D} + BCD$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | | 1 |
| 01 | | | 1 | |
| 11 | | | 1 | |
| 10 | 1 | | | 1 |

BCD

$c_{i+1}$

| $x_i \backslash y_i c_i$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $m_0$ | $m_1$ | 1 ($m_3$) | $m_2$ |
| 1 | $m_4$ | 1 ($m_5$) | 1 ($m_7$) | 1 ($m_6$) |

$x_i c_i$ $\quad$ $y_i c_i$ $\quad$ $x_i y_i$

prime implicants

$$\Rightarrow c_{i+1} = \underbrace{\sum (m_3, m_5, m_6, m_7)}_{\text{SOP in CNDF}} = \underbrace{x_i y_i + y_i c_i + x_i c_i}_{\text{SOP of prime implicants}}$$

$z_i$

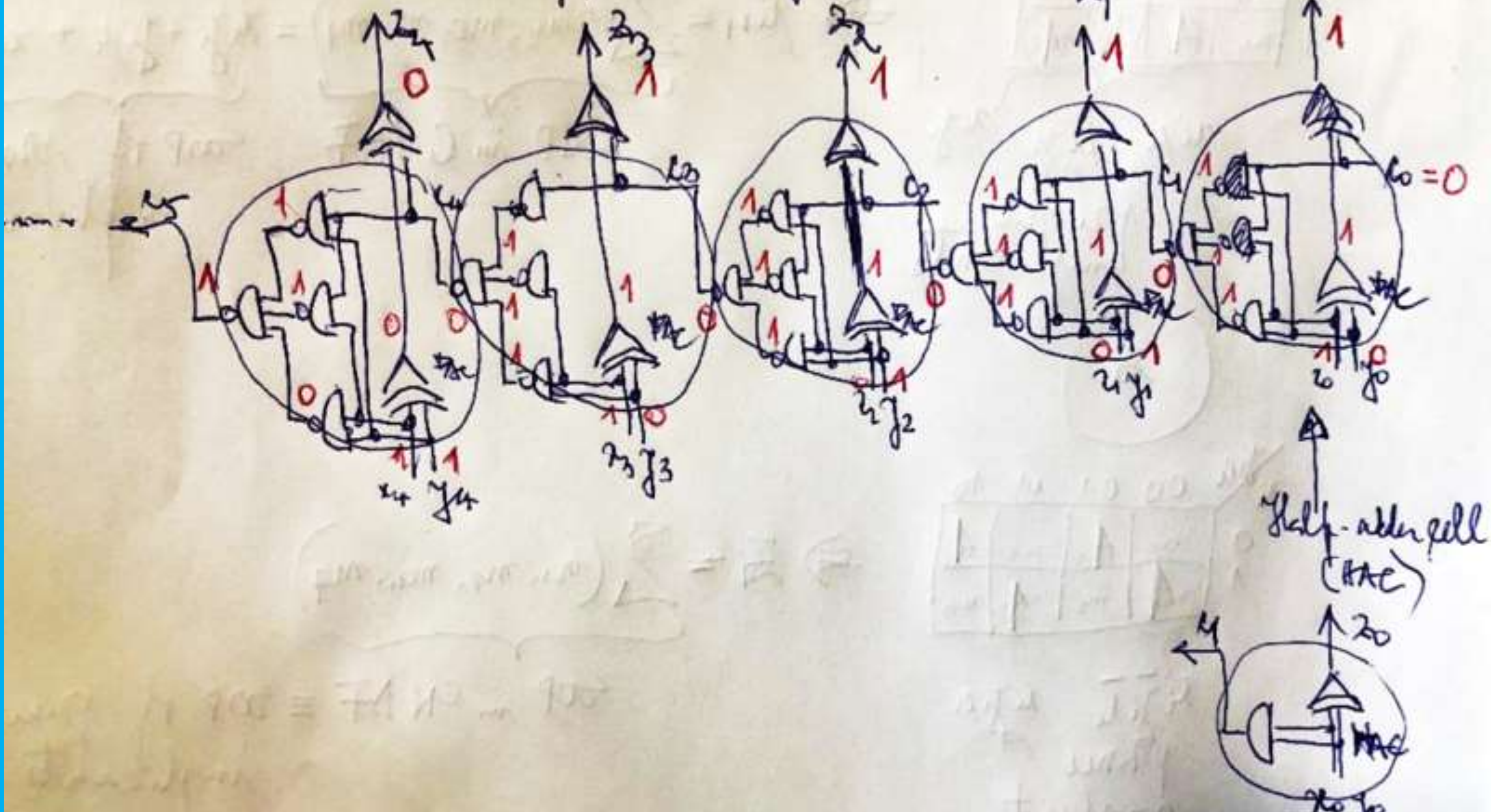
| $x_i \backslash y_i c_i$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $m_0$ | 1 ($m_1$) | $m_3$ | 1 ($m_2$) |
| 1 | 1 ($m_4$) | $m_5$ | 1 ($m_7$) | $m_6$ |

$x_i \bar{y}_i \bar{c}_i$ $\quad$ $x_i y_i c_i$

prime implicant

$$\Rightarrow z_i = \underbrace{\sum (m_1, m_2, m_4, m_7)}_{\text{SOP in CNDF} \,\equiv\, \text{SOP of prime implicants}}$$

# Ripple-carry adder (RCA)




$$\begin{cases} z_0 = x_0 \oplus y_0 \\ r = x_0 y_0 \end{cases}$$

$$F(W,X,Y,Z) = \sum(m_2, m_6, m_8, m_9, m_{10}, m_{11}, m_{14}, m_{15})$$

| WX \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 01 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |
| 11 | $m_{12}$ | $m_{13}$ | $m_{15}$ | $m_{14}$ |
| 10 | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ |

| WX \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | 1 |
| 01 | | | | 1 |
| 11 | | | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

$Y\bar{Z}$

$WY$

$W\bar{X}$

$$F(W,X,Y,Z) = W\bar{X} + WY + Y\bar{Z}$$

don't care

$$f(A,B,C,D) = \sum(m_4, m_8, m_{10}, m_{11}, m_{12}, m_{15}) + \sum_d (m_9, m_{14})$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | 1 | | | |
| 11 | 1 | | 1 | 1* |
| 10 | 1 | 1* | 1 | 1 |

$$f(A,B,C,D) = B\bar{C}\bar{D} + AC + A\bar{D} =$$

essential prime implicants

$$= B\bar{C}\bar{D} + AC + A\bar{B}$$

Binary to Three Excess Encoder

| Inputs | | | | Outputs | | | |
|---|---|---|---|---|---|---|---|
| A | b | c | D | W | X | Y | Z |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | d | d | d | d |
| 1 | 0 | 1 | 1 | d | d | d | d |
| 1 | 1 | 0 | 0 | d | d | d | d |
| 1 | 1 | 0 | 1 | d | d | d | d |
| 1 | 1 | 1 | 0 | d | d | d | d |
| 1 | 1 | 1 | 1 | d | d | d | d |

$$W = \sum(m_5, m_6, m_7, m_8, m_9) + \sum_d(m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15})$$

$$X = \sum(m_1, m_2, m_3, m_4, m_9) + \sum_d(m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15})$$

$Y =$ ----

$Z =$ ----

W

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | 1 | 1 | 1 |
| 11 | 1* | 1* | 1* | 1* |
| 10 | 1 | 1 | 1* | 1* |

$W = BC + BD + A\bar{C} =$
$= BC + BD + A\bar{B}$

essential prime implicants

X

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | 1 | 1 | 1 |
| 01 | 1 | | | |
| 11 | 1* | 1* | 1* | 1* |
| 10 | | 1 | 1* | 1* |

$X = B\bar{C}\bar{D} + \bar{B}D + \bar{B}C$

| Inputs A $2^1$ $a_1$ | A $2^0$ $a_0$ | B $2^1$ $b_1$ | B $2^0$ $b_0$ | Outputs $2^3$ W | $2^2$ X | $2^1$ Y | $2^0$ Z |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

$W = a_1 a_0 b_1 b_0$

X

| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | 1 | 1 |

$X = a_1 \bar{a}_0 b_1 + a_1 b_1 \bar{b}_0$

Y

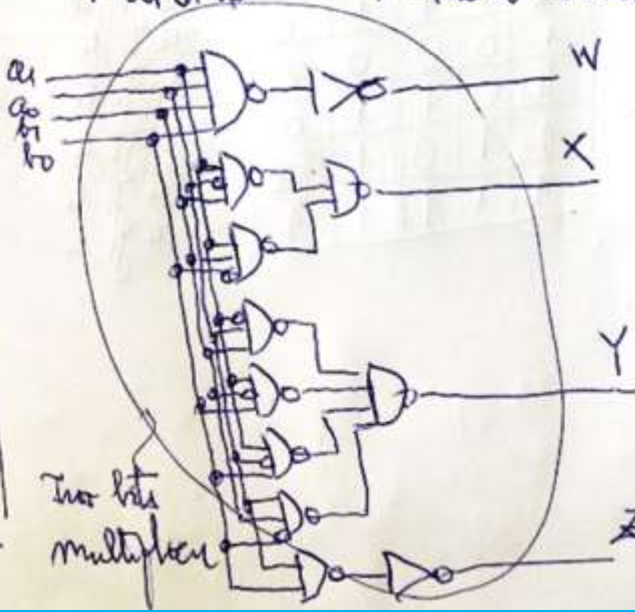
| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | 1 | 1 |
| 11 | | 1 | | 1 |
| 10 | | 1 | 1 | |

$Y = \bar{a}_1 a_0 b_1 + a_1 \bar{b}_1 b_0 + a_1 \bar{a}_0 b_0 + a_0 b_1 \bar{b}_0$

Z

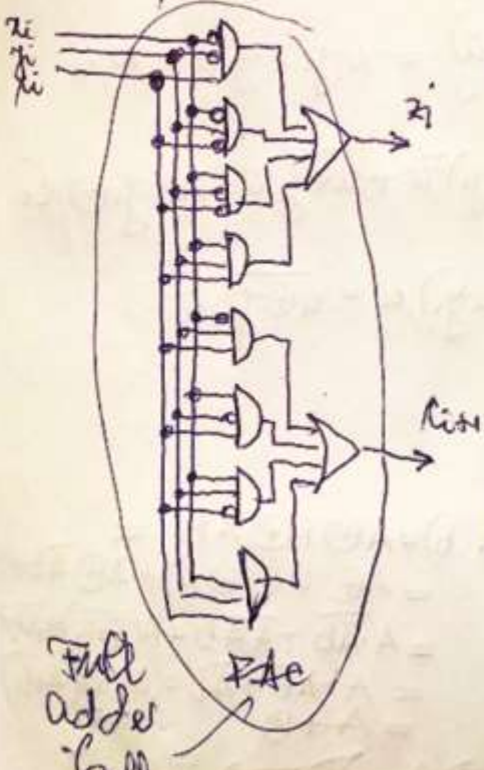
| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | 1 | 1 | |
| 11 | | 1 | 1 | |
| 10 | | | | |

$Z = a_0 b_0$



Problem statement → Truth table construct → Logic equation(s) written → $\boxed{\text{Logic equation(s) minimization}}$ → Logic circuit(s) drawn → Logic circuit(s) built

$$\text{SOP (CNDF)}\begin{cases} z_i = \bar{x}_i \bar{y}_i c_i + \bar{x}_i y_i \bar{c}_i + x_i \bar{y}_i \bar{c}_i + x_i y_i c_i \\ c_{i+1} = \bar{x}_i y_i c_i + x_i \bar{y}_i c_i + x_i y_i \bar{c}_i + x_i y_i c_i \end{cases}$$




Full adder cell

criteria used for minimization
- minimization of the number of variables
- minimization of the number of terms
- global minimization of the number of variables and the terms, so that their sum becomes minimal

methods for logic equation(s) minimization
- using algebraic methods based on the laws and the postulats of Boole's algebra
- using graphical method of Karnaugh (Karnaugh maps)
- using tabular method of Quine - Mc Cluskey

$$F(W,X,Y,Z) = \sum(m_2, m_6, m_8, m_9, m_{10}, m_{11}, m_{14}, m_{15})$$

| WX \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | $m_0$ | $m_1$ | $m_3$ | $m_2$ |
| 01 | $m_4$ | $m_5$ | $m_7$ | $m_6$ |
| 11 | $m_{12}$ | $m_{13}$ | $m_{15}$ | $m_{14}$ |
| 10 | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ |

| WX \ YZ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | 1 |
| 01 | | | | 1 |
| 11 | | | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 |

$Y\bar{Z}$

$WY$

$W\bar{X}$

$$F(W,X,Y,Z) = W\bar{X} + WY + Y\bar{Z}$$

don't care

$$f(A,B,C,D) = \sum(m_4, m_8, m_{10}, m_{11}, m_{12}, m_{15}) + \sum_d(m_9, m_{14})$$

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | 1 | | | |
| 11 | 1 | | 1 | 1* |
| 10 | 1 | 1* | 1 | 1 |

$$f(A,B,C,D) = B\bar{C}\bar{D} + AC + A\bar{D} =$$

essentiol prime implicants

$$= B\bar{C}\bar{D} + AC + A\bar{B}$$

## Binary to Three Excess Encoder

| Inputs | | | | Outputs | | | |
|---|---|---|---|---|---|---|---|
| A | B | C | D | W | X | Y | Z |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | d | d | d | d |
| 1 | 0 | 1 | 1 | d | d | d | d |
| 1 | 1 | 0 | 0 | d | d | d | d |
| 1 | 1 | 0 | 1 | d | d | d | d |
| 1 | 1 | 1 | 0 | d | d | d | d |
| 1 | 1 | 1 | 1 | d | d | d | d |

$W = \sum(m_5, m_6, m_7, m_8, m_9) + \sum_d(m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15})$

$X = \sum(m_1, m_2, m_3, m_4, m_9) + \sum_d(m_{10}, m_{11}, m_{12}, m_{13}, m_{14}, m_{15})$

$Y = \text{----}$

$Z = \text{----}$

W

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | 1 | 1 | 1 |
| 11 | 1* | 1* | 1* | 1* |
| 10 | 1 | 1 | 1* | 1* |

$W = BC + BD + A\bar{C} =$
$= BC + BD + A\bar{B}$

essential prime implicants

X

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | 1 | 1 | 1 |
| 01 | 1 | | | |
| 11 | 1* | 1* | 1* | 1* |
| 10 | | 1 | 1* | 1* |

$X = B\bar{C}\bar{D} + \bar{B}D + \bar{B}C$

| Inputs | | | | outputs | | | |
|---|---|---|---|---|---|---|---|
| A | | B | | | | | |
| $2^1$ $a_1$ | $2^0$ $a_0$ | $2^1$ $b_1$ | $2^0$ $b_0$ | $2^3$ W | $2^2$ X | $2^1$ Y | $2^0$ Z |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

$W = a_1 a_0 b_1 b_0$

X

| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | 1 |
| 10 | | | 1 | 1 |

$X = a_1 \bar{a}_0 b_1 + a_1 b_1 \bar{b}_0$

Y

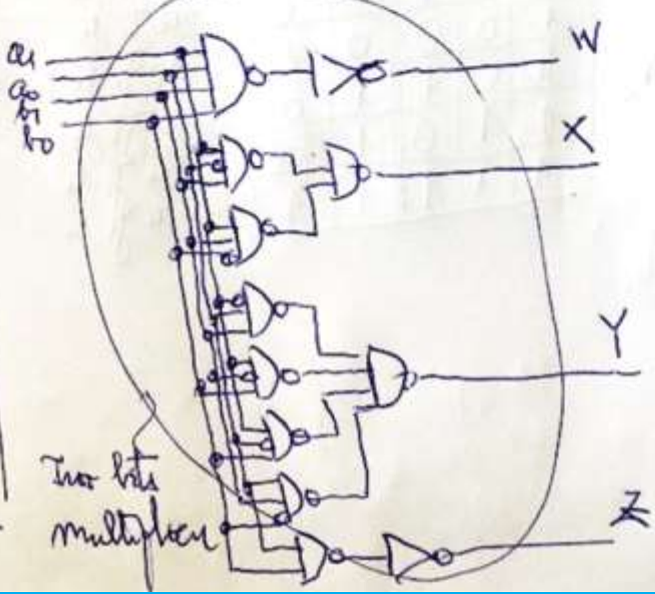
| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | 1 | 1 |
| 11 | | 1 | | 1 |
| 10 | | 1 | 1 | |

$Y = \bar{a}_1 a_0 b_1 + a_1 \bar{b}_1 b_0 + a_1 \bar{a}_0 b_0 + a_0 b_1 \bar{b}_0$

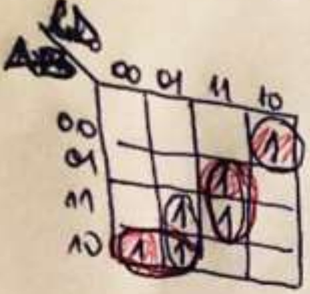
Z

| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | 1 | 1 | |
| 11 | | 1 | 1 | |
| 10 | | | | |

$Z = a_0 b_0$



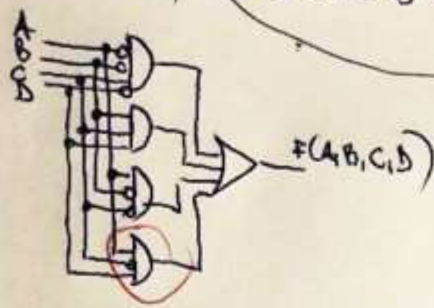
76x) $F(A,B,C,D) = \sum(m_2, m_7, m_8, m_9, m_{13}, m_{15}) = \prod(M_0, M_1, M_3, M_4, M_5, M_6, M_{10}, M_{11}, M_{12}, M_{14})$

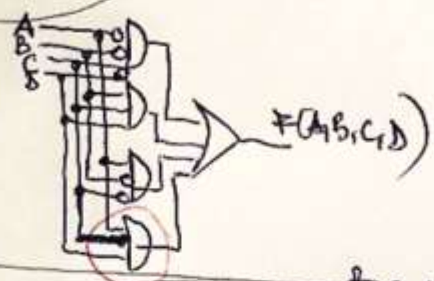


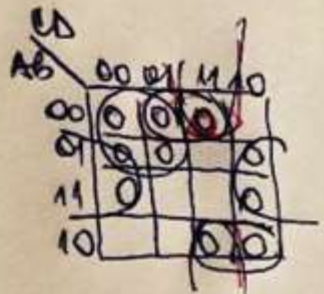

$F_1(A,B,C,D) = \bar{A}\bar{B}C\bar{D} + BCD + A\bar{B}\bar{C} + A\bar{C}D$

$F_2(A,B,C,D) = \bar{A}\bar{B}C\bar{D} + BCD + A\bar{B}\bar{C} + ABD$

essential prime implicants




$F_1'(A,B,C,D) = (A+C)(\bar{B}+D)(A+B+\bar{D})(\bar{A}+B+\bar{C})$ essential prime implicants

$F_2'(A,B,C,D) = (A+C)(\bar{B}+D)(\bar{A}+B+\bar{C})(\bar{B}+\bar{C}+\bar{D})$

essential prime implicants

$F_1'(A,B,C,D) = (A\bar{B} + \bar{B}C + AD + C\bar{D})(\bar{A}\bar{B} + \bar{A}\bar{D} + AB + B + B\bar{D} + A\bar{C} + B\bar{C} + \bar{C}D) =$

$= \bar{A}\bar{B}C\bar{D} + ABD + BCD + A\bar{B}\bar{C} + A\bar{C}D + A\bar{B}\bar{C}D +$

$AB\bar{C}D$ $\bar{A}BCD$ $A\bar{B}\bar{C}D$ $A\bar{B}\bar{C}(1+\bar{D})$ $1$

$ABCD$ $ABCD$ $A\bar{B}\bar{C}D$

$$F(W,X,Y,Z) = \sum (m_2, m_6, m_8, m_9, m_{10}, m_{11}, m_{14}, m_{15})$$

Step 1

| Minterm | W | X | Y | Z | Index |
|---|---|---|---|---|---|
| $m_2$ ✓ | 0 | 0 | 1 | 0 | 1 |
| $m_8$ ✓ | 1 | 0 | 0 | 0 | |
| $m_6$ ✓ | 0 | 1 | 1 | 0 | 2 |
| $m_9$ ✓ | 1 | 0 | 0 | 1 | |
| $m_{10}$ ✓ | 1 | 0 | 1 | 0 | |
| $m_{11}$ ✓ | 1 | 0 | 1 | 1 | 3 |
| $m_{14}$ ✓ | 1 | 1 | 1 | 0 | |
| $m_{15}$ ✓ | 1 | 1 | 1 | 1 | 4 |

The header "Input variables" spans the W, X, Y and Z columns.

