© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Departamento de
Telemática y Electrónica

Escuela Técnica Superior de Ingeniería
y Sistemas de Telecomunicación

Universidad Politécnica de Madrid

Electrónica 2

Programa de Actividades
del Bloque Temático 1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Bloque Temático 1. Sesión 1. Codificación de la Información

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.P1	Presentación de la Asignatura*: normativa, organización, método, evaluación, contacto, tutorías, formación de los grupos y entorno virtual de la asignatura.*	Expositiva	50 min	S1_1 G1	No procede
BT1.P2	Codificación de la información I: *señales digitales, codificación de información con símbolos, concepto de bit y bus (nibble, Byte y Word), número de bits para codificación en sistemas digitales.*	Expositiva	25 min	S1_1 G1	No procede
BT1.P3	Codificación de números naturales I: *sistemas de numeración posicionales, sistema binario natural, conversión binario a decimal, definición de LSb y MSb. conversión decimal a binario por el método de divisiones sucesivas*	Expositiva	25 min	S1_1 G1	No procede
BT1.AINP1	Lectura de la "Guía de la Asignatura Electrónica II"	Lectura Individual	30 min	NP PS1_2	No procede
BT1.AINP2	Ejercicios sobre codificación y decodificación en binario natural. Método de divisiones sucesivas. [25 ejercicios]	Ejercicios Individual	50 min	NP PS1_2	Autoevaluación

Bloque Temático 1. Sesión 2. Codificación de Números

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.P4	Codificación de números naturales II: *tabla de potencias de dos, conversión decimal a binario por el método de descomposición.*	Expositiva	30 min	S1_2 G1	No procede
BT1.AIP1	Ejercicios de conversión decimal a binario: *por divisiones sucesivas, por descomposición (combinaciones de no más de ocho bits)*	Ejercicios Individual	20 min	S1_2 G1	Autoevaluación Por parejas
BT1.P5	Código BCD: *significación y utilidad, conversión binario-BCD y BCD-binario*	Expositiva	10 min	S1_2 G1	No procede
BT1.P6	Codificación de la información II: *Principios generales de codificación.*	Expositiva	20 min	S1_2 G1	No procede
BT1.AIP2	Ejercicios sobre principios generales de codificación y sistema BCD	Ejercicios Individual	20 min	S1_2 G1	Autoevaluación Por parejas
BT1.AINP3	Ejercicios sobre codificación en binario natural por descomposición. [10 ejercicios]	Ejercicios Individual	20 min	NP PS1_3	Autoevaluación
BT1.AINP4	Lectura de la ficha "Sistema hexadecimal y Códigos Alfanuméricos"	Lectura Dirigida	50 min	NP PS1_3	Autoevaluación
BT1.AINP5	Ejercicios sobre principios generales de codificación. [10 ejercicios]	Ejercicios Individual	50 min	NP PS1_3	Autoevaluación
BT1.AINP6	Cuestionario online sobre señales digitales, codificación y sistema binario natural y BCD.	Cuestionario Individual	40 min	online	Autoevaluación

Bloque Temático 1. Sesión 3. Aritmética Binaria

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.P7	Representación de números con signo: codificación y decodificación en complemento a dos. *Propuesta de un ejercicio sencillo*.	Expositiva	30 min	S1_3 G1	No procede
BT1.P8	Suma y resta de números en binario natural. El acarreo de la suma.	Expositiva	25 min	S1_3 G1	No procede
BT1.P9	Suma y resta de números en complemento a dos. El overflow.	Expositiva	25 min	S1_3 G1	No procede
BT1.P10	Extensión del número de bits en binario natural y complemento a 2. Cambio de signo en complemento a 2. División y multiplicación por potencias de 2. Truncamiento y redondeo.	Expositiva	20 min	S1_3 G1	No procede
BT1.AINP7	Ejercicios de conversión entre códigos. [4 hex-bin, 4 bin-hex, 2 dec-hex, 2 hex-dec, 12 de complemento a dos y 2 de código ASCII]	Ejercicios Individual	60 min	NP PS1_4	Autoevaluación
BT1.AINP8	Ejercicios de aritmética binaria. [25 Ejercicios]	Ejercicios Individual	60 min	NP PS1_4	Autoevaluación

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Bloque Temático 1. Sesión 4. Ejercicios sobre Codificación y Aritmética Binaria

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.ATGP1	Torneo EII sobre codificación y aritmética binaria	Aprendizaje Cooperativo	110 min	S1_4 G2	Entregable
BT1.AINP9	Lectura de la ficha sobre "Algebra de Boole y Circuitos Combinacionales".	Lectura	120 min	NP PS1_5	No procede

Bloque Temático 1. Sesión 5. Principios Básicos de los Sistemas Combinacionales

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.P11	Concepto y modelo de funcionamiento de sistemas combinacionales. Caracterización de los sistemas combinacionales: interfaz, esquema de codificación y funcionalidad.	Expositiva	20 min	S1_5 G1	No procede
BT1.P12	Resolución de dudas sobre Algebra de Boole. Notación, evaluación de maxitérminos, minitérminos y expresiones canónicas.	Expositiva	20 min	S1_5 G1	No procede
BT1.AIP3	Ejercicio sobre evaluación de expresiones canónicas.	Ejercicios Individual	15 min	S1_5 G1	Autoevaluación Por parejas
BT1.P13	Tablas de verdad. Obtención de la tabla de verdad a partir de una expresión algebraica. Extracción de expresiones canónicas de tablas de verdad.	Expositiva	30 min	S1_5 G1	No procede
BT1.AIP4	Ejercicio sobre obtención de tablas de verdad a partir de expresiones algebraicas.	Ejercicios Individual	15 min	S1_5 G1	Autoevaluación Por parejas
BT1.AINP10	Ejercicios de extracción de expresiones canónicas a partir de tablas de verdad, cálculo booleano y obtención de tablas de verdad a partir de expresiones.	Ejercicios Individual	75 min	NP PS1_6	Autoevaluación
BT1.AINP11	Cuestionario Moodle sobre caracterización de sistemas digitales (combinacionales / secuenciales)	Ejercicios Individual	15 min	NP PS1_6	Autoevaluación

Bloque Temático 1. Sesión 6. Síntesis de Circuitos Combinacionales

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.AIP5	Ejercicios de caracterización de circuitos combinacionales, cálculo de valor de expresiones booleanas, obtención de tablas de verdad a partir de expresiones algebraicas y obtención de expresiones algebraicas a partir de tablas de verdad	Ejercicios Individual	70 min	S1_6 G1	Autoevaluación
BT1.P14	Metodología de síntesis de sistemas combinacionales: caracterización de la interfaz, formulación de la funcionalidad como una tabla de verdad y extracción de la expresión canónica	Expositiva	30 min	S1_6 G1	No procede
BT1.AINP12	Ejercicios de síntesis de circuitos combinacionales simples	Ejercicios Individual	100 min	NP PS1_7	Autoevaluación
BT1.AINP13	Cuestionario Moodle de códigos y señales digitales.	Cuestionario Individual	50 min	online	Autoevaluación

Bloque Temático 1. Sesión 7. Algebra de Boole y Simplificación

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.ATGP2	Ejercicios de síntesis de circuitos combinacionales simples. Cada miembro del grupo resuelve un ejercicio y posteriormente se realiza una puesta en común con explicación y resolución de dudas de sus compañeros.	Aprendizaje cooperativo	40 min	S1_7 G2	Autoevaluación
BT1.ATGP3	Un miembro del grupo, elegido al azar por el profesor, realizará un ejercicio de síntesis de circuitos combinacionales simples.	Aprendizaje cooperativo	10 min	S1_7 G2	Entregable
BT1.P15	Teoremas básicos del Álgebra de Boole: De Morgan, Adyacencia, Simplificación, Absorción, Idempotencia. Aplicación a la simplificación de expresiones booleanas.	Expositiva	15 min	S1_7 G1	No procede
BT1.AIP6	Ejercicios elementales de simplificación.	Ejercicios Individual	35 min	S1_7 G1	Autoevaluación
BT1.AINP14	Cuestionario Moodle sobre obtención de tablas de verdad a partir de un enunciado y cálculo booleano	Ejercicios Individual	40 min	NP PS1_8	Autoevaluación
BT1.AINP15	Ejercicios sencillos de simplificación algebraica	Ejercicios Individual	60 min	NP PS1_8	Autoevaluación

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Bloque Temático 1. Sesión 8. Puertas lógicas

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.P16	Sistemas combinacionales con indiferencias. Extracción de expresiones algebraicas. Construcción de tablas con indiferencias. Alcance de la metodología de síntesis de sistemas combinacionales basada en tablas de verdad. Tablas de verdad resumidas. Casuística de construcción de las tablas.	Expositiva	50 min	S1_8 G1	No procede
BT1.P17	Puertas lógicas. Puertas XOR y XNOR. Propiedades de XOR y XNOR.	Expositiva	25 min	S1_8 G1	No procede
BT1.P18	Representación de circuitos combinacionales con puertas lógicas. Análisis y síntesis.	Expositiva	25 min	S1_8 G1	No procede
BT1.AINP16	Lectura sobre subsistemas combinacionales y realización de ejercicios de síntesis de sistemas combinacionales.	Lectura	140 min	NP PS1_9	Autoevaluación

Bloque Temático 1. Sesión 9. Ejercicios de Circuitos Combinacionales

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.AIP7	Ejercicios de obtención de tablas de verdad a partir de expresiones algebraicas.	Ejercicios Individual	10 min	S1_9 G1	Autoevaluación Por parejas
BT1.AIP8	Ejercicios de síntesis de circuitos combinacionales.	Ejercicios Individual	20 min	S1_9 G1	Autoevaluación Por parejas
BT1.AIP9	Ejercicios de simplificación.	Ejercicios Individual	20 min	S1_9 G1	Autoevaluación Por parejas
BT1.ATGP4	Ejemplos de diseño de circuitos combinacionales. Ejercicios para realizar en grupo basados en los ejemplos presentados.	Aprendizaje cooperativo	50 min	S1_9 G2	Autoevaluación
BT1.AINP17	Ejercicios sobre caracterización, síntesis, simplificación y representación con puertas lógicas de circuitos combinacionales.	Ejercicios Individual	60 min	NP PS1_10	Autoevaluación
BT1.AINP18	Ejercicios sobre caracterización, síntesis, simplificación y representación con puertas lógicas de circuitos combinacionales.	Ejercicios Individual	60 min	NP PS1_10	Autoevaluación

Bloque Temático 1. Sesión 10. Cronogramas

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.P19	Representación mediante cronogramas del funcionamiento de un circuito combinacional. Aplicación en resultados de simulación. Representación de bits y buses. Representación de indiferencias e indeterminaciones. Realización y análisis de cronogramas.	Expositiva	15 min	S1_10 G1	No procede
BT1.ATGP5	Realización de cronogramas de circuitos ya conocidos. Ejercicios para obtener la tabla de verdad a partir de un cronograma.	Aprendizaje cooperativo	35 min	S1_10 G2	Entregable
BT1.ATGP6	Ejercicios sobre síntesis de subsistemas combinacionales.	Aprendizaje cooperativo	50 min	S1_10 G2	Autoevaluación
BT1.AINP19	Ejercicios sobre síntesis, representación con puertas lógicas y obtención de cronogramas de circuitos combinacionales.	Ejercicios Individual	120 min	NP PS1_11	Autoevaluación

Bloque Temático 1. Sesión 11. Síntesis de sistemas combinacionales

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.ATGP7	Ejercicios sobre síntesis de sistemas combinacionales.	Aprendizaje cooperativo	100 min	S1_11 G2	Entregable

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Bloque Temático 1. Sesión 12. Circuitos combinacionales complejos

Código	Titulo	Modalidad	Duración	Sesión	Evaluación
BT1.P20	Circuitos combinacionales complejos. Ejemplos de diseño: sumadores serie, multiplexores y restadores. Lenguajes de descripción *Hardware* (HDL)	Expositiva	50 min	S1_12 G1	No procede
BT1.ATGNP1	Preparación del examen en grupo.	Repaso	60 min	S1_8 G1	No procede
BT1.AINP20	Preparación individual del examen: repaso del trabajo realizado y realización de ejercicios alternativos si es necesario.	Ejercicios Individual	160 min	NP	No procede
BT2.ATGNP1	Lectura individual del texto Tecnología I. Discusión en grupo de las dudas sobre lo trabajado. Planteamiento de dos dudas, curiosidades, sugerencias, inquietudes, etc., para llevar a la sesión presencial.	Aprendizaje cooperativo	80 min	NP PS2_1	No procede

La notación empleada es la siguiente:

BTn.Pm	Presentación del profesor nº m del Bloque Temático n
BTn.AINPm	Actividad Individual No Presencial nº m del Bloque Temático n
BTn.AIPm	Actividad Individual Presencial nº m del Bloque Temático n
BTn.ATGNPm	Actividad de Trabajo en Grupo No Presencial nº m del Bloque Temático n
BTn.ATGPm	Actividad de Trabajo en Grupo Presencial nº m del Bloque Temático n
Sx_y	Sesión presencial y del bloque x
PSx_y	Actividad a realizar de manera previa a la sesión presencial y del bloque x
NP	Modalidad no presencial
G1	Grupo presencial completo
G2	Grupo presencial desdoblado
G3	Grupo presencial de laboratorio

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P2
Codificación de la Información I

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Introducción a los Sistemas Digitales

- ¿Qué tipo de sistemas electrónicos lleva un teléfono móvil?
- Analógicos
  - Baja frecuencia
    - Altavoz, micrófono
    - Amplificadores
  - Alta frecuencia
    - Emisor de RF
    - Receptor de RF
    - Antenas

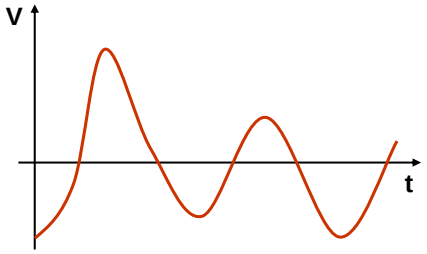
- Digitales
  - DSP
  - Tarjetas inteligentes (SIM)
  - Ampliación de memoria
  - MP3

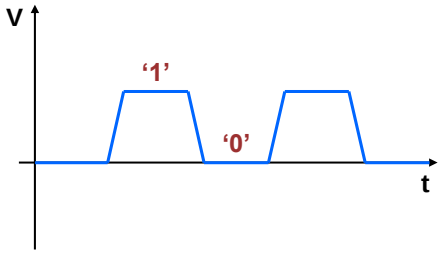
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Introducción a los Sistemas Digitales

- Señal analógica
  - Infinitos valores dentro de un rango
- Señal digital
  - Dos valores: ‘0’ y ‘1’

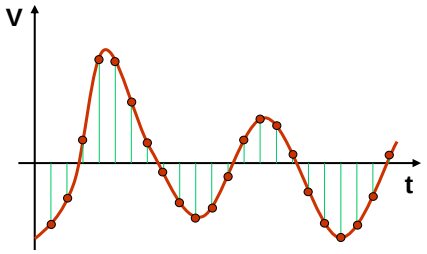




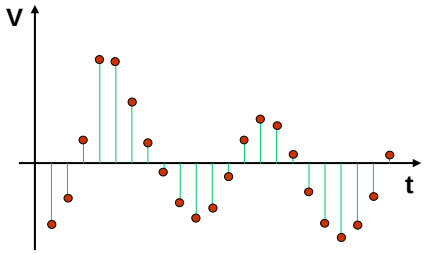
BT1_S1 BT1_P2_3

Introducción a los Sistemas Digitales

- ¿Cómo se convierten señales analógicas a digitales?
  - Primero es necesario seleccionar un número finito de puntos de la señal analógica. Muestreo.



Señal analógica



Señal analógica muestreada

BT1_S1 BT1_P2_4

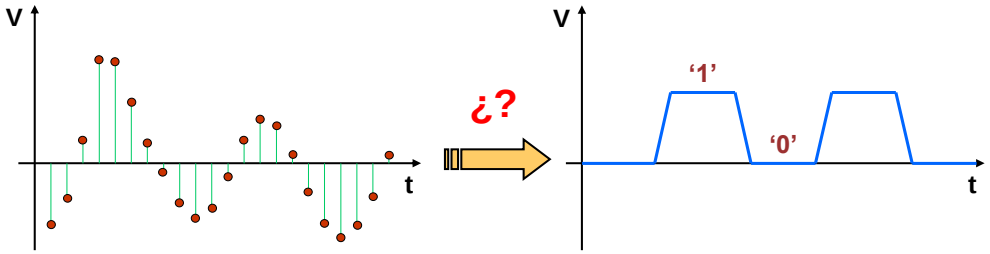
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Introducción a los Sistemas Digitales

- ¿Cómo se convierten señales analógicas a digitales?
  - Después, a cada punto se le hace corresponder un código compuesto exclusivamente por 0s y 1s. Codificación.
  - Codificación, codificación, … uhm, no me viene nada por esto.



Señal analógica muestreada

Señal Digital

Codificación de información con símbolos

- Morse
  - Codificación con puntos y rayas
  - S.O.S ⇒ . . . _ _ _ . . .
- Semáforo
  - Codificación con colores
    - Rojo ⇒ No pasar
    - Amarillo ⇒ Detenerse
    - Verde ⇒ Pasar
- BT2_S3_1
  - Bloque temático II, semana 3, primera clase.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Codificación de información con símbolos

- Información a codificar:
  - Números
    - Naturales
    - Enteros
    - Fracciones
  - Caracteres
    - Mayúsculas
    - Minúsculas
    - Caracteres especiales (letras con tildes, comandos, etc.)
  - Elementos de un conjunto
    - Colores del semáforo
    - Aparcamiento lleno o vacío, …

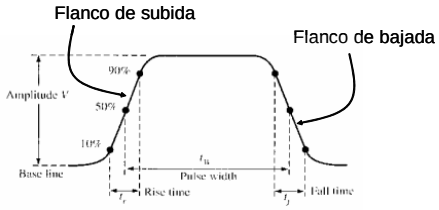
Codificación de información en los Sistemas digitales

- Se codifican empleando combinaciones de ‘1s’ y ‘0s’
- A cada ‘1’ y ‘0’ de estas combinaciones se le denomina BIT.
- Un Bus es un conjunto de BITs agrupados que contiene información codificada de una variable.

Ideal

Forma de Onda Digital

Real



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Codificación de información en los Sistemas digitales

- Nibble: información codificada con 4 bits
  - 1001
- Byte: información codificada con 8 bits.
  - Un byte está compuesto por 2 nibbles.
- Word: información codificada con 16 bits.
  - Un word está compuesto por 2 bytes.
- Double Word: información codificada con 32 bits
  - Un word está compuesto por 4 bytes.
- Quad Word: información codificada con 64 bits
  - Un word está compuesto por 8 bytes.

BT1_S1 BT1_P2_9

Número de bits necesarios

- ¿Cuántos bits se necesitan para codificar objetos (números, caracteres, etc.)?

Con 1 Bit se pueden codificar 2 objetos, propiedades o situaciones diferentes.

Blanco	0
Negro	1

Abierto	0
Cerrado	1

Con 2 Bits se pueden codificar 4 objetos, propiedades o situaciones diferentes.

Blanco	00
Negro	01
Azul	10
Verde	11

Español	00
Inglés	01
Francés	10
Alemán	11

BT1_S1 BT1_P2_10

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P3
Codificación de números naturales I

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas de numeración posicional

- La posición que ocupa cada cifra representa una cantidad.
- El sistemas de numeración posicional más utilizado es el DECIMAL.
- La cifra '1 928' en decimal representa:
  - 1 millar
  - 9 centenas
  - 2 decenas
  - 8 unidades
- No es lo mismo tener 1 928 € que tener 8 291 € ¿o sí?

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas de numeración posicional

- En general, y dependiendo de la base utilizada, B, la cantidad expresada en un sistema de numeración posicional viene dada por el polinomio característico:

$$\ldots + x_2 \cdot B^2 + x_1 \cdot B^1 + x_0 \cdot B^0 + x_{-1} \cdot B^{-1} + x_{-2} \cdot B^{-2} +$$

- En el ejemplo anterior, 1 928, y en decimal (B = 10), el polinomio característico es:

$$1 \cdot 10^3 + 9 \cdot 10^2 + 2 \cdot 10^1 + 8 \cdot 10^0$$

Codificación de números naturales

- Código Binario Natural
- Base: 2
- Símbolos: 0 y 1
- Polinomio característico

$$\ldots + b_4 \cdot 2^4 + b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

- Ejemplo: 0101 $\Rightarrow$ 5

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Conversión de Binario a Decimal

- Se utiliza el polinomio característico de la base 2.

$$\ldots + b_4 \cdot 2^4 + b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

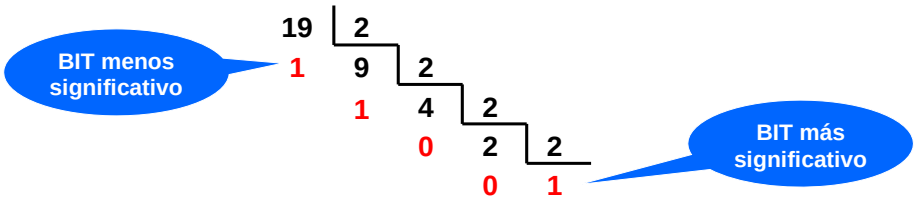
- Ejemplo: 1101

$$1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13_D$$

BT1_S1 BT1_P3_5

Conversión de Decimal a Binario

- Divisiones sucesivas.
  - Ejemplo: 19_D



  - Resultado: 10011_B

MSB LSB

MSB: Most Significant Bit
(*Bit más significativo*)

LSB: Least Significant Bit
(*Bit menos significativo*)

BT1_S1 BT1_P3_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Lectura de la Guía de Aprendizaje de Electrónica 2.**	**CÓDIGO:** **BT1_AINP1**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Lectura. Individual	TIPO:	No presencial	DURACIÓN:	30 minutos

CALENDARIO:	Primera Sesión. Previo a la Segunda Sesión	REQUISITOS:	Ninguno

CRITERIO DE ÉXITO:	

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	No procede

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

LECTURA DE LA GUÍA DE APRENDIZAJE DE ELECTRÓNICA 2

La Guía de Aprendizaje de Electrónica 2 se ha elaborado con la pretensión de que constituya una ayuda básica para orientar a los estudiantes que han decidido cursar esta asignatura. Este documento expone información general sobre la misma, en particular, su sentido y utilidad, los aprendizajes que se van a alcanzar una vez finalizado el cuatrimestre o los contenidos que integran el programa. Asimismo, se detallan los recursos disponibles, así como el programa de actividades y su temporización a lo largo de las 14 semanas que dura el cuatrimestre.

En definitiva, este documento incluye la información necesaria para un seguimiento eficaz de las actividades de aprendizaje y para recibir orientación acerca de cómo afrontar el estudio de los contenidos. De manera que recomendamos su completa lectura antes de iniciar el trabajo propiamente dicho en la asignatura, para lo cual pensamos que puede necesitar alrededor de 30 minutos.

Una vez finalizada la lectura es recomendable dedicar unos minutos al siguiente conjunto de tareas:

- Subrayar sobre el documento aquellos aspectos que considere esenciales.
- Establecer, dentro del horario semanal, las cuatro horas que se dedicarán para realizar las actividades no presenciales que se proponen en la asignatura.
- Conectarse a la página web de la asignatura (moodle) para familiarizarse con su contenido y organización.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre codificación y decodificación en binario natural.**	**CÓDIGO:** **BT1_AINP2**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	50 minutos

CALENDARIO:	Primera Sesión. Previo a Segunda Sesión	REQUISITOS:	

CRITERIO DE ÉXITO:	Realización correcta de al menos 20 de las 25 conversiones

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS SOBRE CODIFICACIÓN Y DECODIFICACIÓN EN BINARIO NATURAL

PARTE I. Ejercicios con Solución Desarrollada.

Se desarrollan dos ejercicios en los que se explica el procedimiento para decodificar números en binario natural mediante la regla de los pesos y codificar números en binario natural por el método de divisiones sucesivas.

EJERCICIO A) Decodificación de números en binario natural

Para decodificar un número expresado en binario natural se procede mediante la regla de los pesos de forma que se muestra a continuación.

El número 10011011_B se transforma mediante la expresión:

$$1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 128 + 16 + 8 + 2 + 1 = 155$$

Por lo que equivale al número decimal 155

EJERCICIO B) Codificación de números por el método de divisiones sucesivas

Este algoritmo para la conversión de números decimales en su equivalente binario natural es el más utilizado cuando el número a convertir es elevado. Se basa en la obtención de los dígitos binarios que representan el número N como los sucesivos restos que quedan de la división de un número X entre dos (por ser el sistema binario un sistema de base dos). Siendo este número X igual al número N en un primer lugar y posteriormente la parte entera de la división efectuada. Este procedimiento se repite hasta que el cociente sea cero.

A continuación se ilustra este procedimiento mediante un ejemplo. Se procederá con la obtención de la expresión en binario natural del número decimal 6237. Para lo cual se procede de la forma siguiente:

6237/2	= 3118	+ 1	→ b_0 = 1
3118/2	= 1559	+ 0	→ b_1 = 0
1559/2	= 779	+ 1	→ b_2 = 1
779/2	= 389	+ 1	→ b_3 = 1
389/2	= 194	+ 1	→ b_4 = 1
194/2	= 97	+ 0	→ b_5 = 0
97/2	= 48	+ 1	→ b_6 = 1
48/2	= 24	+ 0	→ b_7 = 0
24/2	= 12	+ 0	→ b_8 = 0
12/2	= 6	+ 0	→ b_9 = 0
6/2	= 3	+ 0	→ b_{10} = 0
3/2	= 1	+ 1	→ b_{11} = 1
1/2	= 0	+ 1	→ b_{12} = 1

Así pues, el número decimal 6237 puede ser representado en binario natural mediante la expresión 1100001011101.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE II. Ejercicios con Solución Breve.

En lo que sigue se presentan algunos enunciados de ejercicios que deberán ser resueltos sin consultar la solución, tratando de poner en práctica los conocimientos sobre codificación en binario natural adquiridos. Posteriormente se tendrá la oportunidad de contrastar la solución con la que se ofrece y detectar posibles errores.

EJERCICIO 1) Dado el siguiente conjunto de números expresados en binario natural, se pide obtener el número equivalente expresado en decimal para cada uno de ellos.

Nº	Binario Natural	Decimal
1	0111	
2	1111	
3	1001	
4	1101	
5	0101	
6	1 0101	
7	0 1011	
8	1 1111	
9	1 0000	
10	11 1110	
11	01 1010	
12	11 1111	
13	11 0101	
14	01 0100	
15	110 0100	
16	110 1110	
17	001 0110	
18	111 1111	
19	1100 1100	
20	1111 0100	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 2) Codifique en binario natural el siguiente conjunto de números decimales por el método de divisiones sucesivas. Muestre los detalles del procedimiento seguido.

- ✓ 114
- ✓ 205
- ✓ 150
- ✓ 41
- ✓ 79

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE III. Soluciones a los ejercicios con Solución Breve.

EJERCICIO 1)

Nº	Binario Natural	Decimal
1	0111	7
2	1111	15
3	1001	9
4	1101	13
5	0101	5
6	1 0101	21
7	0 1011	11
8	1 1111	31
9	1 0000	16
10	11 1110	62
11	01 1010	26
12	11 1111	63
13	11 0101	53
14	01 0100	20
15	110 0100	100
16	110 1110	110
17	001 0110	22
18	111 1111	127
19	1100 1100	204
20	1111 0100	244

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 2)

- ✓ 114

114/2	= 57	+ 0	→ b_2 = 0
57/2	= 28	+ 1	→ b_3 = 1
28/2	= 14	+ 0	→ b_4 = 0
14/2	= 7	+ 0	→ b_5 = 0
7/2	= 3	+ 1	→ b_6 = 1
3/2	= 1	+ 1	→ b_7 = 1
1/2	= 0	+ 1	→ b_8 = 1

- ✓ 205

205/2	= 102	+ 1	→ b_2 = 1
102/2	= 51	+ 0	→ b_3 = 0
51/2	= 25	+ 1	→ b_4 = 1
25/2	= 12	+ 1	→ b_5 = 1
12/2	= 6	+ 0	→ b_6 = 0
6/2	= 3	+ 0	→ b_7 = 0
3/2	= 1	+ 1	→ b_8 = 1
1/2	= 0	+ 1	→ b_9 = 1

- ✓ 150 → 1001 0110

- ✓ 41 → 10 1001

- ✓ 79 → 100 1111

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P4
Codificación de números naturales II

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tabla de potencias de 2

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

2^9	512
2^{10}	1024
2^{11}	2048
2^{12}	4096
2^{13}	8192
2^{14}	16384
2^{15}	32768
2^{16}	65536

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

7	6	5	4	3	2	1	0
1	1	0	1	1	0	0	1

$11011001_B = 217_D$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Utilización de múltiplos

Kilo	2^{10}	1024	1K	$\approx 10^3$
Mega	2^{20}	1024·1024	1M	$\approx 10^6$
Giga	2^{30}	1024·1024·1024	1G	$\approx 10^9$
Tera	2^{40}	1024·1024·1024·1024	1T	$\approx 10^{12}$

Algunos Ejemplos:

2^{13}	$2^3 \cdot 2^{10}$	8·1024	8K
2^{24}	$2^4 \cdot 2^{20}$	16·1024·1024	16M
2^{32}	$2^2 \cdot 2^{30}$	4·1024·1024·1024	4G
2^{41}	$2^1 \cdot 2^{40}$	2·1024·1024·1024·1024	2T

BT1_S2 BT1_P4_3

Conversión de decimal a binario por descomposición

```
  2 1 7
- 1 2 8   = 2^7
  -----
    8 9
-   6 4   = 2^6
  -----
    2 5
-   1 6   = 2^4
  -----
      9
-     8   = 2^3
  -----
      1   = 2^0
```

$1101\ 1001_B = 217_D$

2^1	2	2^9	512
2^2	4	2^{10}	1024
2^3	8	2^{11}	2048
2^4	16	2^{12}	4096
2^5	32	2^{13}	8192
2^6	64	2^{14}	16384
2^7	128	2^{15}	32768
2^8	256	2^{16}	65536

BT1_S2 BT1_P4_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P5

Código BCD

Código BCD

- BCD (BINARY CODED DECIMAL)
- Representa cada una de las cifras decimales de un número expresado en decimal con 4 bits.
- Un número decimal de n cifras necesita 4·n bits para expresarlo en BCD
  - Cada grupo de 4 bits representa un código decimal entre 0 y 9.
  - Las combinaciones binarias 1010, 1011, 1100, 1101, 1110 y 1111 no son válidas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Conversión Decimal-BCD y BCD-Decimal

Decimal a BCD

3 9 5

0011 1001 0101

BCD a Decimal

0111 0110 0010

7 6 2

Cifra	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

BT1_S2 BT1_P5_3

Diferencia con Binario Natural

Decimal a BCD

1 5 3

0001 0101 0011

Decimal a Binario

1 5 3 = 1001 1001

$$
\begin{array}{rrl}
 & 153 & \\
- & 128 & = 2^7 \\
\hline
 & 25 & \\
- & 16 & = 2^4 \\
\hline
 & 9 & \\
- & 8 & = 2^3 \\
\hline
 & 1 & = 2^0
\end{array}
$$

BT1_S2 BT1_P5_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Conversión Binario a BCD

- Conversión en 2 pasos:
  1. Binario a Decimal
  2. Decimal a BCD

- Ejemplo, convertir $1001\ 0011_B$ a BCD
  1. $1001\ 0011_B = 2^7 + 2^4 + 2^1 + 2^0 = 128 + 16 + 2 + 1 = 147_D$
  2. $147_D = 0001\ 0100\ 0111_{BCD}$

BT1_S2 BT1_P5_5

Conversión BCD a Binario

- Conversión en 2 pasos:
  1. BCD a Decimal
  2. Decimal a Binario

- Ejemplo, convertir $0100\ 1001_{BCD}$ a Binario
  1. $0100\ 1001_{BCD} = 49_D$
  2. $49_D = 0011\ 0001_B$

BT1_S2 BT1_P5_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P6
Codificación de la Información II

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Principios generales de codificación

- ¿Cuántos bits se necesitan para codificar objetos (números, caracteres, etc.)?

Con 1 Bit se pueden codificar 2 objetos, propiedades o situaciones diferentes.

Blanco	0
Negro	1

Abierto	0
Cerrado	1

Con 2 Bits se pueden codificar 4 objetos, propiedades o situaciones diferentes.

Blanco	00
Negro	01
Azul	10
Verde	11

Español	00
Inglés	01
Francés	10
Alemán	11

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Principios generales de codificación

- ¿Cuántos bits se necesitan para codificar objetos (números, caracteres, etc.)?

$$n \geq \log_2 m$$

BT1_S2 BT1_P6_3

Principios generales de codificación

- ¿Cuántos bits se necesitan para codificar objetos (números, caracteres, etc.)?
- Con 'n' bits podemos representar hasta 2^n objetos.
  - El rango de números naturales que se pueden codificar con n BITs es:

$$(0, 2^n-1)$$

  - Es decir, con 4 bits, se pueden representar desde el 0 hasta el 15.

BT1_S2 BT1_P6_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Principios generales de codificación

- ¿Cuántos bits se necesitan para codificar objetos (números, caracteres, etc.)?
- Con 'n' bits podemos representar hasta 2^n objetos.
- Existen distintas opciones de asignación de códigos.
  - A cada elemento del conjunto debe corresponderle un único código de 0s y 1s.
    - Semáforo: rojo → 00, amarillo → 01, verde → 10
  - Se debe utilizar el menor número de bits.
    - Semáforo: rojo → 0000, amarillo → 0001, verde → 0010
  - La asignación debe favorecer las operaciones de codificar y descodificar.

BT1_S2 BT1_P6_5

Principios generales de codificación

- ¿Cuántos bits se necesitan para codificar objetos (números, caracteres, etc.)?
- Con 'n' bits podemos representar hasta 2^n objetos
- Existen distintas opciones de asignación de códigos.
- La elección de un determinado código influye en la construcción de los circuitos.

BT1_S2 BT1_P6_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

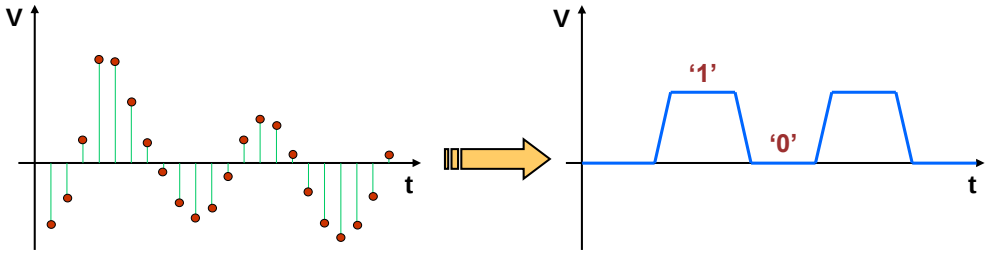
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Principios generales de codificación

- ¿Cómo se convierten señales analógicas a digitales?
  - Después, a cada punto se le hace corresponder un código compuesto exclusivamente por 0s y 1s. Codificación.
  - Codificación, codificación, … ¡ahora si!



