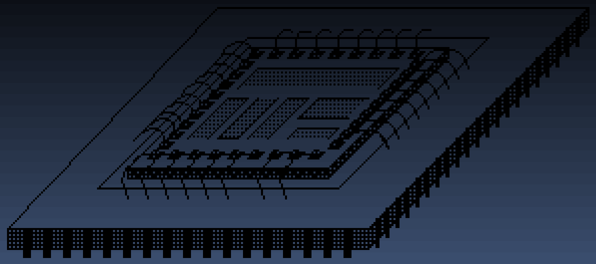

привет
hälsningar
ciao
hola
hallo
hei
hi
γειασου
bonjour
喴
oi

Theory of Digital Machines

CMPT-281



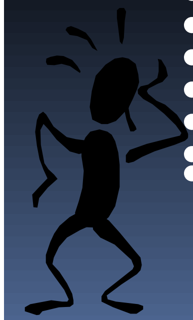
Dr. Antoniou, G

281 Info

Lecture: syllabus

Lab:

Office Hours: syllabus



281 Info

[Lecture:]
[Lab:]
[Office Hours:]

How to write a homework or a project?...Class Notes

go to: **http://canvas.montclair.edu**

281 Info

Lecture:
Lab:
Office Hours:

How to write a homework or a project?...Class Notes

go to: **http://blackboard.montclair.edu**

Grades-

HW & Project	(20% + 20%)
MidTerm	(30%)
Final	(30%)

281 Mechanics

- No Late Homeworks Accepted. Homeworks (hard-copy) must be returned-in at the beginning of the class.

  e.mail your homework >> (CMPT281@gMail.com)
- PowerPoint web-course-lecture-notes are available.
- Read the material before you come to the class …
- The exams are based on the class lectures … and NOT on the incomplete web-course-notes.
- Term Projects will be presented in the classroom
- Attendance is more than essential in order to get a decent grade.

Questions …

CMPT-281; Practical course

- A Very Practical course with countless applications … includes also software (programming) …
- Hardware digital design and Karnaugh maps … will improve your computing understanding
- Learning hardware design will give you even more insight into computing

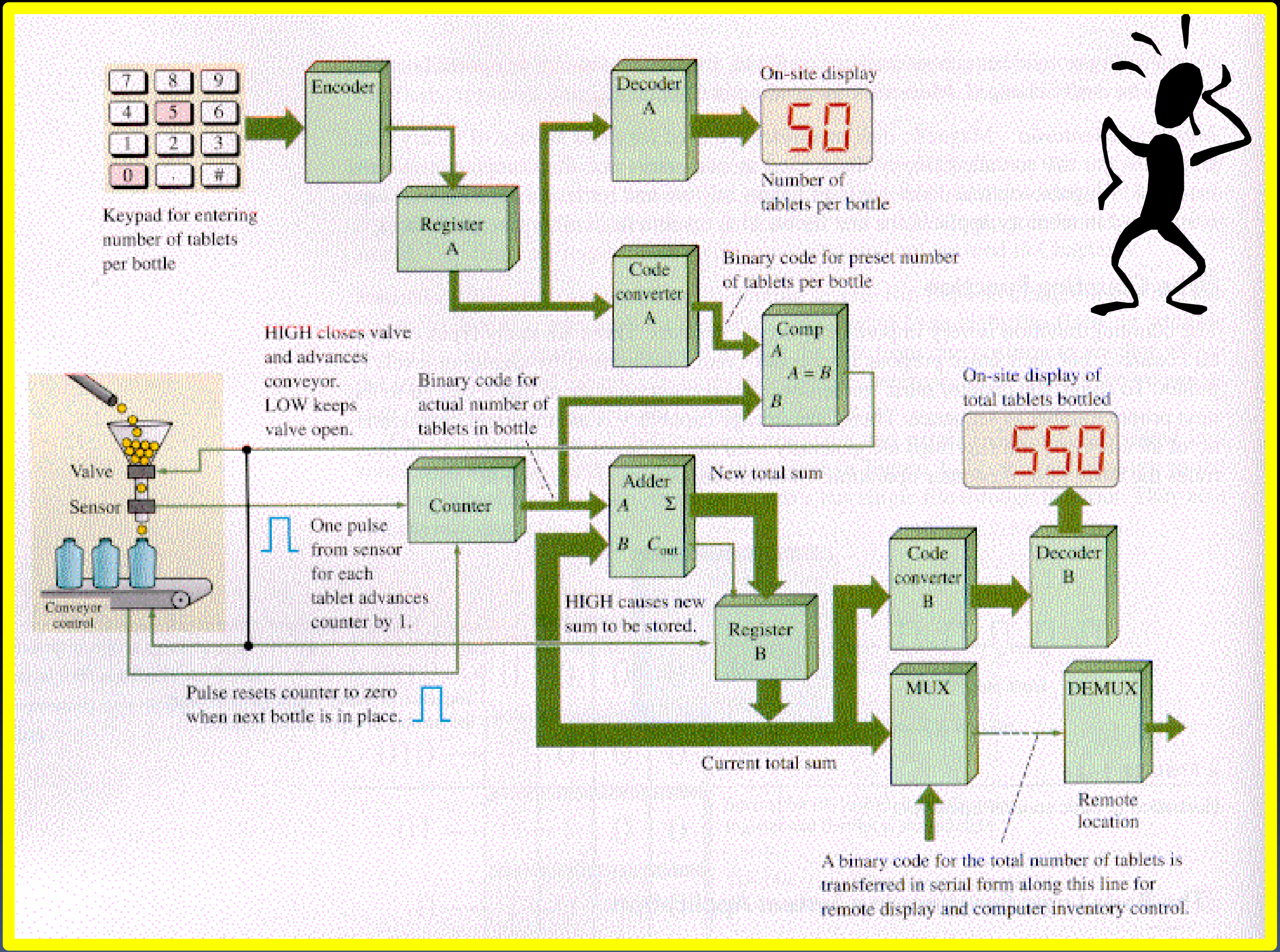
As the first "hardware course" … it is a learning experience … plus 1 or 2-5 more lines in your resume …

CMPT-281

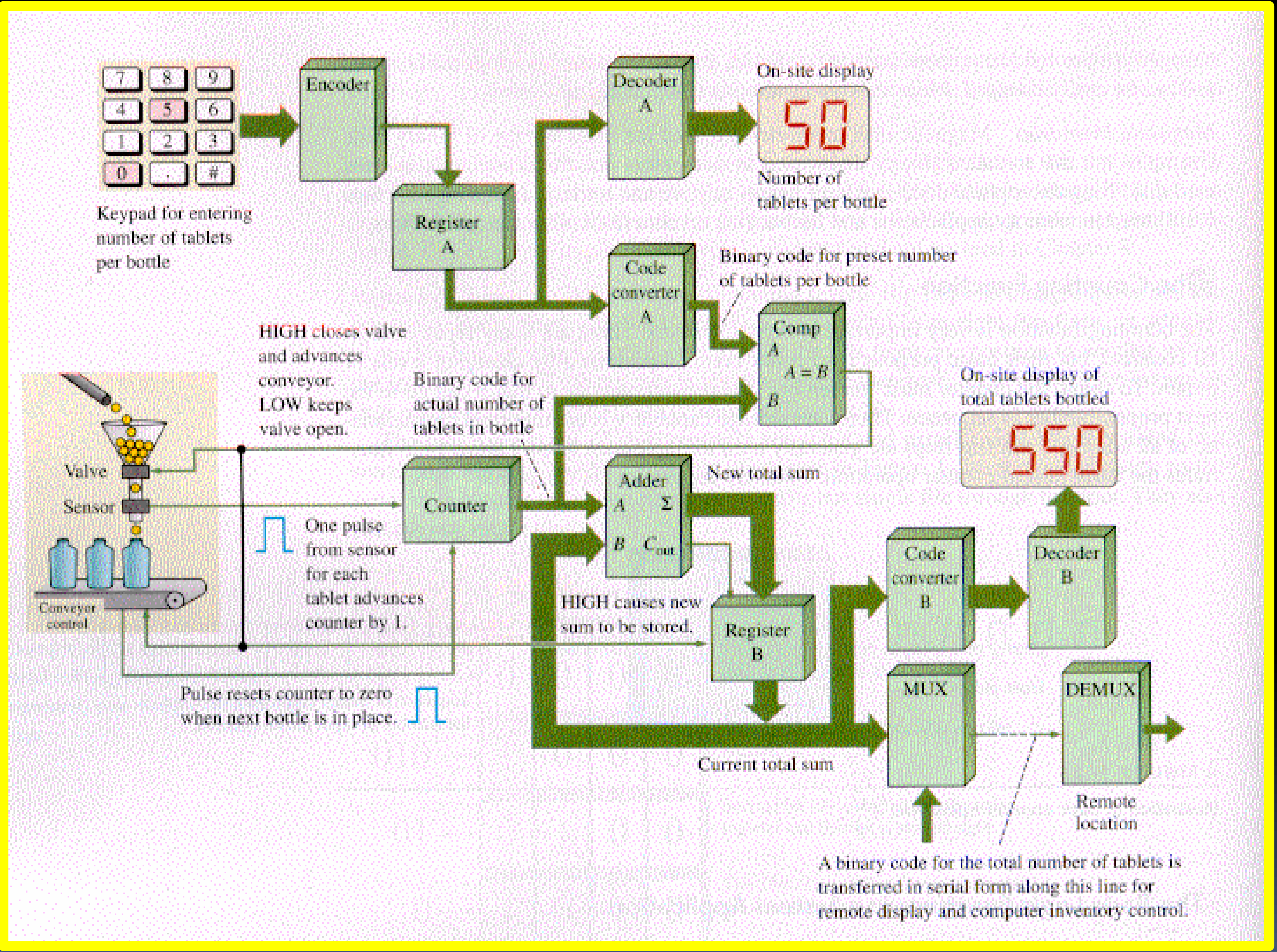
- A Very Practical course with countless applications … includes also software (programming) …
- No prior theoretical knowledge is required
- Computer science background is required
- Programming experience is required

As the first "hardware course" … it is a learning experience … plus 1 or 2-5 more lines in your resume …

Digital Technology; tablet-counting and bottling system



Digital Technology; tablet-counting and bottling system



This is the goal of the course ....

Digital Computers

Digital Computers? …. Why not only Computers

Digit?

