Bun venit

la Facultatea de Automatică și Calculatoare -

specializarea INFORMATICĂ !

Fundamentele calculatoarelor

curs - 2 ore/săptămână -
Mircea Vlăduțiu

lucrări de laborator - 1 oră/săptămână
Larisa Szatmari
Cătălin Vladu

Milestones in Computer History

- Sumerian civilization – Abacus
  ~ 2700 – 2300 BC ~~ 1755 – 1750 the Hammurabi code of laws

(A) Homo Mechanicus Period
- Blaise Pascal (1623 – 1662) – Pascal's Calculator or Pascaline
- Gottfried Leibniz (1646 – 1716) – Leibniz's Calculating Machine
- Charles Babbage (1791 – 1871)
  1822 Difference Engine
  1831 Analytical Engine
  Joseph Marie Jaquard (1752 – 1834)
  Herman Hollerith (1860 – 1929)
- Ada Lovelace (1815 – 1852)
- George Boole (1815 – 1864)

(B) Homo Electromechanicus Period
- Konrad Zuse (1910 – 1995) – Z1, Z2, Z3, Z4
- Howard H. Aiken (1900 – 1973) – IBM Harvard Mark I
- Alan Turing (1912 – 1954) – Turing Machine

Ⓒ Homo Electronicus Period

* Vacuum Tubes Era (First Generation)

- John Vincent Atanasoff (1903-1995) - Atanasoff-Berry Computer
- Presper Eckert (1919-1995) & John Mauchly (1907-1980)
  - ENIAC (Electronic Numerical Integrator and Calculator)

    17468 electronic vacuum tubes, 7200 crystal diodes, 1500 relays, 70000 resistors, 1000 capacitors, ≈ 5000000 hard-soldered joints;

    27 tons; 2,4m x 0,9m x 30m; 167m²; 150 KW;
  - EDVAC (Electronic Discrete Variable Automatic Computer)

    - the first stored-program computer
  - UNIVAC (Universal Automatic Computer)

    - the first general-purpose electronic digital computer designed for business applications

- John von Neumann (1903–1957)
  - IAS (Institute for Advanced Studies) machine at Princeton
    Arthur Burks (1915–2008) & Herman Goldstine (1913–2004)
  - von Neumann architecture vs non-von Neumann architecture
  - "Probabilistic Logics and the Synthesis of Reliable Organism from Unreliable Components" (1956)
- Maurice Wilkes (1913–2010)
  - EDSAC (Electronic Delay Storage Automatic Computer)
  - Microprogramming
    hardwired vs microprogrammed

★ Transistor Era (Second Generation)

- William Schockley (1910–1989) &
  John Bardeen (1908–1991) &
  Walter Brattain (1902–1987)
  1956 Nobel Prize in Physics

⋮

Intel Tick-Tock model → Roadmap

Nehalem		Sandy Bridge		Haswell	
... Nehalem	Westmere	Sandy Bridge	Ivy Bridge	Haswell	Broadwell
45nm	32nm	32nm	22nm	22nm	14nm
Tock	Tick	Tock	Tick	Tock	Tick

Skylake					Icelake	
Skylake	Kaby Lake	Coffee Lake	Whiskey Lake	Cannon Lake	Icelake	Tiger Lake
14nm	14nm	14nm	10nm	10nm	10nm	...

- Intel vs AMD Roadmap AMD - Advanced Micro Devices

{Mainstream Desktop
Mobile Gaming
Mobile U-series
Mobile Ultra-low Voltage (ULV)

... Ryzen 5 - Ryzen 7 - Ryzen 9

Matisse - Renoir - Vermeer - Cezanne ... 7nm

Warhol - Rembrandt ... 6nm

Raphael - Phoenix ... 5nm

* Low Power Problem → Multicore → Manycore
  Intel Prescott ... Kunle Olukotun
  Stanford University

* Development Trends
  → Nanotechnology – RISC–ARM architecture
    Graphene 2D SoC
  → Quantum Computing
    { Paul Benioff, Yuri Manin
    { Richard Feynmann, David Deutsch
    { Superposition principle, Entaglement
    { Qubit
    { Computing power
    {   RSA (Rivest-Shamir-Adleman) cryptosystem
    D-Wave (2007)
  → Bioinspired Computing
    { Artificial Intelligence – Neural Networks
    { Machine Learning – Deep Learning ...
    Deep Blue vs Garry Kasparov 1996/1997
    Google Machine vs Lee Sedol – March 2016
    Alpha Go

CPU (Central Processing Unit)
GPU (Graphics Processing Unit)
TPU (Tensor Processing Unit)

Memristor - Leon Chua
MRAM (Magnetoresistive Random Access Memory)
Neuromorphic Computing
- - - -

(D) Homo Informaticus Period

- Arpanet (Advanced Research Project Agency Network)
  1963 - 1967 - 1969 UCLA
- Transmission Control Program
  1974 TCP/IP (Transmission Control Protocol/ Internet Protocol
  Robert Kahn (1938-) & Vint Cerf (1943-)
- Internet
  Global system of interconnected computer networks that used TCP/IP to communicate between networks and devices
- Tim Berners-Lee (1955-)
  1989 World Wide Web (WWW)

1990 first web browser - CERN Geneva
Resources - Hypertext Transfer Protocol (HTTP)
Web Pages - Hypertext Markup Language (HTML)
• Languages suited to web development

* High-level programming languages
ALGOL
FORTRAN (1957) - John Backus (1924 - 2007)
LISP (1959) - John McCarthy (1927 - 2011)
BASIC (1959) - John Kemeny (1926 - 1992) &
Thomas Kurtz (1928 -)
COBOL (1960) - Grace Murray-Hopper (1906 - 1992)
Time-sharing - Multiprogramming - Multiprocessing
Pascal (1971) - Niklaus Wirth (1934 -)
C (1972) - Dennis Ritchie (1941 - 2011)
C++ (1985) - Bjarne Stroustrup (1950 -)
Java (1995) - James Gosling (1955 -)

C# (2000) - Anders Hejlsberg (1960 -)
Python (1991) - Guido van Rossum (1956-)
PHP (1994) - Hypertext Preprocessor -
Rasmus Lerdorf (1968 -)
JavaScript (1995) - Brendan Eich (1961 -)
Ruby on Rails (2005) - David Heinemeier Hansson (1979-)
⋮

* Operating Systems

- Mainframes Era
  IBM - OS/360, DOS/360, ...
  DEC (Digital Equipment Corporation) - TOPS-10
- Minicomputers Era
  Unix (1973) - Ken Thompson (1943-) &
  Dennis Ritchie (1941-2011)
  ⋮
  Linux (1991) - Linus Torvalds (1969 -)
- Microcomputers Era
  CP/M (1974) - Control Program/Monitor)
  David Kildall (1942-1994)

Windows NT (1993) - Microsoft - versions
Macintosh operating System
* Industry

- Microsoft (1975) - Bill Gates (1955-) &
  Paul Allen (1953 - 2018)
- Apple (1976) - Steve Jobs (1955-2011) &
  Steve Wozniak (1950-) &
  Ronald Wayne (1934-)
- Google (1998) - Larry Page (1973-) &
  Sergey Brin (1973-)
- Amazon (1994) - Jeff Bezos (1964-)
  Blue Origin (2000)
- Pay Pal (1998) - Elon Musk (1971-)
  Space X (2002)
  Bitcoin - Blockchain
- Facebook (2004) - Mark Zuckerberg (1984-)

⋮

Listă bibliografică pentru pregătirea examenului la disciplina “Fundamentele calculatoarelor”-an 1 INF-2021/2022

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.

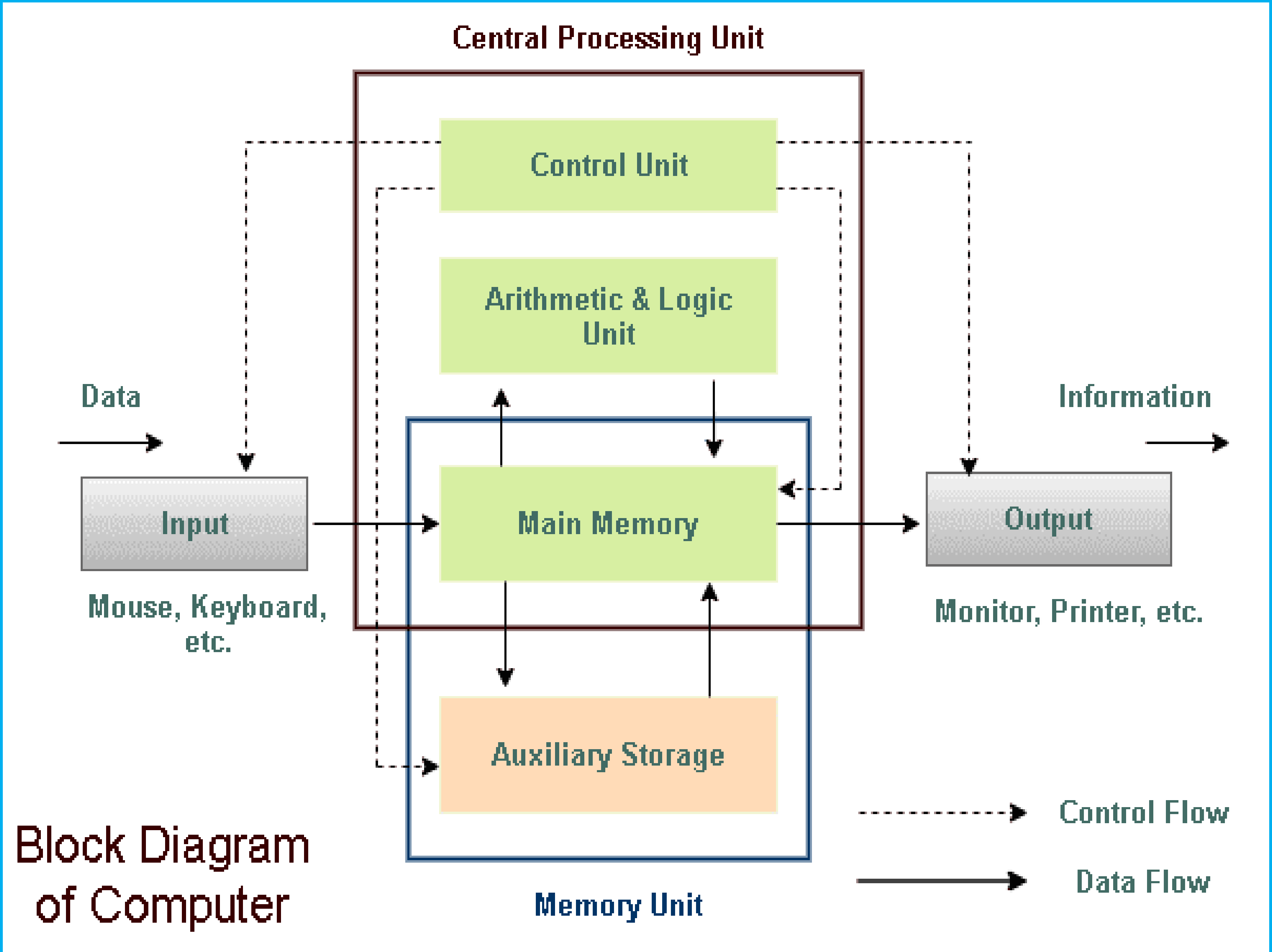
Listă bibliografică pentru pregătirea examenului la disciplina “Fundamentele calculatoarelor”-an 1 INF-2021/2022

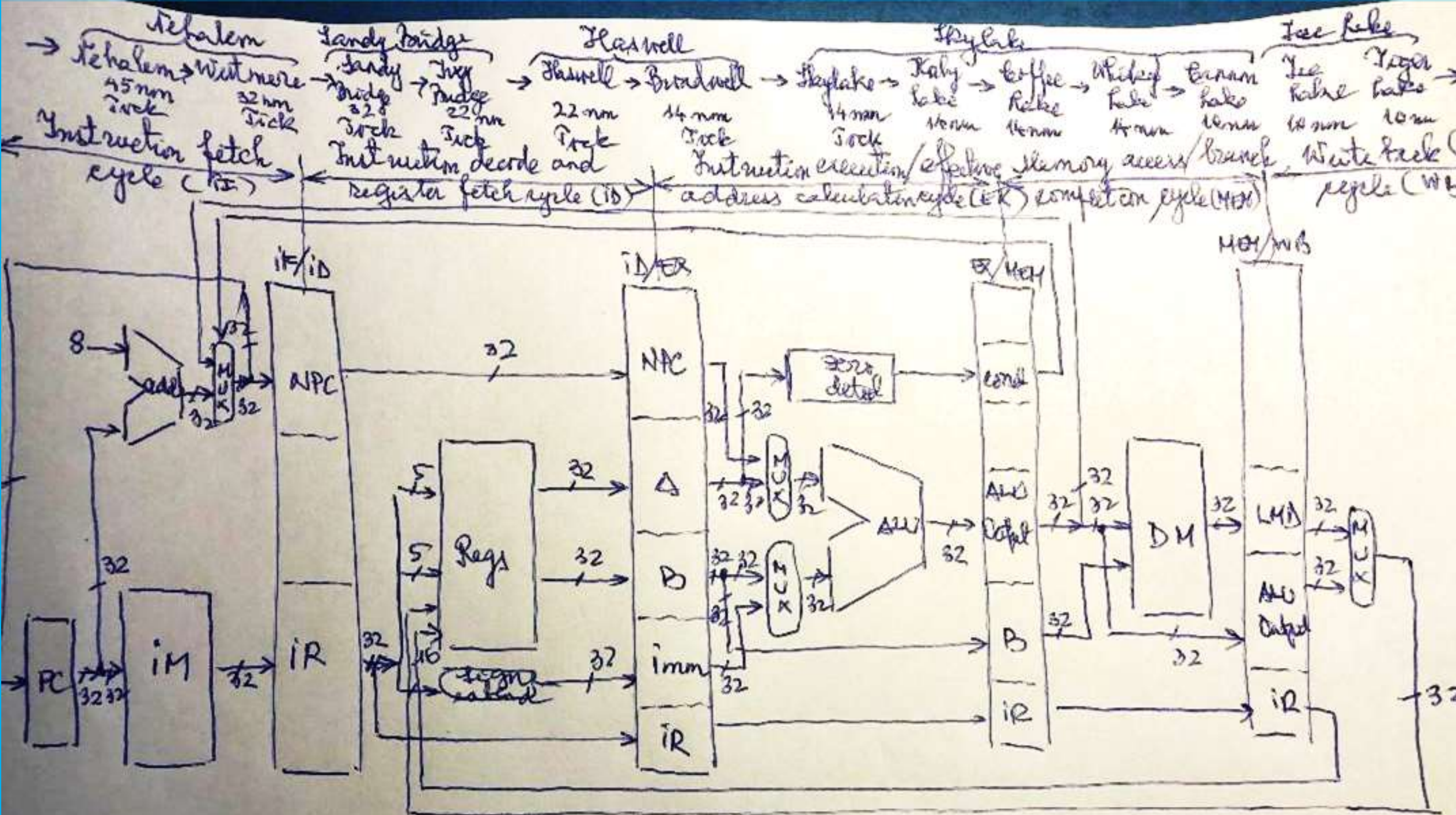
6. Alan B. Marcovits: „Introduction to Logic Desgn” Third Edition, Paperback, 2015.

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

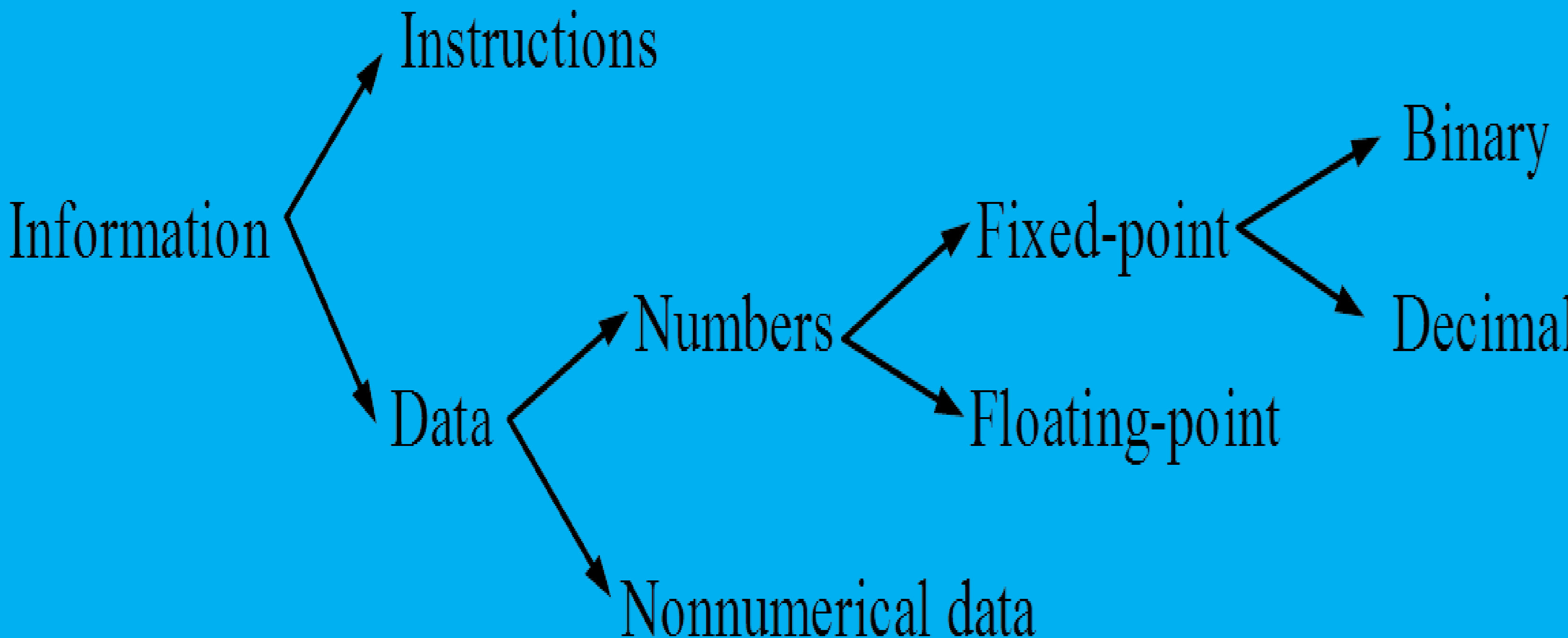

Central Processing Unit
Control Unit
Arithmetic & Logic Unit
Data
Information
Input
Main Memory
Output
Mouse, Keyboard, etc.
Monitor, Printer, etc.
Auxiliary Storage
Control Flow
Data Flow
Block Diagram of Computer
Memory Unit



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


Instructions
Information
Data
Numbers
Fixed-point
Floating-point
Binary
Decimal
Nonnumerical data
Fig.1.1

integer

707_{10} digital to binary conversion

remainder method

1011000011_2

Computer

1011000011_2

binary to digital conversion

707_{10}

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

707	2	
1	353	2
1	176	2
0	88	2
0	44	2
0	22	2
0	11	2
1	5	2
1	2	2
0		

$$707_{10} = 1011000011_2$$

$$1011000011_2 = 1\times2^9+0\times2^8+1\times2^7+1\times2^6+1\times2^1+1\times2^0 =$$
$$= 512+128+64+2+1 = 707_{10}$$

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

radix

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

Fractions

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

$0{,}046875_{10}$

multiplication method

digital to binary conversion

$0{,}000011_2$

Computer

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

$0{,}000011_2$

binary to decimal conversion

$0{,}046875_{10}$

$0{,}046875 \times 2$

$0{,}093750 \times 2$

$0{,}187500 \times 2$

$0{,}375000 \times 2$

$0{,}750000 \times 2$

$1{,}500000 \times 2$

$1{,}\boxed{000000}$

$2^{-1} \quad \cdots \quad 2^{-5}\,2^{-6}$

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

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

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

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)

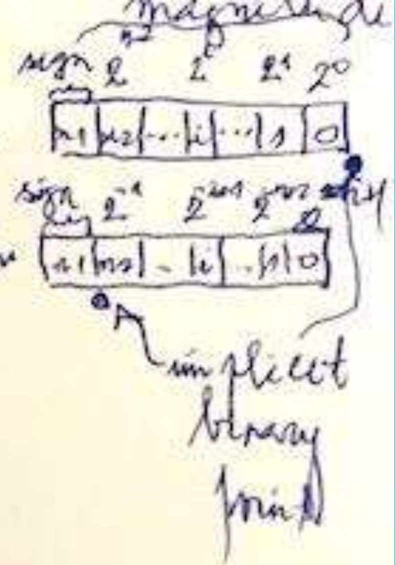
n-1	n-2	…	i	…	1	0

$\underbrace{\qquad\qquad}_{n \text{ bits}}$ computer word

Binary fixed-point numbers → Sign-magnitude (SM) → integer / fraction

→ One's complement (C1)

→ Two's complement (C2)



	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
$N_3 = +0,34375$	0 0 1 0 1 1 0 0	0 0 1 0 1 1	0 0 1 0 1 1 0 0
$N_4 = -0,34375$	1 0 1 0 1 1 0 0	1 1 0 1 0 0 1 1	1 1 0 1 0 1 0 0

SM: sign 2^{6} … 2^{1} 2^{0} (integers); sign 2^{-1} … 2^{-6} 2^{-7} (fractions)

C1: sign 2^{6} … 2^{1} 2^{0}; sign 2^{-1} … 2^{-6} 2^{-7}

C2: sign 2^{6} … 2^{1} 2^{0}; sign 2^{-1} … 2^{-6} 2^{-7}

no weights!

no weights!

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

computer word

$\underbrace{\qquad\qquad}_{n \text{ bits}}$

Binary fixed-point numbers

- Sign-magnitude (SM)
  - integer: sign, magnitude $2^{n-2} \ldots 2^{i} \ldots 2^{1}\ 2^{0}$ — [n-1 | n-2 | ... | i | ... | 1 | 0]
  - fraction: sign, $2^{-1} \ldots 2^{-i} \ldots 2^{-(n-1)}$ — [n-1 | n-2 | ... | i | ... | 1 | 0]; implicit binary point
- One's complement (C1)
- Two's complement (C2)

	SM	C1	C2
	sign 2^{6} --- $2^{1}\ 2^{0}$	sign 2^{6} --- $2^{1}\ 2^{0}$	sign 2^{6} --- $2^{1}\ 2^{0}$
$N_1 = +117_{10}$	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
		no weights	no weights
	sign 2^{-1} ... $2^{-6}\ 2^{-7}$	sign 2^{-1} ... $2^{-6}\ 2^{-7}$	sign 2^{-1} ... $2^{-6}\ 2^{-7}$
$N_3 = +0{,}34575$	0 0 1 1 0 1 1 1 0 0	0 0 1 0 1 1 1	0 0 1 1 0 1 1 0 0
$N_4 = -0{,}34575$	1 0 1 1 0 1 1 0 0	1 1 0 1 0 0 1 1	1 1 0 1 0 1 0 0
		no weights!	no weights!