Step 2

| Minterms | W | X | Y | Z | Index |
|---|---|---|---|---|---|
| $m_2, m_6$ ✓ | 0 | – | 1 | 0 | 1 |
| $m_2, m_{10}$ ✓ | – | 0 | 1 | 0 | |
| $m_8, m_9$ ✓ | 1 | 0 | 0 | – | |
| $m_8, m_{10}$ ✓ | 1 | 0 | – | 0 | |
| $m_6, m_{14}$ ✓ | – | 1 | 1 | 0 | 2 |
| $m_9, m_{11}$ ✓ | 1 | 0 | – | 1 | |
| $m_{10}, m_{11}$ ✓ | 1 | 0 | 1 | – | |
| $m_{10}, m_{14}$ ✓ | 1 | – | 1 | 0 | |

Step $2_1$

| Pairs of minterms | W | X | Y | Z | Index |
|---|---|---|---|---|---|
| $m_{11}, m_{15}$ ✓ | 1 | – | 1 | 1 | 3 |
| $m_{14}, m_{15}$ ✓ | 1 | 1 | 1 | – | |

Step $2_2$

| Pairs of minterms | W | X | Y | Z | Index |
|---|---|---|---|---|---|
| $m_2, m_6, m_{10}, m_{14}$ | – | – | 1 | 0 | 2 |
| ~~$m_2, m_{10}, m_6, m_{14}$~~ | ~~–~~ | ~~–~~ | ~~1~~ | ~~0~~ | |
| $m_8, m_{10}, m_9, m_{11}$ | 1 | 0 | – | – | |
| ~~$m_8, m_9, m_{10}, m_{11}$~~ | ~~1~~ | ~~0~~ | ~~–~~ | ~~–~~ | |
| $m_{10}, m_{11}, m_{14}, m_{15}$ | 1 | – | 1 | – | |
| ~~$m_{10}, m_{14}, m_{11}, m_{15}$~~ | ~~1~~ | ~~–~~ | ~~1~~ | ~~–~~ | |

| Prime implicants \ Minterms | $m_2$ | $m_6$ | $m_8$ | $m_9$ | $m_{10}$ | $m_{11}$ | $m_{14}$ | $m_{15}$ |
|---|---|---|---|---|---|---|---|---|
| $Y\bar{Z}$ | X | X | | | X | | X | |
| $W\bar{X}$ | | | X | X | X | X | | |
| $WY$ | | | | | X | X | X | X |

$$F(W,X,Y,Z) = Y\bar{Z} + W\bar{X} + WY$$

$$F(W,X,Y,Z) = \sum (m_2, m_6, m_8, m_9, m_{10}, m_{11}, m_{14}, m_{15})$$

| | Minterm | W | X | Y | Z | Index |
|---|---|---|---|---|---|---|
| | | Input variables | | | | |
| Step 1 | $m_2$ ✓ | 0 | 0 | 1 | 0 | 1 |
| | $m_8$ ✓ | 1 | 0 | 0 | 0 | |
| | $m_6$ ✓ | 0 | 1 | 1 | 0 | 2 |
| | $m_9$ ✓ | 1 | 0 | 0 | 1 | |
| | $m_{10}$ ✓ | 1 | 0 | 1 | 0 | |
| | $m_{11}$ ✓ | 1 | 0 | 1 | 1 | 3 |
| | $m_{14}$ ✓ | 1 | 1 | 1 | 0 | |
| | $m_{15}$ ✓ | 1 | 1 | 1 | 1 | 4 |
| Step 2 | $m_2, m_6$ ✓ | 0 | – | 1 | 0 | 1 |
| | $m_2, m_{10}$ ✓ | – | 0 | 1 | 0 | |
| | $m_8, m_9$ ✓ | 1 | 0 | 0 | – | |
| | $m_8, m_{10}$ ✓ | 1 | 0 | – | 0 | |
| | $m_6, m_{14}$ ✓ | – | 1 | 1 | 0 | 2 |
| | $m_9, m_{11}$ ✓ | 1 | 0 | – | 1 | |
| | $m_{10}, m_{11}$ ✓ | 1 | 0 | 1 | – | |
| | $m_{10}, m_{14}$ ✓ | 1 | – | 1 | 0 | |

| | Pairs of minterms | W | X | Y | Z | Index |
|---|---|---|---|---|---|---|
| Step $2_1$ | $m_{11}, m_{15}$ ✓ | 1 | – | 1 | 1 | 3 |
| | $m_{14}, m_{15}$ ✓ | 1 | 1 | 1 | – | |
| Step $2_2$ | $m_2, m_6, m_{10}, m_{14}$ | – | – | 1 | 0 | 2 |
| | ~~$m_2, m_{10}, m_6, m_{14}$~~ | ~~–~~ | ~~–~~ | ~~1~~ | ~~0~~ | |
| | $m_8, m_{10}, m_9, m_{11}$ | 1 | 0 | – | – | |
| | ~~$m_8, m_9, m_{10}, m_{11}$~~ | ~~1~~ | ~~0~~ | ~~–~~ | ~~–~~ | |
| | $m_{10}, m_{11}, m_{14}, m_{15}$ | 1 | – | 1 | – | |
| | ~~$m_{10}, m_{14}, m_{11}, m_{15}$~~ | ~~1~~ | ~~–~~ | ~~1~~ | ~~–~~ | |

| Prime implicants \ Minterms | $m_2$ | $m_6$ | $m_8$ | $m_9$ | $m_{10}$ | $m_{11}$ | $m_{14}$ | $m_{15}$ |
|---|---|---|---|---|---|---|---|---|
| $Y\bar{Z}$ | X | X | | | X | | X | |
| $W\bar{X}$ | | | X | X | X | X | | |
| $WY$ | | | | | X | X | X | X |

$$F(W,X,Y,Z) = Y\bar{Z} + W\bar{X} + WY$$

$$f(A,B,C,D) = \sum (m_4, m_8, m_{10}, m_{11}, m_{12}, m_{15}) + \sum_d (m_9, m_{14})$$

| | Minterms | A | B | C | D | Index |
|---|---|---|---|---|---|---|
| | | Input variables | | | | |
| Step 1 | ✓ $m_4$ | 0 | 1 | 0 | 0 | 1 |
| | ✓ $m_8$ | 1 | 0 | 0 | 0 | |
| | ✓ $m_9$ | 1 | 0 | 0 | 1 | 2 |
| | ✓ $m_{10}$ | 1 | 0 | 1 | 0 | |
| | ✓ $m_{12}$ | 1 | 1 | 0 | 0 | |
| | ✓ $m_{11}$ | 1 | 0 | 1 | 1 | 3 |
| | ✓ $m_{14}$ | 1 | 1 | 1 | 0 | |
| | ✓ $m_{15}$ | 1 | 1 | 1 | 1 | 4 |
| Step $2_1$ | * $m_4, m_{12}$ | — | 1 | 0 | 0 | 1 |
| | ✓ $m_8, m_9$ | 1 | 0 | 0 | — | |
| | ✓ $m_8, m_{10}$ | 1 | 0 | — | 0 | |
| | ✓ $m_8, m_{12}$ | 1 | — | 0 | 0 | |
| | ✓ $m_9, m_{11}$ | 1 | 0 | — | 1 | 2 |
| | ✓ $m_{10}, m_{11}$ | 1 | 0 | 1 | — | |
| | ✓ $m_{10}, m_{14}$ | 1 | — | 1 | 0 | |
| | ✓ $m_{12}, m_{14}$ | 1 | 1 | — | 0 | |

| | Pair of minterms | A | B | C | D | Index |
|---|---|---|---|---|---|---|
| | | Input variables | | | | |
| Step $2_1$ | ✓ $m_{11}, m_{15}$ | 1 | — | 1 | 1 | 3 |
| | ✓ $m_{14}, m_{15}$ | 1 | 1 | 1 | — | |
| Step $2_2$ | * $m_8, m_9, m_{10}, m_{11}$ | 1 | 0 | — | — | 1 |
| | ~~$m_8, m_{10}, m_9, m_{11}$~~ | ~~1~~ | ~~0~~ | ~~—~~ | ~~—~~ | |
| | * $m_8, m_{12}, m_{10}, m_{14}$ | 1 | — | — | 0 | |
| | ~~$m_8, m_{10}, m_{12}, m_{14}$~~ | ~~1~~ | ~~—~~ | ~~—~~ | ~~0~~ | |
| | * $m_{10}, m_{11}, m_{14}, m_{15}$ | 1 | — | 1 | — | 2 |
| | ~~$m_{10}, m_{14}, m_{11}, m_{15}$~~ | ~~1~~ | ~~—~~ | ~~1~~ | ~~—~~ | |

| | Prime implicants / Minterms | $m_4$ | $m_8$ | $m_{10}$ | $m_{11}$ | $m_{12}$ | $m_{15}$ |
|---|---|---|---|---|---|---|---|
| Step 3 | $B\bar{C}\bar{D}$ essential | * | | | | * | |
| | $A\bar{B}$ | | * | * | * | | |
| | $A\bar{D}$ | | * | * | | * | |
| | $AC$ essential | | | * | * | | * |

$f_1(A,B,C,D)$

$f_2(A,B,C,D)$

$$f_1(A,B,C,D) = \underbrace{B\bar{C}\bar{D} + AC}_{\text{essential prime implicants}} + A\bar{B}$$

$$f_2(A,B,C,D) = \underbrace{B\bar{C}\bar{D} + AC}_{\text{essential prime implicants}} + A\bar{D}$$

Subiect: Uzitând, pe de o parte, de porți NAND cu 2 și 3 intrări și, pe de altă parte, doar de porți NAND cu 2 intrări, să se proiecteze, folosind metoda Quine-McCluskey, circuitul logic combinațional corespunzător celei mai puțin semnificative ieșiri a unui convertor dinspre exces de 3 în cod 2 din 5.

Subiect: Fiind dată funcția logică (booleană) $S(w, x, y, z) = \sum(m_0, m_1, m_2, m_3, m_8, m_9, m_{11}, m_{15})$ și aplicând metodele Karnaugh, respectiv Quine-McCluskey, să se determine formele minimizate corespunzătoare expresiei date și să se implementeze uzitând de porți NOR cu 2 și 3 intrări.

Subiect: Se consideră expresia booleană $F(x_3, x_2, x_1, x_0) = \sum(m_1, m_2, m_5, m_7, m_8, m_9, m_{10}, m_{13}, m_{15})$ și se cere obținerea tuturor soluțiilor minimizate uzitând de metodele Karnaugh și Quine-McCluskey. Având la dispoziție porți NAND cu 2 și 3 intrări, să se realizeze implementarea soluțiilor minimizate.

# BCD Adder

$X = 278_{10} = 0010\ 0111\ 1000_{BCD}$

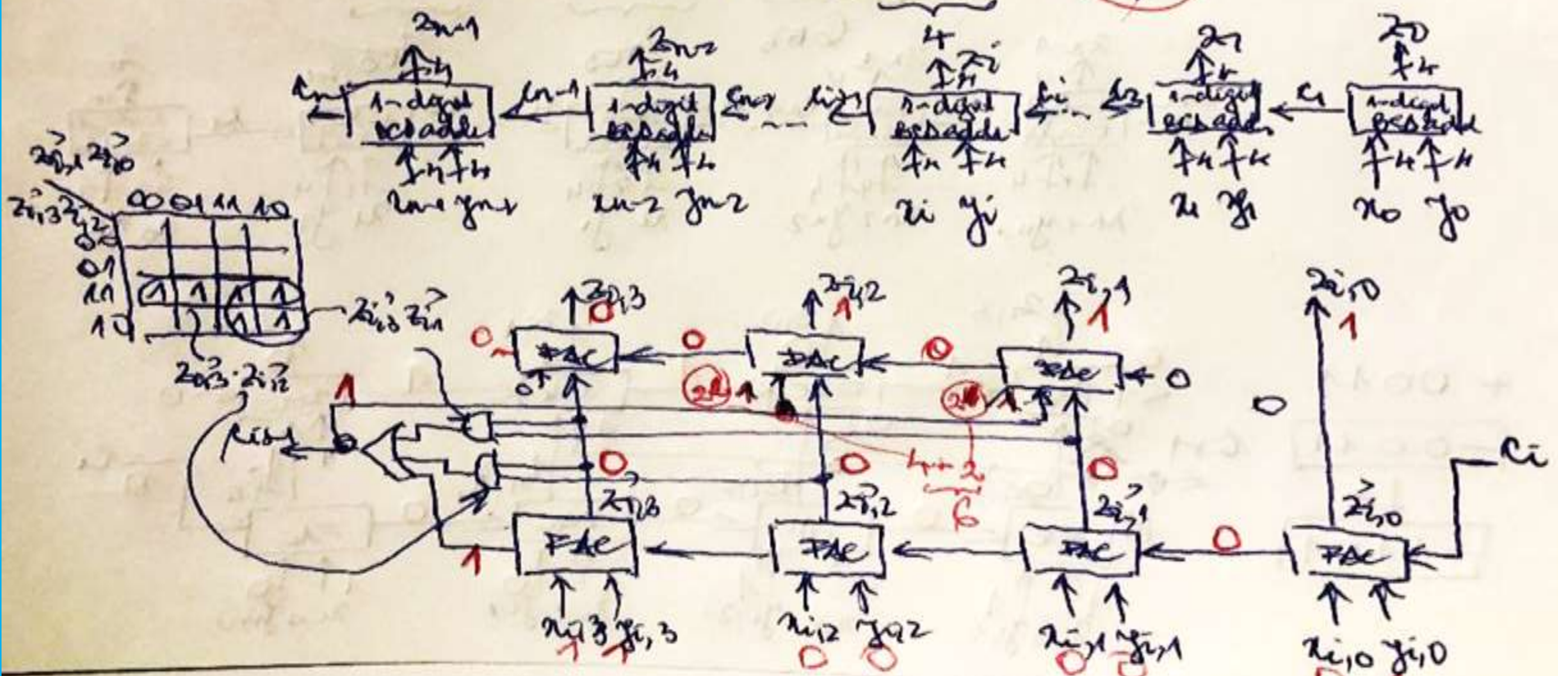
$Y = 369_{10} = 0011\ 0110\ 1001_{BCD}$ +

$Z = 647_{10}$ — $0101\ 1101\ 1\,0001$

$0001 + 0110 = 0111 \;\rightarrow\; 7$

$1101 + 1 = 1110$, $1110 + 0110 = 1\,0100 \;\rightarrow\; 4$

$0101 + 1 = 0110 \;\rightarrow\; 6$



# Excess-Three Adder

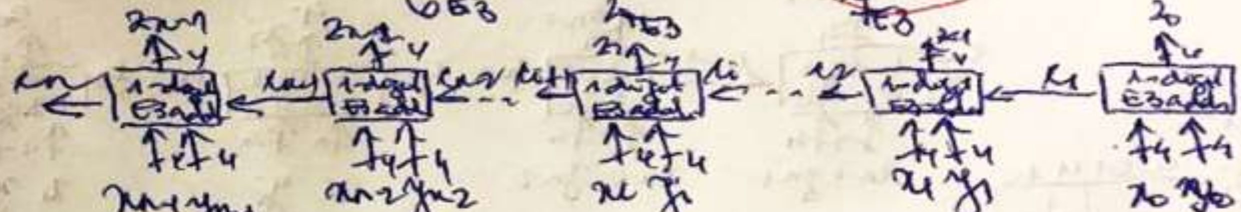
$X = 278_{10} = 0101\ 1010\ 1011_{E3}$ +

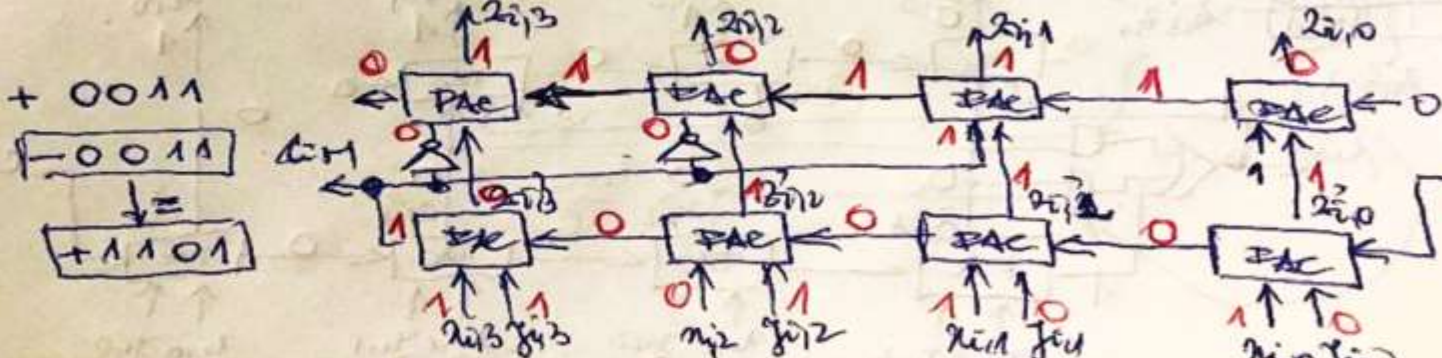
$Y = 369_{10} = 0110\ 1001\ 1100_{E3}$

$Z = 647_{10}$ $\quad 0\ 1100\ 1\ 0100\ 1\ 0111$

$\quad 0011 \quad 0011 \quad 0011$

$\quad \underbrace{1001}_{6_{E3}} \quad \underbrace{0111}_{4_{E3}} \quad \underbrace{1010}_{7_{E3}}$

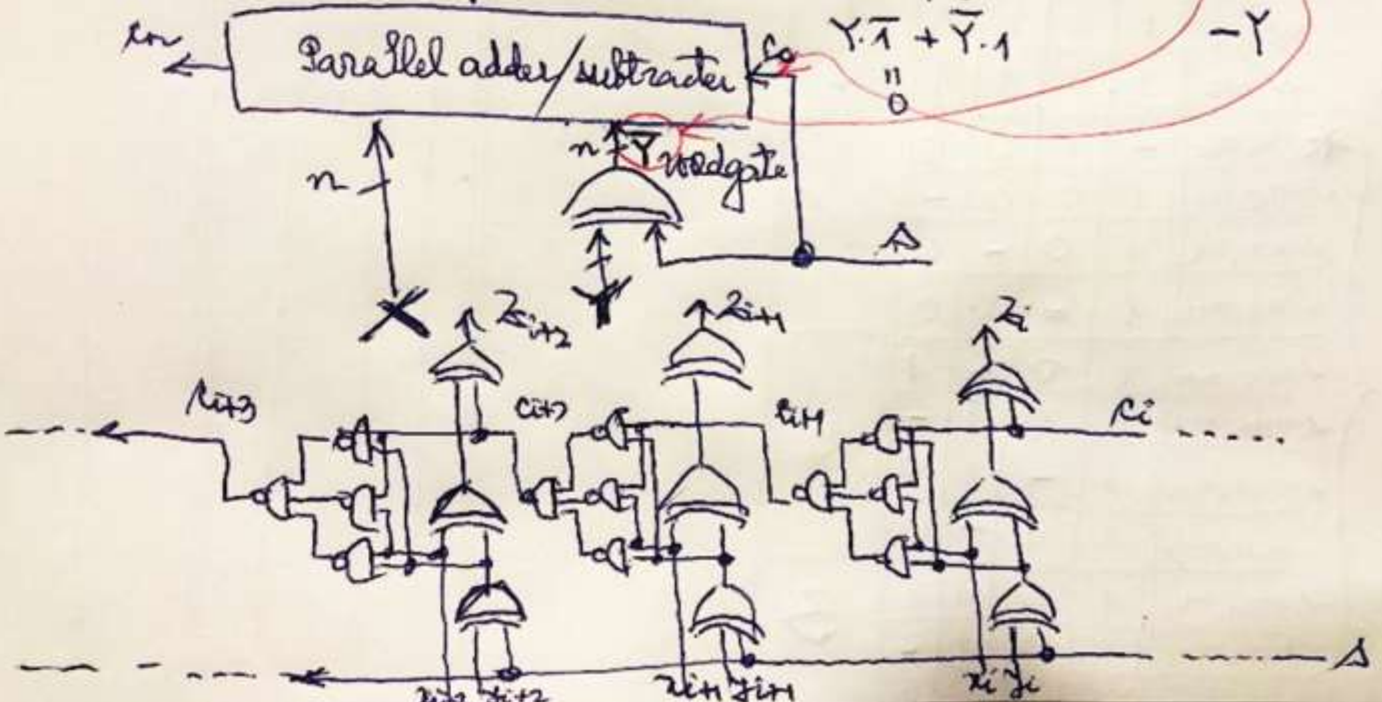


# Subtracters

## (1) Parallel adder/subtracter

$$Z = \begin{cases} \xrightarrow{s=0} X + Y \\ \xrightarrow{s=1} X + \underbrace{(Y \oplus 1)}_{Y\cdot\bar{1} + \bar{Y}\cdot 1} + 1 = X + \underbrace{\bar{Y} + 1}_{-Y} = X - Y \end{cases}$$



# Subtracters

## (2) Parallel subtracter

borrow

$X = 17_{10} = 10001_2 -$

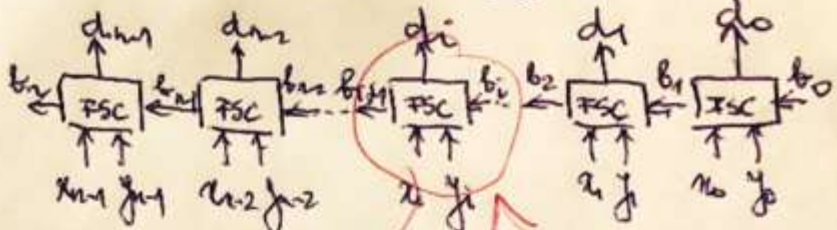
$b_{i+1} + b_i$

$Y = 13_{10} = 01101_2$

$D = 4_{10} \quad 00100$

$4_{10}$

FSC - Full Subtracter Cell




Truth table

$x_i - (y_i + b_i) \Rightarrow$

| $x_i$ | $y_i$ | $b_i$ | $b_{i+1}$ | $d_i$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Logic equations

$$d_i = \sum(m_1, m_2, m_4, m_7) = \bar{x}_i \bar{y}_i b_i + \bar{x}_i y_i \bar{b}_i + x_i \bar{y}_i \bar{b}_i + x_i y_i b_i =$$

$$= (\bar{x}_i \bar{y}_i + x_i y_i) b_i + (\bar{x}_i y_i + x_i \bar{y}_i) \bar{b}_i = x_i \oplus y_i \oplus b_i$$

$$b_{i+1} = \sum(m_1, m_2, m_3, m_7)$$

Logic equation minimization

| $x_i \backslash y_i b_i$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | 1 | 1 | 1 |
| 1 | | | 1 | |

$$\Rightarrow b_{i+1} = \bar{x}_i b_i + \bar{x}_i y_i + y_i b_i$$

Logic circuit drawn

# One's complement adder

$X = +38_{10} = 00100110_{2\text{-}SM}$
$Y = +42_{10} = 00101010_{2\text{-}SM}$
$Z = +80_{10}\ 0\ \underbrace{01010000}_{+80_{10}}{}_{2\text{-}SM}$

$X = +38_{10} = 00100110_{2\text{-}C1}$
$Y = -42_{10} = 11010101_{2\text{-}C1}$
$Z = -4_{10}\ 0\ 11111011_{2\text{-}C1}$