Digital Computers

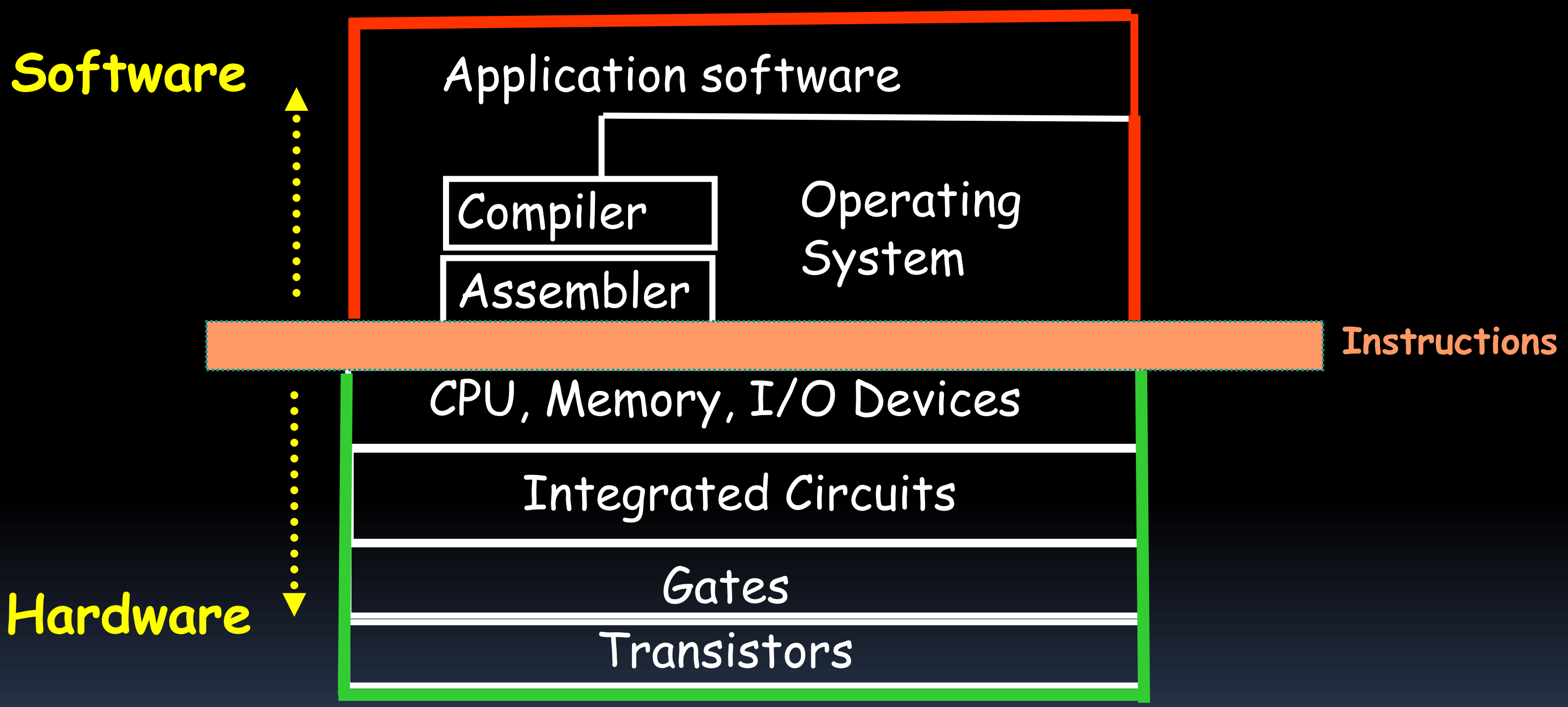
- Software + Hardware

Digital Computers

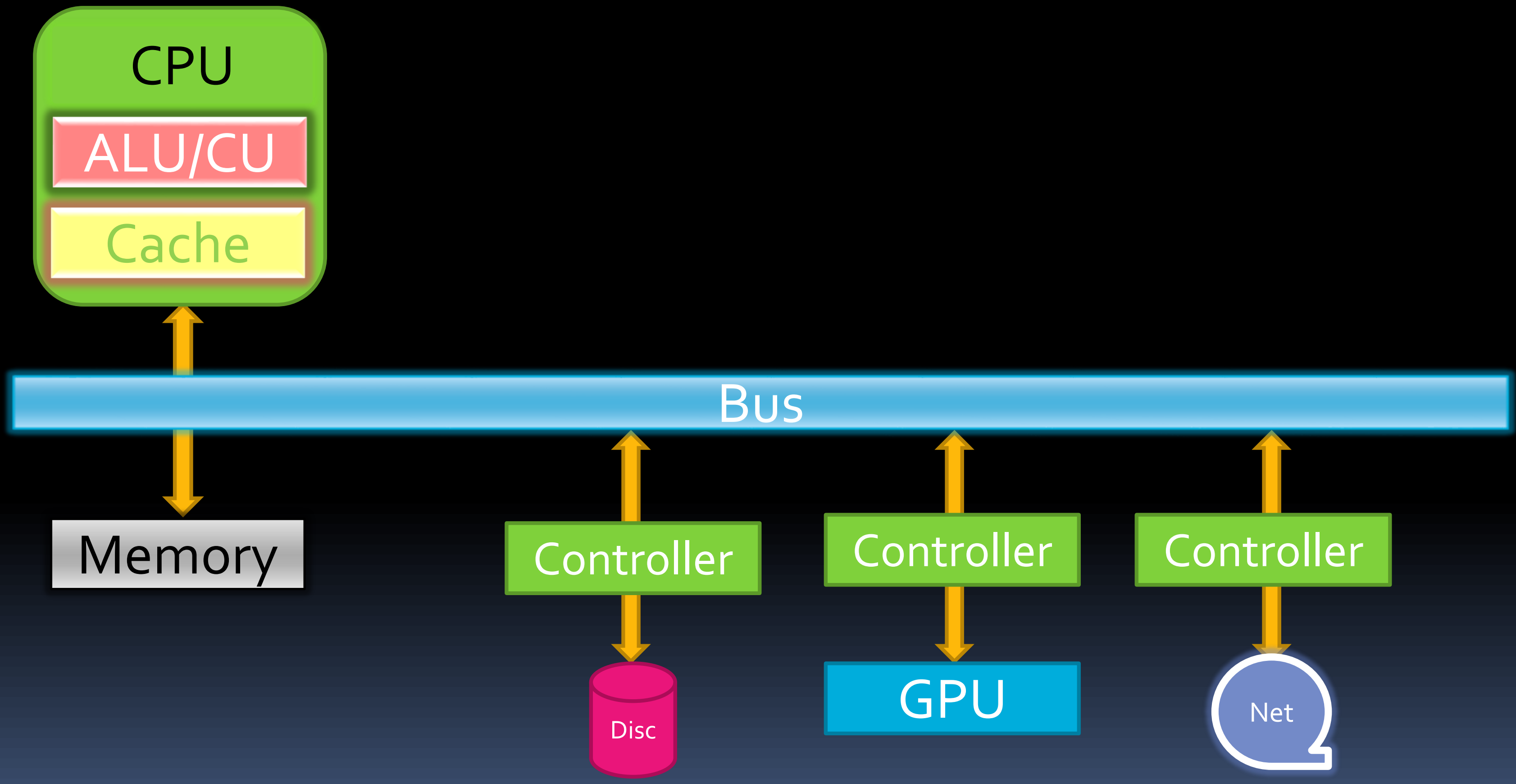
- Software + Hardware

Instructions - programs -

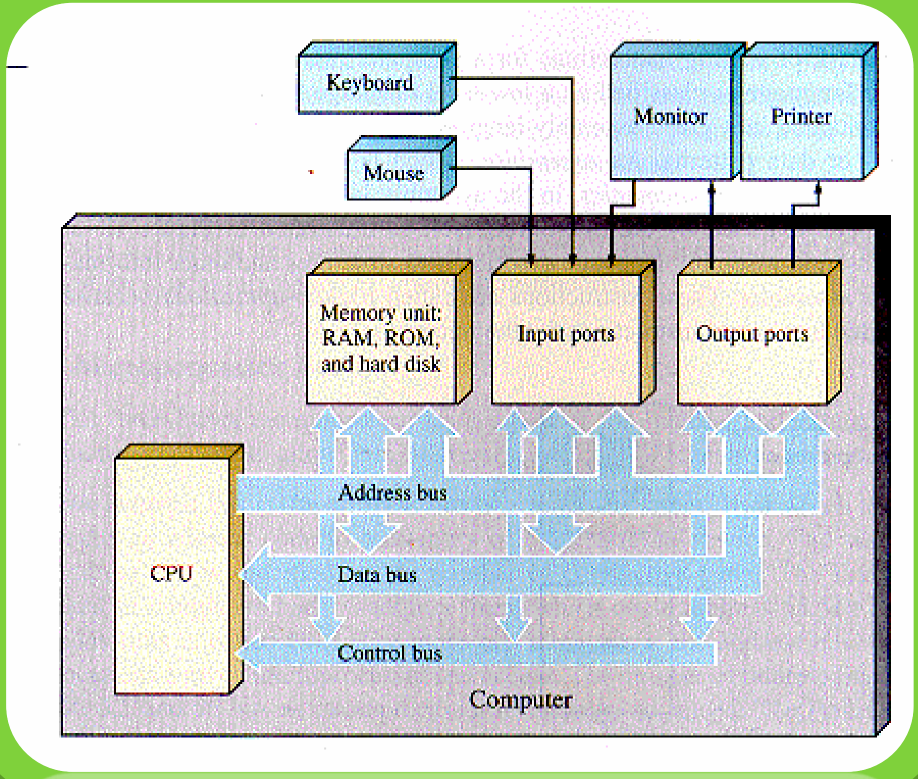
Software & hardware



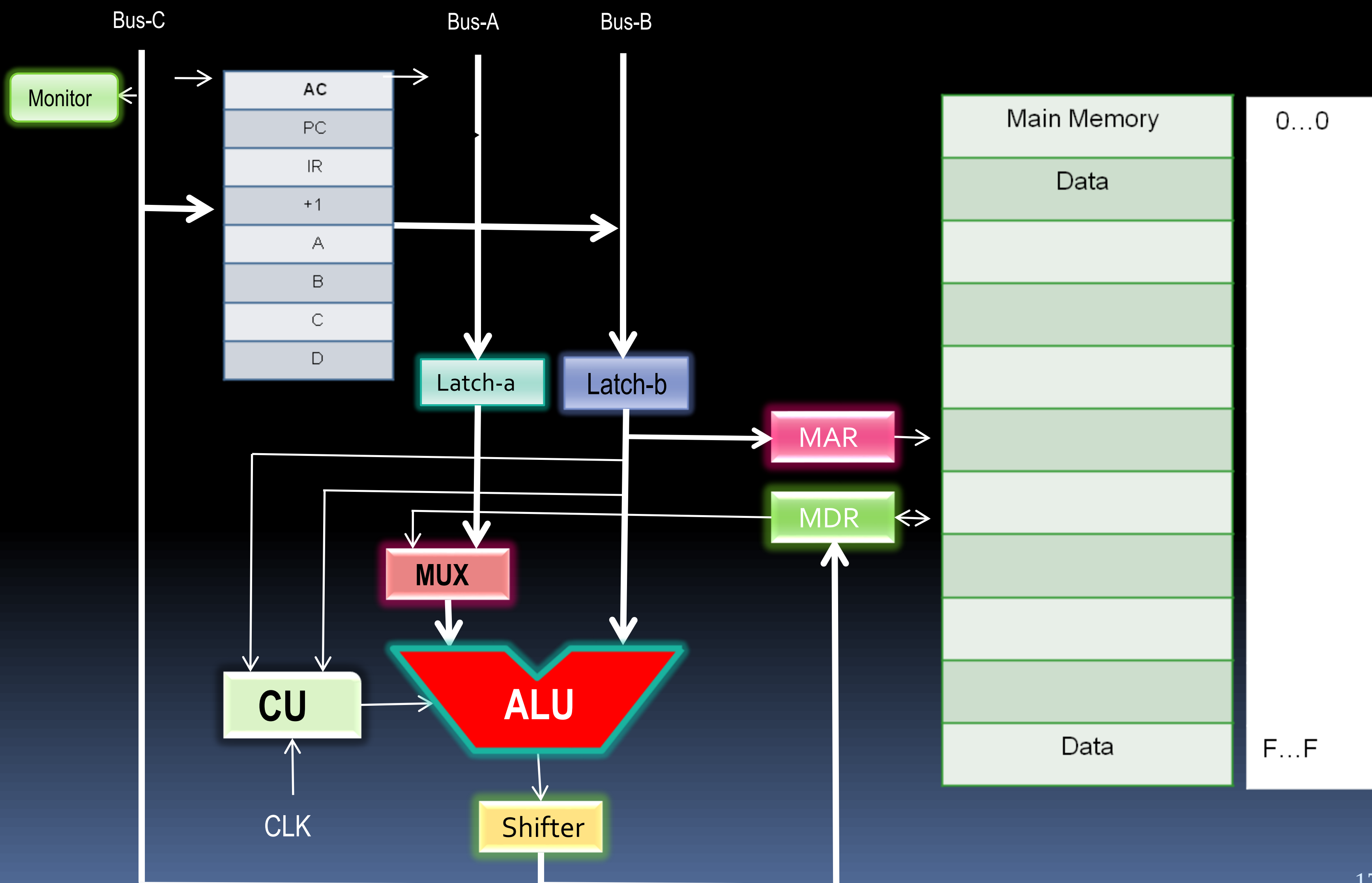
Digital Computer



Digital Computer



CPU-Memory


Bus-C
Bus-A
Bus-B
Monitor
AC
PC
IR
+1
A
B
C
D
Latch-a
Latch-b
MAR
MDR
MUX
ALU
CU
CLK
Shifter
Main Memory
Data
Data
0…0
F…F

Digital computer technology today?

Digital computer technology today?

- PC, PDA, Workstations, MultiCPU computers
- CD and DVD players
- MP3 Players
- CellPhones
- TV
- Internet
- Satellite Radio
- Microwave ovens
- Security systems
- Cars
- GPS Receivers, …

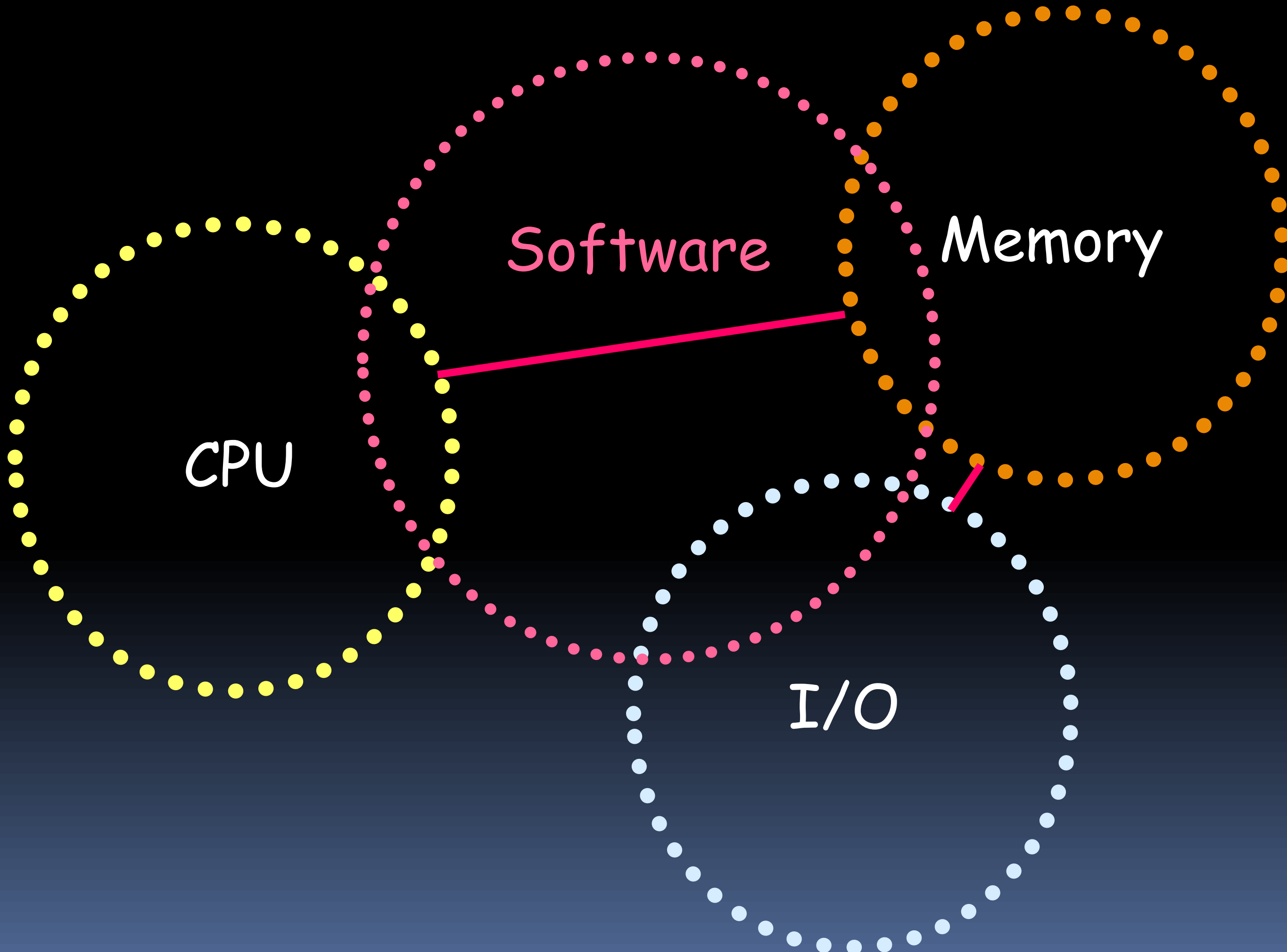
Digital Computers

- Software + Hardware

Hardware:

- CPU [Control Unit / Arithmetic Logic Unit]
- Memory
- Input/Output devices

A Computer System



Digital Computer Design

- Analyze
- Design
- Implement with software based techniques

How?