Señal analógica muestreada

Señal Digital

BT1_S2 BT1_P6_7

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre codificación en binario natural por descomposición.**	**CÓDIGO:** **BT1_AINP3**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	20 minutos

CALENDARIO:	Segunda Sesión. Previo a la Tercera Sesión	REQUISITOS:	

CRITERIO DE ÉXITO:	Realización correcta de al menos ocho de las diez conversiones

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS SOBRE CODIFICACIÓN EN BINARIO NATURAL POR DESCOMPOSICIÓN

PARTE I. Ejercicio con Solución Desarrollada.

EJERCICIO A) Codificación de números por el método de descomposición en potencias de dos.

Este algoritmo suele ser el preferido cuando el número a convertir no es excesivamente grande. Se basa en el hecho de que en cualquier sistema posicional cualquier cifra multiplicada por su peso siempre es estrictamente mayor que la suma de las cifras de su derecha multiplicadas por sus pesos. Por ejemplo en el sistema decimal, con el número 2345 se comprueba que 2000 es mayor que 345, que 300 es mayor que 45, y que 40 es mayor que 5. Un ejemplo de caso límite es el número 1999, en el que se observa que 1000 es mayor que 999 (la cifra máxima que se puede alcanzar con tres dígitos decimales).

En un sistema en base dos, como es el sistema binario natural, esta propiedad puede ser aprovechada para encontrar los dígitos binarios mediante la suma de las potencias de dos (1, 2, 4, 8, 16, 32, 64, 128, 256, 512…) en que puede ser descompuesto cualquier número decimal.

Para una mejor comprensión del algoritmo, se ilustra mediante un ejemplo. Así, si se desea convertir el número 201 a binario natural procederemos de la siguiente manera.

- La potencia octava de dos es 256, pero este número (y ninguna otra potencia superior) no puede aparecer por ser superior al número 201. Así pues, la mayor de las potencias de dos contenida en el número 201 es la séptima potencia, 2^7=128.
- Podemos decir pues que 201 = 2^7 + X. Para proseguir con el procedimiento necesitamos conocer el valor de X, que es 201 – 128 = 73.
- A continuación, se procede de igual forma con la cantidad 73, buscando la mayor potencia de dos que está contenida en esta cifra, resultando ser la sexta potencia de dos, 2^6, es decir, 64. Así pues, ahora se sabe que 201 = $2^7 + 2^6$ + Y. Siendo Y = 9.
- Finalmente, es fácil determinar que el número 9 es la suma de las potencias 2^3 y 2^0, 8 y 1. Por lo que la descomposición completa del número 201 es $2^7 + 2^6 + 2^3 + 2^0$.
- En definitiva se ha comprobado que el número 201 se puede expresar mediante el siguiente conjunto de cifras binarias multiplicadas por su peso correspondiente:

$$201 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

- Con lo que se puede concluir que la expresión en binario natural del número 201 es: 11001001

La aplicación eficiente de este método de codificación obliga a conocer el valor de la sucesión de las primeras potencias de 2 (desde 2^0 hasta 2^{10}) –algo que resulta muy útil también para realizar cálculos relativos al número de bits necesarios para codificar un número o los elementos de un conjunto, por ejemplo-, por lo que debe intentar memorizarlas:

$2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, $2^7 = 128$, $2^8 = 256$, $2^9 = 512$, $2^{10} = 1024$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE II. Ejercicios con Solución Breve.

En lo que sigue se presentan algunos enunciados de ejercicios que deberán ser resueltos sin consultar la solución, tratando de poner en práctica los conocimientos sobre codificación en binario natural adquiridos. Posteriormente se tendrá la oportunidad de contrastar la solución con la que se ofrece y detectar posibles errores.

EJERCICIO 1) Codifique en binario natural el siguiente conjunto de números decimales por el método de descomposición en potencias de dos. Muestre los detalles del procedimiento seguido.

- ✓ 52
- ✓ 101
- ✓ 140
- ✓ 23
- ✓ 230

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 2) Codifique en binario natural el siguiente conjunto de números decimales por el método que prefiera. No es necesario mostrar los detalles del procedimiento seguido.

Nº	Decimal	Nº de Dígitos	Binario Natural
1	39	7 bits	
2	254	mínimo	
3	112	mínimo	
4	200	8 bits	
5	300	8 bits	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE III. Soluciones a los ejercicios con Solución Breve.

EJERCICIO 1)

- ✓ 52

  $52 = 32 + 16 + 4 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$

  El número decimal 52 equivale al 110100 en binario natural

- ✓ 101

  $101 = 64 + 32 + 4 + 1 = 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$

  El número decimal 100 equivale al 1100101 en binario natural

- ✓ 140

  $140 = 128 + 8 + 4 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$

  El número decimal 140 equivale al 10001100 en binario natural

- ✓ 23 → 10111

- ✓ 230 → 11100110

EJERCICIO 2)

Nº	Decimal	Nº de Dígitos	Binario Natural
1	39	7 bits	010 0111
2	254	mínimo	1111 1110
3	112	mínimo	111 0000
4	200	8 bits	1100 1000
5	300	8 bits	IMPOSIBLE

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Sistema Hexadecimal y Códigos Alfanuméricos.**	**CÓDIGO:** **BT1_AINP4**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Lectura Dirigida	TIPO:	No presencial	DURACIÓN:	50 minutos

CALENDARIO:	Segunda Sesión. Previo a la Tercera Sesión	REQUISITOS:	Principios generales de codificación

CRITERIO DE ÉXITO:	Compresión de los conceptos y procedimientos expuestos.

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	No procede

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistema Hexadecimal y Códigos Alfanuméricos

PARTE I. Sistema Hexadecimal

La utilización del sistema de numeración binario natural presenta un inconveniente cuando se trata de representar números grandes. En estas circunstancias, la representación binaria se materializa en una cadena muy larga de unos y ceros, que puede resultar de difícil manejo en las operaciones de codificación y decodificación. Por ejemplo, el número decimal 58469_D se expresaría en binario mediante la cadena siguiente: $1110\ 0100\ 0110\ 0101_B$, cuya interpretación es, a todas luces, incómoda. Una forma de reducir la extensión de estas cadenas de bits pasa por su expresión condensada en el denominado sistema de numeración hexadecimal.

Este sistema de numeración es un sistema posicional de base 16, es decir cada uno de los *dígitos hexadecimales* puede tomar un valor de entre 16 posibles. Para representar estos valores, se utilizan como símbolos los diez números del sistema decimal más un conjunto de seis letras para representar los valores superiores al nueve. Es decir se utilizan los símbolos: "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E" y "F"; para representar los valores: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 y 15, respectivamente.

Así como en el sistema decimal cada uno de los *dígitos decimales* tiene un peso equivalente al de una potencia de diez que se va incrementando a medida que se avanza hacia la izquierda, es decir: unidades (10^0), decenas (10^1), centenas (10^2), millares (10^3), etc. En el sistema hexadecimal, por ser la base dieciséis, cada dígito hexadecimal tiene un peso equivalente al de una potencia de 16 que se va incrementando igualmente a medida que se avanza hacia dígitos superiores. Así, por ejemplo, el número hexadecimal $E465_H$, sería el equivalente decimal:

$$14 \cdot 16^3 + 4 \cdot 16^2 + 6 \cdot 16^1 + 5 \cdot 16^0 = 57344 + 1024 + 96 + 5 = 58469_D$$

Así pues, es fácil deducir que con "n" dígitos hexadecimales se podrán representar 16^n números diferentes, en un rango desde el 0 hasta el $16^n - 1$. Por ejemplo, con 4 dígitos hexadecimales se pueden representar 16^4 (65536_D) números diferentes, estando comprendidos en un rango entre el 0000_H y el $FFFF_H$, o en su equivalente decimal, entre el 0_D y el 65535_D

El procedimiento de conversión de un número hexadecimal en su equivalente decimal, se basa en la regla de los pesos anteriormente descrita. Para llevar a cabo el procedimiento contrario, la conversión de un valor decimal en el número hexadecimal correspondiente, lo más usual es utilizar el mecanismo de descomposición. Este mecanismo consiste, de forma equivalente a como fue expuesto para el caso del sistema binario natural, en la expresión del número como la suma sucesiva de múltiplos de potencias enteras de la base, en el caso hexadecimal: 1, 16, 256, 4096, 65536, 1048576,…

Se ilustra el procedimiento mediante el ejemplo de la conversión del número 38426_D a hexadecimal. En primer lugar, es preciso determinar el número de dígitos hexadecimales que se necesitan para expresar este número. Puesto que el número decimal que se desea convertir es superior al peso del cuarto dígito (4096) pero inferior al quinto (65536), se concluye que el equivalente hexadecimal podrá ser expresado mediante cuatro dígitos.

A continuación, se inicia un proceso iterativo en el que se divide el número por el valor decimal del dígito de mayor peso, (38426/4096). Pues bien, el cociente obtenido será el valor que habrá que asignar al cuarto dígito hexadecimal. A continuación, con el resto de la división anterior (1562) se realiza la división por el peso del tercer dígito (256). Y así se procede sucesivamente, como se muestra.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

$38426_D = 9 \cdot 4096 + 1562 =$
$9 \cdot 4096 + 6 \cdot 256 + 26 =$
$9 \cdot 4096 + 6 \cdot 256 + 1 \cdot 16 + 10$
$9 \cdot 4096 + 6 \cdot 256 + 1 \cdot 16 + 10 \cdot 1 = 961A_H$

Sin embargo, la utilidad del sistema de numeración hexadecimal para facilitar el manejo de números binarios, se basa en el hecho de que la base dieciséis es una potencia exacta de dos, concretamente es 2^4. Esta propiedad posibilita una muy fácil y rápida conversión de números binarios a hexadecimal, y viceversa. Puesto que se da una equivalencia perfecta entre las 16 combinaciones posibles con cuatro bits y los 16 símbolos hexadecimales, como se muestra en la tabla:

Valor decimal	Binario	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Cuando se quiere obtener el valor hexadecimal de una expresión binaria, basta con agrupar en conjuntos de cuatro bits (denominado *nibble*) el número binario que se desea convertir. Este conjunto de cuatro bits podrá tomar un conjunto de 2^4, es decir 16, posibles valores decimales, desde 0 a hasta 15. Posteriormente se procede a representar el dígito hexadecimal al que corresponde el valor de los cuatro bits. Por ejemplo, el número binario $11010111111100110101101010 00_B$, se expresaría en hexadecimal de la siguiente manera:

$$\underbrace{1101}_{D}\ \underbrace{0111}_{7}\ \underbrace{1111}_{F}\ \underbrace{0011}_{3}\ \underbrace{0101}_{5}\ \underbrace{1010}_{A}\ \underbrace{1000}_{8_H}$$

Para el procedimiento inverso, se parte del número hexadecimal y se procede con la sustitución de cada dígito hexadecimal por su correspondiente combinación binaria. Así, por ejemplo, el número $8C5AD3_H$, se transforma en el número $1000\ 1100\ 0101\ 1010\ 1101\ 0011_B$.

Una última cuestión es qué se hace cuando el número de bits en los que viene expresado el número binario no es múltiplo de cuatro. La solución viene de la mano del concepto genérico de "extensión del número de bits de un dato". Hay que recordar que cualquier número binario puede ser expresado en un número mayor

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

de bits con solo añadir tantos ceros a la izquierda como sea preciso para alcanzar el número de bits deseado. Así pues, para realizar la conversión del número 1011000110_B, se procedería a añadir dos ceros a la izquierda para alcanzar un total de 12 bits (múltiplo de cuatro) y posteriormente a aplicar el procedimiento ya descrito. De forma que le número hexadecimal al que daría lugar la combinación binaria propuesta sería el $2C6_H$.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE II. Códigos alfanuméricos

En muchas aplicaciones de los sistemas digitales se requiere la utilización de datos que consisten en letras y otros caracteres simbólicos usualmente utilizados para transmitir mensajes de texto. Al conjunto de símbolos formado por las letras del alfabeto (mayúsculas y minúsculas), los diez símbolos numéricos, los signos de puntuación y otros símbolos que sirven para dar formato al texto (fin de línea, retorno de carro, tabulador, nuevo párrafo, fin de texto, …), se le denomina información alfanumérica. Algunos ejemplos de este tipo de símbolos son: a, b, G, M, á, E, u, ü, /, ¿, ¡, “, #, etc.

La utilización de este tipo de información en un sistema digital pasa por su codificación mediante combinaciones de unos y ceros, como es habitual. Sin embargo, el entendimiento entre los distintos sistemas se fundamenta en la utilización de la misma codificación en todos ellos, es decir, que cada uno de los símbolos tenga la misma combinación asignada. Para lograr este objetivo se han desarrollado estándares internacionales que definen el conjunto de símbolos que se pueden manejar, junto a la representación numérica, o código, que se le asigna. Los más extendidos son el código ASCII y el UNICODE. El primero de ellos es el más veterano, desarrollado en 1963 bajo la denominación *American Standard Code for Information Interchange*, permite la representación de hasta 128 símbolos codificados con siete bits, 256 símbolos codificados con ocho bits en el caso del ASCII extendido, que permiten la escritura de mensajes y el control del formato, según será representado por los sistemas encargados de presentar el mensaje. Sin embargo, este estándar tenía, y tiene, una limitación importante, derivada de que fue originalmente ideado para representar mensajes en lengua inglesa, por lo que carece de algunos símbolos de puntuación como la apertura de interrogación, letras como la “ñ”, o vocales acentuadas; de hecho, experimentó una ampliación en el año 1967 para incluir algunos de estos caracteres. Es por ello, que surgieron posteriormente nuevos estándares que permitían la utilización de símbolos de otras lenguas, griego, chino, japonés, etc., de los cuales el más extendido es el UNICODE, que no trataremos en este documento por exceder el alcance pretendido con una introducción a la codificación, pero del que puede obtenerse información en la dirección http://unicode.org/.

Las siguientes tablas muestran el conjunto de caracteres recogidos en el estándar ASCII, junto al código que se le asigna.

Tabla de caracteres estándar

ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo
0	0	NUL	16	10	DLE	32	20	(espacio)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(	56	38	8
9	9	TAB	25	19	EM	41	29	)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo	ASCII	Hex	Símbolo
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	□

Tabla de caracteres extendida

128	Ç	144	É	160	á	176	░	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	161	í	177	▒	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	162	ó	178	▓	195	├	211	╙	227	π	243	≤
131	â	147	ô	163	ú	179	│	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	164	ñ	180	┤	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	165	Ñ	181	╡	198	╞	214	╓	230	µ	246	÷
134	å	150	û	166	ª	182	╢	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	167	º	183	╖	200	╚	216	╪	232	Φ	248	°
136	ê	152	_	168	¿	184	╕	201	╔	217	┘	233	Θ	249	∙
137	ë	153	Ö	169	_	185	╣	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	170	¬	186	║	203	╦	219	█	235	δ	251	√
139	ï	156	£	171	½	187	╗	204	╠	220	▄	236	∞	252	_
140	î	157	¥	172	¼	188	╝	205	═	221	▌	237	φ	253	²
141	ì	158	_	173	¡	189	╜	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	174	«	190	╛	207	╧	223	▀	239	∩	255	
143	Å	192	└	175	»	191	┐	208	╨	224	α	240	≡		

Así por ejemplo para codificar la cadena “Electrónica 2” se utilizarían los siguientes códigos: 69, 108, 101, 99, 116, 114, 162, 110, 105, 99, 97, 32 y 50.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre principios generales de codificación.**	**CÓDIGO:** **BT1_AINP5**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	50 minutos

CALENDARIO:	Segunda Sesión. Previo a la Tercera Sesión	REQUISITOS:	Sistema binario natural

CRITERIO DE ÉXITO:	Realización correcta de al menos cinco de los ocho ejercicios

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS SOBRE PRINCIPIOS GENERALES DE CODIFICACIÓN

PARTE I. Conceptos Teóricos.

Los sistemas electrónicos procesan señales eléctricas que portan información. Una de las características distintivas de los sistemas electrónicos digitales es que utilizan **señales binarias**: señales que sólo pueden tomar dos valores: 0 y 1.

Este sistema de representación de la información restringe –a cambio, evidentemente, de varias ventajas que compensan esta limitación- la cantidad de información que pueden portar las señales: con una señal binaria, un **bit**[1], sólo se pueden codificar dos estados ({si,no}, {on-off}, {0-1}, etc.). Por ello, cuando en un sistema digital resulta necesario codificar los elementos de un conjunto con una mayor variedad de valores, es necesario recurrir a una agregación de bits -a un **dato** de n bits- con la que se pueda formar un conjunto suficiente de combinaciones, de unos y ceros, que permitan codificar todos los elementos que formen parte del conjunto.

El número de combinaciones distintas, de unos y ceros, que pueden realizarse con un dato de n bits viene dado por la siguiente ecuación:

número de combinaciones = 2^n ; siendo "n" el número de bits

Este número de combinaciones nos indica el número máximo de valores, o elementos, de un conjunto que podemos codificar con ese número de bits. Por otra parte, partiendo del conocimiento del número de elementos a codificar, N, podemos calcular, despejando *n* en la ecuación anterior, el número de bits que debe tener el dato para poder representar con una combinación diferente a cada uno de los N elementos:

$n \geq \log_2 N$

Este cálculo puede realizarse de un modo inmediato si se conoce la sucesión de potencias de dos para los primeros números enteros: $2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$, $2^7=128$, $2^8=256$, $2^9=512$, etcétera. Por ejemplo, para codificar 200 valores distintos necesitamos un dato de 8 bits, con el que pueden formarse ($2^8=256$) 256 combinaciones, porque 7 resultan insuficientes (ya que $2^7=128$).

Una vez calculado el número de bits necesarios para representar una determinada variedad de información, puede abordarse el proceso de **codificación**. La codificación consiste en asignar, a cada elemento del conjunto que queremos codificar, una determinada combinación de ceros y unos. Este proceso conduce a la obtención de un *código*, que queda definido por el conjunto completo de asociaciones realizadas entre elementos codificados y combinaciones de ceros y unos.

Un código puede definirse mediante una tabla que indique explícitamente la asociación entre los elementos del conjunto codificado y las combinaciones (**palabras**) del código; también puede definirse mediante un enunciado que defina la regla de formación de las palabras del código: el código binario natural, por ejemplo, se define así –cada número natural se codifica mediante su expresión en el sistema numérico posicional en base 2.

Para la creación de un código deben seguirse los siguientes criterios:

[1] Un bit, término que en la ingeniería de sistemas digitales se utiliza en ocasiones como sinónimo de *señal digital*, es en realidad una medida de *cantidad de información*: es la máxima información que puede representar una señal digital capaz de tomar dos valores.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

- *A cada elemento del conjunto codificado le debe corresponder una combinación diferente de ceros y unos*.

  Esta regla, de obligado cumplimiento, garantiza que se pueda **descodificar** una palabra del código: si dos elementos tuvieran asociada la misma combinación, no sería posible determinar el valor que ésta codifica.

  Ejemplo: Si en la codificación de un conjunto de colores, con tres bits, asignamos la combinación 001 al verde y al rojo, no podemos determinar, a partir de esta combinación, el color codificado. La descodificación correcta sería: *001 => es verde o rojo*. Si esto es admisible se debe a que hemos determinado incorrectamente el conjunto de valores a codificar; los dos elementos (verde, rojo), se reducen a uno, (verde o rojo).

- *La longitud de las palabras del código debe minimizarse.*

  Esta es una condición deseable. Utilizar un número de bits mayor que el mínimo necesario no ofrece, en la mayor parte de los casos, ninguna ventaja y, en cambio, puede dar lugar a ineficiencias a la hora de diseñar circuitos que procesen la información codificada –porque tiene un formato innecesariamente complejo. Pero esto no es siempre cierto: en ocasiones, raras, el uso de un exceso de bits puede resultar conveniente y facilitar el diseño de circuitos más simples.

  Ejemplo: Para codificar los dígitos decimales, del 0 al 9, basta con utilizar 4 bits. Utilizar 5 bits, por ejemplo, resulta superfluo e inconveniente.

- *La asignación de combinaciones a valores del conjunto de elementos codificados debe facilitar los procesos de codificación y descodificación*.

  En las tareas relacionadas con la ingeniería de sistemas digitales, las operaciones de codificación y descodificación se realizan de manera intensiva, por lo que resulta muy importante que puedan ejecutarse eficazmente. Este criterio puede resultar, en ocasiones, contradictorio con el anterior: la construcción de un código que permita interpretar fácilmente el valor representado por una combinación de ceros y unos puede verse facilitado por el uso de un exceso de bits.

  En muchas ocasiones el conjunto de valores a codificar tienen una relación de orden o, implícitamente, pueden reducirse a una secuencia de números: en estos casos resulta conveniente recurrir a un código numérico estándar (normalmente al binario natural) para construir el código.

  Ejemplo: Asignar a los meses del año números correlativos facilita el cálculo de las palabras del código y su rápida interpretación: es fácil recordar que al mes de marzo le corresponde el número tres; y desde el punto de vista de la decodificación es fácil recordar que el mes con el número cinco es el quinto mes del año, es decir, Mayo

Resulta necesario advertir que, en ocasiones, los dos últimos criterios pueden obviarse, debido al procesamiento específico a que vaya a ser sometida la información.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE II. Ejercicios con Solución Desarrollada.

A continuación se desarrollan dos ejercicios en los que se explicará el procedimiento para codificar y decodificar diferentes conjuntos de elementos. Se resolverán algunos ejemplos variados donde se mostrarán diversas formas de llevar a cabo asignaciones de código en las que se tendrán en cuenta condicionantes de diversa índole.

EJERCICIO A) Codifique los conjuntos de elementos que se indican

A.1) Los días de la semana.

Este es un claro ejemplo de lo que podemos considerar como conjuntos ordenados de elementos. Se trata de conjuntos en los que entre sus elementos se da un cierto orden intrínseco, es decir, los elementos se suceden uno detrás de otro de acuerdo a alguna característica particular (tiempo, tamaño, distancia, valor, etc.). Algunos ejemplos son los días de la semana, los meses del año o las letras del alfabeto. En este tipo de conjuntos lo más razonable es llevar a cabo una codificación mediante la asignación de números consecutivos en orden ascendente.

En nuestro caso, al "lunes" podríamos asignarle el código uno, al "martes" el dos, al "miércoles" el tres y así sucesivamente. Bien entendido que en un sistema digital, en el que solamente podemos trabajar con números binarios, habría que representar este código de acuerdo a algún sistema numérico, el más usual en estos casos es el binario natural. Por lo que la codificación definitiva sería como figura en la primera columna.

Día de la Semana	Codificación Correlativa	Codificación Aleatoria
Lunes	001	011
Martes	010	001
Miércoles	011	101
Jueves	100	100
Viernes	101	111
Sábado	110	010
Domingo	111	110

Este tipo de codificación tiene la ventaja de que el mayor o menor código asignado representa una cierta característica inherente al elemento. En nuestro caso el lugar que ocupa el día en la semana. Esto facilita notablemente la codificación y la decodificación, para comprobarlo bastaría con tratar de memorizar los códigos asignados a cada uno de los días si la codificación fuese la que figura en la segunda columna.

Otra ventaja es que para pasar de un elemento de la lista al siguiente bastaría con incrementar en una unidad del código correspondiente, es decir, el paso del jueves al viernes supone pasar del cuatro al cinco.

A.2) Los planetas del Sistema Solar.

Este es el tipo de conjuntos que son susceptibles de ser ordenados según distintos criterios, según los intereses del procesamiento posterior de los datos. Así podemos pensar en la asignación de un código

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

ascendente según su mayor distancia al sol, o según su tamaño. En la tabla siguiente figuran los códigos asignados según el caso.

PLANETA	CODIFICACIÓN SEGÚN DISTANCIA AL SOL	CODIFICACIÓN SEGÚN DIAMETRO ECUATORIAL
Mercurio	0000	0001
Venus	0001	0011
Tierra	0010	0100
Marte	0011	0010
Júpiter	0100	1000
Saturno	0101	0111
Urano	0110	0110
Neptuno	0111	0101
Plutón	1000	0000

A.3) Cada una de las cinco cámaras para el control del tráfico que ha instalado la DGT en una carretera.

En este ejemplo podríamos considerar razonable una asignación de código correlativo empezando con el uno, como figura en la tercera columna de la tabla, y en muchos casos lo será teniendo en cuenta que de este modo optimizamos el número de bits con los que codificamos (es suficiente la utilización de tres bits). Sin embargo, este procedimiento puede no ser útil según el tipo de procesamiento posterior que se quiera hacer con el elemento. Por ejemplo, podría resultarnos de utilidad para determinada aplicación, que el código identificativo de la cámara fuese el punto kilométrico en el que se ésta se encuentra, codificado en binario natural, como figura en la cuarta columna. En este caso necesitamos ocho bits y podrían ser más si la distancia fuese mayor, por lo que sacrificamos la eficiencia en el número de bits necesarios para codificar la información, en aras de una mayor utilidad del código desde el punto de vista del procesamiento de los datos.

Nº CÁMARA	PUNTO KILOMÉTRICO	CODIFICACIÓN CORRELATIVA	CODIFICACIÓN SEGÚN PUNTO KILOMÉTRICO
1	35	001	00100011
2	60	010	00111100
3	103	011	01100111
4	158	100	10011110
5	203	101	11001011

A.4) Estado civil de una persona *(soltero, casado, separado, viudo, pareja de hecho, otro.)*

En este caso no podemos determinar ninguna característica de orden entre los elementos del conjunto, podemos pensar en otros ejemplos como son, los colores en los que se fabrica un modelo de automóvil, las especies animales que habitan en un parque zoológico, los tipos de madera en los que se

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

dispone de un mueble determinado, o el estado civil de una persona. En estas circunstancias, ante la ausencia de un criterio claro se suele asignar de manera aleatoria los códigos a los distintos elementos. Así pues, podemos codificar el estado civil con un 0 para soltero, un 1 para casado, un 2 para separado, un 3 para viudo, un 4 para pareja de hecho o un 5 para otros casos.

A.5) Equipo encendido o apagado.

En principio podemos pensar que resulta indiferente la asignación del cero a la situación de equipo encendido o de equipo apagado, y del uno para la situación alternativa. Sin embargo, hay que indicar que en este caso tiene lugar una circunstancia de *convenio* o *acuerdo*, en la medida que es algo asumido por la comunidad de profesionales la asignación del código "uno" para reflejar una situación positiva y del "cero" para reflejar lo contrario. Así, podemos encontrar que el 1 lógico se utiliza para indicar aspectos tales como: *sí*, *creciente*, *activado*, *funciona*, *abierto*, *on*,…; mientras que el 0 lógico suele representar la situación contraria, es decir: *no*, *decreciente*, *desactivado*, *no funciona*, *cerrado*, *off*,… En consecuencia, elegimos el "cero" para *equipo apagado* y el "uno" para *equipo encendido*.

EJERCICIO B) Se pide realizar la asignación de código para las cartas de una baraja española.

Se nos plantea codificar una baraja con cuarenta cartas, distribuidas en cuatro palos de diez cartas cada uno, así pues, para llevar a cabo esta codificación necesitaremos, al menos, 6 bits. Otra cuestión es el procedimiento para asignar los números a las cartas. En la segunda y quinta columnas se puede ver el resultado de una asignación correlativa, comenzando desde el cero y acabando en el 39. Sin embargo, podemos efectuar un procedimiento de codificación alternativo, cuyo resultado figura en las columnas tercera y sexta.

Se trata de una codificación en la que utilizan dos bits para codificar el palo al que pertenece la carta (P_1P_0) y otros cuatro bits ($V_3V_2V_1V_0$) para codificar el valor de la carta. Además, se aprecia que no se utiliza el valor cero para ninguna carta y que para la sota, el caballo y el rey se utilizan el diez, el once y el doce, produciéndose un salto desde el valor siete.

Esta manera de proceder presenta varias ventajas. Por una parte resulta muy sencillo localizar el palo al que pertenece la carta, puesto que los dos bits más a la izquierda me ofrecen directamente esa información, cosa que no sucede con la otra codificación. Por otra parte, dos cartas con el mismo valor, dos ases por ejemplo, tienen el mismo código en los cuatro bits más a la derecha del código, esto es de máxima utilidad cuando pretendemos comparar si una carta es mayor o menor que otra. Finalmente, saltarnos los códigos "0000", "1000", y "1001" tiene la ventaja de hacer coincidir el valor de la carta con el código asignado a la misma.

En conclusión, cuando los elementos del conjunto presentan algún tipo de estructura u organización interna, como es el caso que se nos presenta en este ejemplo (elementos que pueden ser clasificados en cuatro grupos de diez cartas según el palo de pertenencia, o en diez grupos de cuatro cartas según el valor de la misma), será oportuno que la forma de codificación refleje esa misma estructura. Esto facilitará los procedimientos de codificación y decodificación, así como el procesamiento posterior que se realice con los datos. Así pues, será común encontrar que los bits con los que codificamos los elementos estén divididos en subconjuntos que, a su vez, dividen el grupo total en subgrupos homogéneos que comparten una determinada característica (oros, copas, ases, reyes, etc.)

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

CARTA	CODIFICACIÓN CORRELATIVA	CODIFICACIÓN POR PALO Y VALOR	CARTA	CODIFICACIÓN CORRELATIVA	CODIFICACIÓN POR PALO Y VALOR
	$C_5C_4C_3C_2C_1C_0$	$P_1P_0V_3V_2V_1V_0$		$C_5C_4C_3C_2C_1C_0$	$P_1P_0V_3V_2V_1V_0$
as de oros	0 0 0 0 0 0	**0 0 0 0 0 1**	as de espadas	0 1 0 1 0 0	**1 0 0 0 0 1**
dos de oros	0 0 0 0 0 1	**0 0** 0 0 1 0	dos de espadas	0 1 0 1 0 1	**1 0** 0 0 1 0
tres de oros	0 0 0 0 1 0	**0 0** 0 0 1 1	tres de espadas	0 1 0 1 1 0	**1 0** 0 0 1 1
cuatro de oros	0 0 0 0 1 1	**0 0** 0 1 0 0	cuatro de espadas	0 1 0 1 1 1	**1 0** 0 1 0 0
cinco de oros	0 0 0 1 0 0	**0 0** 0 1 0 1	cinco de espadas	0 1 1 0 0 0	**1 0** 0 1 0 1
seis de oros	0 0 0 1 0 1	**0 0** 0 1 1 0	seis de espadas	0 1 1 0 0 1	**1 0** 0 1 1 0
siete de oros	0 0 0 1 1 0	**0 0** 0 1 1 1	siete de espadas	0 1 1 0 1 0	**1 0** 0 1 1 1
sota de oros	0 0 0 1 1 1	**0 0** 1 0 1 0	sota de espadas	0 1 1 0 1 1	**1 0** 1 0 1 0
caballo de oros	0 0 1 0 0 0	**0 0** 1 0 1 1	caballo de espadas	0 1 1 1 0 0	**1 0** 1 0 1 1
rey de oros	0 0 1 0 0 1	**0 0** 1 1 0 0	rey de espadas	0 1 1 1 0 1	**1 0** 1 1 0 0
as de copas	0 0 1 0 1 0	**0 1 0 0 0 1**	as de bastos	0 1 1 1 1 0	**1 1 0 0 0 1**
dos de copas	0 0 1 0 1 1	**0 1** 0 0 1 0	dos de bastos	0 1 1 1 1 1	**1 1** 0 0 1 0
tres de copas	0 0 1 1 0 0	**0 1** 0 0 1 1	tres de bastos	1 0 0 0 0 0	**1 1** 0 0 1 1
cuatro de copas	0 0 1 1 0 1	**0 1** 0 1 0 0	cuatro de bastos	1 0 0 0 0 1	**1 1** 0 1 0 0
cinco de copas	0 0 1 1 1 0	**0 1** 0 1 0 1	cinco de bastos	1 0 0 0 1 0	**1 1** 0 1 0 1
seis de copas	0 0 1 1 1 1	**0 1** 0 1 1 0	seis de bastos	1 0 0 0 1 1	**1 1** 0 1 1 0
siete de copas	0 1 0 0 0 0	**0 1** 0 1 1 1	siete de bastos	1 0 0 1 0 0	**1 1** 0 1 1 1
sota de copas	0 1 0 0 0 1	**0 1** 1 0 1 0	sota de bastos	1 0 0 1 0 1	**1 1** 1 0 1 0
caballo de copas	0 1 0 0 1 0	**0 1** 1 0 1 1	caballo de bastos	1 0 0 1 1 0	**1 1** 1 0 1 1
rey de copas	0 1 0 0 1 1	**0 1** 1 1 0 0	rey de bastos	1 0 0 1 1 1	**1 1** 1 1 0 0

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE III. Ejercicios con Solución Breve.

En lo que sigue se presentan algunos enunciados de ejercicios que deberán ser resueltos sin consultar la solución, tratando de poner en práctica los conocimientos sobre codificación adquiridos. Posteriormente se tendrá la oportunidad de contrastar la solución con la que se ofrece y detectar posibles errores. Se ha de tener en cuenta, que en algunos ejercicios hay más de una solución válida y que la decisión para elegir la mejor estará guiada por criterios de eficacia del código utilizado.

EJERCICIO 1) Indique de cuántas teclas consta el teclado de tu ordenador y, por lo tanto, cuántos bits serían necesarios para codificarlas.

EJERCICIO 2) Proponga un esquema de codificación para las matrículas de los vehículos en España.

EJERCICIO 3) Se dispone de un depósito en el que se han colocado cinco sensores de nivel en el lateral, de forma que cuando el sensor es alcanzado por el líquido entrega un “uno”, y un “cero” en caso contrario. Cada sensor está distanciado del siguiente de forma que, en el paso entre uno y otro, el líquido aumenta en cinco litros. ¿Cómo sería el código que entregarían los cinco sensores según se va incrementado el contenido del depósito?

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 4) Cuantos bits serían necesarios para codificar el peso que mide una báscula con un rango entre 0 y 200Kg, con una precisión de 100gr.

EJERCICIO 5) Cuántos bits serán necesarios para almacenar la información de una fotografía digital de 640x480 pixels, en los que cada pixel se codifica con el nivel de rojo, verde y azul, en ocho bits cada uno.

EJERCICIO 6) Indique como llevaría a cabo la codificación de la temperatura en un recinto, en un rango entre 10º y 35º centígrado. Con una resolución de un grado.

EJERCICIO 7) Indique como efectuaría la codificación del movimiento de subida y bajada del cristal de la ventanilla de un coche.

EJERCICIO 8) Proponga un esquema de codificación para las provincias y Comunidades Autónomas españolas

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE IV. Soluciones a los ejercicios con Solución Breve.

EJERCICIO 1)

El teclado con el que estoy escribiendo este texto dispone de 102 teclas, por lo que necesitaría 7 bits para codificar las teclas.