C1

$$X_{C1} = \bar{X} =$$

≥ 0:

$$\begin{array}{cccccccc} \text{sign} & 2^{n-2} & & 2^i & & 2^1 & 2^0 \\ 0 & x_{n-2} & \dots & x_i & \dots & x_1 & x_0 \end{array}$$

the 1 complement

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

≤ 0:

$$1\ \bar{x}_{n-2} \dots \bar{x}_i \dots \bar{x}_1\ \bar{x}_0$$

negative numbers

$$\begin{array}{ccccccccc} \text{sign} & 2^{n-2} & 2^{n-3} & & 2^i & & 2^1 & 2^0 \\ 1 & 1 & 1 & \dots & 1 & \dots & 1 & 1 \end{array}$$

$$X_{C1} = 111\dots1\dots11 - \underbrace{x_{n-2}\, x_{n-3} \dots x_i \dots x_1\, x_0}_{\text{magnitud } X_M}$$

$$\to 2^{n-1} + 2^{n-2} + \dots + 2^i + \dots + 2^1 + 2^0 =$$

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

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

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

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

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

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

$$1\times2^{n-1} + 1\times2^{n-2} + 1\times2^{n-3} + \cdots + 1\times2^i + \cdots + 1\times2^1 + 1\times2^0 =$$

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

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

$$= 2^n - 1$$

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

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

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

$$X_{C2} = -X = X_{C1} + 000\cdots0\cdots01 \quad \text{for fractions}$$

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

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

$$1\times2^0 + 1\times2^{-1} + 1\times2^{-2} + \cdots + 1\times2^{-i} + \cdots + 1\times2^{-n+2} + 1\times2^{-n+1} =$$

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

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

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

$$= 2 - X_M$$

↑ name of C2

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

* 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

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

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

$N = -88_{10}$

| 1 | 0 1 0 1 0 0 0 |

James Robertson

$$\begin{cases} X_{C2} = -x_{n-1}\cdot 2^{n-1} + \sum\limits_{i=0}^{n-2} x'_i \cdot 2^i & \text{for } X \text{ integer} \\ X_{C2} = -x_{n-1}\cdot 2^{0} + \sum\limits_{i=1}^{n-1} x'_{n-i-1} \cdot 2^{-i} & \text{for } X \text{ fractional} \end{cases}$$

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

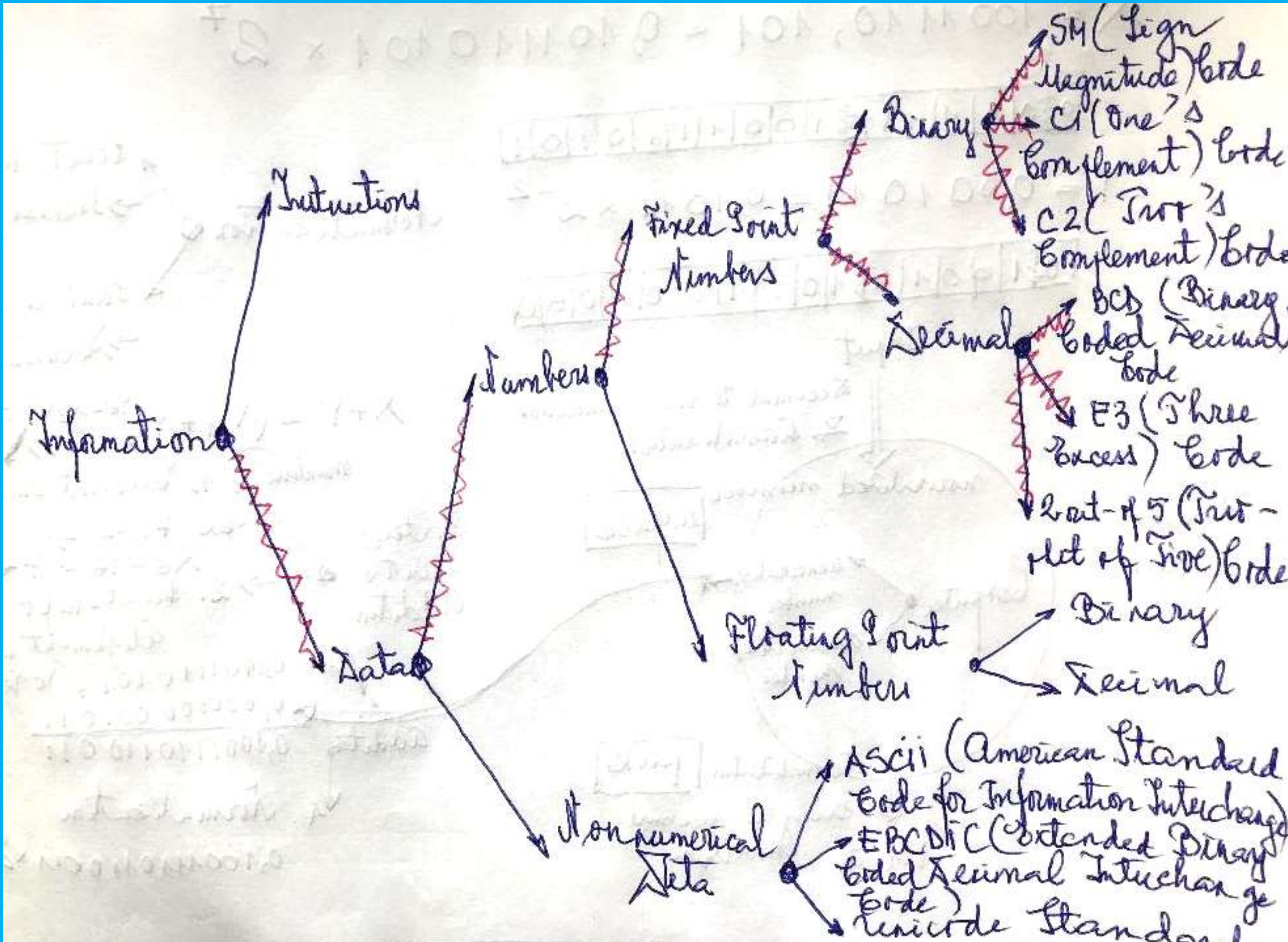
$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

| 1 | 0 1 0 1 0 0 0 | $\rightarrow X_{C2} = -1\times 2^7 + \sum\limits_{i=0}^{6} x'_i \cdot 2^i = -1\times 2^7 + (1\times 2^5 + 1\times 2^3) =$

$= -128 + 32 + 8 = -88$

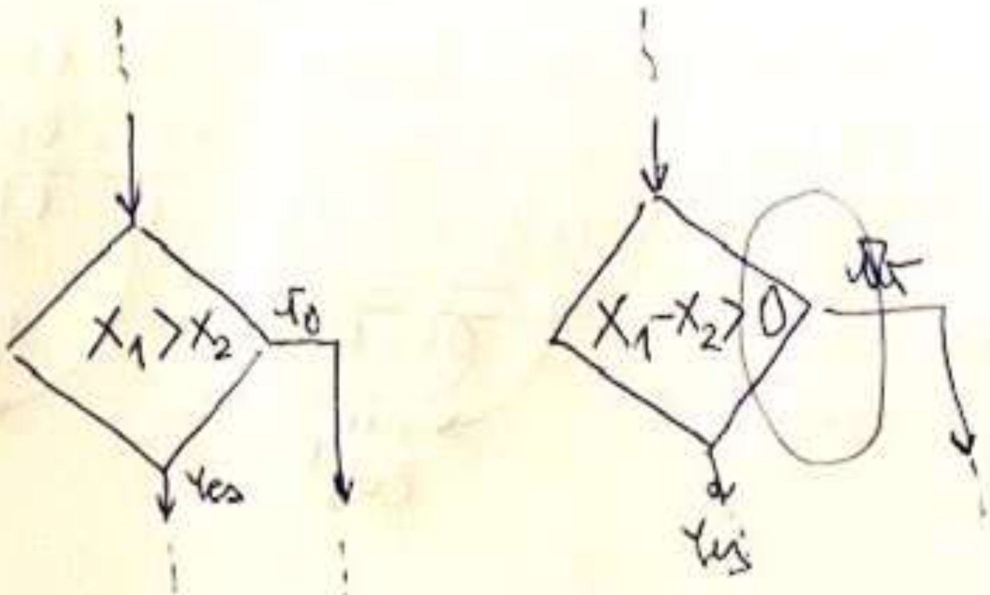
* C2's anomaly

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


Information
Instructions
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
2out-of 5 (Two-out of Five) Code
Floating Point Number
Binary
Decimal
Nonnumerical Data
ASCII (American Standard Code for Information Interchange)
EBCDIC (Extended Binary Coded Decimal Interchange Code)
Unicode Standard

0's representation

- SM
  - (+) 0 : 0 0 ... 0 ... 0 0
  - (−) 0 : 1 0 ... 0 ... 0 0
- C1
  - (+) 0 : 0 0 ... 0 ... 0 0
  - (−) 0 : 1 1 ... 1 ... 1 1
- C2 → (+) 0 : 0 0 ... 0 0 0



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

Decimal

$$\begin{array}{r} 7+ \\ 6 \\ \hline 1\,3 \end{array}$$

carry · sum

10^1 Digits

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

2^1

carry bit · sum bit

Binary

sign magnitude

SM

$$\begin{array}{l} X = X_S\,X_M \;+ \\ Y = Y_S\,Y_M \\ \hline Z = Z_S\,Z_M \end{array}
\qquad
\begin{array}{l} X = 0\,X_M \;+ \\ Y = 0\,Y_M \\ \hline Z = 0\,Z_M \end{array}
\qquad
\begin{array}{l} X = 0\,X_M \;+ \\ Y = 1\,Y_M \\ \hline Z = 1\,Z_M \end{array}
\qquad
\begin{array}{l} X = 1\,X_M \\ Y = 0\,Y_M \\ \hline Z = 1\,Z_M \end{array}
\qquad
\begin{array}{l} X = 1\,X_M \\ Y = 1\,Y_M \\ \hline Z = 10\,Z_M \end{array}$$

$X_M > Y_M$ · $X_M < Y_M$

$Z_M = (X_M + Y_M)$

$Z_M = (X_M + Y_M)$ ← false → $Z_M = (X_M + Y_M)$

$Z_M = (X_M - Y_M)$ ← correct → $Z_M = (Y_M - X_M)$

$Z_M = (X_M + Y_M)$ fal…

$Z_M = 1(X_M + Y_M)$ cor…

3 cases false Result!

timepenalty!

1 | 0 | 1 | … |

sign

Z_M

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

a

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

b

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

c

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

d

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

$$\begin{array}{l} X = +3 + \\ Y = -2 \\ \hline Z = +1 \end{array} \qquad \begin{array}{r} X_{SM} = 0\,011 + \\ Y_{SM} = 1\,010 \\ \hline 1101 \end{array}$$

-5 false

$X_M \geq Y_M$

(3) (2)

magnitude comparison!

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

$+(3-2) = +1$

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

0 must be corrected!

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

carry out

must be corrected

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

end around correction

Weighted vs nonweighted fixed point codes

weighted

Sign-magnitude

- integer: sign 2^{n-2} … 2^i … 2^0

n-1	n-2	…	i	…	0

- fraction: sign 2^{-1} … 2^{1+i-n} … 2^{1-n}

n-1	n-2	…	i	…	0

Binary coded decimal

sign 10^{n-2} … 10^i … 10^0

n-1	n-2	…	i	…	0

sign $2^3\ 2^2\ 2^1\ 2^0$ … $2^3\ 2^2\ 2^1\ 2^0$ … $2^3\ 2^2\ 2^1\ 2^0$

$10^i \cdot 2^j$

nonweighted

One's complement

Two's complement

Three excess

Two-out-of-five

Binary vs Decimal

$n = 10 \rightarrow 2^{10} = 1024 \cong 1000 = 10$

n bits $\rightarrow 2^n$ codes (words)

$\frac{n}{4}$ decimal digits $\rightarrow 2^{0{,}83n}$ codes

$$\left.\begin{array}{l} 10 \text{ ---- } 3 \\ x \text{ ---- } \frac{n}{4} \end{array}\right\} \Rightarrow x = \frac{10n}{12} \cong 0{,}83n$$

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

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

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

$\Rightarrow$ 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 $0 \leq x_i < r$
- scientific notation: $N = M \cdot B^E$

E ← exponent

M ← mantissa

B ← base

X_S X_E X_M

sign	exponent	mantissa
$\mid$	$\mid \ldots \mid$	$\mid \ldots \mid$

$$\boxed{X = (-1)^{\overbrace{X_S}^{\text{sign}}} \cdot \underbrace{X_B}_{\text{base}}^{\overbrace{X_E}^{\text{exponent}}} \cdot \underbrace{X_M}_{\text{mantissa}}}$$

Input

$\Downarrow$ normalized floating point number

Computer

floating point number → unnormalized

floating point number → denormalized

$\Downarrow$ normalized

Output floating point number

normalized floating point number

$\xrightarrow{SM}$ $\overbrace{0}^{\text{sign}}, \overbrace{1 \ldots}^{\text{magnitude}}$ $\underbrace{0,1\ldots}_{\text{mantissa}}$

$\xrightarrow{C2}$ $\overbrace{x_S}^{\text{sign}}, x_{M\,msb} \ldots$ $\underbrace{x_S, x_{M\,msb}\ldots}_{\text{mantissa}}$

Mantissa problems

Error minimum X_E - k bits

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

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

Bias (Excess)

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

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

Exponent problem

$X = 100111 0,101 = 0,10111 0101 \times 2^7$

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

Normalization
- Point leftshift → Increasing exponent
- Point Rightshift → Decreasing exponent

Input

Decimal to binary conversion & Normalization

normalized number

unpack

Mantissa Problem

Computer
- unnormalized number
- denormalized number

Normalization

pack

Binary to decimal

Output conversion

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

mantissa

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
3. Addition

ex

$$\begin{array}{r} 0,10011101 01 \\ + \; 0,000000000 01011 \\ \hline 0,10011101 10011 \end{array}$$

4. Normalization

$0,10011101,10011 \times 2^7$

Normalization rules → Mantissa

SM:

$$\overbrace{0/1 \ddagger \; - - - - \;}^{X_E}\overbrace{\ddagger 1 | \; - - - \; |}^{X_M}$$

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

C2:

$$0/1 \ddagger \; \ddagger 1/0 | \; — \; |$$

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

* Observation:

- through normalization operation

$$\frac{1}{2} \leq \underbrace{|X_M|}_{\text{absolute value}} < 1$$

Exponent Problem

$X = +0,2_{10} = +0,0010001,\dots_2$

0,2 × 2
0,4 × 2
0,8 × 2
1,6 × 2
0,2 × 2
0,4 × 2
0,8 × 2
1,6 × 2
0,2

$$\overbrace{0 \ddagger 0|0|1|0|0|1|0}^{X'}$$

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

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

* from conversion algorithm → truncation error

Normalization rules → Mantissa

SM:

$\overbrace{0/1 \ldots\ldots}^{X_E}\overbrace{1 \ldots\ldots}^{X_M}$

for normalization, point leftshift ⇄ increasing exponent or point rightshift ⇄ decreasing exponent

C2:

| 0/1 | ---- | 1/0 | — |

for normalization, point leftshift ⇄ increasing exponent or point rightshift ⇄ decreasing exponent, but WARNING! for negative mantisse rightshift introducing 1_s!!

* Observation:

- through normalization operation

$$\frac{1}{2} \le |X_M| < 1$$

absolute value

Exponent Problem

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

0,2 × 2
0,4 × 2
0,8 × 2
1,6 × 2
0,2 × 2
0,4 × 2
0,8 × 2
1,6 × 2
0,2

X': | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

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

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

* from conversion algorithm → truncation error

Errors → Truncation errors

Errors → Rounding errors → Rounding process ⇒ Time penalty!

* By floating point operation → Partial result O mente

because errors → $Z = (-1)^{x_{n-1}} \cdot X_M \cdot 2^{X_E}$ → significant errors

X_M very small; X_E large

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 | 0 | 0 — 0 | 0 | 1 — 1 |

e bits

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

0 representation as $\boxed{0/1} \; \underbrace{\boxed{0 \,|\, \cdots \,|\, 0}}_{x_E} \; \underbrace{\boxed{0 \,|\, \cdots \,|\, 0}}_{x_M}$

and we have

- SM $\rightarrow$ $\boxed{0/1} \; \boxed{1 \,|\, 1 \,|\, \cdots \,|\, 1 \,|\, 1} \; \boxed{0 \,|\, \cdots \,|\, 0}$
- $\rightarrow$ $\boxed{0/1} \; \boxed{0 \,|\, \cdots \,|\, 0 \,|\, 0} \; \boxed{0 \,|\, \cdots \,|\, 0}$

$\rightarrow$ Excess (Biased) Representation

- SM $\rightarrow$ Excess (Bias) $= 2^{e-1} - 1$
- C2 $\rightarrow$ Excess (Bias) $= 2^{e-1}$

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

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

* 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
(June, 1985 Kahan, 1987 → 854,
2008, Zuras & Cowlishaw, revision 2019 -
Hough & Cowlishaw)

mainframes
↓ IBM Floating Point „Standard"

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

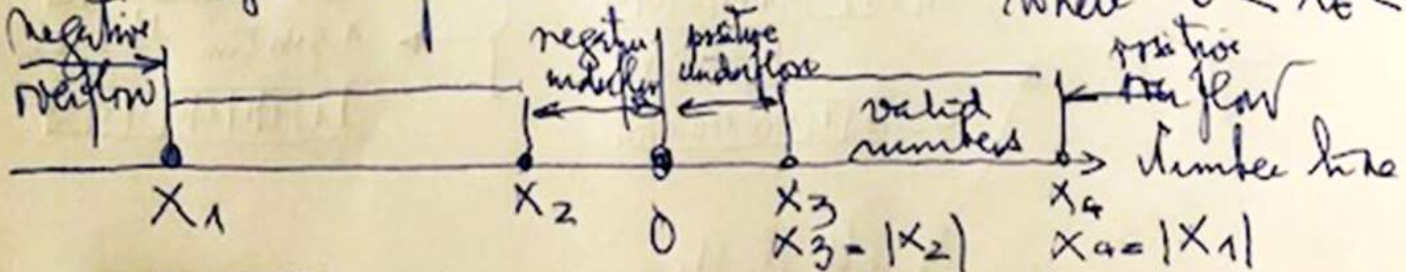
x_S, x_E

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

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

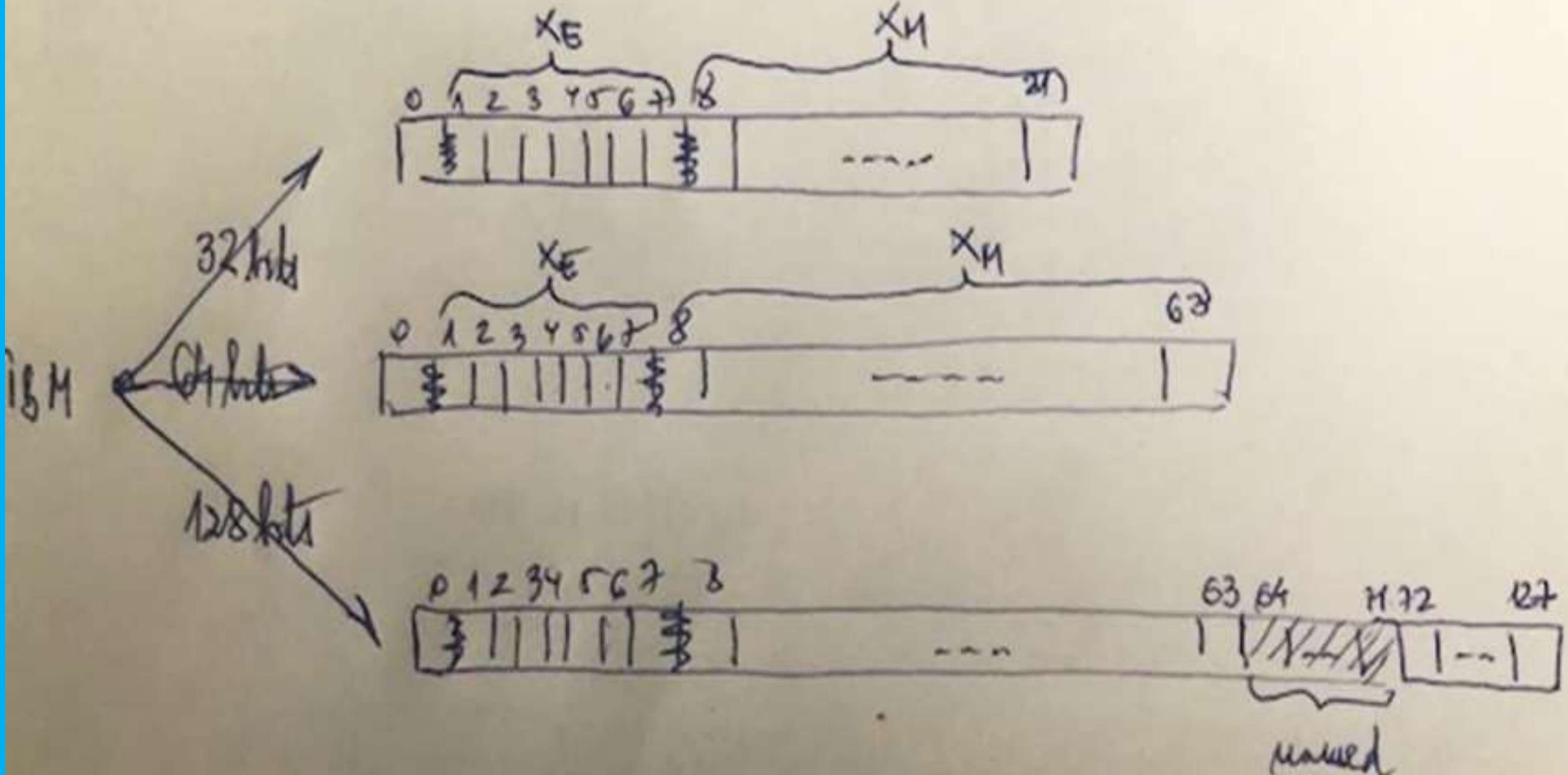
hidden bit

1	0	0	0	0	0	0	0	1	0	...	0

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

IBM Floating Point „Standard"

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



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

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

$794 \,|\, 2$
$0 \quad 397 \,|\, 2$
$1 \quad 198 \,|\, 2$
$0 \quad 99 \,|\, 2$
$1 \quad 49 \,|\, 2$
$1 \quad 24 \,|\, 2$
$0 \quad 12 \,|\, 2$
$0 \quad 6 \,|\, 2$
$0 \quad 3 \,|\, 2$
$1 \quad 1$

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

$$X = -1{,}100011010000010111 \times 2^9$$

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

$$\underbrace{1100}_{C}\ \underbrace{0100}_{4}\ \underbrace{0100}_{4}\ \underbrace{0110}_{6}\ \underbrace{1000}_{8}\ \underbrace{0101}_{5}\ \underbrace{1100}_{C}\ \underbrace{0000}_{0}$$

$$X_{IEEE} = C44685C0_{16}$$

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

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

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

$$\underbrace{1\,|\,1\,|\,0\,|\,0}_{C}\,|\,\underbrace{0\,|\,0\,|\,1\,|\,1}_{3}\,|\,\underbrace{0\,|\,0\,|\,1\,|\,1}_{3}\,|\,\underbrace{0\,|\,0\,|\,0\,|\,1}_{1}\,|\,\underbrace{1\,|\,0\,|\,1\,|\,0}_{A}\,|\,\underbrace{0\,|\,0\,|\,0\,|\,1}_{1}\,|\,\underbrace{0\,|\,1\,|\,1\,|\,1}_{7}\,|\,\underbrace{0\,|\,0\,|\,0\,|\,0}_{0}$$

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

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

$$0\,|\,1\,|\,0\,|\,0\,|\,0\,|\,1\,|\,0\,|\,1\,|\,1\,|\,0\,|\,1\,|\,0\,|\,1\,|\,0\,|\,1\,|\,1\,|\,1\,|\,1\,|\,0\,|\,1\,|\,0\,|\,0\,|\,0\,|\,0\,|\,1\,|\,1\,|\,1\,|\,1\,|\,0\,|\,0\,|\,0\,|\,0$$

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

$$X_{754} = (-1)^0 \cdot 2^{12} \cdot 1,01010111101000001111 =$$

hidden bit

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

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

$$= +5498,1171875$$

$X_{ibm} = 454BD070_{16}$

0	1	0	0	0	1	0	1	0	1	0	0	1	0	1	1	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0

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

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

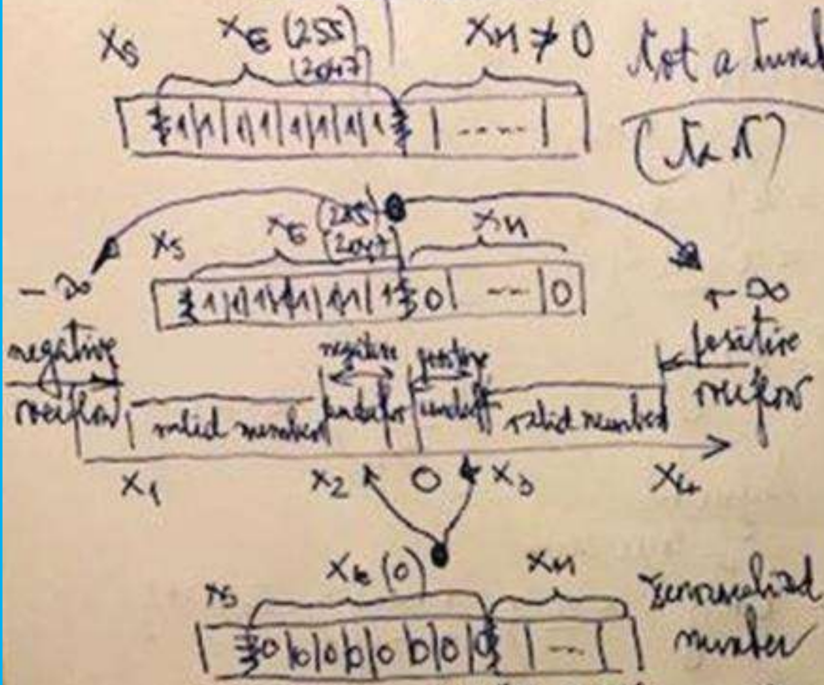
$= +\,1010\,1011\,1101\,0000\,1111 =$

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

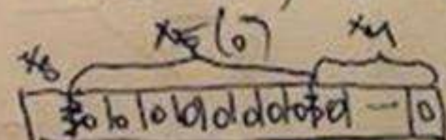
$+ 256 + 8 + 4 + 2 + 1) = +\,703759_{10}$

Exceptions

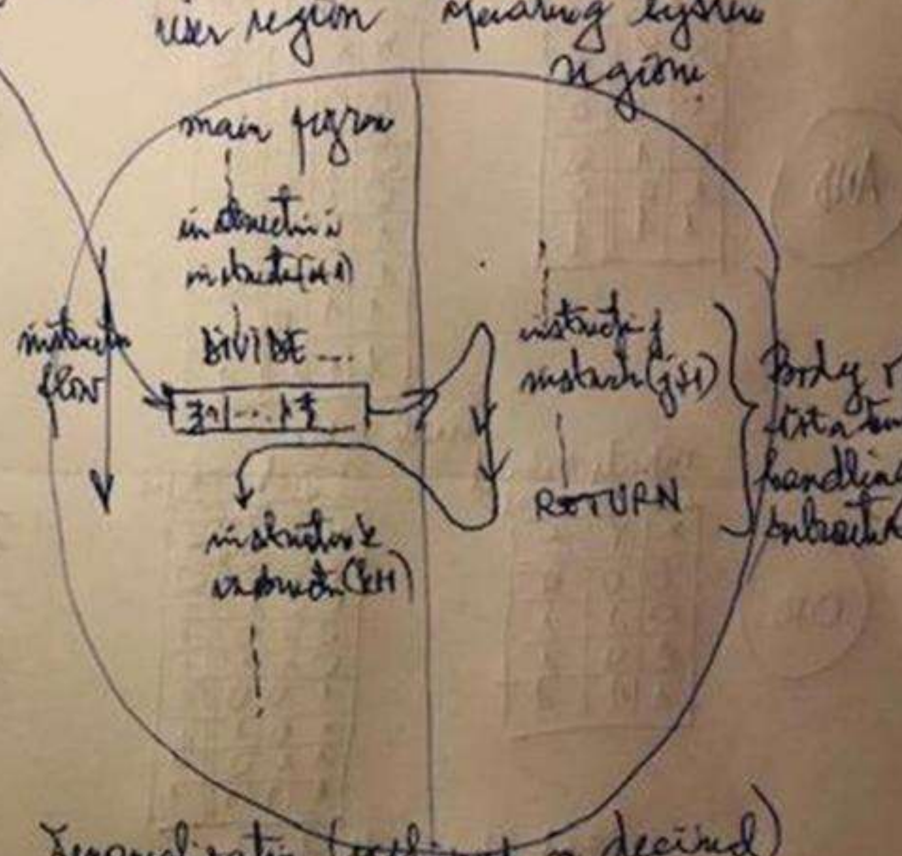
Not a Number (NaN) → divide by zero; square root of a negative number



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



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



Denormalization (explained in decimal)