Using VHDL

VHDL (V Hardware Description Language).

where, V = VHSIC (Very High Speed Integrated Circuits)

- CD--- in your book
- Can also download for free: www.altera.com

QUARTUS® II

Also available in the CS-Laboratory (RI-105)

Digital Computer Design

- Analyze
- Design
- Implement with software based techniques
- Realize using hardware

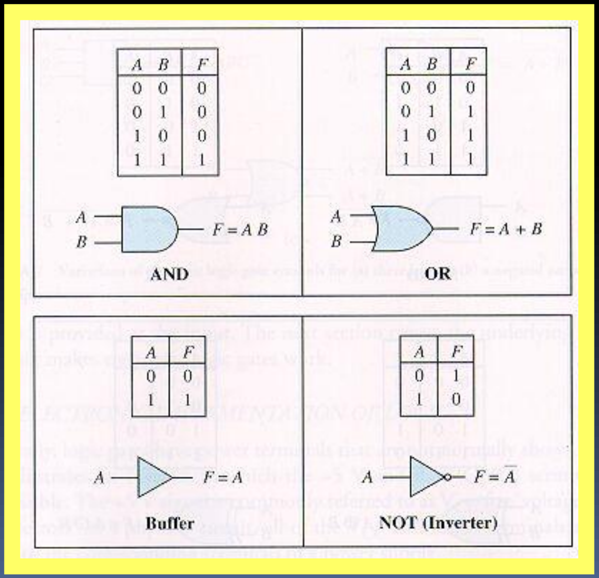
Digital Computer Design

- Analyze
- Design
- Implement with software based techniques
- Realize using hardware

Hardware?

Hardware

- Gates ++



Final step-Test

- Analyze
- Design
- Implement with software based techniques
- Realize using hardware
- Test

Digital Computer Fundamentals

Digital computers use two states

- 2 States: 0 and 1 = Binary System
- 0 = OFF, 1 = ON

Bi in Latin = 2

Digital computers; 2 voltage levels

- 2 States: 0 and 1 = Binary System

- 0 = OFF, 1 = ON
- 0 = Down, 1 = Up
- 0 = 0 volts, 1 = 3-5 volts

An array of 1’s and 0’s

- 2 States: 0 and 1 = Binary System
- 0 = OFF, 1 = ON
- 0 = Down, 1 = Up
- 0 = 0 volts, 1 = 3-5 volts

1 0 1 1 0

A digital signal… 10110…

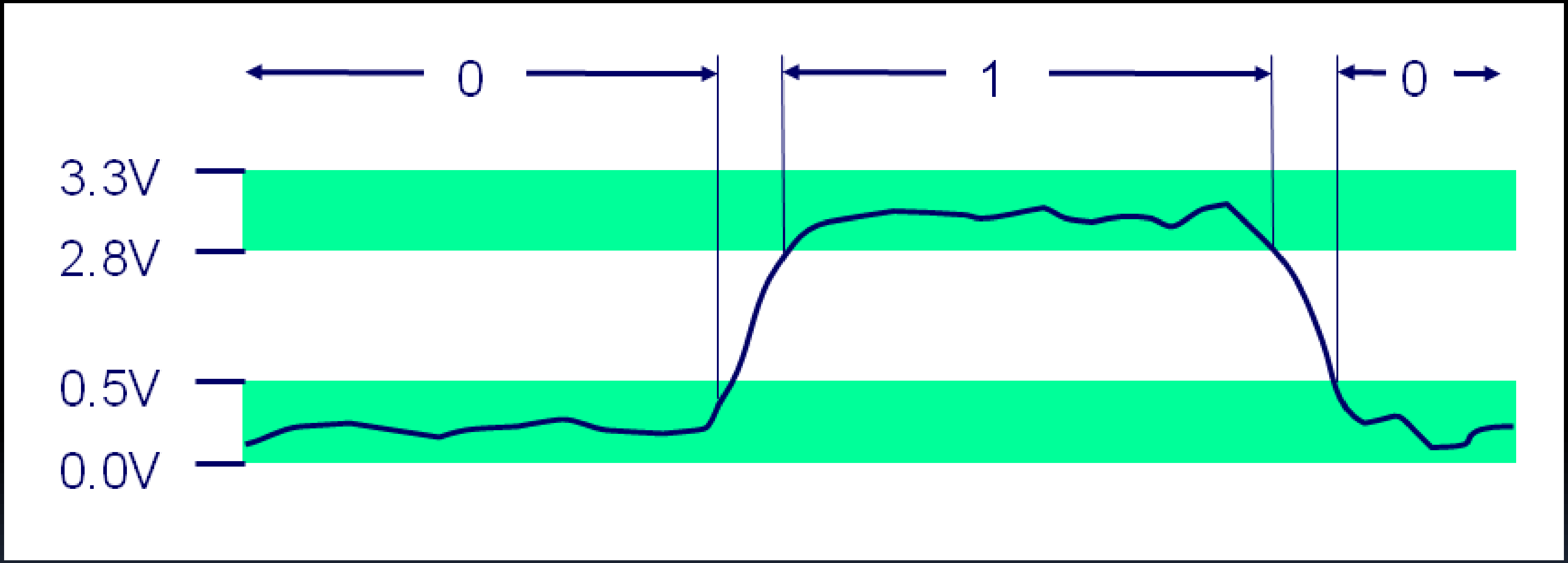
- 2 States: 0 and 1 = Binary System
- 0 = OFF, 1 = ON
- 0 = Down, 1 = Up
- 0 = 0 volts, 1 = 3-5 volts

A digital signal… 10110…

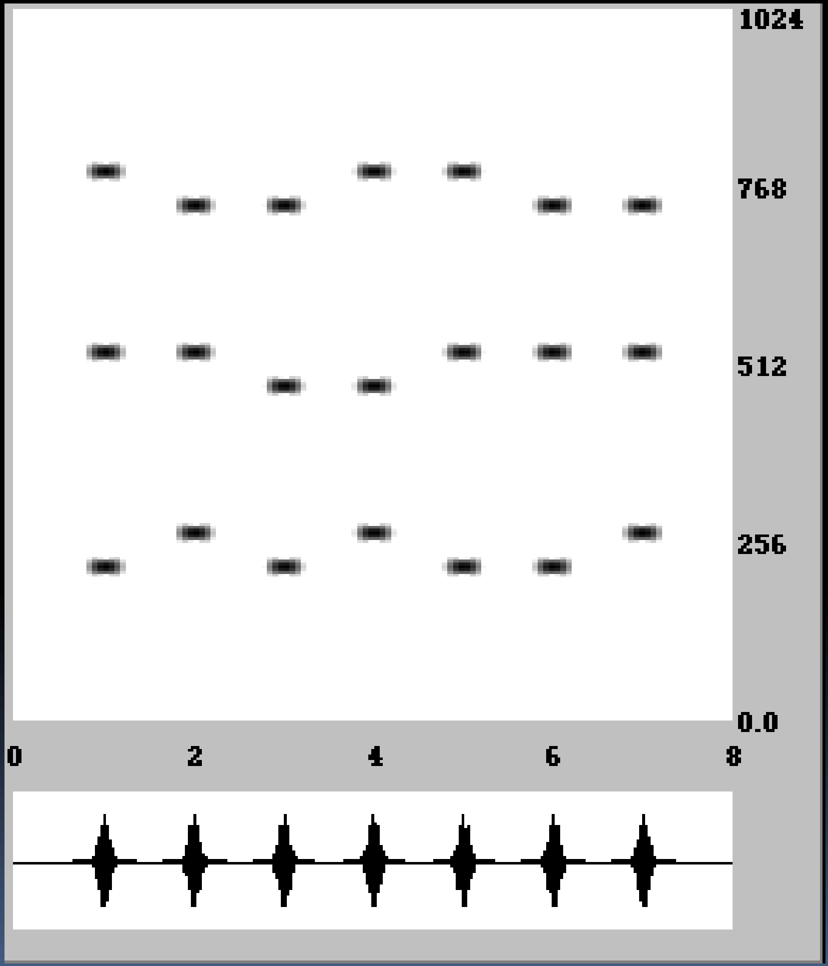
- 2 States: 0 and 1 = Binary System

- 0 = OFF, 1 = ON
- 0 = Down, 1 = Up
- 0 = 0 volts, 1 = 3-5 volts

It is all voltage … electricity

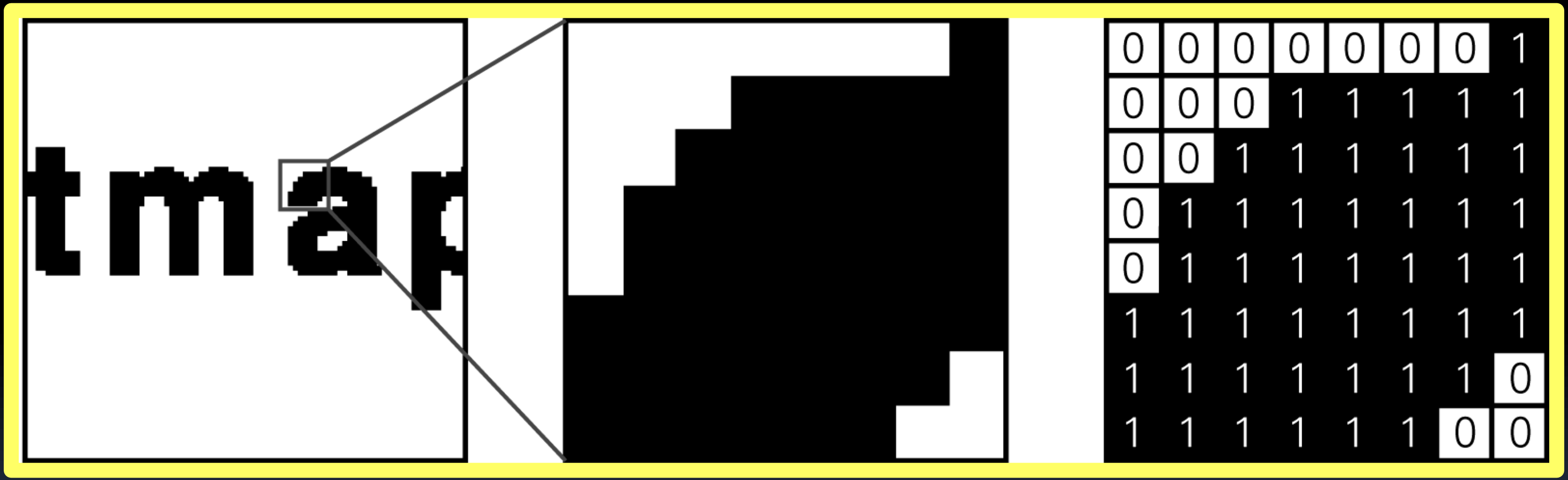


Actual binary voltage values



http://www.uwec.edu

All … are 1's and 0's



Black-and-White or bi-level digital image

DIGITAL COMPUTERS

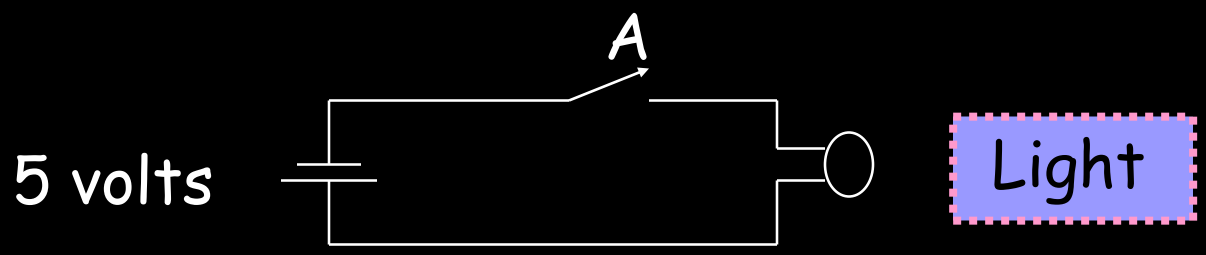
↕

Binary Logic

The binary logic is implemented with switches

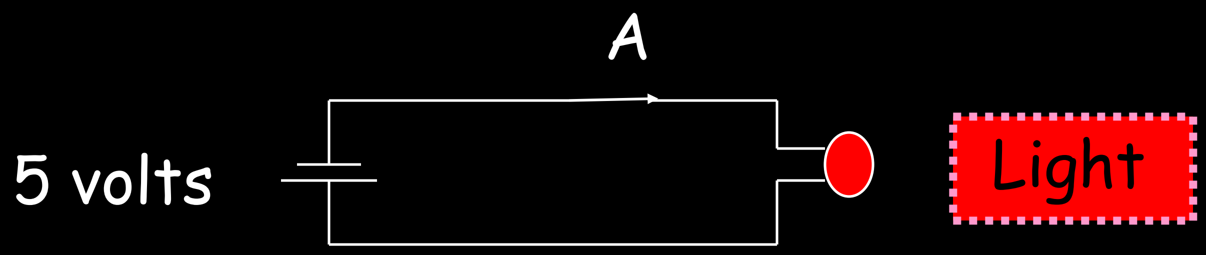
- Relays → Mechanical
- Semiconductors → Electronic