EJERCICIO 2)

Se propone codificar por una parte el número y cada una de las tres letras por separado. La matrícula dispone de 4 cifras decimales, por lo que el número a codificar estará comprendido entre 0000 y 9999, así pues, necesitaremos 14 bits. Las letras válidas en las matrículas son 21, por lo que necesitaremos 5 bits para codificar cada una. Por lo tanto, el esquema será el siguiente:

$$N_{13}N_{12}N_{11}N_{10}N_9N_8N_7N_6N_5N_4N_3N_2N_1N_0C_4C_3C_2C_1C_0B_4B_3B_2B_1B_0A_4A_3A_2A_1A_0$$

EJERCICIO 3)

Hemos de considerar que, cuando un sensor el alcanzado por el líquido, también lo son todos los que están por debajo. Así pues, el código entregado por el conjunto de los sensores será el siguiente.

Nº Sensor Cubierto	Litros Almacenados	Codificación Entregada
		$S_5S_4S_3S_2S_1$
Ninguno	0 litros	0 0 0 0 0
1	5 litros	0 0 0 0 1
2	10 litros	0 0 0 1 1
3	15 litros	0 0 1 1 1
4	20 litros	0 1 1 1 1
5	25 litros	1 1 1 1 1

EJERCICIO 4)

El número total de décimas de kilogramo, o centenares de gramos, que hay es de 200x10 = 2000. Para codificar esta cantidad de niveles necesitamos 11 bits (que permitiría alcanzar un máximo de 2048 posibles niveles)

EJERCICIO 5)

Para codificar los niveles de RVA de cada uno de los pixels de la fotografía se necesitan 8x3 = 24 bits. Puesto que la fotografía completa tiene 640x480 = 307200. Para la fotografía necesitamos pues 7.372.800 bits

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 6)

Tenemos dos alternativas, codificar exactamente el número de grados que componen el rango, es decir, 25 niveles. Esto nos daría un abanico de valores correlativos desde 00000 hasta 11001. O bien, codificar de manera que el código asignado sea el valor exacto de la temperatura del recinto, con lo cual el rango de valores oscilaría entre 001010 hasta 100011. Como se ve esta segunda opción tiene la desventaja de necesitar un bit más, pero la ventaja de indicar la temperatura precisa que hay en el recinto sin un procesamiento adicional.

EJERCICIO 7)

Por lo explicado en los ejemplos resueltos anteriormente presentados, lo más oportuno sería codificar con un "uno" la situación de cristal subiendo y con un "cero" la contraria, es decir, cristal bajando.

EJERCICIO 8)

En España hay un total de 19 comunidades autónomas y un conjunto global de 52 provincias y ciudades autónomas. La C.A. con más provincias es Castilla y León con un total de nueve, por otra parte hay varias C.A. con una sola provincia.

Un esquema de codificación simple es codificar las provincias por orden alfabético con seis bits, utilizando los códigos desde el 000000 hasta el 110011.

Una forma alternativa que, aunque necesita mayor cantidad de bits, permite una mejor organización y una más simple codificación y decodificación, consiste en codificar por una parte la C.A y por otra el número de provincia dentro de la comunidad.

Para codificar las 19 C.A. necesitamos 5 bits y podemos asignar un código por orden alfabético desde el 00000 hasta el 10010. Para codificar la provincia dentro de la comunidad, también asignado por orden alfabético, el número de bits viene determinado por aquella más numerosa en provincias. Así pues necesitaríamos 4 bits para codificar las 9 provincias de Castilla y León.

Así pues el esquema sería el siguiente: $CA_4CA_3CA_2CA_1CA_0P_3P_2P_1P_0$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P7
Representación de Números con Signo

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Números en Complemento a 2

- Complemento a 2 (C2) para codificar números con signo
- Los números positivos se codifican igual que en binario natural. El bit más significativo ha de ser '0'.

Dec	C2
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111

Dec	C2
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Números negativos en Complemento a 2

- En complemento a 2, el rango de números enteros que se pueden codificar con 'n' bits es el siguiente: $(-2^{n-1}, 2^{n-1}-1)$
- Ejemplos:
  - Con 4 bits: $(-2^{4-1}, 2^{4-1} - 1) = (-8, +7)$
  - Con 6 bits: $(-2^{6-1}, 2^{6-1} - 1) = (-32, +31)$
- Rangos en binario natural y C2 con 10 bits
  - BN $\Rightarrow (0, 2^{10} - 1) = (0, 1023)$
  - C2 $\Rightarrow (-2^{9}, 2^{9} - 1) = (-512, 511)$

BT1_S3 BT1_P7_3

Números negativos en Complemento a 2

- Procedimiento para codificar un número negativo:
- Invertir e incrementar
  1. Se codifica en binario natural el módulo del número negativo. El bit más significativo ha de ser '0'.
  2. Se invierten todos los bits.
  3. Se incrementa en uno el número resultante.

BT1_S3 BT1_P7_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Números negativos en Complemento a 2

- Ejemplo, codificar en C2 el número -25 en 6 bits
  1. $01\ 1001_B$ ($+25_D$)
  2. 10 0110 (invertir bits)
  3. $10\ 0111_{C2}$ (incrementar)

BT1_S3 BT1_P7_5

Números negativos en Complemento a 2

- Otro procedimiento para codificar un número negativo:
- Invertir después del primer '1':
  1. Se codifica en binario natural el módulo del número negativo. El bit más significativo ha de ser '0'.
  2. Se invierten todos los bits que hay después del primer '1' que se encuentre empezando a buscar por la derecha.

BT1_S3 BT1_P7_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Números negativos en Complemento a 2

- Ejemplo, codificar en C2 el número -20 en 6 bits
  1. $01\ 0100_B$ ($+20_D$)
  2. $10\ 1100_{C2}$ (invertir bits)
- Ambos procedimientos también se pueden emplear para cambiar de signo a cualquier número en C2.

BT1_S3 BT1_P7_7

Números negativos en Complemento a 2

- Siempre hay que tener en cuenta el rango de números que se pueden codificar con n bits:

$$(-2^{n-1}, 2^{n-1}-1)$$

- Ejemplo problemático, codificar -8 con 4 bits:
  - Se codifica en binario natural: 1000
  - Se invierten todos los bits después del primer '1': 1000
  - ¿Qué sucedió? ¿-8 = 8?
  - Sucede que para codificar el +8 hacen falta 5 bits:
    - 01000 (+8) $\Rightarrow$ 11000 (-8)

BT1_S3 BT1_P7_8

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Números negativos en Complemento a 2

- Ejemplos sencillos
  1. Expresa el número -93 en C2
  2. ¿Qué número decimal es el número 101001 en C2?
  3. Cambia de signo el número 11010.
- Solución
  1. 64+16+8+4+1 = 01011101 -> 10100011
  2. 010111 = 16 + 4 + 2 + 1 = 23 $\Rightarrow$ -23
  3. 00101 -> 00110 (+6)

BT1_S3 BT1_P7_9

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P8

Suma y resta de números en binario natural

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Suma en Binario Natural

- Suma de 2 dígitos en Binario Natural

$$\begin{array}{r} X \\ + \quad Y \\ \hline C \quad S \end{array}$$

Acarreo "Carry" → C

$$\begin{array}{r} 0 \\ + \quad 0 \\ \hline 0 \quad 0 \end{array} \qquad \begin{array}{r} 1 \\ + \quad 0 \\ \hline 0 \quad 1 \end{array} \qquad \begin{array}{r} 0 \\ + \quad 1 \\ \hline 0 \quad 1 \end{array} \qquad \begin{array}{r} 1 \\ + \quad 1 \\ \hline 1 \quad 0 \end{array}$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Suma en Binario Natural

- Suma de 2 dígitos en Binario Natural con acarreo

$$\begin{array}{r} C_I \\ X \\ + \quad Y \\ \hline C_O \; S \end{array}$$

Acarreo "Carry" → C_O

$$\begin{array}{r} 1 \\ 0 \\ + \; 0 \\ \hline 0\;1 \end{array} \qquad \begin{array}{r} 1 \\ 1 \\ + \; 0 \\ \hline 1\;0 \end{array} \qquad \begin{array}{r} 1 \\ 0 \\ + \; 1 \\ \hline 1\;0 \end{array} \qquad \begin{array}{r} 1 \\ 1 \\ + \; 1 \\ \hline 1\;1 \end{array}$$

Suma en Binario Natural

- Suma de 2 números de 'n' bit
  - Procedimiento análogo a la suma en decimal

Acarreo Inicial

Suma con 8 bits

Acarreos

$$\begin{array}{r} 0\;1\;0 \\ 2\;1\;9 \\ + \; 1\;2\;3 \\ \hline 3\;4\;2 \end{array} \qquad \begin{array}{r} 1\;1\;1\;1\;0\;1\;1\;0 \\ 1\;1\;0\;1\;1\;0\;1\;1 \\ + \; 0\;1\;1\;1\;1\;0\;1\;1 \\ \hline 1\;0\;1\;0\;1\;0\;1\;1\;0 \end{array}$$

Acarreo Final → 1 0 1 0 1 0 1 1 0

Se dice que se ha generado un acarreo

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Suma en Binario Natural

- Desbordamiento
  - El Acarreo Final es 1 cuando el resultado de la suma no se puede expresar con el mismo número de bits que los sumandos.
  - Al producirse un desbordamiento en la suma en Binario Natural hace falta un bit más para poder expresar el resultado. El bit de acarreo final también realiza esta función.

BT1_S3 BT1_P8_5

Resta en Binario Natural

- Resta de 2 dígitos en Binario Natural

$$\begin{array}{r} X \\ -\ Y \\ \hline R \end{array}$$

Minuendo ≥ Sustraendo

$$\begin{array}{r} 0 \\ -\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ -\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ -\ 1 \\ \hline ? \end{array} \qquad \begin{array}{r} 1 \\ -\ 1 \\ \hline 0 \end{array}$$

BT1_S3 BT1_P8_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Resta en Binario Natural

- Resta de 2 números de 'n' bits
  - Procedimiento análogo a la resta en decimal

Acarreo Inicial

Resta con 8 bits

Acarreos o "Borrows"

2 1 9	1 1 0 1 1 0 1 1
- 1 0 0	- 1 1 0 0 0 0 0 0
1 2 1	0 1 1 1 1 0 0 1
0 9 8	0 1 1 0 0 0 1 0

Minuendo ≥ Sustraendo

Resta en Binario Natural

- Resta de 2 dígitos en Binario Natural con acarreo

Acarreo o "Borrow" → B_O

$$\begin{array}{r} X \\ -\ B_I \\ Y \\ \hline R \end{array}$$

0	0	1	0	1
1 0	1 1	0 1	1 1	1 1
- 1	- 0	- 0	- 1	- 1
1	1	0	0	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P9

Suma y resta de números en complemento a dos

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Suma en Complemento a 2

- Suma de 2 números expresados en complemento a 2
  - No se hace un tratamiento distinto del bit de signo
  - Para obtener el resultado de la suma se emplea el mismo procedimiento que la suma en Binario Natural

Suma con 4 bits

	0 0 1 0 (+2)		1 1 1 0 (-2)
+	0 1 0 0 (+4)	Se ignora el acarreo	+ 1 1 0 0 (-4)
	0 1 1 0 (+6)		1 1 0 1 0 (-6)

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Suma en Complemento a 2

- Desbordamiento u "Overflow": Si la suma excede el rango de valores para números de 'n' bits en complemento a 2.
- La ocurrencia de Desbordamiento u "Overflow" sucede cuando el signo de la suma es diferente al de los sumandos.

Suma con 4 bits

```
  0 1 0 0 (+4)
+ 0 1 0 1 (+5)
-------------
  1 0 0 1 (-7)
```

Se ignora el acarreo

```
  1 1 0 1 (-3)
+ 1 0 1 0 (-6)
---------------
1 0 1 1 1 (+7)
```

BT1_S3 BT1_P9_3

Resta en Complemento a 2

- La resta de 2 números expresados en complemento a 2
  - R = X – Y = X + (-Y) = X + C2(Y)
  - No se hace un tratamiento distinto del bit de signo
  - Se suma como en Binario Natural

Ejemplo: (+2) – (+4)

Resta con 4 bits

Se ignora el acarreo

```
  0 0 1 0 (+2)
- 0 1 0 0 (+4)
-------------
1 1 1 1 0 ( -2)
```

```
  0 0 1 0 (+2)
+ 1 1 0 0 ( -4)
---------------
  1 1 1 0 ( -2)
```

BT1_S3 BT1_P9_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Resta en Complemento a 2

- Desbordamiento u "Overflow": Si la resta excede el rango de valores para números de 'n' bits en complemento a 2.
- La ocurrencia de Desbordamiento u "Overflow" sucede cuando los signos de los operandos son distintos y el signo de la resta es diferente al signo del minuendo.

(+)	(+)	(+)	(+)	(-)	(-)	(-)	(-)
- (+)	- (+)	- (-)	- (-)	- (+)	- (+)	- (-)	- (-)
(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)
Posible	Posible	Posible	Imposible	Imposible	Posible	Posible	Posible

BT1_S3 BT1_P9_5

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Resta en Complemento a 2

- Desbordamiento u "Overflow": Si la resta excede el rango de valores para números de 'n' bits en complemento a 2.
- La ocurrencia de Desbordamiento u "Overflow" sucede cuando los signos de los operandos son distintos y el signo de la resta es diferente al signo del minuendo.

Resta con 4 bits

0 1 0 0 (+4)		1 1 0 1 (-3)
- 1 0 1 1 (-5)	Se ignora el acarreo	- 0 1 1 0 (+6)
1 0 0 1 (-7)		1 0 1 1 1 (+7)

BT1_S3 BT1_P9_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Acarreo y Overflow

- IMPORTANTE:
  - El acarreo tiene sentido utilizarlo en las operaciones de suma y resta de números SIN SIGNO.
  - El Overflow tiene sentido emplearlo en las operaciones CON SIGNO.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P10

Multiplicación y División. Extensión de signo. Truncamiento y Redondeo

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Multiplicación y División por potencias de 2

- Multiplicación por 2^n

$$0\,0\,1\,0\,1\,0 \begin{cases} \xrightarrow{\cdot\, 2^1} 0\,1\,0\,1\,0\,0 \\ \xrightarrow{\cdot\, 2^2} 1\,0\,1\,0\,0\,0 \end{cases}$$

- División por 2^n

BN $$0\,0\,1\,0\,1\,0 \begin{cases} \xrightarrow{\div\, 2^1} 0\,0\,0\,1\,0\,1 \\ \xrightarrow{\div\, 2^2} 0\,0\,0\,0\,1\,0 \end{cases}$$

C2 $$1\,1\,1\,0\,1\,0 \begin{cases} \xrightarrow{\div\, 2^1} 1\,1\,1\,1\,0\,1 \\ \xrightarrow{\div\, 2^2} 1\,1\,1\,1\,1\,0 \end{cases}$$

Extensión de signo

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Extensión de signo

- Se trata de expresar una cantidad con más bits.
  - Si el número es positivo o cero se añaden tantos 0s como sean necesarios a su izquierda.
  - Si el número es negativo se añaden tantos 1s como sean necesarios a su izquierda.
- Ejemplos:
  - Exprese 01001 (+9) en 8 bits ⇒ 0000 1001
  - Exprese 10111 (-9) en 10 bits ⇒ 11 1111 0111

Truncamiento

- Sirve para reducir el número de cifras necesarias para expresar una cantidad dada.
- Elimina las cifras a la derecha de una dada.
  - Ejemplos
    - $27\,000_D$ se puede expresar, en cientos, como 270_D
    - $27\,485_D$, en cientos, es 274. Se comete un error.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Truncamiento

- Números con 'n' cifras, un número X se trunca a 'm' cifras ($n \geq m$) eliminando las n-m cifras menos significativas.
  - Truncar los números:
    - $123\ 456\ 789_D$ a 5 cifras $\Rightarrow 12\ 345_D$
    - $1001\ 0110_B$ a 6 bits $\Rightarrow 10\ 0101_B$
    - $1\ 899_D$ a 2 cifras $\Rightarrow 18$ ¿No sería mejor aproximar a 19_D?

Redondeo

- El truncamiento aproxima por abajo y no es siempre lo mejor.
- El redondeo es otra forma de reducir el número de cifras.
- El redondeo a 'm' cifras de un número X, expresado en base 'b', con n cifras ($n \geq m$) se aproxima truncando a m cifras a la cantidad:

$$X + \frac{b^{n-m}}{2}$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Redondeo

- Ejemplos
  - $1\,899_D$ se puede redondear, en cientos, como

$$1899 + \frac{10^{4-2}}{2} = 1949 \Rightarrow 19$$

1001 0110 se puede redondear a 5 bits, como

$$1001\,0110 + \frac{2^{8-5}}{2} = 1001\,0110 + 100 = 1001\,1010 \Rightarrow 1\,0011$$

- El redondeo permite aproximar al entero más próximo tanto por exceso como por defecto.

BT1_S3 BT1_P10_7

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre conversión entre códigos.**	**CÓDIGO:** **BT1_AINP7**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	60 minutos

CALENDARIO:	Tercera Sesión. Previo a la Cuarta Sesión	REQUISITOS:	Binario natural

CRITERIO DE ÉXITO:	Realización correcta de al menos 20 de las 30 conversiones

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS SOBRE CONVERSIÓN ENTRE CÓDIGOS

PARTE I. Conceptos Teóricos.

El código binario natural y el código BCD permiten codificar números naturales (los números positivos y el cero). Sin embargo, en muchas aplicaciones digitales es necesario trabajar con números enteros con signo, por lo que es necesario representar números positivos y negativos. El código utilizado normalmente para la representación de estos números es el código **complemento a dos**, aunque existen otros que, bien han caído en desuso (como el código **complemento a uno**), o bien se usan sólo en raras ocasiones (como el código **signo y magnitud**). El código complemento a dos, que es el que se va a describir aquí, tiene una estructura más compleja que los códigos complemento a uno y signo y magnitud, pero su uso facilita enormemente la construcción de circuitos digitales que realizan operaciones aritméticas; por este motivo es el utilizado en la práctica totalidad de los sistemas digitales.

Estructura del código complemento a 2

De los **n** bits que componen la expresión de un número en **complemento a dos**, **n-1** bits representan el módulo del número, y el bit restante su signo. El **bit de signo** es el primero por la izquierda; de acuerdo con esto, la estructura de los **n** bits de un número codificado en complemento a dos es:

$$\mathbf{S\ M_{n-2}\ M_{n-2}\ M_{n-3}\ldots\ldots\ldots M_1\ M_0}$$

Donde **S** es el bit de signo y $\mathbf{M_i}$ los bits que codifican el módulo

Codificación de números en complemento a 2

Signo

El valor del **bit de signo** para los números positivos y el cero es **0**, para los números negativos vale **1**.

Módulo

La codificación del módulo del número se realiza de diferente manera según sea positivo o negativo. Si se utilizan **n** bits para representar el número, y se dispone por tanto de **n-1** bits para la codificación del **módulo**, el procedimiento a seguir es el siguiente:

- Cuando el número es **positivo**, su módulo se codifica expresándolo directamente en binario natural con n-1 bits.
  - Obsérvese que, por tanto, con **n** bits el mayor número positivo que se puede codificar en complemento a 2 es $\mathbf{2^{n-1}-1}$.
  - Por ejemplo, el número +12, en complemento a 2, utilizando 6 bits se compondría de la siguiente manera:
    - Bit de signo: 0
    - Módulo: 12 en binario natural en (6-1) 5 bits: 0 1100
    - +12 en complemento a 2 en 6 bits: 00 1100
  - Con 6 bits el mayor número positivo que puede representarse en código complemento a 2 es el +31 (2^5-1): 01 1111

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

- Cuando el número es **negativo**, se codifica expresando en binario natural el **complemento a 2** con **n-1** bits del módulo del número. Siendo el **complemento a 2 con k bits** de un número positivo, **X**, otro número positivo de valor $\mathbf{2^k-X}$; por tanto, X debe ser menor o igual que 2^k.
    - El **complemento a 2 con n-1 bits** del módulo se calcula como: $\mathbf{2^{n-1}-}$ **módulo**
    - Por ejemplo, si el número a codificar es el -12, sabemos que su módulo es 12, por lo que el complemento a 2 con 5 bits vale: $2^5-12 = 20$
    - El número -12 se codificaría con 6 bits en complemento a 2, de la siguiente manera:
        - Bit de signo: 1
        - Complemento a 2 del módulo, 12, en 5 bits (6-1 bits): $2^5-12 = 20$; que en binario natural es el 1 0100
        - Así pues, el número -12 en complemento a 2 en 6 bits se representa como 11 0100
    - El complemento a 2 con n-1 bits del módulo ($\mathbf{2^{n-1}-}$ **módulo**) solo puede calcularse para números negativos cuyo módulo esté entre **1** y $\mathbf{2^{n-1}}$; por tanto, el número negativo de mayor módulo que puede representarse con n bits es $\mathbf{-2^{n-1}}$.
    - El número negativo de menor valor que puede codificarse en complemento a 2 con 6 bits es el -32 (-2^5), cuyo código es ($2^5-32=0$) 10 0000

Rango de codificación

El rango de valores numéricos que pueden codificarse con n bits va desde -2^{n-1} hasta $+2^{n-1}-1$. Observe que pueden codificarse menos números positivos que negativos porque el cero se codifica como positivo (bit de signo 0).

- Para codificar el número +16 se precisan como mínimo 6 bits: el bit de signo y cinco adicionales para el módulo, ya que con 4 bits para la codificación del módulo sólo se puede representar hasta el +15 ($+2^4-1$). El código del número +16 en 6 bits sería 01 0000
- Para codificar el número -16 se precisan como mínimo 5 bits: el bit de signo y cuatro adicionales para el módulo, ya que con 4 bits se puede representar hasta el -16 (-2^4). El código del número -16 en 5 bits sería 10000
- En complemento a 2 pueden representarse:
    - Con 4 bits desde el -8 (-2^3) hasta el +7 ($+2^3-1$) (1000 al 0111)
    - Con 8 bits desde el -128 (-2^7) hasta el +127 ($+2^7-1$) (1000 0000 al 0111 1111)

Decodificación

Puesto que el procedimiento de codificación en código complemento a dos es distinto para los números positivos y negativos, a la hora de decodificar una determinada combinación también hay que proceder de diferente forma con los números positivos y negativos:

- En primer lugar, hay que observar el valor del bit de signo para determinar el procedimiento de cálculo del módulo del número: si vale 0 se decodificará como un número positivo, si vale 1 como negativo.
- Si el número es positivo, se calcula el valor del módulo interpretando los bits correspondientes en binario natural.
    - 0 1001: Bit de signo 0 , Módulo 1001 -> +9
    - 0 1000 0111: Bit de signo 0, Módulo 1000 0111 -> +135

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

- Si el número es negativo, se calcula el valor del módulo interpretando los bits correspondientes en binario natural y operando, con el valor obtenido, para realizar la operación inversa a la codificación: módulo = 2^{n-1}-$valor_{BN}$.
    - 1 1001: Bit de signo 1, $Valor_{BN}$ 1001 = 9, Módulo 2^4-9 = 7 -> -7
    - 1 1000 0111: Bit de signo 1, $Valor_{BN}$ 1000 0111 = 135, Módulo 2^8-135 = 121 -> -121

Extensión del número de bits

Una operación frecuente en la codificación de números es la obtención de su expresión con un número de bits **n**, partiendo de una codificación con un menor número de bits –digamos **m**. En código binario natural esta operación se realiza añadiendo **n-m** ceros a la izquierda de la palabra código de **m** bits.

- Para extender la codificación del 14 en binario natural desde 4 a 8 bits: 1110 -> 0000 1110

En código complemento a 2, para pasar de **m** a **n** bits, se procede de la siguiente manera:

- Se añaden **n-m** bits a la izquierda de valor igual al bit de signo del número
    - Para extender la codificación del +7 desde 4 hasta 8 bits: 0111 -> 0000 0111
    - Para extender la codificación del -7 desde 4 hasta 8 bits: 1001 -> 1111 1001
- A esta operación de expansión del número de bits en complemento a dos se la conoce comúnmente como **extensión de signo**.

Interpretación de códigos numéricos

Una ristra de bits que representa un número sólo es interpretable si se conoce el código empleado en la codificación del número. En cualquier contexto de aplicación de un código debe estar claramente definido cuál se está empleando: binario natural, complemento a dos, BCD, o cualquier otro. En la tabla siguiente se muestra las dieciséis posibles combinaciones que podemos realizar con cuatro bits y su correspondiente equivalencia decimal como números en binario natural y en complemento a dos.

Combinación binaria	Decimal B.N.	Decimal c.a.2
0000	+0	+0
0001	+1	+1
0010	+2	+2
0011	+3	+3
0100	+4	+4
0101	+5	+5
0110	+6	+6
0111	+7	+7
1000	+8	-8
1001	+9	-7
1010	+10	-6
1011	+11	-5
1100	+12	-4
1101	+13	-3
1110	+14	-2
1111	+15	-1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Consideraciones finales

El proceso de codificación y decodificación de números negativos en complemento a 2 puede resultar un poco latoso –no ocurre lo mismo con los positivos, pues se tratan como en binario natural- cuando resulta necesario realizarlo sobre un buen número de datos y con rapidez, por lo que es habitual el uso de procedimientos heurísticos que evitan la necesidad de echar la cuenta del complemento a 2 (2^{n-1}- X). En los siguientes ejercicios se presentan algunas de ellas: es recomendable que las conozca y elija alguna –la que prefiera- para agilizar sus cálculos.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicios con Solución Desarrollada.

EJERCICIO A) Se desea codificar los números +24 y -24 utilizando código en complemento a dos.

Para ello se debe determinar en primer lugar el número de bits que son necesarios para llevar a cabo esta codificación. Según la regla anteriormente expuesta, sabemos que con cuatro bits el rango de números codificables va desde el -8 hasta el +7, que con cinco bits va desde el -16 hasta el +15 y que con seis bits podemos representar números entre el -32 y el +31. Así pues, se concluye que con seis o más bits se podrán representar los números que se nos pide. La opción de seis bits es la que optimiza el número de bits utilizados.

Para representar el +24: por tratarse de un número positivo se codifica con el bit de signo a 0 y el módulo en binario natural con 5 bits (1 1000). Sería, por tanto, el 01 1000.

En cuanto al -24: por tratarse de un número negativo su bit de signo vale 1, y su módulo debe representarse como su complemento a dos en 5 bits: $M = 2^5 – 24 = 32 – 24 = 8$ (0 1000). El número -24 se codifica con la cadena de bits 10 1000.

Una forma alternativa para la codificación de un número negativo en complemento a 2 con n bits –repetimos con el -24 en 6 bits- es la siguiente:

- Se parte de la codificación de su módulo en **binario natural** con n bits
  - -24 -> 24; 24 en binario natural con 6 bits: 01 1000
- Se cambian ceros por unos, y unos por ceros (se invierten todos sus bits).
  - 01 1000 -> 10 0111
  - Con la inversión de los bits se obtiene lo que se conoce como el “complemento a uno” del número.
- Se suma 1 a esta cantidad, con lo que se obtiene el código del número negativo en complemento a dos.
  - 10 0111 + 1 = 10 1000

Otra forma rápida de calcular la expresión de un número negativo en complemento a dos consiste en:

- Se parte de su codificación en **binario natural** con n bits
  - -24 -> +24; 24 en binario natural con 6 bits: 01 1000
- La combinación obtenida se altera del siguiente modo: comenzando por la derecha, se dejan los bits como están hasta que se encuentra el primer uno –éste también se deja como está-, y a partir de aquí se invierten unos y ceros
  - 01 1000 -> **10** 1000

Si se quisiera representar este mismo número con una mayor cantidad de bits se procedería a **extender el signo**. Por ejemplo con 10 bits, el número -24 se representaría como:

- **11 11**10 1000

En el caso del +24, también es fácil obtener su expresión con más bits; con 8, por ejemplo:
- **00**01 1000

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO B) Se desea decodificar los números 01001 y 10111 sabiendo que están expresados en complemento a dos.

Para decodificar un número positivo (aquel cuyo bit de signo es cero), basta con sumar los pesos de aquellos dígitos que valen la unidad (como en el sistema binario natural). Así pues, en el primero de los números se tiene que:

- 0 1001 -> $1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 1 = +9$

En el caso del número negativo se puede evitar el cálculo del complemento a dos, de la siguiente manera:

- Se computa la suma de los pesos de los bits, como en binario natural e incluyendo al bit de signo, pero el peso de éste se resta:

- 10111 -> $-1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 1 = -16 + 4 + 2 + 1 = -9$

Otro procedimiento rápido para obtener el módulo de un número negativo es éste:

- Se invierten todos los bits de la combinación.
    - 10111 -> 01000
- Se suma 1 y se decodifica en binario natural
    - 01000 + 1 = 01001 = 9
- Luego el número es el -9

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE II. Ejercicios con Solución Breve.

En la tabla se presenta un conjunto de ejercicios que deberán ser resueltos sin consultar la solución, tratando de poner en práctica los conocimientos adquiridos sobre conversión entre los distintos códigos que se han abordado hasta el momento. Posteriormente se tendrá la oportunidad de contrastar la solución con la que se ofrece y detectar posibles errores.

Tenga en cuenta que alguno de los ejercicios puede no tener y que la interpretación de los subíndices debe hacerse de la siguiente manera:

B: binario natural
D: decimal
C2: complemento a dos
H: hexadecimal
BCD: BCD
X-n: el código indicado por X con el número de dígitos expresado por n
X-MIN: el código indicado por X con el número mínimo de dígitos necesarios

Nº	Código Inicial	Código Final
1	$1100\ 0011_{B}$	H-4
2	$1111\ 0001\ 0000_{B}$	H
3	$1000\ 1001\ 0000_{BCD}$	D-4
4	$0101\ 0110\ 0110_{BCD}$	D
5	$0000\ 1111_{H}$	D
6	$AF3_{H}$	D
7	$1000\ 1111_{C2-8}$	D
8	$1111\ 0011\ 1000_{C2-12}$	D
9	$010\ 0011\ 1000_{C2-11}$	D
10	315_{D}	H-4
11	315_{D}	BCD-4
12	-315_{D}	C2-9
13	188_{D}	H-MIN
14	-66_{D}	C2-MIN
15	FE_{H}	BCD-2

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Nº	Código Inicial	Código Final
16	230_H	BCD-4
17	-129_D	C2-MIN
18	-15_D	C2-7
19	$1001\ 1110_B$	D
20	$1001\ 0000\ 0000\ 1001_{BCD}$	D
21	FAC_H	B
22	$21AF_H$	B-14
23	$1101\ 0110\ 0110_{C2}$	D
24	$01\ 0011\ 0001_{C2}$	D
25	$4\ 227_D$	B-16
26	$43\ 690_D$	H-4
27	-33_D	C2-MIN
28	16_D	C2-6
29	77_D	B-8
30	77_D	BCD-4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE III. Soluciones a los ejercicios con Solución Breve.

Nº	Código Inicial	Código Final
1	$1100\ 0011_{B}$	$00C3_{H-4}$
2	$1111\ 0001\ 0000_{B}$	$F10_{H}$
3	$1000\ 1001\ 0000_{BCD}$	0890_{D-4}
4	$0101\ 0110\ 0110_{BCD}$	566_{D}
5	$0000\ 1111_{H}$	4369_{D}
6	$AF3_{H}$	2803_{D}
7	$1000\ 1111_{C2-8}$	-113_{D}
8	$1111\ 0011\ 1000_{C2-12}$	-200_{D}
9	$010\ 0011\ 1000_{C2-11}$	568_{D}
10	315_{D}	$013B_{H-4}$
11	315_{D}	$0000\ 0011\ 0001\ 0101_{BCD-4}$
12	-315_{D}	IMPOSIBLE $_{C2-9}$
13	188_{D}	BC_{H-MIN}
14	-66_{D}	10111110_{C2-MIN}
15	FE_{H}	IMPOSIBLE $_{BCD-2}$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Nº	Código Inicial	Código Final
16	230_H	$0000\ 0101\ 0110\ 0000_{BCD\text{-}4}$
17	-129_D	$1\ 0111\ 1111_{C2\text{-}MIN}$
18	-15_D	$111\ 0001_{C2\text{-}7}$
19	$1001\ 1110_B$	158_D
20	$1001\ 0000\ 0000\ 1001_{BCD}$	$9\ 009_D$
21	FAC_H	$1111\ 1010\ 1100_B$
22	$21AF_H$	$10\ 0001\ 1010\ 1111_{B\text{-}14}$
23	$1101\ 0110\ 0110_{C2}$	-666_D
24	$01\ 0011\ 0001_{C2}$	$+305_D$
25	$4\ 227_D$	$0001\ 0000\ 1000\ 0011_{B\text{-}16}$
26	$43\ 690_D$	$AAAA_{H\text{-}4}$
27	-33_D	$101\ 1111_{C2\text{-}MIN}$
28	16_D	$01\ 0000_{C2\text{-}6}$
29	77_D	$0100\ 1101_{B\text{-}8}$
30	77_D	$0000\ 0000\ 0111\ 0111_{BCD\text{-}4}$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre aritmética binaria.**	**CÓDIGO:** **BT1_AINP8**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	60 minutos

CALENDARIO:	Tercera Sesión. Previo a la Cuarta Sesión	REQUISITOS:	Códigos numéricos

CRITERIO DE ÉXITO:	Realización correcta de al menos 15 de las 25 conversiones

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS SOBRE ARITMÉTICA BINARIA

Realice la operación aritmética entre los números que se indica en cada caso. Tenga en cuenta que alguna de las operaciones puede no tener solución y que el resultado debe estar expresado en el mismo número de bits que los operandos.

Nº	Código	Operación	Operando A	Operando B	Resultado	Carry	OVF
1	COMP A 2	A – B	0111	1111			
2	COMP A 2	A + B	1100	1111			
3	COMP A 2	A – B	1111	0110			
4	COMP A 2	A + B	0100	1101			
5	COMP A 2	A – B	0110	0100			
6	BIN. NAT.	A + B	1110	0100			
7	BIN. NAT.	A + B	1101	0101			
8	BIN. NAT.	A – B	1110	0111			
9	BIN. NAT.	A + B	0011	1001			
10	COMP A 2	A / 4	1011000	------			
11	BIN. NAT.	A + B	0110	0101			
12	BIN. NAT.	A + B	1001	1101			
13	BIN. NAT.	A – B	1010	0101			
14	BIN. NAT.	A – B	0111	0100			
15	COMP A 2	A + B	0101	0010			
16	COMP A 2	A + B	0110	0111			
17	COMP A 2	A + B	1011	0110			
18	COMP A 2	A – B	1101	1101			
19	COMP A 2	A – B	1110	1100			
20	COMP A 2	A / 8	1011000	------			
21	BIN. NAT.	A + B	0101	1101			
22	BIN. NAT.	A - B	1011	0111			
23	COMP A 2	A + B	0110	1100			
24	COMP A 2	A + B	0110	0011			
25	COMP A 2	A + B	1001	1110			

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones a los ejercicios de aritmética binaria.