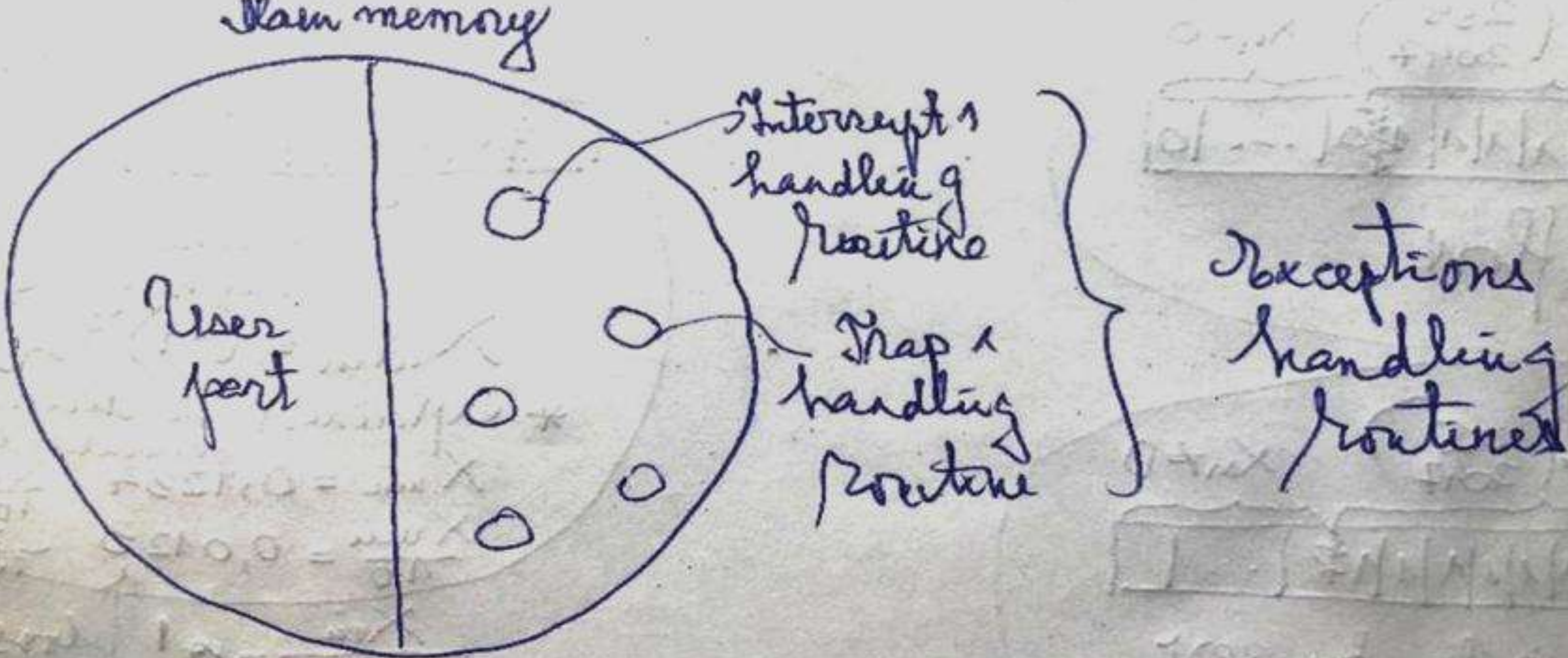
$p = 4$ normalized

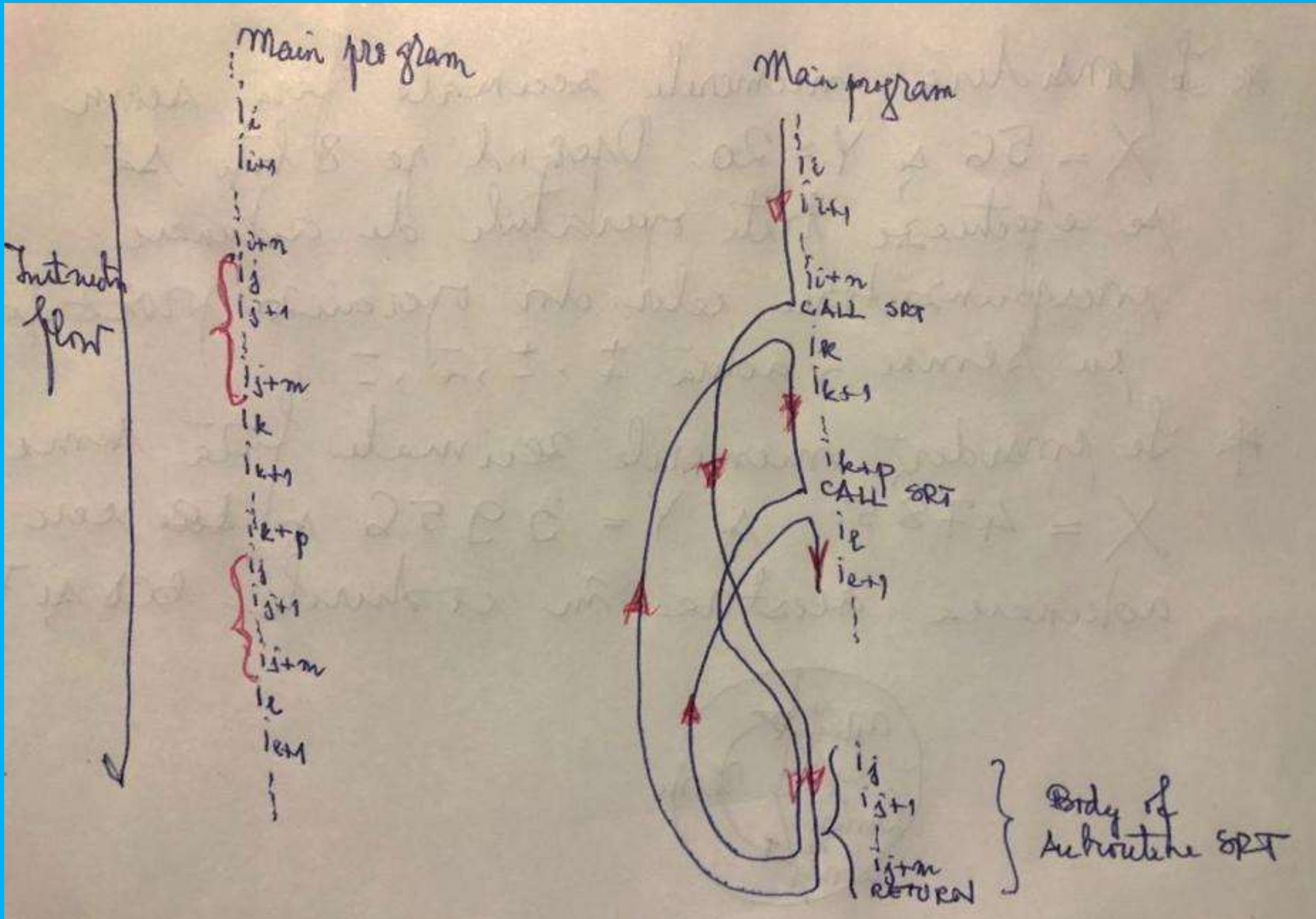
$X_{min} = 0{,}1234$ $\qquad \frac{X_{min}}{100} = 0{,}0012$

$\frac{X_{min}}{10} = 0{,}0123$ $\qquad \frac{X_{min}}{1000} = 0{,}0001$

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

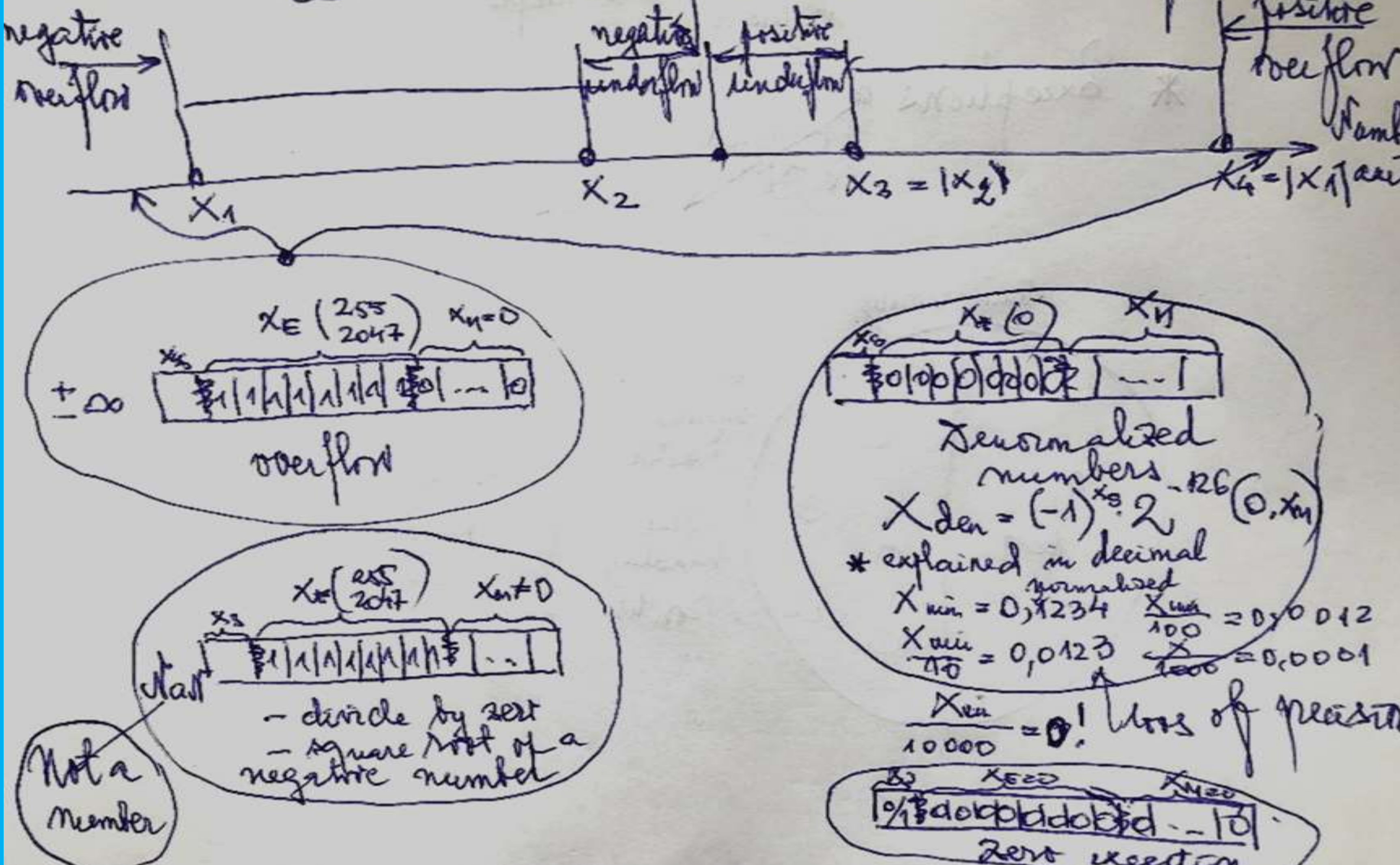
* Exceptions $\begin{cases} \xrightarrow{\text{produced by hardware}} \text{Interrupts} \\ \xrightarrow{\text{produced by software}} \text{Traps} \end{cases}$

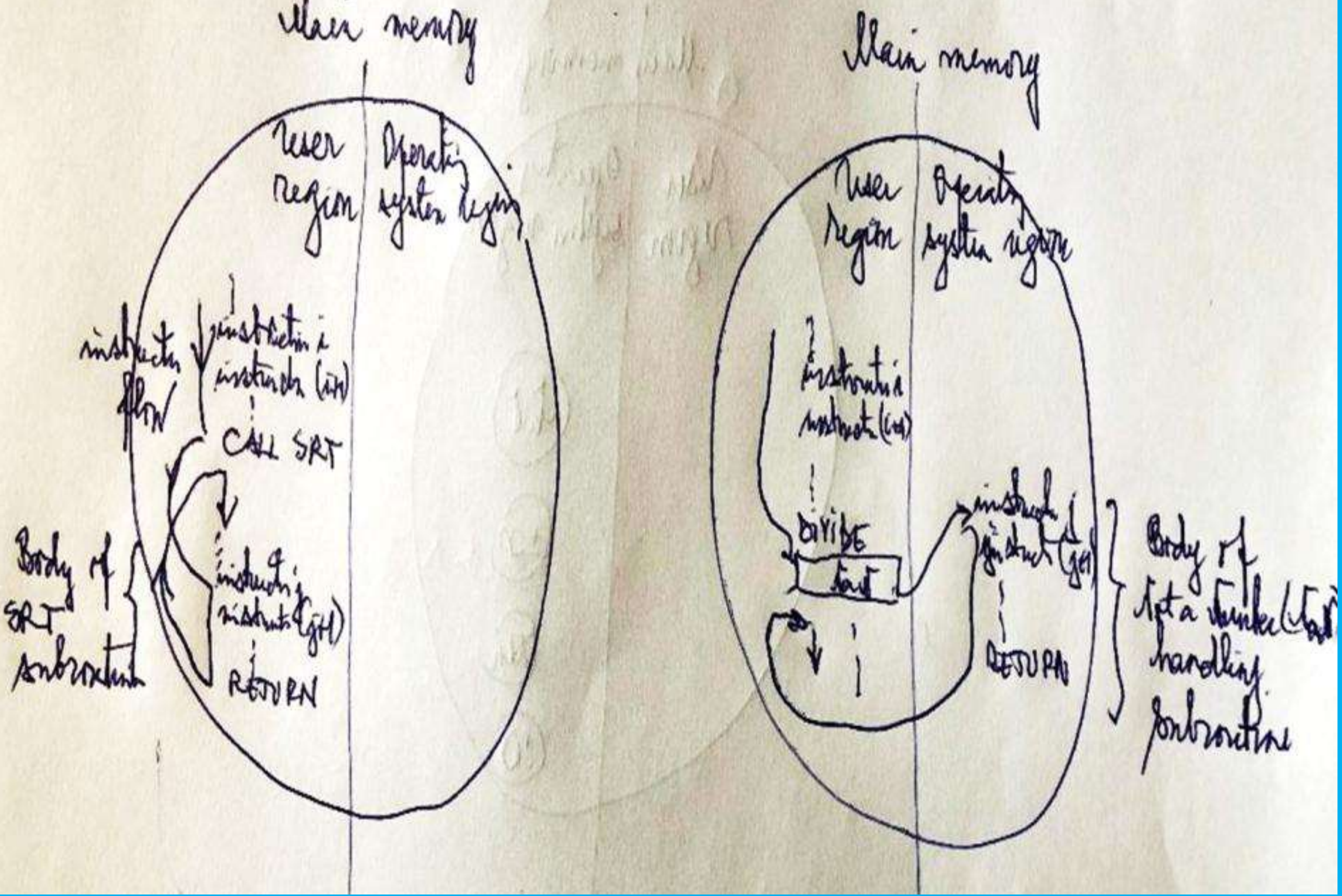


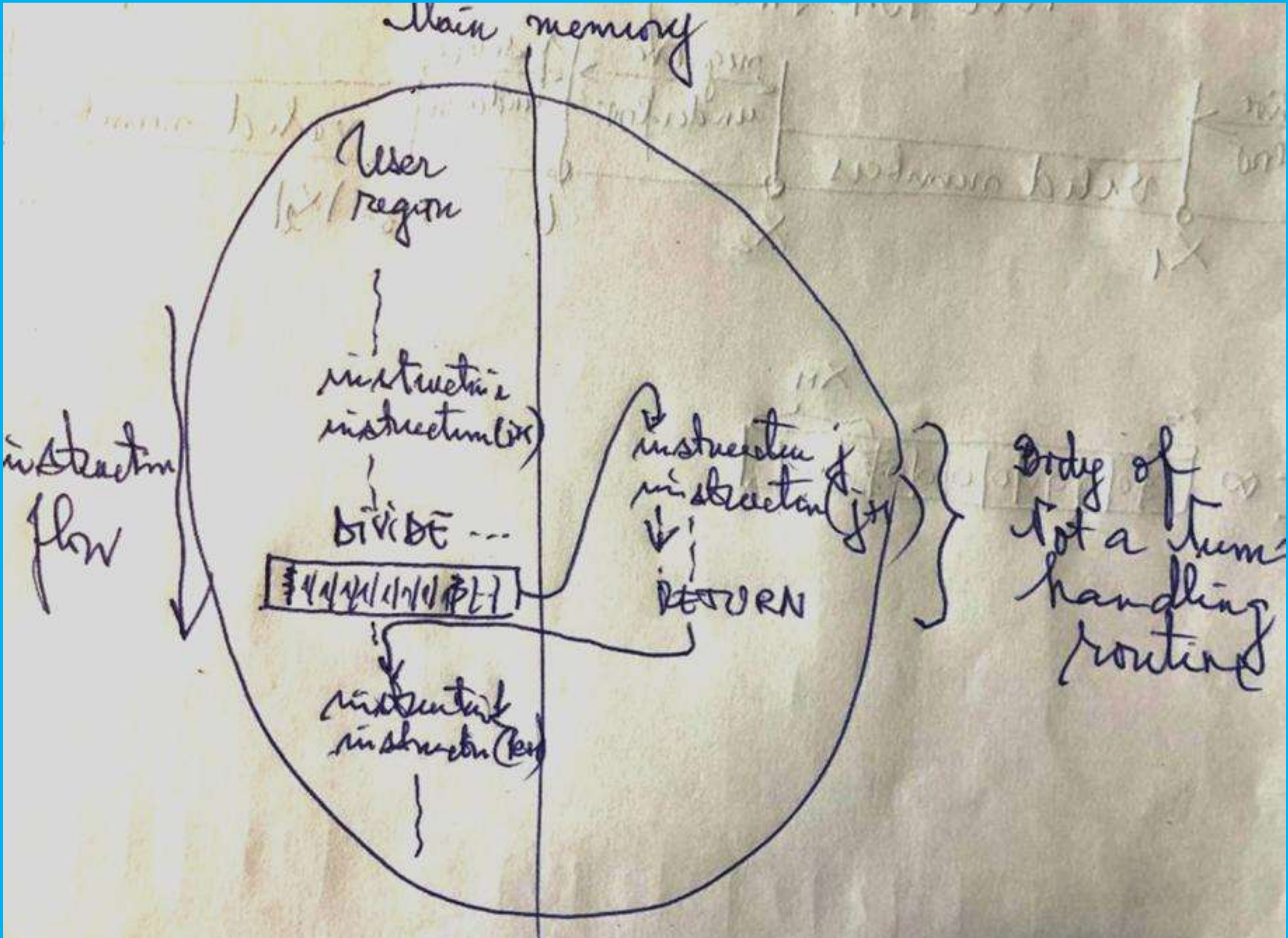


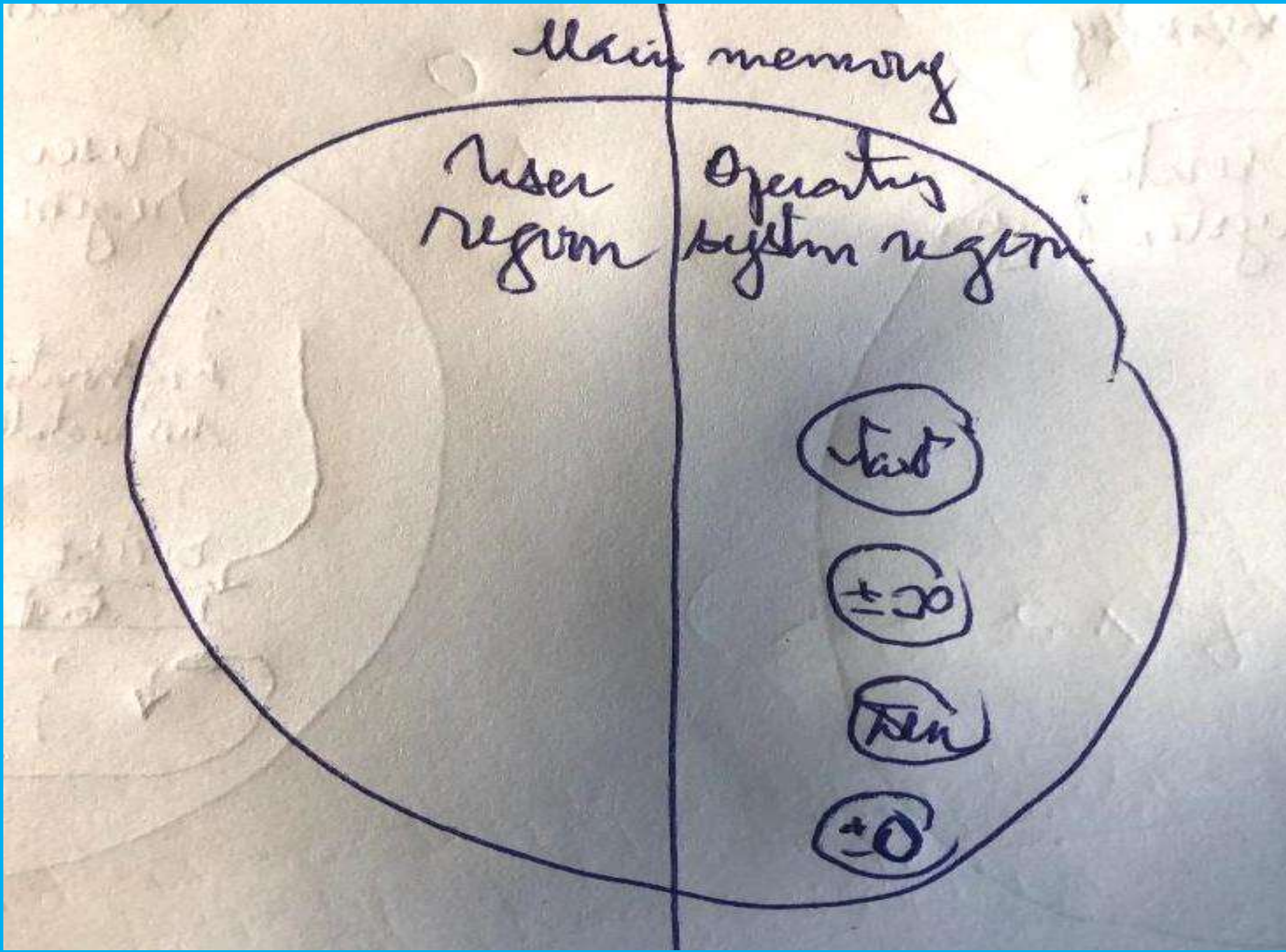
Main program
Intruction flow
i_i
i_{i+1}
i_{i+n}
i_j
i_{j+1}
i_{j+m}
i_k
i_{k+1}
i_{k+p}
i_j
i_{j+1}
i_{j+m}
i_l
i_{l+1}
Main program
i_i
i_{i+1}
i_{i+n}
CALL SRT
i_k
i_{k+1}
i_{k+p}
CALL SRT
i_l
i_{l+1}
i_j
i_{j+1}
i_{j+m}
RETURN
Body of subroutine SRT

IEEE 754 Floating Point Standard Exceptions




Main memory
User region
Operating system region
instruction flow
instruction i
CALL SRT
RETURN
Body of SRT subroutine
Main memory
User region
Operating system region
DIVIDE
RETURN
Body of
handling subroutine


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


Main memory
User region
Operating system region
±∞
±0

Representation of Floating Point Numbers in Single Precision *IEEE 754 Standard*

$$\text{Value} = N = (-1)^S \times 2^{E-127} \times (1.M)$$

1	8	23
S	E	M

sign

0 < E < 255
Actual exponent is:
e = E - 127

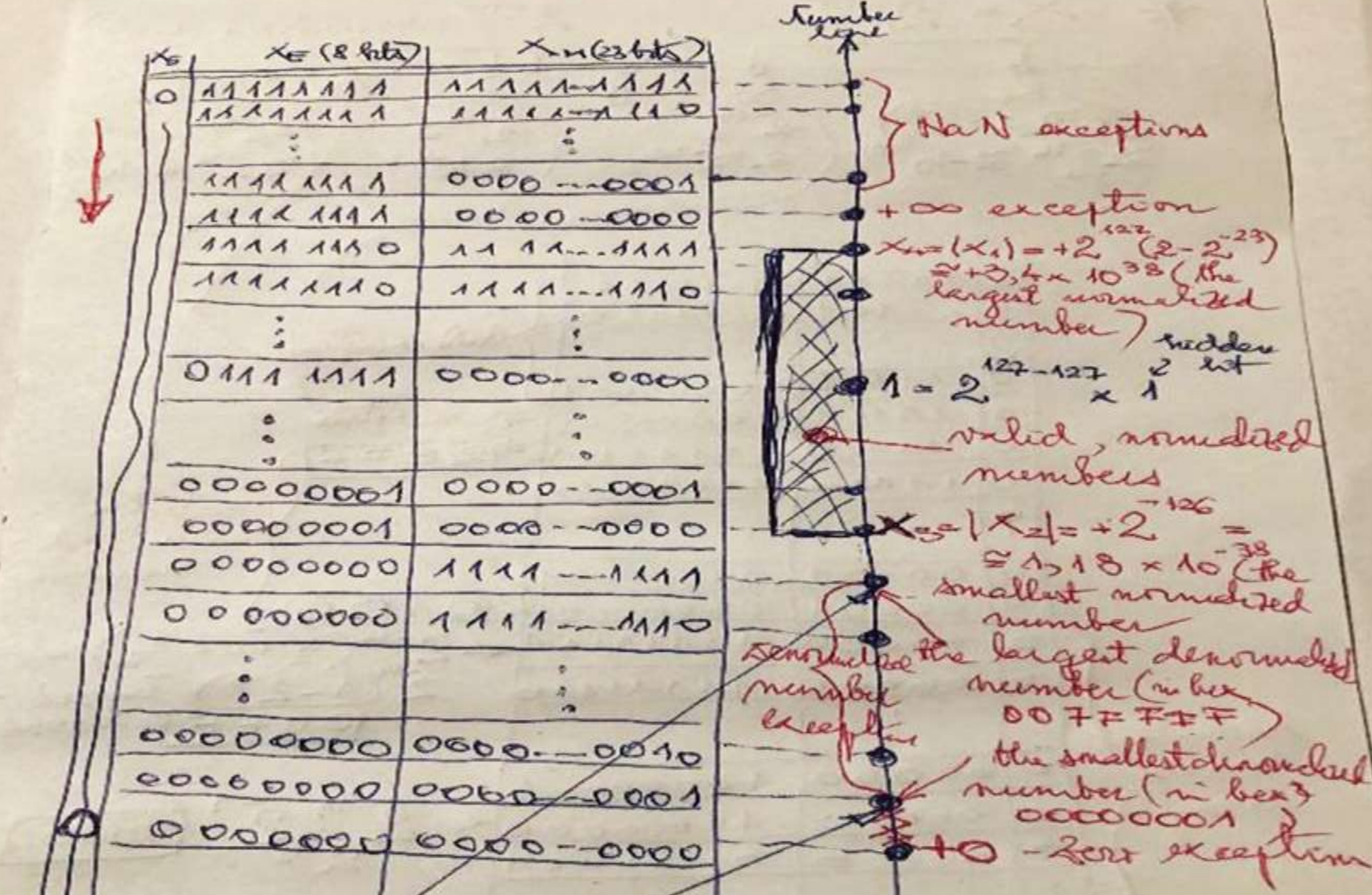
exponent:
excess 127
binary integer
added

mantissa:
sign + magnitude, normalized
binary significand with
a hidden integer bit: 1.M

Example: 0 = 0 00000000 0 . . . 0 -1.5 = 1 01111111 10 . . . 0

Magnitude of numbers that can be represented is in the range: $2^{-126}(1.0)$ to $2^{127}(2 - 2^{-23})$