Basic Principle: Switch



A	Light
Open	0

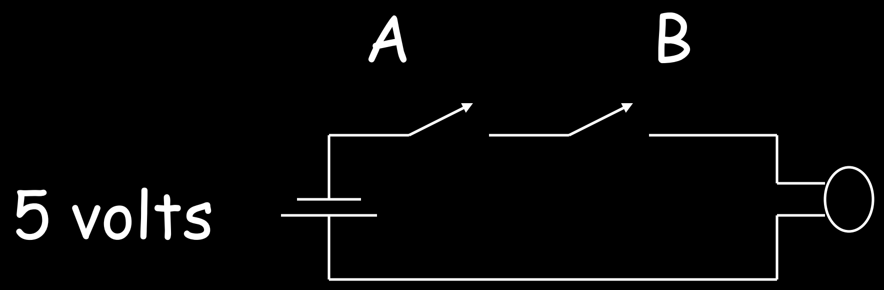
Basic Principle: Switch



A	Light
Open	0
Closed	1

Truth Table

AND

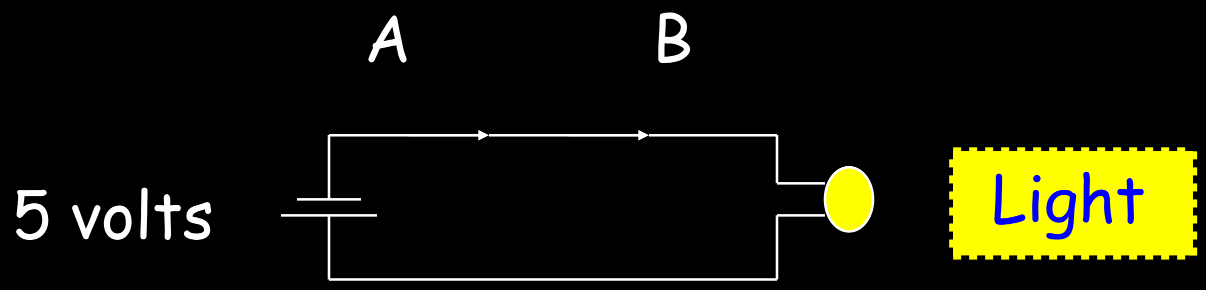


A	B	Light
0	0	
0	1	
1	0	
1	1	

Truth Table

AND

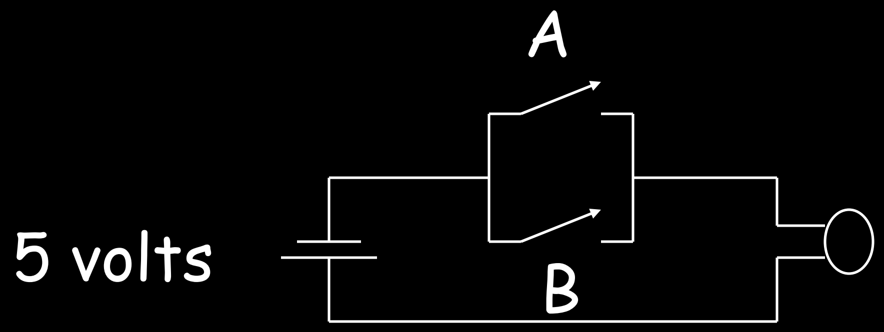
A (and) B = A • B



A	B	Light
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table

OR

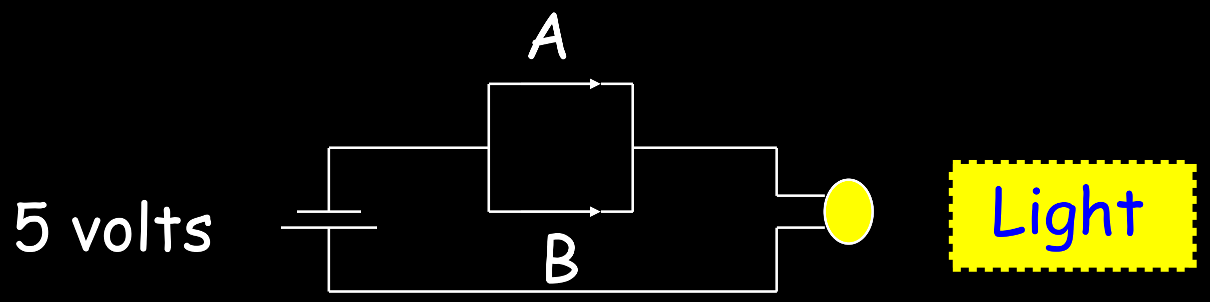


A	B	Light
0	0	
0	1	
1	0	
1	1	

Truth Table

OR

A (or) B = A + **B**



A	B	Light
0	0	0
0	1	1
1	0	1
1	1	1

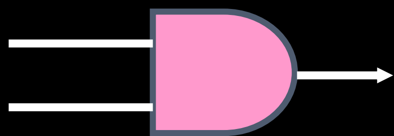
Truth Table

NOT

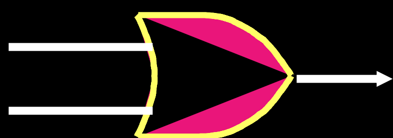
A_1	$\overline{A_1}$
0	1
1	0

Gate symbols

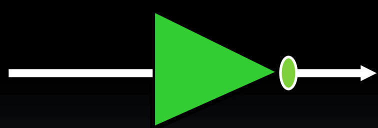
Gate Delay (nsec)



AND 2.4

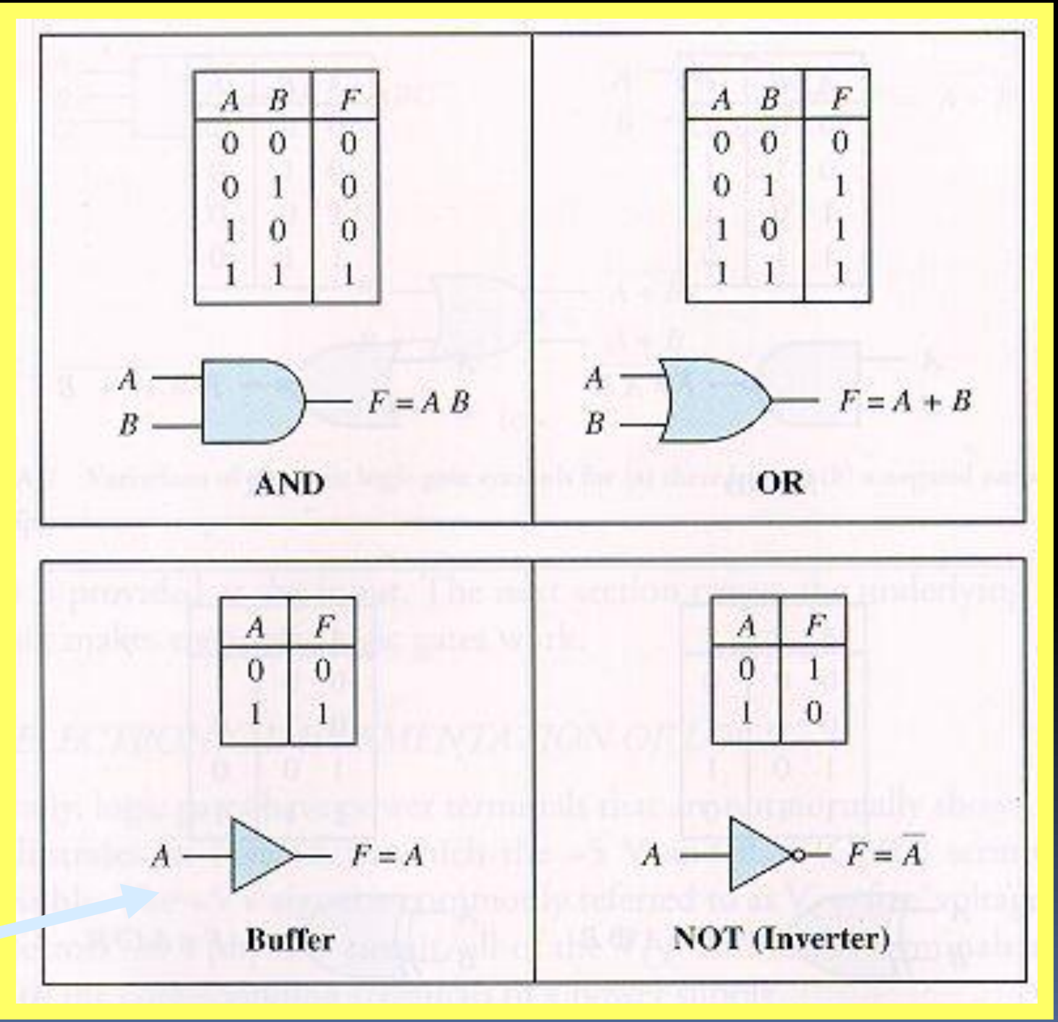


OR 2.4



NOT 1.0

The 4 basic gates and truth tables



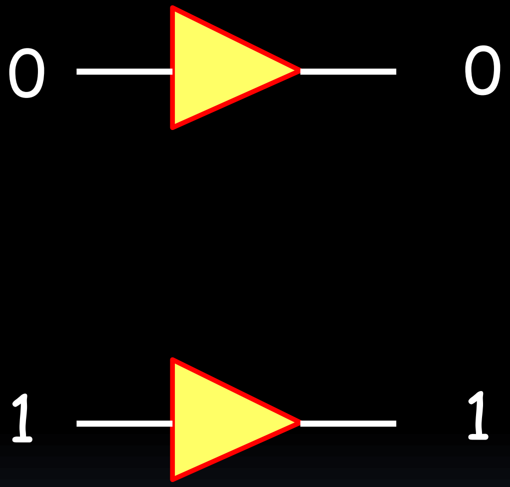
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

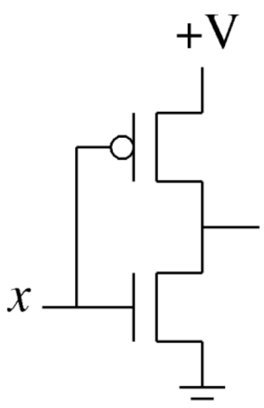
A	F
0	0
1	1

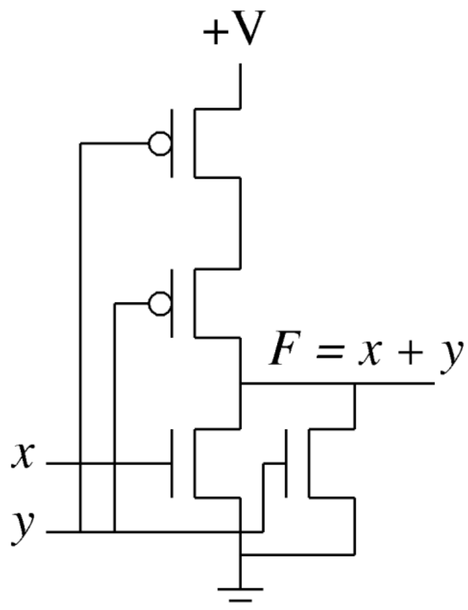
A	F
0	1
1	0

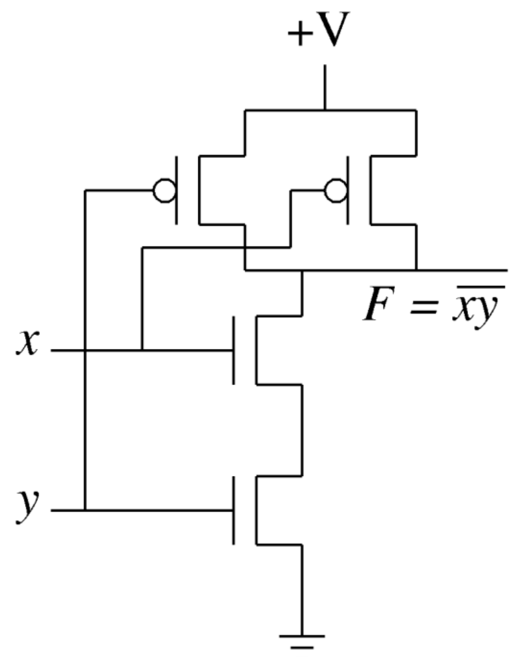
Buffer



The reality – Transistors (CMOS)