Nº	Código	Operación	Operando A	Operando B	Resultado	Carry	OVF
1	COMP A 2	A – B	0111	1111	1000	------	SI
2	COMP A 2	A + B	1100	1111	1011	------	NO
3	COMP A 2	A – B	1111	0110	1001	------	NO
4	COMP A 2	A + B	0100	1101	0001	------	NO
5	COMP A 2	A – B	0110	0100	0010	------	NO
6	BIN. NAT.	A + B	1110	0100	0010	SI	------
7	BIN. NAT.	A + B	1101	0101	0010	SI	------
8	BIN. NAT.	A – B	1110	0111	0111	------	------
9	BIN. NAT.	A + B	0011	1001	1100	NO	------
10	COMP A 2	A / 4	1011000	------	1110110	------	------
11	BIN. NAT.	A + B	0110	0101	1011	NO	------
12	BIN. NAT.	A + B	1001	1101	0110	SI	------
13	BIN. NAT.	A – B	1010	0101	0101	------	------
14	BIN. NAT.	A – B	0111	0100	0011	------	------
15	COMP A 2	A + B	0101	0010	0111	------	NO
16	COMP A 2	A + B	0110	0111	1101	------	SI
17	COMP A 2	A + B	1011	0110	0001	------	NO
18	COMP A 2	A – B	1101	1101	0000	------	NO
19	COMP A 2	A – B	1110	1100	0010	------	NO
20	COMP A 2	A / 8	1011000	------	1111 011	------	------
21	BIN. NAT.	A + B	0101	1101	0010	SI	-----
22	BIN. NAT.	A - B	1011	0111	0100	-----	-----
23	COMP A 2	A + B	0110	1100	0010	-----	NO
24	COMP A 2	A + B	0110	0011	1001	-----	SI
25	COMP A 2	A + B	1001	1110	0111	-----	SI

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Torneo EII de Codificación y Aritmética Binaria.**	**CÓDIGO:** **BT1_ATGP1**

GRUPO		FECHA:	

MODALIDAD:	Aprendizaje Cooperativo	TIPO:	Presencial	DURACIÓN:	110 minutos
CALENDARIO:	Cuarta Sesión.	REQUISITOS:			

CRITERIO DE ÉXITO:	Realización correcta de los ejercicios. Situación competitiva, evaluación referida a la norma.
INTERDEPENDENCIA POSITIVA:	En situación competitiva un miembro menos supone una desventaja difícilmente superable.
EXIGIBILIDAD INDIVIDUAL:	La puntuación del grupo está en relación directa con el trabajo de cada miembro individual.
ROLES:	No hay distribución diferenciada de roles entre los miembros
HABILIDADES SOCIALES:	Resistencia a la presión de una situación competitiva sin perder el apoyo mutuo. Distribución de trabajo y revisión que optimice el número de fallos y a velocidad.
REFLEXIÓN DE GRUPO:	Evaluación de los resultados de la estrategia elegida. ¿Se ha obtenido el mejor rendimiento del que el grupo es capaz?

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

TORNEO EII DE CODIFICACIÓN Y ARITMÉTICA BINARIA

PARTE I. Descripción de la Actividad.

a) La clase se distribuye para llevar a cabo una actividad de aprendizaje cooperativo en grupos G4.
b) Se dedican los diez primeros minutos para explicar la dinámica del torneo y que los equipos se pongan de acuerdo en la estrategia para la resolución de las tareas. También se podrá dedicar un tiempo a que los grupos resuelvan dudas relacionadas con las actividades individuales no presenciales BTI_AINP7 y BTI_AINP8, sobre codificación y aritmética binaria.
c) Se realizan cuatro rondas con una duración de 20 minutos cada una, de acuerdo a la siguiente reglamentación:
    a. Las dos primeras rondas abordan tareas de conversión entre diversos códigos, las otras dos versan sobre suma en binario natural, suma y resta en c.a.2, extensión del número de bits en c.a.2, cambio de signo, truncamiento y redondeo, y multiplicación y división por potencias de dos.
    b. Cada ronda consiste en un periodo de entrenamiento de 10 minutos en el que se tratarán de resolver aquellas dudas que tienen los miembros del grupo.
    c. A continuación se pasa a la resolución de un conjunto de 20 preguntas, con un tiempo máximo de 6 minutos para hacerlo.
    d. Cada grupo repartirá las tareas de la forma que le parezca oportuna.
    e. Cuando un grupo considere que ha contestado de manera correcta a las 20 preguntas puede detener el juego antes de haber transcurrido los 6 minutos.
    f. Durante los siguientes cuatro minutos cada grupo entrega su solución al siguientes grupo (el grupo 1 al grupo 2, el 2 al 3, y así sucesivamente) para que se lo corrija de acuerdo a la solución que entrega el profesor.
    g. Los siguientes cinco minutos se dedican a determinar la puntuación de la ronda por parte del profesor y por parte de los grupos a resolver las dudas que hayan quedado y a revisar las ventajas e inconvenientes de su estrategia y su desempeño como grupo.
    h. La puntuación se otorgará de la manera siguiente:
        i. Los grupos que no hayan detenido el juego reciben un punto por cada respuesta correcta y cero puntos por las incorrectas.
        ii. El grupo que haya detenido el juego recibe una bonificación a partir de 16 aciertos si es de 5 miembros, 14 si es de 4 y 12 si es de 3 miembros, pudiendo llegar a obtener un máximo de 30 puntos. Sin embargo, si obtiene menos de 16 aciertos, se le restará un punto por cada uno de los grupos que haya superado los 16 aciertos.
d) Finalizadas las cuatro rondas se suman los puntos obtenidos en cada una de ellas. Los grupos obtendrán una puntuación entre 0 y 9 puntos proporcional al número de respuestas correctas, es decir, entre 0 y 80 aciertos. El grupo campeón obtendrá un plus de un punto sobre la puntuación que haya obtenido.
e) El grupo con mejor puntuación de todo el curso pasará a formar parte del Salón de la Fama de Electrónica II.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PRIMERA RONDA

GRUPO DE CLASE: _____ **GRUPO: _____**

Tabla con las 20 preguntas de la primera ronda. Realizar las siguientes conversiones entre código BCD, hexadecimal, binario natural y complemento a dos.

Nº	Código Inicial	Código Final	Corrección
1	0101 1010 B	D	
2	1010 0101 B	D	
3	1011 1001 1110 B	D	
4	18 D	C2-MIN	
5	1011 1001 1110 BCD	D	
6	0001 0000 0000 BCD	D	
7	AF10 H	D	
8	0010 1111 C2-8	D	
9	1111 0111 0111 0001 C2-16	D	
10	666 D	B-MIN	
11	300 D	B-10	
12	173 D	B-8	
13	751 D	BCD-4	
14	1187 D	H-4	
15	301 D	H-2	
16	-58 D	C2-MIN	
17	-200 D	C2-8	
18	1001 0001 B	BCD-MIN	
19	7 557 D	B-16	
20	6 144 D	H-4	

NÚMERO DE ACIERTOS:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

SEGUNDA RONDA

GRUPO DE CLASE: _____ **GRUPO: _____**

Tabla con las 20 preguntas de la segunda ronda. Realizar las siguientes conversiones entre código BCD, hexadecimal, binario natural y complemento a dos.

Nº	Código Inicial		Código Final	Corrección
1	1111 0001	B	D	
2	1111 0001 0000	B	D	
3	0000 0001 0001	BCD	D-3	
4	1001 1001 0110	BCD	D	
5	0001 0000	H	D	
6	1FF	H	D	
7	1001 1111	C2-8	D	
8	1111 1000 0000 0001	C2-16	D	
9	1101 1101	C2	D	
10	1030	D	B-12	
11	77	D	B-8	
12	77	D	BCD-4	
13	1024	D	H-4	
14	231	D	H-MIN	
15	-58	D	C2-MIN	
16	-2	D	C2-8	
17	-31	D	C2-MIN	
18	257	D	C2-MIN	
19	16	D	C2-6	
20	-5	D	C2-5	

NÚMERO DE ACIERTOS:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

TERCERA RONDA

GRUPO DE CLASE:_____ **GRUPO: _____**

Tabla con las 20 preguntas de la tercera ronda. Realizar las siguientes operaciones aritméticas.

Nº	Operación	Operando A	Operando B	Resultado	Carry	OVF	Corr
1	A + B	11 1001 $_{B-6}$	00 0101 $_{B-6}$	$_{B-6}$			
2	A + B	11 0001 $_{B}$	01 0001 $_{B}$	$_{B}$			
3	A + B	10 0101 $_{B}$	00 0101 $_{B}$	$_{B}$			
4	A + B	1 1110 $_{C2}$	0 1100 $_{C2}$	$_{C2}$			
5	A + B	0 1110 $_{C2}$	1 0011 $_{C2}$	$_{C2}$			
6	A + B	1011 $_{C2}$	1011 $_{C2}$	$_{C2}$			
7	A + B	0110 $_{C2-4}$	0011 $_{C2-4}$	$_{C2-4}$			
8	A + B	0110 $_{C2-4}$	1100 $_{C2-4}$	$_{C2-4}$			
9	A + B	0011 $_{C2-4}$	1110 $_{C2-4}$	$_{C2-4}$			
10	A - B	1110 $_{C2-4}$	0111 $_{C2-4}$	$_{C2-4}$			
11	A - B	01 1001 $_{B-6}$	00 0101 $_{B-6}$	$_{B-6}$			
12	A - B	1 1101 $_{C2}$	0 1010 $_{C2}$	$_{C2}$			
13	A - B	0110 $_{C2-4}$	1000 $_{C2-4}$	$_{C2-4}$			
14	A - B	0 0110 $_{C2}$	1 0001 $_{C2}$	$_{C2}$			
15	A / 8	10 1101 1100 $_{C2}$	----------	$_{C2}$			
16	A · 4	00 0110 0101 $_{C2}$	----------	$_{C2}$			
17	-A	1011 0101 $_{C2}$	----------	$_{C2}$			
18	Ext. de bits	01 0011 $_{C2-6}$	----------	$_{C2-8}$			
19	Truncamiento	0011 1010 $_{B-8}$	----------	$_{B-6}$			
20	Redondeo	100 1111 $_{B-7}$	----------	$_{B-5}$			

NÚMERO DE ACIERTOS:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

CUARTA RONDA

GRUPO DE CLASE: _____ **GRUPO: _____**

Tabla con las 20 preguntas de la cuarta ronda. Realizar las siguientes operaciones aritméticas.

Nº	Operación	Operando A	Operando B	Resultado	Carry	OVF	Corr
1	A + B	10 0101 $_{B}$	00 0101 $_{B}$	$_{B}$			
2	A + B	01 0011 $_{B}$	00 1100 $_{B}$	$_{B}$			
3	A + B	11 0011 $_{B}$	01 1010 $_{B}$	$_{B}$			
4	A + B	0111 1110 $_{C2}$	0000 0011 $_{C2}$	$_{C2}$			
5	A + B	0000 1011 $_{C2}$	0000 1011 $_{C2}$	$_{C2}$			
6	A + B	11 0011 $_{C2}$	01 1010 $_{C2}$	$_{C2}$			
7	A + B	1100 0111 $_{C2}$	1100 1001 $_{C2}$	$_{C2}$			
8	A + B	11 0001 $_{B}$	01 0001 $_{B}$	$_{B}$			
9	A + B	0000 0111 $_{C2}$	1111 1001 $_{C2}$	$_{C2}$			
10	A – B	1 1111 $_{C2}$	0 1010 $_{C2}$	$_{C2}$			
11	A - B	0111 $_{C2-4}$	1111 $_{C2-4}$	$_{C2-4}$			
12	A – B	0 1111$_{C2}$	1 1001 $_{C2}$	$_{C2}$			
13	A – B	0101 1011 $_{C2}$	0110 0011 $_{C2}$	$_{C2}$			
14	A – B	10 1001 $_{C2}$	01 0001 $_{C2}$	$_{C2}$			
15	A / 16	1101 1100 $_{C2}$	----------	$_{C2}$			
16	A · 16	1111 1101 $_{C2}$	----------	$_{C2}$			
17	-A	1100 0110 $_{C2}$	----------	$_{C2}$			
18	Ext. de bits	100 0111 $_{C2-7}$	----------	$_{C2-8}$			
19	Truncamiento	0101 0101 $_{B-8}$	----------	$_{B-5}$			
20	Redondeo	1011 0001 $_{B-8}$	----------	$_{B-6}$			

NÚMERO DE ACIERTOS:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Lectura sobre sistemas combinacionales y Álgebra de Boole.**	**CÓDIGO:** **BT1_AINP9**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Lectura. Individual	TIPO:	No presencial	DURACIÓN:	120 minutos

CALENDARIO:	Cuarta Sesión. Previo a la Quinta Sesión	REQUISITOS:	Ninguno

CRITERIO DE ÉXITO:	No aplicable

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	No aplicable

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

La siguiente lectura expone los principios fundamentales del Algebra de Boole, así como algunas técnicas básicas para la representación de la funcionalidad de un sistema digital combinacional. Realice una lectura con detenimiento del texto, tratando de comprender tanto los conceptos teóricos como los ejemplos prácticos que se presentan.

Anote en una hoja aquellas dudas que no haya sido capaz de resolver, así como la numeración de aquellos ejemplos cuya solución no comprenda plenamente. En la primera sesión presencial de la tercera semana del bloque temático, el profesor llevará a cabo una presentación resumida de los aspectos esenciales de esta lectura y resolverá las dudas que se planteen en la sesión.

Asimismo, con el fin de asentar un poco más lo tratado, se plantearán para su resolución algunos ejercicios en los que se pondrán en práctica algunas de las técnicas abordadas en la lectura.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas combinacionales

En términos generales se puede decir que la función de los sistemas digitales consiste en procesar información. La información a procesar se presenta en la entrada del sistema y el producto resultante del procesamiento es entregado por el sistema en su salida. La función de procesamiento específica de cada sistema determina la relación que existe entre sus entradas y salidas. La figura 1 ilustra está definición genérica de los sistemas digitales.

Figura 1

Un ejemplo, conceptualmente sencillo, de sistema digital se muestra en la figura 2. Se trata de un sumador. La entrada del sumador son dos números, A y B, expresados en un determinado código. La función de procesamiento consiste en obtener la suma de los dos números. La salida es otro número, S, cuyo valor se corresponde con la suma de A y B.

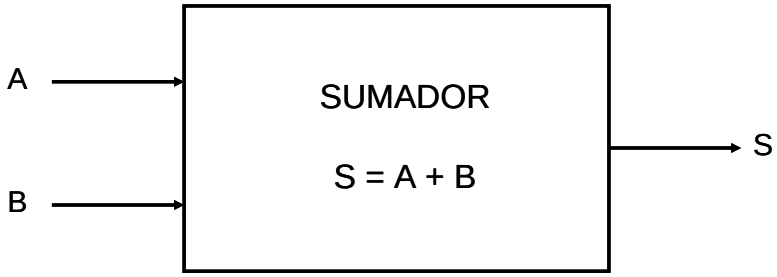


Figura 2

Atendiendo a las características de su función de procesamiento, se distinguen dos clases de sistemas digitales: los sistemas *combinacionales* y los sistemas *secuenciales*. La característica que los diferencia es la memoria. Los sistemas combinacionales *no tienen memoria*, mientras que los secuenciales sí.

La ausencia de memoria se manifiesta en el hecho de que la combinación de unos y ceros que aparece en la salida de un sistema combinacional está determinada, *exclusivamente*, por la combinación presente en la entrada en cada momento –no depende, por tanto, de valores de entrada previos.

Lo expresado anteriormente implica que, ante una determinada combinación de unos y ceros en la entrada, un sistema combinacional responde, siempre, con la misma combinación en su salida. Existe, por tanto, una *relación invariable en el tiempo* entre cada combinación de entrada y

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

el valor correspondiente que el circuito entrega en la salida para esa combinación. En consecuencia, la combinación que aparece en la salida de un circuito combinacional no cambia mientras la combinación que haya en su entrada no se modifique.

Vamos a ver estas propiedades materializadas en el ejemplo de la figura 2. En la figura 3, se muestra un circuito sumador en el que las entradas, A y B, toman los valores 2 y 2, respectivamente. Puesto que la función del circuito es calcular la suma de A y B, el valor de salida debe ser 4.

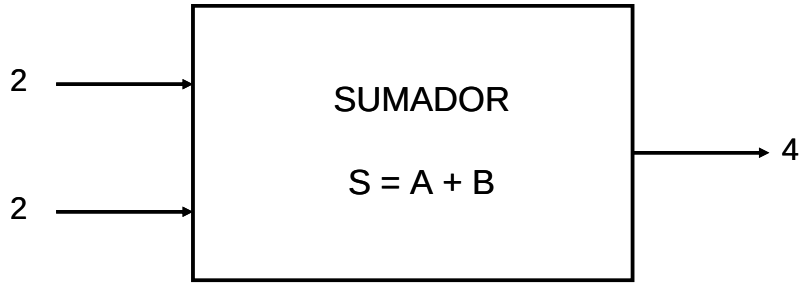


Figura 3

Para evaluar el resultado de la suma basta con *leer* el valor presente en las entradas, resultando irrelevante el que tomen en cualquier otro momento (el circuito no necesita tener memoria para realizar la suma). Es evidente, también, que a un determinado valor de entrada, 2 y 2, le corresponde siempre el mismo valor de salida, 4 –esto no se verifica al revés, obviamente la salida también valdrá 4, por ejemplo, cuando las entradas A y B valgan 1 y 3. Observe, por último, que mientras las entradas A y B valgan 2, la salida mantendrá el valor 4 y sólo cuando cambien las combinaciones de entrada de modo que su suma sea distinta de 4, variará la salida,

Además de conocer las propiedades que caracterizan a un circuito combinacional, es conveniente ser capaz de identificar cuándo una determinada función *no puede realizarse* con este tipo de circuitos. Si para realizar una determinada operación un circuito necesita disponer de *memoria*, no será posible realizarla con un circuito combinacional. A continuación se verá un ejemplo que ilustra esta situación. El sistema de la figura 4 indica en su salida el valor correspondiente a la suma acumulada de los números que se presentan en su entrada.

Figura 4

En el ejemplo, la serie de números que se entrega al circuito es 1, 3, 2, 3; si el valor inicial acumulado es 0, la serie de números que aparecerá en la salida será: 1 (valor acumulado + 1, cuando la entrada vale 1), 4 (valor acumulado + 3, cuando la entrada vale 3), 6 (valor acumulado + 2, cuando la entrada vale 2) y 9 (valor acumulado + 3, la segunda vez que la entrada vale 3). El

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

circuito necesita recordar el valor acumulado para calcular el valor de salida, por tanto ha de disponer de *memoria*. Observando la entrada y salida del circuito, la memoria asociada a la función se pone de manifiesto en el hecho de que para un mismo valor de entrada la salida puede tomar distintos valores (en el ejemplo, cuando la entrada vale 3, la salida vale en una ocasión 4 y en otra 9), lo que indica que la salida depende de *algo más* que del valor actual de la entrada.

1. Sistemas combinacionales y lógica

La lógica es la ciencia que estudia las reglas del pensamiento racional, de los razonamientos. Aristóteles la definió como *el arte de la argumentación correcta y verdadera*. La parte de su obra dedicada al estudio del silogismo –un razonamiento formado por dos premisas (proposiciones) y una conclusión que se deduce de éstas- es la precursora de esta rama del conocimiento. A lo largo de los siglos XIX y XX el trabajo de un destacado grupo de matemáticos y pensadores condujo a la creación de la lógica formal o simbólica, que consiste, en esencia, en una formulación matemática de las proposiciones y las reglas lógicas. Uno de los resultados más conocidos de estos trabajos, por ser la base de la computación y el diseño de los sistemas digitales, es el **álgebra de Boole**. Debe su nombre al matemático inglés George Boole (1815-1864), el primero en establecer una relación entre el lenguaje algebraico y las formas lógicas y silogismos. La aplicación del álgebra de Boole al diseño digital se debe a Claude Elwood Shannon (1916-2001), que demostró que el funcionamiento de los circuitos de conmutación y los circuitos digitales se puede representar mediante funciones de un tipo de álgebra de Boole llamada álgebra de conmutación.

El álgebra de conmutación que se utiliza en la ingeniería de sistemas digitales define tres operaciones sobre un conjunto de dos elementos, el 0 y el 1. Los sistemas combinacionales realizan funciones que se construyen mediante la combinación de estas operaciones (operaciones lógicas o booleanas). La simplicidad de las operaciones lógicas y de su realización electrónica, por un lado, y la complejidad del procesamiento que puede realizarse combinando un gran número de ellas, por otro, son una de las claves del éxito de la electrónica digital. El otro factor decisivo de este éxito es el vertiginoso aumento de la escala de integración electrónica, que permite disponer de chips muy baratos que contienen millones de operadores lógicos.

2. Álgebra de Boole. Definición. Propiedades, operaciones y teoremas. Notación

<u>*Definición y postulados*</u>

Se dice que un conjunto, A, con al menos dos elementos, tiene una estructura de álgebra de Boole si, dadas dos operaciones binarias[1], denominadas suma (simbolizada por +) y producto[2] (simbolizada por •), y una operación monaria[3] denominada complemento (simbolizada por ¯), dichas operaciones poseen las siguientes propiedades (denominadas axiomas de la estructura):

A1. *La suma y el producto son operaciones conmutativas:*

[1] El calificativo binario hace referencia en este contexto a operaciones que se aplican a dos elementos.
[2] Las operaciones del álgebra de Boole se denominan y representan de diversas formas. Aquí se ha optado por una de las más utilizadas en matemáticas. Como se verá más adelante, es común en la ingeniería de circuitos digitales denominar AND (Y) al producto, OR (O) a la suma y NOT (NO) al complemento.
[3] Una operación monaria es la que se aplica a un único elemento.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

$$\forall a,b \in A, \quad a+b=b+a \quad y \quad a \cdot b = b \cdot a$$

A2. *La suma y el producto son operaciones asociativas[4]:*

$$\forall a,b,c \in A, \quad (a+b)+c=a+(b+c) \quad y \quad (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

A3. *El elemento neutro[5] de la suma es el 0 y el elemento neutro del producto es el 1:*

$$\forall a \in A, \quad a+0=a \quad y \quad a \cdot 1 = a$$

A4. *Cada operación binaria es distributiva respecto de la otra:*

$$\forall a,b,c \in A, \quad a \cdot (b+c) = a \cdot b + a \cdot c \quad y \quad a+(b \cdot c) = (a+b) \cdot (a+c)$$

A5. *Todos los elementos del conjunto tienen un complemento, que es único[6], cumpliéndose que la suma de un elemento más su complemento es igual a 1 y el producto de un elemento por su complemento es 0:*

$$\forall a \in A \; \exists \bar{a} \in A, \quad a+\bar{a}=1 \quad y \quad a \cdot \bar{a} = 0$$

El álgebra de Boole que se utiliza en la ingeniería de los sistemas digitales –y, por tanto, la que específicamente nos interesa- es la definida sobre el conjunto {0, 1}. Se la conoce como *álgebra de conmutación*. En este álgebra las operaciones suma, producto y complemento también se denominan, respectivamente, *OR*, *AND* y *NOT*; en este capítulo, por simplicidad, no utilizaremos esta notación. A partir de ahora, particularizaremos el estudio al álgebra de conmutación, aunque nos referiremos a ella indistintamente como álgebra de conmutación o álgebra de Boole.

Operaciones del álgebra de conmutación

Una vez presentadas las propiedades que definen el álgebra, vamos a definir[7] a continuación, particularizando ya para el álgebra de conmutación, sus tres operaciones básicas. Para dominar el cálculo y el análisis de expresiones lógicas (booleanas) resulta imprescindible manejarlas con soltura, una habilidad que no resulta difícil adquirir porque, afortunadamente, en el álgebra de conmutación –al igual que en otras álgebras de Boole definidas sobre conjuntos de dos elementos- estas operaciones son muy sencillas.

1.- Complemento: *El complemento del 0 es el 1 y el complemento del 1 es el 0.*

$$\bar{0}=1 \qquad \bar{1}=0$$

2.- Suma: *La suma de dos elementos vale 0 sólo si ambos son 0, en cualquier otro caso vale 1.*

[4] La propiedad asociativa no es propiamente un axioma, ya que puede derivarse de los demás en una definición rigurosamente matemática del álgebra.
[5] En matemáticas también se lo denomina elemento *identidad*.
[6] La unicidad del elemento neutro incluida en este axioma es en realidad un teorema del álgebra.
[7] En el álgebra de conmutación las operaciones se derivan de los axiomas del álgebra.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

+	**0**	**1**
0	*0*	*1*
1	*1*	*1*

La suma de un conjunto de elementos vale 0 únicamente si todos los valores que se suman valen 0; basta con que uno de los valores que se suma tome el valor 1 para que la suma valga 1.

$$0+0+0+\ldots+0+\ldots+0+0+0=0$$
$$0+0+0+\ldots+1+\ldots+0+0+0=1$$

3.- Producto: *El producto de dos elementos vale 1 sólo si ambos son 1, en cualquier otro caso vale 0.*

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

•	**0**	**1**
0	*0*	*0*
1	*0*	*1*

El producto de un conjunto de elementos vale 1 únicamente si todos ellos valen 1; basta con que uno de los valores del producto tome el valor 0 para que el resultado valga 0.

$$1\cdot1\cdot1\cdot\ldots\cdot1\cdot\ldots\cdot1\cdot1\cdot1=1$$
$$1\cdot1\cdot1\cdot\ldots\cdot0\cdot\ldots\cdot1\cdot1\cdot1=0$$

Ejemplo 3.1*: La suma de tres variables* $a+\overline{b}+c$ *vale 0 sólo si a es 0, b es 1 (y su complementario[8], por tanto, 0) y c es 0; cualquier otra combinación de valores de las variables da lugar a que la suma valga 1.*

Ejemplo 3.2*: Si consideramos el producto de tres variables* $\overline{a}\cdot\overline{b}\cdot c$*, su valor será 1 en el caso de que a y b valgan 0 (sus complementarios, 1) y c sea igual a 1; para cualquier otra combinación de valores el producto valdrá 0.*

Ejemplo 3.3*: La expresión* $(a+b)\cdot\overline{c}$ *toma el valor 1 para las combinaciones de valores en que c vale 0 y, a o b (al menos una de ellas, o las dos) valen 1.*

Por comodidad suele omitirse el símbolo del producto entre variables o entre ellas y los paréntesis –esto es lo que suele hacerse también en el álgebra corriente.

3. Cálculo booleano. Evaluación de expresiones.

Notación y orden de precedencia de las operaciones.

Cualquier expresión o función booleana está compuesta por literales que representan variables y constantes (0 y 1), los símbolos de las operaciones del álgebra, paréntesis y símbolos de igualdad. El orden de precedencia en la evaluación de las operaciones, de mayor a menor, es: complemento, producto y suma.

[8] Las variables complementadas se denominan también variables *negadas*. Por ejemplo, se puede aludir al término $\overline{a}$ diciendo "a complementada" o "a negada". En este texto se utilizará indistintamente cualquiera de estas dos formas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

***Ejemplo 4.1**: Para evaluar la expresión* $a\cdot\overline{b}+b\cdot\overline{c}\cdot d+b$ *se calcularían primero los productos* $a\cdot\overline{b}$ *y* $b\cdot\overline{c}\cdot d$ *, después se calcularía la suma de ambos productos y b. Si, por ejemplo, a y b valen 0 y c y d valen 1:*

$$a\cdot\overline{b}+b\cdot\overline{c}\cdot d+b=0\cdot\overline{0}+0\cdot\overline{1}\cdot 1+0=0\cdot 1+0\cdot 0\cdot 1+0=0+0+0=0$$

La evaluación de sumas y productos puede alterarse mediante el uso de paréntesis o cuando se complementan expresiones –el complemento, en este caso, realiza el mismo papel que los paréntesis. Para realizar el cálculo de una expresión complementada, primero se calcula el valor de la expresión y luego se complementa.

***Ejemplo 4.2**: En la expresión* $a\cdot(\overline{b+c+d})$ *pueden eliminarse los paréntesis porque el complemento de la suma de variables establece una precedencia en el cálculo del complemento de la expresión sobre el producto por la variable a. Esta expresión se escribiría normalmente*[9] $a\cdot\overline{b+c+d}$ *. Si todas las variables valen 1:*

$$a\cdot\overline{b+c+d}=1\cdot\overline{1+1+1}=1\cdot\overline{1}=1\cdot 0=0$$

Si todas las variables valen 0:

$$a\cdot\overline{b+c+d}=0\cdot\overline{0+0+0}=0\cdot\overline{0}=0\cdot 1=0$$

Si a vale 1 y el resto de las variables 0:

$$a\cdot\overline{b+c+d}=1\cdot\overline{0+0+0}=1\cdot\overline{0}=1\cdot 1=1$$

***Ejemplo 4.3**: La función* $F=(\overline{a+b}+\overline{c}d)\overline{\overline{b}c}$ *se evaluaría en el siguiente orden:*

$$\left.\begin{array}{l}\overline{a+b}\\ \overline{c}d\end{array}\right\}\longrightarrow\left.\begin{array}{l}\overline{a+b}+\overline{c}d\\ \overline{\overline{b}c}\end{array}\right\}\longrightarrow(\overline{a+b}+\overline{c}d)\overline{\overline{b}c}$$

En primer lugar, $\overline{a+b}$ *ó* $\overline{c}d$ *, después su suma para obtener* $(\overline{a+b}+\overline{c}d)$*. A continuación,* $\overline{\overline{b}c}$ *-que también podría haberse calculado antes que el paréntesis- y, por último, se calcularía el producto de estos dos términos para evaluar la expresión completa.*

Si, por ejemplo, a y c valen 0 y b y d valen 1, el valor de la función sería:

$$1^{o}\left\{\begin{array}{l}\overline{a+b}=\overline{0+1}=\overline{1}=0\\ \overline{c}d=\overline{0}\cdot 1=1\cdot 1=1\end{array}\right.$$

$$2^{o}\left\{\begin{array}{l}\overline{a+b}+\overline{c}d=0+1=1\\ \overline{\overline{b}c}=\overline{\overline{1}\cdot 0}=\overline{0\cdot 0}=\overline{0}=1\end{array}\right.$$

$$3^{o}\left\{(\overline{a+b}+\overline{c}d)\overline{\overline{b}c}=1\cdot 1=1\right.$$

[9] En ocasiones se conservan los paréntesis para mejorar la legibilidad, en particular cuando se trata, como en el ejemplo, de un producto. En el caso de una suma, p. ej: $a+\overline{b+c+d}$ el problema de legibilidad desaparece y no son necesarios.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Observe cómo para evaluar una expresión negada primero se realiza el cálculo de la expresión y luego se complementa.

Cuando a y c valen 1 y b y d valen 0:

$$F = (\overline{a+b} + \bar{c}d)\overline{\bar{b}c} = (\overline{1+0} + \bar{1}\cdot 0)\cdot\overline{\bar{0}\cdot 1} = (\bar{1} + 0\cdot 0)\cdot\overline{1\cdot 1} = (0+0)\cdot\bar{1} = 0\cdot 0 = 0$$

Cuando todas las variables valen 0:

$$F = (\overline{a+b} + \bar{c}d)\overline{\bar{b}c} = (\overline{0+0} + \bar{0}\cdot 0)\cdot\overline{\bar{0}\cdot 0} = (\bar{0} + 1\cdot 0)\cdot\overline{1\cdot 0} = (1+0)\cdot\bar{0} = 1\cdot 1 = 1$$

Cuando todas las variables valen 1:

$$F = (\overline{a+b} + \bar{c}d)\overline{\bar{b}c} = (\overline{1+1} + \bar{1}\cdot 1)\cdot\overline{\bar{1}\cdot 1} = (\bar{1} + 0\cdot 1)\cdot\overline{0\cdot 1} = (0+0)\cdot\bar{0} = 0\cdot 1 = 0$$

Las complementaciones anidadas se resuelven con reglas análogas a las de los paréntesis, evaluando primero las más internas.

Ejemplo 4.4: *El cálculo del valor de la expresión* $\overline{c + \overline{d\cdot(\bar{b}+a)}\cdot(c+d)}$*, cuando a y b valen 1 y c y d valen 0, se realizaría de la siguiente manera:*

En primer lugar se evalúa $\overline{d\cdot(\bar{b}+a)}$*:*

$$\overline{d\cdot(\bar{b}+a)} = \overline{0\cdot(\bar{1}+1)} = \overline{0\cdot(0+1)} = \overline{0\cdot 1} = \bar{0} = 1$$

Después, la suma $(c+d)$*:*

$$(c+d) = (0+0) = 0$$

A continuación, el producto de $\overline{d\cdot(\bar{b}+a)}$ *y* $(c+d)$*:*

$$\overline{d\cdot(\bar{b}+a)}\cdot(c+d) = 1\cdot 0 = 0$$

Se suman $\overline{d\cdot(\bar{b}+a)}\cdot(c+d)$ *y c:*

$$c + \overline{d\cdot(\bar{b}+a)}\cdot(c+d) = 0+0$$

Y, por último, se resuelve el complemento externo:

$$\overline{c + \overline{d\cdot(\bar{b}+a)}\cdot(c+d)} = \bar{0} = 1$$

Los nombres de las funciones y variables se representan habitualmente mediante una letra. En la aplicación del álgebra al diseño y análisis de circuitos muchas veces se utilizan subíndices numéricos y acrónimos o nombres en lugar de letras[10].

[10] Esta notación, que será la más habitual durante el transcurso de la asignatura, no se utilizará en este texto.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejemplo 4.5: *En la función booleana* $DATA = ENABLE \cdot (S_0 + S_1)$ *los nombres de las variables son ENABLE, S_0 y S_1, el de la función es DATA.*

Ejemplo 4.6: *El producto de dos variables* $b \cdot c$ *se reduce a* bc*. Una expresión como* $a \cdot (b + c \cdot d)$*, se escribe* $a(b + cd)$*.*

Ejemplo 4.7: *El valor de* $D_0 = \overline{(A_0 + A_1 + \overline{A_2})(\overline{A_0}\,\overline{A_1}A_2 + A_1) + \overline{\overline{\overline{A_0}A_1 + A_0\overline{A_2}}}}$ *cuando A_0 y A_1 valen 0 y A_2 vale 1, se calcularía de la siguiente manera:*

$$D_0 = \overline{(0+0+\overline{1})(\overline{0}\cdot\overline{0}\cdot 1+0)+\overline{\overline{\overline{0}\cdot 0+0\cdot\overline{1}}}} = \overline{(0+0+0)(1\cdot 1\cdot 1+0)+\overline{1\cdot 0+0\cdot 0}}$$
$$D_0 = \overline{0\cdot(1+0)+\overline{0+0}} = \overline{0\cdot 1+\overline{0}} = \overline{0+1} = \overline{1} = 0$$

Si A_0 vale 1 y A_1 y A_2 valen 0:

$$D_0 = \overline{(1+0+\overline{0})(\overline{1}\cdot\overline{0}\cdot 0+0)+\overline{\overline{\overline{1}\cdot 0+1\cdot\overline{0}}}} = \overline{(1+0+1)(0\cdot 1\cdot 0+0)+\overline{0\cdot 0+1\cdot 1}}$$
$$D_0 = \overline{1\cdot(0+0)+\overline{0+1}} = \overline{1\cdot 0+\overline{1}} = \overline{0+0} = \overline{0} = 1$$

<u>*Evaluación eficiente de expresiones.*</u>

Ejemplo 4.8: *En una suma de términos basta con que uno de ellos valga 1 para que toda la expresión tome este valor; por tanto, una buena estrategia para calcular el valor de una expresión de este tipo consiste en evaluar primero los términos más sencillos. De este modo la operación finaliza cuando se encuentra un término que valga 1 ó, en el peor caso, cuando se comprueba que todos valen 0.*

El valor de la función, $F = \overline{a\overline{b}c + b\overline{d}\cdot\overline{cd+ba}} + ac$*, cuando todas las variables valen 1, resulta evidente que vale 1, pues el producto ac vale 1:*

$$F = \overline{a\overline{b}c + b\overline{d}\cdot\overline{cd+ba}} + ac = \overline{a\overline{b}c + b\overline{d}\cdot\overline{cd+ba}} + 1 = 1$$

Ejemplo 4.9: *Cuando el formato de la expresión se reduce a un producto de términos, basta con que uno de ellos valga 0 para que la expresión tome también este valor. Por ejemplo, si tratamos de calcular el valor de:*

$$\overline{a + \overline{b}(c + \overline{a}d)}\cdot(b + c\overline{d})\cdot(c + d)$$

cuando todas las variables valen 0, concluimos inmediatamente que la expresión vale 0, pues c+d es 0.