$\downarrow C1 \to SM$

$\underbrace{10000100}_{-4_{10}}$

$X = -38_{10} = 11011001_{2\text{-}C1}$
$Y = +42_{10} = 00101010_{2\text{-}C1}$
$Z = +4_{10}\ \boxed{1}\ \boxed{0}0000011$

end around carry 1

$\underbrace{00000100}_{+4_{10}}$

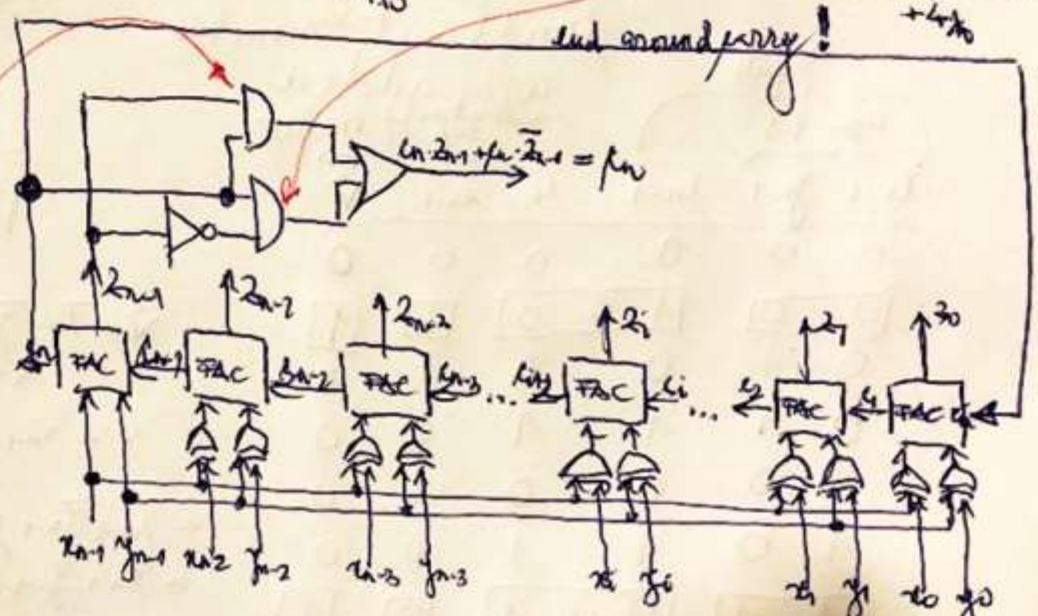
$X = -38_{10} = 11011001_{2\text{-}C1}$
$Y = -42_{10} = 11010101_{2\text{-}C1}$
$Z = -80_{10}\ \boxed{1}\ \boxed{1}0101110$

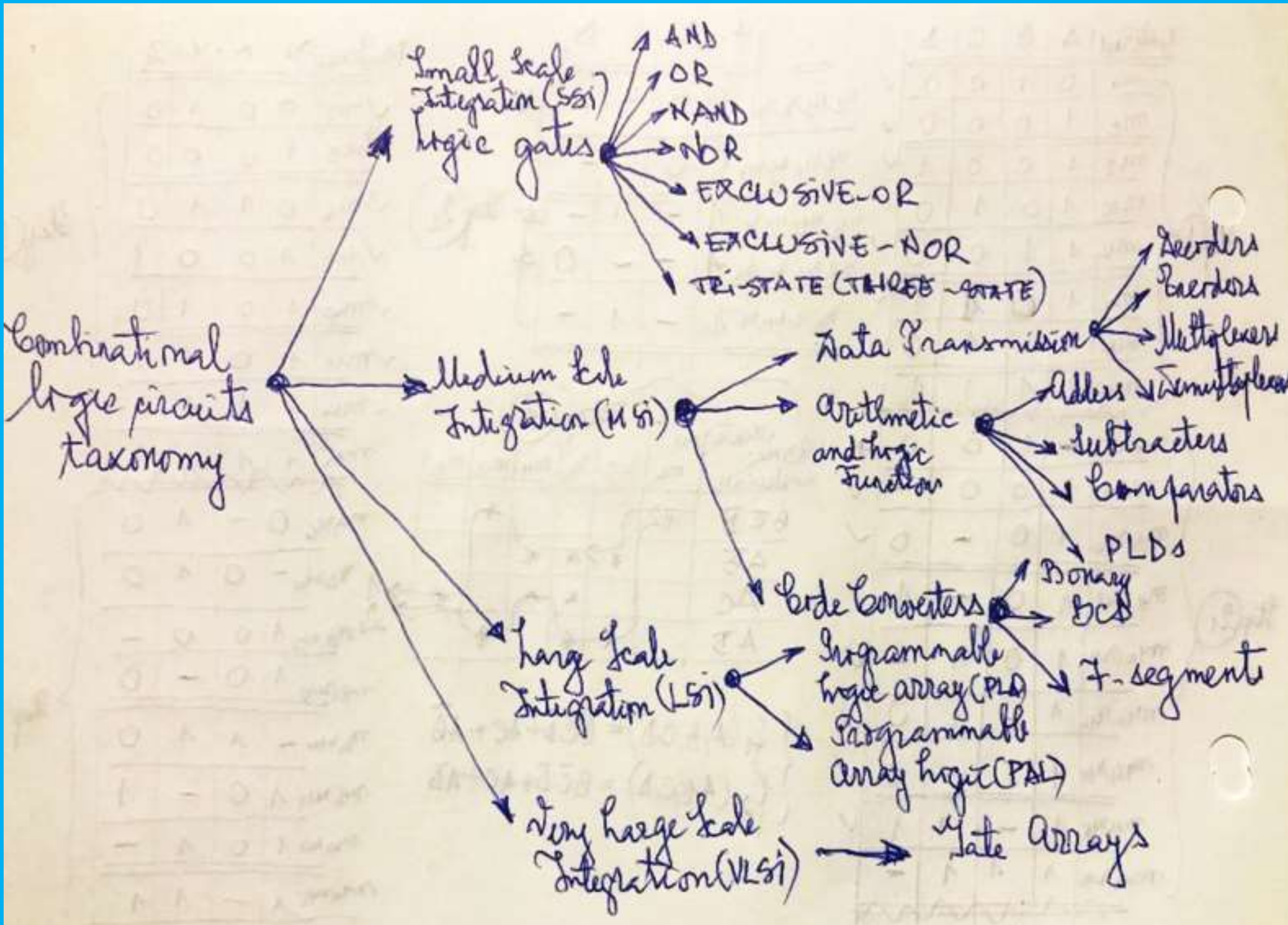
$\longrightarrow 1$

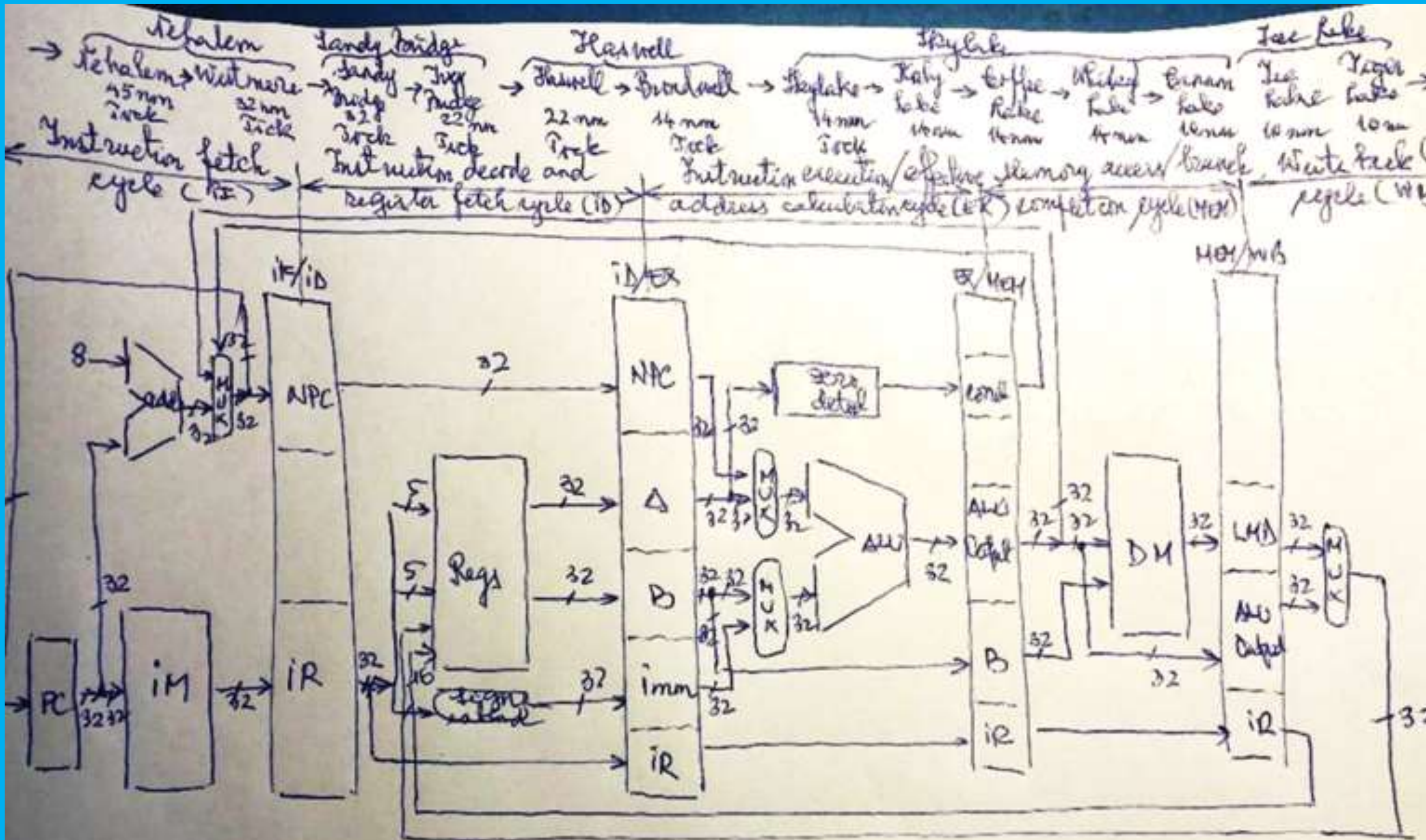
$10101111$

$\downarrow C1 \to SM$

$\underbrace{11010000}_{-80_{10}}$




Combinational logic circuits taxonomy
Small Scale Integration (SSI) logic gates
AND
OR
NAND
NOR
EXCLUSIVE-OR
EXCLUSIVE-NOR
TRI-STATE (THREE-STATE)
Medium Scale Integration (MSI)
Data Transmission
Decoders
Encoders
Multiplexers
Demultiplexers
Arithmetic and logic Function
Adders
Subtracters
Comparators
PLDs
Code Converters
Binary
BCD
7-segment
Large Scale Integration (LSI)
Programmable logic array (PLA)
Programmable array logic (PAL)
Very Large Scale Integration (VLSI)
Gate Arrays




PC - Program counter
iM - Instruction memory
iR - Instruction register
NPC - Next program counter

MUX - Multiplexer
Regs - Register file
A, B, imm - Buffer register
↑ Immediate

ALU - Arithmetic/logic unit
DM - Data memory
LMD - Load memory data register

# Addition overflow detection

**Definition:** Overflow is detected when two signed 2's complement numbers are added

- if both operands are positive and the sum is negative
- if both operands are negative and the sum is positive

inputs: sign bits — outputs: sign bit, overflow

| $x_{n-1}$ | $y_{n-1}$ | $c_{n-1}$ | $c_n$ | $z_{n-1}$ | $V$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| $\boxed{0}$ | $\boxed{0}$ | $\boxed{1}$ ⊕ | $\boxed{0}$ | $\boxed{1}$ | $\boxed{1}$ |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| $\boxed{1}$ | $\boxed{1}$ | $\boxed{0}$ ⊕ | $\boxed{1}$ | $\boxed{0}$ | $\boxed{1}$ |
| 1 | 1 | 1 | 1 | 1 | 0 |

$$\boxed{V = c_n \oplus c_{n-1}}$$

$$V = \overline{x}_{n-1}\overline{y}_{n-1}c_{n-1} + x_{n-1}y_{n-1}\overline{c}_{n-1}$$

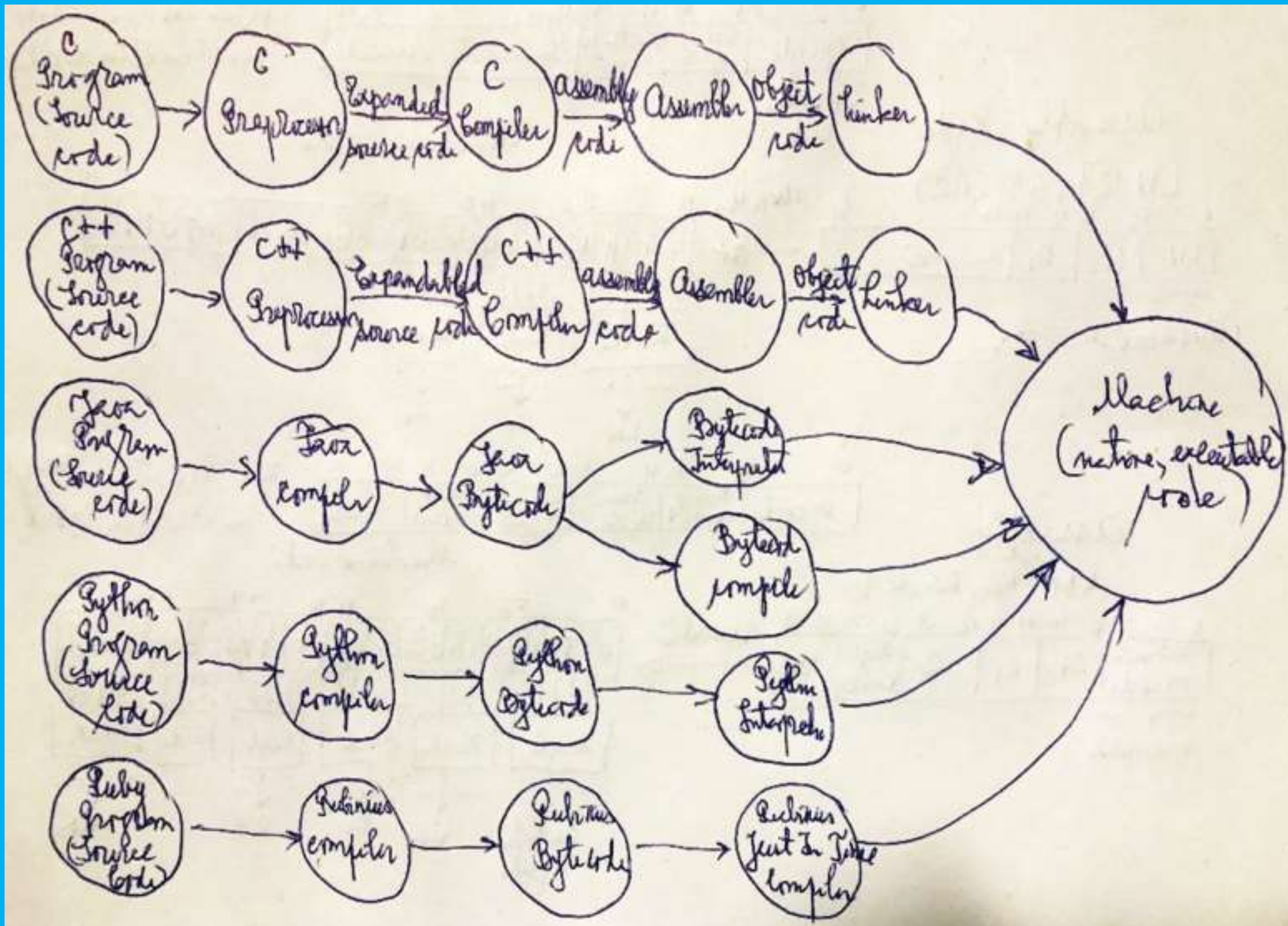
$$A + B = A \oplus B \oplus AB$$

$$V = \overline{x}_{n-1}\overline{y}_{n-1}c_{n-1} \oplus x_{n-1}y_{n-1}\underbrace{\overline{c}_{n-1}}_{1 \oplus c_{n-1}} =$$

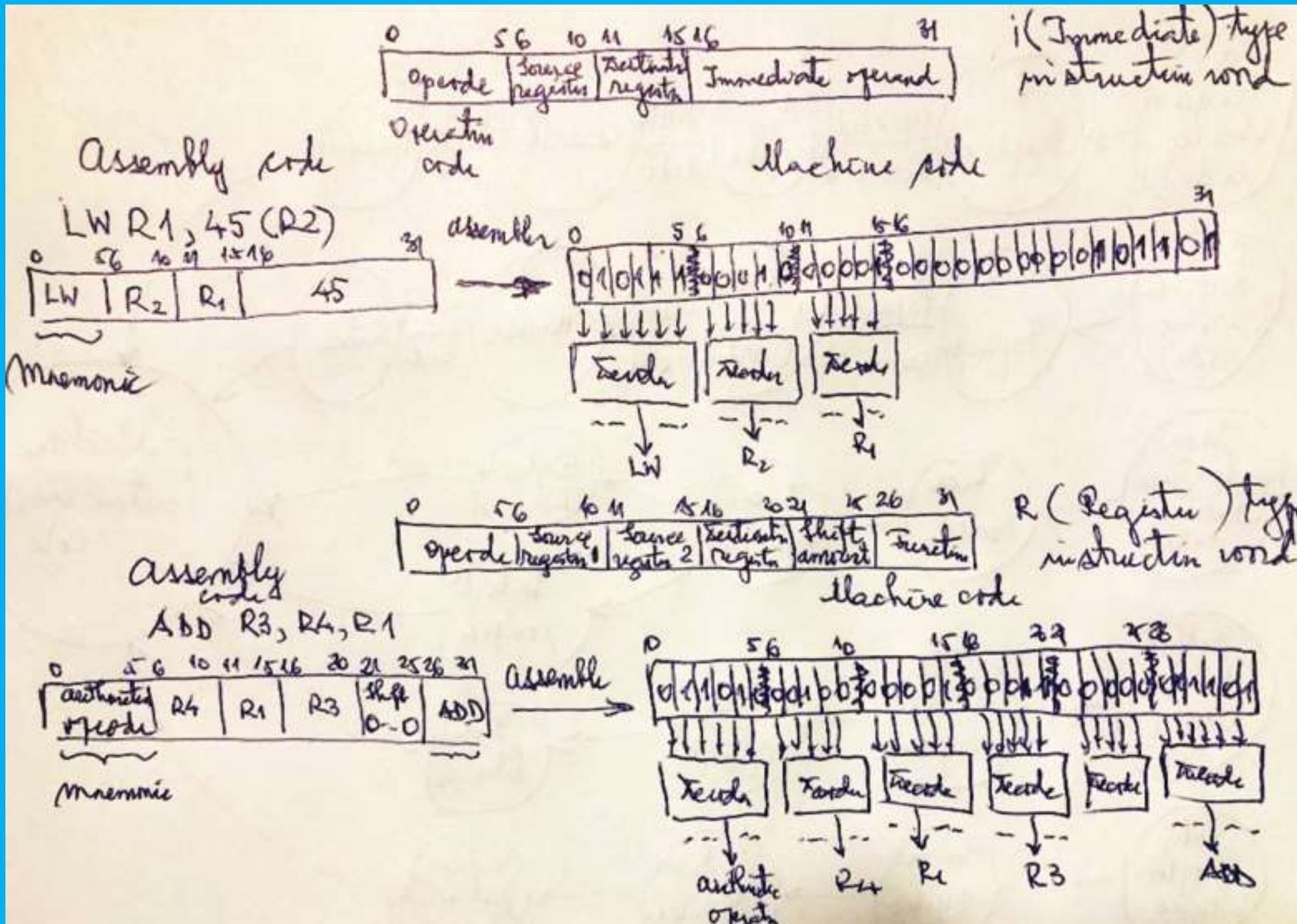
$$= \overline{x}_{n-1}\overline{y}_{n-1}c_{n-1} \oplus x_{n-1}y_{n-1} \oplus x_{n-1}y_{n-1}c_{n-1} =$$

$$= x_{n-1}y_{n-1} \oplus \underbrace{(\overline{x}_{n-1}\overline{y}_{n-1} \oplus x_{n-1}y_{n-1})}_{(x_{n-1} \oplus y_{n-1}) \oplus 1} c_{n-1} =$$

$$= \underbrace{x_{n-1}y_{n-1} \oplus x_{n-1}c_{n-1} \oplus y_{n-1}c_{n-1}}_{c_n} \oplus c_{n-1} = c_n \oplus c_{n-1}$$



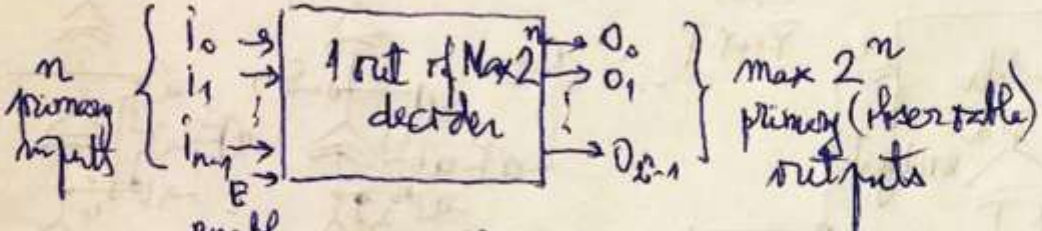
C Program (Source code)
C Preprocessor
Expanded source code
C Compiler
assembly code
Assembler
object code
Linker
C++ Program (Source code)
C++ Preprocessor
Expanded source code
C++ Compiler
assembly code
Assembler
object code
Linker
Java Program (Source code)
Java compiler
Java Bytecode
Bytecode Interpreter
Bytecode compiler
Machine (native, executable code)
Python Program (Source code)
Python compiler
Python Bytecode
Python Interpreter
Ruby Program (Source code)
Rubinius compiler
Rubinius Bytecode
Rubinius Just In Time compiler

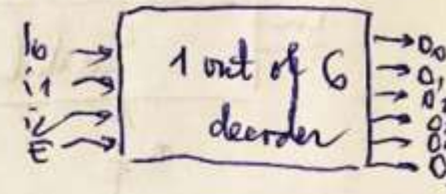

i (Immediate) type instruction word
0 5 6 10 11 15 16 31
Opcode
Source register
Destination register
Immediate operand
Operation code
Assembly code
Machine code
LW R1, 45 (R2)
assembler
LW R2 R1 45
Mnemonic
Decoder
LW
R2
R1
R (Register) type instruction word
Opcode
Source register 1
Source register 2
Destination register
Shift amount
Function
assembly code
ADD R3, R4, R1
arithmetic opcode
R4
R1
R3
Shift 0-0
ADD
assemble
Mnemonic
arithmetic operation
R4
R1
R3
ADD

# Decoders

* Where used?