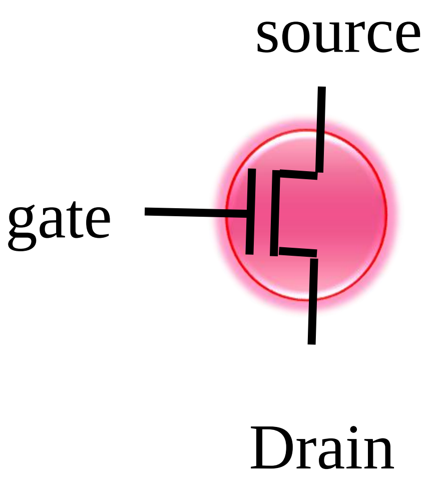


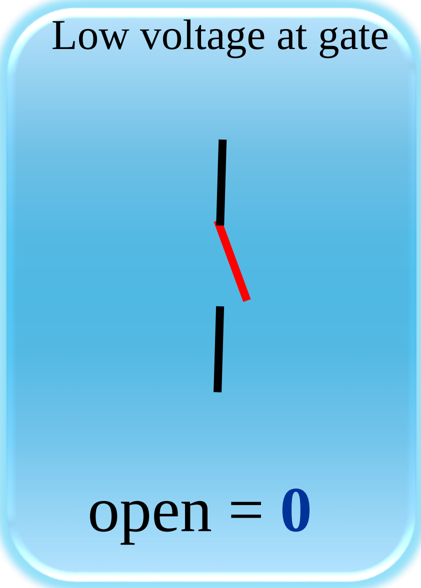


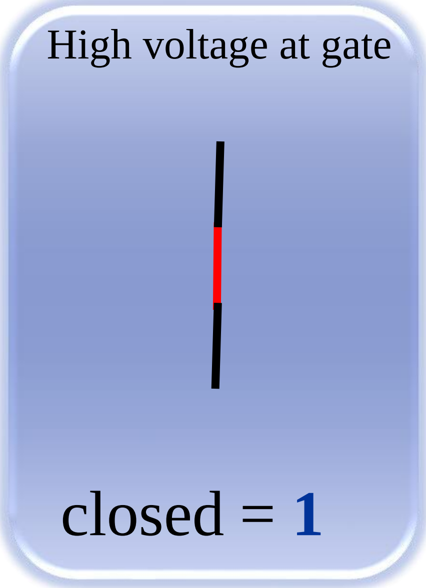


***CMOS* = *C*omplementary *Me*tal *Ox*ide *Se*miconductor**

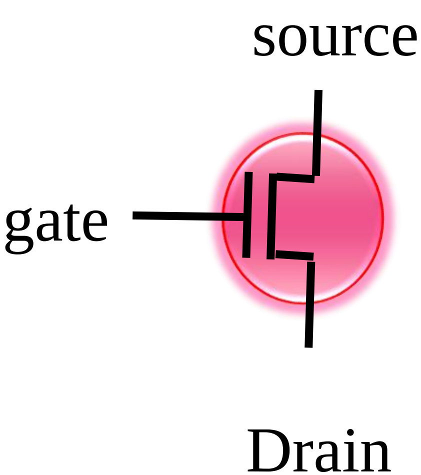
Transistor (CMOS) .... 0 and 1

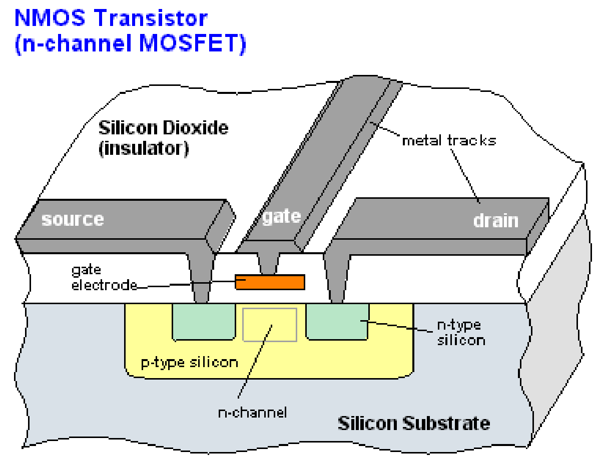


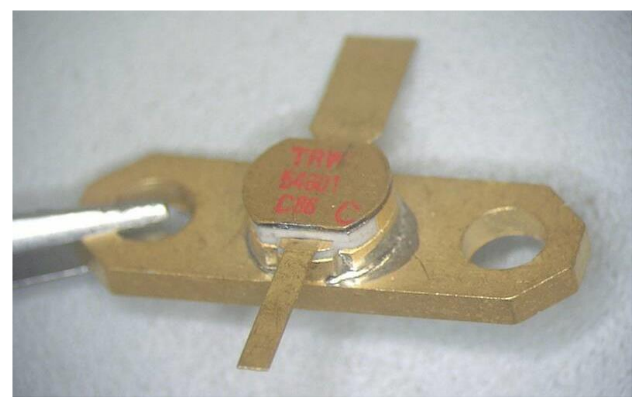




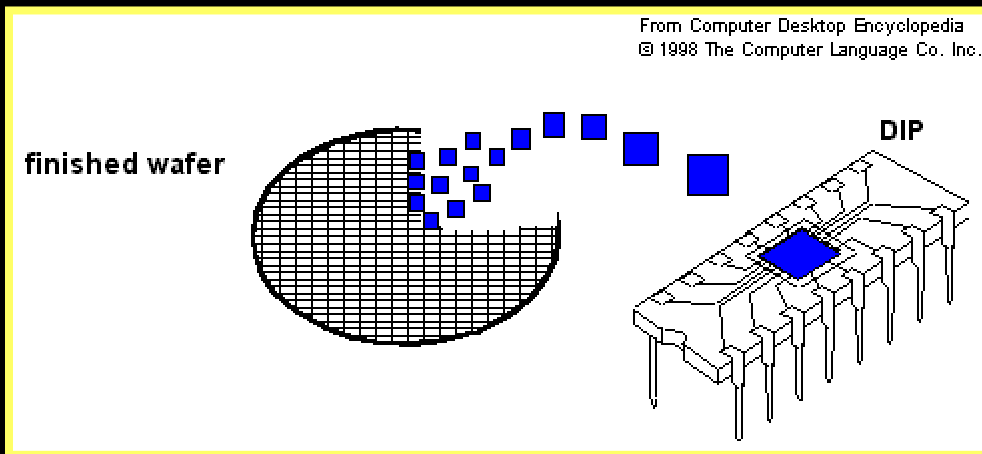
Transistor

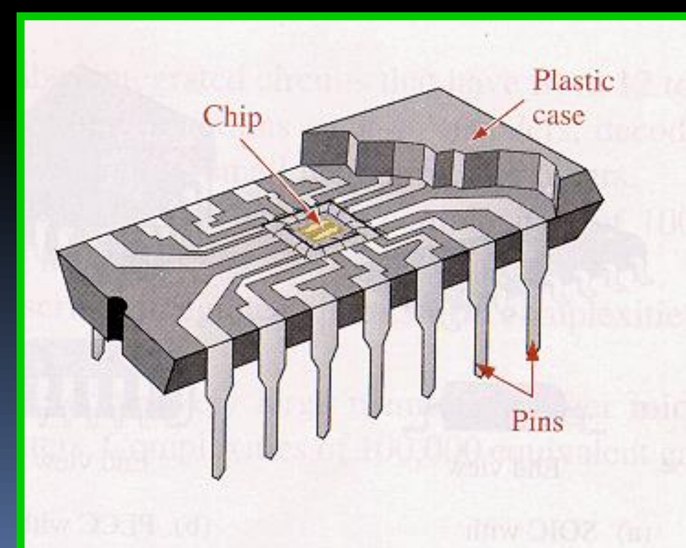

source
gate
Drain


NMOS Transistor
(n-channel MOSFET)
Silicon Dioxide
(insulator)
metal tracks
source
gate
drain
gate
electrode
n-type
silicon
p-type silicon
n-channel
Silicon Substrate