Which is approximately: 1.8×10^{-38} to 3.40×10^{38}

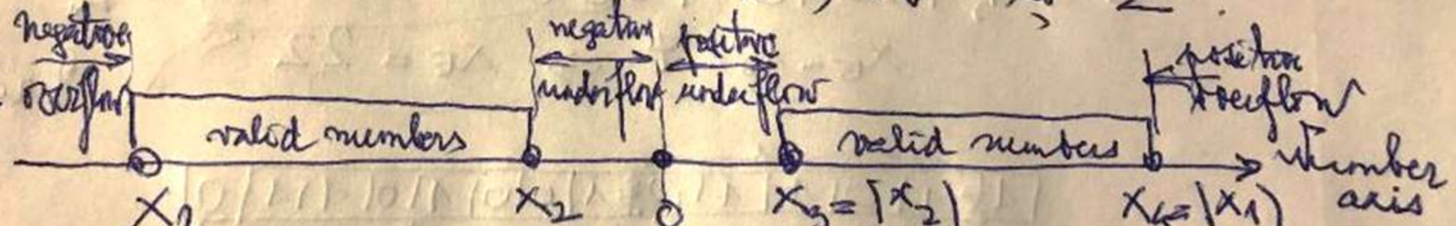


$$+2^{-126}\cdot 2^{-23} = 2^{-149}$$

$$2^{-126}(2^{-1}+2^{-2}+\dots+2^{-23}) = 2^{-126}(1-2^{-23})$$

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:



X1

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 --- 3
16 --- x

x = 4.8

X_2

1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0

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

$$10 - 3$$
$$14 - x$$
$$x = 4,2$$

2a $X^* = -203,08886 71875_{10} = -11001011,000 1011 011_2 =$

$= -1,10010110 00010110 11 \times 2^7$ truncation error

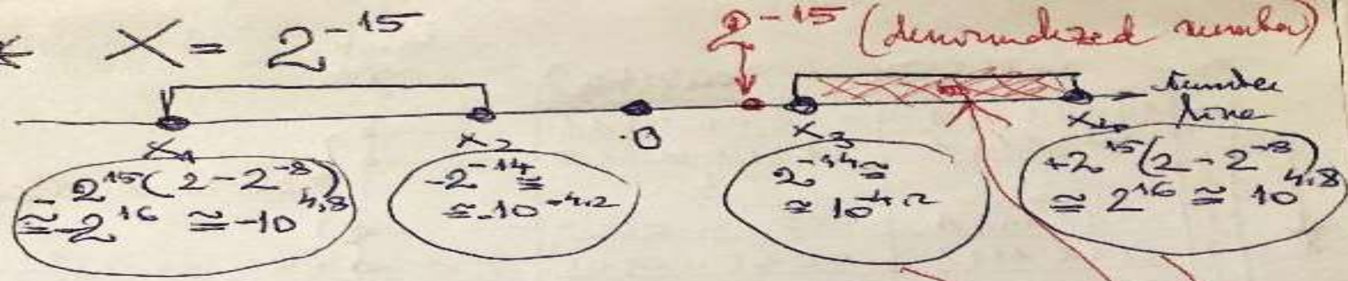
$x_E - 15 = 7 \Rightarrow x_E = 22$

1	1	0	1	1	0	1	0	0	1	0	1	1	0

3 6 9 6

$\Rightarrow X^* = 3696_H$

$* \quad X = 2^{-15}$



0	11111	11111111	NaN exception
0	11111	11111110	
⋮			
0	11111	00000001	
0	11111	00000000	← ∞ exception
0	11110	11111111	← $\boxed{2EFF}$ valid normalized numbers
0	11110	11111110	
⋮			
0	00001	00000001	
0	00001	00000000	← $\boxed{0100}$
0	00000	11111111	← $2^{-14}\left(\frac{1}{2}+\dots+\frac{1}{2^8}\right) = 2^{-14}(1-2^{-8})$ largest denormal $\boxed{00FF}$
0	00000	11111110	
	⋮		
0	00000	10000001	
0	00000	10000000	← $2^{-14} \cdot 2^{-1}$ 2^{-15} $\boxed{0080}$
0	00000	01111111	
⋮			
0	00000	00000001	← $2^{-14} \cdot 2^{-8}$ smallest denormal
0	00000	00000000	+0 Zero exceptions
1	00000	00000000	−0
⋮	⋮	⋮	

denormalized numbers exception

* $X = 2^{12} = 2^{12} \cdot (1 + 0 + \ldots + 0)$

↑ hidden !!!

$X_E - 15 = 12 \longrightarrow X_E = 27_{10} = (16 + 8 + 2 + 1)_{10} = 11011_2$

0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0

1 B 0 0

$X = 2^{12} = 1B00_H$

* $X = -1{,}5_{10} = -1{,}1_2 = -1{,}1 \times 2^0$

$X_E - 15 = 0 \longrightarrow X_E = 15_{10} = 01111_2$

1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0

2 F 8 0

$X = -1{,}5_{10} = 2F80_H$

* 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 $\overset{+}{+}, \overset{+}{-}, \overset{-}{+}, \overset{-}{-}$.

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

(I.) The Representation of Numbers in Computing Systems

(II.) Boolean Algebra

* History - George Boole (1815-1864), Edward Huntington (1874-1952), Claude Shannon (1916-2001) - "the father of information theory", Augustus De Morgan (1806-1871)
* variables > literals

Logic Functions

Truth table

AND

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

for n input variable → 2^n lines

Symbol

$Z = x \cdot y$ (logic product) $= xy =$

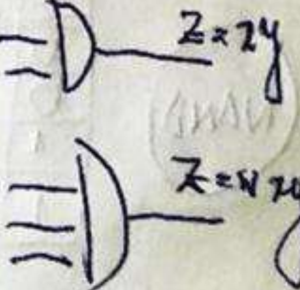
$=$ x and y

$(Z = x \wedge y)$

$Z = wxy$

* 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	

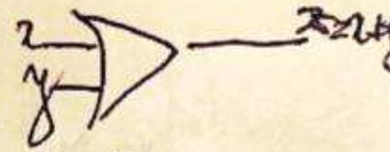
Expression

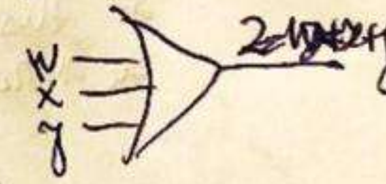
$Z = x + y = x$ OR y

$(Z = x \vee y)$

$Z = w + y + z$

* 1's dominance
* disjunction



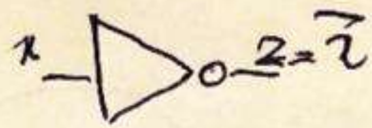


NOT

Input	Output
x	z
0	1
1	0

complement

$Z = \overline{x}$



NAND

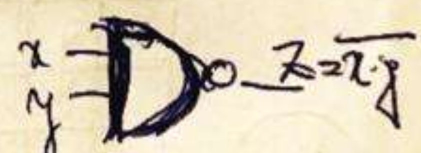
Inputs		Output
x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

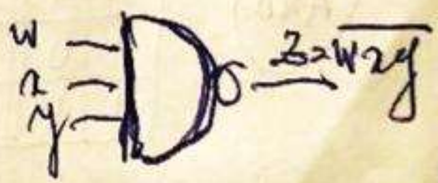
Inputs			Output
x	y	z	z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

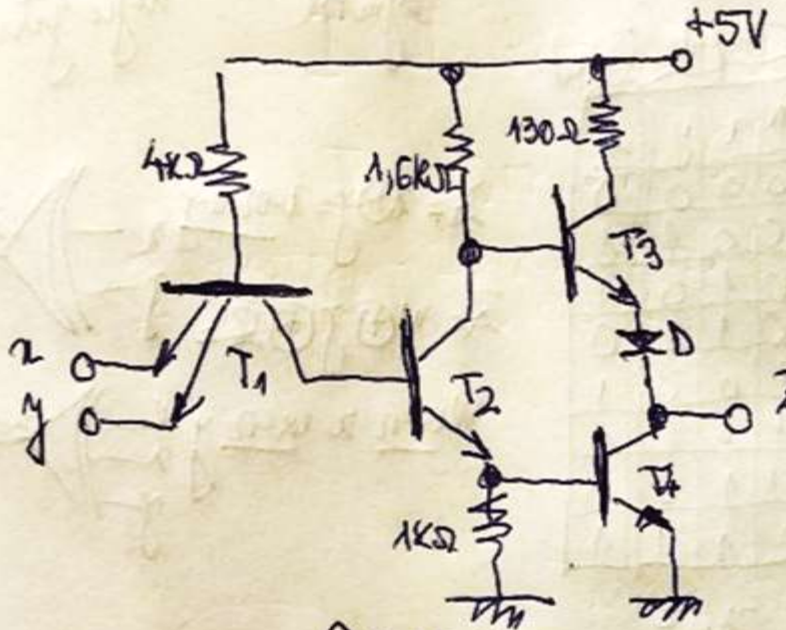
$Z = \overline{x \cdot y} = \overline{xy} =$

$= x$ nand y

$Z = \overline{wxy}$







Jack Kilby – Texas Instruments

Voltage

0 —— L (Low)

1 —— H (High)

Truth tables

NOR

Inputs / Output

x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

Inputs / Output

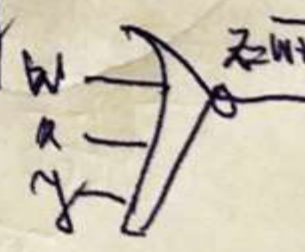
w	q	y	z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Symbol

Logic gate

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

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



Truth tables

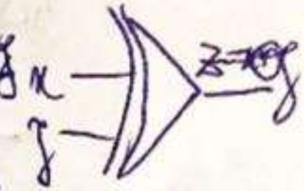
Symbol — Logic gates

Exclusive OR (EX-OR)

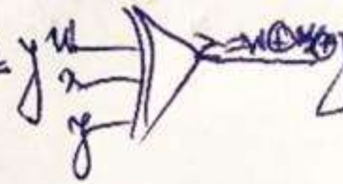
Inputs		Output
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

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



$Z = w \oplus y \oplus z =$ 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

$Z = \overline{w \oplus y \oplus z} =$ 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
$\bar{\bar{x}} = x$		Double negation law
$x \cdot x = x$	$x + x = x$	Idempotence laws
$x y = y x$	$x + y = y + x$	Commutativity laws
$x \cdot (yz) = (xy)z = (xz)y$	$x + (y+z) = (x+y) + z = (x+z) + y$	Associativity laws
$x \cdot (y+z) = xy + xz$	$x + yz = (x+y)(x+z)$	Distributivity laws

* 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 generalised to an arbitrary number of terms

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

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

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	$\overline{x \cdot y}$	$\overline{x}$	$\overline{y}$	$\overline{x}+\overline{y}$	$\overline{x+y}$	$\overline{x}$	$\overline{y}$	$\overline{x} \cdot \overline{y}$
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

$\equiv$ q.e.d

$\equiv$ 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

=

③ $F(A,B,C,D) = (B\bar{C}+\bar{A}D)(A\bar{B}+C\bar{D}) = \underbrace{A\bar{B}B\bar{C}}_{0} + \underbrace{A\bar{A}\bar{B}D}_{0} +$

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

(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

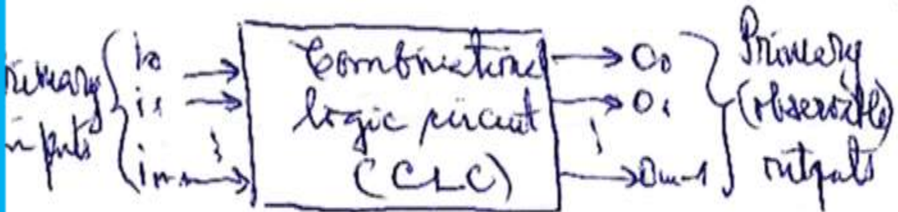
(5) $F(A,B,C,D) = \bar{A}B(\bar{D} + \bar{C}D) + B(A + \bar{A}CD) =$

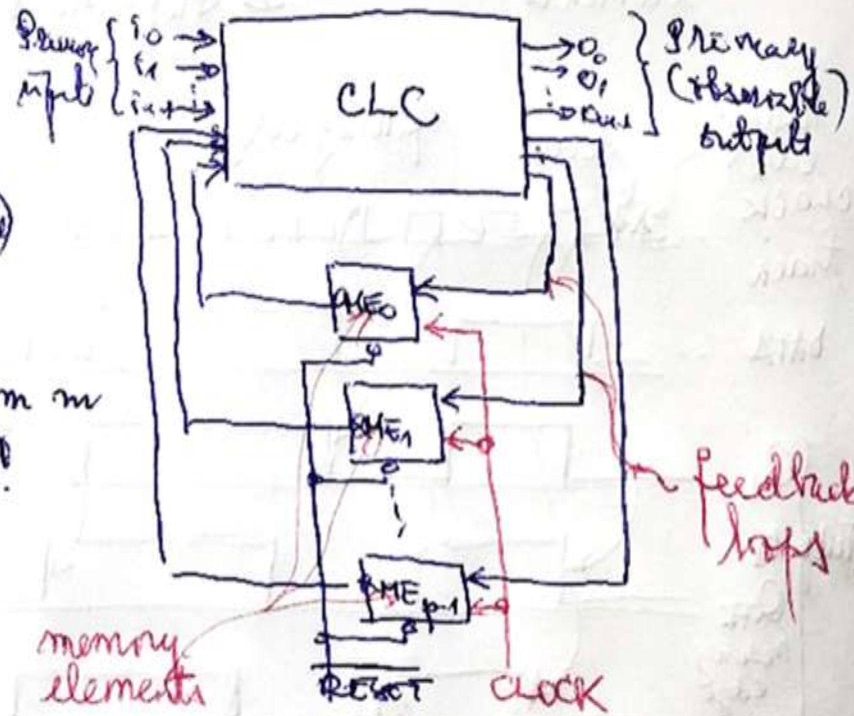
$= \bar{A}B(\bar{D}+\bar{C}) + B(A+CD) = B(\bar{A}\bar{D} + \bar{A}\bar{C} + A + CD) =$

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

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

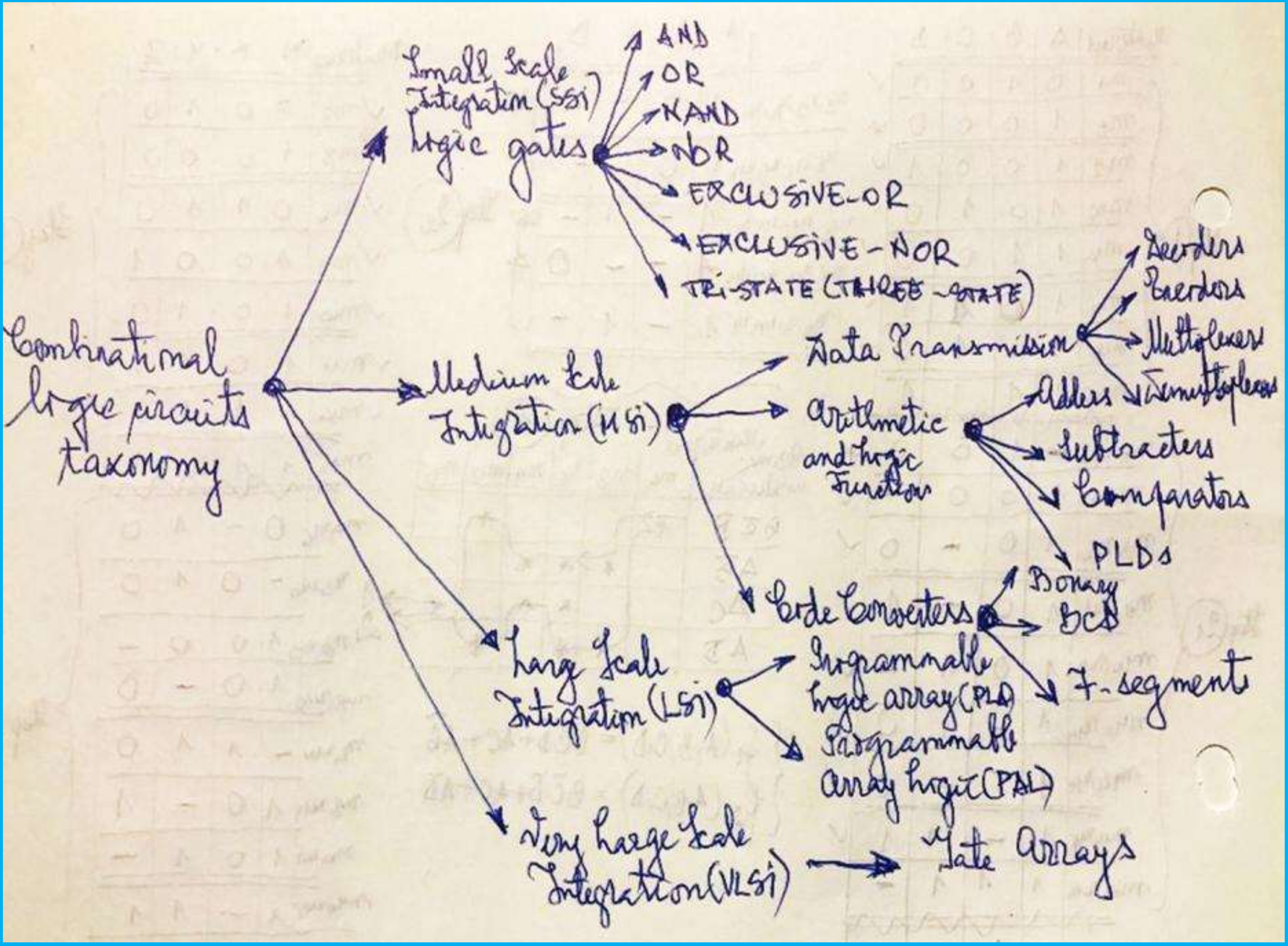
Combinational vs sequential logic circuits





* 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-segments
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)

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

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

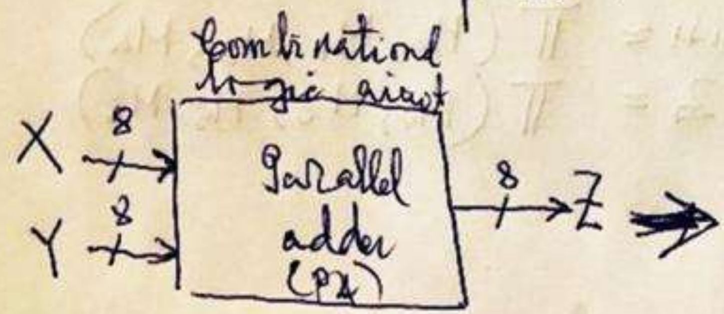
false!

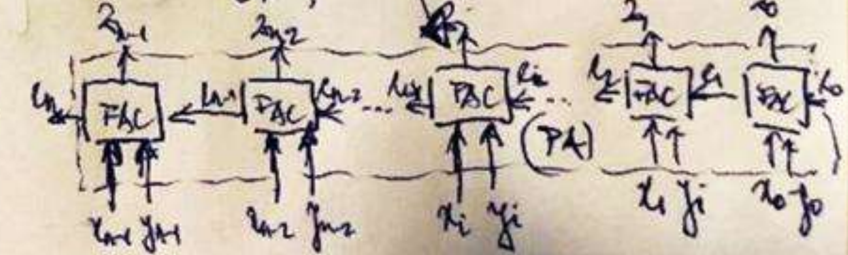
end-around carry $\rightarrow$ $+1$

$$\underbrace{00001111}_{+15_{10}}$$

OK!

ok!





Problem statement → Truth table constructed → logic equation(s) written → logic equation(s) implemented → 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

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

$x_i + y_i + c_i \rightarrow z_i$

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

arithmetic!

Problem statement → Truth table constructed → $\boxed{\text{logic equation(s) written}}$ → 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)
$$\begin{cases} c_{i+1} = \bar{x}_i y_i c_i + x_i \bar{y}_i c_i + x_i y_i \bar{c}_i + x_i y_i c_i \\ z_i = \bar{x}_i \bar{y}_i c_i + \bar{x}_i y_i \bar{c}_i + x_i \bar{y}_i \bar{c}_i + x_i y_i c_i \end{cases}$$

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

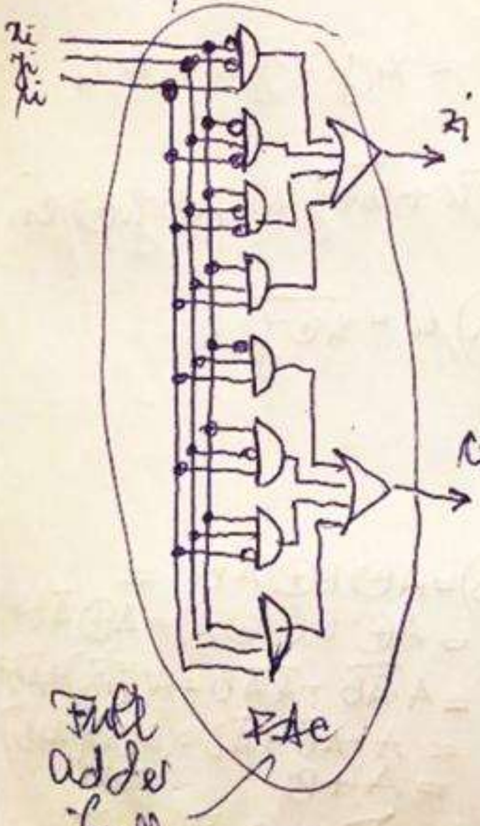
SOP (CNDF)
$$\begin{cases} c_{i+1} = \sum (m_3, m_5, m_6, m_7) \\ z_i = \sum (m_1, m_2, m_4, m_7) \end{cases}$$

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

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

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 postulates of Boole's algebra
- using graphical method of Karnaugh (Karnaugh maps)
- using tabular method of Quine - McCluskey

Ex 1

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

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

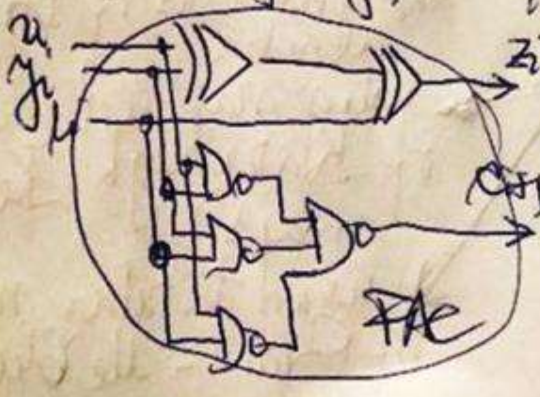
Ex 2

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

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

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

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



Ex. 3

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

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}

$$F(A,B,C,D) = \sum(m_5, m_7, m_{13}, m_{15}) = BD$$

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

$$F(A,B,C,D) = \sum(m_5, m_6, m_7, m_{13}, m_{14}, m_{15}) = BC + BD$$

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

BC

BD

$$F(A,B,C,D) = \sum(m_3, m_5, m_6, m_7, m_{13}, m_{14}, m_{15}) = \bar{A}CD + BC + BD$$

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

$\bar{A}CD$

BC

BD

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

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

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

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

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

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

BCD

b_{i+1}

$x_i \backslash y_i b_i$	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

$x_i b_i$ $y_i b_i$ $x_i y_i$

prime implicants

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

Z_i

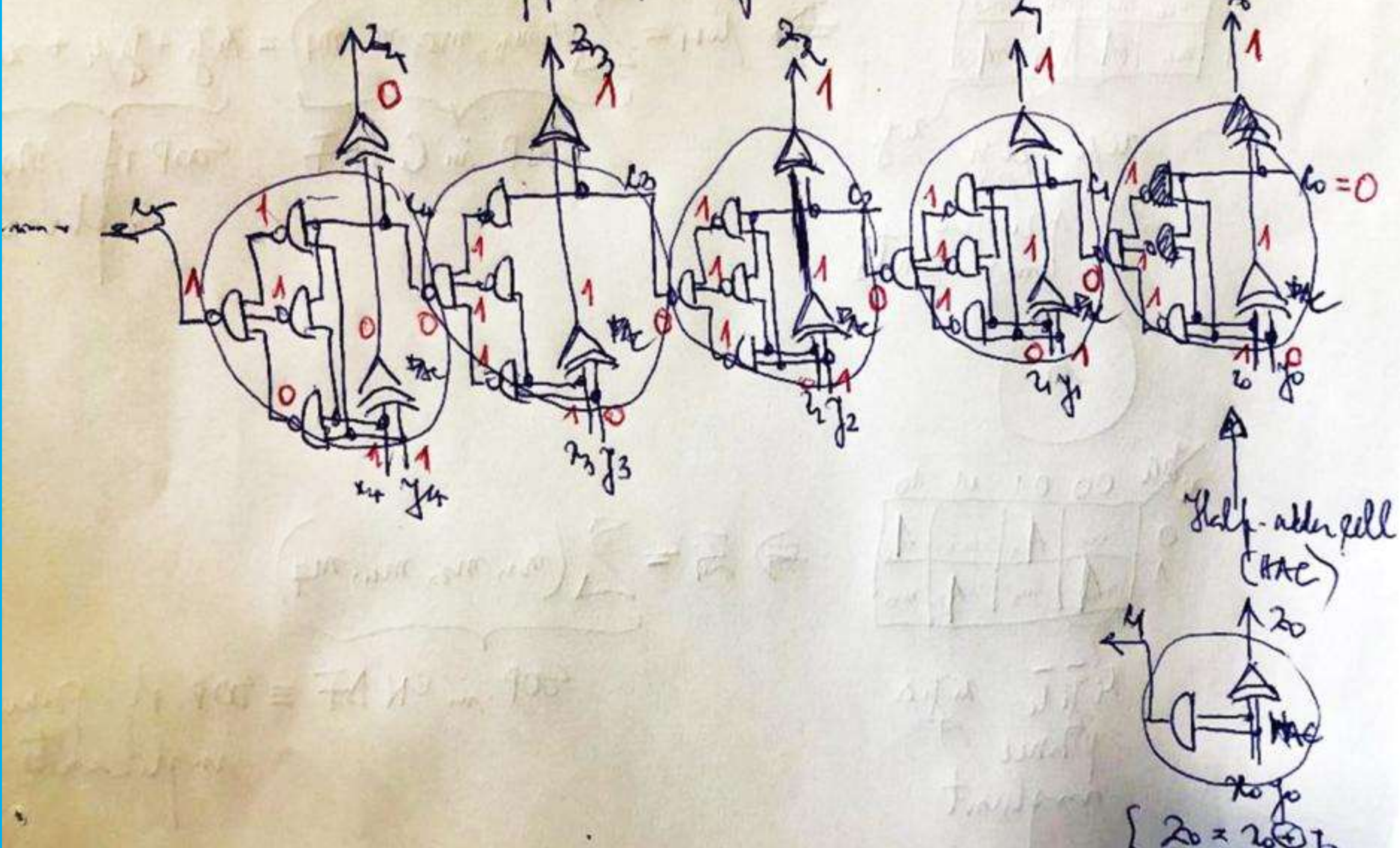
$x_i \backslash y_i b_i$	00	01	11	10
0	m_0	1 m_1	m_3	1 m_2
1	1 m_4	m_5	1 m_7	m_6

