AN INTRODUCTION TO DIGITAL LOGIC HARDWARE *

by

MIGUEL I. ESCOTO JR.*

Brief History

* Started when man first learned to count—associate number names with objects in a group
* 4000 to 3000 B.C. use of the abacus.
  Beads were called calculi by the Romans
* Led to arithmetic and all kinds of calculating devices
  Napier's bones (first slide rule)
  Pascal's calculator (first adding machine) 1642
  Leibnig (first multiplying machine) 1671
* 1801 Jaquard invented 1st automatic loom main feature was to use punched cards where needle passed through holes and stitched a pattern onto the cloth.
* 1833 Babbage visualized the 1st computer a machine that used punched cards to carry out arithmetic calculations automatically.
* 1854 Boole found a new way of thinking, a new way to reason things out. His symbol logic is called Boolean Algebra
* 1950 first electronic computer based on Babbage idea appeared
  1st generation used vacuum tubes
  2nd generation used transistors (late 50's)
  3rd generation used transistor and integrated circuits
  4th generation extensive use of integrated circuits

DIGITAL vs. ANALOG

A digital source can produce only a finite set of discrete symbols. We say that the alphabet size is countable, i.e., finite.

Signals which are digital are restricted to known set of values.

Examples:

A teletype machine — digital	Current flow } analog
Voice conversation	Light intensity } analog

KEY CONCEPTS

Finite vs. infinite
Continously variable vs. discrete
Estimation vs. decision
Signal set vs. signal space

* Instructor in Electrical Engineering Department, College of Engineering, University of the Philippines.

DIGITAL LOGIC rests on Boolean Algebra — a systematic way symbols are used for logic decisions. A logic circuit is thus are which follows Boolean Algebra.

The order to simplify circuit design and improve reliability, the BINARY NUMBER SYSTEM is generally employed in digital equipment.

For example in electronic digital instruments, the digits are usually represented by different potentials. If the decimal system were used, the circuits would have to be capable of differentiating accurately between ten levels representing the various digits. This of course is possible with careful circuit design but the circuits would be rather complex and the chance of errors is quite high.

*A Byte is an eight bit word
*The weight of each bit position rests as 2^x, where x is the bit position

Examples·

Ozeth, 1st, 2nd, 3rd — bit position

$$(1010)_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + 0 \times 2^0$$

base 2 (Binary) $= (10)_{10}$ ← Base 10 (Decimal)

-4, -3, -2, -1, 0 — bit position

$$(1.1101)_2 = (1 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2}) + (0 \times 2^{-3}) + (1 \times 2^{-4})$$

Binary Pt. ↗

$$= 1 + 1/2 + 1/4 + 0 + 1/16 = (1.8125)_{10}$$

THE BINARY SYSTEM

* 2 digits are used 1 and Ø
* Basic unit of information is the Bit which may assume either a 1 or a Ø.
* Combination of bits form a word thus
  1010 is a four bit word
  10100101 is an eight bit word

CONVERSION OF NUMBER SYSTEMS

Binary/Octal to decimal

Rules:

- multiply the most significant (octal) digit by 8
- add to this product the value of the next significant digit and multiply by 8
- repeat this up to the least significant digit (LSD)
- add the value of the LSD to the last product
- the result of this is the required decimal number

Example:

Binary: 111/011/101/001

Octal : 7 3 5 1

Conversion :

$$\{[(7\times 8) + 3]\, 8 + 5\}\, 8 + 1$$

$$56 + 3 = 59$$

$$59 \times 8 = 472; \quad 472 + 5 = 477$$

$$477 \times 8 = 3816; \quad 3816 + 1 = (3817)_{10}$$

$$(7351)_8 = (111\ 011\ 101\ 001)_2 = (3817)_{10}$$

Base 8 (Octal) — Base 2 (Binary) — Base 10 (Decimal)

DECIMAL TO OCTAL/BINARY

Rules:

1. Divide the decimal number by 8 and write down the remainder (r_1)
2. Divide the quotient of the preceeding stage by 8 again and write down the remainder (r_2)
3. Repeat step 2 until the quotient is 0. The required # is then r_n, r_{n-1}, r_{n-2} .... $r_2 r_1$

Example: Decimal $(3817)_{10}$

Conversion

```
3817 ÷ 8 = 477 ÷ 8 = 59 ÷ 8 = 7 ÷ 8 = 0
32           40         56       0
 61           77        3=r3     7=r4
 56           72
  57          5=r2
  56
   1=r1
```

$$(3817)_{10} = (7351)_8 = (111\ 011\ 101\ 001)_2$$

Boolean Algebra

A summary

Laws:

$\overline{0} = 1$	$x + 0 = x$	$0 \cdot x = 0$
$1 = 0$	$x + 1 = 1$	$x \cdot 1 = x$
$\overline{\overline{x}} = x$	$x + x = x$	
	$x + \overline{x} = 1$	$x \cdot \overline{x} = 0$

Commutative Laws:

$xy = yx$ $\qquad x + y = y + x$

Associative Laws:

$xyz = x(yz) = z\,(xy)$

$x + y + z = x + (y + z) = y + (x + z) = z + (x + y)$

Distributive Laws:

$x(y + z) = xy + xz$ $\qquad (x + y)\,(x + z) = x + yz$

Absorption Laws:

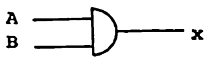
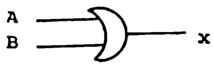
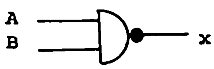
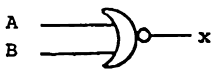
$x + xy = x$ $\qquad x + \overline{x}y = x + y$ $\qquad x(\overline{x} + y) = xy$

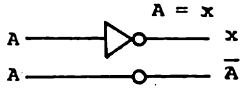
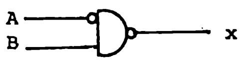
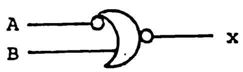
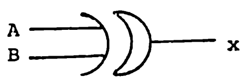
De Morgan's Laws:

$$\overline{x + y + z + \ldots N} = \overline{x} \cdot \overline{y} \cdot \overline{z} \ldots \overline{N}$$

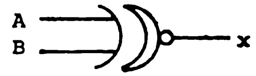
$$\overline{x \cdot y \cdot z \ldots N} = \overline{x} + \overline{y} + \overline{z} + \ldots \overline{N}$$

STANDARD SYMBOLS FOR LOGIC ELEMENTS

CIRCUIT	AMERICAN STANDARD	BOOLEAN FUNCTION
AND		$x = A \bullet B$
OR		$x = A + B$
NAND		$x = \overline{A \cdot B}$
NOR		$x = \overline{A + B}$

CIRCUIT	AMERICAN	BOOLEAN FUNCTION
NOT	$\bar{A} = x$  Buffer inverter inverter	$A \nvDash \bar{A} = x$
NAND with one inverting input		$x = \overline{\bar{A} \quad B}$
NOR with one inverting input		$x = \overline{\bar{A} + B}$
EXCLUSIVE OR		$x = A + B = A\bar{B}$ $+ A\bar{B}$

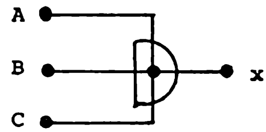
EXCLUSIVE OR
use as comparator



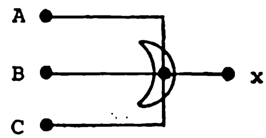
$x = \overline{A + B}$

$= AB + \overline{AB}$

DISTRIBUTED or WIRED AND

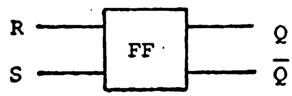


DISTRIBUTED or WIRED OR



DELAY

FLIP-FLOP (RS)



LOGIC FAMILIES

Family	*Type*	*Technology*	*Initial*
* Diode logic	non-saturating	Bi-polar	D.L.
* Resistor-transistor logic	saturating	Bi-polar	R.T.L.
* Diode Transistor logic	saturating	Bi-polar	D.T.L.
* Transistor-transistor logic	saturating	Bi-polar	TTL
Sub-Families:			
Normal	saturating	Bi-polar	(TTL)SN
Low power	saturating	Bi-polar	(TTL)L
Low power Schottkv	saturating	Bi-polar	(TTL)LS
High threshold logic	saturating	Bi-polar	(TTL)H.T.L.
Open collector	saturating	Bi-polar	(TTL)O.C.
* Emitter coupled logic	non-saturating	Bi-polar	E.C.L.

* Integrated injection logic	saturating	Bi-polar	I^2L
* N channel MOS	saturating	metal oxide semiconductor	N-MOS
P channel MOS	saturating	(MOSFET)	P-MOS
Complementary MOS	saturating		CMOS
Silicon on sappline	saturating	MOS	SOS MOS

EXAMPLES IN DIGITAL SYSTEMS:

* Design of simple "Water Level indicator" combinational logic
* Design of a simple "Kilometer per liter Computer" sequential logic
* Numerical method for contouring (Position and feed rate control)
* A/D, D/A conversion logic circuits
    simultaneous conversion (clockless)
    counter-method
    continous conversion
    successive approximation

Others dual slope integration
    pulse width modulation
    ramp and comparator method

* Parity Bit generation and detection
* Other logical circuit tricks using propagation delay.

REFERENCES:

* Digital Instrument Course by A. J. Bouwen Philips (c) 1971
  Part 1 Basic Binary Theory and Logic Circuits
* Digital Principles and Applications by Albert Paul Malvino &
  McGraw Hill (c) 1975 Donald P. Leach
* Engineering Electronics with Industrial Applications and Control
  by John D. Ryder (c) 1957 McGraw Hill
* Digital Computer Fundamentals by Thomas Bartel 3rd Edition
  McGraw Hill (c) 1972
* Digital Electronics by Millman and Taub McGraw Hill