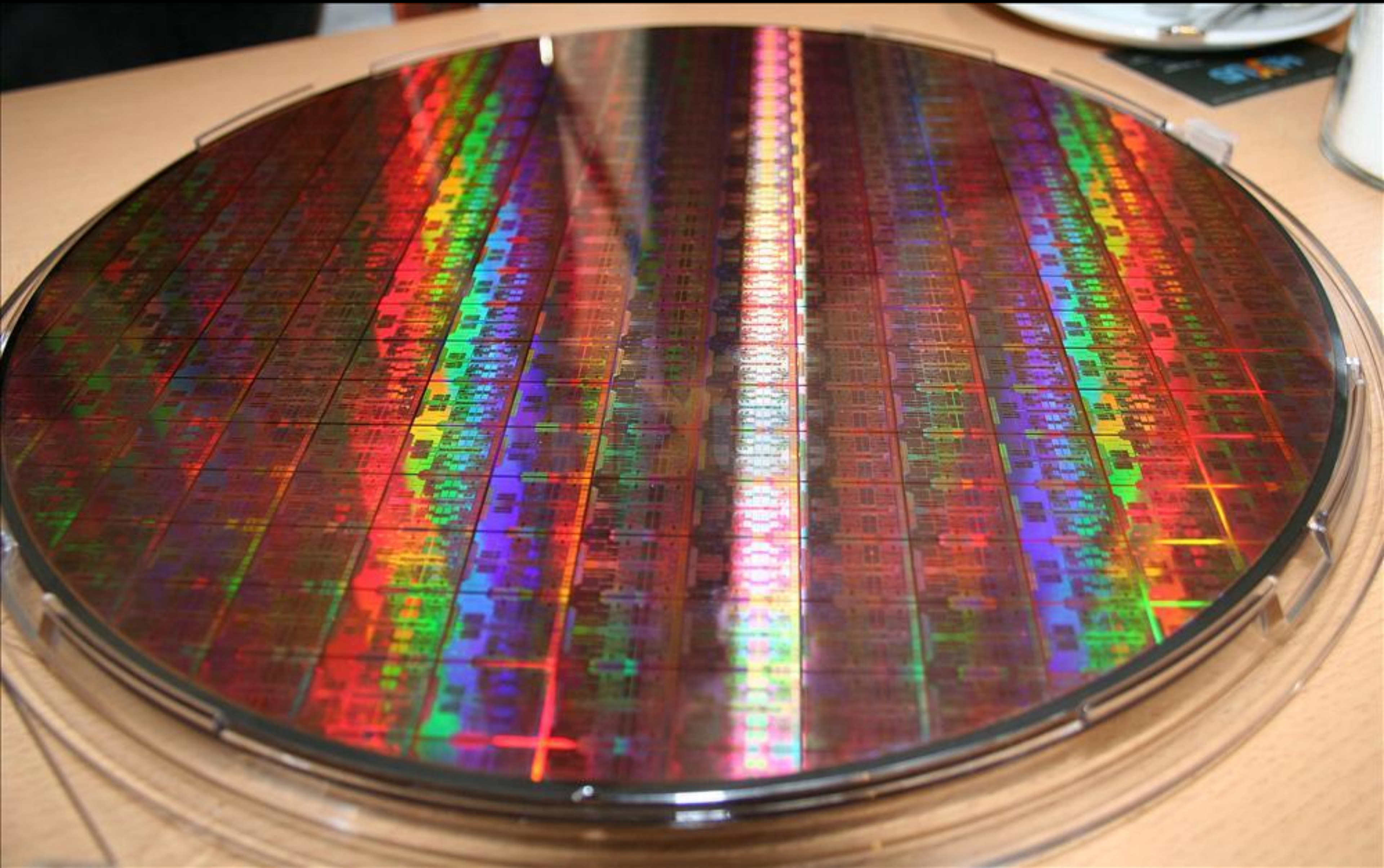


Chip + housing

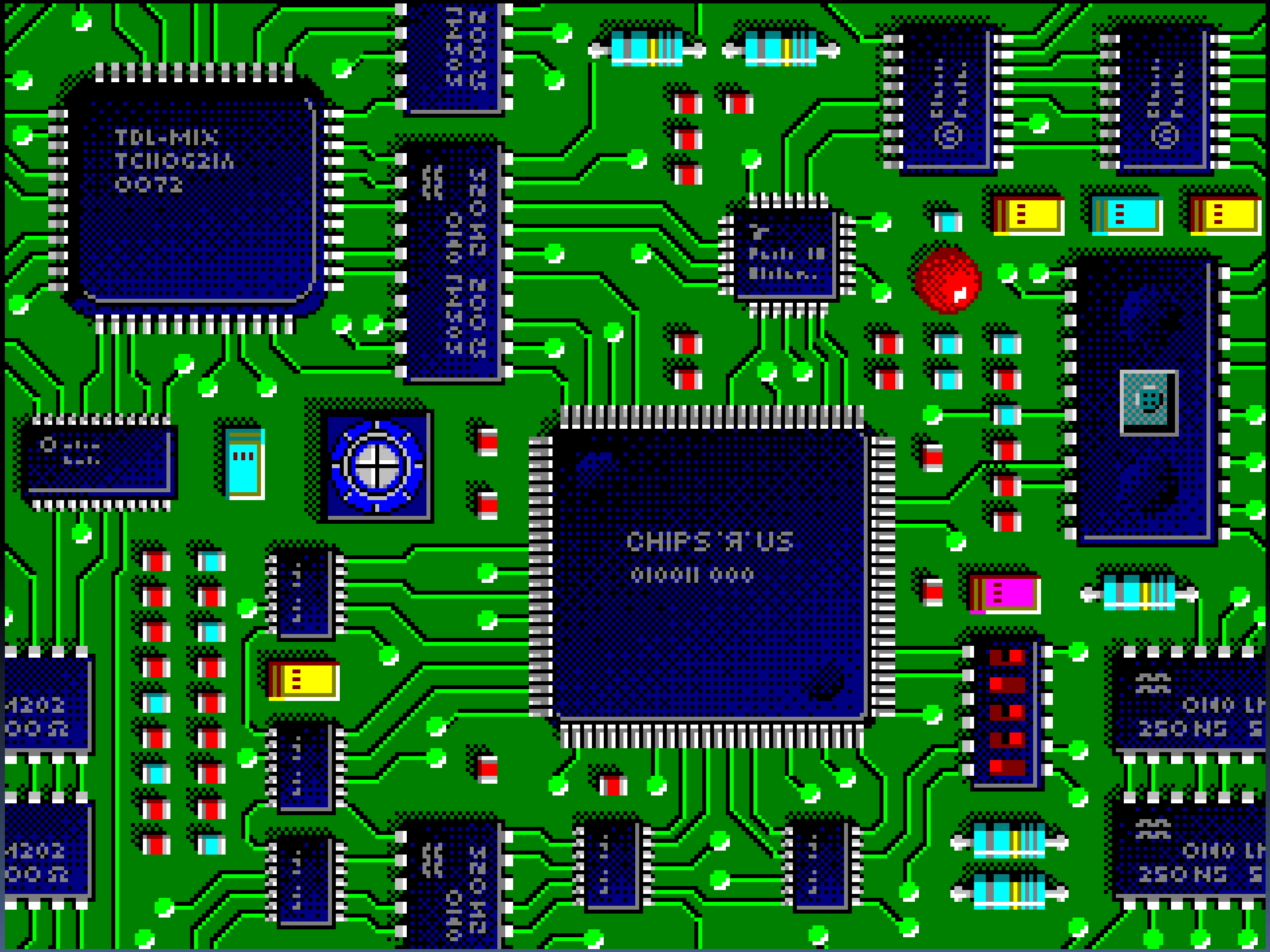




Finished wafer

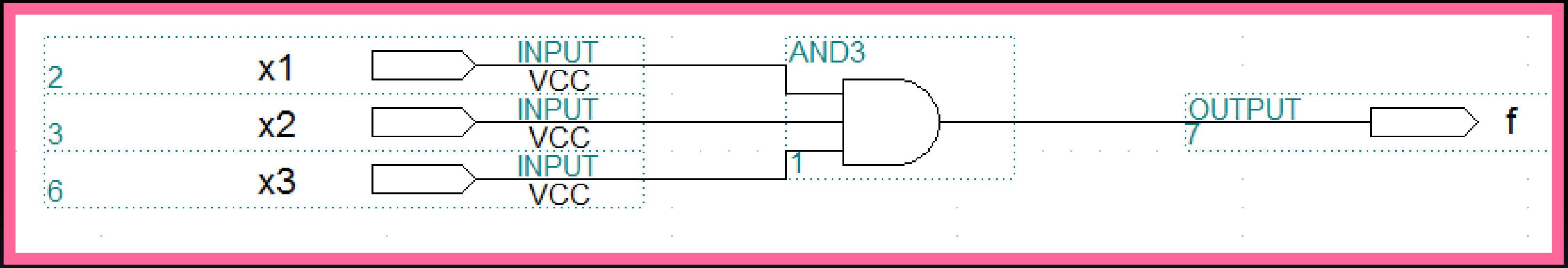


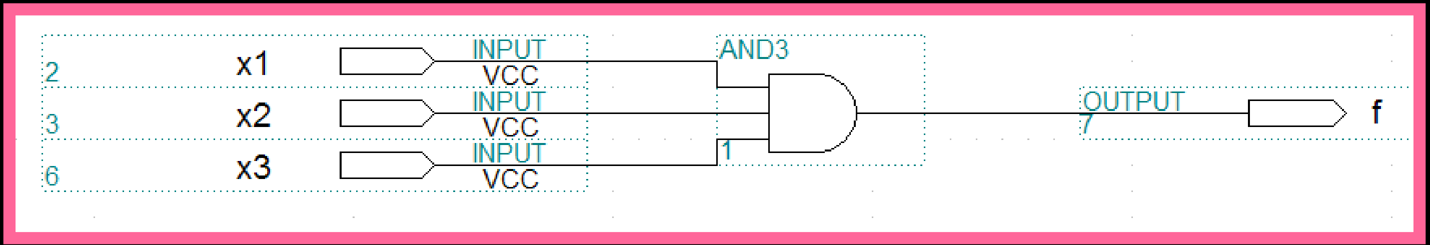
Circuit Board



VHDL & AND gate

AND gate

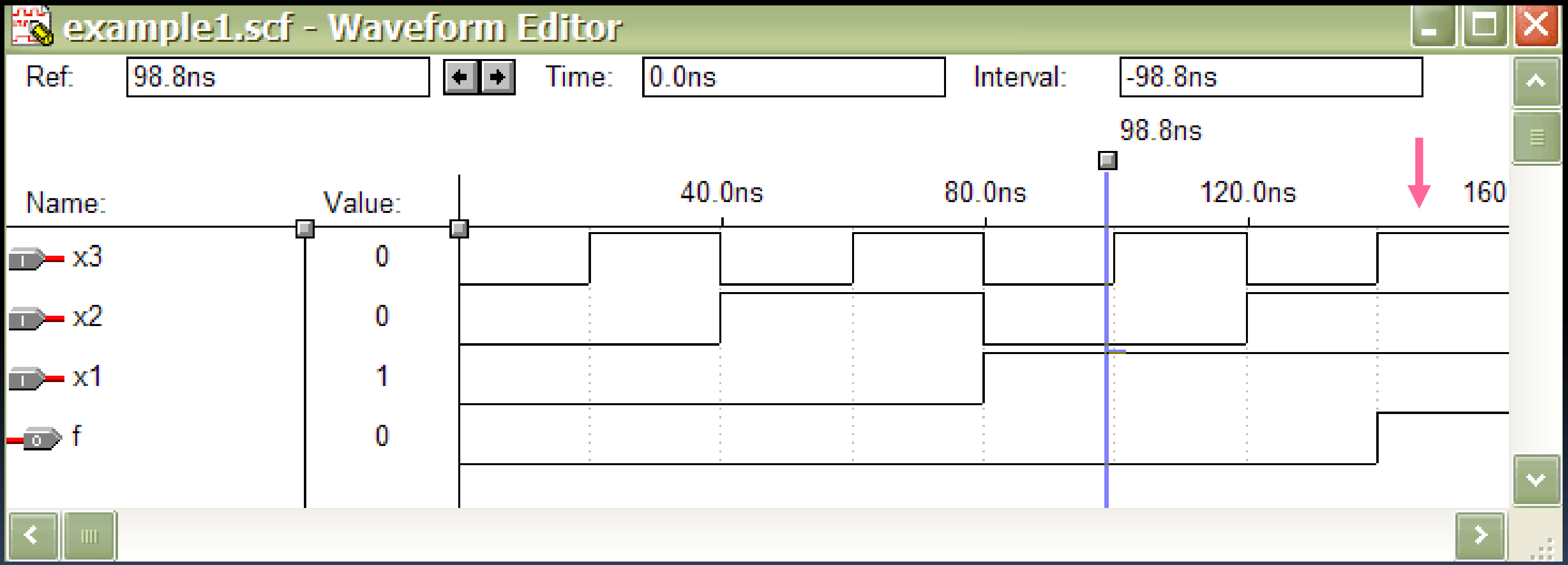




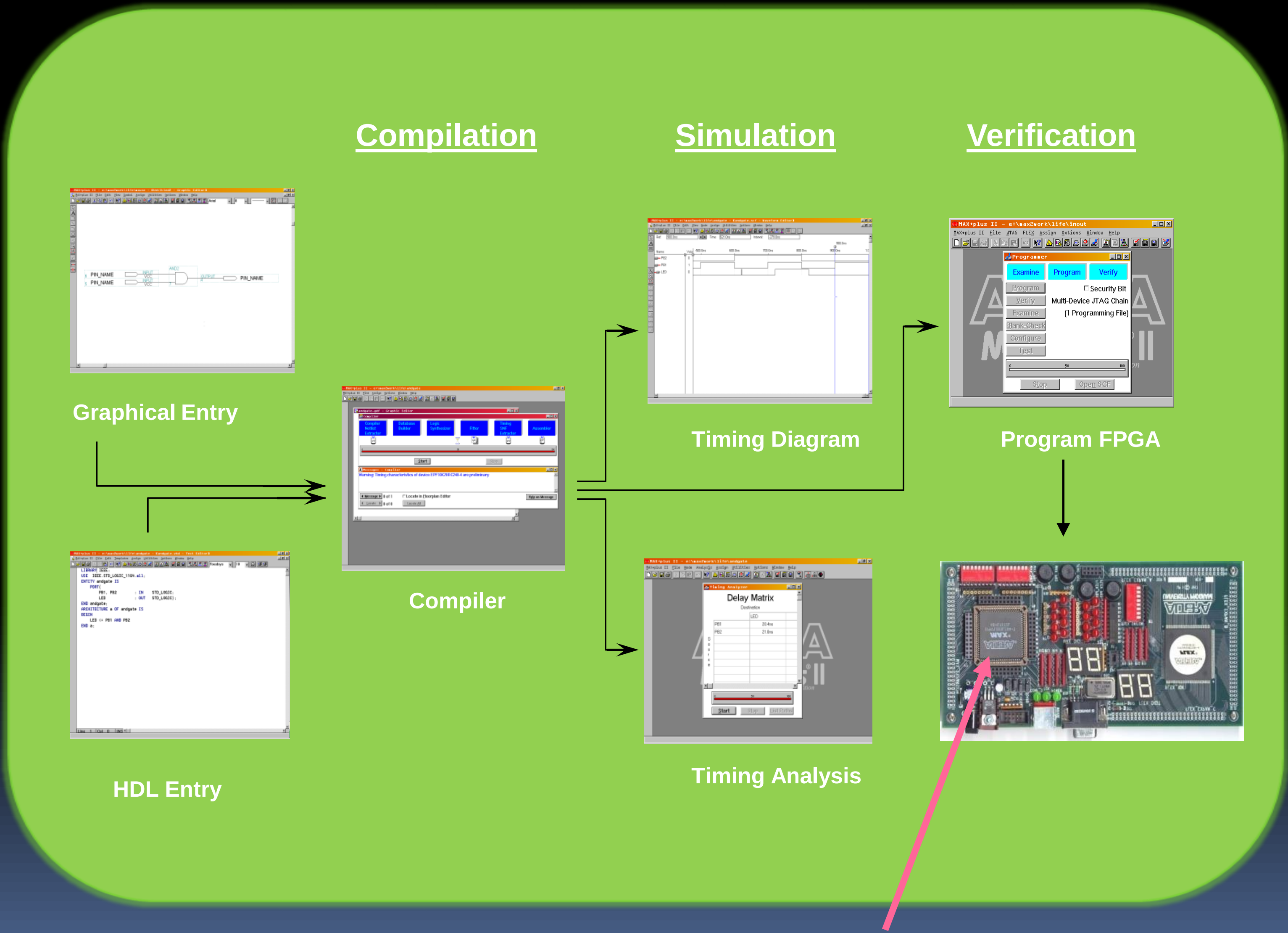

```
ENTITY example1 IS
        PORT ( x1, x2, x3    :        IN BIT;
                       f     :        OUT BIT ) ;
END example1 ;

ARCHITECTURE LogicFunc OF example1 IS
BEGIN
        f <= (x1 AND x2 AND x3);
END LogicFunc ;
```

VHDL code; AND gate

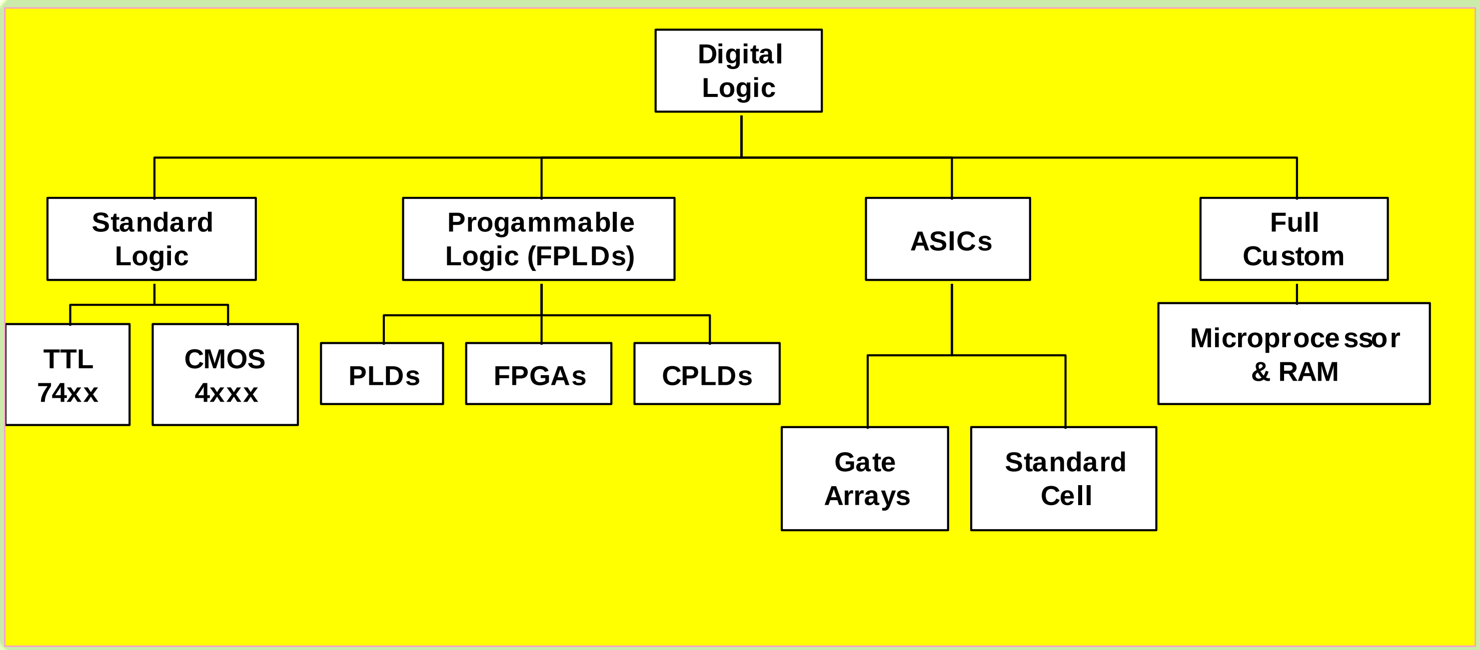
VHDL Simulation; AND gate



FPGA; Create a “working chip”



Digital technology today…



Number systems

- Decimal
- Binary
- Octal
- Hexadecimal

Binary

- Binary numbers: $base_2$

  { 0, 1}

0/1 are Binary digits

Bi; derives from the Latin stem bi, meaning 2

Notes: Binary, Bit

- Binary = 2

"There are only 10 types of people in the world those that understand binary and those that don't!"