Ejemplo 4.10: *Un formato muy utilizado para describir las ecuaciones booleanas consiste en expresarlas como una suma de productos o un producto de sumas de todas las variables de la función. La evaluación de este tipo de expresiones para una determinada combinación de valores es inmediata.*

Tomemos, por ejemplo, la función $F = a\overline{b}c + \overline{a}\,\overline{b}\,\overline{c} + ab\overline{c}$ ***.*** *Para que F valga 1 basta con que uno de los productos que se suman tome este valor. Comprobar esta condición resulta muy sencillo, porque para una determinada combinación de valores de las variables sólo existe un producto de todas ellas cuyo valor sea 1. Si, por ejemplo, a y b valen 0 y c vale 1, el único producto de a, b y c que vale 1 es* $\overline{a}\,\overline{b}c$*. La función del*

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

ejemplo vale 0 para estos valores de las variables porque $\bar{a}\bar{b}c$ *no es uno de los sumandos. Observando los productos de la función, se deduce que sólo hay tres combinaciones de valores para las que F vale 1:*

$$a=1 \quad b=0 \quad c=1 \rightarrow \quad a\bar{b}c=1$$
$$a=0 \quad b=0 \quad c=0 \rightarrow \quad \bar{a}\bar{b}\bar{c}=1$$
$$a=1 \quad b=1 \quad c=0 \rightarrow \quad ab\bar{c}=1$$

Para cualquier otra combinación la función toma el valor 0. En todos los productos que se suman en F aparecen las tres variables. Si alguna variable no forma parte de un producto el valor al que se evalúa éste no depende, evidentemente, de ella. Por tanto basta con que el valor de las variables que aparecen valga 1, si no están complementadas, ó 0, si lo están, para que el producto valga 1.

Ejemplo 4.11: *La ecuación* $F=(a+\bar{b}+\bar{c})(\bar{a}+\bar{b}+\bar{c})(a+b+c)$ *es el producto de un conjunto de sumas. Valdrá, por tanto, 0 cuando alguna de las sumas sea 0. Al igual que en los productos, una suma de todas las variables de una función sólo puede valer 0 para una determinada combinación de valores:*

$$a=0 \quad b=1 \quad c=1 \rightarrow \quad (a+\bar{b}+\bar{c})=0$$
$$a=1 \quad b=1 \quad c=1 \rightarrow \quad (\bar{a}+\bar{b}+\bar{c})=0$$
$$a=0 \quad b=0 \quad c=0 \rightarrow \quad (a+b+c)=0$$

Por tanto, F valdrá 0 para esas tres combinaciones de a, b y c y 1 para cualquier otra.

Ejemplo 4.12: *Las propiedades de la suma y el producto pueden combinarse para deducir fácilmente las combinaciones de valores de variables que hacen que una función se evalúe a 1 ó a 0. Por ejemplo, la función* $F=\overline{a(\bar{b}+c)}+\bar{b}c$ *valdrá 1 cuando* $\overline{a(\bar{b}+c)}$ *o* $\bar{b}c$ *valgan 1, en cualquier otro caso valdrá 0. El primer término,* $\overline{a(\bar{b}+c)}=\bar{a}+b\bar{c}$ *, vale 1 cuando a es 0, ó cuando, simultáneamente, b valga 1 y c, 0:*

$$a=0 \rightarrow F=1 \qquad\qquad b\bar{c}=1 \rightarrow F=1$$

$\underline{a}$	$\underline{b}$	$\underline{c}$
0	0	0
0	0	1
0	1	0
0	1	1

$\underline{a}$	$\underline{b}$	$\underline{c}$
0	1	0
1	1	0

El segundo término vale 1 cuando b es 0 y c, 1:

$$\bar{b}c=1 \rightarrow F=1$$

$\underline{a}$	$\underline{b}$	$\underline{c}$
0	0	1
1	0	1

El conjunto de combinaciones que hacen que la función valga 1 es, por tanto:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

a	*b*	*c*
0	0	0
0	0	1
0	1	0
0	1	1
1	0	1
1	1	0

Para cualquier otra combinación la función vale 0.

4. Tablas de verdad

Puesto que en el álgebra de conmutación una variable solo puede valer 0 ó 1, con un conjunto de n variables pueden formarse únicamente 2^n combinaciones distintas de valores. Este hecho permite que las funciones del álgebra de conmutación puedan describirse mediante tablas que indican el valor que toma la función para cada una de estas combinaciones. Estas tablas reciben el nombre de *tablas de verdad*.

En las tablas de verdad existe una columna para cada variable y otra para la función. Las columnas de las variables se sitúan a la izquierda de la tabla y la de la función a la derecha. El número de filas de la tabla depende exclusivamente del número de variables que tenga la función, porque cada fila se corresponde con una combinación de valores de éstas. Obviamente, el número de filas de la tabla de verdad de una función de n variables es 2^n. Esta relación limita el uso de las tablas a funciones con pocas variables; desde un punto de vista práctico, a funciones de menos de 6 variables (una tabla de 2^6 = 64 filas, resulta *demasiado grande*)[11].

En la figura 5 se muestra una función de tres variables descrita mediante su tabla de verdad. Observe el orden en que están dispuestas las combinaciones en la tabla. Se corresponde con una lista ordenada de acuerdo con el valor binario de la combinación: la primera fila es la combinación 000 (0 en decimal), la segunda es 001 (1 en decimal), la tercera, 010 (2 en decimal) y así sucesivamente hasta la combinación 111 (7 en decimal). El uso de esta ordenación resulta muy conveniente desde el punto de vista práctico.

a	**b**	**c**	**F**
0	**0**	**0**	**0**
0	**0**	**1**	**1**
0	**1**	**0**	**1**
0	**1**	**1**	**0**
1	**0**	**0**	**0**
1	**0**	**1**	**1**
1	**1**	**0**	**0**
1	**1**	**1**	**0**

Figura 5

[11] Esto es en general; más adelante se verá que pueden utilizarse en determinadas circunstancias para definir funciones con un mayor número de variables.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

El cálculo del valor de la función en este formato es inmediato; basta con observar el valor de F en la fila correspondiente a la combinación de valores que se está evaluando. Por ejemplo, para la combinación a = 0, b = 0 y c = 1, la función vale 1 porque en la fila correspondiente a esta combinación (la segunda), el valor de F es 1.

Obtención de tablas de verdad a partir de expresiones algebraicas

Dada la expresión algebraica de una función booleana, su representación en una tabla de verdad se realiza calculando su valor para todas las posibles combinaciones de las variables. El valor calculado se indica en la fila correspondiente a la combinación. Veamos algunos ejemplos.

Ejemplo 5.1: *Se trata de realizar la tabla de verdad de la función* $F = a + bc$. *La función tiene tres variables y, por tanto, el número de combinaciones que pueden formarse con ellas es 8 (2^3). El cálculo del valor de la función para esas ocho combinaciones puede afrontarse evaluando la función para cada una de ellas:*

$$
\begin{array}{llllll}
a=0 & b=0 & c=0 & F= & 0+0\cdot 0 & =0 \\
a=0 & b=0 & c=1 & F= & 0+0\cdot 1 & =0 \\
a=0 & b=1 & c=0 & F= & 0+1\cdot 0 & =0 \\
a=0 & b=1 & c=1 & F= & 0+1\cdot 1 & =1 \\
a=1 & b=0 & c=0 & F= & 1+0\cdot 0 & =1 \\
a=1 & b=0 & c=1 & F= & 1+0\cdot 1 & =1 \\
a=1 & b=1 & c=0 & F= & 1+1\cdot 0 & =1 \\
a=1 & b=1 & c=1 & F= & 1+1\cdot 1 & =1
\end{array}
$$

Normalmente resulta más eficiente analizar los términos de la función con el fin de determinar las combinaciones para las que vale 1 ó 0 –lo que sea más fácil: una vez que se deducen, por ejemplo, las combinaciones para las que vale 1, para el resto la función vale 0.

En la ecuación del ejemplo resulta muy sencillo determinar las combinaciones para las que la función vale 1: la ecuación se compone de dos términos que se suman, a y bc, por lo tanto valdrá 1 en todas las combinaciones para las que al menos uno de ellos valga 1. El término a vale 1 en todas las combinaciones en las que la variable a toma este valor (cuatro en este caso); el término bc, en todas en las que b y c valgan 1 (dos combinaciones). En la figura 6 se muestran todas las combinaciones para las que la función vale 1 –son cinco en total porque hay una, a = 1, b =1 y c = 1, que cumple las dos condiciones- y la tabla de verdad completa.

a	b	c	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Figura 6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejemplo 5.2: *La tabla de verdad de la función* $F=(b+\overline{c})(c+\overline{a+b})$ *se obtiene inmediatamente advirtiendo que, puesto que la función consiste en el producto de dos términos, vale 0 en las combinaciones en las que* $(b+\overline{c})$ *o* $(c+\overline{a+b})$ *valen 0:*

$(b+\overline{c})=0$ *si b = 0 y c = 1*

$(c+\overline{a+b})=0$ *si c = 0 y, a = 1 ó b = 1 (*$\overline{a+b}$ *vale 0 cuando a ó b valen 1)*

La tabla de verdad de la función es:

a	b	c	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Formas canónicas. Minitérminos y maxitérminos

En el álgebra de Boole una función puede expresarse algebraicamente de múltiples formas. Una propiedad muy importante de las funciones lógicas es que se pueden describir mediante lo que se conoce como las *formas canónicas* (o *formas normalizadas*) de la función. El interés de las formas canónicas se deriva de que pueden extraerse directamente de la tabla de verdad de la función.

Las formas canónicas son dos:

1. La primera forma canónica, denominada *forma canónica disyuntiva*, expresa la función como una suma de *minitérminos*. Un minitérmino es un producto del que forman parte todas las variables de la función; cada variable aparece una sola vez y puede estar complementada o sin complementar. Esto implica que una función de N variables puede tener hasta 2^N minitérminos. La característica esencial de un minitérmino es que sólo vale 1 para una determinada combinación de valores de las variables que lo forman y que, para esa combinación, cualquier otro minitérmino vale 0.

   Ejemplo 5.3: *La función de tres variables* $F=\overline{a}\overline{b}c+\overline{a}bc+a\overline{b}c+ab\overline{c}$ *está expresada en forma canónica disyuntiva como suma de cuatro minitérminos.*

   Ejemplo 5.4: *El minitérmino* $\overline{a}\overline{b}\overline{c}$ *vale 1 si, y solo si, las tres variables valen 0. El resto de los minitérminos que pueden formarse con a, b y c valdrán 0 para esta combinación de valores – porque en todos ellos alguno de los elementos del producto que constituye el minitérmino tomará el valor 0.*

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

2. La segunda forma canónica, conocida como *forma canónica conjuntiva*, consiste en un producto de *maxitérminos*. Un maxitérmino es una suma en la que aparecen, al igual que en los minitérminos, todas las variables, una sola vez cada una, pudiendo, también, estar o no complementadas. Nuevamente, esto implica que una función de N variables puede tener hasta 2^N maxitérminos. Un maxitérmino se caracteriza porque sólo vale 0 para una determinada combinación de valores de las variables que lo forman y porque, para esa combinación, cualquier otro maxitérmino vale 1.

Ejemplo 5.5: *La función* $G=(\overline{a}+\overline{b}+\overline{c})\cdot(\overline{a}+b+\overline{c})\cdot(a+b+c)\cdot(a+\overline{b}+\overline{c})$, *está expresada en forma canónica conjuntiva como producto de cuatro maxitérminos.*

Ejemplo 5.6: *El maxitérmino* $a+\overline{b}+\overline{c}$ *sólo vale 0 si a es 0, b es 1 y c es 1. En cualquier otro maxitérmino formado con a, b y c alguno de los elementos de la suma valdrá 1 para esta combinación y por tanto todos ellos valdrán también 1.*

Ejemplo 5.7: *Con tres variables a, b y c, se pueden formar los siguientes minitérminos y maxitérminos (están todos los posibles):*

Minitérminos	***Maxitérminos***
$\overline{a}\overline{b}\overline{c}$	$\overline{a}+\overline{b}+\overline{c}$
$\overline{a}\overline{b}c$	$\overline{a}+\overline{b}+c$
$\overline{a}b\overline{c}$	$\overline{a}+b+\overline{c}$
$\overline{a}bc$	$\overline{a}+b+c$
$a\overline{b}\overline{c}$	$a+\overline{b}+\overline{c}$
$a\overline{b}c$	$a+\overline{b}+c$
$ab\overline{c}$	$a+b+\overline{c}$
abc	$a+b+c$

Ejemplo 5.8: *Las funciones* $H=\overline{a}c+a(\overline{b}c+b\overline{c})$ *y* $J=(\overline{a}+\overline{c})\cdot(a+(b+c)\cdot(\overline{b}+\overline{c}))$ *no están expresadas en forma canónica.*

<u>*Tablas de verdad de la forma canónica disyuntiva*</u>

Una función expresada en forma canónica como suma de minitérminos vale 1 para el conjunto de combinaciones de valores que hacen que alguno de los minitérminos que contiene valga 1 -el número de combinaciones para las que la función vale 1 es igual al número de minitérminos de la función. Para cualquier otra combinación la función vale 0.

Ejemplo 5.9: *La función* $F=\overline{a}\overline{b}c+a\overline{b}\overline{c}+abc$ *tiene tres minitérminos. La combinación concreta de valores que hace que un minitérmino valga 1 depende exclusivamente de que las variables que lo forman aparezcan complementadas o sin complementar. Si una variable aparece sin complementar debe valer 1 para que el minitérmino valga 1; si aparece complementada, debe valer 0. Las tres combinaciones de valores para las que la función vale 1 son:*

a = 0, b = 0, c = 1, por $\overline{a}\overline{b}c$, *porque* $\overline{0}\cdot\overline{0}\cdot 1=1$
a = 1, b = 0, c = 0, por $a\overline{b}\overline{c}$, *porque* $1\cdot\overline{0}\cdot\overline{0}=1$
a = 1, b = 1, c = 1, por abc, *porque* $1\cdot 1\cdot 1=1$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

La tabla de verdad de esta función sería:

a	***b***	***c***	***F***
0	*0*	*0*	*0*
0	*0*	*1*	*1*
0	*1*	*0*	*0*
0	*1*	*1*	*0*
1	*0*	*0*	*1*
1	*0*	*1*	*0*
1	*1*	*0*	*0*
1	*1*	*1*	*1*

Obtención de la forma canónica disyuntiva a partir de una tabla de verdad

De una tabla de verdad podemos obtener el conjunto de combinaciones de valores de las variables para las que la función vale 1. Si construimos una función que sume los minitérminos que valen 1 para esas combinaciones, la expresión resultante se corresponderá con la función definida por la tabla de verdad.

Por ejemplo, la tabla de verdad de la figura, define una función de 3 variables.

a	**b**	**c**	**F**
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

La función vale 1 para cuatro combinaciones de valores de las variables *a*, *b* y *c*: 000, 001, 101 y 111. Los minitérminos correspondientes a estas combinaciones se obtienen complementado las variables que valen 0 y dejando sin complementar las que valen 1. Los minitérminos son:

$$\begin{array}{ll} 000 & \bar{a}\bar{b}\bar{c} \\ 001 & \bar{a}\bar{b}c \\ 101 & a\bar{b}c \\ 111 & abc \end{array}$$

La función definida por la tabla de verdad puede expresarse algebraicamente como suma de estos cuatro minitérminos:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

$$F = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + a\bar{b}c + abc$$

En resumen, para obtener la expresión algebraica correspondiente a la forma canónica disyuntiva (suma de minitérminos) de una función, a partir de su tabla de verdad, hay que realizar las siguientes operaciones:

1. Seleccionar las filas de la tabla en las que la función vale 1.
2. La combinación de valores de las variables en cada fila seleccionada se corresponde con un minitérmino: el único que se evalúa a 1 para esa combinación. Debe deducirse la expresión de cada uno de estos minitérminos, teniendo en cuenta que las variables que valen 0 en la combinación aparecen en él negadas y las que valen 1 sin negar
3. La expresión de la función se construye sumando los minitérminos correspondientes a todas las filas seleccionadas.

Tablas de verdad de la forma canónica conjuntiva

Las funciones expresadas en forma canónica conjuntiva valen 0 para las combinaciones de valores de variables que hacen que alguno de los maxitérminos que contiene valga 0. El número de estas combinaciones será igual, por tanto, al de maxitérminos de la función. Para cualquier otra combinación la función vale 1.

Ejemplo 5.10: *La función* $F = (\bar{a} + \bar{b} + c) \cdot (a + \bar{b} + \bar{c}) \cdot (a + b + c)$ *tiene tres maxitérminos.*

La combinación concreta de valores que hace que un maxitérmino valga 0 depende exclusivamente de que las variables que lo forman aparezcan complementadas o sin complementar. Si una variable aparece sin complementar debe valer 0 para que el maxitérmino valga 0, si aparece complementada, debe valer 1. Las tres combinaciones de valores correspondientes a los maxitérminos de la función son:

a = 1, b = 1, c = 0, para $\bar{a} + \bar{b} + c$ *porque* $\bar{1} + \bar{1} + 0 = 0$
a = 0, b = 1, c = 1, para $a + \bar{b} + \bar{c}$ *porque* $0 + \bar{1} + \bar{1} = 0$
a = 0, b = 0, c = 0, para $a + b + c$ *porque* $0 + 0 + 0 = 0$

La tabla de verdad de esta función sería:

a	*b*	*c*	*F*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Obtención de la forma canónica conjuntiva a partir de una tabla de verdad

Si seleccionamos en la tabla de verdad de una función el conjunto de combinaciones de valores de las variables para las que la función vale 0, y construimos una expresión que realice el producto de los maxitérminos que valen 0 para esas combinaciones, dicha expresión se corresponderá con la función definida por la tabla de verdad.

Veamos un ejemplo. La tabla de verdad de la figura, define una función de 2 variables.

a	b	F
0	0	1
0	1	0
1	0	0
1	1	0

La función vale 0 para tres combinaciones de valores de las variables *a* y *b*: 01, 10 y 11. Los maxitérminos correspondientes a estas combinaciones se obtienen complementado las variables que valen 1 y dejando sin complementar las que valen 0. Son los siguientes:

$$01 \quad a+\overline{b}$$
$$10 \quad \overline{a}+b$$
$$11 \quad \overline{a}+\overline{b}$$

La forma canónica conjuntiva correspondiente a la función definida en la tabla de verdad consiste en el producto de estos tres maxitérminos:

$$F=(a+\overline{b})\cdot(\overline{a}+b)\cdot(\overline{a}+\overline{b})$$

En definitiva, para obtener la expresión algebraica correspondiente a la forma canónica conjuntiva (producto de maxitérminos) de una función, a partir de su tabla de verdad, hay que realizar las siguientes operaciones:

1. Seleccionar las filas de la tabla en las que la función vale 0.

2. La combinación de valores de las variables en cada fila seleccionada se corresponde con un maxitérmino, el único que se evalúa a 0 para esa combinación. Debe deducirse la expresión de cada uno de estos maxitérminos, teniendo en cuenta que las variables que valen 0 en la combinación aparecen en él sin negar y las que valen 1, negadas.

3. La expresión de la función se construye haciendo el producto de los maxitérminos correspondientes a todas las filas seleccionadas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejemplo 5.11: *Vamos a obtener la expresión de la forma canónica conjuntiva y disyuntiva para las funciones definidas por las siguientes tablas de verdad:*

a	*b*	*c*	*d*	*F*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

a	*b*	*c*	*d*	*G*
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

En la tabla de verdad de la función F, hay cinco combinaciones para las que la función vale 0: 0000, 0001, 0010, 0100 y 0111. Por tanto, la forma canónica conjuntiva de la función contendrá los 5 maxitérminos siguientes:

$$a+b+c+d$$
$$a+b+c+\overline{d}$$
$$a+b+\overline{c}+d$$
$$a+\overline{b}+c+d$$
$$a+\overline{b}+\overline{c}+\overline{d}$$

Y su expresión algebraica sería:

$$F=(a+b+c+d)(a+b+c+\overline{d})(a+b+\overline{c}+d)(a+\overline{b}+c+d)(a+\overline{b}+\overline{c}+\overline{d})$$

La forma canónica disyuntiva de la función F tiene once minitérminos, los correspondientes a las combinaciones: 0011, 0101, 0110, 1000, 1001, 1010, 1011, 1100, 1101, 1110 y 1111. La expresión de la función en esta forma canónica es:

$$F=\overline{a}\overline{b}cd+\overline{a}b\overline{c}d+\overline{a}bc\overline{d}+a\overline{b}\overline{c}\overline{d}+a\overline{b}\overline{c}d+a\overline{b}c\overline{d}+a\overline{b}cd+ab\overline{c}\overline{d}+ab\overline{c}d+abc\overline{d}+abcd$$

En la función G hay dos combinaciones de valores para los que vale 0 y catorce para las que vale 1.

La forma canónica conjuntiva de G es:

$$G=(a+\overline{b}+\overline{c}+d)(\overline{a}+b+\overline{c}+\overline{d})$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

La forma canónica disyuntiva de G es:

$$G = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd + \bar{a}b\bar{c}\bar{d} + \bar{a}b\bar{c}d + \bar{a}bcd + a\bar{b}\bar{c}\bar{d} + a\bar{b}\bar{c}d +$$
$$+ a\bar{b}c\bar{d} + ab\bar{c}\bar{d} + ab\bar{c}d + abc\bar{d} + abcd$$

Como puede observarse en el ejemplo, la expresión normalizada de una función puede ser considerablemente larga.

Puesto que un minitérmino (o maxitérmino) tiene asociado una sola combinación de valores para la que vale 1 (0), puede denotarse por el valor binario correspondiente a esa combinación (en el caso de los maxitérminos, por el de la combinación que hace que valga 0). Esta característica se utiliza para expresar las formas canónicas con una notación simple y compacta. En esta notación un minitérmino se identifica mediante la letra *m* minúscula con un subíndice; el subíndice indica el valor binario de la combinación de valores que hace que el minitérmino valga 1.

Por ejemplo, m_4, en una función de 5 variables (*a*, *b*, *c*, *d*, *e*), es el minitérmino que vale 1 para la combinación 00100, es decir, $\bar{a}\bar{b}c\bar{d}\bar{e}$. En una función de 3 variables (*x*, *y*, *z*), m_4 se corresponde con el minitérmino $x\bar{y}\bar{z}$.

La expresión $F = m_1 + m_3 + m_6$, correspondiente a una función de tres variables (*a*, *b*, *c*), equivale a $F = \bar{a}\bar{b}c + \bar{a}bc + ab\bar{c}$. Las variables con las que se construyen los minitérminos –las variables de la función- suelen indicarse en la propia expresión: $F(a,b,c) = m_1 + m_3 + m_6$.

Es frecuente que al utilizar esta notación se haga uso del símbolo Σ (sumatorio) para simplificar aun más la expresión.

Ejemplo 5.12: *La expresión* $F(a,b,c) = m_1 + m_3 + m_6$ *equivale a* $F(a,b,c) = \sum m(1,3,6)$.

Para la forma canónica conjuntiva (producto de sumas) la notación es análoga. Los maxitérminos se indican con la letra *M* mayúscula y el subíndice correspondiente a la combinación de valores para la que vale 0.

Ejemplo 5.13: *La expresión* $F(a,b,c) = M_1 \cdot M_3 \cdot M_6$, *que también puede expresarse como* $F(a,b,c) = \prod M(1,3,6)$, *equivale a* $F(a,b,c) = (a + b + \bar{c}) \cdot (a + \bar{b} + \bar{c}) \cdot (\bar{a} + \bar{b} + c)$.

Ejemplo 5.14: *Las funciones F y G del ejemplo 4.12 se representarían de la siguiente manera:*

$$F(a,b,c,d) = \sum m(3,5,6,8,9,10,11,12,13,14,15)$$
$$F(a,b,c,d) = \prod M(0,1,2,4,7)$$
$$G(a,b,c,d) = \sum m(0,1,2,3,4,5,7,8,9,10,12,13,1,4,15)$$
$$G(a,b,c,d) = \prod M(6,11)$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P11

Concepto y caracterización de los sistemas combinacionales

Sistemas digitales

- Sistemas que realizan el procesado de información digital
- Sistema = interfaz + funcionalidad

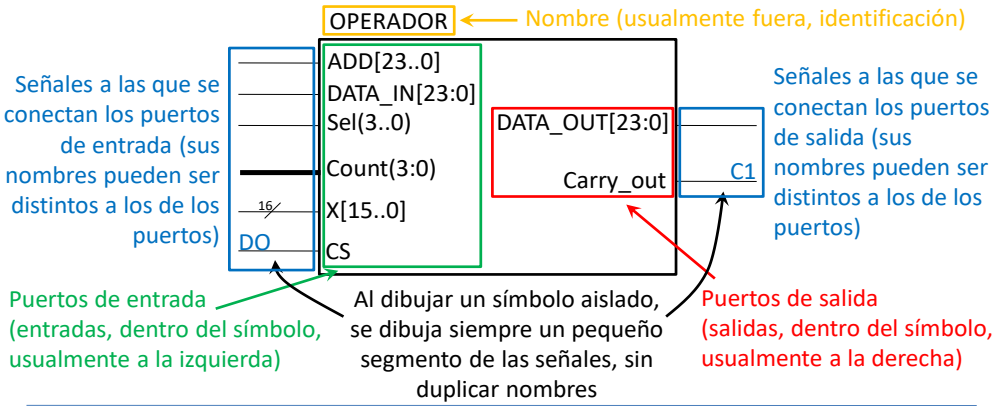
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas digitales: interfaz y símbolo

- Símbolo: representación gráfica de la interfaz
  - Representa al sistema dentro de un circuito mayor
  - No es una descripción completa de la interfaz (no define cómo se codifican los puertos)



BT1_S5 BT1_P11_3

Sistemas digitales: funcionalidad

- La descripción de la funcionalidad de un sistema ha de ser:
  - Completa: el comportamiento de las salidas debe estar definido para todas las combinaciones de las entradas
  - No ambigua: la descripción de la funcionalidad no debe dar lugar a varias interpretaciones
  - Consistente: una parte de la funcionalidad no debe entrar en conflicto con otras

BT1_S5 BT1_P11_4

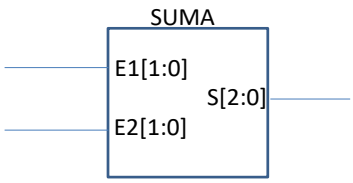
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas digitales: ejemplos

- *Ejemplo 1: Sumador. El sistema realiza la suma de dos datos de entrada de 2 bits en binario natural. La salida se genera en 3 bits.*

INTERFAZ	**ENTRADAS**	
	E1[1:0] E2[1:0]	Binario natural
	SALIDAS	
	S[2:0]	Binario natural
FUNCION	S=E1+E2	El sistema obtiene en la salida la suma de las dos entradas



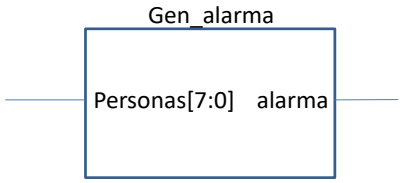
Sería incompleta si no se indicara el número de bits o el código utilizado.

BT1_S5 BT1_P11_5

Sistemas digitales: ejemplos

- *Ejemplo 2: Alarma. El sistema tiene como entrada el número de personas en un local (binario natural, 8 bits) y genera una salida de alarma (a 1 lógico) cuando este número supera los 200.*

INTERFAZ	**ENTRADAS**	
	Personas[7:0]	Número de personas en el local, en binario natural
	SALIDAS	
	alarma	
FUNCION	•alarma = 0 si personas ≤ 200 •alarma = 1 si personas > 200	El sistema activa la alarma si el número de personas excede las 200



Sería ambigua si se utilizara la expresión "al llegar a los 200", no queda claro que pasa con 201 o más

BT1_S5 BT1_P11_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

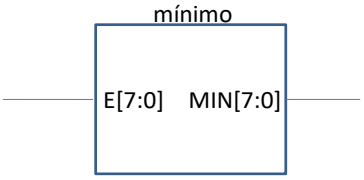
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas digitales: ejemplos

- *Ejemplo 3: Mínimo. El sistema obtiene en cada instante a su salida el mínimo de entre todos los valores presentados a la entrada (el valor actual y los presentados anteriormente). La entrada y la salida están codificadas en complemento a 2 de 8 bits.*

INTERFAZ	ENTRADAS	
	E [7:0]	Complemento a 2
	SALIDAS	
	MIN [7:0]	Complemento a 2
FUNCION	S = min (E1, E2,…, EN)	La salida en cada instante es el mínimo de los valores presentados a la entrada hasta ese instante



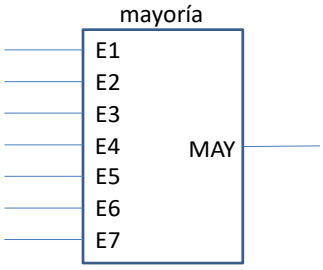
Sería inconsistente si se utilizara una entrada en c.a.2 y una salida en binario natural.

BT1_S5 BT1_P11_7

Sistemas digitales: ejemplos

- *Ejemplo 4: Mayoría. El sistema tiene 7 entradas y una salida, todas de un bit. Cada entrada representa un voto afirmativo (1) o negativo (0). La salida indica si el voto mayoritario es afirmativo (1) o negativo (0).*

INTERFAZ	ENTRADAS	
	E1, E2, E3, E4, E5, E6, E7	7 entradas de un bit
	SALIDAS	
	MAY	Salida de un bit
FUNCION		La salida es 1 si la mayoría de las entradas son 1; en caso contrario es 0



Sería incompleta si fueran pares y no se indicara que hacer en caso de empate.

BT1_S5 BT1_P11_8

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas digitales: ejemplos

- *Ejemplo 5: Detección de monedas. El sistema recibe la información del peso de una moneda en el rango entre 0 y 30 g. con una resolución de 5 g. y decide el tipo de moneda en función del peso. Si pesa menos de 5 g es de 5 c, entre 5 g y 9 g es de 10 c, entre 10 g y 19 g es de 50 c, entre 20 g Y 24 g es de 1 €, a partir de 25 g es de 2 €.*

BT1_S5 BT1_P11_9

Sistemas digitales: ejemplos

- *Codificación de las entradas:*
  - *Una opción sería proveer al sistema de 7 entradas, cada una de ellas se activaría (1) cuando se diera una de las 7 combinaciones.*
  - *Otra opción sería codificar cada combinación utilizando un número de variables más pequeño. En este caso el número mínimo de variables es 3.*

Peso de la moneda	Entradas (cod. con 3 bits)			Entradas (codificación con 7 bits)						
	e2	e1	e0	e6	e5	e4	e3	e2	e1	e0
0 g	0	0	0	0	0	0	0	0	0	1
5 g	0	0	1	0	0	0	0	0	1	0
10 g	0	1	0	0	0	0	0	1	0	0
15 g	0	1	1	0	0	0	1	0	0	0
20 g	1	0	0	0	0	1	0	0	0	0
25 g	1	0	1	0	1	0	0	0	0	0
30 g	1	1	0	1	0	0	0	0	0	0

BT1_S5 BT1_P11_10

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas digitales: ejemplos

- *Codificación de las salidas: hay que distinguir entre 5 monedas diferentes*
  - *Puede dotarse al sistema de 5 salidas*
  - *Pueden codificarse las 5 combinaciones con 3 variables*

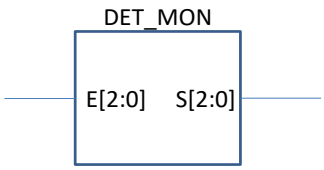
Valor de la moneda	Salidas (cod. con 3 bits)			Salidas (codificación con 5 bits)				
	s2	s1	s0	s4	s3	s2	s1	s0
5 c	0	0	0	0	0	0	0	1
10 c	0	0	1	0	0	0	1	0
50 c	0	1	0	0	0	1	0	0
1 €	0	1	1	0	1	0	0	0
2 €	1	0	0	1	0	0	0	0

BT1_S5 BT1_P11_11

Sistemas digitales: ejemplos

- *Interfaz y función*

INTERFAZ	ENTRADAS	
	E[2:0]	Bus de 3 bits
	SALIDAS	
	S[2:0]	Bus de 3 bits
FUNCION	peso < 5g => 5c 5g ≤ peso < 10g => 10c 10g ≤ peso < 19g => 50c 20g ≤ peso < 25g => 1€ 25g ≤ peso ≤ 30g => 2€	El sistema detecta si la moneda es de 5 c, 10 c, 50 c, 1 € o 2 € en función de su peso.



BT1_S5 BT1_P11_12

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

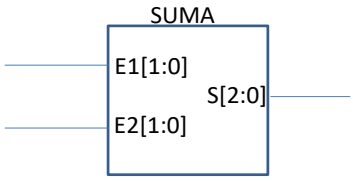
Sistemas (digitales) combinacionales

- Sistemas digitales cuyas salidas en un instante t dependen exclusivamente de los valores de las entradas en ese mismo instante t
- Sistemas "sin memoria"

BT1_S5 BT1_P11_13

Sistemas combinacionales

- *Ejemplo : Sumador.*



Entradas	E2	E1	E2	E1	E2	E1	E2	E1	E2	E1
	00	00	01	00	01	11	00	00	00	01
Salida	000		001		100		000		001	

BT1_S5 BT1_P11_14

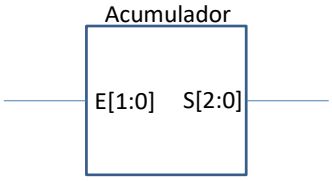
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sistemas no combinacionales

- *Ejemplo : Acumulador*.