$x_i \bar{y}_i \bar{b}_i$ $x_i y_i b_i$

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				1
10			1	1

Y

a_1a_0 \ b_1b_0	00	01	11	10
00				
01			1	1
11		1		1
10		1	1	

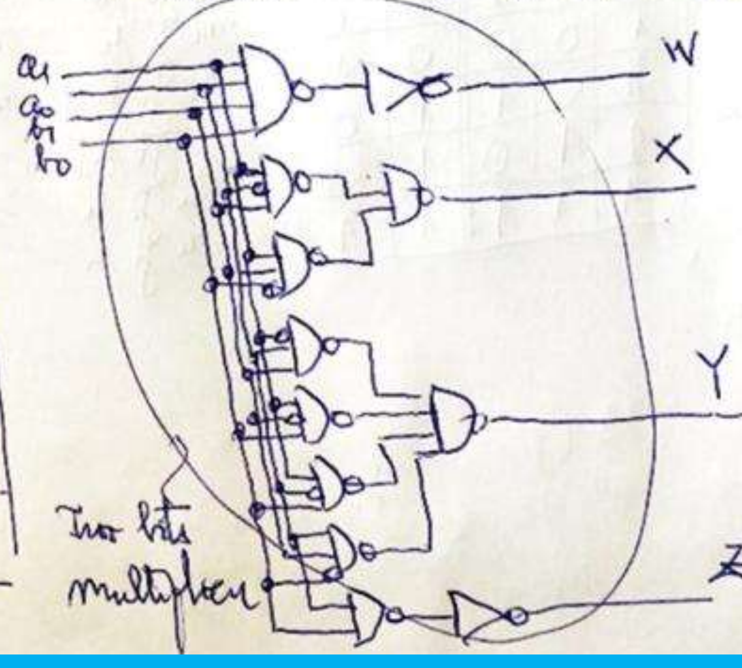
Z

a_1a_0 \ b_1b_0	00	01	11	10
00				
01		1	1	
11		1	1	
10				

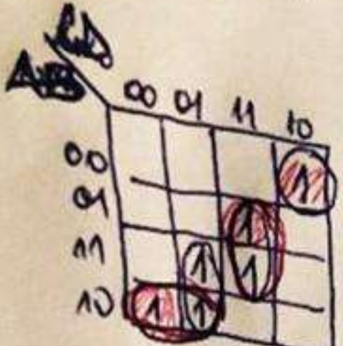
$X = a_1\bar{a}_0 b_1 + a_1 b_1 \bar{b}_0$

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

$Z = a_0 b_0$



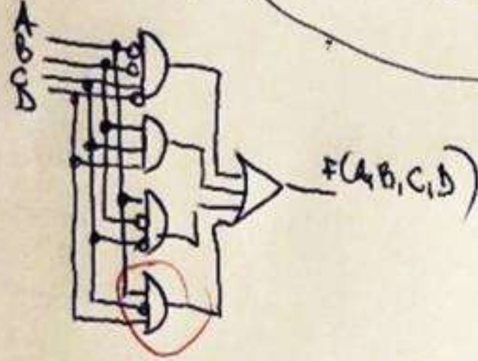
7ex $F(A,B,C,D) = \sum(m_2, m_7, m_8, m_9, m_{13}, m_{15}) = \prod(M_0, M_1, M_3, M_4, M_5, M_6, M_{10}, M_{11}, M_{12}, M_{14})$

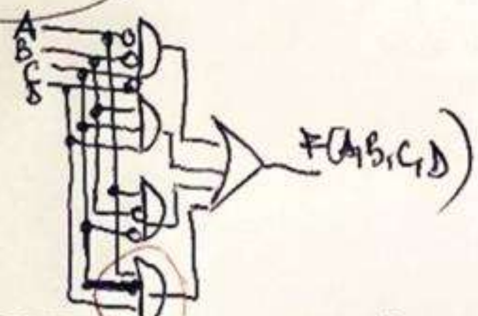


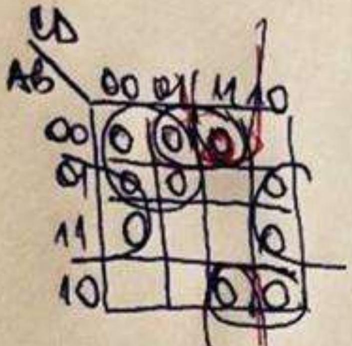
$F_1(A,B,C,D) = \bar{A}\bar{B}C\bar{D} + BCD + A\bar{B}\bar{C} + A\bar{C}D$

essential prime implicants

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







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

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

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

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

$ABCD$ $ABCD$ $AB\bar{C}D$

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

	Minterms	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}
$Y\bar{Z}$	×	×			×		×	
$W\bar{X}$			×	×	×	×		
WY					×	×	×	×

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

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

Step 1

Minterms	A	B	C	D	Index
✓ 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

Pair of minterms	A	B	C	D	Index
* m_4, m_{12}	–	1	0	0	
✓ m_8, m_9	1	0	0	–	1
✓ 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	
✓ m_{11}, m_{15}	1	–	1	1	3
✓ m_{14}, m_{15}	1	1	1	–	

Step 2_2

Pair of minterms	A	B	C	D	Index
* 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~~	~~–~~	

Step 3

Prime implicants \ Minterms	m_4	m_8	m_{10}	m_{11}	m_{12}	m_{15}
$B\bar{C}\bar{D}$ 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: Utilizând, 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 din cod 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 utilizând 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 utilizând de metodele Karnaugh și Quine-McCluskey. Având la dispoziție porți NAND cu 2 și 3 intrări, să se realizeze implementarea soluțiilor minimizate.

BCD Adder

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

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

$Z = 647_{10}$

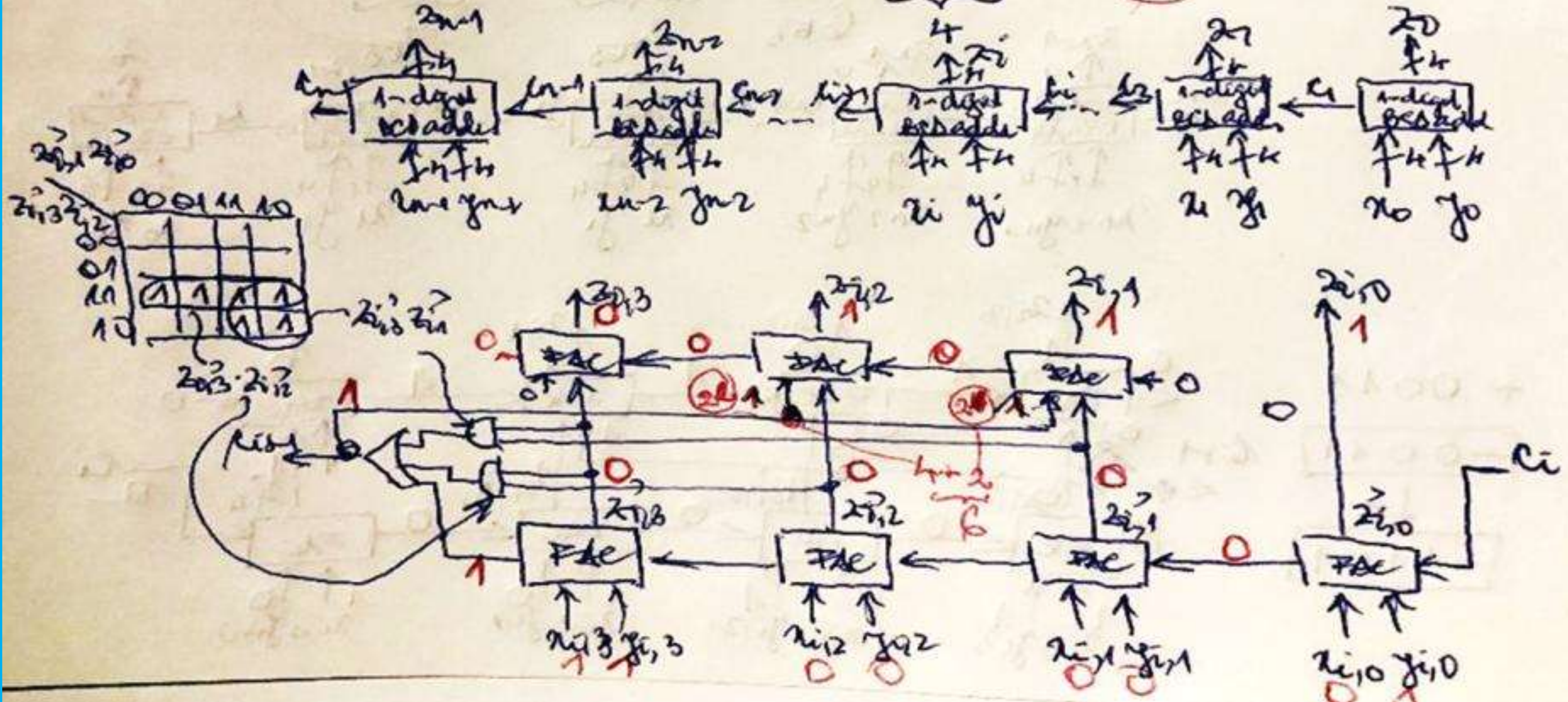
$0101\ \ 1101\ \ (1)\ 0001$

$+\ 0110$

$0111 \Rightarrow 7$

$1110 + 0110 = (1)\ 0100 \Rightarrow 4$

$0101 + 1 = 0110 \Rightarrow 6$



Excess-Three Adder

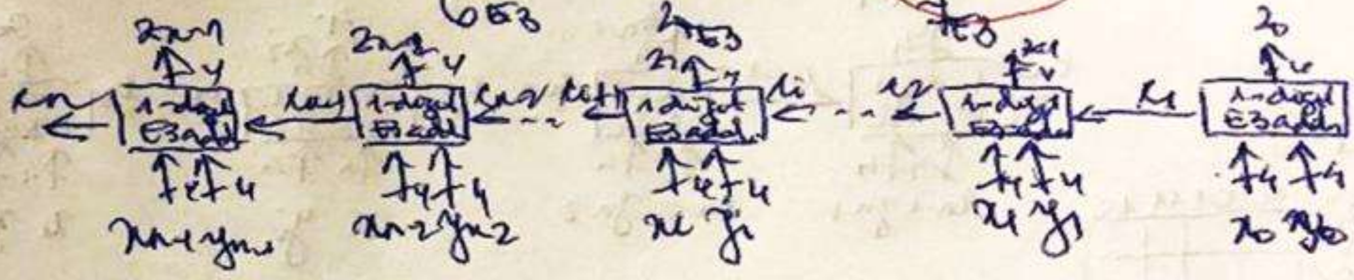
$X = 278_{10} = 0101\ 1010\ 1011_{E3}$ +

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

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

$\boxed{0011}$ $\boxed{0011}$ $\boxed{0011}$

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

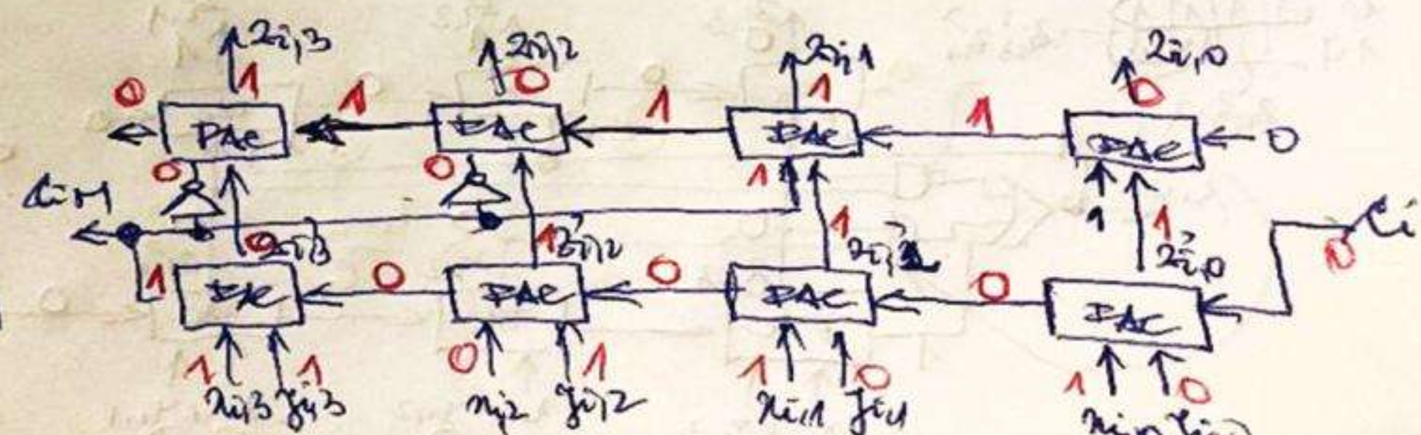


+ 0011

$\boxed{-0011}$

↓=

$\boxed{+1101}$



Subtracters

② Parallel subtracter

borrow

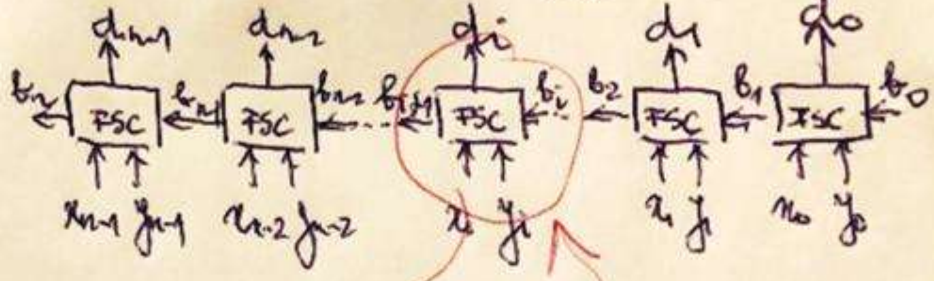
$X = 17_{10} = 10001_2$

$b_{i+1} \leftarrow b_i$

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

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

FSC - Full Subtracter Cell



Truth table

x_i	y_i	b_i	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

$x_i - (y_i + b_i) \Rightarrow$

Logic equations

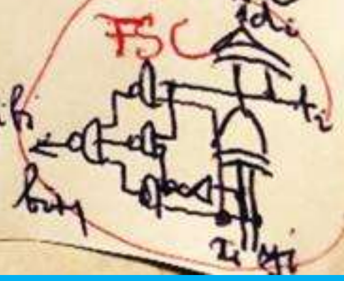
$$\begin{cases} d_i = \sum(m_1, m_2, m_4, m_7) = \bar{x}_i\bar{y}_i b_i + \bar{x}_i y_i \bar{b}_i + x_i \bar{y}_i \bar{b}_i + x_i y_i b_i = \\ \quad = (\bar{x}_i\bar{y}_i + x_i y_i) b_i + (\bar{x}_i y_i + x_i \bar{y}_i)\bar{b}_i = x_i \oplus y_i \oplus b_i \\ b_{i+1} = \sum(m_1, m_2, m_3, m_7) \end{cases}$$

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



Subtracters

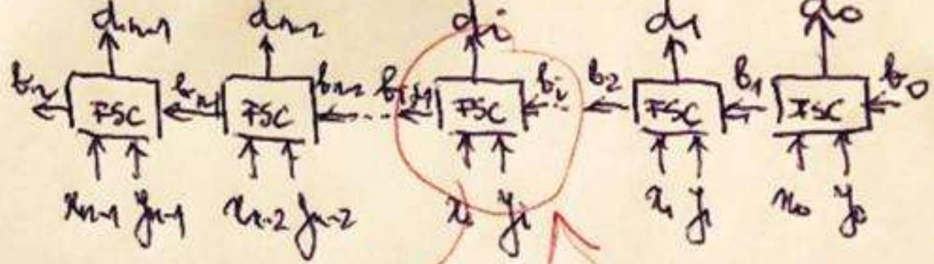
(2) Parallel subtracter

FSC - Full Subtracter Cell

$$X = 17_{10} = 10001_2 -$$
$$Y = 13_{10} = 01101_2$$
$$D = 4_{10} \qquad \underbrace{00100}_{4_{10}}$$

borrow

$b_{i+1} \leftarrow b_i$



Truth table

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

$x_i - (y_i + b_i) \Rightarrow$

Logic equations

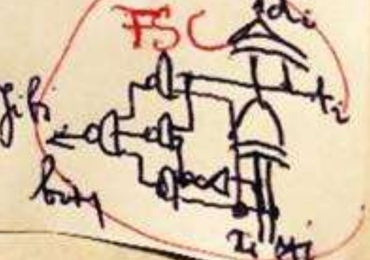
$$\begin{cases} d_i = \sum(m_1, m_2, m_4, m_7) = \bar{x}_i\bar{y}_i b_i + \bar{x}_i y_i \bar{b}_i + x_i\bar{y}_i\bar{b}_i + x_i y_i b_i = \\ \quad = (\bar{x}_i\bar{y}_i + x_i y_i)b_i + (\bar{x}_i y_i + x_i\bar{y}_i)\bar{b}_i = x_i \oplus y_i \oplus b_i \\ b_{i+1} = \sum(m_1, m_2, m_3, m_7) \end{cases}$$

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



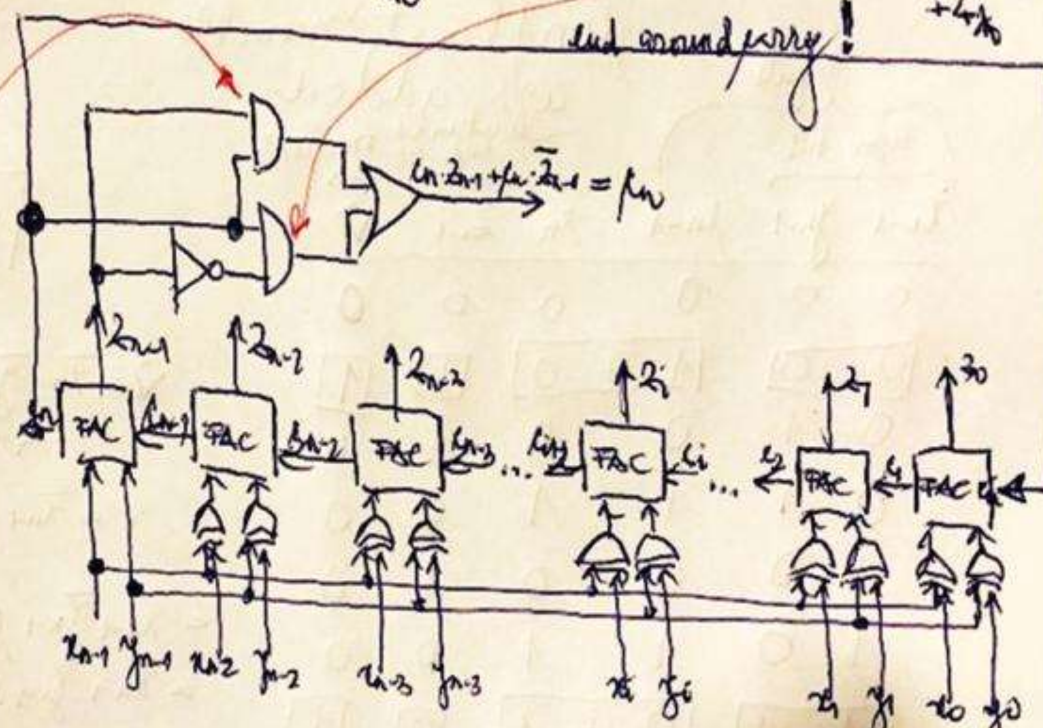
One's complement adder

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

$X = +38_{10} = 00100110_{2\text{-}C1}$
$Y = -42_{10} = 11010101_{2\text{-}C1}$
$Z = -4_{10} = 0\ 11111011_{2\text{-}C1}$
$\downarrow C1 \to SM$
$\underbrace{10000100}_{-4_{10}}$

$X = -38_{10} = 11011001_{2\text{-}C1}$
$Y = +42_{10} = 00101010_{2\text{-}C1}$
$Z = +4_{10} = \boxed{1}\ \boxed{0}0000011$
end around carry 1
$\underbrace{00000100}_{+4_{10}}$

$X = -38_{10} = 11011001_{2\text{-}C1}$
$Y = -42_{10} = 11010101_{2\text{-}C1}$
$Z = -80_{10} = \boxed{1}\boxed{1}0101110$
$\rightarrow 1$
10101111
$\downarrow C1 \to SM$
$\underbrace{11010000}_{-80_{10}}$



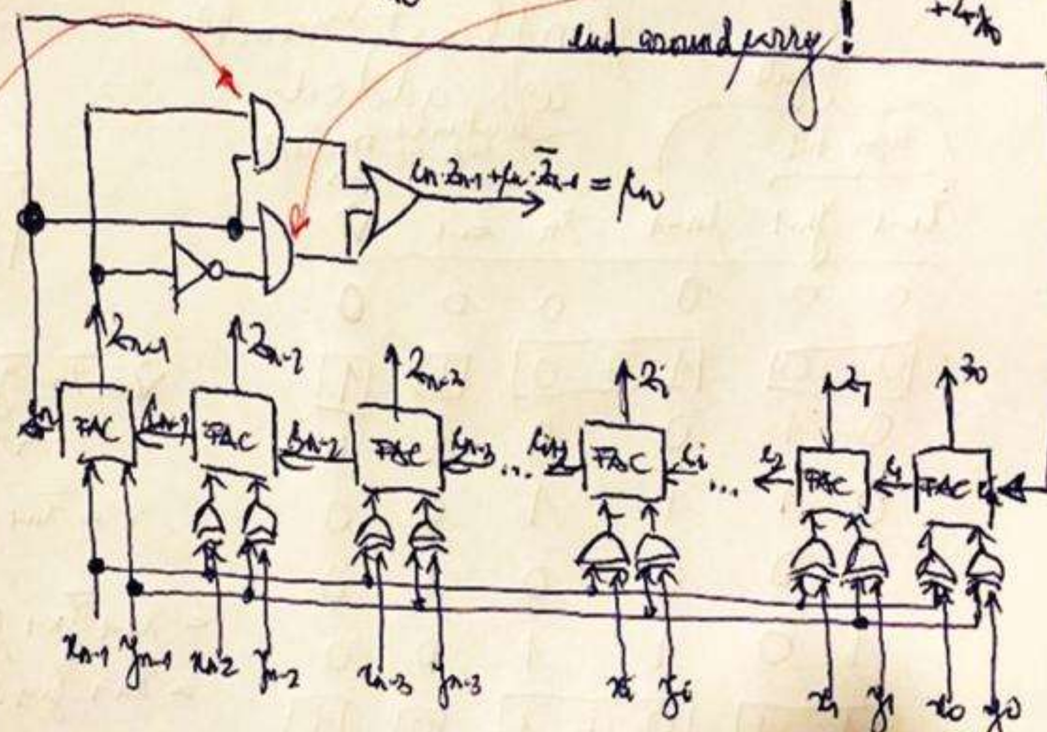
One's complement adder

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

$$\begin{aligned}
X &= +38_{10} = 00100110_{2\text{-}C1}\\
Y &= -42_{10} = 11010101_{2\text{-}C1}\\
\hline
Z &= -4_{10}\ 0\ 11111011_{2\text{-}C1}\\
&\quad \downarrow C1 \to SM\\
&\quad \underbrace{10000100}_{-4_{10}}
\end{aligned}$$

$$\begin{aligned}
X &= -38_{10} = 11011001_{2\text{-}C1}\\
Y &= +42_{10} = 00101010_{2\text{-}C1}\\
\hline
Z &= +4_{10}\ \boxed{1}\ \boxed{0}\ 0000011\\
&\quad \text{end around carry } 1\\
\hline
&\quad \underbrace{00000100}_{+4_{10}}
\end{aligned}$$

$$\begin{aligned}
X &= -38_{10} = 11011001_{2\text{-}C1}\\
Y &= -42_{10} = 11010101_{2\text{-}C1}\\
\hline
Z &= -80_{10}\ \boxed{1}\ \boxed{1}\ 0101110\\
&\quad \longrightarrow 1\\
\hline
&\quad 10101111\\
&\quad \downarrow C1 \to SM\\
&\quad \underbrace{11010000}_{-80_{10}}
\end{aligned}$$



Addition overflow detection

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

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

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

inputs: sign bits x_{n-1}	y_{n-1}	c_{n-1}	outputs: c_n	sign bit z_{n-1}	overflow V
0	0	0	0	0	0
0	0	1 ⊕	0	1	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
1	1	0 ⊕	1	0	1
1	1	1	1	1	0