Block diagram



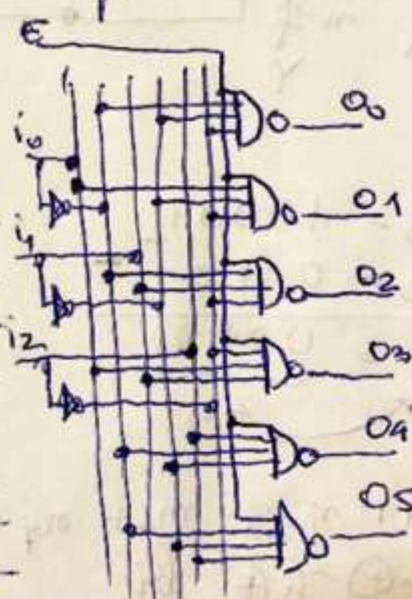



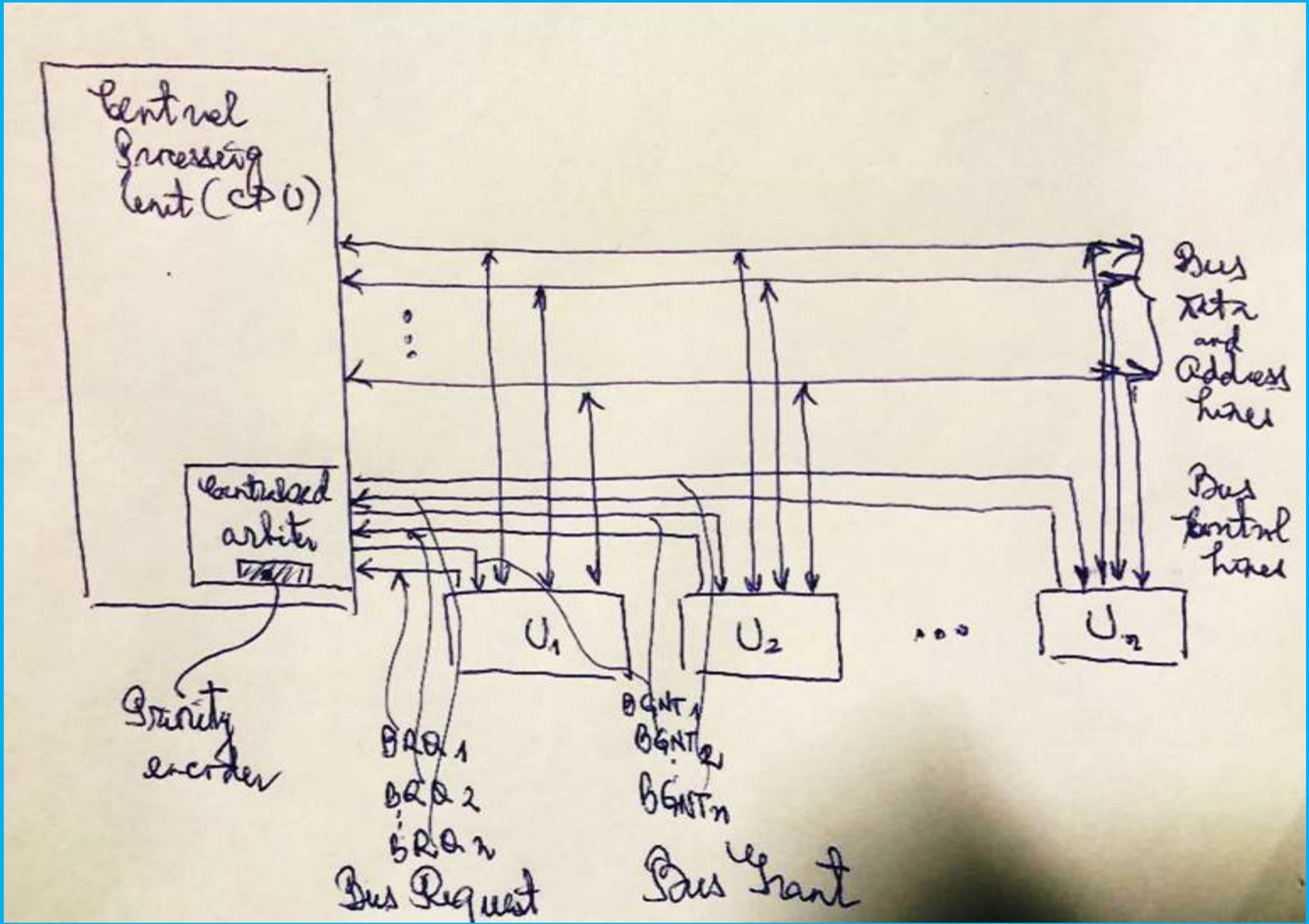

enable input

Truth table

don't care

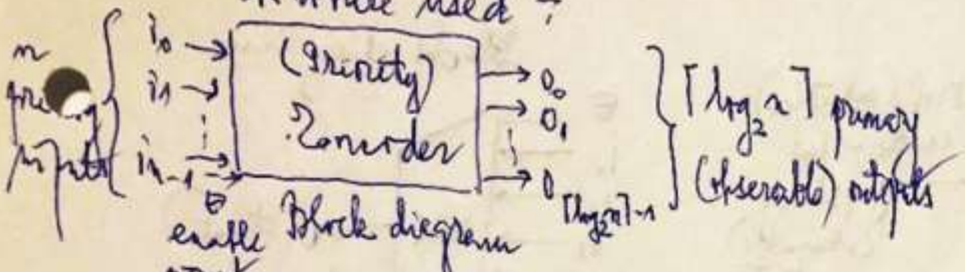
| E (enable input) | $i_2$ (primary inputs) | $i_1$ | $i_0$ | $O_5$ (primary outputs) | $O_4$ | $O_3$ | $O_2$ | $O_1$ | $O_0$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | x | x | x | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |




Central Processing Unit (CPU)
Centralized arbiter
Priority encoder
Bus Data and Address lines
Bus Control lines
$U_1$
$U_2$
$U_n$
BRQ 1
BRQ 2
BRQ n
Bus Request
BGNT 1
BGNT 2
BGNT n
Bus Grant

# Encoders

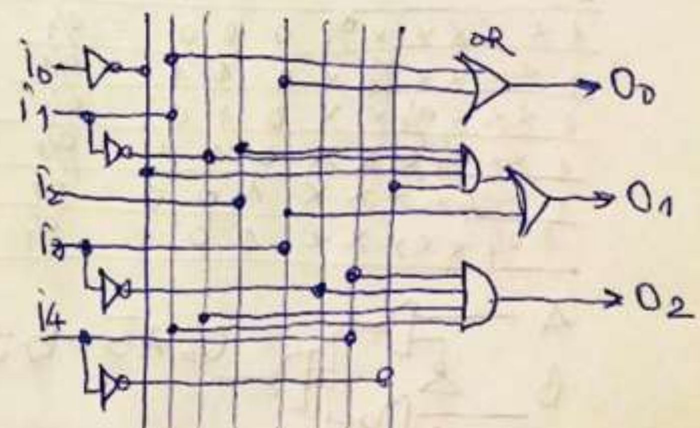
* where used ?





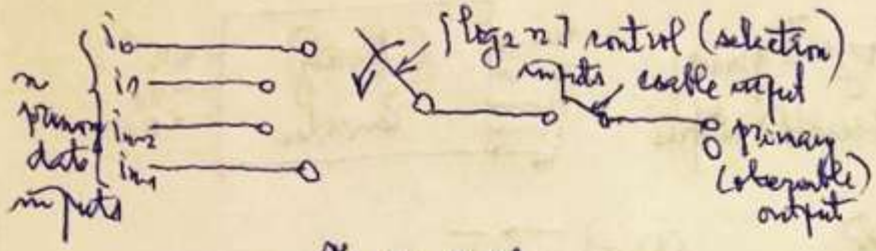
| | primary inputs | | | | | primary (observable) outputs | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $i_4$ | $i_3$ | $i_2$ | $i_1$ | $i_0$ | $O_2$ | $O_1$ | $O_0$ | |
| Priority: | 4 | 1 | 5 | 2 | 3 | $2^2$ | $2^1$ | $2^0$ | weight |
| | x | 1 | x | x | x | 0 | 1 | 1 | |
| | x | 0 | x | 1 | x | 0 | 0 | 1 | |
| | x | 0 | x | 0 | 1 | 0 | 0 | 0 | |
| | 1 | 0 | x | 0 | 0 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |

$$\begin{cases} O_2 = i_4 \cdot \bar{i}_3 \cdot \bar{i}_1 \cdot \bar{i}_0 \\ O_1 = i_3 + \bar{i}_4 \cdot \bar{i}_3 \cdot i_2 \cdot \bar{i}_1 \cdot \bar{i}_0 = i_3 + \bar{i}_4\, i_2 \bar{i}_1 \cdot \bar{i}_0 \\ O_0 = i_3 + \bar{i}_3 \cdot i_1 = i_3 + i_1 \end{cases}$$

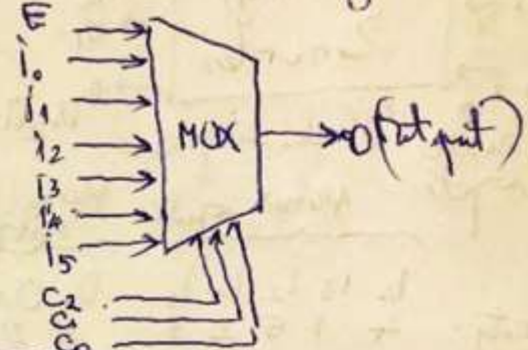


# Multiplexers

* where used?




Block diagram

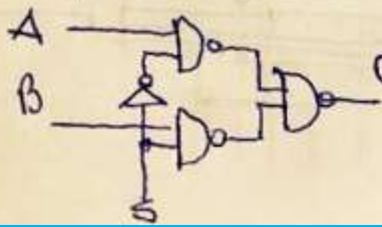



Truth table

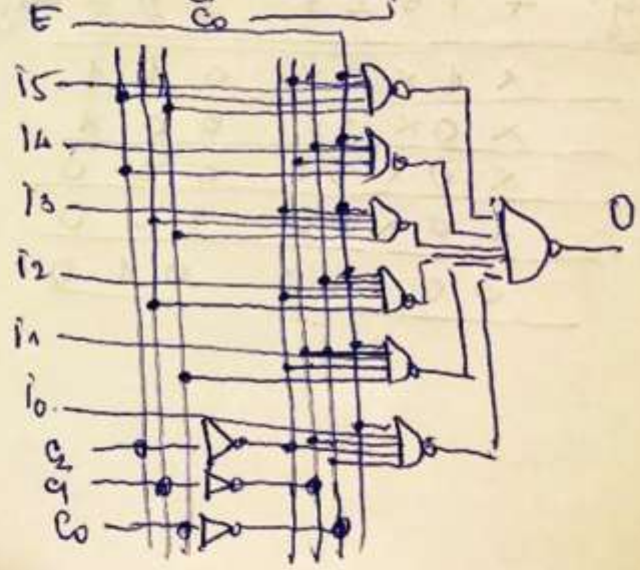
don't care

| E | $i_5$ | $i_4$ | $i_3$ | $i_2$ | $i_1$ | $i_0$ | $C_2$ | $C_1$ | $C_0$ | output |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | x | x | x | x | x | x | x | x | x | 0 |
| 1 | x | x | x | x | x | 0/1 | 0 | 0 | 0 | 0/1 |
| 1 | x | x | x | x | 0/1 | x | 0 | 0 | 1 | 0/1 |
| 1 | x | x | x | 0/1 | x | x | 0 | 1 | 0 | 0/1 |
| 1 | x | x | 0/1 | x | x | x | 0 | 1 | 1 | 0/1 |
| 1 | x | 0/1 | x | x | x | x | 1 | 0 | 0 | 0/1 |
| 1 | 0/1 | x | x | x | x | x | 1 | 0 | 1 | 0/1 |

(inputs: E, $i_5$ … $i_0$; control (selection): $C_2$ $C_1$ $C_0$)

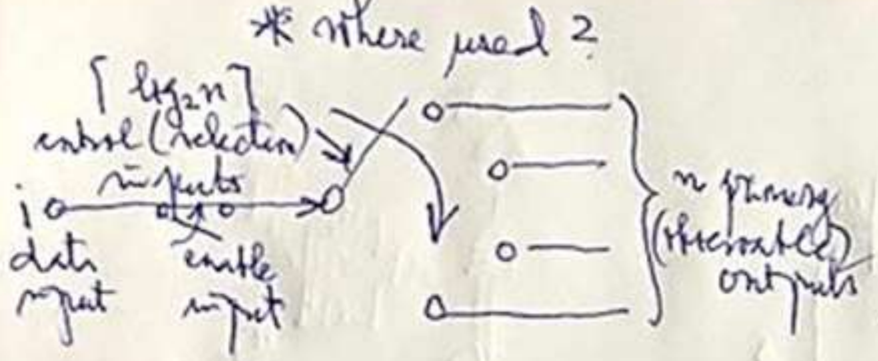


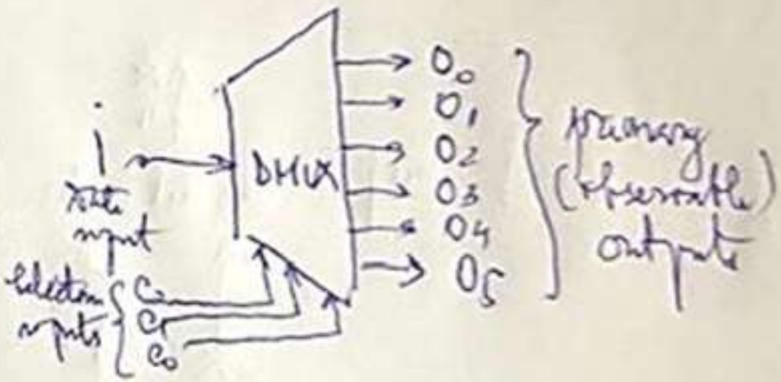

$O = A\bar{S} + B \cdot S$



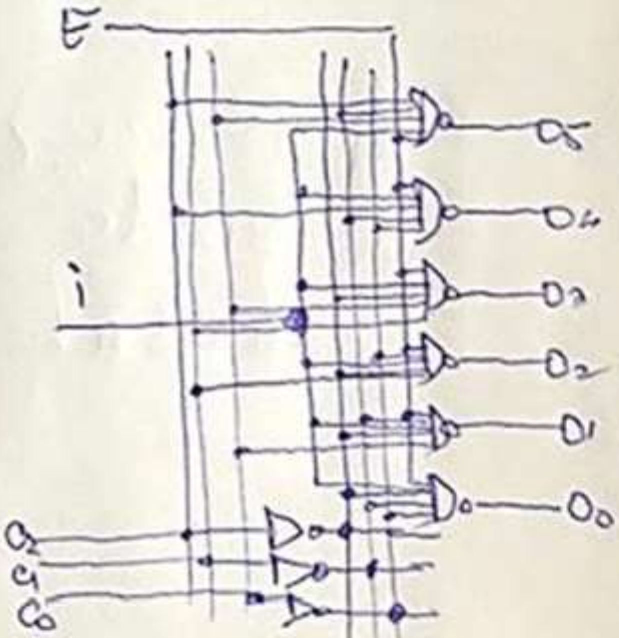
# Demultiplexers

* where used ?




| E | i | $C_2$ $C_1$ $C_0$ | $O_5$ | $O_4$ | $O_3$ | $O_2$ | $O_1$ | $O_0$ |
|---|---|---|---|---|---|---|---|---|
| 0 | x | x x x | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0/1 | 0 0 0 | 1 | 1 | 1 | 1 | 1 | 0/1 |
| 1 | 0/1 | 0 0 1 | 1 | 1 | 1 | 1 | 0/1 | 1 |
| 1 | 0/1 | 0 1 0 | 1 | 1 | 1 | 0/1 | 1 | 1 |
| 1 | 0/1 | 0 1 1 | 1 | 1 | 0/1 | 1 | 1 | 1 |
| 1 | 0/1 | 1 0 0 | 1 | 0/1 | 1 | 1 | 1 | 1 |
| 1 | 0/1 | 1 0 1 | 0/1 | 1 | 1 | 1 | 1 | 1 |



# Comparators

A :: B

| Inputs | | | | Outputs | | |
|---|---|---|---|---|---|---|
| A | | B | | | | |
| $2^1$ | $2^0$ | $2^1$ | $2^0$ | | | |
| $a_1$ | $a_0$ | $b_1$ | $b_0$ | A>B | A=B | A<B |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

A > B

| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | 1 | | | |
| 11 | 1 | 1 | | 1 |
| 10 | 1 | 1 | | |

A < B

| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | 1 | 1 | 1 |
| 01 | | | 1 | 1 |
| 11 | | | | |
| 10 | | | 1 | |

$$(A>B) = a_1\overline{b_1} + a_1a_0\overline{b_0} + \overline{a_0}\,\overline{b_1}\,\overline{b_0}$$

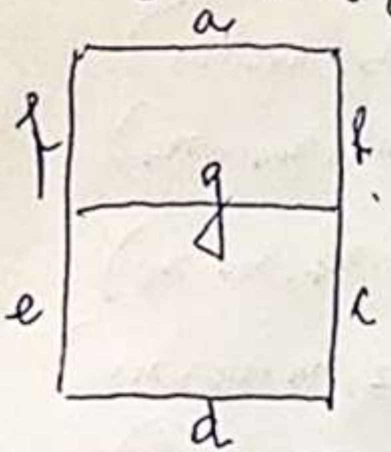
$$(A<B) = \overline{a_1}b_1 + \overline{a_1}\,\overline{a_0}b_0 + \overline{a_0}b_1b_0$$

A = B

| $a_1a_0$ \ $b_1b_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | | |
| 01 | | 1 | | |
| 11 | | | 1 | |
| 10 | | | | 1 |

$$(A=B) = \overline{a_1}\,\overline{a_0}\,\overline{b_1}\,\overline{b_0} + \overline{a_1}a_0\overline{b_1}b_0 + a_1a_0b_1b_0 + a_1\overline{a_0}b_1\overline{b_0}$$

# Binary to 7 Segments Converters




$a = 2+3+5+6+7+8+9+0$

$b = 1+2+3+4+7+8+9+0$

$c = 1+3+4+5+6+7+8+9+0$

$d = 2+3+5+6+8+9+0$

$e = 2+6+8+\cancel{+}0$

$f = 4+5+6+8+9+0$

$g = 2+3+4+5+6+8+9$

| A | B | C | D | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

(a)

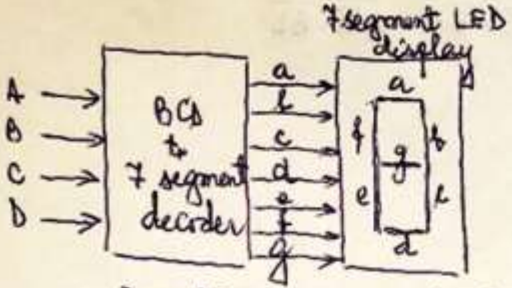
| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | 1 | 1 |
| 01 | | 1 | 1 | 1 |
| 11 | | | | |
| 10 | 1 | 1 | | |

$a = \bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C} + \bar{A}BD + \bar{A}C$

(g)

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | 1 | 1 | 0 | 1 |
| 11 | | | | |
| 10 | 1 | 1 | | |

$g = \bar{A}B\bar{D} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C}$




$$\begin{cases} a = 0+2+3+5+6+7+8+9 \\ b = 0+1+2+3+4+7+8+9 \\ c = 0+1+3+4+5+6+7+8+9 \\ d = 0+2+3+5+6+8+9 \\ e = 0+2+6+8 \\ f = 0+4+5+6+8+9 \\ g = 2+3+4+5+6+8+9 \end{cases}$$

Inputs; Outputs

| A ($2^3$) | B ($2^2$) | C ($2^1$) | D ($2^0$) | a | b | c | d | e | f | g |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | d | d | d | d | d | d | d |
| 1 | 0 | 1 | 1 | d | d | d | d | d | d | d |
| 1 | 1 | 0 | 0 | d | d | d | d | d | d | d |
| 1 | 1 | 0 | 1 | d | d | d | d | d | d | d |
| 1 | 1 | 1 | 0 | d | d | d | d | d | d | d |
| 1 | 1 | 1 | 1 | d | d | d | d | d | d | d |

a

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | 1 | 1 |
| 01 | | 1 | 1 | 1 |
| 11 | d | d | d | d |
| 10 | 1 | 1 | d | d |

$a = A + C + BD + \bar{B}\bar{D}$

c

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 1 | |
| 01 | 1 | 1 | 1 | 1 |
| 11 | d | d | d | d |
| 10 | 1 | 1 | d | d |

$c = B + \bar{C} + D$

e

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | | | 1 |
| 01 | | | | 1 |
| 11 | d | d | d | d |
| 10 | 1 | d | d | d |

$e = \bar{B}\bar{D} + C\bar{D}$

g

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | 1 | 1 |
| 01 | 1 | 1 | | 1 |
| 11 | d | d | d | d |
| 10 | 1 | 1 | d | d |

essential

$g = B\bar{C} + \bar{B}C + B\bar{D} + A\bar{C} =$

$= B\bar{C} + \bar{B}D + A\bar{B} + C\bar{D} =$

(g)

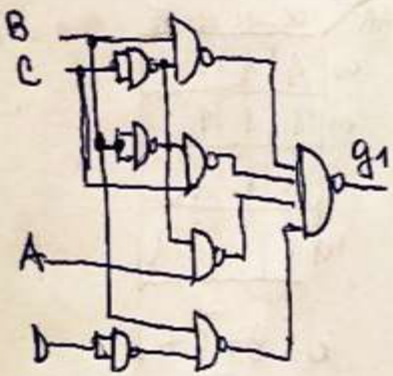
| AB \ CD | 00 | 01 | 11 | 10 |
| --- | --- | --- | --- | --- |
| 00 | | | 1 | 1 |
| 01 | 1 | 1 | | 1 |
| 11 | d | d | d | d |
| 10 | 1 | 1 | d | d |

$$g_1 = \overline{\overline{B\bar{C} + \bar{B}C + A\bar{C} + B\bar{D}}} = \overline{\overline{B\bar{C}} \cdot \overline{\bar{B}C} \cdot \overline{A\bar{C}} \cdot \overline{B\bar{D}}}$$