Notes: Binary, Bit, Digit

- Binary = 2
- Bit = Binary digit

"There are only 10 types of people in the world those that understand binary and those that don't!"

Notes: Binary, Bit, Digit

- Binary = 2
- Bit = Binary digit
- Digitus = fingers in Latin

↓

Digital Systems

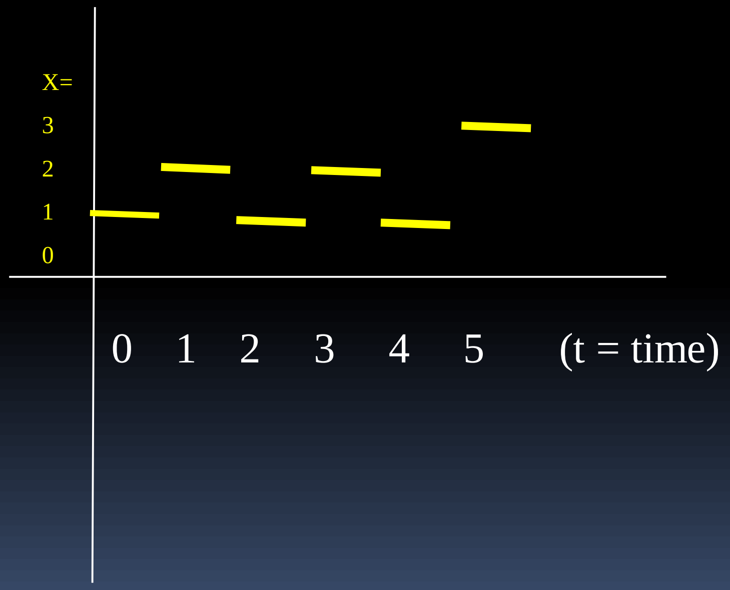
"There are only 10 types of people in the world those that understand binary and those that don't!"

Digital System

- The inputs and the outputs of a digital system are defined over a finite number of discrete values.....

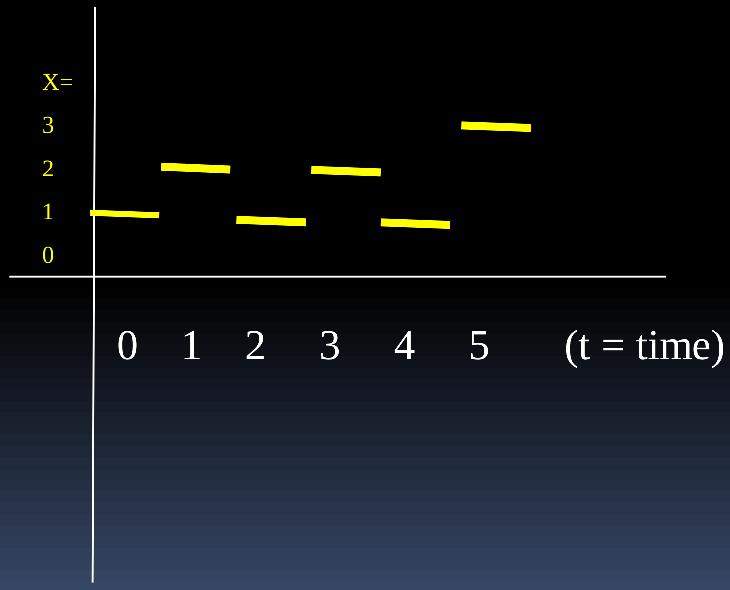
Discrete values…

- The inputs and the outputs of a digital system are defined over a finite number of discrete values…..



Discrete values…

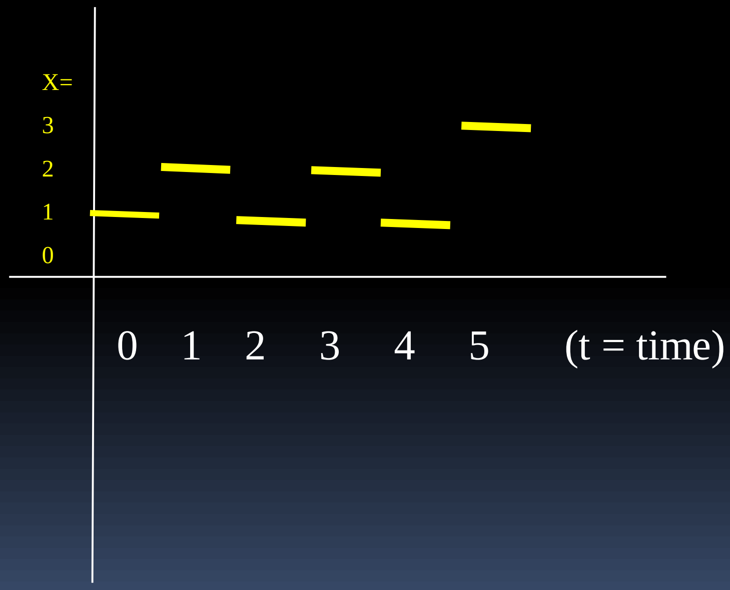
- The inputs and the outputs of a digital system are defined over a finite number of discrete values…..



t:	0	1	2	3	4	5
x:						

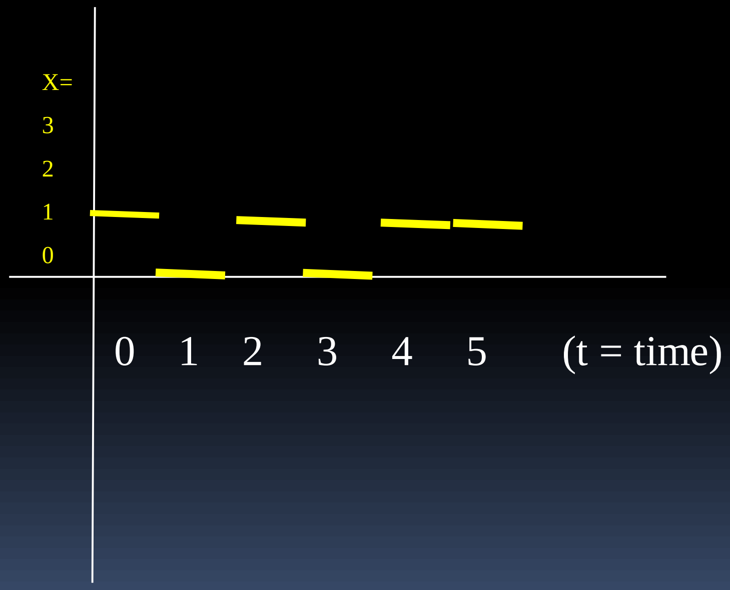
Discrete values…

- The inputs and the outputs of a digital system are defined over a finite number of discrete values…..



t:	0	1	2	3	4	5
x:	1	2	1	2	1	3

Binary values … are discrete



t:	0	1	2	3	4	5
x:						

Binary values … are discrete

t:	0	1	2	3	4	5
x:	1	0	1	0	1	1

Binary (2 discrete values) logic is the basis in computing technology

- In computing we also use the:
  - Octal system
  - Hexadecimal system
- Why do we use Octal and Hex systems?

Why Octal and Hex ?

- Compact way of displaying or writing binary numbers
- Used by computer scientists (computer networks, programming, hardware-memory) to represent binary numbers in a compact form …

Binary system

- Binary numbers: $base_2$
  → { 0, 1}

Octal system

- Binary numbers: $base_2$

  ➡ { 0, 1}

Greek stem; octa

- Octal numbers: $base_8$

  ➡ {0 1 2 3 4 5 6 7}

Hexadecimal system

- Binary numbers: $base_2$

  → { 0, 1}

Greek stems; octa

- Octal numbers: $base_8$

  → {0 1 2 3 4 5 6 7}

and

hexadeca

- Hexadecimal numbers: $base_{16}$

  → {0 1 2 3 4 5 6 7 8 9 A B C D E F}

Conversion between systems

$(?)_{10} \leftrightarrow (?)_{2} \leftrightarrow (?)_{8} \leftrightarrow (?)_{16}$

Decimal number

- $(953.78)_{10}$

Decimal number: analysis

- $base_{10} \rightarrow base_{10}$
  - $(953.78)_{10}$

    $= 900+50+3+0.7+0.08$

Decimal number: analysis

- $base_{10} \rightarrow base_{10}$
  - $(953.78)_{10}$

    = 900 +50 +3 +0.7 +0.08

    = 9*100+5*10+3*1+7*0.1+8*0.01

Powers of 10

- $base_{10} \rightarrow base_{10}$
  - $(9 \quad 5 \quad 3 \quad .7 \quad 8)_{10}$

$= 900 \quad +50 \quad +3 \quad +0.7 \quad +0.08$

$= 9*100 \quad +5*10 \quad +3*1 \quad +7*0.1 \quad +8*0.01$

= 9*10^2+5*10^1+3*10^0+7*10^{-1} + 8*10^{-2}

We multiply each digit by the appropriate power of 10,

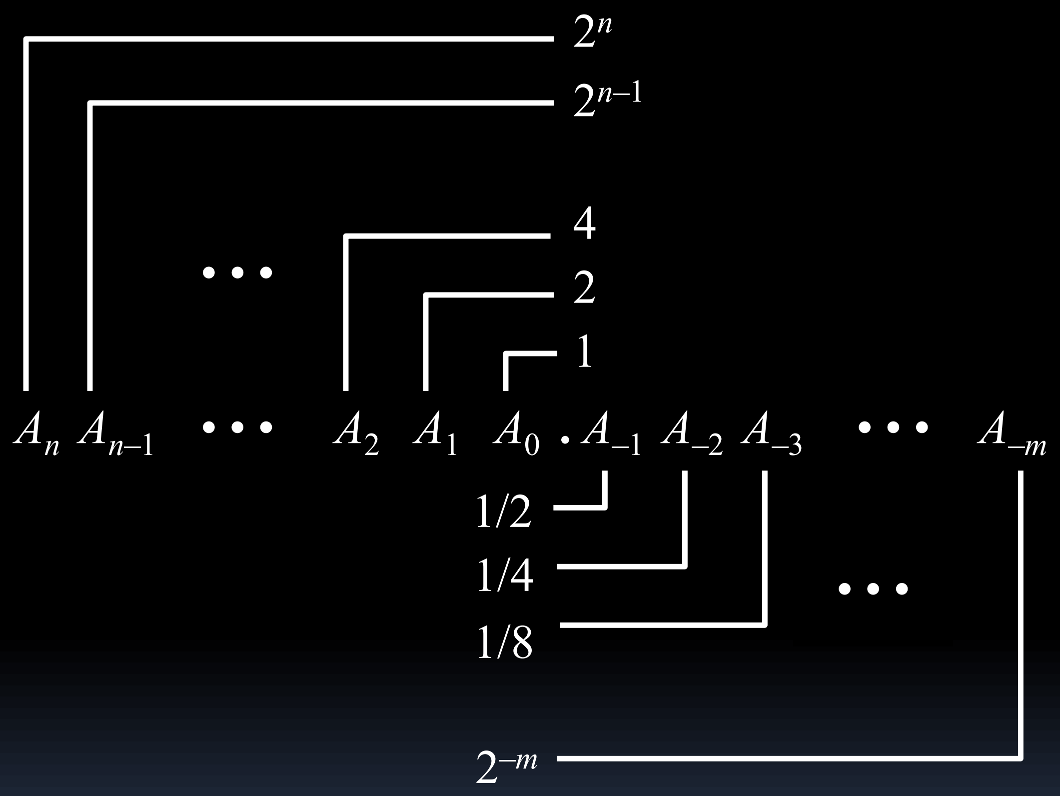
A number … positional notation

$(A_n \ldots A_2 A_1 A_0 \,.\, A_{-1} A_{-2} \ldots A_{-m})_r$

Integer Radix Point Fraction

r is the base

In polynomial form … r = 2



$$\sum_{i=-m}^{n} A_i \cdot r^i$$

Binary: Base-2

- $base_{10} \rightarrow base_{10}$
  - $(953.78)_{10}$

    $= 900 \quad +50 \quad +3 \quad +0.7 \quad +0.08$

    $= 9*100 \quad +5*10 \quad +3*1 \quad +7*0.1 \quad +8*0.01$

    $= 9*10^2+5*10^1+3*10^0+7*10^{-1} + 8*10^{-2}$
- Convert: From $base_2 \rightarrow base_{10}$
  - $(1\,0\,1\,1\,.\,1\,1)_2$