Entradas	E	E	E	E	E
	00	01	11	01	01
Salida	000	001	100	101	110

BT1_S5 BT1_P11_15

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P12

Resolución de Dudas sobre Algebra de Boole.

Notación y evaluación de expresiones.
Minitérminos y Maxitérminos

Álgebra de Conmutación

- B (A ∈ {0,1}, +, ·, ¯)
- Axiomas:

Axioma	Enunciado	
A1	Propiedad conmutativa	$\forall a,b \in A: a+b=b+a$ $\forall a,b \in A: a \cdot b = b \cdot a$
A2	Propiedad asociativa	$\forall a,b,c \in A: (a+b)+c=a+(b+c)$ $\forall a,b,c \in A: (a \cdot b) \cdot c = a \cdot (b \cdot c)$
A3	Existencia de elemento neutro para + y ·	$\forall a \in A, a+0=a$ $\forall a \in A, a \cdot 1 = a$
A4	Distributividad de + con respecto a · y de · con respecto a +	$\forall a,b,c \in A: a+(b \cdot c)=(a+b) \cdot (a+c)$ $\forall a,b,c \in A: a \cdot (b+c)=(a \cdot b)+(a \cdot c)$
A5	Cada elemento de A tiene un complemento que es único	$\forall a \in A, \exists \bar{a} \in A / a+\bar{a}=1$ $\forall a \in A, \exists \bar{a} \in A / a \cdot \bar{a}=0$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Álgebra de Conmutación

- Definición de las operaciones:

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

A	comp (A)
0	1
1	0

BT1_S5 BT1_P12_3

Álgebra de Conmutación

- Reglas del álgebra:
  - A + 0 = A
  - A + 1 = 1
  - A · 0 = 0
  - A · 1 = A
  - A + A = A
  - A + /A = 1
  - A · A = A
  - A · (/A) = 0
  - //A = A

BT1_S5 BT1_P12_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Notación en Funciones Booleanas

- Ejemplo de función booleana.

$$f = (\overline{a+b} + \bar{c}\,d)\overline{\bar{b}\,c}$$

- Ausencia de '·'
- Precedencia: '¯', '·' y '+' (por ese orden)
- Los paréntesis y el operador complemento modifican la precedencia

BT1_S5 BT1_P12_5

Evaluación de funciones booleanas

$$f = (\overline{a+b} + \bar{c}\,d)\overline{\bar{b}\,c}$$

a	b	c	d
0	1	0	1

1. $\overline{a+b} = \overline{0+1} = \bar{1} = 0$

   $\bar{c}\,d = 1 \cdot 1 = 1$

2. $(\overline{a+b} + \bar{c}\,d) = 0 + 1 = 1$

   $\overline{\bar{b}\,c} = \overline{\bar{1} \cdot 0} = \overline{0 \cdot 0} = \bar{0} = 1$

3. $(\overline{a+b} + \bar{c}\,d)\overline{\bar{b}\,c} = 1 \cdot 1 = 1$

BT1_S5 BT1_P12_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Evaluación eficiente de funciones booleanas

$$f = a(\overline{a+b} + \bar{c}d)\overline{\bar{b}c} + d$$

a	b	c	d
0	1	0	1

Si d = 1 la función vale 1, independientemente de los valores que tomen el resto de variables (x+1=1)

$$f = a(\overline{a+b} + \bar{c}d)\overline{\bar{b}c} + d$$

a	b	c	d
0	1	0	0

Si d = 0, cuando a=0 la función vale 0, independientemente de los valores que tomen el resto de variables (x·0=0)

BT1_S5 BT1_P12_7

Expresiones canónicas: disyuntiva (suma de minitérminos)

minitérmino	Subíndice (binario)	expresión
m_0	00	$\bar{a}\,\bar{b}$
m_1	01	$\bar{a}\,b$
m_2	10	$a\,\bar{b}$
m_3	11	$a\,b$

El minitérmino vale '1' cuando la combinación de las variables (en este caso a y b) coincide con el valor binario del subíndice

La función vale '1' para las combinaciones en las que alguno de sus minitérminos vale '1'

- Ejemplos:

función	Notación abreviada
$f = \bar{a}\,\bar{b} + \bar{a}\,b + a\,\bar{b}$	$f(a,b) = m_0 + m_1 + m_2 = \sum m(0,1,2)$
$f = a\,b$	$f(a,b) = m_3$
$f = \bar{a}\,\bar{b} + a\,b$	$f(a,b) = m_0 + m_3 = \sum m(0,3)$
$f = \bar{a}\,\bar{b} + \bar{a}\,b + a\,\bar{b} + a\,b$	$f(a,b) = m_0 + m_1 + m_2 + m_3 = \sum m(0,1,2,3)$

BT1_S5 BT1_P12_8

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Expresiones canónicas: disyuntiva (suma de minitérminos)

minitérmino	subindice	expresión
m_0	000	$\bar{a}\bar{b}\bar{c}$
m_1	001	$\bar{a}\bar{b}c$
m_2	010	$\bar{a}b\bar{c}$
m_3	011	$\bar{a}bc$

minitérmino	subindice	expresión
m_4	100	$a\bar{b}\bar{c}$
m_5	101	$a\bar{b}c$
m_6	110	$ab\bar{c}$
m_7	111	abc

- Ejemplos:

función	Notación abreviada
$f=\bar{a}\bar{b}\bar{c}+\bar{a}bc+a\bar{b}\bar{c}$	$f(a,b,c)=m_0+m_3+m_4=\sum m(0,3,4)$
$f=abc$	$f(a,b,c)=m_7$
$f=\bar{a}\bar{b}\bar{c}+a\bar{b}\bar{c}+a\bar{b}c+abc$	$f(a,b,c)=m_0+m_4+m_5+m_7=\sum m(0,4,5,7)$
$f=\bar{a}\bar{b}\bar{c}+\bar{a}\bar{b}c+\bar{a}b\bar{c}+\bar{a}bc$ $+a\bar{b}\bar{c}+a\bar{b}c+ab\bar{c}+abc$	$f(a,b,c)=m_0+m_1+m_2+m_3$ $+m_4+m_5+m_6+m_7=\sum m(0,1,2,3,4,5,6,7)$

BT1_S5 BT1_P12_9

Evaluación de expresiones canónicas

$f=\bar{a}\bar{b}+a\bar{b}+\bar{a}b$

a	b
0	1

$f=\bar{a}\bar{b}\bar{c}+abc+a\bar{b}\bar{c}+a\bar{b}c$

a	b	c
0	1	0

$f=\bar{a}\bar{b}\bar{c}\bar{d}+abcd+a\bar{b}\bar{c}d$

a	b	c	d
0	1	0	1

- Solo un minitérmino puede valer 1 para cada combinación de variables

BT1_S5 BT1_P12_10

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Expresiones canónicas: conjuntiva (prod. de maxitérminos)

maxitérmino	Subíndice (binario)	expresión
M_0	00	$a+b$
M_1	01	$a+\overline{b}$
M_2	10	$\overline{a}+b$
M_3	11	$\overline{a}+\overline{b}$

El maxitérmino vale '0' cuando la combinación de las variables (en este caso a y b) coincide con el valor binario del subíndice

La función vale '0' para las combinaciones en las que alguno de sus maxitérminos vale '0'

- Ejemplos:

función	Notación abreviada
$f=(a+b)(a+\overline{b})(\overline{a}+b)$	$f(a,b)=M_0M_1M_2=\prod M(0,1,2)$
$f=\overline{a}+\overline{b}$	$f(a,b)=M_3$
$f=(a+b)\ (\overline{a}+\overline{b})$	$f(a,b)=M_0M_3=\prod M(0,3)$
$f=(\overline{a}+\overline{b})\ (\overline{a}+b)(a+\overline{b})(a+b)$	$f(a,b)=M_0M_1M_2M_3=\prod M(0,1,2,3)$

BT1_S5 BT1_P12_11

Expresiones canónicas: conjuntiva (prod. de maxitérminos)

maxitérmino	subíndice	expresión
M_0	000	$a+b+c$
M_1	001	$a+b+\overline{c}$
M_2	010	$a+\overline{b}+c$
M_3	011	$a+\overline{b}+\overline{c}$

maxitérmino	subíndice	expresión
M_4	100	$\overline{a}+b+c$
M_5	101	$\overline{a}+b+\overline{c}$
M_6	110	$\overline{a}+\overline{b}+c$
M_7	111	$\overline{a}+\overline{b}+\overline{c}$

- Ejemplos:

función	Notación abreviada
$f=(a+b+c)(a+\overline{b}+\overline{c})(\overline{a}+b+c)$	$f(a,b,c)=M_0M_3M_4=\prod M(0,3,4)$
$f=\overline{a}+\overline{b}+\overline{c}$	$f(a,b,c)=M_7$
$f=(\overline{a}+\overline{b}+\overline{c})(\overline{a}+b+c)(a+b+c)$	$f(a,b,c)=M_0M_4M_7=\prod M(0,4,7)$
$f=(\overline{a}+\overline{b}+\overline{c})(a+b+c)(\overline{a}+b+c)$ $(a+\overline{b}+\overline{c})(a+\overline{b}+c)(a+b+\overline{c})$	$f(a,b,c)=M_0M_1M_2M_3M_4M_7$ $=\prod M(0,1,2,3,4,7)$

BT1_S5 BT1_P12_12

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Evaluación de expresiones canónicas

$$f = (a+b)\ (\bar{a}+\bar{b})$$

a	b
0	1

$$f = (\bar{a}+\bar{b}+\bar{c})(a+b+c)(\bar{a}+b+c)$$

a	b	c
1	0	0

$$f = (\bar{a}+\bar{b}+\bar{c}+\bar{d})(a+b+c+d)$$
$$(a+\bar{b}+\bar{c}+d)$$

a	b	c	d
0	1	1	0

- Solo un maxitérmino puede valer 0 para cada combinación de variables

BT1_S5 BT1_P12_13

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1_P13

Obtención de tablas de verdad. Síntesis de funciones.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad

- Dos variables
- Tres variables
- Cuatro variables
- …
- Multifunción

e_3	e_2	e_1	e_0	x	y	z
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	1	0	0
0	1	1	0	0	0	0
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	0	1	1	1	0	0
1	1	0	0	0	1	1
1	1	0	1	1	1	1
1	1	1	0	0	1	0
1	1	1	1	1	1	0

- Número de filas: $2^{\text{número de variables}}$
- Posibles funciones: $2^{\text{número de filas}}$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad

- Obtención a partir de una función

$$f = a + b\bar{c}$$

- *f=1 si a = 1*
- *f=1 si b = 1 y c = 0*
- *f=0 si a = 0 y b = 0*
- *f=0 si a = 0 y c = 1*

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

BT1_S5 BT1_P13_3

Tablas de verdad

- Obtención a partir de expresiones canónicas

$$f = \bar{a}\,\bar{b}cd + \bar{a}\,b\bar{c}\bar{d} + \bar{a}\,b\bar{c}d + \bar{a}\,bcd + a\,\bar{b}\bar{c}d$$
$$a\,\bar{b}cd + a\,b\bar{c}d + a\,bcd$$

$$f = \sum m(3,4,5,7,9,11,13,15)$$

min.	a	b	c	d	f
m_0	0	0	0	0	0
m_1	0	0	0	1	0
m_2	0	0	1	0	0
m_3	0	0	1	1	1
m_4	0	1	0	0	1
m_5	0	1	0	1	1
m_6	0	1	1	0	0
m_7	0	1	1	1	1
m_8	1	0	0	0	0
m_9	1	0	0	1	1
m_{10}	1	0	1	0	0
m_{11}	1	0	1	1	1
m_{12}	1	1	0	0	0
m_{13}	1	1	0	1	1
m_{14}	1	1	1	0	0
m_{15}	1	1	1	1	1

BT1_S5 BT1_P13_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad

- Obtención a partir de expresiones canónicas

$$f = (\bar{a}+\bar{b}+\bar{c}+\bar{d})(a+b+c+d)\ (a+\bar{b}+\bar{c}+d)\ (a+\bar{b}+\bar{c}+\bar{d})$$

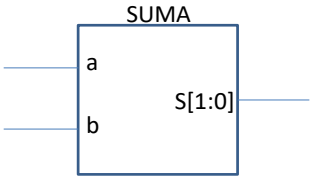
$$f = \prod M(0,6,7,15)$$

max.	a	b	c	d	f
M_0	0	0	0	0	0
M_1	0	0	0	1	1
M_2	0	0	1	0	1
M_3	0	0	1	1	1
M_4	0	1	0	0	1
M_5	0	1	0	1	1
M_6	0	1	1	0	0
M_7	0	1	1	1	0
M_8	1	0	0	0	1
M_9	1	0	0	1	1
M_{10}	1	0	1	0	1
M_{11}	1	0	1	1	1
M_{12}	1	1	0	0	1
M_{13}	1	1	0	1	1
M_{14}	1	1	1	0	1
M_{15}	1	1	1	1	0

BT1_S5 BT1_P13_5

Tablas de verdad

- Relación con la funcionalidad de un sistema combinacional
- *Ejemplo: Sumador de 1 bit. La salida es la suma binaria de las dos entradas.*



a	b	S_1	S_0	comentario
0	0	0	0	0+0 = 0
0	1	0	1	0+1 = 1
1	0	0	1	1+0 = 1
1	1	1	0	1+1 = 2

BT1_S5 BT1_P13_6

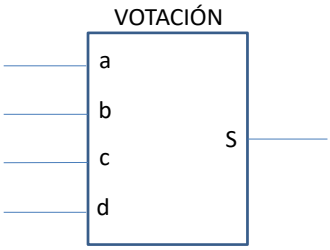
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad

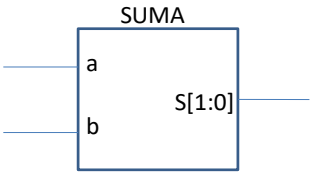
- Relación con la funcionalidad de un sistema combinacional
- *Ejemplo: Votación. 4 votantes. La salida es SI (1) si hay mayoría de votos a favor (1).*



a	b	c	d	s
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Síntesis de funciones

- Suma de minitérminos



Max.	a	b	s_1	s_0
M_0	0	0	0	0
M_1	0	1	0	1
M_2	1	0	0	1
M_3	1	1	1	0

$$s_0 = \sum m(1,2) = \bar{a}b + a\bar{b}$$

$$s_1 = m_3 = ab$$

- Producto de maxitérminos

$$s_0 = \prod M(0,3) = (a+b)(\bar{a}+\bar{b})$$

$$s_1 = \prod M(0,1,2) = (\bar{a}+b)(a+\bar{b})(a+b)$$

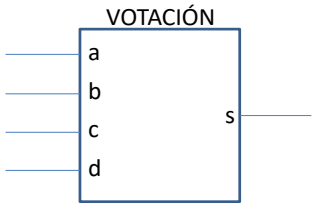
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Síntesis de funciones

- Suma de minitérminos



minitérm.	a	b	c	d	s
m_0	0	0	0	0	0
m_1	0	0	0	1	0
m_2	0	0	1	0	0
m_3	0	0	1	1	0
m_4	0	1	0	0	0
m_5	0	1	0	1	0
m_6	0	1	1	0	0
m_7	0	1	1	1	1
m_8	1	0	0	0	0
m_9	1	0	0	1	0
m_{10}	1	0	1	0	0
m_{11}	1	0	1	1	1
m_{12}	1	1	0	0	0
m_{13}	1	1	0	1	1
m_{14}	1	1	1	0	1
m_{15}	1	1	1	1	1

$$s = \sum m(7,11,13,14,15) \quad = \bar{a}bcd + a\bar{b}cd + a\bar{b}\bar{c}d + abc\bar{d} + abcd$$

- Producto de maxitérminos

$$s = \prod M(0,1,2,3,4,5,6,8,9,10,12)$$
$$= (a+b+c+d)(a+b+c+\bar{d})(a+b+\bar{c}+d)(a+b+\bar{c}+\bar{d})$$
$$(a+\bar{b}+c+d)(a+\bar{b}+c+\bar{d})(a+\bar{b}+\bar{c}+d)(\bar{a}+b+c+d)$$
$$(\bar{a}+b+c+\bar{d})(\bar{a}+b+\bar{c}+d)(\bar{a}+\bar{b}+c+d)$$

BT1_S5 BT1_P13_9

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre extracción de expresiones canónicas, cálculo booleano y obtención de tablas de verdad a partir de expresiones algebraicas.**	**CÓDIGO:** **BT1_AINP10**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Trabajo individual.	TIPO:	No presencial	DURACIÓN:	75 minutos

CALENDARIO:	Quinta Sesión. Previo a la Sexta Sesión	REQUISITOS:	

CRITERIO DE ÉXITO:	Realizar correctamente 5 de los 6 ejercicios de extracción de expresiones canónicas, 7 de los 8 de evaluación de expresiones, y 6 de los 8 de obtención de tablas de verdad.

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	puntos

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJEMPLO DE EXTRACCIÓN DE EXPRESIONES CANÓNICAS A PARTIR DE TABLAS DE VERDAD

Supongamos que se desea obtener una expresión algebraica para la siguiente tabla de verdad:

a	b	c	Valor
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

El procedimiento más sencillo es obtener una de las dos formas normales o canónicas, la disyuntiva (suma de minitérminos) o la conjuntiva (producto de maxitérminos). Para obtener la primera debemos fijarnos en las filas de la tabla para las que la función vale 1:

a	b	c	Valor	
0	0	0	0	
0	0	1	1	$m1$
0	1	0	1	$m2$
0	1	1	1	$m3$
1	0	0	1	$m4$
1	0	1	0	
1	1	0	1	$m6$
1	1	1	0	

A partir de esta información vemos que para obtener la función debemos sumar los minitérminos 1, 2, 3, 4 y 6:

$$f(a,b,c) = \underbrace{\overline{a}\overline{b}c}_{m1} + \underbrace{\overline{a}b\overline{c}}_{m2} + \underbrace{\overline{a}bc}_{m3} + \underbrace{a\overline{b}\overline{c}}_{m4} + \underbrace{ab\overline{c}}_{m6} = \sum m(1,2,3,4,6)$$

Para obtener la forma normal conjuntiva debemos fijarnos en las filas para las que la función vale 0:

a	b	c	Valor	
0	0	0	0	$M0$
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	1	
1	0	1	0	$M5$
1	1	0	1	
1	1	1	0	$M7$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Vemos que la función se puede escribir como el producto de los maxitérminos 0, 5 y 7:

$$f(a,b,c) = \underbrace{(a+b+c)}_{M0}\underbrace{(\overline{a}+b+\overline{c})}_{M5}\underbrace{(\overline{a}+\overline{b}+\overline{c})}_{M7} = \prod M(0,5,7)$$

Observe lo siguiente:

- Si el número de unos en la tabla de verdad es menor que el de ceros es conveniente utilizar la forma normal disyuntiva, porque tendrá menos términos que la forma normal conjuntiva. Por el contrario, si el número de unos es mayor que el de ceros será más ventajoso emplear la forma normal conjuntiva.

- Dada la tabla de verdad de una función lógica, el número total de términos que aparecen en ambas formas normales (es decir, el número de minitérminos que aparecen en la forma normal disyuntiva más el número de maxitérminos que aparecen en la forma normal conjuntiva) siempre es igual al número de filas de la tabla de verdad.

- Los minitérminos que aparecen en la forma normal disyuntiva de una función se corresponden con los maxitérminos que no aparecen en la forma normal conjuntiva. De igual modo, los maxitérminos que aparecen en la forma normal conjuntiva se corresponden con los minitérminos que no aparecen en la forma normal disyuntiva.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS DE EXTRACCIÓN DE EXPRESIONES CANÓNICAS A PARTIR DE TABLAS DE VERDAD

1.- Obtenga la forma normal conjuntiva de la función lógica cuya tabla de verdad se muestra a continuación:

a	b	c	Valor
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

2.- Obtenga la forma normal disyuntiva de la función lógica cuya tabla de verdad se muestra a continuación:

a	b	c	Valor
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

3.- Obtenga la forma normal conjuntiva de la función lógica cuya tabla de verdad se muestra a continuación:

a	b	c	Valor
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

4.- Obtenga la forma normal disyuntiva de la función lógica cuya tabla de verdad se muestra a continuación:

a	b	c	Valor
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

5.- Obtenga la forma normal conjuntiva de la función lógica cuya tabla de verdad se muestra a continuación:

a	b	c	Valor
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

6.- Obtenga la forma normal disyuntiva de la función lógica cuya tabla de verdad se muestra a continuación:

a	b	c	Valor
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	0
1	1	1	0

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJEMPLOS DE EVALUACIÓN DE EXPRESIONES BOOLEANAS

Ejemplo 1.- Supongamos que se desea evaluar la siguiente expresión para los valores a=1, b=1, c=0, d=0:

$$(a+b+c)(\overline{a}+b+\overline{c})+a\,\overline{b}\,c+\overline{d}$$

Como la expresión es una suma de tres términos, es suficiente comprobar si alguno de ellos vale 1 para los valores dados de las variables. Como puede verse fácilmente, el último término ($\overline{d}$) vale 1 para d=0, por lo que el valor de la expresión es 1 independientemente de lo que valgan los otros dos términos de la suma.

Supongamos ahora que se desea evaluar la expresión para los valores a=1, b=1, c=0, d=1. Vemos fácilmente en este caso que tanto $\overline{d}$ como $a\,\overline{b}\,c$ valen 0, por lo que el valor de la expresión coincide con el valor del término $(a+b+c)(\overline{a}+b+\overline{c})$. Esto es un producto de sumas, y es fácil comprobar que ambas sumas valen 1 para la combinación de valores de las variables dada, por lo que el valor de la expresión completa es nuevamente 1.

Ejemplo 2.- Supongamos que se desea evaluar la siguiente expresión para los valores a=1, b=1, c=0, d=0,e=1:

$$a(\overline{b}+c(\overline{d+e}))$$

Evaluando la expresión de fuera a dentro, comprobamos primero que a=1, por lo que hay que evaluar $\overline{b}+c(\overline{d+e})$. Como b=1, $\overline{b}$=0 y hay que evaluar $c(\overline{d+e})$. Viendo que c=0 ya se puede concluir que $c(\overline{d+e})$=0 independientemente del valor de $\overline{d+e}$, y por tanto el valor de la expresión es 0.

Si evaluamos la expresión para a=1, b=1, c=1, d=0, e=1 habría que llegar a evaluar $\overline{d+e}$, cuyo valor es 1 si y sólo si d=e=0. Como no es este el caso, $\overline{d+e}$=1 y por tanto el valor de la expresión es 1.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS DE EVALUACIÓN DE EXPRESIONES BOOLEANAS

Evalúe las siguientes expresiones booleanas para los valores indicados de sus variables.

Nº	Expresión	Valores de las variables	Resultado
1	$a+bc+\overline{a}b$	a=1, b=0, c=1	
2	$\overline{a}(\overline{c+\overline{b}})+ac$	a=1, b=1, c=0	
3	$a+a\overline{bc}+\overline{\overline{a+c}b}$	a=0, b=0, c=1	
4	$(\overline{a}+b+c)(a+\overline{b}+c)(a+b+\overline{c})$	a=1, b=1, c=1	
5	$\overline{(b+c)(a+b)(\overline{b}+\overline{c})}$	a=0, b=1, c=1	
6	$\overline{(a+b)}+bc+\overline{a}b+a\overline{b}$	a=1, b=1, c=0	
7	$a+\overline{a}(b+c)(\overline{a+c})$	a=0, b=0,c=0	
8	$ab+(\overline{a}+\overline{b}+c)(a+b+\overline{c})$	a=1, b=0, c=0	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJEMPLOS DE OBTENCIÓN DE TABLAS DE VERDAD A PARTIR DE EXPRESIONES ALGEBRAICAS

Ejemplo 1.- Supongamos que se desea obtener la tabla de verdad correspondiente a la siguiente expresión:

$$(a+b+c)(\overline{a}+b+\overline{c})(\overline{a}+\overline{b}+\overline{c})$$

Comenzaremos dibujando la tabla vacía. Como hay tres variables, la tabla tendrá $2^3 = 8$ filas:

a	b	c	Valor
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

La expresión corresponde a la forma normal conjuntiva (producto de maxitérminos) de una función lógica de tres variables, por lo que la expresión valdrá 0 para las tres combinaciones de valores a las que corresponden los tres maxitérminos presentes:

a	b	c	Valor
0	0	0	0
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	0
1	1	0	
1	1	1	0

En el resto de casos la función vale 1, por lo que podemos completar el resto de la tabla:

a	b	c	Valor
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejemplo 2.- Supongamos que se desea obtener la tabla de verdad correspondiente a la siguiente expresión:

$$(a+b+c)(\overline{a}+b+\overline{c})+a\,\overline{b}\,c+\overline{d}$$

Comenzaremos dibujando la tabla vacía. Como hay cuatro variables, la tabla tendrá 2^4=16 filas (por economía de espacio se representa la tabla partida en dos):

a	b	c	d	Valor
0	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	

a	b	c	d	Valor
1	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	

La expresión es una suma de tres términos: $(a+b+c)(\overline{a}+b+\overline{c})$, $a\,\overline{b}\,c$, y $\overline{d}$. Comenzaremos buscando las combinaciones de entrada que hacen que cada uno de estos términos valga 1, ya que en una suma lógica el resultado es 1 si uno cualquiera de los operandos es 1. Empezaremos examinando el término que sea más general; es decir, aquel que fije el valor de la expresión para más combinaciones de valores de las variables. En este caso comenzamos por el último término, $\overline{d}$, que vale 1 para las ocho combinaciones en las que d=0. De este modo podemos rellenar ya la mitad de las filas de la tabla de verdad:

a	b	c	d	Valor
0	0	0	0	1
0	0	0	1	
0	0	1	0	1
0	0	1	1	
0	1	0	0	1
0	1	0	1	
0	1	1	0	1
0	1	1	1	

a	b	c	d	Valor
1	0	0	0	1
1	0	0	1	
1	0	1	0	1
1	0	1	1	
1	1	0	0	1
1	1	0	1	
1	1	1	0	1
1	1	1	1	

El término central, $a\,\overline{b}\,c$, vale 1 para las dos combinaciones en las que a=1, b=0 y c=1. Una de ellas (a=1, b=0, c=1 y d=0) ya tiene valor asignado en el paso anterior, pero la otra no, así que lo anotamos ahora:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

a	b	c	d	Valor
0	0	0	0	1
0	0	0	1	
0	0	1	0	1
0	0	1	1	
0	1	0	0	1
0	1	0	1	
0	1	1	0	1
0	1	1	1	

a	b	c	d	Valor
1	0	0	0	1
1	0	0	1	
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	
1	1	1	0	1
1	1	1	1	

En el resto de las filas el valor de los dos términos analizados hasta ahora es 0, por lo que el valor de la expresión en esas filas coincide con el valor de $(a+b+c)(\overline{a}+b+\overline{c})$. Esto es un producto de sumas, que vale 0 si alguna de las sumas vale 0, o 1 en caso contrario. La suma $a+b+c$ vale 0 para a=b=c=0, y $\overline{a}+b+\overline{c}$ vale 0 para a=1, b=0 y c=1, de modo que la expresión vale 0 para aquellas de las cuatro combinaciones 0000, 0001, 1010 y 1011 (valores de a, b, c y d, respectivamente) que no tuvieran ya un valor anotado:

a	b	c	d	Valor
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	
0	1	0	0	1
0	1	0	1	
0	1	1	0	1
0	1	1	1	

a	b	c	d	Valor
1	0	0	0	1
1	0	0	1	
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	
1	1	1	0	1
1	1	1	1	

Fíjese en que si la fila ya tenía un 1 anotado significa que alguno de los otros dos términos que ya habíamos analizado vale 1 para esa combinación de valores, por lo que el valor de la suma final para ese caso no depende ya del término que estamos analizando ahora.

Como decíamos antes, para el resto de combinaciones de valores el producto de sumas vale 1, de modo que ya podemos completar la tabla:

a	b	c	d	Valor
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1

a	b	c	d	Valor
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejemplo 3.- Supongamos que se desea obtener la tabla de verdad que corresponde a la siguiente expresión:

$$a(\overline{b} + c(\overline{d + e}))$$

Comenzamos dibujando la tabla vacía. En este caso tenemos cinco variables, luego la tabla de verdad tendrá 2^5=32 filas:

a	b	c	d	e	Valor
0	0	0	0	0	
0	0	0	0	1	
0	0	0	1	0	
0	0	0	1	1	
0	0	1	0	0	
0	0	1	0	1	
0	0	1	1	0	
0	0	1	1	1	
0	1	0	0	0	
0	1	0	0	1	
0	1	0	1	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	0	1	
0	1	1	1	0	
0	1	1	1	1	

a	b	c	d	e	Valor
1	0	0	0	0	
1	0	0	0	1	
1	0	0	1	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	0	1	
1	0	1	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Evaluando la expresión de fuera a dentro, comprobamos que la operación más externa es el producto de a por otra subexpresión, luego siempre que $a = 0$ la expresión valdrá 0. Esto nos permite rellenar ya la mitad de la tabla:

a	b	c	d	e	Valor
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

a	b	c	d	e	Valor
1	0	0	0	0	
1	0	0	0	1	
1	0	0	1	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	0	1	
1	0	1	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

En las filas restantes tenemos que la expresión valdrá 1 si $\overline{b} + c\left(\overline{d + e}\right)$=1, y 0 en caso contrario. Esta subexpresión vale 1 cuando b=0 ó $c\left(\overline{d + e}\right)$=1, lo que nos permite rellenar ya con el valor 1 la mitad de las filas que nos quedan:

a	b	c	d	e	Valor
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

a	b	c	d	e	Valor
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

El valor de la expresión en las filas que faltan coincide con el de la subexpresión $c\left(\overline{d+e}\right)$, que vale 0 cuando c=0, lo que de nuevo nos permite rellenar la mitad de las filas aún vacías:

a	b	c	d	e	Valor
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

a	b	c	d	e	Valor
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

Ya para finalizar, $\overline{d+e}$ vale 1 si d y e valen 0, y 0 en caso contrario. Con esta información podemos rellenar las cuatro filas restantes:

a	b	c	d	e	Valor
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

a	b	c	d	e	Valor
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	0

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS DE OBTENCIÓN DE TABLAS DE VERDAD A PARTIR DE EXPRESIONES ALGEBRAICAS

Obtenga las tablas de verdad correspondientes a las siguientes expresiones algebraicas:

1 $a+bc+\overline{a}b$

2 $\overline{a}(\overline{c+\overline{b}})+ac$

3 $a+a\overline{bc}+\overline{\overline{a+c}b}$

4 $(\overline{a}+b+c)(a+\overline{b}+c)(a+b+\overline{c})$

5 $\overline{(b+c)(a+b)(\overline{b}+\overline{c})}$

6 $\overline{(a+b)}+bc+\overline{a}b+a\overline{b}$

7 $a+\overline{a}(b+c)(\overline{a+c})$

8 $ab+(\overline{a}+\overline{b}+c)(a+b+\overline{c})$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones a los ejercicios propuestos sobre extracción de expresiones canónicas a partir de la tabla de verdad

Ejercicio 1 – $f(a,b,c) = \prod M(4,5,6,7) = (\overline{a}+b+c)(\overline{a}+b+\overline{c})(\overline{a}+\overline{b}+c)(\overline{a}+\overline{b}+\overline{c})$

Ejercicio 2 – $f(a,b,c) = m0 = \overline{a}\overline{b}\overline{c}$

Ejercicio 3 – $f(a,b,c) = \prod M(0,2,5,6) = (a+b+c)(a+\overline{b}+c)(\overline{a}+b+\overline{c})(\overline{a}+\overline{b}+c)$

Ejercicio 4 – $f(a,b,c) = \sum m(1,2,3,5,6) = \overline{a}\overline{b}c + \overline{a}b\overline{c} + \overline{a}bc + a\overline{b}c + ab\overline{c}$

Ejercicio 5 – $f(a,b,c) = \prod M(2,3,4,5) = (a+\overline{b}+c)(a+\overline{b}+\overline{c})(\overline{a}+b+c)(\overline{a}+b+\overline{c})$

Ejercicio 6 – $f(a,b,c) = \sum m(2,3,4) = \overline{a}b\overline{c} + \overline{a}bc + a\overline{b}\overline{c}$

Soluciones a los ejercicios propuestos sobre evaluación de expresiones booleanas

Nº	Expresión	Valores de las variables	Resultado
1	$a+bc+\overline{a}b$	a=1, b=0, c=1	1
2	$\overline{a}(\overline{c+\overline{b}})+ac$	a=1, b=1, c=0	0
3	$a+a\overline{bc}+\overline{\overline{a+c}b}$	a=0, b=0, c=1	1
4	$(\overline{a}+b+c)(a+\overline{b}+c)(a+b+\overline{c})$	a=1, b=1, c=1	1
5	$\overline{(b+c)(a+b)(\overline{b}+\overline{c})}$	a=0, b=1, c=1	1
6	$\overline{(a+b)}+bc+\overline{a}b+a\overline{b}$	a=1, b=1, c=0	0
7	$a+\overline{a}(b+c)(\overline{a+c})$	a=0, b=0,c=0	0
8	$ab+(\overline{a}+\overline{b}+c)(a+b+\overline{c})$	a=1, b=0, c=0	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones a los ejercicios propuestos sobre obtención de tablas de verdad a partir de expresiones algebraicas

Expresión 1 –

a	b	c	Valor
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Expresión 2 –

a	b	c	Valor
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Expresión 3 –

a	b	c	Valor
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Expresión 4 –

a	b	c	Valor
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	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Expresión 5 –

a	b	c	Valor
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Expresión 6 –

a	b	c	Valor
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	0
1	1	1	1

Expresión 7 –

a	b	c	Valor
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Expresión 8 –

a	b	c	Valor
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

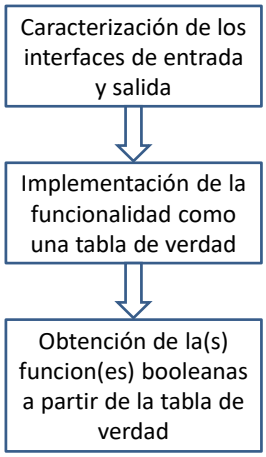
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P14

Metodología de síntesis de sistemas combinacionales

Metodología de síntesis de sistemas combinacionales



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Metodología de síntesis de sistemas combinacionales: ejemplo 1

- *Realice un multiplicador de números en binario natural. El rango de valores en las entradas del multiplicador deberá estar entre 0 y 3.*

Multiplicador: Caracterización de los interfaces

- Interfaz de entrada: para representar valores en binario natural entre el 0 y el 3 son necesarios dos bits para cada entrada:

Multiplicando	A1	A0
0	0	0
1	0	1
2	1	0
3	1	1

Multiplicador	B1	B0
0	0	0
1	0	1
2	1	0
3	1	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

