

EL253 - Sistemas Digitales

Semestre 2020-2

Profesor Kalun José Lau Gan

Sesión de Teoría Semana 1

1

Preguntas previas

2

Agenda

- Revisión de circuitos digitales
- Jerarquía en el diseño digital
- El VHDL
- Características del VHDL y sus estándares
- Estructura de un código en VHDL

3

Revisión de circuitos digitales

- Sistemas numéricos: binario, octal, decimal y hexadecimal

Ej: Convertir $13_{DEC} \rightarrow$ octal

Procedimiento: Llevar todo número a binario

Tabla de pesos:

$1024 \quad 512 \quad 256 \quad 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$
 $0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1$
 Resultado en binario

Para octal agrupamos de tres en tres los dígitos binarios $\Rightarrow 15_{oct}$

Ej: Convertir $4835_{DEC} \rightarrow$ hexadecimal

$4096 \quad 2048 \quad 1024 \quad 512 \quad 256 \quad 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$
 $1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1$
 Resultado en binario
 Para hexadecimal se agrupan de 4 en 4 los dígitos binarios $\Rightarrow 12E3_{HEX}$

Convertir $384_{HEX} \rightarrow$ decimal

\rightarrow pasamos a BIN:

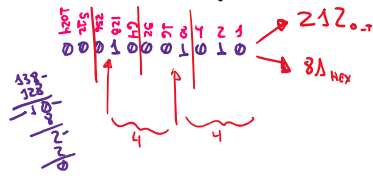
$0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0$
 $512 \quad 256 \quad 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1$
 $512 + 256 + 128 + 64 + 16 + 8 = 900$
 $\Rightarrow 384_{HEX} \rightarrow 900_{DEC}$

4

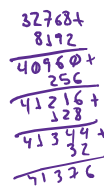
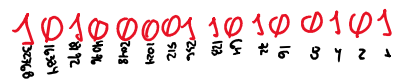
Ejercicios

Convertir $138_{DEC} \rightarrow octal$

1º Pasarlo a binario:

Convertir $A1A5_{HEX} \rightarrow Decimal$

Pasando a binario:

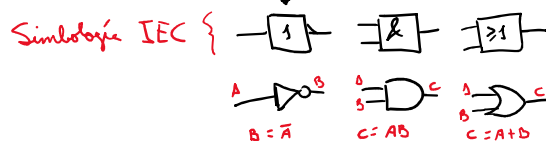
Convertir $33333_{DEC} \rightarrow HEX$ 

5

Revisión de circuitos digitales

• Álgebra de Boole

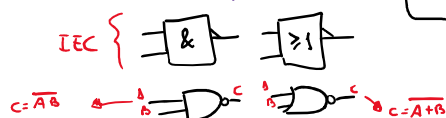
- Operadores básicos: NOT, AND, OR



- Operadores exclusivos: XOR, XNOR



- Operadores universales: NAND, NOR



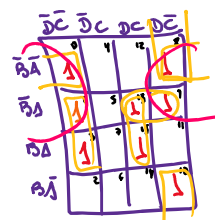
• Leyes del Álgebra de Boole

• Simplificación de funciones

• Mapas de Veitch-Karnaugh

→ Obtener una función lógica a partir de una tabla de verdad.

Ej: $F = \sum_m(0, 1, 3, 8, 9, 10, 13, 15)$



$$F_{DCBA} = \bar{C}\bar{B} + \bar{D}\bar{C}A + DCA + D\bar{C}\bar{A}$$

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



$$F_{DCBA} = \bar{C}\bar{B} + D\bar{B}A + \bar{D}\bar{C}A + DCA + D\bar{C}\bar{A}$$

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

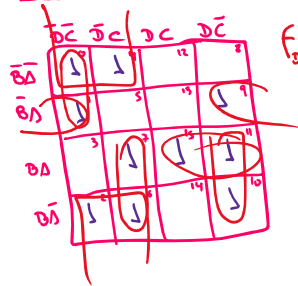
$$= A(\bar{D}\bar{C}) + \bar{A}\bar{C}(\bar{B} + D) + D\bar{B}A$$

6

Ejercicio

- Obtener la función lógica simplificada de lo siguiente:

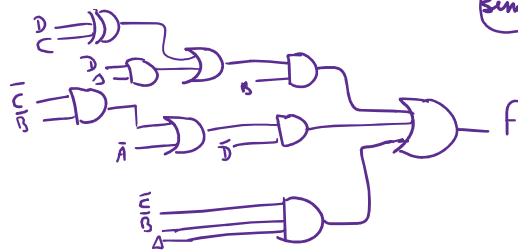
$$f_{DCBA} = \sum_m (0, 1, 2, 4, 6, 7, 9, 10, 11, 15)$$



$$f_{DCBA} = \bar{D}\bar{C}\bar{B} + \bar{D}C\bar{B} + D\bar{C}\bar{B} + \bar{D}\bar{A} + \bar{C}\bar{B}A + DBA$$

$$= B[(D\oplus C) + DA] + \bar{D}(\bar{C}\bar{B} + \bar{A}) + \bar{C}\bar{B}A$$

Luego de obtener la función se debe de aplicar las propiedades de Boole para simplificarla aún más.