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

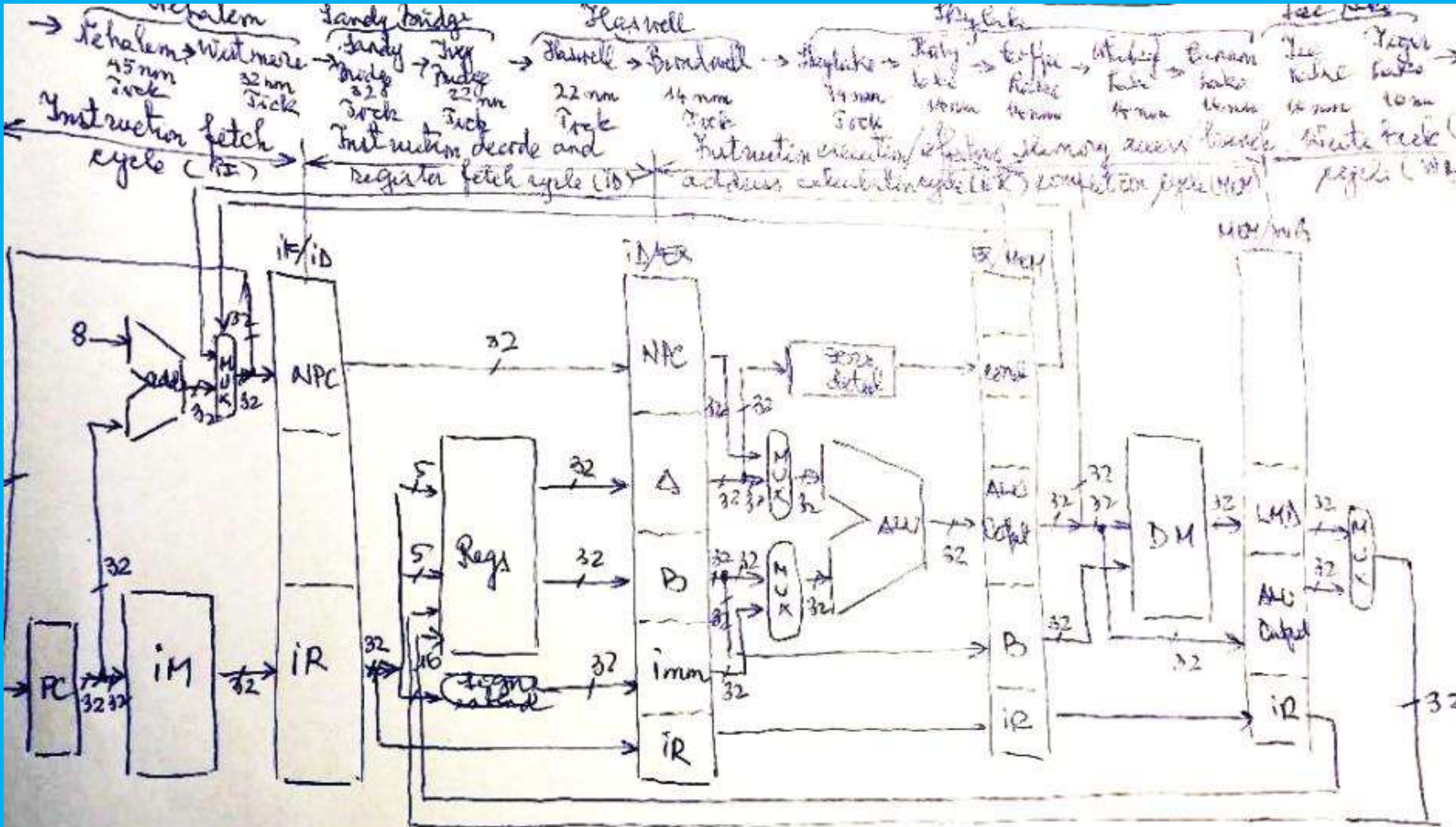
$$A+B = A \oplus B \oplus AB$$

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

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

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

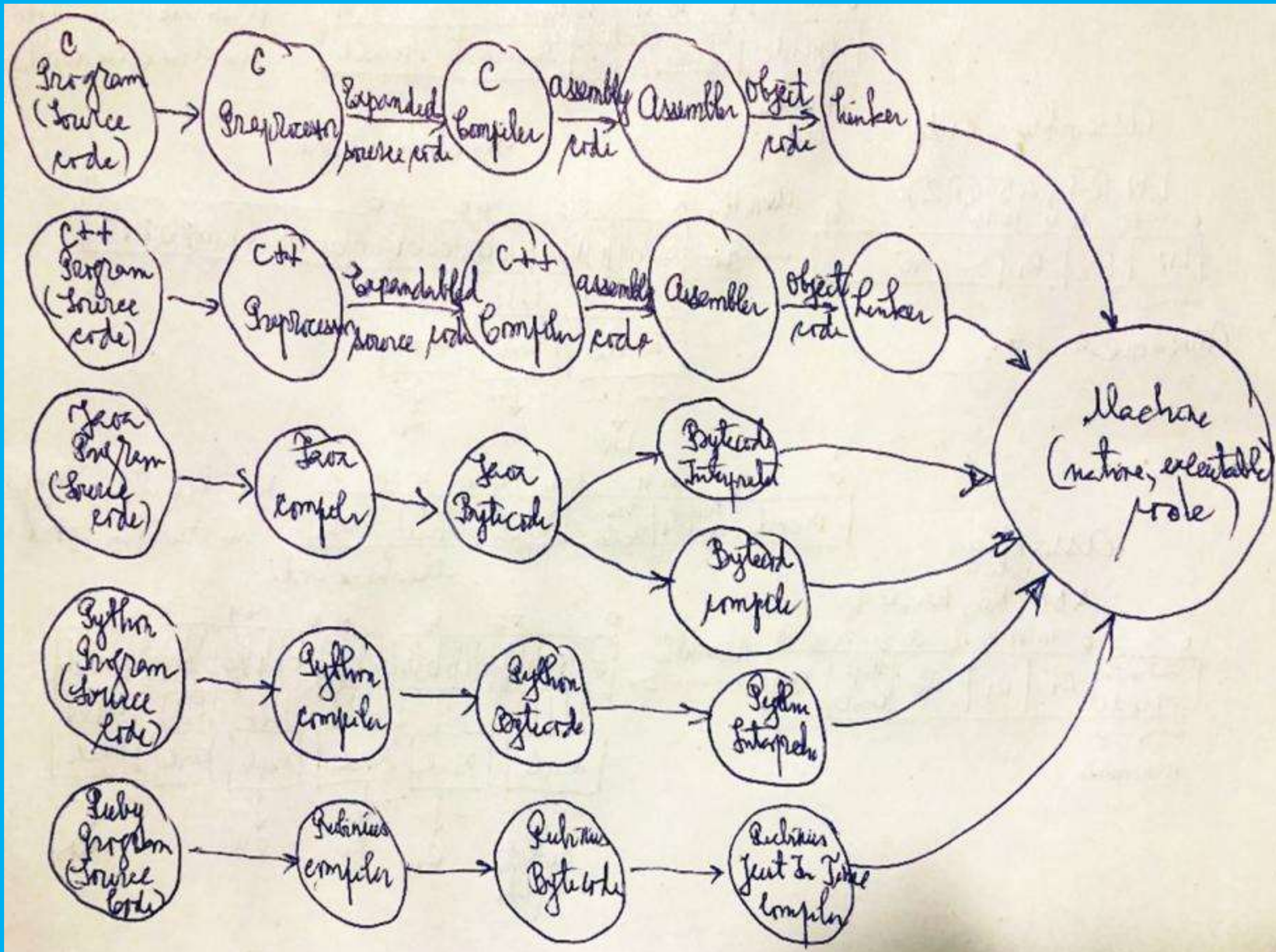
$$= \underbrace{x_{n-1}y_{n-1} \oplus x_{n-1}c_{n-1} \oplus y_{n-1}c_{n-1}}_{c_n} \oplus c_{n-1} = c_n \oplus c_{n-1}$$



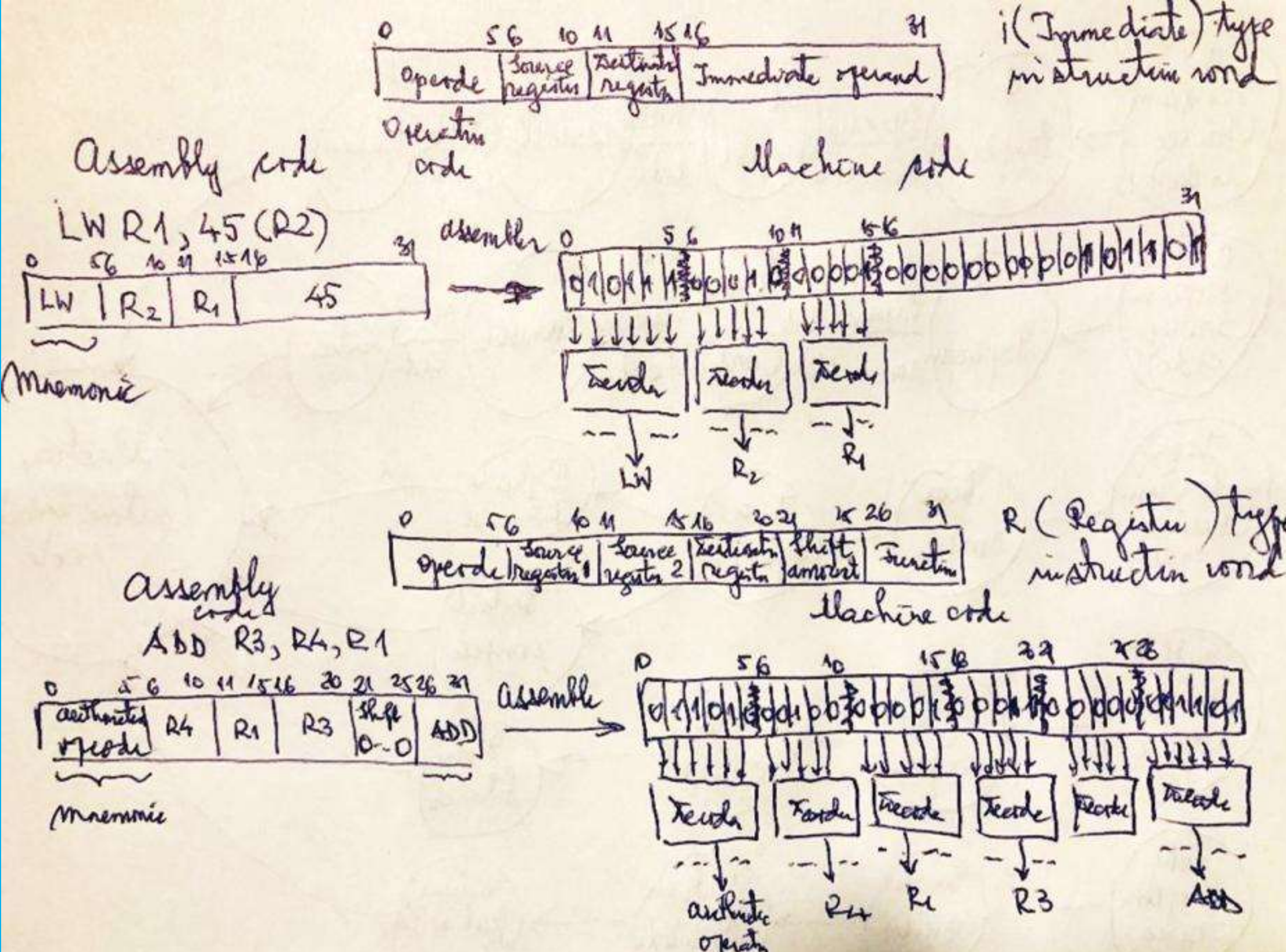
PC - Program counter
iM - Instruction memory
iR - Instruction register
NPC - Next program counter

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

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



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
Python Program (Source code)
Python compiler
Python Bytecode
Python Interpreter
Ruby Program (Source code)
Rubinius compiler
Rubinius Bytecode
Rubinius Just In Time compiler
Machine (native, executable code)

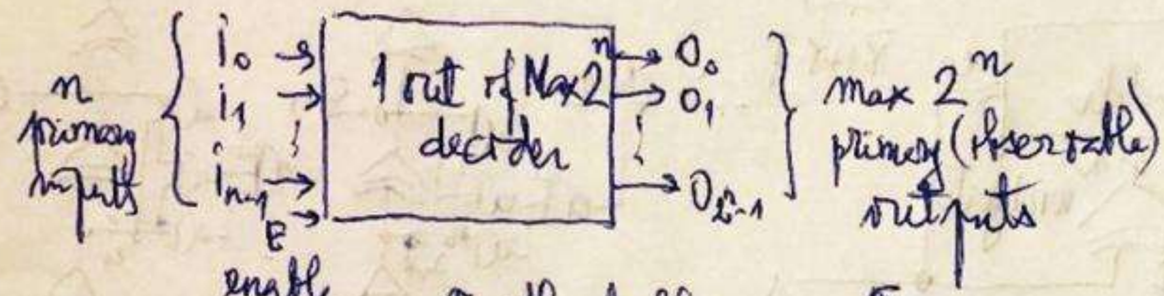


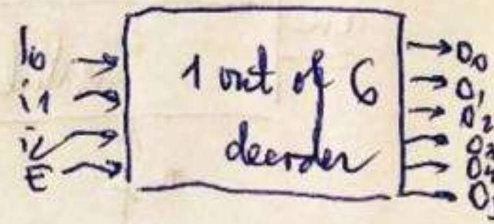
i(Immediate) type instruction word
0 5 6 10 11 15 16 31
Opcode
Source register
Destination register
Immediate operand
Operation code
Assembly code
Machine code
LW R1, 45 (R2)
assembler
LW
R2
R1
45
Mnemonic
Decoder
R (Register) type instruction word
Opcode
Source register 1
Source register 2
Destination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

Decoders

* Where used?

Block diagram



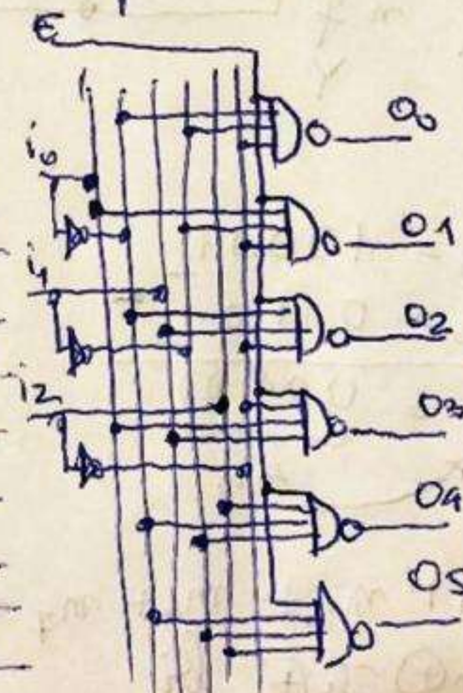


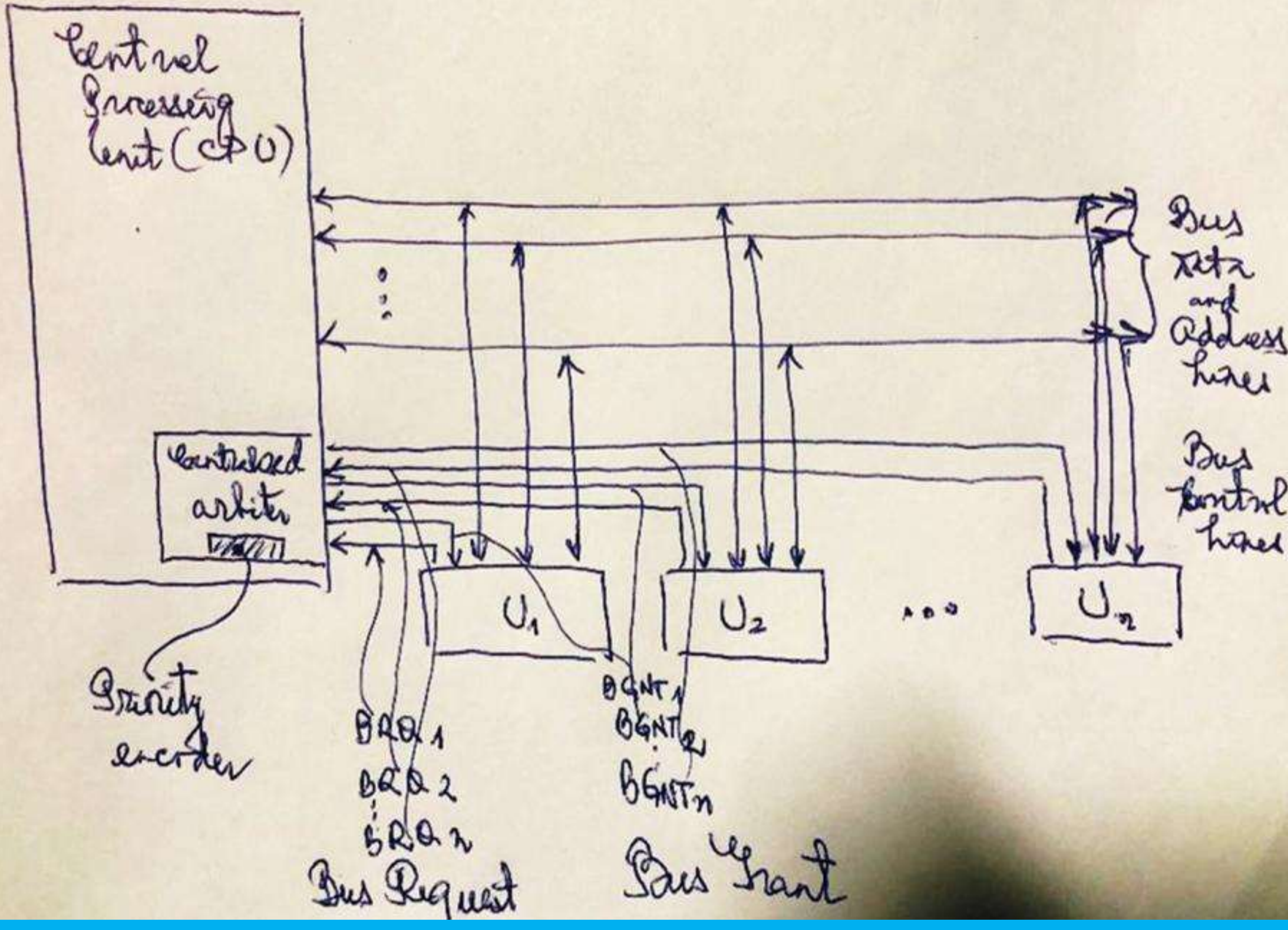
Truth table

enable input — primary inputs — primary outputs

don't care

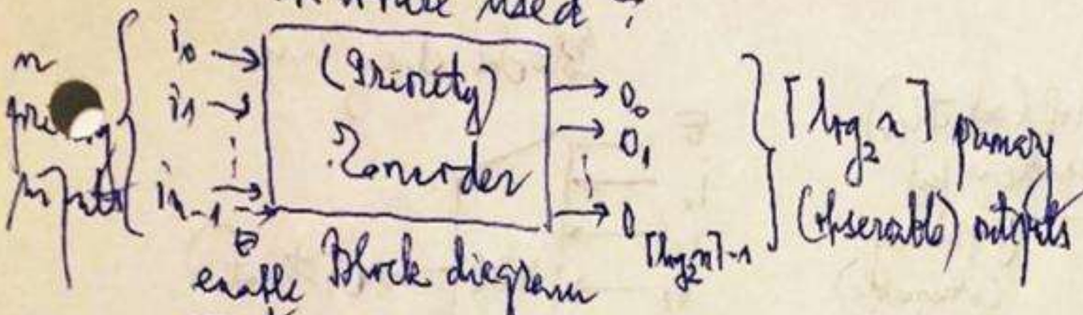
E	i_2	i_1	i_0	O_5	O_4	O_3	O_2	O_1	O_0
0	x	x	x	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0
1	0	0	1	1	1	1	1	0	1
1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	1	0	1	1	1
1	1	0	0	1	0	1	1	1	1
1	1	0	1	0	1	1	1	1	1




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

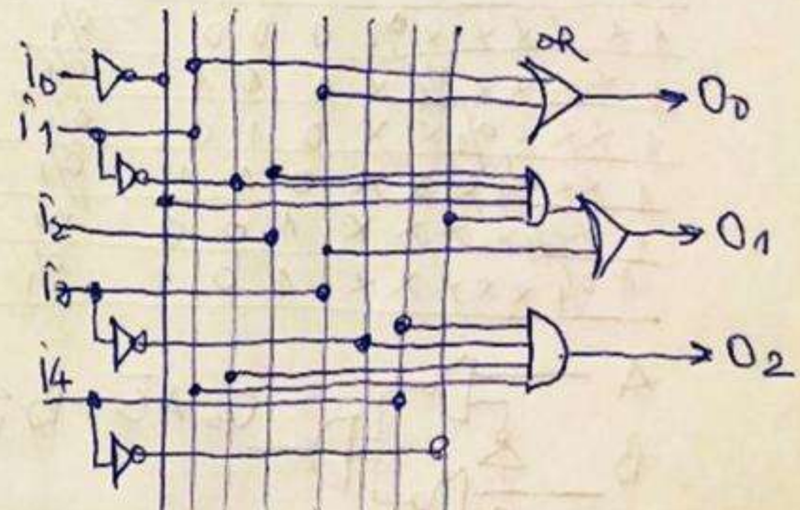
* where used ?



Block diagram

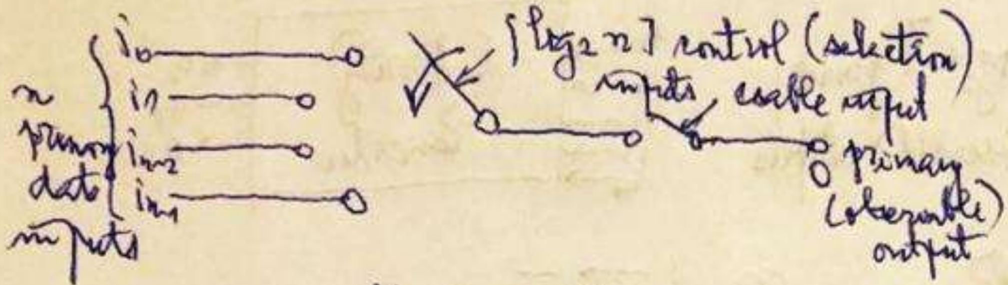
	primary inputs					primary (observable) outputs			
	i_4	i_3	i_2	i_1	i_0	O_2	O_1	O_0	
Priority:	4	1	5	2	3	2^2	2^1	2^0	weight
	x	1	x	x	x	0	1	1	
	x	0	x	1	x	0	0	1	
	x	0	x	0	1	0	0	0	
	1	0	x	0	0	1	0	0	
	0	0	1	0	0	0	1	0	

$$
\begin{cases}
O_2 = i_4 \cdot \overline{i_3} \cdot \overline{i_1} \cdot \overline{i_0} \\
O_1 = i_3 + \overline{i_4} \cdot \overline{i_3} \cdot i_2 \cdot \overline{i_1} \cdot \overline{i_0} = i_3 + \overline{i_4} \cdot i_2 \overline{i_1} \cdot \overline{i_0} \\
O_0 = i_3 + \overline{i_3} \cdot i_1 = i_3 + i_1
\end{cases}
$$

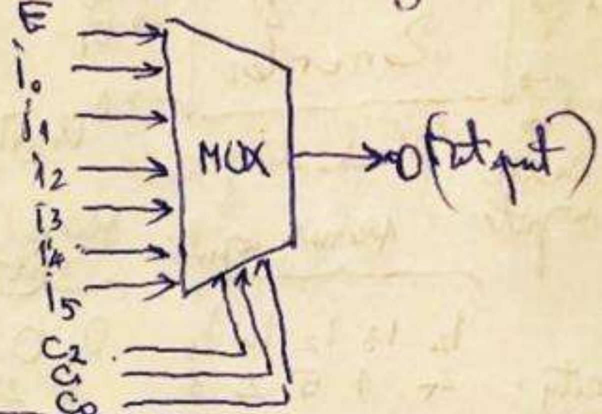


Multiplexers

* where used?



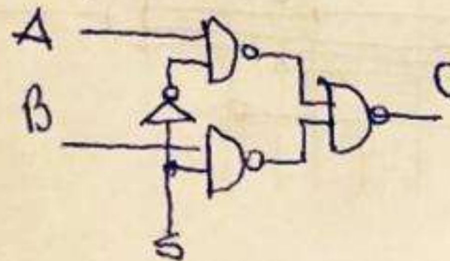
Block diagram



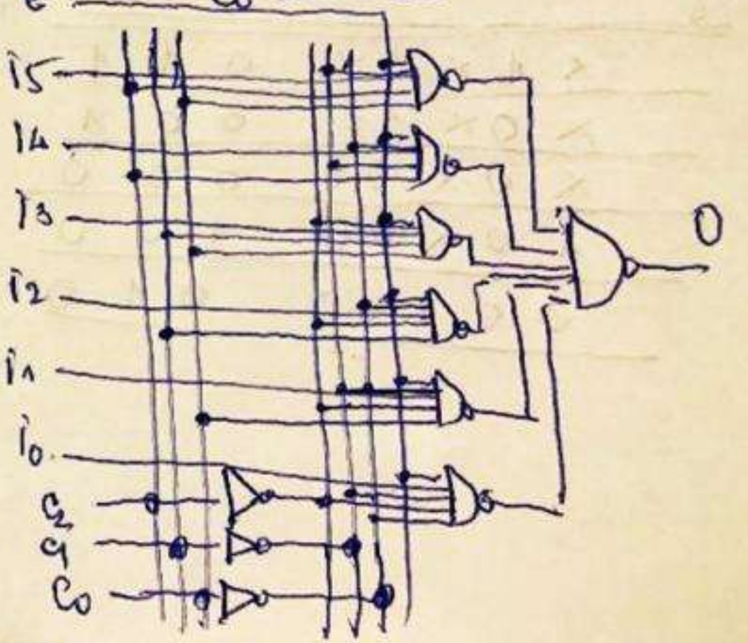
Truth table

don't care

inputs							control (selection)			output
E	i_5	i_4	i_3	i_2	i_1	i_0	C_2	C_1	C_0	O
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

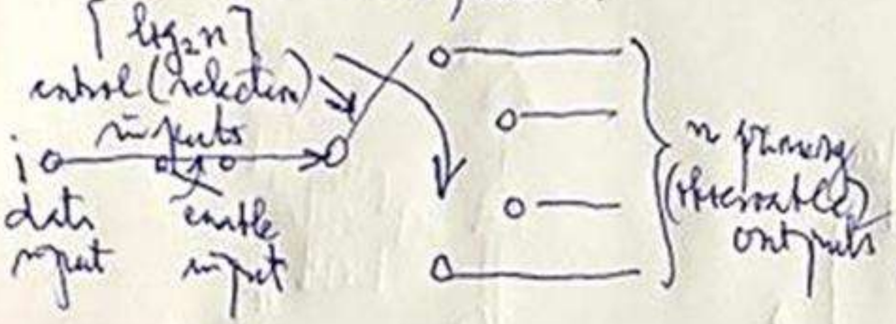


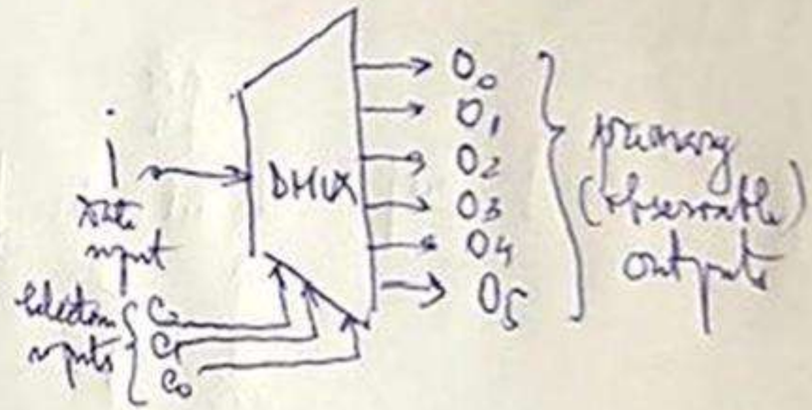
$$O = A\bar{S} + B \cdot S$$



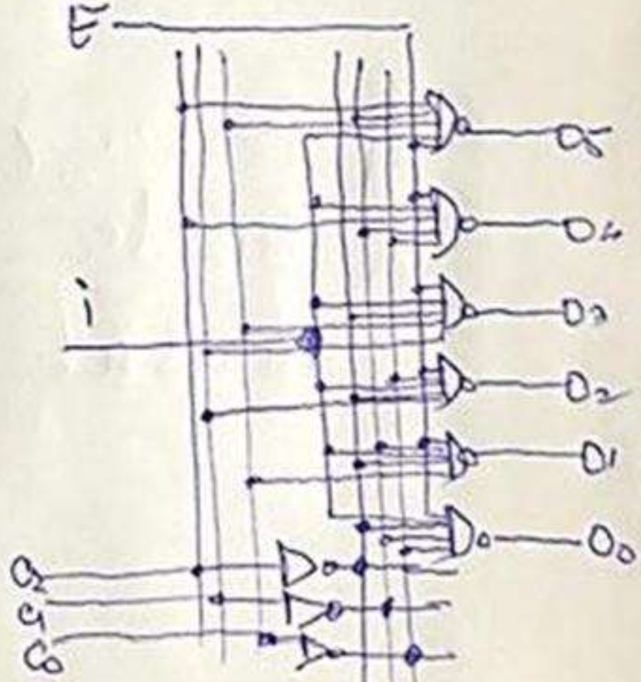
Demultiplexers

* 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

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 > 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	