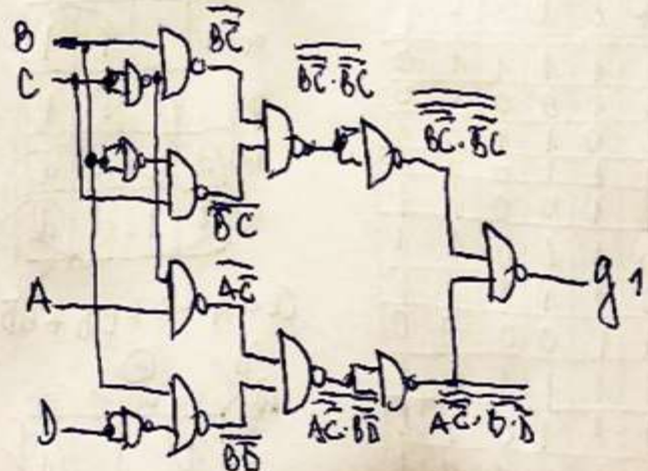
$$g_2 = B\bar{C} + \bar{B}C + A\bar{C} + C\bar{D}$$

$$g_3 = B\bar{C} + \bar{B}C + A\bar{B} + B\bar{D}$$

$$g_4 = B\bar{C} + \bar{B}C + A\bar{B} + C\bar{D}$$

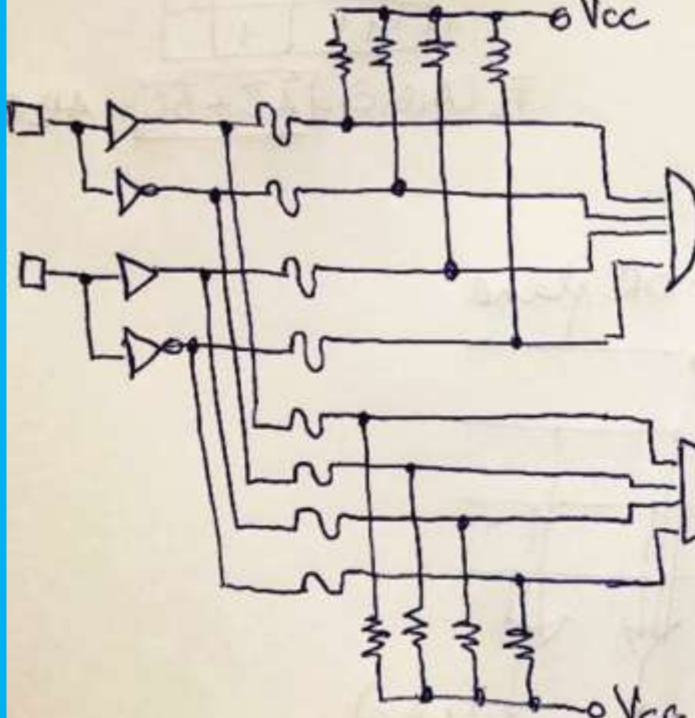



$$g_1 = \overline{\overline{B\bar{C} + \bar{B}C + A\bar{C} + B\bar{D}}} = \overline{\overline{B\bar{C} + \bar{B}C} \cdot \overline{A\bar{C} + B\bar{D}}} = \overline{\overline{\overline{\overline{B\bar{C}} \cdot \overline{\bar{B}C}}} \cdot \overline{\overline{\overline{A\bar{C}} \cdot \overline{B\bar{D}}}}} =$$

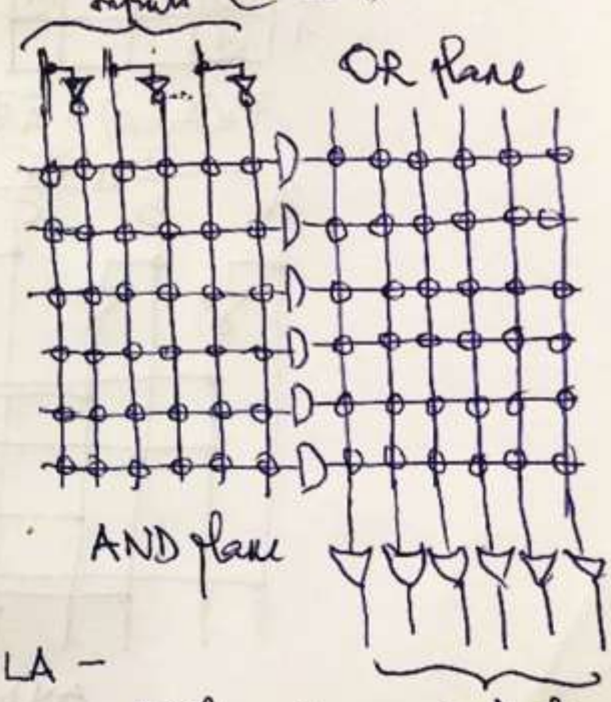


# Programmable Logic Devices (PLDs)

## Programmable Array Logic (PAL)




## Programmable Logic Array (PLA)




FPLA –
Field Programmable Logic Array

# PLA Implementation

$F_1(A,B,C) = \sum(m_0, m_1, m_2, m_4)$

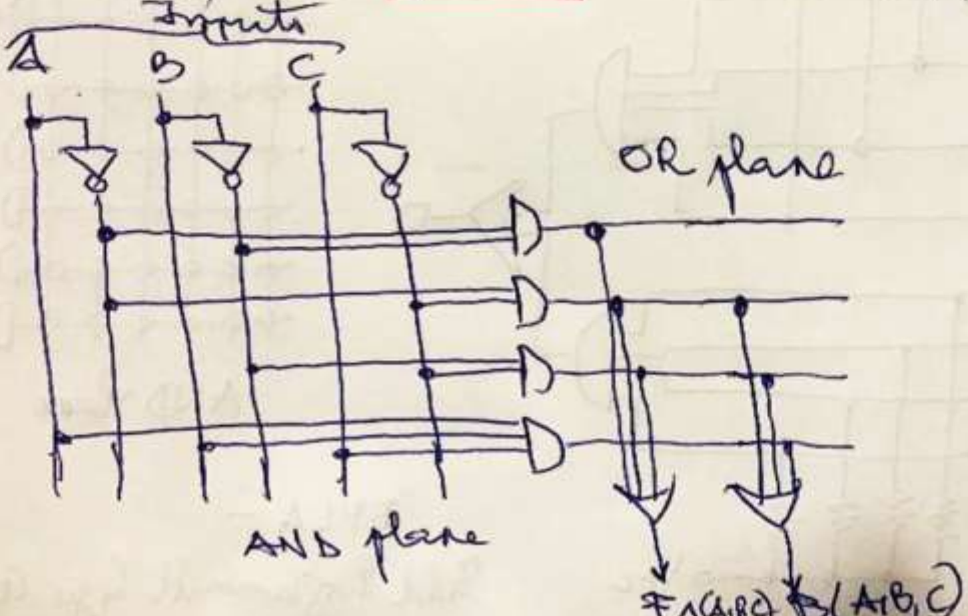
| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | | 1 |
| 1 | 1 | | | |

$F_1(A,B,C) = \bar{A}\bar{B} + \boxed{\bar{A}\bar{C} + \bar{B}\bar{C}}$

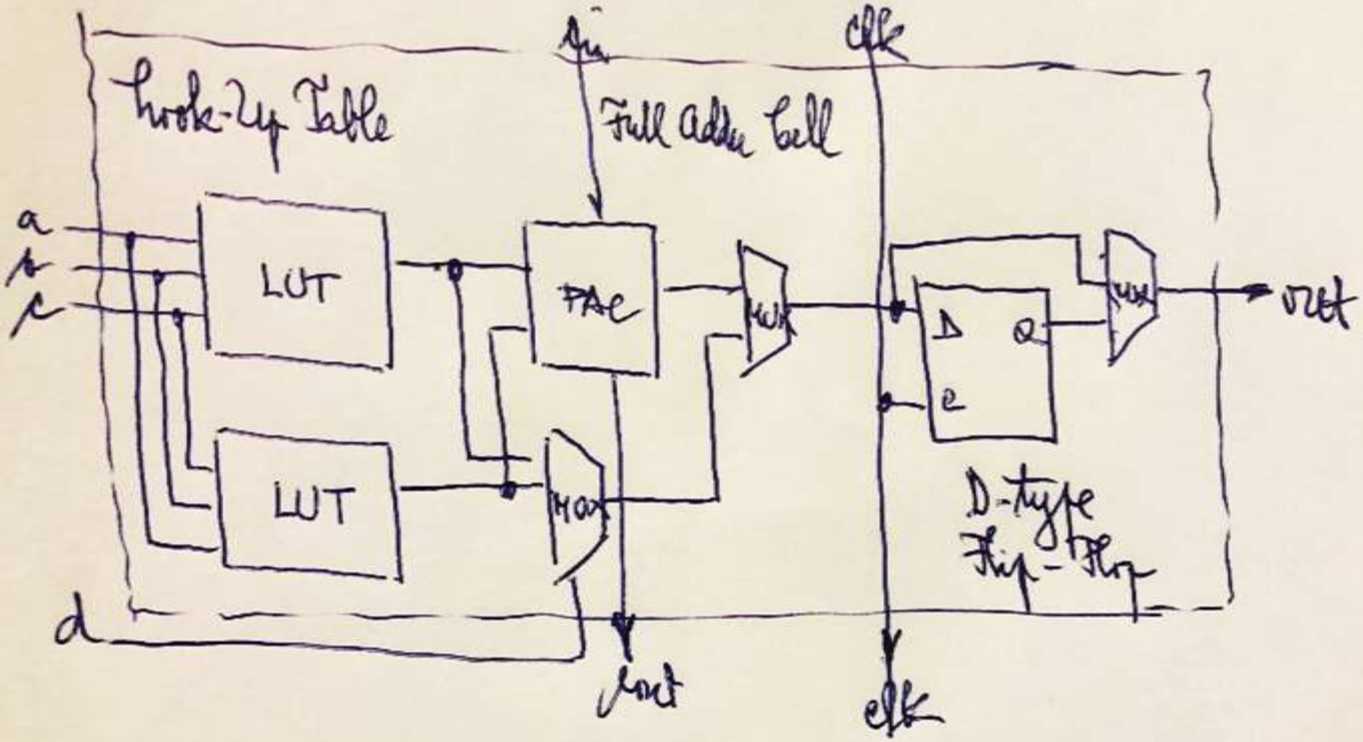
$F_2(A,B,C) = \sum(m_0, m_2, m_4, m_7)$

| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | | | 1 |
| 1 | 1 | | 1 | |

$F_2(A,B,C) = \boxed{\bar{A}\bar{C} + \bar{B}\bar{C}} + ABC$

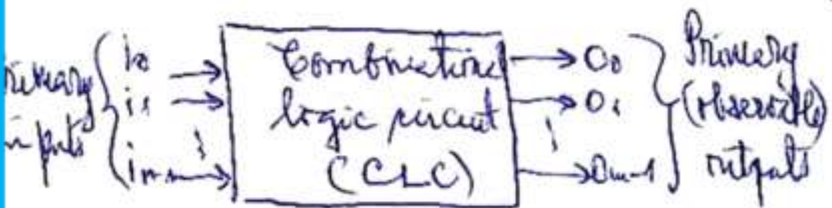


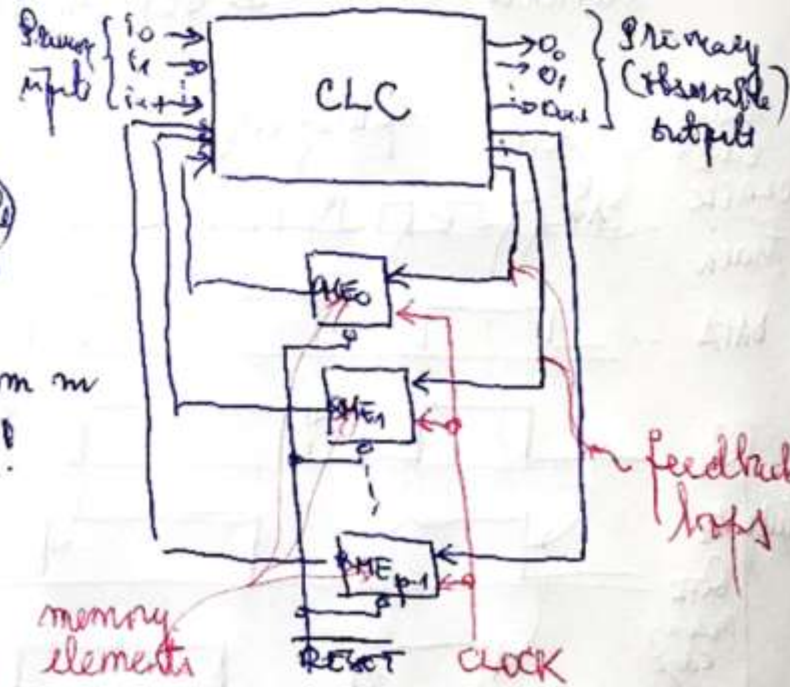
* Field Programmable Gate Array (FPGA)




* Application Specific Integrated Circuit (ASIC)

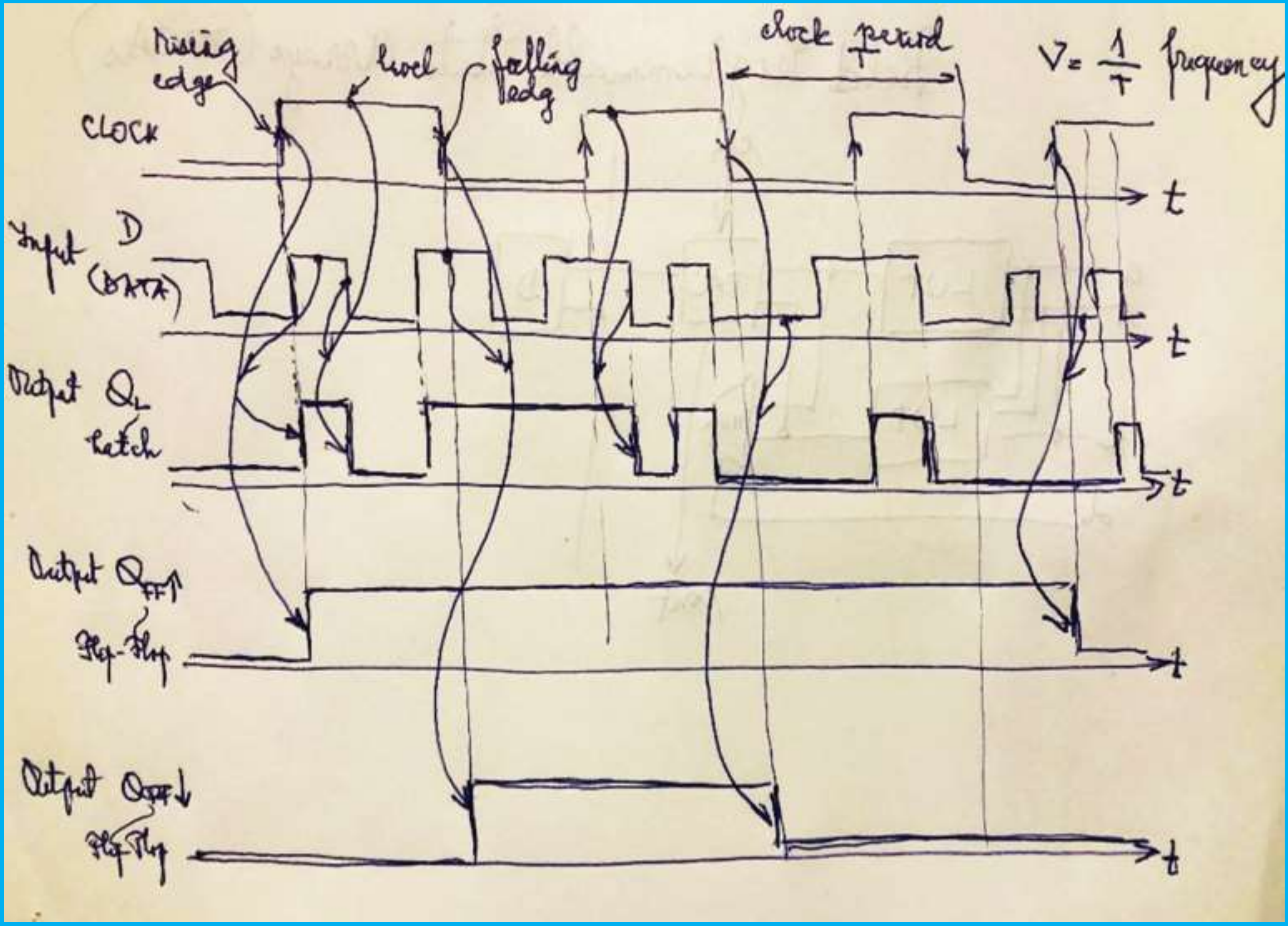
# Combinational vs sequential logic circuits



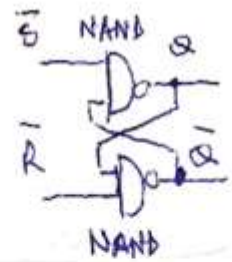

* n in general different from m
* no memory elements!
* no feedback loops

* Sequential logic circuit
  - Synchronous (with CLOCK)
  - Asynchronous (without CLOCK)

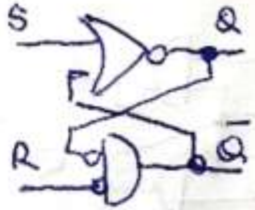


rising edge
level
falling edge
clock period T
CLOCK
t
Input D (DATA)
Output $Q_L$ Latch
Output $Q_{FF}\uparrow$ Flip-Flop
Output $Q_{FF}\downarrow$ Flip-Flop

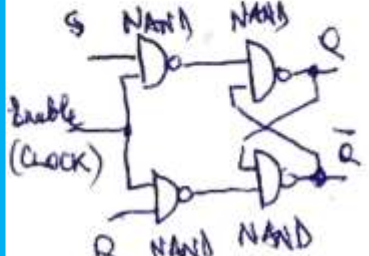
# S-R (Set-Reset) Memory element



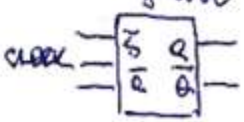

| Inputs $\bar{S}$ | $\bar{R}$ | Outputs Q | $\bar{Q}$ | |
|---|---|---|---|---|
| ~~0~~ | ~~0~~ | ~~1~~ | ~~1~~ | not allowed |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 1 | Q | $\bar{Q}$ | |


| Inputs S | R | Outputs Q | $\bar{Q}$ | |
|---|---|---|---|---|
| 0 | 0 | Q | $\bar{Q}$ | |
| 0 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | |
| ~~1~~ | ~~1~~ | ~~0~~ | ~~0~~ | not allowed |




symbol




| Inputs Q(t) | S | R | Output Q(t+1) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | not allowed |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | not allowed |

Karnaugh map

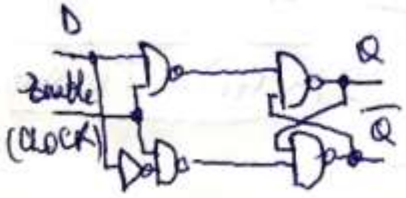
| Q(t) \ SR | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | d | 1 |
| 1 | 1 | | d | 1 |

don't care

Characteristic equation

$$\boxed{Q(t+1) = S + \bar{R} \cdot Q(t)}$$

# D (Delay) Memory element



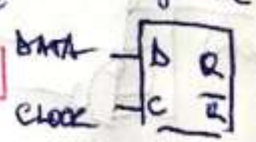

| Inputs Q(t) | D | Output Q(t+1) |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| Q(t) \ D | 0 | 1 |
|---|---|---|
| 0 | | 1 |
| 1 | | 1 |

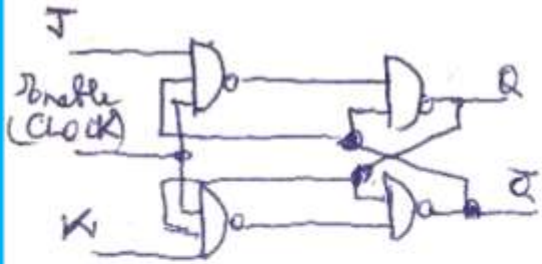
Characteristic equation

$$\boxed{Q(t+1) = D}$$

Symbol



# J-K (Jack Kilby) Memory element




| Inputs | | | Output |
|---|---|---|---|
| Q(t) | J | K | Q(t+1) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

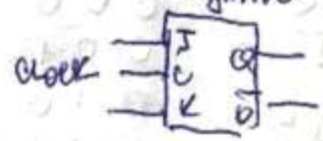
Karnaugh map

| Q(t) \ JK | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | 1 | 1 |
| 1 | 1 | | | 1 |

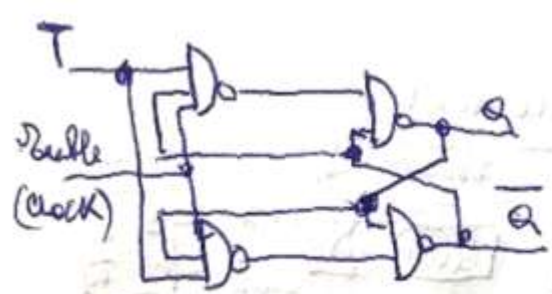
Characteristic equation

$$\boxed{Q(t+1) = J\cdot\overline{Q(t)} + \overline{K}\cdot Q(t)}$$

Symbol




# T (Toggle) Memory element




| Inputs | | Output |
|---|---|---|
| Q(t) | T | Q(t+1) |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