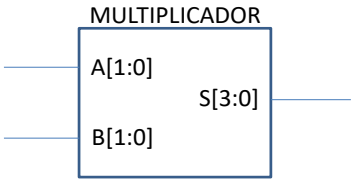
Multiplicador: Caracterización de los interfaces

- Interfaz de salida: el producto de 2 números que están en el rango de 0 a 3 está en el rango de 0 a 9. Para representar este producto en binario natural hacen falta 4 bits:

Salida	S3	S2	S1	S0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

BT1_S6 BT1_P14_5

Multiplicador: Tabla de verdad



A_1	A_0	B_1	B_0	S_3	S_2	S_1	S_0
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

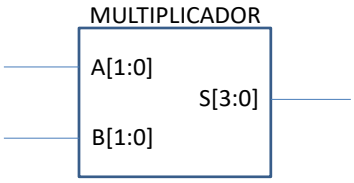
BT1_S6 BT1_P14_6

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Multiplicador: Síntesis de funciones

A_1	A_0	B_1	B_0	S_3	S_2	S_1	S_0
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



$$S_3 = m_{15} = A_1\, A_0\, B_1\, B_0$$

$$S_2 = \sum m(10,11,14) = A_1\, \overline{A_0}\, B_1\, \overline{B_0} + A_1\, A_0\, B_1\, \overline{B_0} + A_1\, \overline{A_0}\, B_1\, B_0$$

$$S_1 = \sum m(6,7,9,11,13,14) =$$

$$= \overline{A_1}\, A_0\, B_1\, \overline{B_0} + \overline{A_1}\, A_0\, B_1 B_0 + A_1 \overline{A_0}\, \overline{B_1}\, B_0 + A_1 \overline{A_0}\, B_1\, B_0 + A_1 A_0\, \overline{B_1}\, B_0 + A_1 A_0\, B_1\, \overline{B_0}$$

$$S_0 = \sum m(5,7,13,15) = \overline{A_1}\, A_0\, \overline{B_1}\, B_0 + \overline{A_1} A_0 B_1 B_0 + A_1\, A_0\, \overline{B_1}\, B_0 + A_1\, A_0\, B_1\, B_0$$

Metodología de síntesis de sistemas combinacionales: ejemplo 2

- *Realice un sistema generador de recomendaciones en función de la velocidad del viento. El sistema obtiene la información de la velocidad del viento de un instrumento capaz de realizar mediciones de velocidad entre 0 y 300 km/h con una resolución de 20 km/h. Las recomendaciones que realiza son las siguientes:*
  - *Entre 0 y 59 km/h: realice sus actividades con normalidad*
  - *Entre 60 y 99 km/h: realice sus actividades con precaución*
  - *Entre 100 y 159 km/h: realice solo los desplazamientos precisos*
  - *Entre 160 y 199 km/h: Asegure puertas y ventanas. Salga de casa únicamente en caso de urgencia*
  - *Entre 200 y 300 km/h: Asegure puertas y ventanas. No salga de casa bajo ningún concepto*

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Recomendador: Caracterización de los interfaces

- Interfaz de entrada: entre 0 y 300 en pasos de 20 existen 15 escalones, que se pueden representar en binario con 4 variables:

Velocidad del viento	v_3	v_2	v_1	v_0
0 ≤ v < 20 km/h	0	0	0	0
20 ≤ v < 40 km/h	0	0	0	1
40 ≤ v < 60 km/h	0	0	1	0
60 ≤ v < 80 km/h	0	0	1	1
80 ≤ v < 100 km/h	0	1	0	0
100 ≤ v < 120 km/h	0	1	0	1
120 ≤ v < 140 km/h	0	1	1	0
140 ≤ v < 160 km/h	0	1	1	1

Velocidad del viento	v_3	v_2	v_1	v_0
160 ≤ v < 180 km/h	1	0	0	0
180 ≤ v < 200 km/h	1	0	0	1
200 ≤ v < 220 km/h	1	0	1	0
220 ≤ v < 240 km/h	1	0	1	1
240 ≤ v < 260 km/h	1	1	0	0
260 ≤ v < 280 km/h	1	1	0	1
280 ≤ v ≤ 300 km/h	1	1	1	0

BT1_S6 BT1_P14_9

Recomendador: Caracterización de los interfaces

- Interfaz de salida: el sistema realiza 5 recomendaciones diferentes, que pueden codificarse con 3 variables

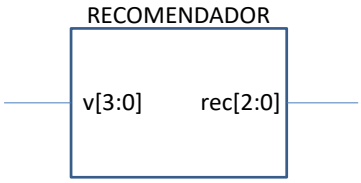
Recomendación	rec_2	rec_1	rec_0
Realice sus actividades con normalidad	0	0	0
Realice sus actividades con precaución	0	0	1
Realice solo los desplazamientos precisos	0	1	0
Asegure puertas y ventanas. Salga de casa únicamente en caso de urgencia.	0	1	1
Asegure puertas y ventanas. No salga de casa por ningún concepto	1	0	0

BT1_S6 BT1_P14_10

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Recomendador: Tabla de verdad

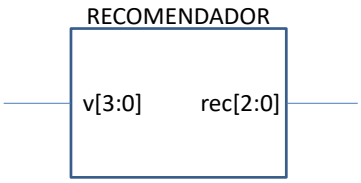


v_3	v_2	v_1	v_0	r_2	r_1	r_0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	1	1
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	0

La última fila corresponde con una velocidad del viento que no puede darse. En este caso, el valor de la función puede ser cualquiera

Recomendador: Síntesis de funciones

v_3	v_2	v_1	v_0	r_2	r_1	r_0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	1	1
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	0
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	0	0



$$r_0 = \sum m(3,4,8,9) = \overline{v_3}\,\overline{v_2}\,v_1\,v_0 + \overline{v_3}\,v_2\,\overline{v_1}\,\overline{v_0} + v_3\,\overline{v_2}\,\overline{v_1}\,\overline{v_0} + v_3\,\overline{v_2}\,\overline{v_1}\,v_0$$

$$r_1 = \sum m(5,6,7,8,9) = \overline{v_3}\,v_2\,\overline{v_1}\,v_0 + \overline{v_3}\,v_2\,v_1\,\overline{v_0} + \overline{v_3}\,v_2\,v_1\,v_0 + v_3\,\overline{v_2}\,\overline{v_1}\,\overline{v_0} + v_3\,\overline{v_2}\,\overline{v_1}\,v_0$$

$$r_2 = \sum m(10,11,12,13,14) = v_3\,\overline{v_2}\,v_1\,\overline{v_0} + v_3\,\overline{v_2}\,v_1\,v_0 + v_3\,v_2\,\overline{v_1}\,\overline{v_0} + v_3\,v_2\,\overline{v_1}\,v_0 + v_3\,v_2\,v_1\,\overline{v_0}$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios de síntesis de circuitos combinacionales simples.**	**CÓDIGO:** **BT1_AINP12**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Trabajo individual.	TIPO:	No presencial	DURACIÓN:	100 minutos

CALENDARIO:	Sexta Sesión. Previo a Séptima Sesión	REQUISITOS:	

CRITERIO DE ÉXITO:	Realización correcta de tres de los cuatro ejercicios propuestos.

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	puntos

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS DE SÍNTESIS DE SISTEMAS COMBINACIONALES SIMPLES

A continuación se presentan dos ejemplos desarrollados de síntesis de sistemas combinacionales sencillos. Léalos atentamente, y realice a continuación los dos ejercicios que se proponen tras los ejemplos.

Ejemplo 1

Obtenga una expresión lógica que corresponda a un circuito combinacional que determine si un número entre 0 y 15 presentado a su entrada es o no primo.

Siguiendo los pasos de la metodología de síntesis de sistemas combinacionales, la primera tarea es identificar las entradas y salidas del sistema. En este caso la tarea es sencilla; el sistema debe tener una entrada en la que se presente el número cuya primalidad se desea comprobar, y una salida que permita saber si el número presentado en la entrada es primo o no.

El siguiente paso es determinar la codificación de entradas y salidas. En principio, y ya que la entrada es un número natural, parece razonable codificarlo en binario natural. Necesitaremos para ello cuatro bits que llamaremos n_3, n_2, n_1 y n_0. En cuanto a la salida, únicamente es necesario distinguir entre dos condiciones mutuamente excluyentes (un número es primo o no lo es, pero no puede serlo y a la vez no serlo), por lo que es suficiente una salida de un bit a la que llamaremos *p*, y a la que arbitrariamente asignaremos el valor "1" cuando el número en la entrada sea primo, y "0" cuando no lo sea.

Una vez hecho esto el siguiente paso es obtener la tabla de verdad del sistema, que quedaría así:

n_3	n_2	n_1	n_0	p
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Ya sólo resta obtener una expresión algebraica que corresponda a esta tabla de verdad. Un posible método es acudir a las formas canónicas; en este caso la función lógica que buscamos tiene seis unos y diez ceros, por lo que es recomendable utilizar la forma normal disyuntiva o suma de minitérminos:

$$p = \sum m(2,3,5,7,11,13)$$
$$= \overline{n_3}\,\overline{n_2}\,n_1\,\overline{n_0} + \overline{n_3}\,\overline{n_2}\,n_1\,n_0 + \overline{n_3}\,n_2\,\overline{n_1}\,n_0 + \overline{n_3}\,n_2\,n_1\,n_0 + n_3\,\overline{n_2}\,n_1\,n_0 + n_3\,n_2\,\overline{n_1}\,n_0$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Otra opción menos formal es examinar visualmente la tabla de verdad para extraer una suma de productos no canónicos (es decir, que no incluyan a todas las variables de entrada y no sean por tanto minitérminos). Por ejemplo, observando las filas que corresponden a los valores de entrada 2 y 3 podemos ver que el producto $\overline{n_3}\,\overline{n_2}\,n_1$ vale 1 para esas dos combinaciones de entrada y 0 para cualquier otra; a partir de las entradas 5 y 7 podemos hacer una deducción similar para el producto $\overline{n_3}\,n_2\,n_0$; y podemos repetir el proceso usando las entradas 3 y 11 ($\overline{n_2}\,n_1\,n_0$) y las entradas 5 y 13 ($n_2\,\overline{n_1}\,n_0$). Si luego sumamos los productos así identificados obtenemos la siguiente expresión:

$$p = \overline{n_3}\,\overline{n_2}\,n_1 + \overline{n_3}\,n_2\,n_0 + \overline{n_2}\,n_1\,n_0 + n_2\,\overline{n_1}\,n_0$$

Que es totalmente equivalente a la forma canónica obtenida en primer lugar.

Ejemplo 2

Sintetice un sistema combinacional que haga de árbitro en el juego de “piedra, papel o tijera”: dos jugadores eligen simultáneamente una jugada que puede ser “piedra”, “papel” o “tijera”. Las jugadas están ordenadas circularmente, de modo que “piedra” gana a “tijera”, ésta gana a “papel” y éste a su vez a “piedra”. Si ambos jugadores eligen la misma jugada se produce un empate. El sistema debe decir cuál es el resultado a partir de las jugadas elegidas por ambos jugadores.

Siguiendo la metodología expuesta en clase, empezaremos identificando cuáles son las entradas y salidas y qué valores pueden tomar. Puesto que se debe decidir el resultado del juego a partir de las jugadas escogidas por cada jugador, éstas deben ser las entradas del sistema. Por otro lado debe existir una salida en la que se indique el resultado del juego.

Veamos ahora la codificación de entradas y salidas. En cuanto a las entradas, ambas deben indicar una de tres posibles jugadas; por tanto cada entrada debe tener al menos dos bits. Como en principio no hay ningún otro criterio, elegimos la siguiente codificación:

Jugada	Codificación
Piedra	01
Papel	10
Tijera	11

En cuanto a la salida existen tres posibles resultados: gana el primer jugador, gana el segundo o se produce un empate. De nuevo son necesarios al menos dos bits para codificar la salida, y sobra al menos una combinación de salida. Supongamos que elegimos la siguiente codificación:

Resultado	Codificación
Gana el primer jugador	10
Gana el segundo jugador	01
Empate	11

El siguiente paso es obtener la tabla de verdad del sistema. Si llamamos *a* y *b* a las entradas correspondientes al primer y segundo jugador, respectivamente, y *r* a la salida, la tabla de verdad es la que se muestra a continuación. En la tabla se ha indicado a la derecha el significado de cada código para que resulte más sencillo interpretarla. Para las combinaciones de entrada que no son válidas se ha utilizado la combinación de salida que tampoco lo es.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

a_1	a_0		b_1	b_0		r_1	r_0	
0	0		0	0		0	0	No válida
0	0		0	1	Piedra	0	0	No válida
0	0		1	0	Papel	0	0	No válida
0	0		1	1	Tijera	0	0	No válida
0	1	Piedra	0	0		0	0	No válida
0	1	Piedra	0	1	Piedra	1	1	Empate
0	1	Piedra	1	0	Papel	0	1	Gana B
0	1	Piedra	1	1	Tijera	1	0	Gana A
1	0	Papel	0	0		0	0	No válida
1	0	Papel	0	1	Piedra	1	0	Gana A
1	0	Papel	1	0	Papel	1	1	Empate
1	0	Papel	1	1	Tijera	0	1	Gana B
1	1	Tijera	0	0		0	0	No válida
1	1	Tijera	0	1	Piedra	0	1	Gana B
1	1	Tijera	1	0	Papel	1	0	Gana A
1	1	Tijera	1	1	Tijera	1	1	Empate

Nos queda extraer de la tabla de verdad unas expresiones algebraicas que permitan calcular r_1 y r_0 a partir de a_1, a_0, b_1 y b_0. Como ambas funciones tienen menos unos que ceros, emplearemos la forma canónica disyuntiva:

$$r_1 = \sum m(5,7,9,10,14,15)$$
$$= \overline{a_1}\, a_0\, \overline{b_1}\, b_0 + \overline{a_1}\, a_0\, b_1\, b_0 + a_1\, \overline{a_0}\, \overline{b_1}\, b_0 + a_1\, \overline{a_0}\, b_1\, \overline{b_0} + a_1\, a_0\, b_1\, \overline{b_0} + a_1\, a_0\, b_1\, b_0$$
$$r_0 = \sum m(5,6,10,11,13,15)$$
$$= \overline{a_1}\, a_0\, \overline{b_1}\, b_0 + \overline{a_1}\, a_0\, b_1\, \overline{b_0} + a_1\, \overline{a_0}\, b_1\, \overline{b_0} + a_1\, \overline{a_0}\, b_1\, b_0 + a_1\, a_0\, \overline{b_1}\, b_0 + a_1\, a_0\, b_1\, b_0$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicios

Ejercicio 1 – Sintetice un sistema combinacional que permita averiguar si un número entre 0 y 15 es múltiplo de 2 y mayor que 7.

Ejercicio 2 – Sintetice un sistema combinacional que examine tres bits e informe a su salida de si los tres son iguales o no y, en este último caso, indique cuál de los tres es distinto a los otros dos.

Ejercicio 3 – Realice un módulo que eleve al cuadrado números entre 0 y 4.

Ejercicio 4 – Realice un módulo que reciba como entrada, debidamente codificada, una palabra de la frase "quien lee mucho y viaja mucho ve mucho y sabe mucho" y obtenga en su salida el número de letras que tiene esa palabra.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones a los ejercicios propuestos

A continuación se ofrece una solución para cada uno de los ejercicios propuestos en el apartado anterior. Fíjese en que la solución nunca es única, sino que depende de la codificación elegida para entradas y salidas; en algunas ocasiones la elección de la codificación es casi automática (por ejemplo, cuando los datos a codificar son números naturales entre cero y un máximo), pero en otros puede haber una amplia variedad de posibilidades razonables.

Ejercicio 1

En este caso la entrada es un número natural entre 0 y 15, por lo que se codificará en binario natural con 4 bits. La salida debe indicar si se produce o no una condición, por lo que se codificará con un bit que valdrá 1 cuando se cumpla la condición indicada en el enunciado, y 0 cuando no se cumpla. Llamando *N* a la entrada y *CUMPLE* a la salida, la tabla de verdad queda así:

N_3	N_2	N_1	N_0	*CUMPLE*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

La función *CUMPLE* se puede expresar como suma de minitérminos de la siguiente manera:

$$CUMPLE = N_3\overline{N_2}\,\overline{N_1}\,\overline{N_0} + N_3\overline{N_2}N_1\overline{N_0} + N_3N_2\overline{N_1}\,\overline{N_0} + N_3N_2N_1\overline{N_0} = \sum m(8,10,12,14)$$

Simplificando o por inspección de la tabla de verdad se puede llegar a la siguiente expresión simplificada:

$$CUMPLE = N_3\overline{N_0}$$

Ejercicio 2

El enunciado establece claramente que la entrada está formada por tres bits, a los que llamaremos *A*, *B* y *C*. Para elegir la codificación de la salida hay que ver cuántas posibilidades existen. Representaremos para ello la tabla de verdad del circuito omitiendo por el momento la salida:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

A	*B*	*C*
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Vemos que o bien las tres entradas son iguales, o bien una de ellas es distinta a las otras dos. Como hay tres entradas, este segundo caso comprende tres posibilidades (*A* distinto de *B* y *C* -011 y 100-, *B* distinto de *A* y *C* -010 y 101-, y *C* distinto de *B* y *A* -001 y 110-). Hay que distinguir por tanto entre cuatro posibilidades, por lo que la salida debe tener al menos dos bits. Elegiremos la siguiente codificación para la salida, a la que llamaremos *S*:

Caso	S_1	S_0
Las tres entradas iguales	0	0
A distinto de *B* y *C*	0	1
B distinto de *A* y *C*	1	0
C distinto de *A* y *B*	1	1

La tabla de verdad queda como sigue:

A	*B*	*C*	S_1	S_0
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Y de ella se pueden extraer las siguientes funciones:

$$S_1 = \sum m(1,2,5,6) = \overline{B}\,C + B\,\overline{C}$$
$$S_0 = \sum m(1,3,4,6) = \overline{A}\,C + A\,\overline{C}$$

Ejercicio 3

Tanto la entrada como la salida son números naturales cuyo límite inferior es 0, por lo que se codificarán en binario natural. Como la entrada puede valer entre 0 y 4 serán necesarios 3 bits para representarla, y la salida puede tomar los valores 0, 1, 4, 9 y 16, por lo que son necesarios 5 bits. Llamando *E* a la entrada y *S* a la salida la tabla de verdad queda como sigue:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

E_2	E_1	E_0	S_4	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	1	0	0	0	1	0	0
0	1	1	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Fíjese en que hay tres combinaciones de entrada que en principio no podrían aparecer, y para ellas se ha escogido un valor cero para todas las salidas. De la tabla se pueden extraer las siguientes expresiones:

$$
\begin{aligned}
S_4 &= E_2\overline{E_1}\,\overline{E_0} \\
S_3 &= \overline{E_2}E_1E_0 \\
S_2 &= \overline{E_2}E_1\overline{E_0} \\
S_1 &= 0 \\
S_0 &= \sum m(1,3) = \overline{E_2}E_0
\end{aligned}
$$

Otra posibilidad sería no codificar la salida en binario natural, sino asignar una combinación de unos y ceros a cada uno de los posibles valores de salida; como sólo hay cinco posibles, sería suficiente con utilizar tres bits. El escoger una u otra posibilidad depende del contexto en el que se desee utilizar este circuito más que de cualquier otro factor, por lo que con los datos de que se dispone no es posible decidir cuál es la opción más adecuada.

Ejercicio 4

La codificación de la salida en este ejercicio es sencilla, ya que es un número natural entre 1 y 5. Se codificará en binario natural con 3 bits, y la llamaremos *S*. La codificación de la entrada, a la que llamaremos *E*, requiere algo más de atención, aunque tampoco es complicada. En primer lugar es necesario ver cuántos datos distintos puede haber en la misma; en este caso, hay que ver cuántas palabras distintas tiene la frase indicada en el enunciado. Si escribimos la frase y tachamos las palabras que aparezcan repetidas podremos contarlas fácilmente: "Quien lee mucho y viaja ~~mucho~~ ve ~~mucho~~ ~~y~~ sabe ~~mucho~~". En total hay 7 palabras distintas, que se pueden codificar con tres bits como sigue:

Palabra	E_2	E_1	E_0
quien	0	0	0
lee	0	0	1
mucho	0	1	0
y	0	1	1
viaja	1	0	0
ve	1	0	1
sabe	1	1	0

La tabla de verdad del circuito queda por tanto como sigue:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

E_2	E_1	E_0	S_2	S_1	S_0
0	0	0	1	0	1
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0

De esta tabla pueden derivarse las siguientes ecuaciones:

$$S_2 = \sum m(0,2,4,6) = \overline{E_0}$$
$$S_1 = \sum m(1,5) = \overline{E_1}E_0$$
$$S_0 = \sum m(0,1,2,3,4) = \overline{E_2} + \overline{E_1}\,\overline{E_0}$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre síntesis de sistemas combinacionales simples.**	**CÓDIGO:** **BT1_ATGP2**

GRUPO		FECHA:	

MODALIDAD:	Aprendizaje Cooperativo	TIPO:	Presencial	DURACIÓN:	40 minutos
CALENDARIO:	Séptima Sesión	REQUISITOS:	Metodología de síntesis de circuitos combinacionales		

CRITERIO DE ÉXITO:	Al menos tres ejercicios bien resueltos.
INTERDEPENDENCIA POSITIVA:	En el tiempo disponible no pueden completar los cuatro ejercicios si no trabajan durante cierto tiempo en paralelo.
EXIGIBILIDAD INDIVIDUAL:	La puntuación del grupo tiene una componente relacionada con la realización de un ejercicio de manera individual por un miembro del grupo elegido al azar.
ROLES:	**Secretario**: toma nota de las decisiones alcanzadas y rellena las hojas de resultados. **Controlador de Tiempo**: controla el tiempo que queda para la finalización de la tarea. **Moderador**: dirige la discusión, controla los turnos de palabra y garantiza la participación por igual de todos los miembros.
HABILIDADES SOCIALES:	Capacidad para explicar al resto del equipo la solución que cada miembro individual ha realizado de su ejercicio.
REFLEXIÓN DE GRUPO:	Evaluación de los resultados obtenidos. ¿Hemos trabajado todos por igual?.¿Hemos tenido problemas para entendernos?. ¿Se ha obtenido el mejor rendimiento del que el grupo es capaz?

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE I. Descripción de la Actividad

a) Lectura conjunta de los enunciados de los cuatro ejercicios que se plantean para que quede claro que se pide en cada uno de ellos. Discusión sobre la interpretación de los enunciados. (15 minutos).
b) Cada miembro del grupo se queda con un ejercicio. Pensará su solución y la pondrá por escrito en una hoja. (10 minutos).
c) Puesta en común de los resultados alcanzados por cada miembro del grupo y análisis de la coherencia de las cuatro soluciones propuestas. Solución de discordancias y propuesta de una solución consensuada. (15 minutos).
d) Realización de un test de final de actividad por uno de los miembros del grupo elegido al azar. La calificación obtenida será aplicable al resto de miembros del grupo. (10 minutos).

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE II. Enunciados de los Ejercicios

EJERCICIO 1) Paridad.

Realice un módulo que detecte el número de unos de una palabra de 4 bits y genere a su salida un 1 si éste es par (paridad par) y un 0 si es impar (paridad impar). Si el número de unos es cero se considerará paridad par.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 2) Desplazador

Realice un módulo que reciba como entrada una palabra de 3 bits y obtenga a su salida, también en 3 bits, la misma palabra desplazada un bit a la derecha (metiendo un cero por la izquierda) o a la izquierda (introduciendo un cero por la derecha), en función del estado de una entrada de control. El desplazamiento será a la izquierda si la entrada de control es 0 y a la derecha si es 1. En los desplazamientos a la izquierda los datos que ocupen el MSB se pierden. Lo mismo ocurre con los datos que ocupen el LSB en los desplazamientos a la derecha.

- Si control = 0 y entrada = “e3 e2 e1”; entonces salida = “e2 e1 0”
- Si control = 1 y entrada = “e3 e2 e1”; entonces salida = “0 e3 e2”

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 3) Tiempo de espera

Realice un módulo que reciba como entrada el número de aviones en espera de despegue (hasta 7) e informe del tiempo medio de espera para que despegue el último de la fila, en minutos. Suponga que, una vez en espera de despegue, un avión tarda un tiempo medio de 2 minutos en despegar.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 4) Avisador velocidad

Realice un módulo que reciba como entrada el tipo de carretera por la que circula un vehículo (autopista, autovía, nacional, comarcal, local o camino) y genere a partir de este dato la información sobre la velocidad máxima que puede alcanzar el vehículo en esa vía (120 en autopista y autovía, 110 en nacional, 90 en comarcal y local y 60 Km/h en camino). Utilice un esquema de codificación que reduzca al mínimo el número de bits, tanto a la entrada como a la salida.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1	
TÍTULO DE LA ACTIVIDAD: **Test final sobre síntesis de sistemas combinacionales simples.**	**CÓDIGO:** **BTI_ATGP3**

FECHA:	

NOMBRE:		APELLIDOS:	

GRUPO		CALIFICACIÓN: [entre 0 y 10 puntos]	puntos

EJERCICIO. Incrementador

Realice un módulo incrementador (la salida es la entrada más uno) para números en complemento a 2 de 3 bits. La salida debe dimensionarse de forma que no exista la posibilidad de desbordamiento.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica II
Bloque temático I

BT1.P15

Teoremas básicos del Algebra de Boole

Álgebra de Boole. Teoremas básicos

Teoremas básicos del Álgebra de Boole		
Principio de dualidad. Dado un teorema puede obtenerse el dual intercambiando las operaciones '+' y '·' y los elementos neutros.		
Teorema de idempotencia	$a+a=a$	$a\cdot a=a$
Teorema de absorción	$a+a\cdot b=a$	$a\cdot(a+b)=a$
Teorema de adyacencia	$a\cdot b+a\cdot\overline{b}=a$	$(a+b)\cdot(a+\overline{b})=a$
Teorema de simplificación	$a+\overline{a}\cdot b=a+b$	$a\cdot(\overline{a}+b)=a\cdot b$
Teorema de De Morgan	$\overline{a_o\cdot a_1\cdot\ldots\cdot a_n}=\overline{a_o}+\overline{a_1}+\ldots+\overline{a_n}$	$\overline{a_o+a_1+\ldots+a_n}=\overline{a_o}\cdot\overline{a_1}\cdot\ldots\cdot\overline{a_n}$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Álgebra de Boole. Simplificación

Teorema de absorción	$a + a \cdot b = a$	$a \cdot (a + b) = a$

$$\bar{a}b\bar{c}d + \bar{a}d = \bar{a}db\bar{c} + \bar{a}d = \bar{a}d$$

$$(a + b + \bar{c} + \bar{d})(b + \bar{c}) = (b + \bar{c})(b + \bar{c} + a + \bar{d}) = (b + \bar{c})$$

Álgebra de Boole. Simplificación

Teorema de simplificación	$a + \bar{a} \cdot b = a + b$	$a \cdot (\bar{a} + b) = a \cdot b$

$$a + \bar{a}b\bar{c}d = a + b\bar{c}d$$

$$\overline{ab} + ab(c + d) = \overline{ab} + c + d$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Álgebra de Boole. Simplificación

Teorema de adyacencia	$a \cdot b + a \cdot \bar{b} = a$	$(a+b)\cdot(a+\bar{b}) = a$

$$\bar{a}b\bar{c}d + \bar{a}bcd = \bar{a}bd$$

$$(a+b+\bar{c}+\bar{d})(a+b+\bar{c}+d) = a+b+\bar{c}$$

Álgebra de Boole. Simplificación

Teorema de idempotencia	$a + a = a$	$a \cdot a = a$

Absorción
($a+ab=a$)

$$\bar{a}+\bar{a}bd+\bar{a}\bar{c} = \bar{a}+\bar{a}+\bar{a}bd+\bar{a}\bar{c} = \bar{a}+\bar{a} = \bar{a}$$

Simplificación
($a+\bar{a}b=a+b$)

$$a+\bar{a}b+\bar{a}cd = a+a+\bar{a}b+\bar{a}cd = a+b+a+cd = a+b+cd$$

Adyacencia
($ab+a\bar{b}=a$)

$$a\bar{b}c+a\bar{b}\bar{c}+abc = a\bar{b}c+a\bar{b}c+a\bar{b}\bar{c}+abc = a\bar{b}+ac$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios de simplificación de expresiones algebraicas.**	**CÓDIGO:** **BT1_AINP15**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Trabajo individual.	TIPO:	No presencial	DURACIÓN:	30 minutos

CALENDARIO:	Séptima Semana. Previo a Octava Sesión.	REQUISITOS:	

CRITERIO DE ÉXITO:	Realización correcta de cinco de los siete ejercicios propuestos.

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	puntos

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIOS DE SIMPLIFICACIÓN DE EXPRESIONES ALGEBRAICAS

Ejercicio 1 – Compruebe mediante manipulaciones algebraicas que $ab = (a+b)\left(a+\overline{b}\right)(\overline{a}+b)$.

Ejercicio 2 – Compruebe mediante manipulaciones algebraicas (y sin aplicar el principio de dualidad) que $a+b = a\overline{b}+\overline{a}b+ab$.

Ejercicio 3 – Simplifique mediante manipulaciones algebraicas la expresión $(a+b+c+d)\left(a+b+c+\overline{d}\right)(a+b+\overline{c}+d)(a+b+\overline{c}+\overline{d})$.

Ejercicio 4 – Obtenga mediante manipulaciones algebraicas la expresión más sencilla para la función $f(a,b,c) = \sum m(1,4,5)$.

Ejercicio 5 – Obtenga mediante manipulaciones algebraicas la expresión más sencilla para la función $f(a,b,c,d) = \overline{\overline{ab}+\overline{c}}+ab\overline{c}+abcd$.

Ejercicio 6 – Obtenga mediante manipulaciones algebraicas la expresión más sencilla para la función $f(a,b) = \overline{b\,(\,\overline{a}+b\,)}$.

Ejercicio 7 – Compruebe mediante manipulaciones algebraicas que $a\overline{b}c+abd+a\overline{c}d = a\overline{b}c+ad$.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones a los ejercicios propuestos

Ejercicio 1 – Compruebe mediante manipulaciones algebraicas que $ab = (a+b)\left(a+\overline{b}\right)(\overline{a}+b)$.

$$\begin{aligned} ab &= (a+b)\left(a+\overline{b}\right)(\overline{a}+b) \\ &= (a+b)\left(a+\overline{b}\right)(a+b)(\overline{a}+b) && \text{Aplicando idempotencia para duplicar } (a+b) \\ &= ab && \text{Aplicando adyacencia dos veces} \end{aligned}$$

Ejercicio 2 – Compruebe mediante manipulaciones algebraicas (y sin aplicar el principio de dualidad) que $a+b = a\overline{b}+\overline{a}b+ab$.

$$\begin{aligned} a+b &= a\overline{b}+\overline{a}b+ab \\ &= a\overline{b}+ab+\overline{a}b+ab && \text{Aplicando idempotencia para duplicar } ab \\ &= a+b && \text{Aplicando adyacencia dos veces} \end{aligned}$$

Ejercicio 3 – Simplifique mediante manipulaciones algebraicas la expresión $(a+b+c+d)\left(a+b+c+\overline{d}\right)(a+b+\overline{c}+d)(a+b+\overline{c}+\overline{d})$.

$$\begin{aligned} &(a+b+c+d)\left(a+b+c+\overline{d}\right)(a+b+\overline{c}+d)(a+b+\overline{c}+\overline{d}) \\ &= (a+b+c)(a+b+\overline{c}) && \text{Aplicando adyacencia dos veces} \\ &= a+b && \text{Aplicando adyacencia una vez más} \end{aligned}$$

Ejercicio 4 – Obtenga mediante manipulaciones algebraicas la expresión más sencilla para la función $f(a,b,c) = \sum m(1,4,5)$.

$$\begin{aligned} \sum m(1,4,5) &= \overline{a}\overline{b}c+a\overline{b}\overline{c}+a\overline{b}c \\ &= \overline{a}\overline{b}c+a\overline{b}c+a\overline{b}\overline{c}+a\overline{b}c && \text{Aplicando idempotencia para duplicar } a\overline{b}c \\ &= \overline{b}c+a\overline{b} && \text{Aplicando adyacencia dos veces} \\ &= \overline{b}(a+c) \end{aligned}$$

Ejercicio 5 – Obtenga mediante manipulaciones algebraicas la expresión más sencilla para la función $f(a,b,c,d) = \overline{\overline{ab}+\overline{c}}+ab\overline{c}+abcd$.

$$\begin{aligned} \overline{\overline{ab}+\overline{c}}+ab\overline{c}+abcd &= abc+ab\overline{c}+abcd && \text{Aplicando De Morgan al primer término} \\ &= ab+abcd && \text{Aplicando adyacencia} \\ &= ab && \text{Aplicando absorción} \end{aligned}$$

Ejercicio 6 – Obtenga mediante manipulaciones algebraicas la expresión más sencilla para la función $f(a,b) = \overline{b\,(\overline{a}+b\,)}$.

$$\begin{aligned} \overline{b\,(\overline{a}+b\,)} &= \overline{b}+\overline{(\overline{a}+b\,)} && \text{Aplicando De Morgan} \\ &= \overline{b}+a\overline{b} && \text{Aplicando De Morgan al segundo término} \\ &= \overline{b} && \text{Aplicando absorción} \end{aligned}$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 7 – Compruebe mediante manipulaciones algebraicas que $a\overline{b}c + abd + a\overline{c}d = a\overline{b}c + ad$.

$$
\begin{aligned}
a\overline{b}c + abd + a\overline{c}d &= a\left(\overline{b}c + bd + \overline{c}d\right) \\
&= a\left(\overline{b}c + d(b + \overline{c})\right) \\
&= a\left(\overline{b}c + d\left(\overline{\overline{b}c}\right)\right) && \text{Aplicando De Morgan a } (b + \overline{c}) \\
&= a\left(\overline{b}c + d\right) && \text{Aplicando simplificación} \\
&= a\overline{b}c + ad
\end{aligned}
$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica 2
Bloque Temático 1

Sistemas combinacionales con indiferencias. Extracción de expresiones algebraicas. Tablas de verdad resumidas. Límite de la metodología de síntesis de sistemas combinacionales mediante tabla.

Tablas de verdad con indiferencias

- Son tablas de verdad en las que el valor de la función para algunas de las filas es indiferente (puede ser cualquiera, 0 ó 1).
- Esto es algo con lo que ya nos hemos encontrado en alguna ocasión: Recomendador, exponenciador, cuentaletras, avisador de velocidad, quiniela. En todos estos sistemas hay combinaciones de las entradas que no tienen sentido y para las cuales nos es indiferente el comportamiento del sistema en sus salidas.