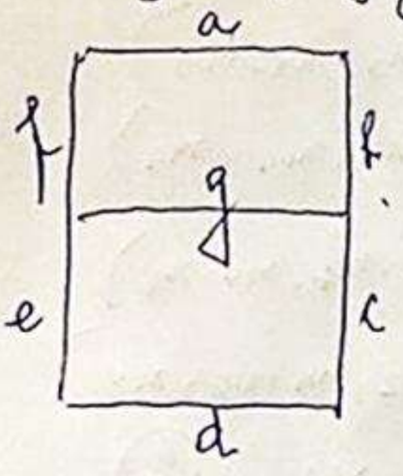
$(A>B) = a_1\overline{b_1} + a_1a_0\overline{b_0} + \overline{a_0}\,\overline{b_1}\,\overline{b_0}$ $\quad (A<B) = \overline{a_1}b_1 + \overline{a_1}\,\overline{a_0}b_0 + \overline{a_0}b_1b_0$

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



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

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

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

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

$e = 2+6+8+0$

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

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

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)

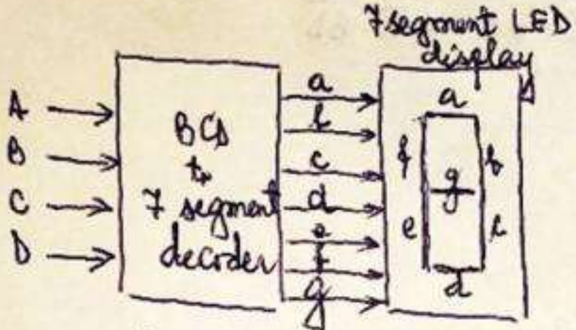
AB \ CD	00	01	11	10
00	1		1	1
01		1	1	1
11				
10	1	1		

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

(g)

AB \ CD	00	01	11	10
00			1	1
01	1	1	Φ	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						
2^3	2^2	2^1	2^0							
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
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

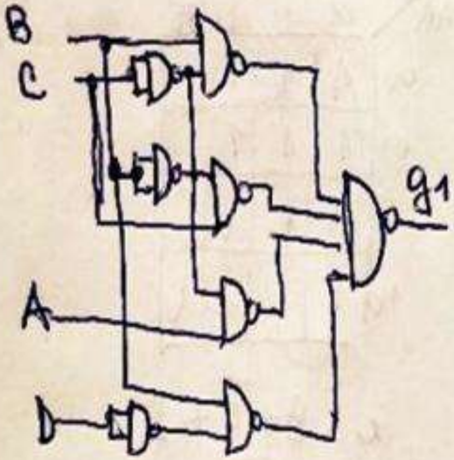
AB \ CD	00	01	11	10
00			1	1
01	1	1		1
11	d	d	d	d
10	1	1	d	d

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

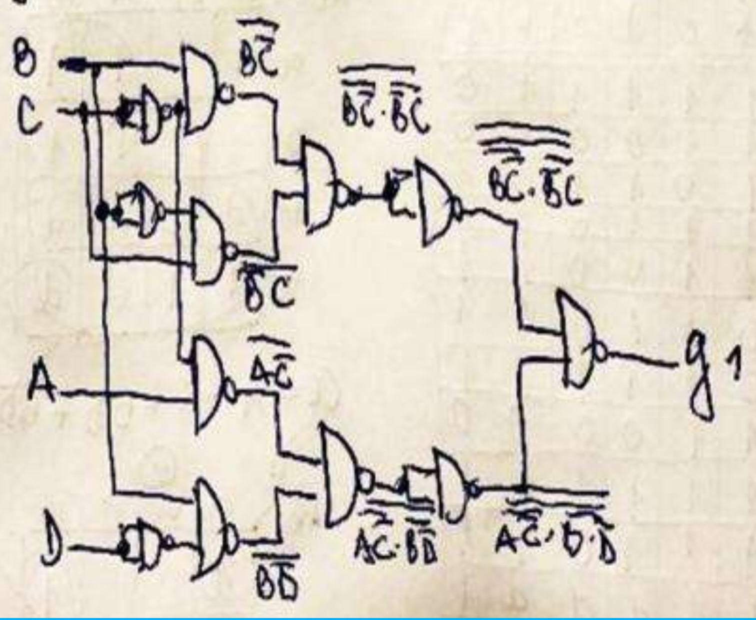
$$g_2 = B\bar{C} + \bar{B}C + A\bar{C} + C\bar{D}$$

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

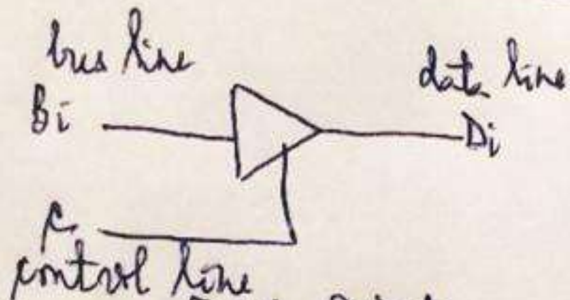
$$g_4 = B\bar{C} + \bar{B}C + A\bar{B} + C\bar{D}$$

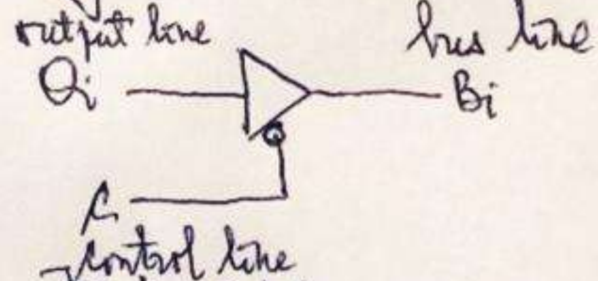


$$g_1 = \overline{\overline{B\bar{C} + \bar{B}C + A\bar{C} + B\bar{D}}} = \overline{\overline{B\bar{C} + \bar{B}C} \cdot \overline{A\bar{C} + B\bar{D}}} = \overline{\overline{\overline{\overline{B\bar{C}} \cdot \overline{\bar{B}C}}} \cdot \overline{\overline{\overline{A\bar{C}} \cdot \overline{B\bar{D}}}}}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



Tristate logic gates

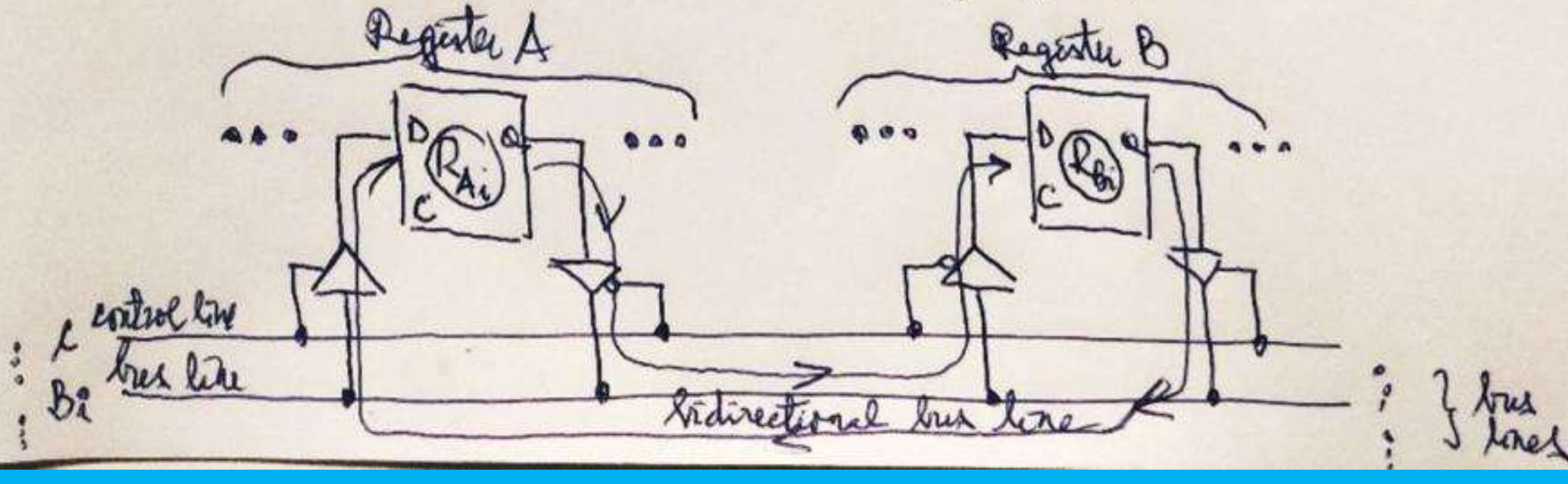




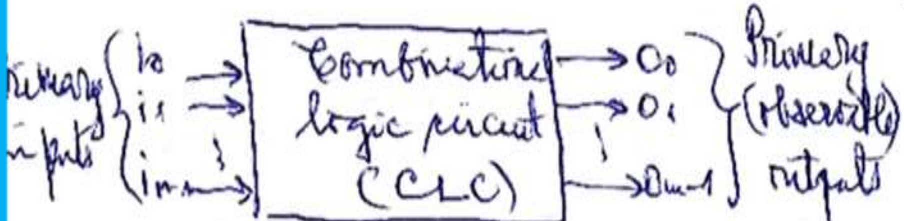
Inputs		Output
Bi	c	Di
0	0	Hi
0	1	0
1	0	Hi
1	1	1

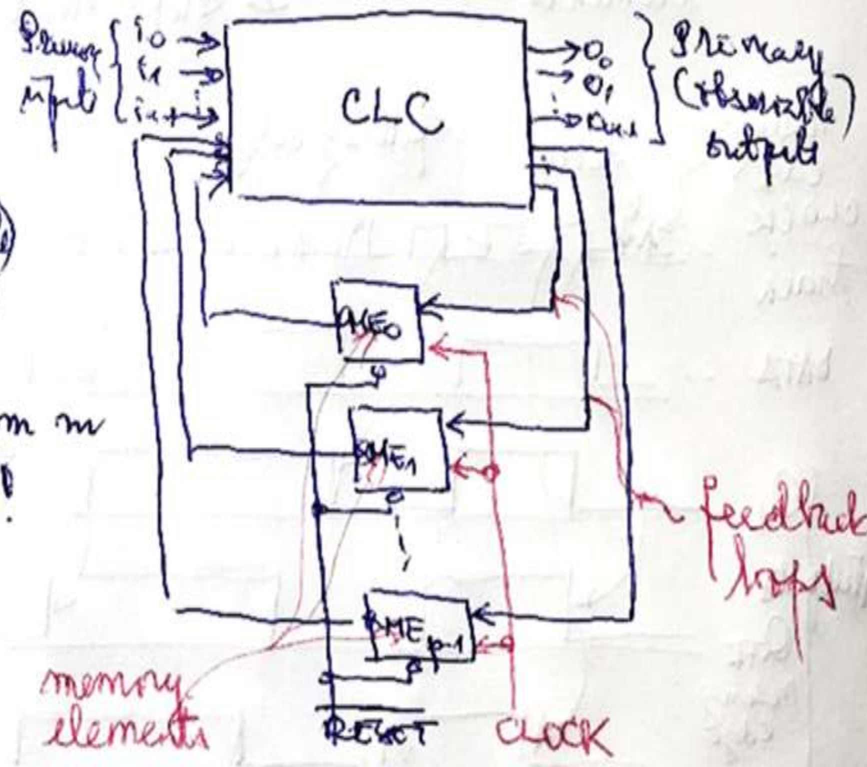
Inputs		Output
Qi	c	Bi
0	0	0
0	1	Hi
1	0	1
1	1	Hi

Hi - High Impedance



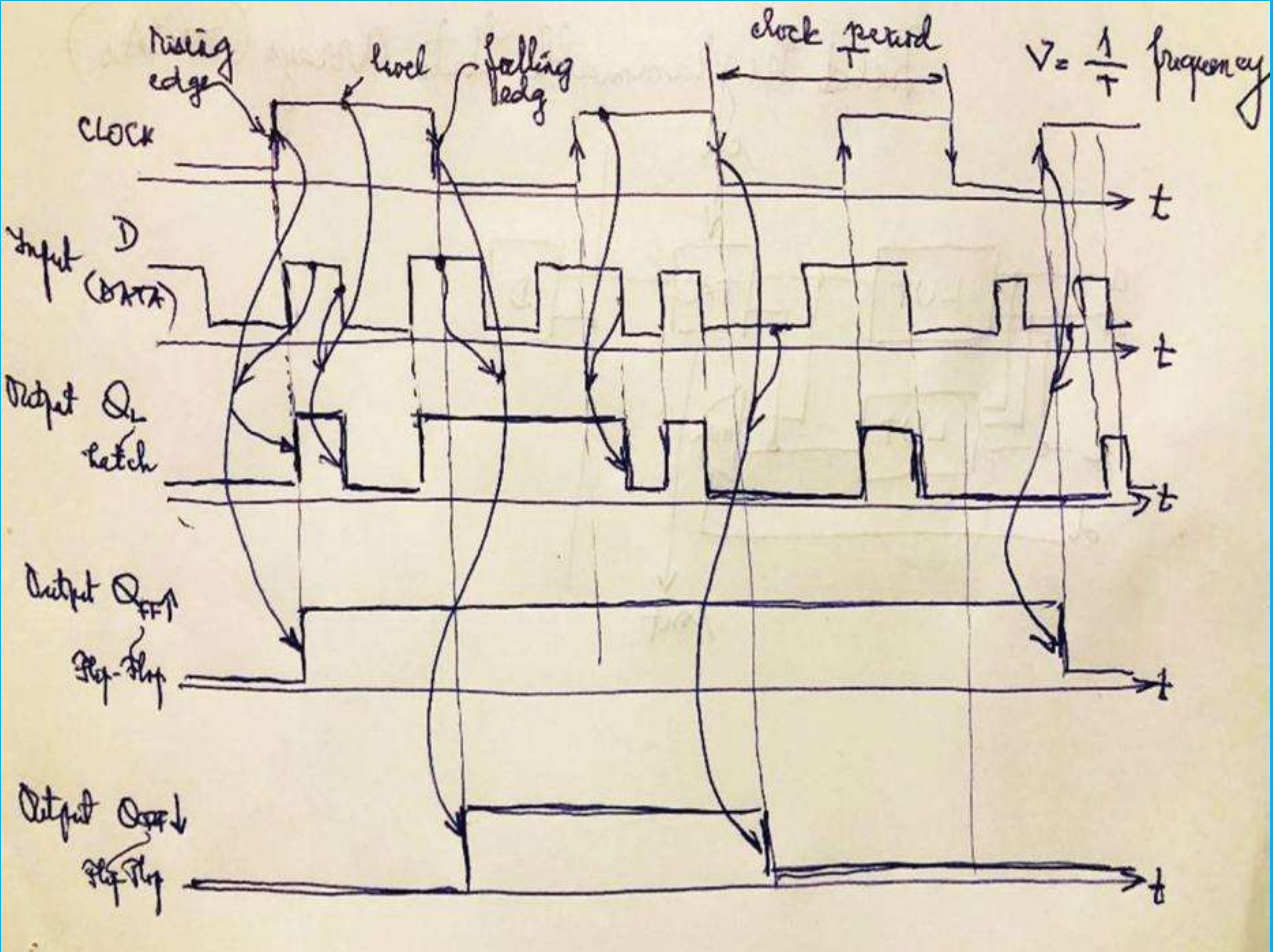
Combinational vs sequential logic circuits





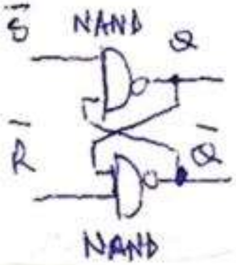
* n in general different from m
* no memory elements!
* no feedback loops

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

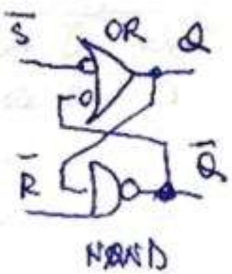


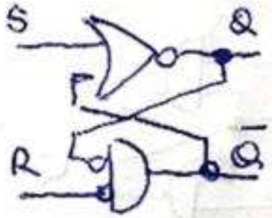
rising edge
level
falling edge
clock period
T
ν = 1/T frequency
CLOCK
t
Input D (DATA)
t
Output Q_L Latch
t
Output $Q_{FF\uparrow}$ Flip-Flop
t
Output $Q_{FF\downarrow}$ Flip-Flop
t

S-R (Set-Reset) Memory element

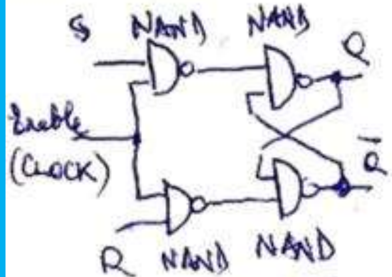


Inputs $\bar{S}$	$\bar{R}$	Output Q	$\bar{Q}$	
~~0~~	~~0~~	~~1~~	~~1~~	not allowed
0	1	1	0	
1	0	0	1	
1	1	Q	$\bar{Q}$	

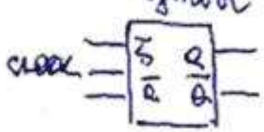




Inputs S	R	Outputs Q	$\bar{Q}$	
0	0	Q	$\bar{Q}$	
0	1	1	0	
1	0	0	1	
~~1~~	~~1~~	~~0~~	~~0~~	not allowed



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

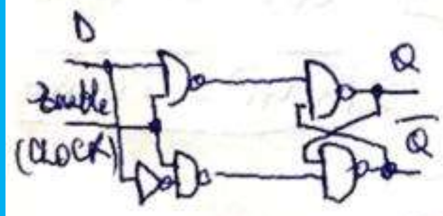
Q(t) \ SR	00	01	11	10
0	.		d	1
1	1		d	1

don't care

Characteristic equation

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

D (Delay) Memory element



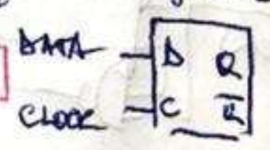
Inputs Q(t)	D	Output Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

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

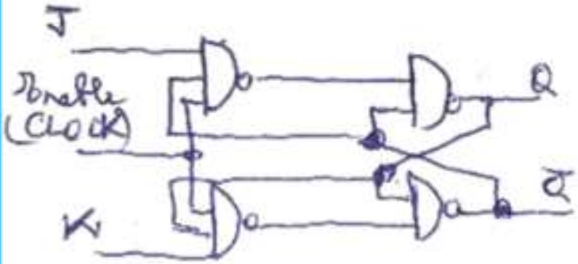
Characteristic equation

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

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

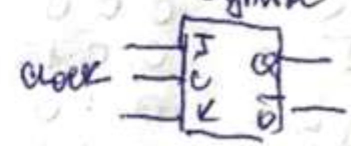
Karnaugh map

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

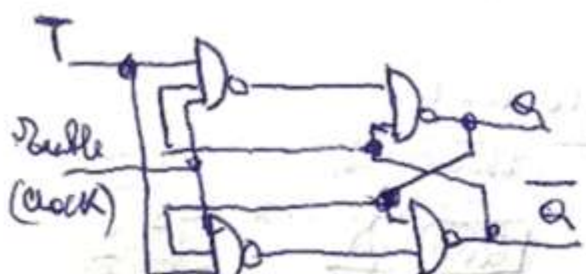
Characteristic equation

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

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

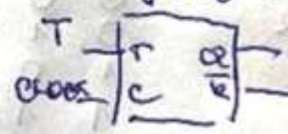
Q(t) \ T	0	1
0		1
1	1	

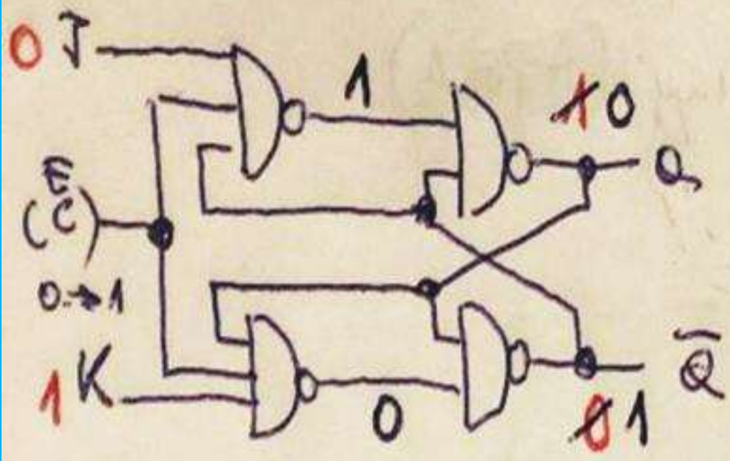
Characteristic equation

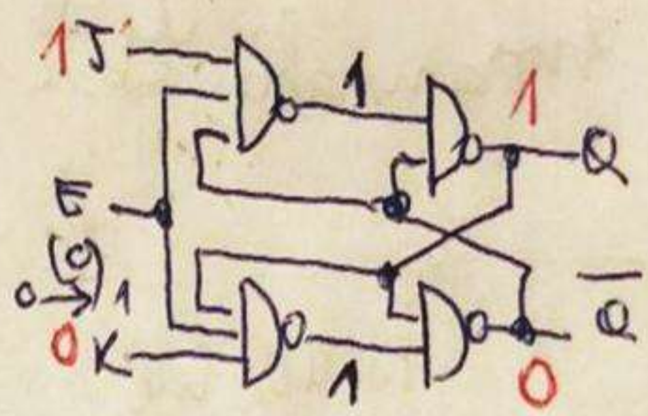
$$Q(t+1) = T\cdot\overline{Q(t)} + Q(t)\cdot\overline{T}$$

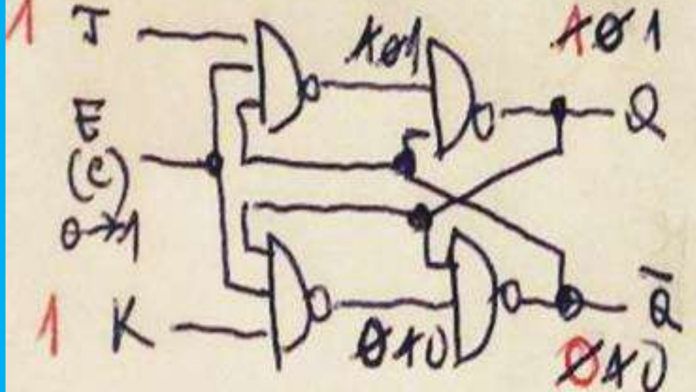
$$\Rightarrow \boxed{Q(t+1) = T \oplus Q(t)}$$