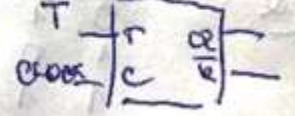
Karnaugh map

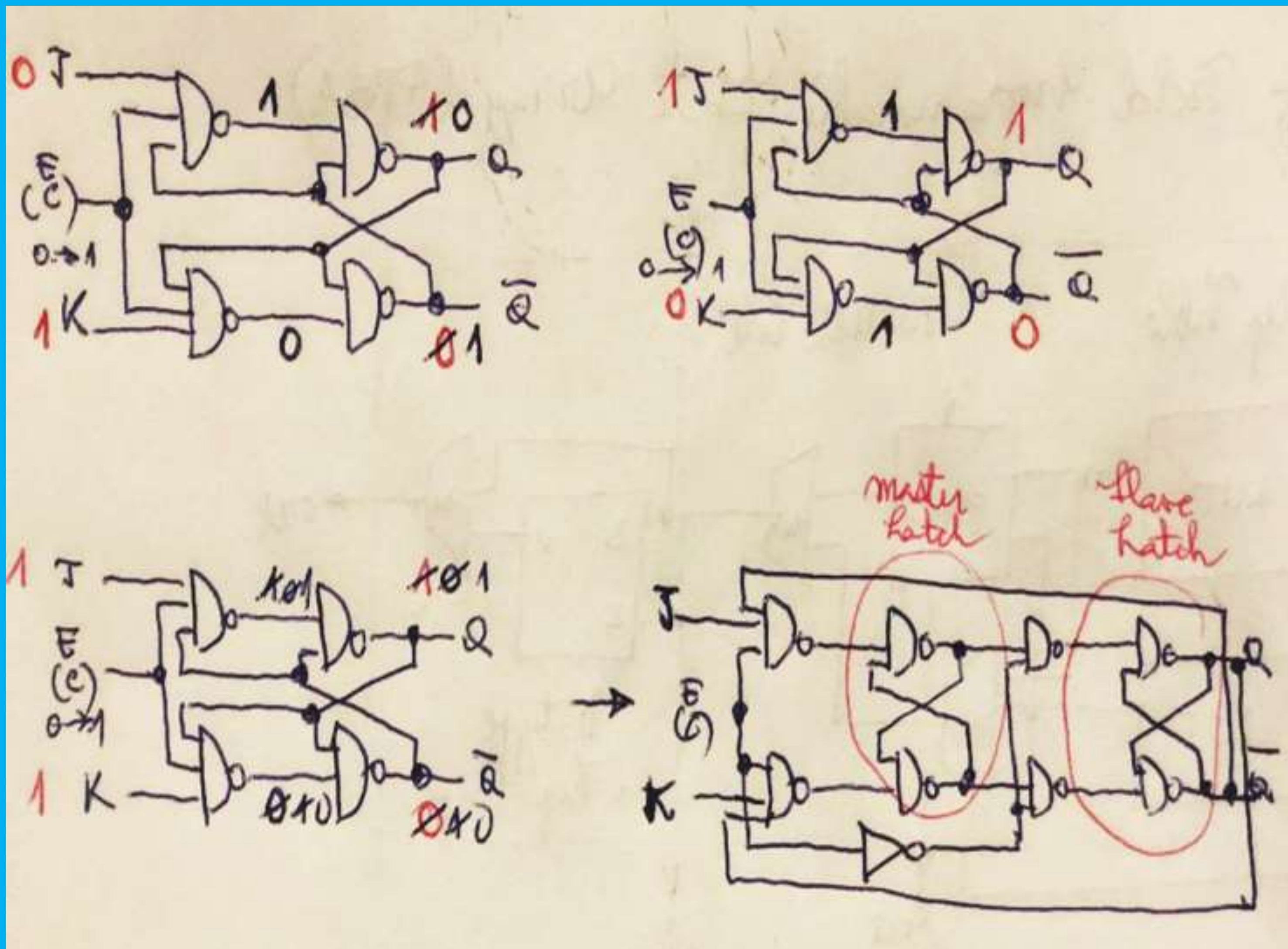
| Q(t) \ T | 0 | 1 |
|---|---|---|
| 0 | | 1 |
| 1 | 1 | |

Characteristic equation

$$Q(t+1) = T\cdot\overline{Q(t)} + Q(t)\cdot\overline{T}$$

Symbol $\Rightarrow$ $\boxed{Q(t+1) = T \oplus Q(t)}$

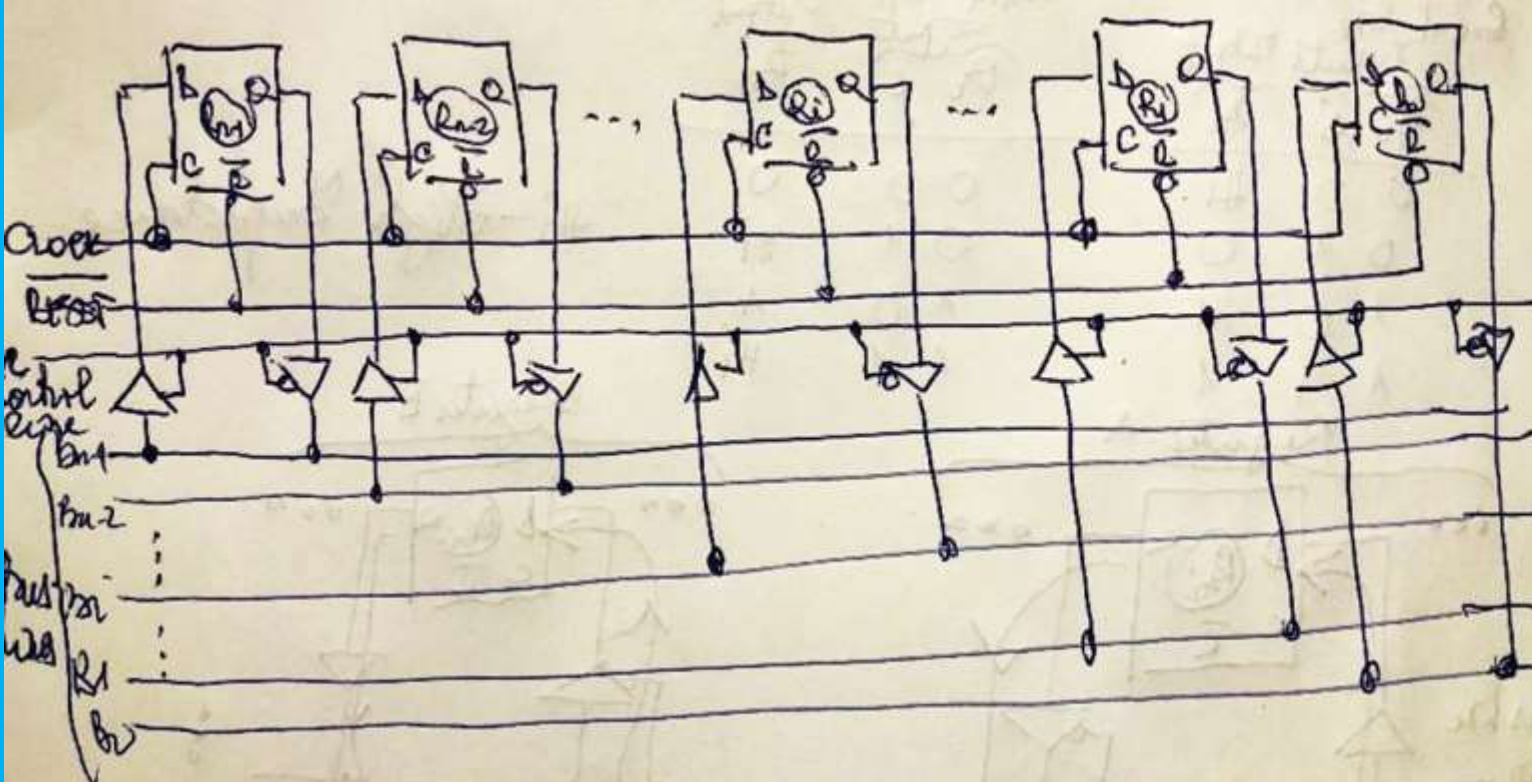




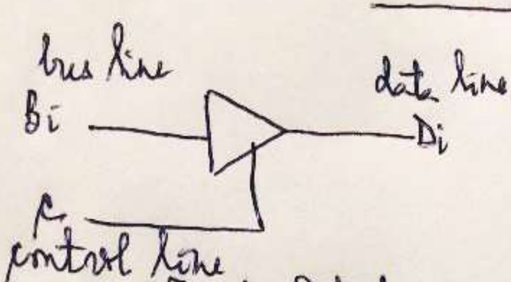
J
K
E
C
Q
$\overline{Q}$
0→1
master latch
slave latch

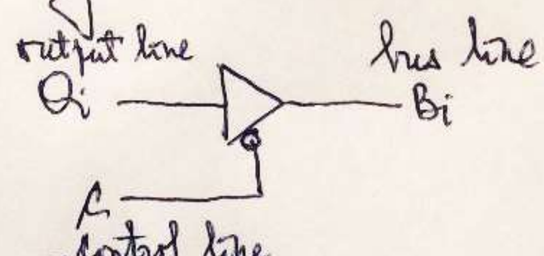
# MSI sequential logic circuits

## Registers



# Tristate logic gates

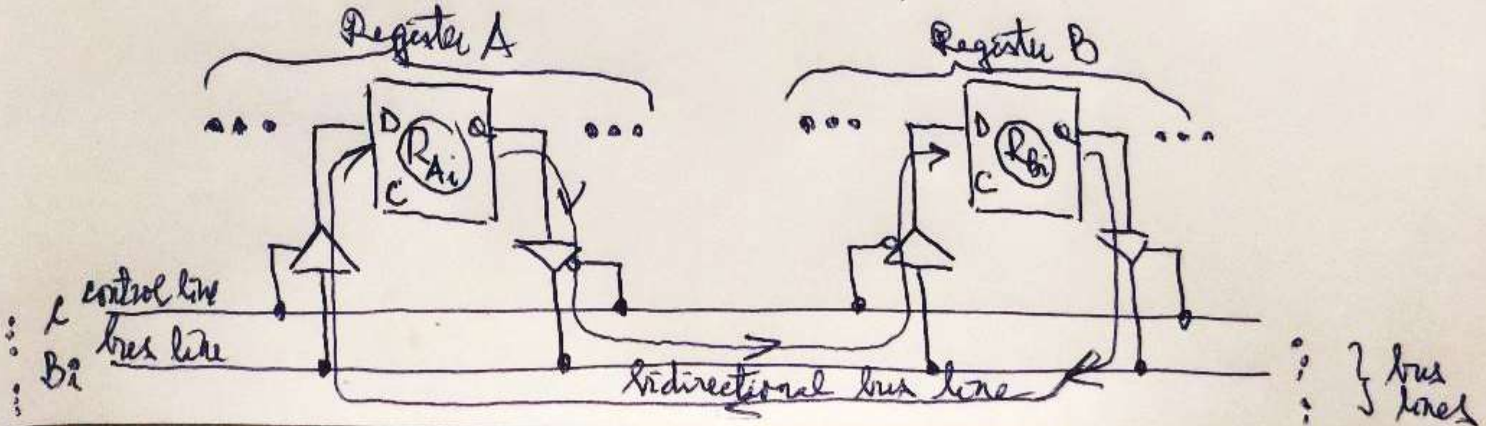


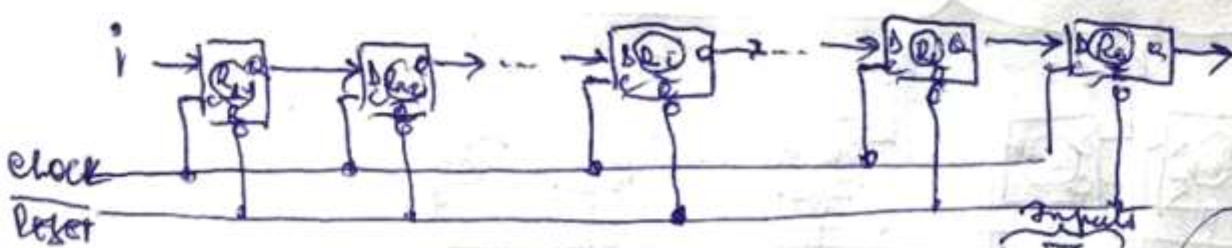

| Inputs | | Output |
|---|---|---|
| Bi | C | Di |
| 0 | 0 | Hi |
| 0 | 1 | 0 |
| 1 | 0 | Hi |
| 1 | 1 | 1 |

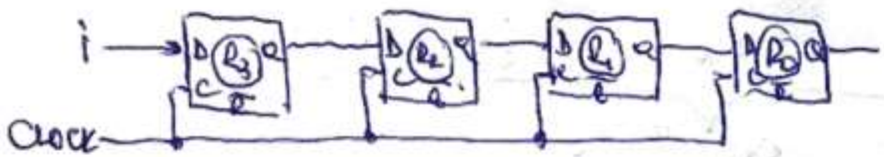
| Inputs | | Output |
|---|---|---|
| Qi | C | Bi |
| 0 | 0 | 0 |
| 0 | 1 | Hi |
| 1 | 0 | 1 |
| 1 | 1 | Hi |

Hi - High Impedance



# Shiftregisters



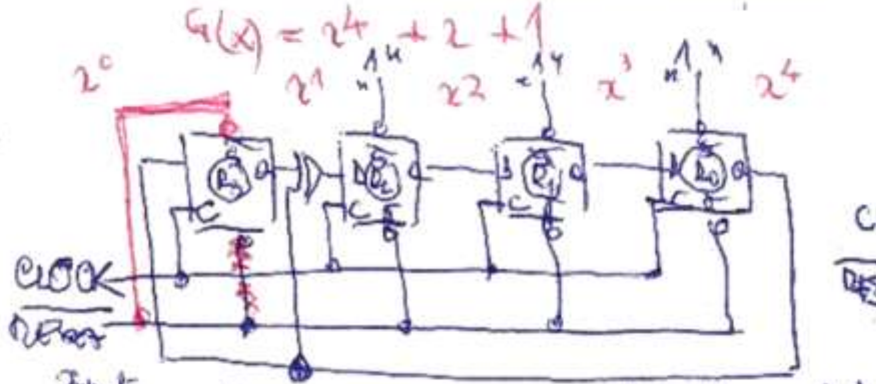

inputs

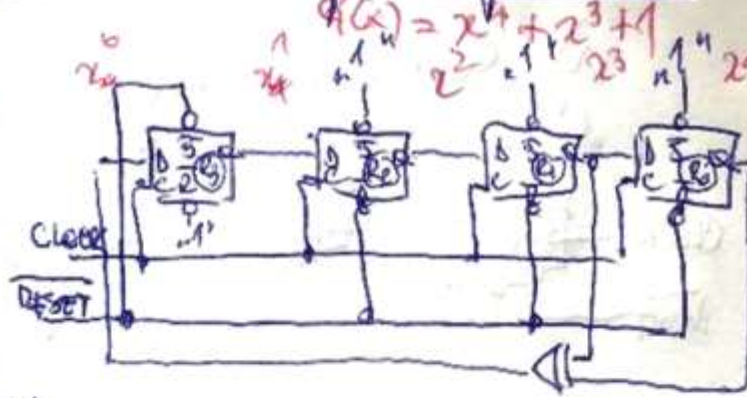
don't care

| I | R | C | $Q_{R3}$ | $Q_{R2}$ | $Q_{R1}$ | $Q_{R0}$ |
|---|---|---|---|---|---|---|
| x | 0 | x | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 |

* Right-shift
* left-shift
* Right/left-shift

# LFSR - Linear Feedback Shift Register




Inputs ← Decimal equivalent

| $\bar{R}$ | C | $Q_{R3}$ ($2^3$) | $Q_{R2}$ ($2^2$) | $Q_{R1}$ ($2^1$) | $Q_{R0}$ ($2^0$) | DE |
|---|---|---|---|---|---|---|
| 0 | X | 1 | 0 | 0 | 0 | 8 |
| 1 | 1 | 0 | 1 | 0 | 0 | 4 |
| 1 | 1 | 0 | 0 | 1 | 0 | 2 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 12 |
| 1 | 1 | 0 | 1 | 1 | 0 | 6 |
| 1 | 1 | 0 | 0 | 1 | 1 | 3 |
| 1 | 1 | 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 1 | 0 | 1 | 0 | 10 |
| 1 | 1 | 0 | 1 | 0 | 1 | 5 |
| 1 | 1 | 1 | 1 | 1 | 0 | 14 |
| 1 | 1 | 0 | 1 | 1 | 1 | 7 |
| 1 | 1 | 1 | 1 | 1 | 1 | 15 |
| 1 | 1 | 1 | 0 | 1 | 1 | 11 |
| 1 | 1 | 1 | 0 | 0 | 1 | 9 |
| 1 | 1 | 1 | 0 | 0 | 0 | 8 |

PRPG - Pseudo-Random Pattern Generator

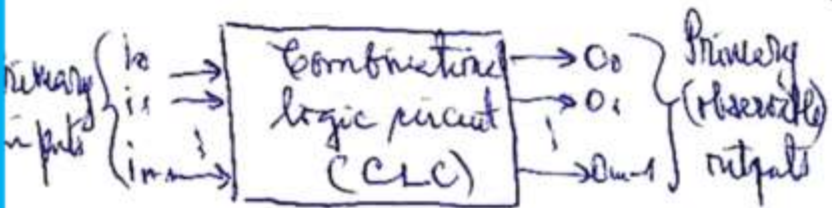
| $\bar{R}$ | C | $Q_{R3}$ | $Q_{R2}$ | $Q_{R1}$ | $Q_{R0}$ | DE |
|---|---|---|---|---|---|---|
| 0 | X | 1 | 0 | 0 | 0 | 8 |
| 1 | 1 | 0 | 1 | 0 | 0 | 4 |
| 1 | 1 | 0 | 0 | 1 | 0 | 2 |
| 1 | 1 | 1 | 0 | 0 | 1 | 9 |
| 1 | 1 | 1 | 1 | 0 | 0 | 12 |
| 1 | 1 | 0 | 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 0 | 1 | 1 | 11 |
| 1 | 1 | 0 | 1 | 0 | 1 | 5 |
| 1 | 1 | 1 | 0 | 1 | 0 | 10 |
| 1 | 1 | 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 1 | 1 | 1 | 0 | 14 |
| 1 | 1 | 1 | 1 | 1 | 1 | 15 |
| 1 | 1 | 0 | 1 | 1 | 1 | 7 |
| 1 | 1 | 0 | 0 | 1 | 1 | 3 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 8 |

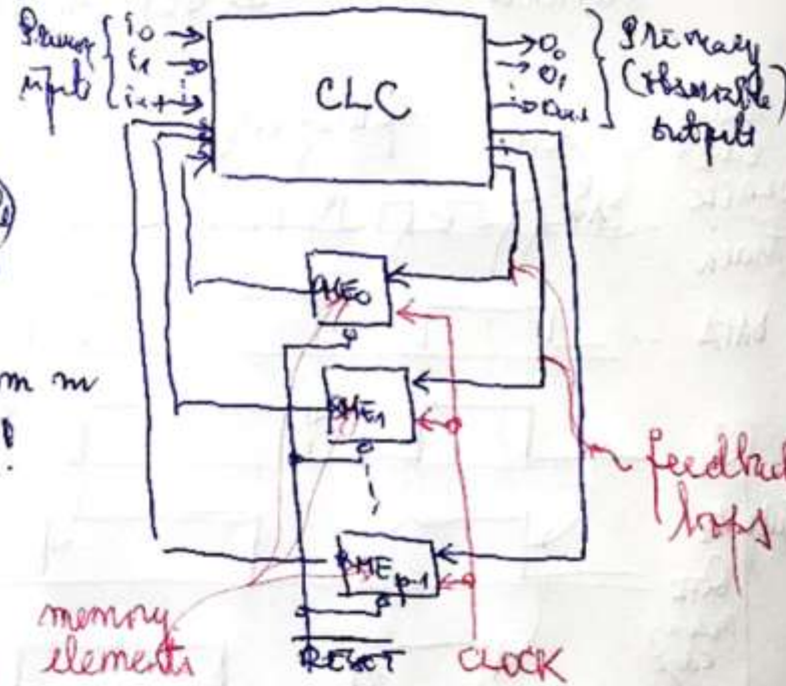
La mulți ani,

sănătate, gând pozitiv și speranță în mai bine!

Mult succes în prima voastră sesiune de examene în calitate de studenți!