7

Revisión de circuitos digitales

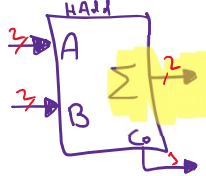
- Circuitos MSI combinacionales
 - Sumadores/Restadores
 - Decodificadores/Codificadores
 - Comparadores de magnitud
 - Multiplexores y demultiplexores
 - ALU

8

Revisión de circuitos digitales

• Sumadores

• Medio



$$\begin{array}{r} \text{BINI} \\ 1 + 10 \\ \hline 10 \\ \text{CR} \end{array} \quad \begin{array}{r} 10 + 1 \\ \hline 011 \\ \text{CR} \end{array}$$

El acarreo solo puede ser "1" o "0" en cualquier base

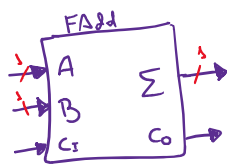
En decimal:

$$\begin{array}{r} 1 + 9 + 99 + \\ \hline 2 \quad 18 \quad 198 \\ \text{CR} \quad \text{CR} \end{array}$$

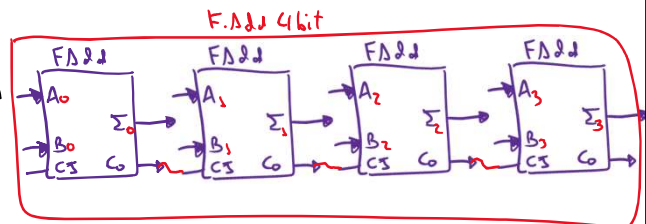
En hexadecimal:

$$\begin{array}{r} 1 + 9 + F + \\ \hline 2 \quad 12 \quad 1E \\ \text{CR} \quad \text{CR} \end{array}$$

• Completo



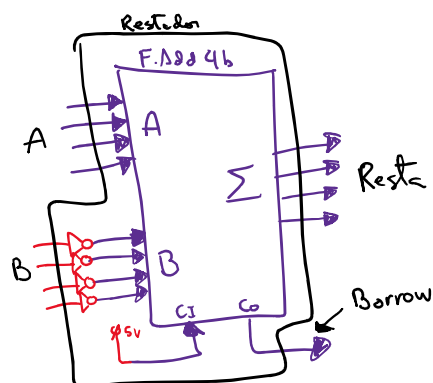
Sirve para hacer sumadores de mayor cantidad de bits con solo conectar en cascade.



9

Revisión de circuitos digitales

• Restador empleando FAdd:

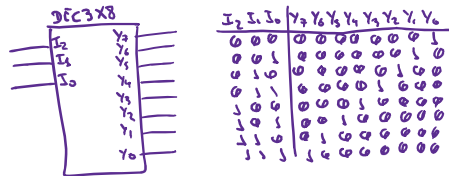


10

Revisión de circuitos digitales

• Decodificadores

• Binarios

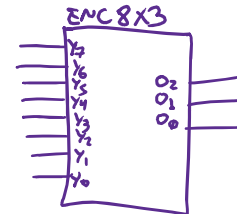


• BCD-7Seg



• Codificadores binarios

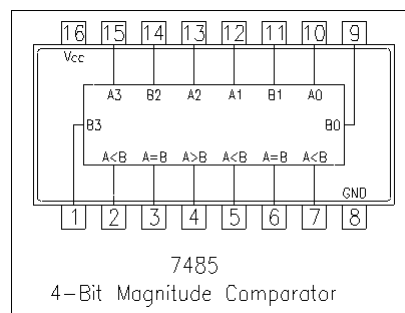
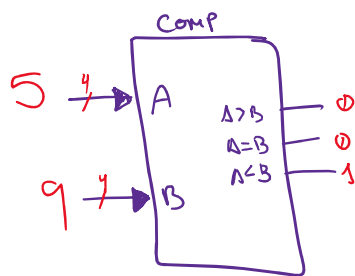
→ La función inversa de un decodificador binario