Base-2 to Base-10

- $base_{10} \rightarrow base_{10}$
  - $(953.78)_{10}$

    = 900 +50 +3 +0.7 +0.08

    = 9*100 +5*10 +3*1 +7*0.1 +8*0.01

    = 9*10^2+5*10^1+3*10^0+7*10^{-1} + 8*10^{-2}
- Convert: From $base_2 \rightarrow base_{10}$
  - $(1\,0\,1\,1\,.\,1\,1)_2$

    = 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1} +1*2^{-2}

    = 8 + 0 + 2 + 1 + 0.5 + 0.25

    = $(11.75)_{10}$

We multiply each digit by the appropriate power of 2,

Another example

- $(1\,1\,0\,1\,0)_2$
  $= 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$
  $= 16 + 8 + 0 + 2 + 0$
  $= (26)_{10}$

Base-8 to Base-10

- $Base_8 \rightarrow Base_{10}$
  - $(1\,4\,7\,.\,3)_8 =$

Base-8 to Base-10

- $Base_8 \rightarrow Base_{10}$
  - $(1\ 4\ 7\ .\ 3)_8 = 1 * 8^2 + 4 * 8^1 + 7 * 8^0 + 3 * 8^{-1}$
    $= (103.375)_{10}$
- $Base_{16} \rightarrow Base_{10}$
  - $(B\ 6\ 5\ F)_{16} =$

Base-16 to Base-10

- $Base_{16} \rightarrow Base_{10}$
  - $(B\ 6\ 5\ F)_{16} = 11*16^3 + 6*16^2 + 5*16^1 + 15*16^0$
    $= 45056 + 1536 + 80 + 15$
    $= (46687)_{10}$

Base-10 ⇨ Base-2

- $(2\ 6)_{10} = (?)_2$

Use division

26/2 =   ?

Base-10 ⇨ Base-2

- $(2\ 6)_{10} = (?)_2$
- Division
  - 26 / 2 = 13 0/2
  - 13 / 2 = 6 1/2

Base-10 ⇨ Base-2

- $(2\ 6)_{10} = (?)_2$
- Division
  - 26 / 2 = 13 0/2
  - 13 / 2 = 6 1/2
  - 6 / 2 = 3 0/2
  - 3 / 2 = 1 1/2
  - 1 / 2 = 0 1/2

Base-10 ⇨ Base-2

- $(2\ 6)_{10} = (?)_2$
- Division
  - 26 / 2 = 13 0/2
  - 13 / 2 = 6 1/2
  - 6 / 2 = 3 0/2
  - 3 / 2 = 1 1/2
  - 1 / 2 = 0 1/2

Write remainders of division above (bottom to top)

→ $(1\ 1\ 0\ 1\ 0)_2$

Another way: Using weights

- Weights:

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Example:

$(1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)_2 = (?)_{10}$

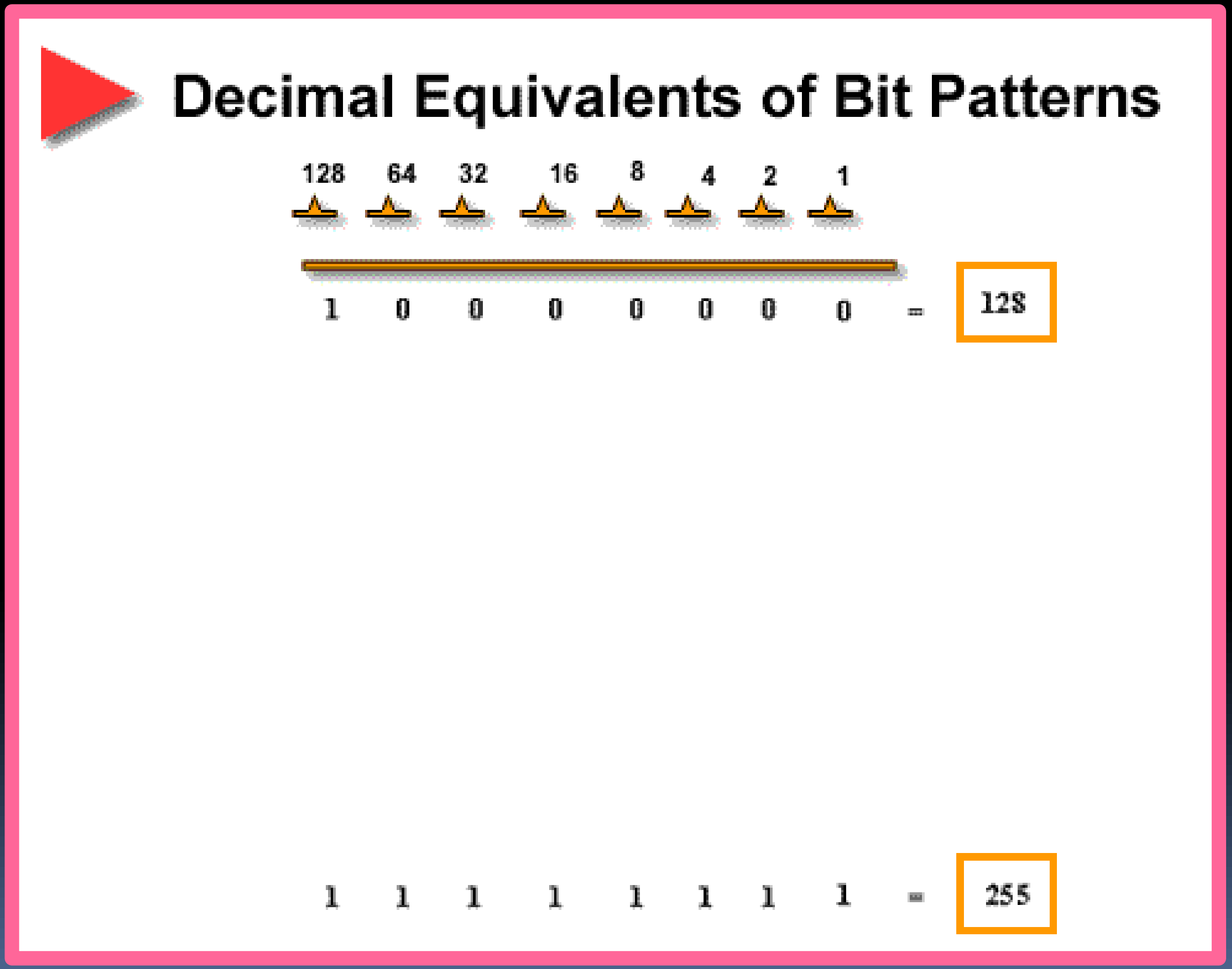
Example: Using weights

- Weights:

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
1	0	0	1	1	1	0	0	1
256			+32	+16	+8			+1

$= (313)_{10}$

Example: Using weights



Another example

- $(625)_{10} \rightarrow (?)_2$

Another example

- $(625)_{10} \rightarrow (?)_2$
- 625/2 = 312 1/2
- 312/2 = 156 0/2
- 156/2 = 78 0/2
- 78/2 = 39 0/2
- 39/2 = 19 1/2
- 19/2 = 9 1/2
- 9/2 = 4 1/2
- 4/2 = 2 0/2
- 2/2 = 1 0/2
- 1/2 = 0 1/2

= 1001110001

Conversion from different bases

- $base_{10} \longrightarrow base_8$
- $(\ 1\ 5\ 3\ )_{10} = (?)_8$

Base-10 to Base-8

- $base_{10} \rightarrow base_8$
- $(1\ 5\ 3)_{10} = (?)_8$
  - 153 / 8 = 19 1/8
  - 19 / 8 = 2 3/8
  - 2 / 8 = 0 2/8

$= (231)_8$

Base-10 to Base-16

- $base_{10} \rightarrow base_{16}$
- $(2\ 4\ 5)_{10} = (\ ?\ )_{16}$

Base-10 to Base-16

- $base_{10} \rightarrow base_{16}$
- $(\,2\;4\;5)_{10} = (\;?\;)_{16}$
  - $245 / 16 = 15 \quad 5/16$
  - $15 / 16 = 0 \quad 15/16$

  $= (F5)_{16}$

Base-10 ⇨ Base-2

- $(0\,.\,6\,8\,7\,5)_{10} = (?)_2$
- Lets try to solve this problem on your own.................
- **You have … 3 minutes**

Base-10 ⇨ Base-2

- $(0.6875)_{10} = (?)_2$
  - $0.6875 * 2 = 1.3750$
  - $0.3750 * 2 = 0.7500$
  - $0.750 * 2 = 1.5000$
  - $0.500 * 2 = 1.0000$
  - $0.000 * 2 = 0.0000$

  $= (0.10110)_2$

Note:

- The conversion of decimal numbers with both integer and fraction parts is done by converting the integer and the fraction and then combining the two answers.
- Example:
  - $(41.6875)_{10} \rightarrow (?)_2$

$(41)_{10} = (101001)_2$ (STEP -1)

$(0.6875)_{10} = (0.10110)_2$ (STEP-2)

$= (101001.10110)_2$

Conversion from binary to octal

$(010110001101011.111100000010)_2 = (?)_8$

Conversion from binary to octal

(010 110 001 101 011 . 111 100 000 010)$_2$ = (?)$_8$

A

- Step 1: Start from A, go left ⟹ 3 digits
- Step 2: Start from A, go right ⟹ 3 digits
  - 011 = 3   111 = 7
  - 101 = 5   100 = 4
  - 001 = 1   000 = 0
  - 110 = 6   010 = 2
  - 010 = 2

  = (26153.7402)$_8$

Conversion from binary to hex

$$(10110001101011.11110010)_2 = (?)_{16}$$

Conversion from binary to hex

- $(10\ 1100\ 0110\ 1011\ .\ 1111\ 0010)_2 = (?)_{16}$

  B

- Step 1: Start from B, go left → 4 digits
- Step 2: Start from B, go right → 4 digits
  - 1011 = B 1111 = F
  - 0110 = 6 0010 = 2
  - 1100 = C
  - 0010 = 2

  $= (2C6B.F2)_{16}$

Conversion from hex to binary

- $(306.D)_{16} = (?)_2$

Conversion from hex to binary

- $(306.D)_{16} = (?)_2$

$= (0011\ 0000\ 0110\ .\ 1101)_2$

END1

Conclusion

- Converting from base "r" to base 10, form the appropriate sum of powers in base "r"
- Converting from a decimal integer to base "r", divide by "r", use the remainders as coefficients
- Converting a decimal fraction to base "r", multiply by "r", use the integers as coefficients
- Converting octal and hex to binary, form groups of 3 or 4 digits

END1+

Application: computer networks

Dotted Decimal Notation … http://

http://csam.montclair.edu

- Dotted-decimal: http://130.68.20.55

Internet addresses: URL & IP

- A website address is represented as a URL (text based), for example:
  - www.montclair.edu
- There is also a numerical representation for the text based website address, called an Internet Protocol address (IP address), for example:
  - www.montclair.edu ➔ 130.68.1.10
- More examples:
  - cs.montclair.edu ➔ 130.68.20.101
  - csam.montclair.edu ➔ 130.68.20.108
- The text based website address is much easier for us to remember than an IP address
  - Telephone numbers, for instance:
    - 1-800-flowers ➔ 1-800-356-9377
- The text is **much easier** to remember…

Internet addresses: IP address…

- Are unique…
- Universal…
- Used also for routing purposes…

Internet addresses: IP

- Dotted-decimal: 130.68.1.12
- Binary: 4 byte address with total 32 bits

Let us convert 130.68.1.12 into Binary, that has 32 bits (4 x 8)…

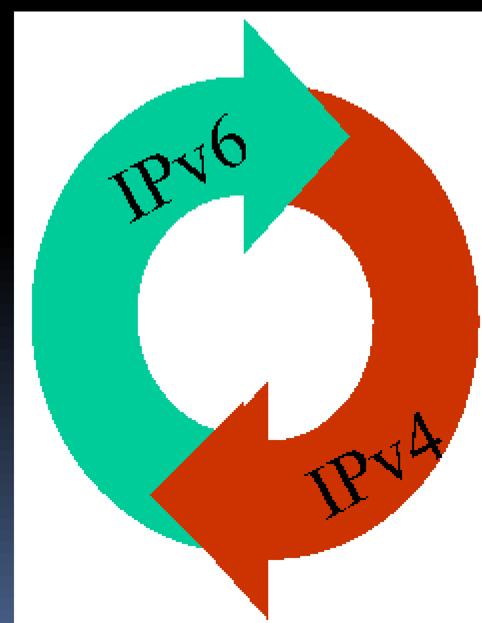
dotted-decimal & binary

- Dotted-decimal: 130 . 68 . 1 . 12
- Binary:

10000010 | 01000100 | 00000001 | 00001100

2^{32} Different addresses

IPv4 → IPv6



IPv4 → IPv6

END