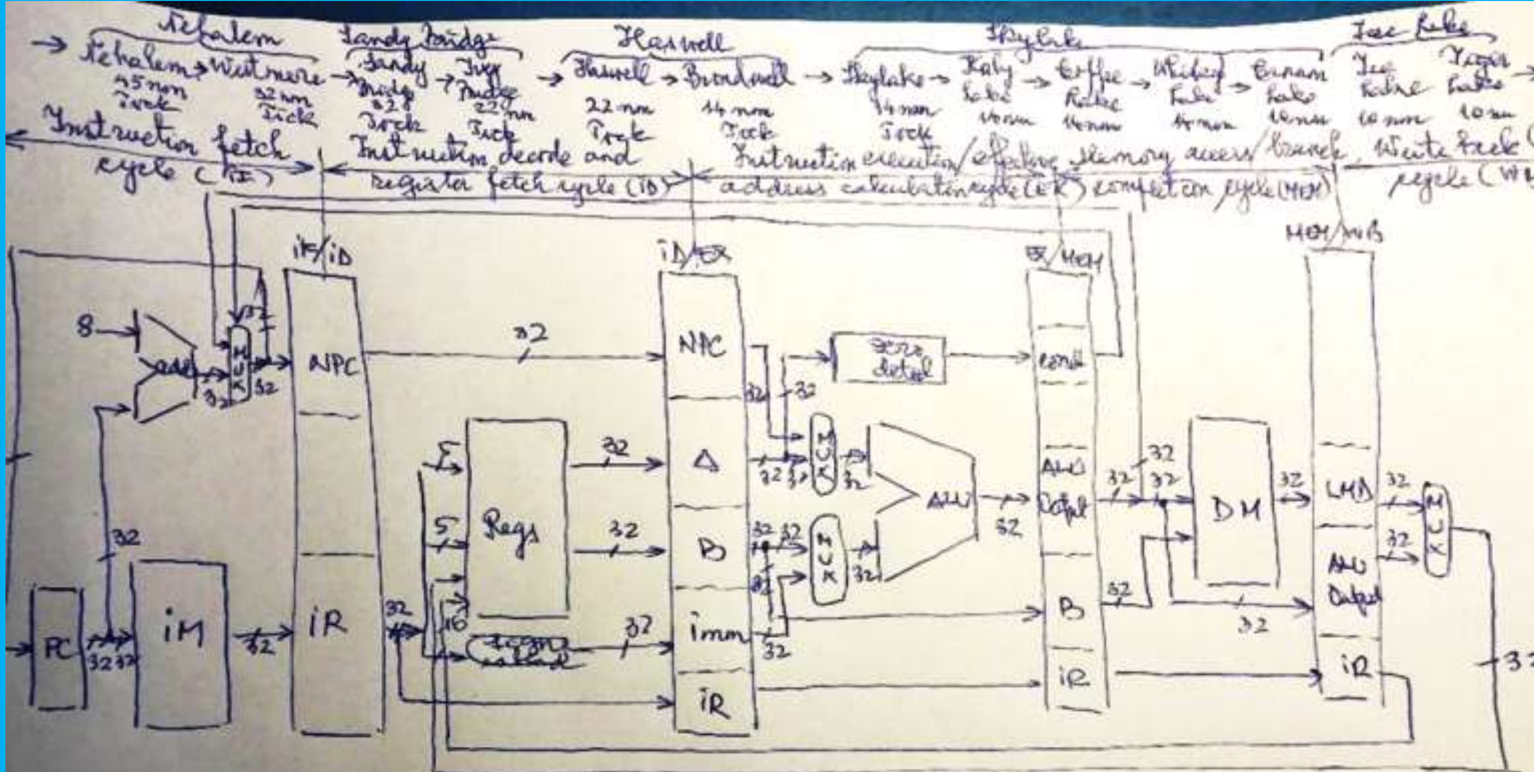
# Combinational vs sequential logic circuits




* $n$ in general different from $m$
* no memory elements!
* no feedback loops

* Sequential logic circuit
  - Synchronous (with CLOCK)
  - Asynchronous (without CLOCK)




PC - Program counter
iM - Instruction memory
iR - Instruction register
NPC - Next program counter

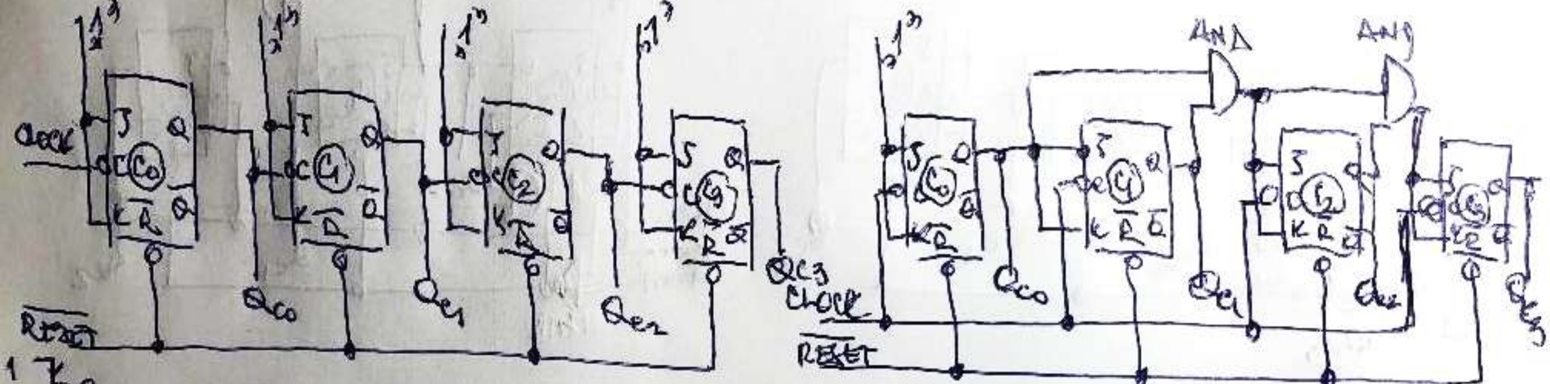
MUX - Multiplexer
Regs - Register file
A, B, imm - Buffer register
↑ Immediate

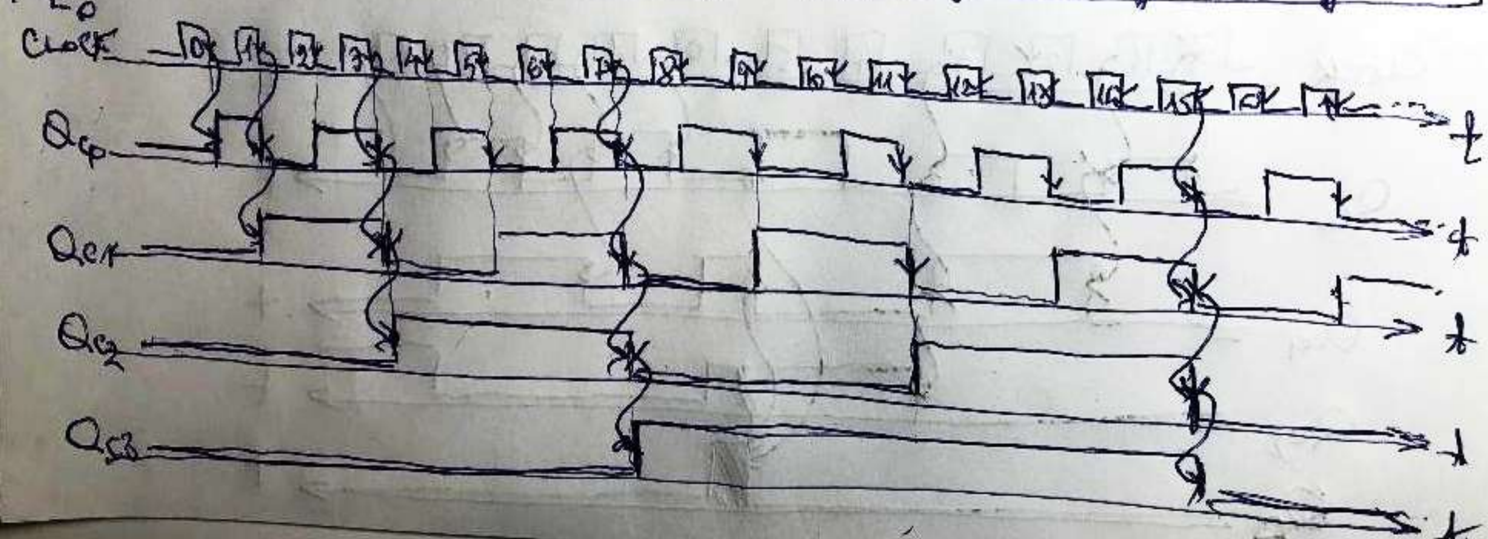
ALU - Arithmetic/logic unit
DM - Data memory
LMD - Load memory data register

# Counters

a) Asynchronous

b) Synchronous

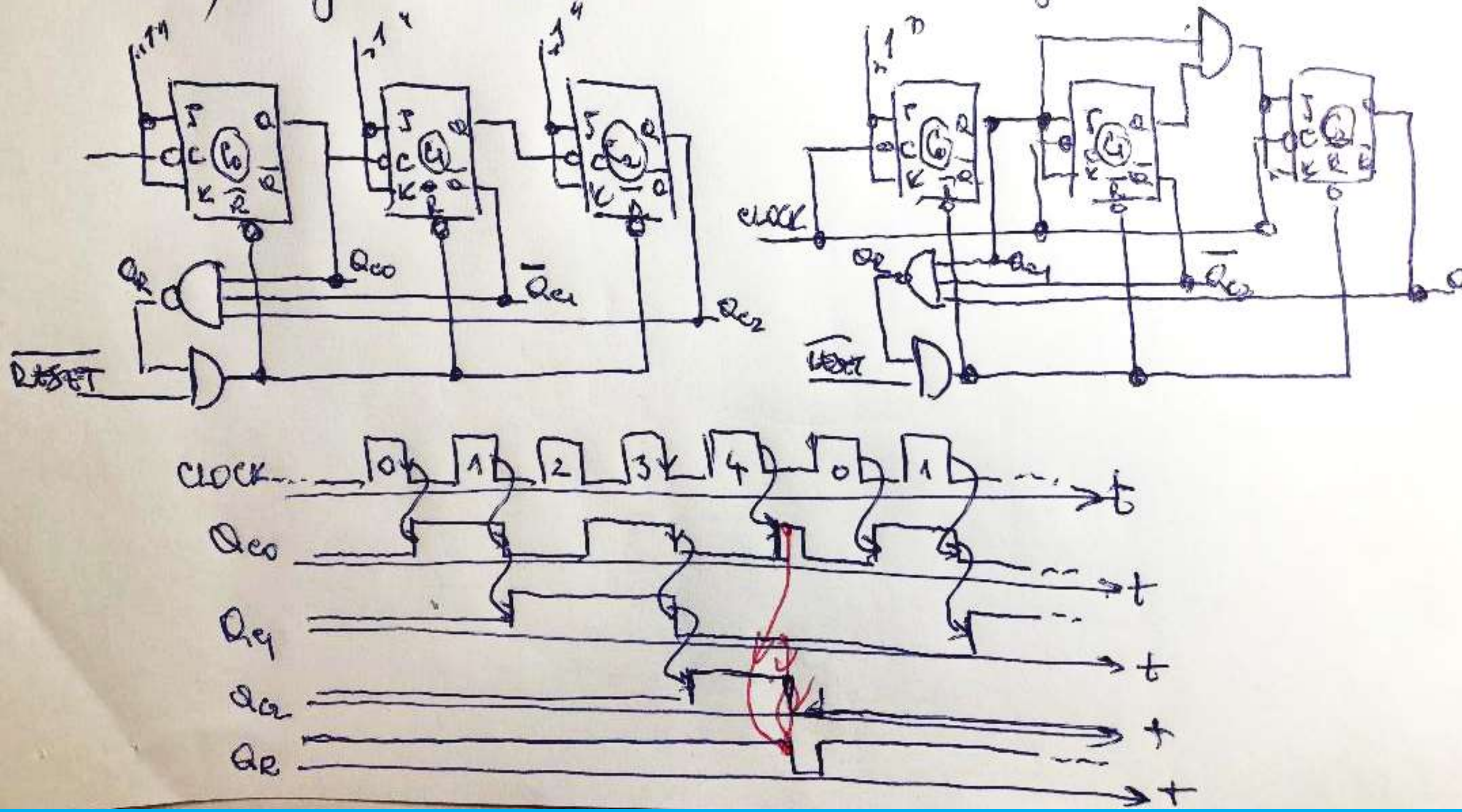


# Modulo 5 binary counter

## a) Asynchronous

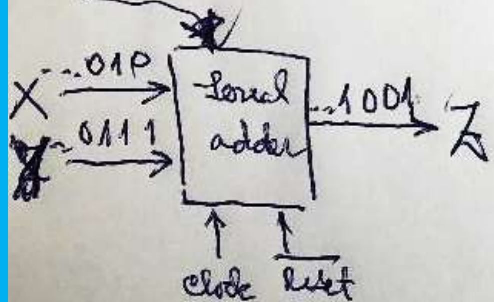
## b) Synchronous



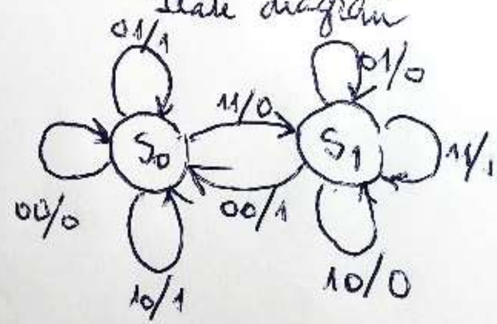
# Design of sequential logic circuits

Problem statement → State table construct → Choice of types of memory elements → Transition table construct → Logic equation minimization → Logic circuits drawn

Ex: Serial adder - least significant digit first




State diagram




$S_0$ - no carry

$S_1$ - carry

| State \ Input xy | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| $S_0$ | $S_0/0$ | $S_0/1$ | $S_1/0$ | $S_0/1$ |
| $S_1$ | $S_0/1$ | $S_1/0$ | $S_1/1$ | $S_1/0$ |

| State variable W \ Input xy | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0/0 | 0/1 | 1/0 | 0/1 |
| 1 | 0/1 | 1/0 | 1/1 | 1/0 |

# D type flip-flop

inputs $W(t+1) = D$ outputs

| w | x | y | D | Z |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

D

| w \ xy | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | 1 | |
| 1 | | 1 | 1 | 1 |

$D = xy + xw + yw$

Z

| w \ xy | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | 1 | | 1 |
| 1 | 1 | | 1 | |

$Z = x \oplus y \oplus w$

# J-K type flip-flop

$w(t+1) = J\,\overline{w(t)}$ or $\overline{K}\,w(t)$

inputs outputs

| w | x | y | J | K | Z |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | d | 0 |
| 0 | 0 | 1 | 0 | d | 1 |
| 0 | 1 | 0 | 0 | d | 1 |
| 0 | 1 | 1 | 1 | d | 0 |
| 1 | 0 | 0 | d | 1 | 1 |
| 1 | 0 | 1 | d | 0 | 0 |
| 1 | 1 | 0 | d | 0 | 0 |
| 1 | 1 | 1 | d | 0 | 1 |

don't care

J

| w \ xy | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | 1 | |
| 1 | d | d | d | d |

$J = xy$

K

| w \ xy | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | d | d | d | d |
| 1 | 1 | | | |

$K = \overline{x} \cdot \overline{y}$

$$w(t+1) = J \cdot \overline{w(t)} + \overline{K}\, w(t)$$

$$0 = J \cdot \overline{0} + \overline{K} \cdot 0$$

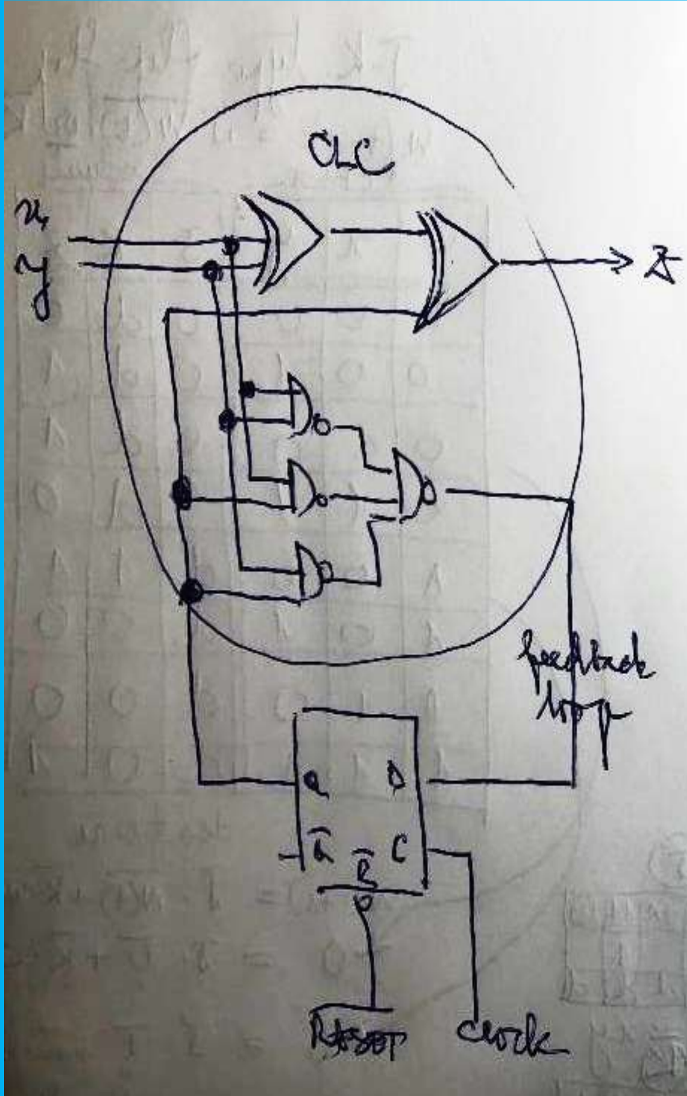
$$0 = J \cdot \underset{0}{\overset{1}{\overline{1}}} + \overset{0}{\overline{K}} \cdot 1$$