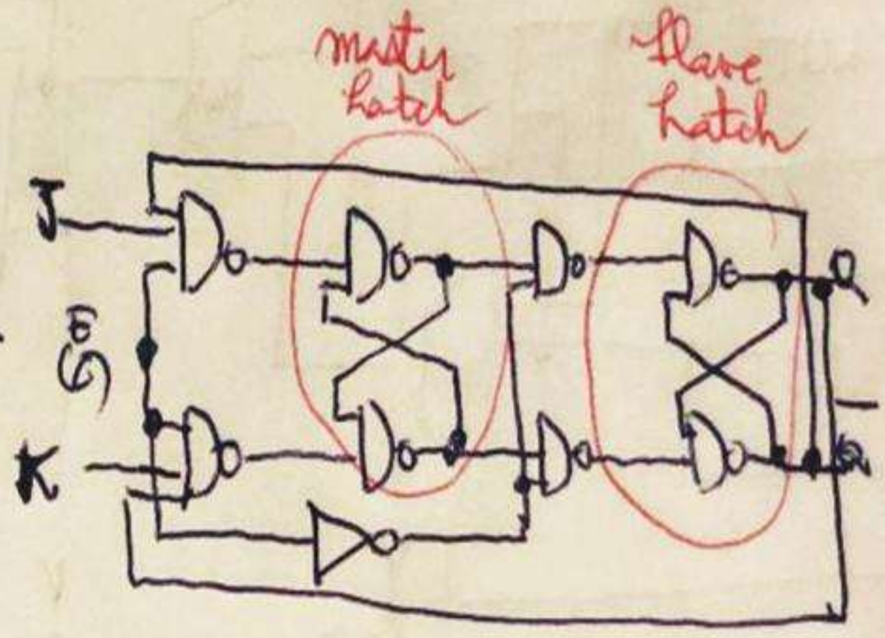
Symbol




0 J
E
(C)
0→1
1 K
1
0
1 0
Q
0 1
$\overline{Q}$

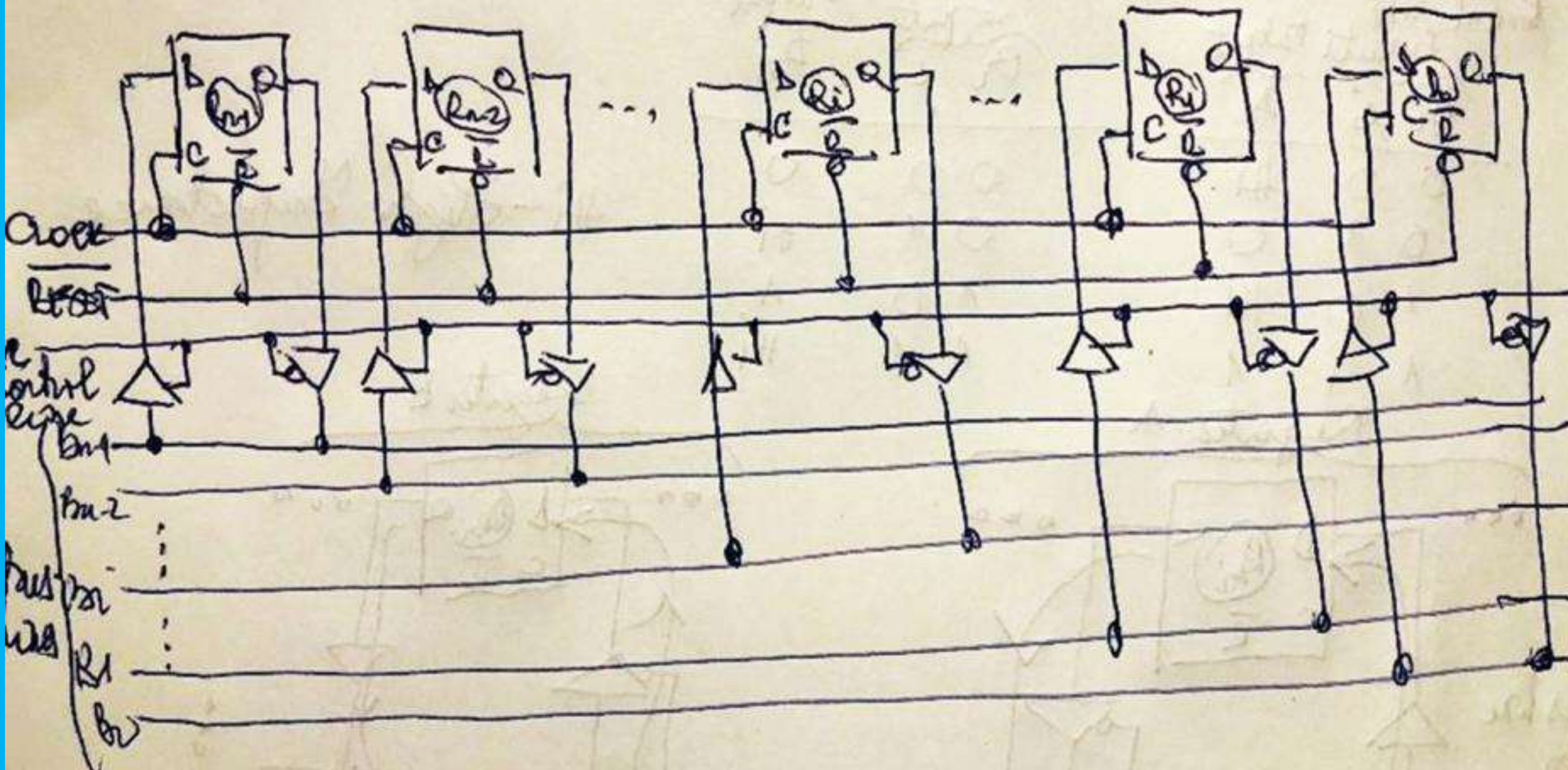

1 J
E
(C)
0→1
0 K
1
1
1
Q
0
$\overline{Q}$


1 J
E
(C)
0→1
1 K
1 0 1
1 0 1
Q
0 1 0
0 1 0
$\overline{Q}$

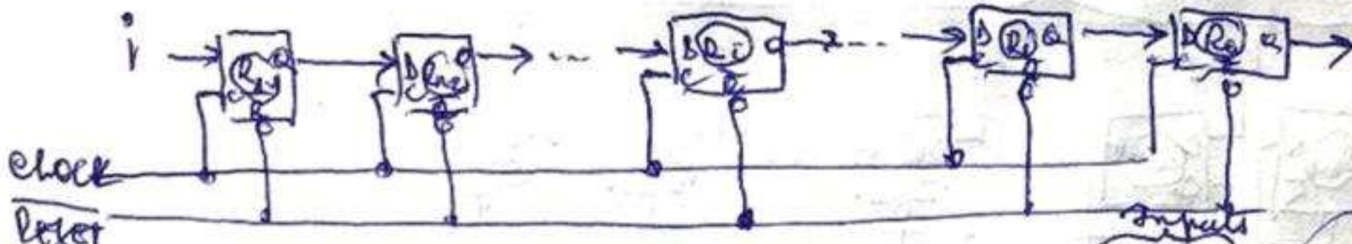

master latch
slave latch
J
E
(C)
K
Q
$\overline{Q}$

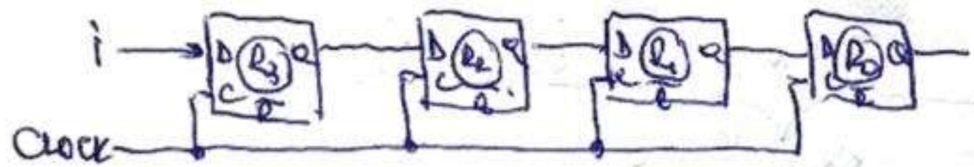
MSI sequential logic circuits

Registers



Shiftregisters





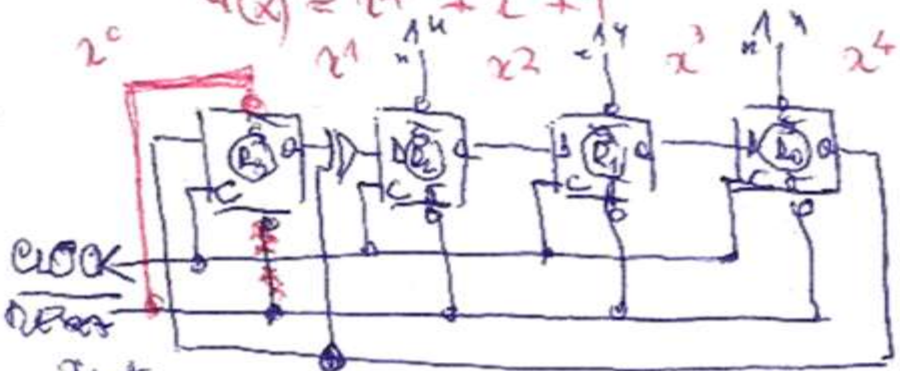
* Right - shift
* left - shift
* Right/left - shift

inputs — don't care

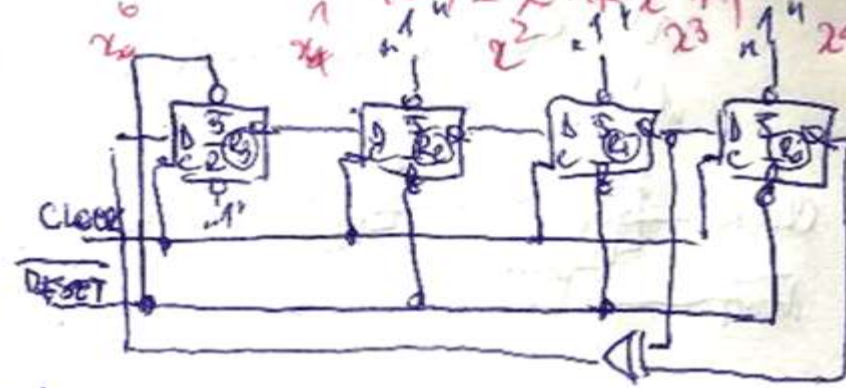
I	R	C	Q_{R3}	Q_{R2}	Q_{R1}	Q_{R0}
x	0	x	0	0	0	0
1	1	1	1	0	0	0
0	1	1	0	1	0	0
1	1	1	1	0	1	0
1	1	1	1	1	0	1
0	1	1	0	1	1	0
1	1	1	1	0	1	1
0	1	1	0	1	0	1
1	1	1	1	0	1	0
1	1	1	1	1	0	1

LFSR - Linear Feedback Shift Register

$G(x) = x^4 + x + 1$



$G(x) = x^4 + x^3 + 1$



Inputs

Decimal equivalent

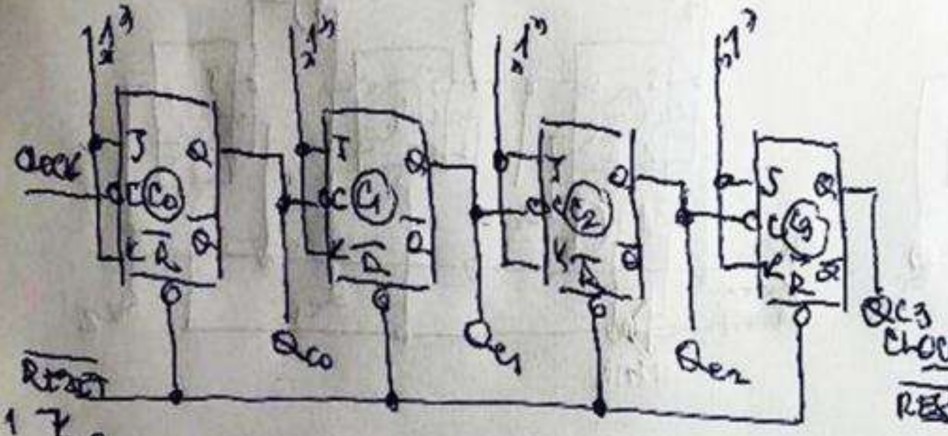
$\bar{R}$	C	Q_{R3} (2^3)	Q_{R2} (2^2)	Q_{R1} (2^1)	Q_{R0} (2^0)	DE
0	X	1	0	0	0	8
1	1	0	1	0	0	4
1	1	0	0	1	0	2
1	1	0	0	0	1	1
1	1	1	1	0	0	12
1	1	0	1	1	0	6
1	1	0	0	1	1	3
1	1	1	1	0	1	13
1	1	1	0	1	0	10
1	1	0	1	0	1	5
1	1	1	1	1	0	14
1	1	0	1	1	1	7
1	1	1	1	1	1	15
1	1	1	0	1	1	11
1	1	1	0	0	1	9
1	1	1	0	0	0	8

PRPG - Pseudo-Random Pattern Generator

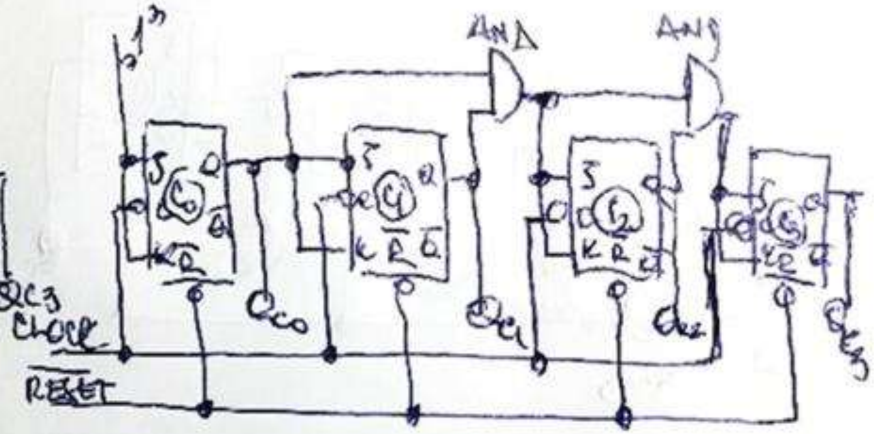
$\bar{R}$	C	Q_{R3}	Q_{R2}	Q_{R1}	Q_{R0}	DE
0	X	1	0	0	0	8
1	1	0	1	0	0	4
1	1	0	0	1	0	2
1	1	1	0	0	1	9
1	1	1	1	0	0	12
1	1	0	1	1	0	6
1	1	1	0	1	1	11
1	1	0	1	0	1	5
1	1	1	0	1	0	10
1	1	1	1	0	1	13
1	1	1	1	1	0	14
1	1	1	1	1	1	15
1	1	0	1	1	1	7
1	1	0	0	1	1	3
1	1	0	0	0	1	1
1	1	1	0	0	0	8

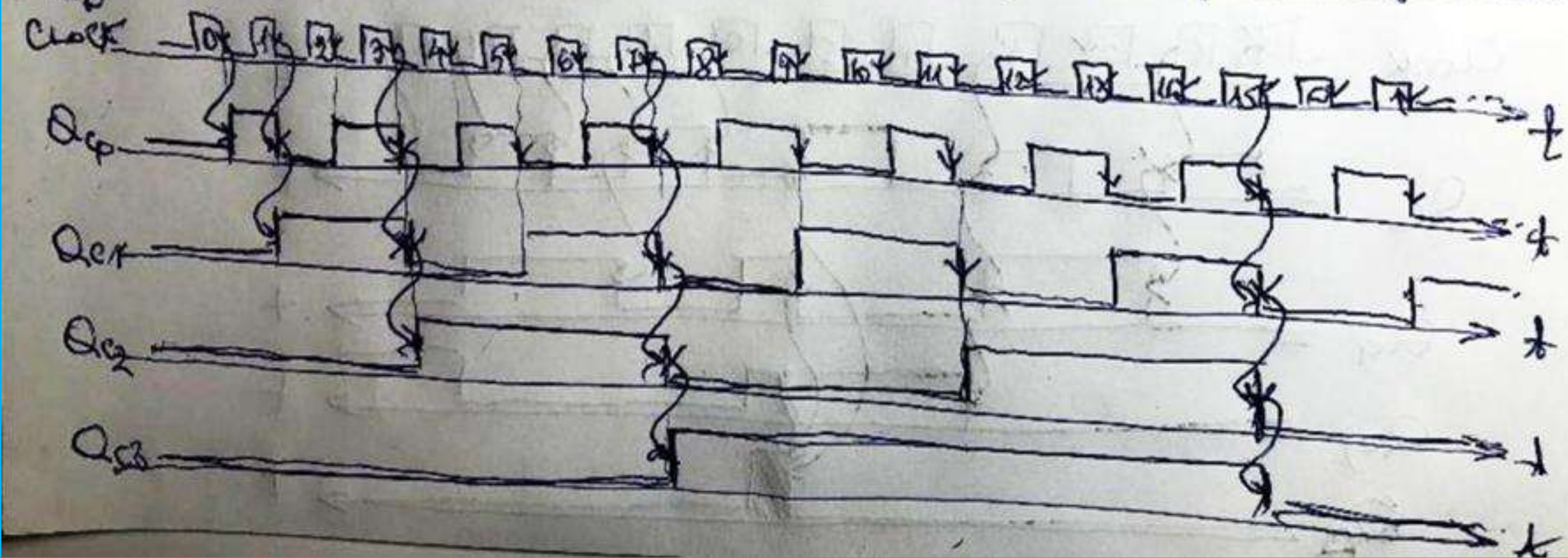
Counters

a) Asynchronous



b) Synchronous

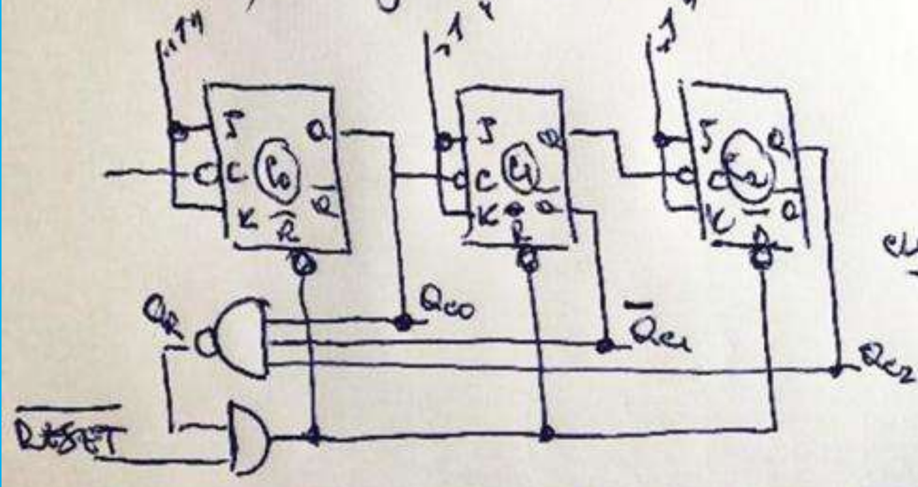


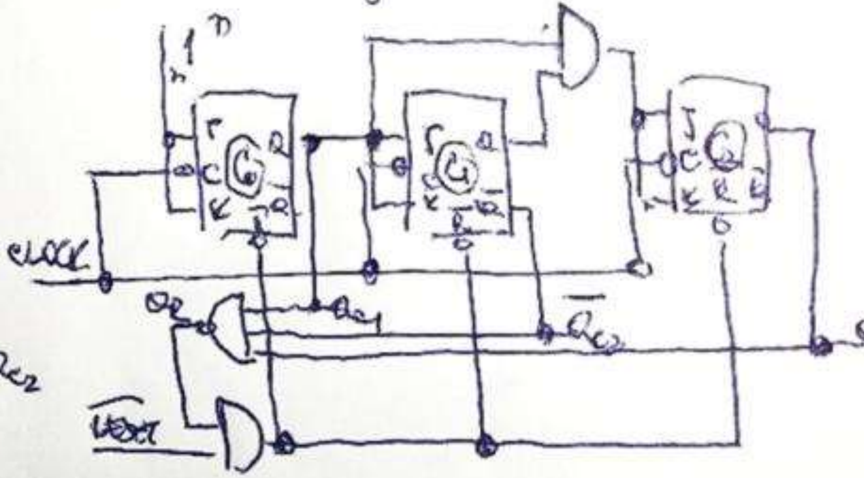


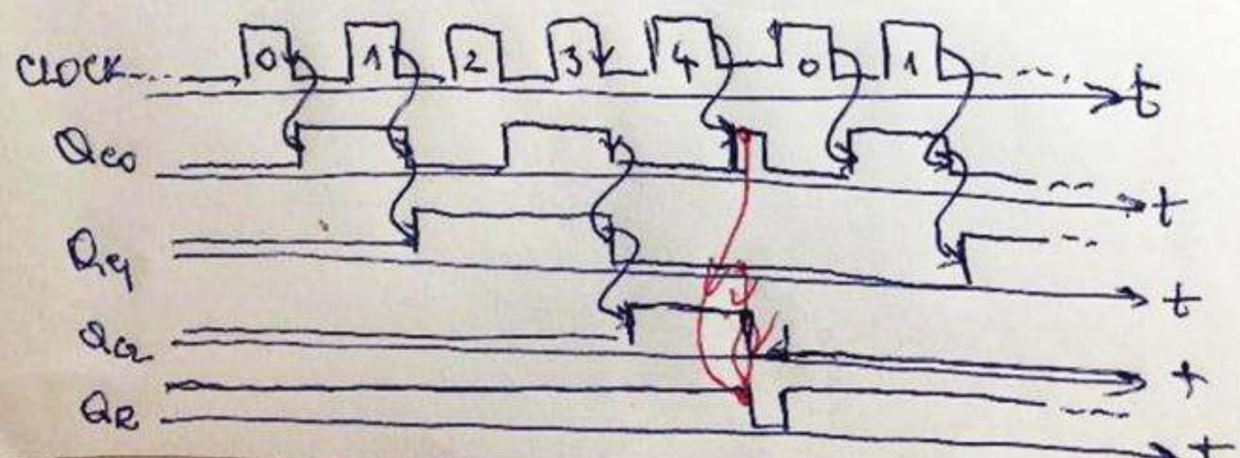
Modulo 5 binary counter

a) Asynchronous

b) Synchronous



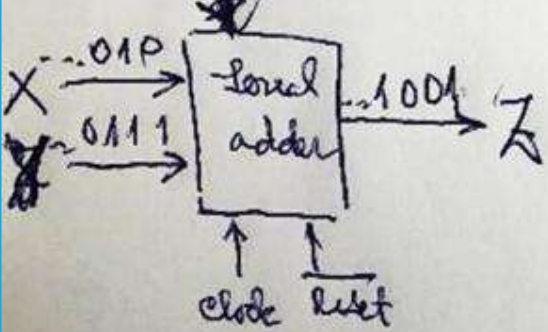




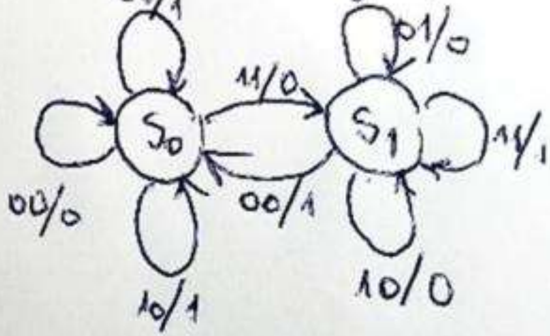
Design of sequential logic circuits

Problem statement → State table construct → Choice of types of memory elements → Transition table construct → Logic equation minimization → Logic schematic 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 $w(t+1) = D$ Output

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	?		

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

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

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

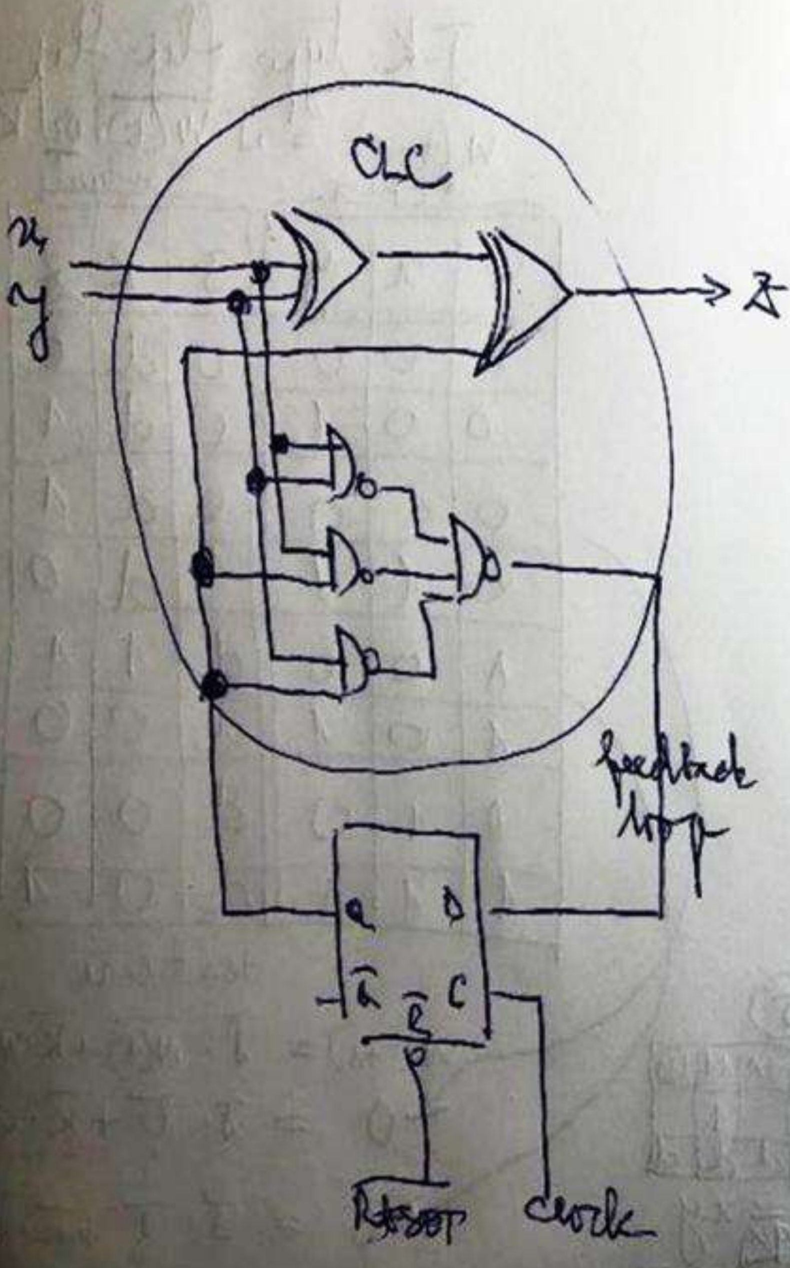
$0 = J \cdot \overline{1} + \overline{K} \cdot 1$

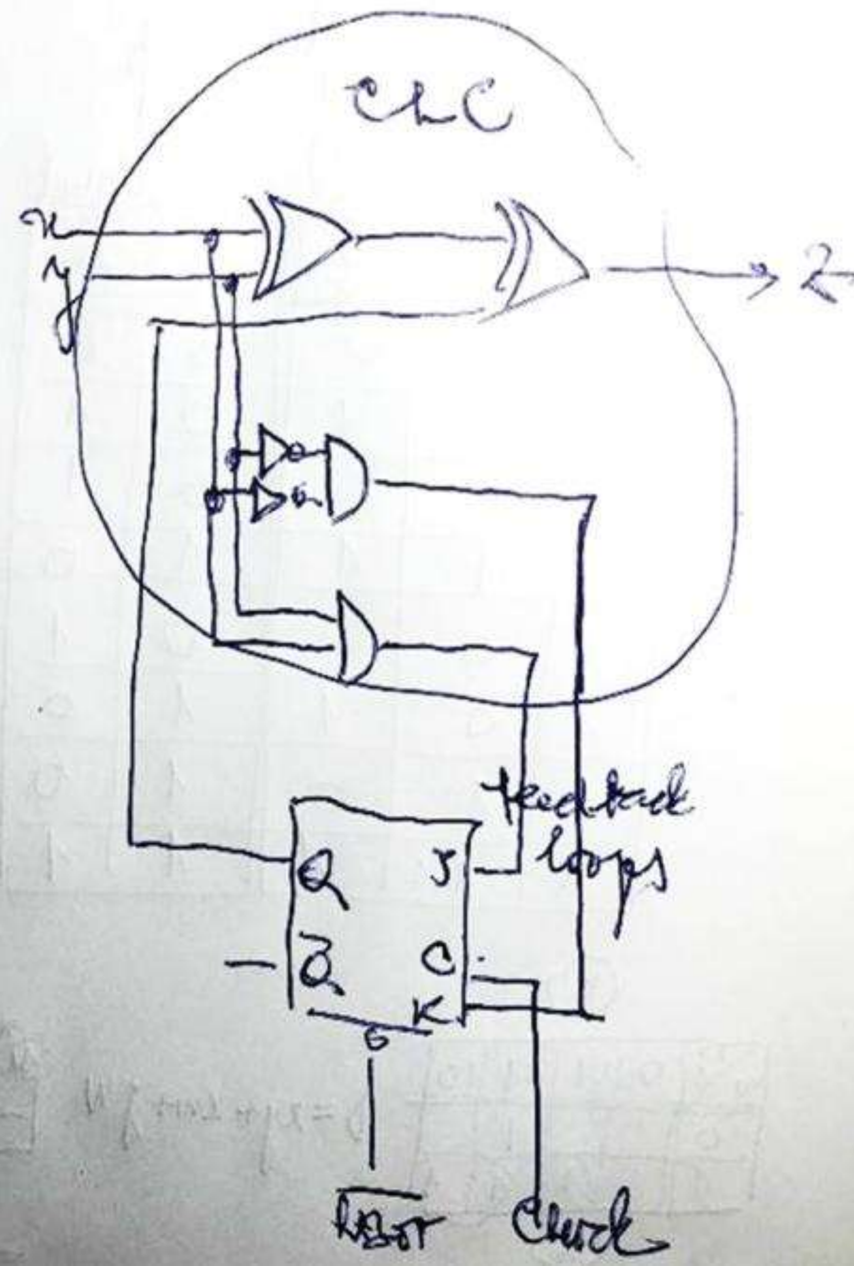
$\Downarrow$

$\overline{K} = 0$

$\downarrow$

$K = 1$


CLC
x
y
z
feedback loop
Q
D
C
Reset
Clock


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