11

Revisión de circuitos digitales

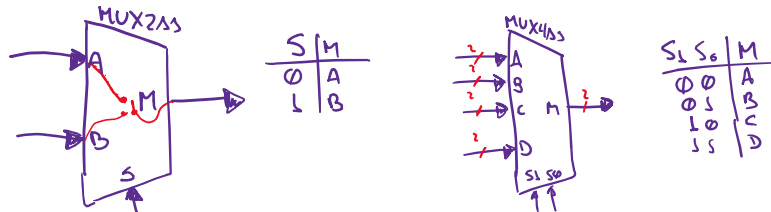
Comparadores de magnitud



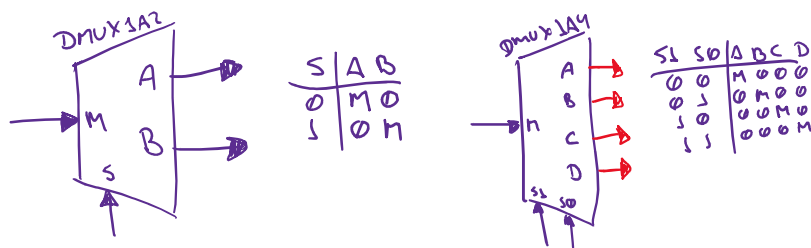
12

Revisión de circuitos digitales

- Multiplexores → Selector de datos

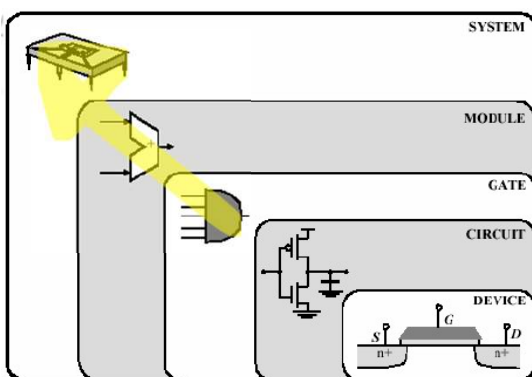


- Demultiplexores → Distribuidor de datos



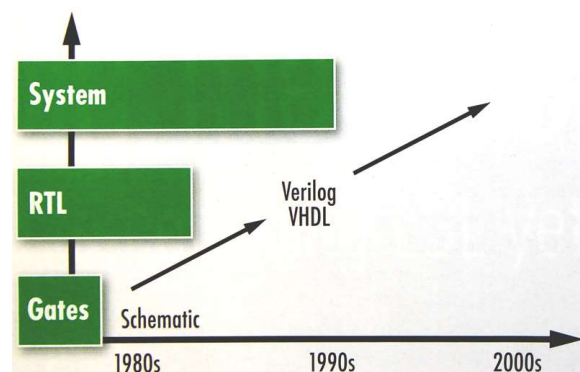
13

Jerarquía de diseño digital



Diferentes procedimientos para dar solución a un problema

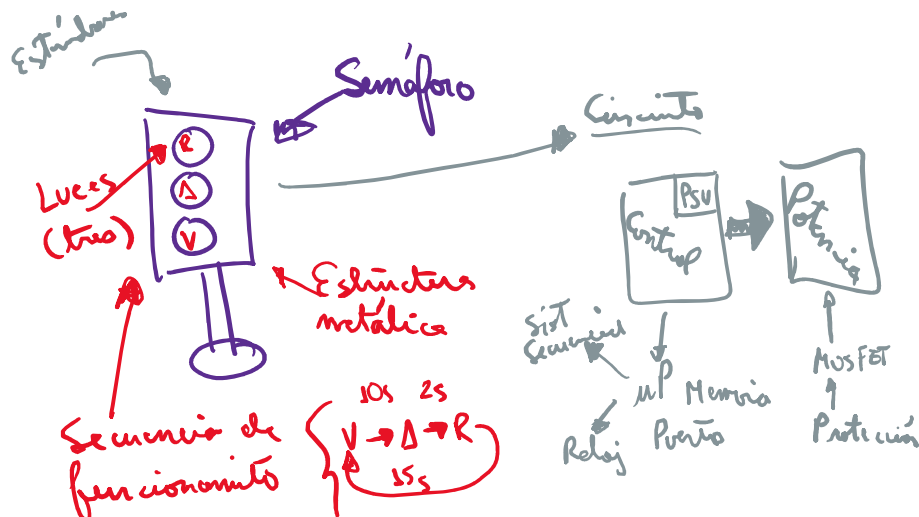
Más detalle



14

Metodología TOP-DOWN

- Especifica el funcionamiento desde lo mas genérico hacia lo mas específico.