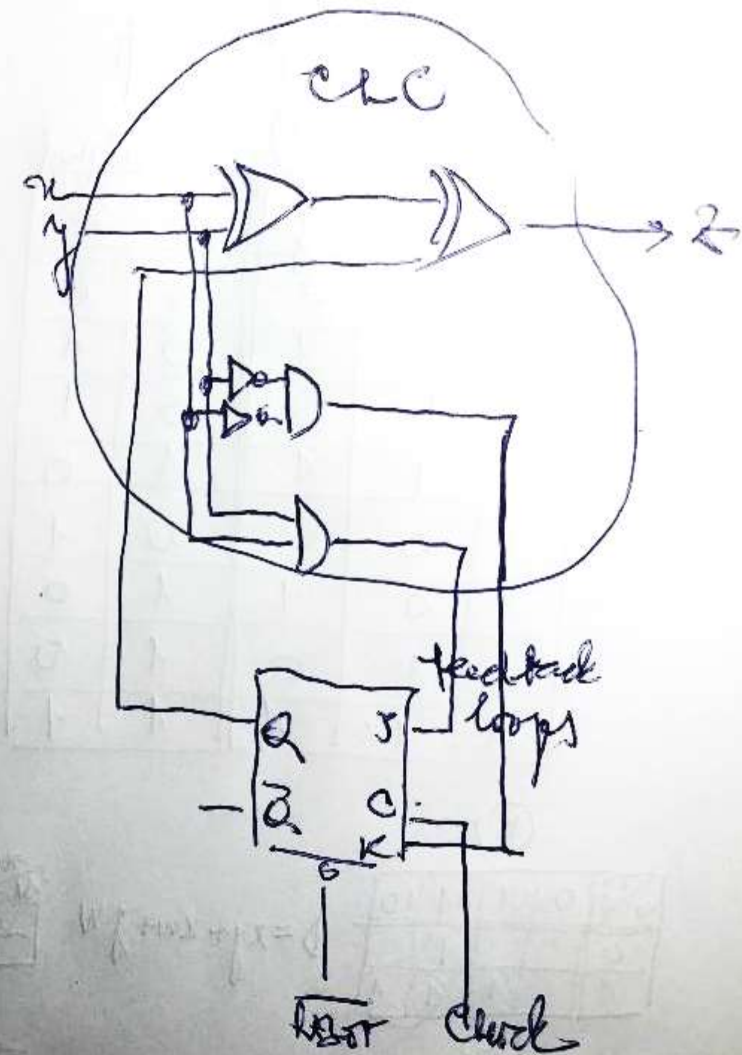
$\Downarrow$

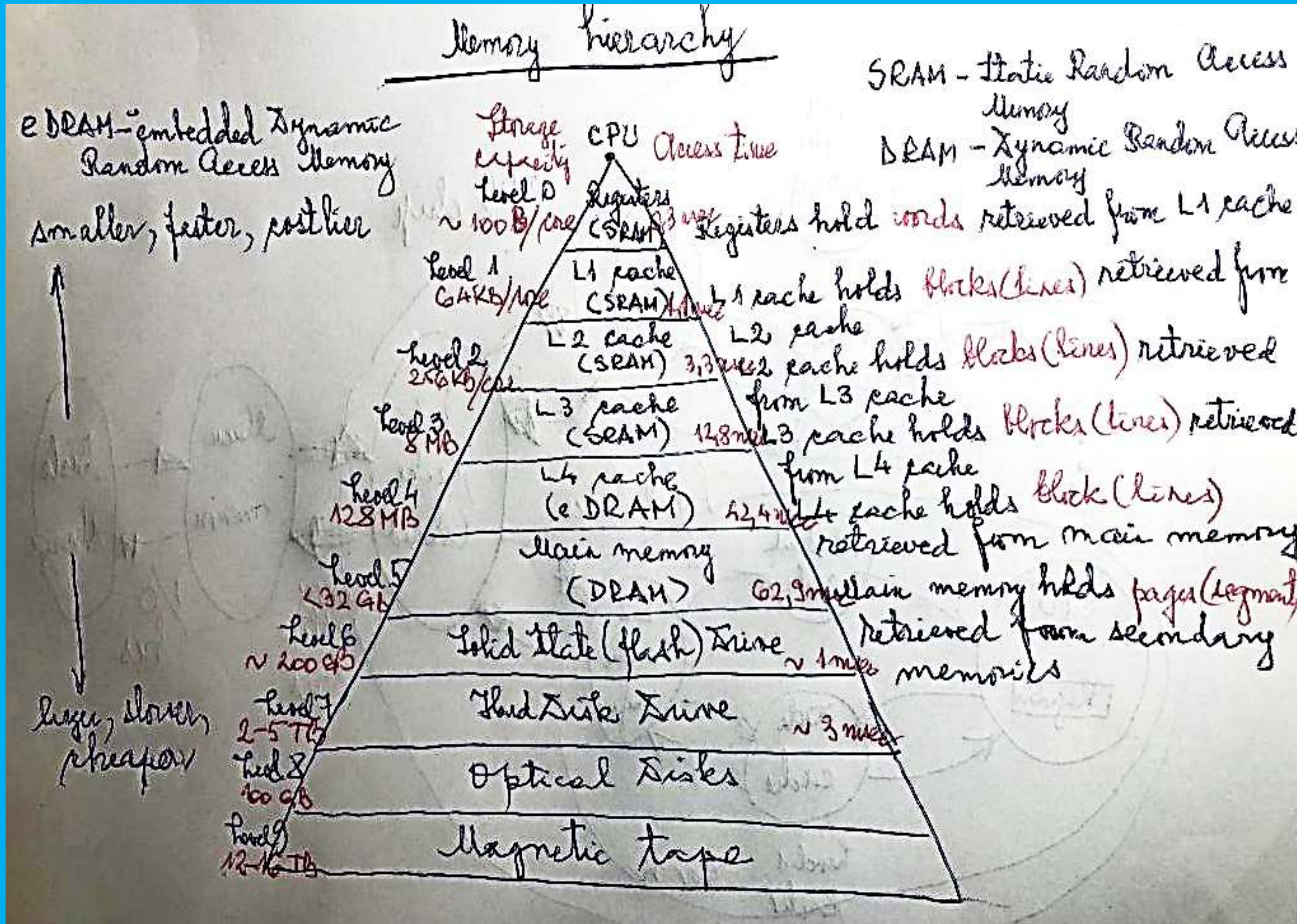
$\overline{K} = 0$

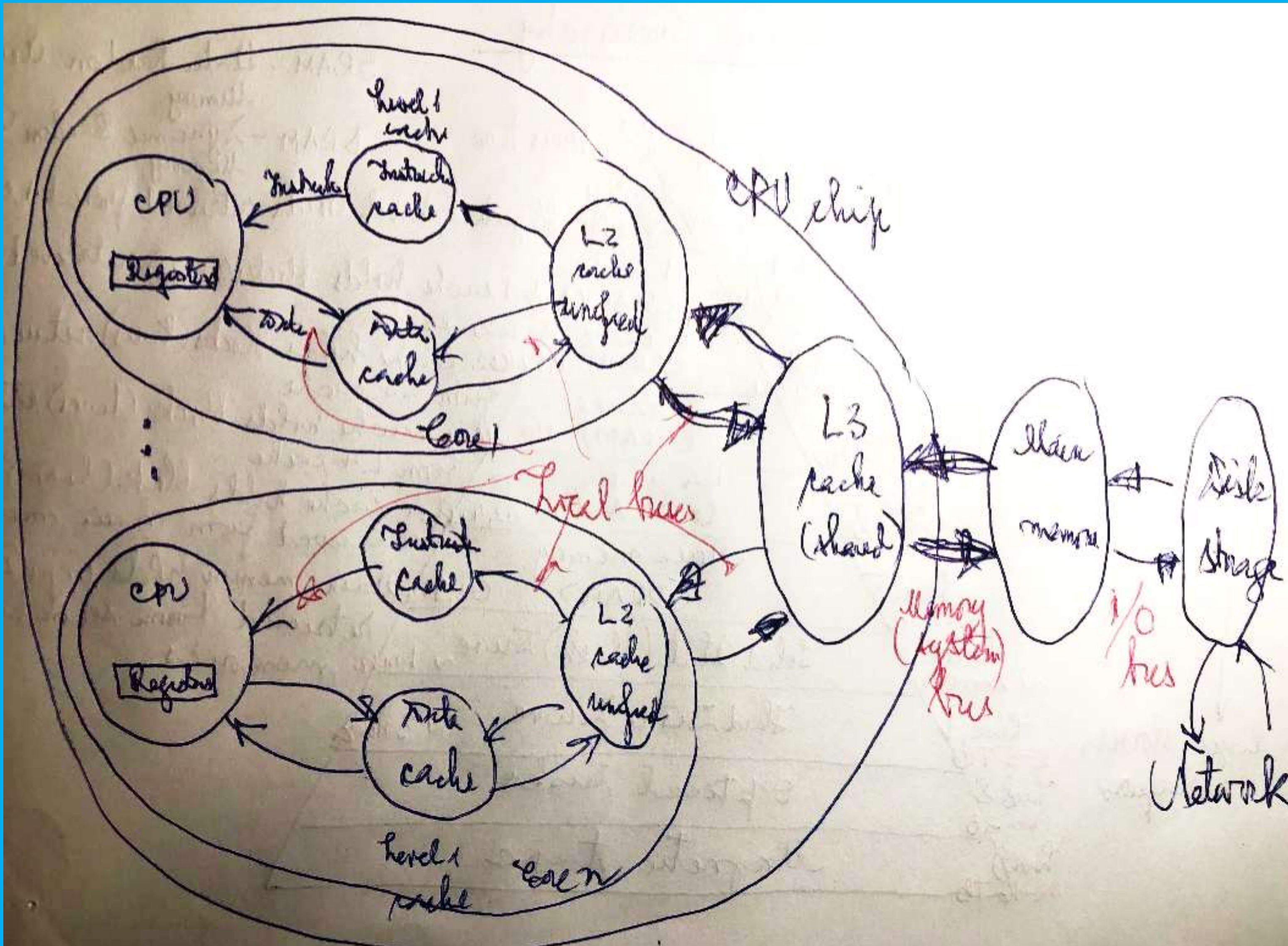
$\downarrow$

$K = 1$


CLC
x
y
z
feedback loop
Q
D
C
Reset
Clock


CLC
x
y
z
feedback loops
Q
J
C
K
Reset
Clock


Memory hierarchy
SRAM - Static Random Access Memory
DRAM - Dynamic Random Access Memory
eDRAM - embedded Dynamic Random Access Memory
smaller, faster, costlier
larger, slower, cheaper
Storage capacity
CPU
Access time
Level 0
~100B/core
Registers (SRAM)
0,3 nsec
Registers hold words retrieved from L1 cache
Level 1
64KB/core
L1 cache (SRAM)
1,1 nsec
L1 cache holds blocks (lines) retrieved from L2 cache
Level 2
256KB/core
L2 cache (SRAM)
3,3 nsec
L2 cache holds blocks (lines) retrieved from L3 cache
Level 3
8MB
L3 cache (SRAM)
12,8 nsec
L3 cache holds blocks (lines) retrieved from L4 cache
Level 4
128MB
L4 cache (eDRAM)
42,4 nsec
L4 cache holds block (lines) retrieved from main memory
Level 5
<32GB
Main memory (DRAM)
62,9 nsec
Main memory holds pages (segments) retrieved from secondary memories
Level 6
~200GB
Solid State (flash) Drive
~1 msec
Level 7
2-5TB
Hard Disk Drive
~3 msec
Level 8
100GB
Optical Disks
Level 9
12-16TB
Magnetic tape



CPU chip
Level 1 cache
CPU
Registers
Instruction cache
Data cache
L2 cache unified
Core 1
L3 cache (shared)
Main memory
Disk storage
Network
Memory (system) bus
I/O bus
Level 1 cache
Core n

-1-

Subiectele și șablonul de corectură pentru examenul la disciplina „Fundamentele calculatoarelor" – an I INF – din 12.01.2020, orele 16-20, sala A101

Start – 1p

Parte teoretică

• Subiectul ① Să se trateze matematic corecțiile care intervin la operația de (5p) adunare a numerelor reprezentate în virgulă fixă.

Soluție: • La reprezentarea numerelor binare în virgulă fixă intervine corecția de „end around carry" în 2 cazuri din 4 la adunarea numerelor reprezentate în complement de 1, anume când:

0,25 • $X>0, Y<0$ și $|X|>|Y|$

0,125 $\begin{cases} X = 0\, x_{n-2} x_{n-3} \dots x_i \dots x_1 x_0 = X_M \\ \overline{Y} = 1\, \bar{y}_{n-2}\bar{y}_{n-3}\dots\bar{y}_i\dots\bar{y}_1\bar{y}_0 = 111\dots1\dots11 - 0\, y_{n-2}y_{n-3}\dots y_1 y_0 = \end{cases}$ ← partea de mărime a lui X

$= 2^n - 1 - Y_M$ 0,125

$X+\overline{Y} = X_M + 2^n - 1 - Y_M = \underbrace{2^n}_{} + \underbrace{(X_M - Y_M)}_{\text{valoarea rezultată}} - 1$ 0,25

trebuie corectată

depășește capacitatea registrului de lungime n

⇒ corecția constă în adunarea unității binare care depășește capacitatea registrului la cel mai puțin semnificativ rang al sumei printr-o legătură de corecție prezentă la sumatorul binar paralel în această variantă de unde și denumirea de „end around carry".

0,25 • $X<0, Y<0$ ⇒ avem în maniera descrisă mai sus:

0,25 $\begin{cases} \overline{X} = 1\,\bar{x}_{n-2}\dots\bar{x}_i\dots\bar{x}_1\bar{x}_0 = 2^n - 1 - X_M \\ \overline{Y} = 1\,\bar{y}_{n-2}\dots\bar{y}_i\dots\bar{y}_1\bar{y}_0 = 2^n - 1 - Y_M \end{cases}$

$\overline{X}+\overline{Y} = 2^n - 1 - X_M + 2^n - 1 - Y_M = 2^n + \underbrace{2^n - 1 - (X_M+Y_M)}_{\text{rezultatul sumei în C1}} - 1$ 0,25

end around carry

0,25

• La reprezentarea numerelor zecimale în virgulă fixă intervine corecția de „6" la adunarea numerelor reprezentate în codul BCD (binary coded decimal), anume în 2 situații:

• când avem carry la adunarea echivalentelor binare a unor cifre zecimale, adică, spre ex.: 0,25

$$\begin{array}{r} 9+ \\ 8 \\ \hline 17 \end{array} \xrightarrow{\text{în BCD}} \begin{array}{r} 1001+ \\ 1000 \\ \hline 1\ 0001 \end{array}$$ 0,25

are valoarea $10^1=10$ are valoarea de $2^4=16=10+6$

„6" trebuie „scăzut în urmă" pentru ca transportul să aibă doar valoarea 10, deci

$$\begin{array}{r} 1001+ \\ 1000 \\ \hline 1\ 0001+ \\ 1\ 0110 \\ \hline 10+ⓑ\ 0111 \end{array}$$ 0,5

0,25 • când la adunarea reprezentărilor binare a două cifre zecimale nu apare carry și valoarea echivalentului binar al cifrei zecimale sumă este mai mare de 9, adică, spre ex.:

$$\begin{array}{r} 6+ \\ 7 \\ \hline 13 \end{array} \longrightarrow \begin{array}{r} 0110+ \\ 0111 \\ \hline 0\ 1101 \end{array} \Rightarrow \begin{array}{r} 0110+ \\ 0111 \\ \hline 1101 = \\ 1010 \end{array} \Rightarrow \begin{array}{r} 1101+ \\ 0110 \\ \hline 0011 \end{array}$$

$13 \downarrow 10^1$ $\qquad$ $0 = 10 - 10$

0,25 $\qquad$ 0,25

se impune scăderea lui 10, dar aceasta este echivalentă cu adunarea complementului de 2 a lui 10, care este 6

• La reprezentarea numerelor zecimale de virgulă fixă avem o proiectare de 3^n l., adunarea numerelor în
0,25 codul E3 (Three excess), când operația binară se realizează în manieră binară convențională și nu în maniera „decalată" ca la adunarea
0,25 BCD, și anume în 2 situații:

• când avem carry la adunarea echivalentelor binare a două cifre zecimale, spre exemplu

$$\begin{array}{r} 6+ \\ 7 \\ \hline 13 \end{array} \longrightarrow \begin{array}{r} 1001_{E3}+ \\ 1010_{E3} \\ \hline (1)0011 \end{array} \Rightarrow \begin{array}{r} 1001+ \\ 1010 \\ \hline 1\ 0011+ \\ 0011 \\ \hline 0110 \end{array} = 3_{E3}$$

0,25 → 3+3 $\qquad$ 0,25

echivalent cu adunarea lui 2 și a excesului de 3

• când nu avem carry la adunarea echivalentelor binare a două cifre zecimale, spre exemplu

$$\begin{array}{r} 1+ \\ 3 \\ \hline 4 \end{array} \longrightarrow \begin{array}{r} 0100_{E3}+ \\ 0110_{E3} \\ \hline 0\ 1010 \end{array} \Rightarrow \begin{array}{r} 0100+ \\ 0110 \\ \hline 1010- \\ 0011 \\ \hline 0111 \end{array} = 4_{E3}$$

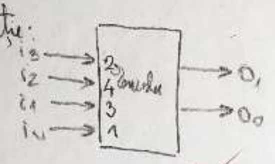
0,25 $3 - 3$ $\qquad$ 0,25

pentru exces de 3

mai menționăm că scăderea lui 3 este echivalentă cu adunarea complementului de 2 a lui 3, care este $0011_{2-SM} = 1101_{2-C2}$

• Subiectul (2) Să se proiecteze un codificator binar de prioritate
(2p) cu 4 intrări $i_3 i_2 i_1 i_0$ (fără intrare de enable) cu prioritățile $i_3 i_2 i_1 i_0 = 2\ 4\ 3\ 1$, unde 4 este cel mai prioritar și 1 cel mai puțin prioritar. La implementare se vor folosi doar porți logice NOR cu 2 intrări.

Soluție:




0,25

| Inputs | | | | Outputs | |
|---|---|---|---|---|---|
| $i_3$ | $i_2$ | $i_1$ | $i_0$ | $O_1$ | $O_0$ |
| x | 1 | x | x | 1 | 0 |
| x | 0 | 1 | x | 0 | 1 |
| 1 | 0 | 0 | x | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 |

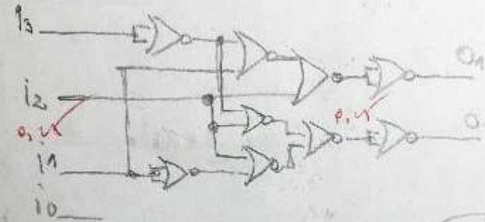
0,25

0,25 $O_1 = i_2 + i_3\bar{i}_2\bar{i}_1 = i_2 + i_3\bar{i}_1 = \overline{\overline{i_2 + i_3 \cdot \bar{i}_1}} = \overline{\overline{i_2} + \overline{\bar{i}_3 + i_1}}$

0,25 $O_0 = \bar{i}_2 i_1 + i_3 \cdot \bar{i}_2 \cdot \bar{i}_1 = \bar{i}_2(i_1 + i_3 \cdot \bar{i}_1) = \bar{i}_2 i_1 + i_3 \cdot \bar{i}_2 = \overline{\overline{\bar{i}_2 i_1 + i_3 \bar{i}_2}}$

0,5

$= \overline{\overline{i_2 + \bar{i}_1} + \overline{\bar{i}_3 + i_2}}$



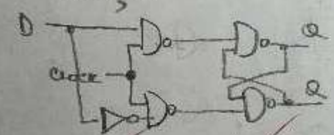

0,25 0,25 0,25

(cu deducția ecuației caracteristice)

**Subiectul 3** Să se trateze elementul de memorare de tip D și

(2p) Să construiască cu acesta un registru cu 4 ranguri și un registru de deplasare (la dreapta) cu 3 ranguri cu tabelul de adevăr asociat atunci când la intrarea în primul rang se aplică succesiv secvența binară 1101 (starea inițială toate 000)

Soluție:



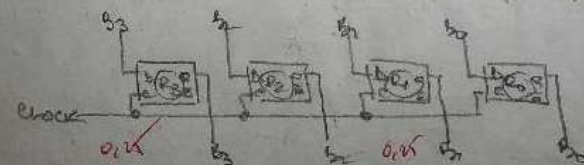

0,25 0,25

| $Q(t)$ | $D$ | $Q(t+1)$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

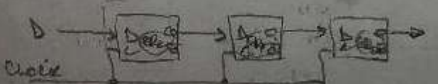
0,25

$Q(t+1) = \overline{Q(t)} \cdot D + Q(t) \cdot D =$
$= D(\overline{Q(t)} + Q(t)) = D$

0,25




0,25 0,25




0,25

| D | C (clock) | $SR_2$ | $SR_1$ | $SR_0$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |

0,25

Barem – 1p

Subiectul (1) Utilizând, pe de o parte, de porți NAND cu 2 și 3 intrări și, pe de altă parte, de porți NAND doar cu 2 intrări, să se proiecteze, folosind metoda Quine-McCluskey, sinteza celui mai puțin semnificativ bit a unui convertor din cod exces de 3 în cod 2 din 5.

(5p)

| | A | B | C | D | $X_4$ | $X_3$ | $X_2$ | $X_1$ | $X_0$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 6 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 7 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 8 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 9 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

0,25

| | A | B | C | D | |
|---|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 | ✓ |
| $m_1$ | 0 | 0 | 0 | 1 | ✓ |
| $m_2$ | 0 | 0 | 1 | 0 | ✓ |
| $m_4$ | 0 | 1 | 0 | 0 | ✓ |
| $m_5$ | 0 | 1 | 0 | 1 | ✓ |
| $m_{10}$ | 1 | 0 | 1 | 0 | ✓ |
| $m_7$ | 0 | 1 | 1 | 1 | ✓ |
| $m_{13}$ | 1 | 1 | 0 | 1 | ✓ |
| $m_{14}$ | 1 | 1 | 1 | 0 | ✓ |
| $m_{15}$ | 1 | 1 | 1 | 1 | ✓ |

0,25

| | A | B | C | D | |
|---|---|---|---|---|---|
| $(m_0, m_1)$ | 0 | 0 | 0 | – | ✓ |
| $(m_0, m_2)$ | 0 | 0 | – | 0 | * |
| $(m_0, m_4)$ | 0 | – | 0 | 0 | ✓ |
| $(m_1, m_5)$ | 0 | – | 0 | 1 | ✓ |
| $(m_2, m_{10})$ | – | 0 | 1 | 0 | * |
| $(m_4, m_5)$ | 0 | 1 | 0 | – | ✓ |
| $(m_5, m_7)$ | 0 | 1 | – | 1 | ✓ |
| $(m_5, m_{13})$ | – | 1 | 0 | 1 | ✓ |
| $(m_{10}, m_{14})$ | 1 | – | 1 | 0 | * |
| $(m_7, m_{15})$ | – | 1 | 1 | 1 | ✓ |
| $(m_{13}, m_{15})$ | 1 | 1 | – | 1 | ✓ |
| $(m_{14}, m_{15})$ | 1 | 1 | 1 | – | * |

0,25 0,25 0,25

| | A | B | C | D | |
|---|---|---|---|---|---|
| $(m_0, m_1, m_4, m_5)$ | 0 | – | 0 | – | * |
| $(m_5, m_7, m_{13}, m_{15})$ | – | 1 | – | 1 | * |

0,25

0,25 0,25

$X_0=\sum(m_4, m_5, m_7, m_{10})+\sum_d(m_0, m_1, m_2, m_{13}, m_{14}, m_{15})$

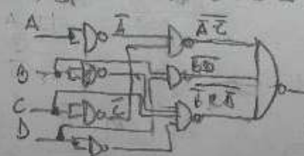
0,25

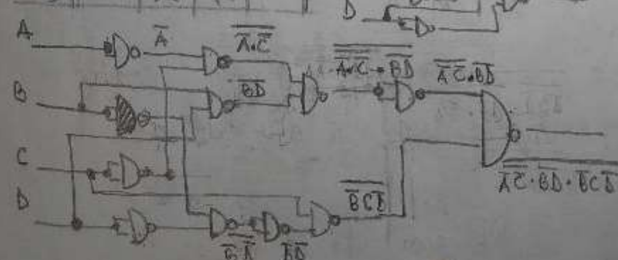
| Implicanți primi | $m_4$ | $m_5$ | $m_7$ | $m_{10}$ |
|---|---|---|---|---|
| ~~$AB\bar C$~~ | | | | |
| $\bar B C\bar D$ | | | | * |
| $AC\bar D$ | | | | * |
| ~~$ABC$~~ | | | | |
| $\bar A\bar C$ | * | * | | |
| $BD$ | | * | * | |

0,25 $X_{01}=\bar A\bar C+BD+\bar B C\bar D$

$X_{02}=\bar A\bar C+BD+AC\bar D$ 0,25

$X_{01}=\overline{\overline{\bar A\bar C}\cdot\overline{BD}\cdot\overline{\bar B C\bar D}}=$




0,25 + 0,25 Varianta 1: 5 x NAND 2 intrări + 2 NAND 3 intrări

0,25 + 0,25 Varianta 2: 4 NAND 2 intrări + 2 NAND 3 intrări

0,25 Varianta 3: 12 x NAND 2 intrări

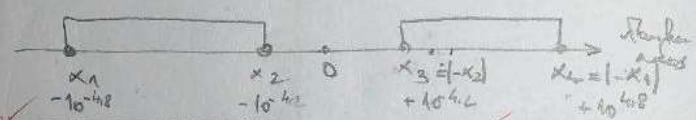
0,25 Varianta 4: 11 x NAND 2 intrări

4p

Subiectul (2) Se consideră un format (ipotetic) de virgulă flotantă pe 14 biți în 5 biți alocați exponentului și 8 biți alocați mantisei. Adoptând normele specifice standardului IEEE 754 la formatul dat, să se determine câmpul valoric al numerelor de reprezentare valide și să se prezinte în secvențe de semne hexazecimale reprezentările numerelor zecimale $-203,0888671875_{10}$, $2^{-15}$ și $2^{12}$.

Soluție: $\boxed{x_S \mid \underbrace{\;|\;|\;|\;|\;}_{x_E} \mid \underbrace{\;|\;|\;|\;|\;|\;|\;|\;}_{x_M}}$

0,25 $X = (-1)^{x_S} \cdot 2^{x_E - 15} (1.x_M)$ unde $0 < x_E < 31$




0,25 $X_1$: $\boxed{1 \mid 1\,1\,1\,1\,0 \mid 1\,1\,1\,1\,1\,1\,1\,1}$

0,25 $X_2$: $\boxed{1 \mid 0\,0\,0\,0\,1 \mid 0\,0\,0\,0\,0\,0\,0\,0}$

0,25 $X_1 = (-1)^1 \cdot 2^{30-15} \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{2^8}\right) = -2^{15} \cdot \frac{511}{2^8} \cong -2^{16} \cong -10^{4,8}$ 0,25

$$\begin{array}{l} 10 \dots 3 \\ 16 \dots x \\ \hline x = 4,8 \end{array}$$

0,25 $X_2 = (-1)^1 \cdot 2^{1-15} (1 + 0 \dots + 0) = -2^{-14} \cong -10^{4,2}$ 0,25

$$\begin{array}{l} 10 \dots 3 \\ 14 \dots x \\ \hline x = -4,2 \end{array}$$

$X^* = -203,0888671875_{10} = -11001011,0001011011_2 =$ 0,25 0,25

$= -1,1001011000 1011011 \times 2^7$

$x_E - 15 = 7 \Rightarrow x_E = 22$ 0,25

0,25 $\boxed{1 \mid 1\,0\,1\,1\,0 \mid 1\,0\,0\,1\,0\,1\,1\,0}$ $\quad X = 3696_H$ 0,25

$\underbrace{1\,1\,0\,1}_{3}\ \underbrace{1\,0\,1\,1}_{6}\ \underbrace{0\,0\,1\,0}_{9}\ \underbrace{1\,1\,0}_{6}$

$X^{**} = 2^{-12} = 0,000000000001 = 1 \times 2^{-12}$

$x_E - 15 = -12 \Rightarrow x_E = 3$ 0,25

$\boxed{0 \mid 0\,0\,0\,1\,1 \mid 0\,0\,0\,0\,0\,0\,0\,0}$ $\quad X = 0300_H$

0,25 $\quad 0 \quad 3 \quad 0 \quad 0$ $\quad$ 0,25

$X^{***} = 2^{-15} = 0,000000000000001 = 1 \times 2^{-15}$

$x_E - 15 = -15 \Rightarrow x_E = 0$

0,25 $\boxed{0 \mid 0\,0\,0\,0\,0 \mid 0\,0\,0\,0\,0\,0\,0\,0}$ $\quad X = 0000_H$