15

El VHDL

- Las siglas: VHSIC-HDL (Very High Speed Integrated Circuit Hardware Description Language)
- Fue orientado a la describir funcionalmente **de manera formal** el cómo funciona un circuito integrado.
- ¿Cuándo se concibieron los HDL (VHDL y Verilog) y por que?
 - 1981 por el US Department of Defense
 - 1983 se formalizó el equipo de trabajo
 - 1985 VHDL revisión 7.2 (IBM y TI)
 - 1987 Estándar IEEE 1076 - 1987

16

El VHDL

- Inicialmente para describir

- Altera Corporation (1983)

Síntesis
Lógica



VHDL

- Fase de simulación
- Fase de implementación (síntesis lógica)



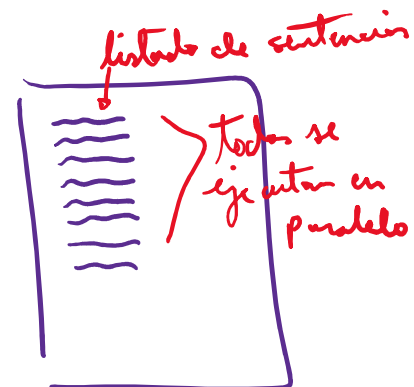
IBM PC
evolución

FPGA 1990
(Xilinx)

17

El VHDL - Características

- Actualmente se usa para modelado (descripción), simulación y síntesis (implementación) de circuitos.
- Estándar IEEE 1076, primera revisión en 1987, última revisión 2019(*) pero los desarrolladores soportan el 2008.
- Adoptado por los fabricantes de IC
- Lenguaje de naturaleza concurrente
- Modela retardos
- Modularidad (componentes – RTL)
- Diferentes estilos para describir
- **Independiente del fabricante y la tecnología**

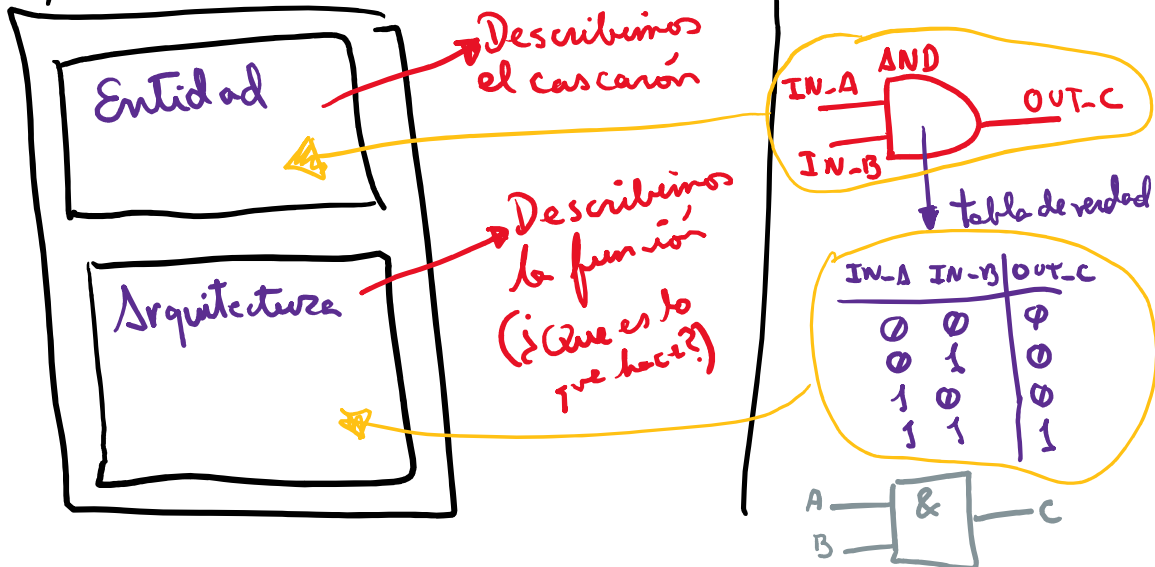


(*) <https://standards.ieee.org/standard/1076-2019.html>

18

Estructura de un código descriptivo en VHDL

*.vhd ó *.vhdl



19

Estándar IEEE 1164

- *Multivalued Logic System for VHDL Model Interoperability*
- Establece los niveles lógicos presentes en un circuito

- (5 minutos) ¿Cuántos niveles lógicos contempla?

std_logic

| Character | Value |
|-----------|-----------------------------------|
| 'U' | uninitialized |
| 'X' | strong drive, unknown logic value |
| '0' | strong drive, logic zero |
| '1' | strong drive, logic one |
| 'Z' | high impedance |
| 'W' | weak drive, unknown logic value |
| 'L' | weak drive, logic zero |
| 'H' | weak drive, logic one |
| '-' | don't care |

20

Modelo de la AND en VHDL:

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity AND_2 is
    port (IN-A, IN-B: in std_logic;
          OUT-C: out std_logic);
end AND_2;

architecture Flujos of AND_2 is
begin
    OUT-C <= IN-A and IN-B;
end Flujos;

```

Declaración de
 librería, IEEE y
 sus estándares

Declaración
 de la entidad

Declaración
 de la arquitectura

21

Fin de la sesión de teoría

22