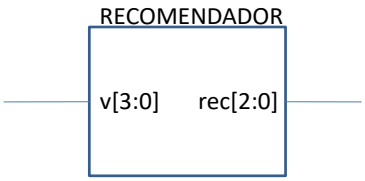
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejemplo: Recomendador



v_3	v_2	v_1	v_0	r_2	r_1	r_0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	1	1
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	X	X	X
1	1	1	1	X	X	X

En el ejemplo del recomendador se eligió arbitrariamente el 0 para todos los valores. La condición de indiferencia se representa con una X

Las dos últimas filas corresponden con velocidades del viento que no pueden darse. En estos casos, los valores de las funciones pueden ser cualesquiera

BT1_S8 BT1_P16_3

Tablas de verdad con indiferencias

- Realmente representan un conjunto de posibles funciones.

a	b	f
0	0	X
0	1	0
1	0	1
1	1	1

a	b	f
0	0	0
0	1	0
1	0	1
1	1	1

$\longrightarrow f = a$

a	b	f
0	0	1
0	1	0
1	0	1
1	1	1

$\longrightarrow f = a + \bar{b}$

BT1_S8 BT1_P16_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad con indiferencias

- Una tabla de verdad de 2 variables puede representar 16 funciones diferentes:

a	b	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$f_0 = 0$ $\quad f_4 = \bar{a}b$ $\quad f_8 = \bar{a}\bar{b}$ $\quad f_{12} = \bar{a}$

$f_1 = ab$ $\quad f_5 = b$ $\quad f_9 = ab + \bar{a}\bar{b}$ $\quad f_{13} = \bar{a} + b$

$f_2 = a\bar{b}$ $\quad f_6 = a\bar{b} + \bar{a}b$ $\quad f_{10} = \bar{b}$ $\quad f_{14} = \bar{a} + \bar{b}$

$f_3 = a$ $\quad f_7 = a + b$ $\quad f_{11} = a + \bar{b}$ $\quad f_{15} = 1$

Tablas de verdad con indiferencias

- Con N variables la tabla tiene 2^N filas …
- … y puede representar 2^{2^N} funciones diferentes.

Número de variables	Número de filas	Número de posibles funciones
2	4	16
3	8	256
4	16	65536
5	32	4.294.967.296

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad con indiferencias

- Extracción de funciones:

a	b	f
0	0	X
0	1	0
1	0	1
1	1	1

a	b	f
0	0	1
0	1	0
1	0	1
1	1	1

$$f = a + \bar{b}$$

La opción más simple: minimizar el número de términos de la función.

Esta opción no siempre da lugar a la expresión más simple.

a	b	f
0	0	0
0	1	0
1	0	1
1	1	1

$$f = a$$

Tablas de verdad con indiferencias

- Extracción de funciones. Ejemplo.

a	b	c	f	f_0	f_1	f_2	f_3
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	0	X	0	0	1	[illegible]
0	1	1	1	1	1	1	[illegible]
1	0	0	0	0	0	0	[illegible]
1	0	1	X	0	1	0	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

minimiza el número de términos de la función.

$$f_0 = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}bc = \bar{a}(\bar{b}+c)$$

$$f_1 = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}bc + a\bar{b}c = \bar{a}(\bar{b}+c)$$

$$f_2 = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}bc = \bar{a}$$

$$f_3 = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}c = \bar{a} + \bar{b}c$$

opción más simple (podría haberse localizado en la tabla a simple vista).

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad con indiferencias

- Extracción de funciones. Ejemplo.

e_3	e_2	e_1	e_0	s
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	X
0	1	0	0	X
0	1	0	1	0
0	1	1	0	X
0	1	1	1	X
1	0	0	0	X
1	0	0	1	0
1	0	1	0	X
1	0	1	1	0
1	1	0	0	X
1	1	0	1	X
1	1	1	0	1
1	1	1	1	X

Esta tabla representa 512 funciones diferentes (¿por qué?).

Tablas de verdad con indiferencias

- Solución 1: todas las indiferencias a 0 (mínimo número de términos).

e_3	e_2	e_1	e_0	s
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

$$s = \bar{e}_3\,\bar{e}_2\,\bar{e}_1\,\bar{e}_0 + \bar{e}_3\,\bar{e}_2\,e_1\,\bar{e}_0 + e_3\,e_2\,e_1\,\bar{e}_0$$
$$= \bar{e}_3\,\bar{e}_2\,\bar{e}_0 + e_3\,e_2\,e_1\,\bar{e}_0$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad con indiferencias

- Solución 2: a cada indiferencia se le asigna el valor complementario de la variable e_0.

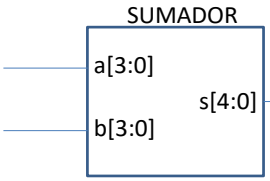
e_3	e_2	e_1	e_0	s
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

$$s = \overline{e_0}$$

BT1_S8 BT1_P16_11

Limitaciones de la metodología

- Ejemplo 1: *realice un sistema capaz de sumar dos números de 4 bits codificados en binario natural.*



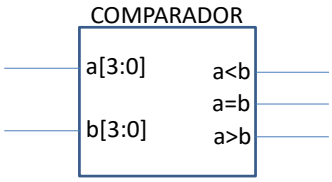
b_3	b_2	b_1	b_0	a_3	a_2	a_1	a_0	s_4	s_3	s_2	s_1	s_0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	1	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	1	0	1	0	0	1	0	1
0	0	0	0	0	1	1	0	0	0	1	1	0
0	0	0	0	0	1	1	1	0	0	1	1	1
0	0	0	0	1	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	1	0	1	0	0	1
0	0	0	0	1	0	1	0	0	1	0	1	0
0	0	0	0	1	0	1	1	0	1	0	1	1
0	0	0	0	1	1	0	0	0	1	1	0	0
0	0	0	0	1	1	0	1	0	1	1	0	1
0	0	0	0	1	1	1	0	0	1	1	1	0
0	0	0	0	1	1	1	1	0	1	1	1	1
0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0	0	0	1	0

BT1_S8 BT1_P16_12

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Limitaciones de la metodología

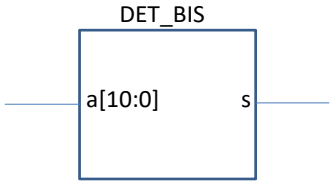
- Ejemplo 2: *realice un sistema capaz de comparar dos números de 4 bits codificados en binario natural y generar 3 salidas: a>b, a<b y a=b.*



b_3	b_2	b_1	b_0	a_3	a_2	a_1	a_0	a<b	a=b	a>b
0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	1	0	0	1
0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	1
0	0	0	0	0	1	1	0	0	0	1
0	0	0	0	0	1	1	1	0	0	1
0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	1	0	0	1	0	0	1
0	0	0	0	1	0	1	0	0	0	1
0	0	0	0	1	0	1	1	0	0	1
0	0	0	0	1	1	0	0	0	0	1
0	0	0	0	1	1	0	1	0	0	1
0	0	0	0	1	1	1	0	0	0	1
0	0	0	0	1	1	1	1	0	0	1
0	0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	0	1	0	1	0

Limitaciones de la metodología

- Ejemplo 3: *realice un sistema capaz de detectar si un año es o no bisiesto, desde el año 1 hasta la actualidad.*



- La tabla de verdad tendría 2048 entradas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad resumidas

- Son variantes en la forma de expresar las tablas de verdad que permiten definir funciones de un gran número de variables.

Tablas de verdad resumidas

- Funciones con un número pequeño de minitérminos:

$$s = e_7\, e_6\, \bar{e}_5\, e_4\, \bar{e}_3\, \bar{e}_2\, \bar{e}_1\, \bar{e}_0 + e_7 e_6\, \bar{e}_5\, e_4\, \bar{e}_3\, \bar{e}_2\, e_1\, \bar{e}_0 + e_7\, \bar{e}_6\, e_5\, e_4 e_3\, e_2\, e_1\, \bar{e}_0$$

e_7	e_6	e_5	e_4	e_3	e_2	e_1	e_0	s
1	1	0	1	0	0	0	0	1
1	1	0	1	0	0	1	0	1
1	0	1	1	1	1	1	0	1
Otras combinaciones								0

Simplemente aparecen solo las filas para las cuales la función vale 1.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad resumidas

- Funciones con un número pequeño de maxitérminos:

$$s = (e_4 + \bar{e}_3 + \bar{e}_2 + \bar{e}_1 + \bar{e}_0)(\bar{e}_4 + \bar{e}_3 + \bar{e}_2 + \bar{e}_1 + \bar{e}_0)$$
$$(e_4 + e_3 + e_2 + \bar{e}_1 + \bar{e}_0)(e_4 + \bar{e}_3 + \bar{e}_2 + e_1 + e_0)$$

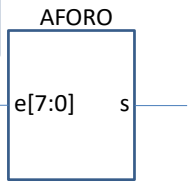
Simplemente aparecen solo las filas para las cuales la función vale 0.

e_4	e_3	e_2	e_1	e_0	s
0	1	1	1	1	0
1	1	1	1	1	0
0	0	0	1	1	0
0	1	1	0	0	0
Otras combinaciones					1

Tablas de verdad resumidas

- Ejemplo 1: *Suponga que un local tiene un aforo de 200 personas y que se dispone de un sistema que es capaz de llevar la cuenta del número de personas que entran. Realice un sistema que detecte cuándo se alcanza el aforo del local.*

Con este número de entradas la tabla de verdad completa tiene 256 filas.



e_7	e_6	e_5	e_4	e_3	e_2	e_1	e_0	s
1	1	0	0	1	0	0	0	1
Otras combinaciones								0

$$s = e_7\, e_6\, \bar{e}_5\, \bar{e}_4\, e_3\, \bar{e}_2\, \bar{e}_1\, \bar{e}_0$$

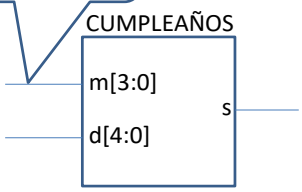
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad resumidas

- Ejemplo 2: *Realice un sistema que detecte cuándo es el cumpleaños de cada miembro de una familia de 3. Las fechas de los cumpleaños son: 22-01, 11-05 y 3-09.*

Con este número de entradas la tabla de verdad completa tiene 512 filas.



d_4	d_3	d_2	d_1	d_0	m_3	m_2	m_1	m_0	s
1	0	1	1	0	0	0	0	1	1
0	1	0	1	1	0	1	0	1	1
0	0	0	1	1	1	0	0	1	1
Otras combinaciones									0

$$s = d_4\,\overline{d_3}\,d_2\,d_1\,\overline{d_0}\,\overline{m_3}\,\overline{m_2}\,\overline{m_1}\,m_0 + \overline{d_4}\,d_3\,\overline{d_2}\,d_1\,d_0\,\overline{m_3}\,m_2\,\overline{m_1}\,m_0$$
$$+\,\overline{d_4}\,\overline{d_3}\,\overline{d_2}\,d_1\,d_0\,m_3\,\overline{m_2}\,\overline{m_1}\,m_0$$

Tablas de verdad resumidas

- Agrupación de términos:

a	b	c	d	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

a	b	c	d	f
0	0	X	X	1
0	1	0	X	0
0	1	1	0	0
0	1	1	1	1
1	0	X	X	0
1	1	0	X	0
1	1	1	X	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad resumidas

- Ejemplo: *Un consejo de administración está compuesto por accionistas de los tipos A, B y C. Cada accionista puede votar solo SI o NO. El resultado de la votación es SI siempre que voten SI:*
  - *el accionista tipo A, o bien,*
  - *los dos accionistas tipo B y al menos cuatro accionistas tipo C o,*
  - *un accionista tipo B y los cinco accionistas tipo C.*

  En los otros casos, el resultado es NO. Realice un sistema que genere el resultado de las votaciones.

A	B_1	B_0	C_4	C_3	C_2	C_1	C_0	s
1	X	X	X	X	X	X	X	1
X	1	1	1	1	1	1	X	1
X	1	1	1	1	1	X	1	1
X	1	1	1	1	X	1	1	1
X	1	1	1	X	1	1	1	1
X	1	1	X	1	1	1	1	1
X	1	X	1	1	1	1	1	1
X	X	1	1	1	1	1	1	1
Otras combinaciones								0

BT1_S8 BT1_P16_21

Tablas de verdad resumidas

- Extracción de las expresiones:

a	b	c	d	f
0	0	X	X	1
0	1	0	X	0
0	1	1	0	0
0	1	1	1	1
1	0	X	X	0
1	1	0	X	0
1	1	1	X	1

Para ambas expresiones se procede como en las tablas de verdad completas, pero sin tener en cuenta las variables que están a X.

$$f = \bar{a}\bar{b} + \bar{a}bcd + abc$$

$$f = (a+\bar{b}+c)(a+\bar{b}+\bar{c}+d)(\bar{a}+b)(\bar{a}+\bar{b}+c)$$

BT1_S8 BT1_P16_22

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad resumidas

- Extracción de las expresiones:

A	B_1	B_0	C_4	C_3	C_2	C_1	C_0	s
1	X	X	X	X	X	X	X	1
X	1	1	1	1	1	1	X	1
X	1	1	1	1	1	X	1	1
X	1	1	1	1	X	1	1	1
X	1	1	1	X	1	1	1	1
X	1	1	X	1	1	1	1	1
X	1	X	1	1	1	1	1	1
X	X	1	1	1	1	1	1	1
Otras combinaciones								0

$$s = A + B_1 B_0 C_4 C_3 C_2 C_1 + B_1 B_0 C_4 C_3 C_1 C_0 + B_1 B_0 C_4 C_2 C_1 C_0$$
$$+ B_1 B_0 C_3 C_2 C_1 C_0 + B_1 C_4 C_3 C_2 C_1 C_0 + B_0 C_4 C_3 C_2 C_1 C_0$$

Tablas de verdad resumidas

- El valor de una función puede ser una variable o su negada (y, en general, una expresión).

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

a	f
0	c
1	b

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Tablas de verdad resumidas

- Obtención de la expresión booleana.

a	f
0	c
1	b

Suma de productos: cada fila constituye un producto que se obtiene multiplicando el minitérmino por el valor de la función.

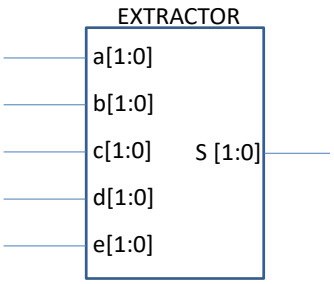
$$f = \bar{a}c + ab$$

$$f = (a+c)(\bar{a}+b)$$

Producto de sumas: cada fila constituye una suma que se obtiene sumando el maxitérmino con el valor de la función.

Tablas de verdad resumidas

- Ejemplo: *Extractor de campos. Una trama de bits tiene 5 campos de 2 bits cada uno, llamados A, B, C, D y E. El primer campo (A) identifica, con un código binario, el campo que debe ser extraído, que necesariamente ha de ser uno de los otros 4. Diseñe el circuito extractor de campos.*



a_1	a_0	s_1	s_0
0	0	b_1	b_0
0	1	c_1	c_0
1	0	d_1	d_0
1	1	e_1	e_0

$$s_0 = \bar{a}_1 \bar{a}_0 \, b_0 + \bar{a}_1 a_0 \, c_0 + a_1 \bar{a}_0 \, d_0 + a_1 a_0 \, e_0$$

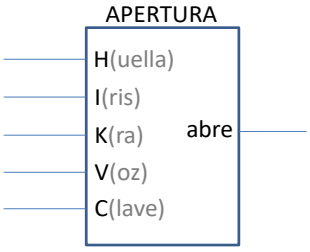
$$s_1 = \bar{a}_1 \bar{a}_0 \, b_1 + \bar{a}_1 a_0 \, c_1 + a_1 \bar{a}_0 \, d_1 + a_1 a_0 \, e_1$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

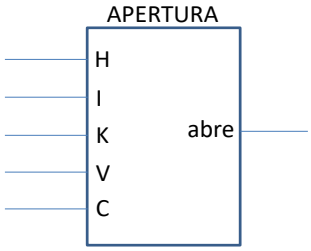
Ejercicio

- *Un sistema de apertura de puertas basado en reconocimiento contiene un detector de huella dactilar, un detector de iris, un reconocedor facial, un reconocedor de voz y un teclado para la introducción de una clave. Para realizar la apertura el sistema requiere siempre la introducción correcta de la clave y, además, un reconocimiento positivo del detector de huella o del de iris o, si no, el reconocimiento de la voz y de la cara. Obtenga la tabla de verdad del sistema y las ecuaciones booleanas.*



clave	iris	huella	voz	cara	
C	I	H	V	K	abre

Solución



clave	iris	huella	voz	cara	
C	I	H	V	K	abre
1	1	X	X	X	1
1	X	1	X	X	1
1	X	X	1	1	1
		Resto			0

$$abre = CI + CH + CVK = C\left(I + H + VK\right)$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

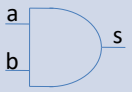
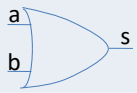
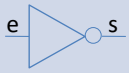
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica 2
Bloque Temático 1

Puertas Lógicas

Puertas lógicas

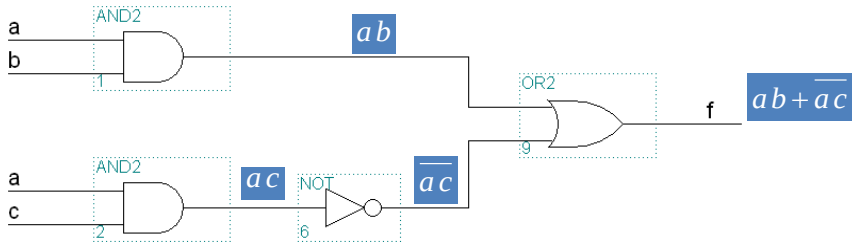
- *Las puertas lógicas simbolizan circuitos que realizan operaciones booleanas básicas y permiten representar gráficamente sistemas combinacionales.*

símbolo	puerta	Función lógica
	AND	$s = a \cdot b$
	OR	$s = a + b$
	INVERSOR	$s = \bar{e}$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Puertas lógicas

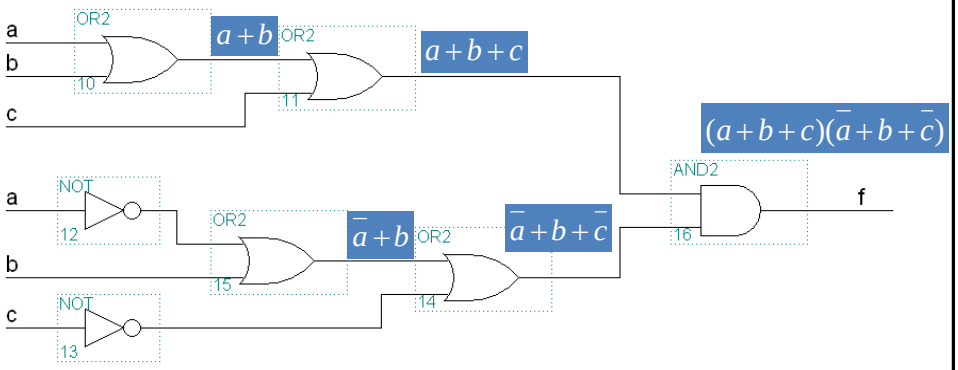
- *Ejemplo:* $f = a\,b + \overline{a\,c}$



BT1_S8 BT1_P17_3

Puertas lógicas

- *Ejemplo:* $f = (a + b + c)(\overline{a} + b + \overline{c})$



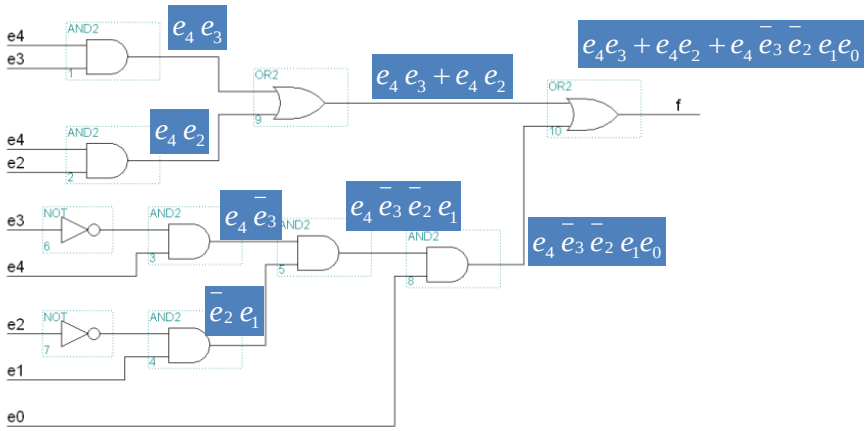
BT1_S8 BT1_P17_4

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

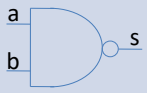
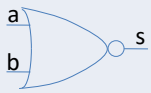
Puertas lógicas

- *Ejemplo:* $f = e_4\ e_3 + e_4\ e_2 + e_4\ \bar{e}_3\,\bar{e}_2\ e_1\ e_0$



Puertas lógicas

- *Puertas NAND y NOR.*

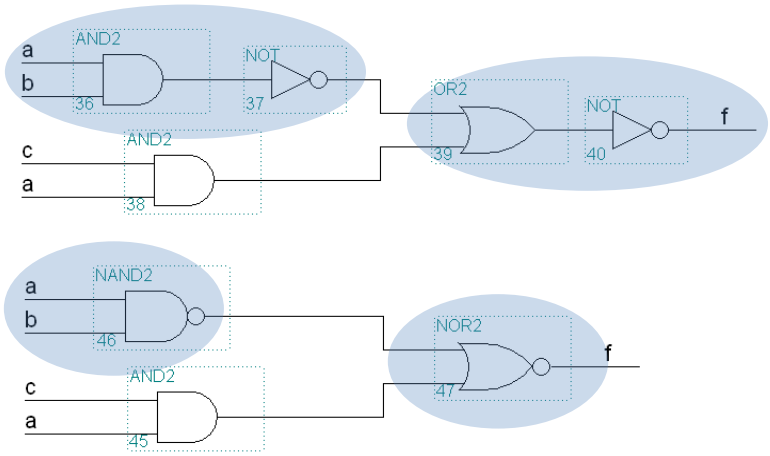
símbolo	puerta	Función lógica
	NAND	$s = \overline{a \cdot b}$
	NOR	$s = \overline{a + b}$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Puertas lógicas

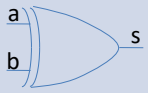
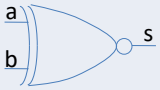
- *Ejemplo:* $f = \overline{\overline{ab} + ac}$



Puertas lógicas

- *Puertas XOR y XNOR.*

S es b si a vale 0, si no es /b («inversor programable»). S es 1 si a y b son distintos.

símbolo	puerta	lógica
	XOR	$s = a \cdot \overline{b} + \overline{a} \cdot b = a \oplus b$
	XNOR	$s = \overline{a \cdot \overline{b} + \overline{a} \cdot b} = a \cdot b + \overline{a} \cdot \overline{b} = \overline{a \oplus b}$

S es 1 si a y b son iguales y es 0 si son diferentes (comparador)

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

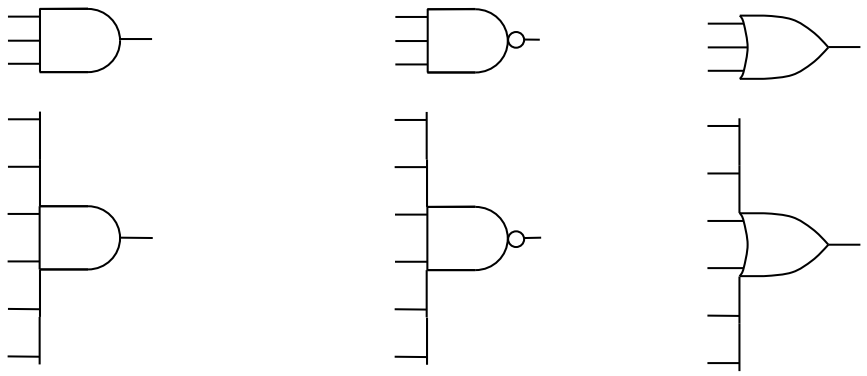
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Puertas lógicas

- Generalización a más entradas (I)



Salida a '1' cuando todas las entradas son '1'

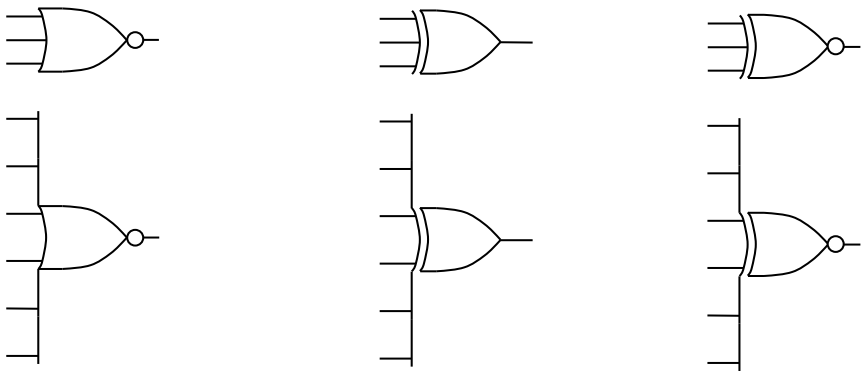
Salida a '1' cuando alguna o algunas entradas son '0'

Salida a '1' cuando alguna o algunas entradas son '1'

BT1_S8 BT1_P17_9

Puertas lógicas

- Generalización a más entradas (II)



Salida a '0' cuando alguna o algunas entradas son '1'

Salida a '1' cuando hay un número impar de entradas a '1' (usualmente no disponible)

Salida a '0' cuando hay un número impar de entradas a '1' (usualmente no disponible)

BT1_S8 BT1_P17_10

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

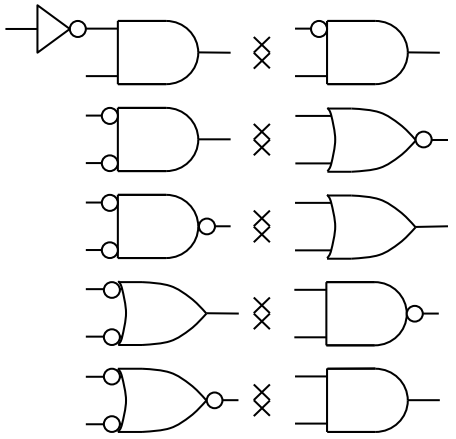
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Puertas lógicas

- Otros símbolos

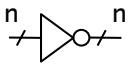


Muy frecuente, también con NAND, OR, NOR, NOT, XOR y XNOR

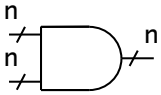
Poco frecuentes, por DeMorgan

Puertas lógicas

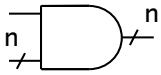
- Con buses



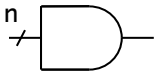
n inversores



n puertas de 2 entradas



n puertas de 2 entradas, una entrada común



una puerta de n entradas

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica 2
Bloque Temático 1

Representación de circuitos combinacionales con puertas lógicas. Análisis y síntesis.

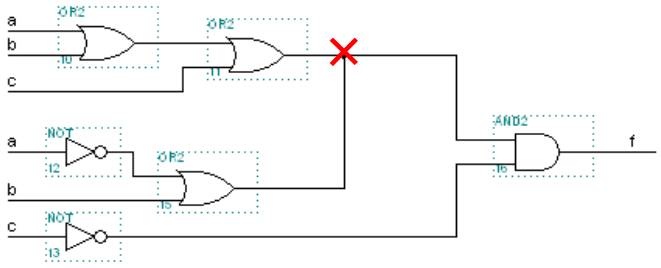
Circuitos con puertas lógicas

- Reglas:
  - Entradas a la izquierda, salidas a la derecha (recomendación)
  - No se conectan entre sí las salidas
  - No se dejan entradas sin conectar
  - No hay bucles

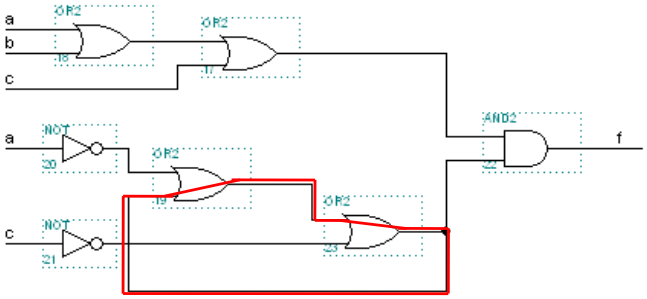
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Circuitos con puertas lógicas



Circuitos con puertas lógicas

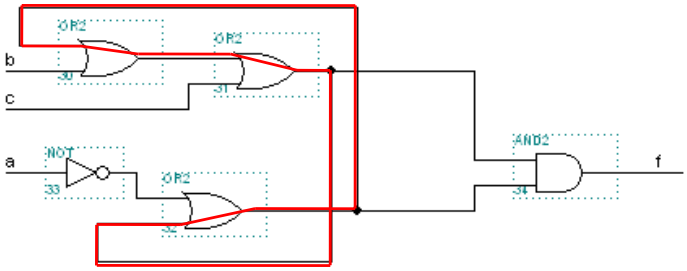


© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

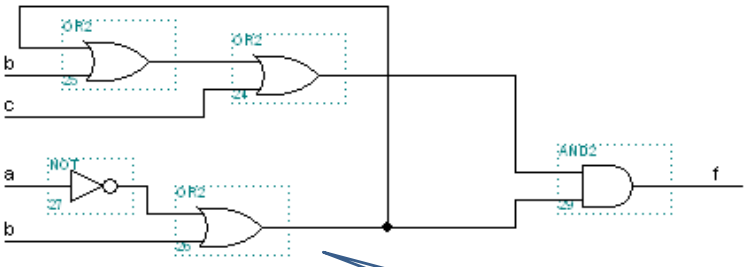
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Circuitos con puertas lógicas



BT1_S8 BT1_P18_5

Circuitos con puertas lógicas



En este caso no hay nada que objetar

BT1_S8 BT1_P18_6

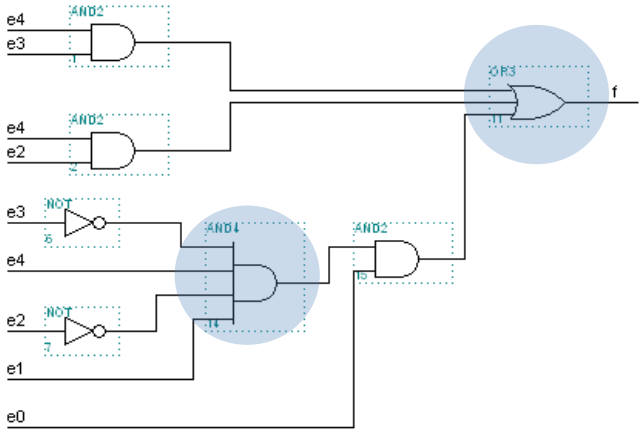
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Circuitos con puertas lógicas

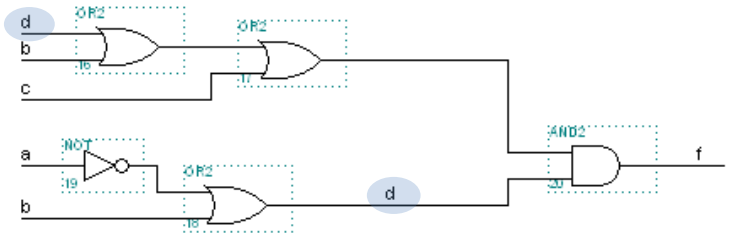
- Puertas con más de 2 entradas



BT1_S8 BT1_P18_7

Circuitos con puertas lógicas

- Conexión mediante etiquetas



BT1_S8 BT1_P18_8

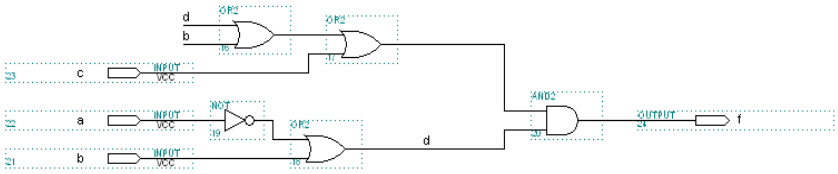
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Circuitos con puertas lógicas

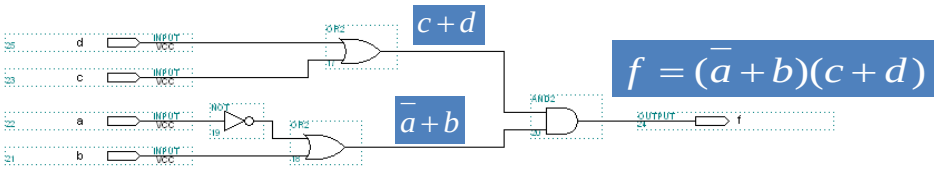
- Puertos de entrada y salida



BT1_S8 BT1_P18_9

Análisis de circuitos

- Ejemplo: Obtenga la función booleana correspondiente a la salida del siguiente circuito.



BT1_S8 BT1_P18_10

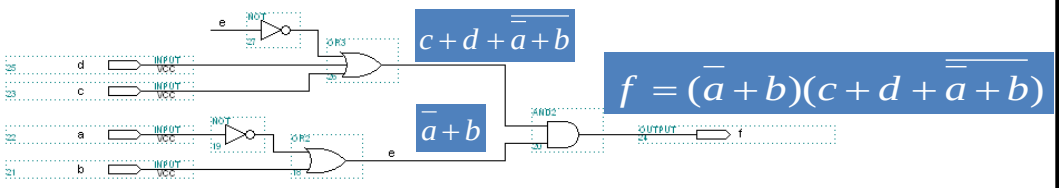
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Análisis de circuitos

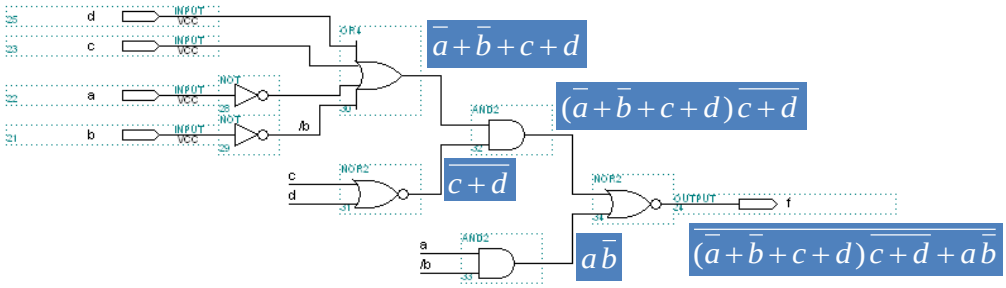
- Ejemplo: Obtenga la función booleana correspondiente a la salida del siguiente circuito.



Síntesis de circuitos

- Ejemplo: Obtenga un circuito que represente la siguiente función booleana:

$$f=\overline{(\overline{a}+\overline{b}+c+d)\,\overline{c+d}+a\overline{b}}$$



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Lectura sobre subsistemas combinacionales y realización de ejercicios de síntesis de sistemas combinacionales.**	**CÓDIGO:** **BT1_AINP16**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Lectura y ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	140 minutos

CALENDARIO:	Octava Sesión. Previo a la Novena Sesión	REQUISITOS:	Metodología de síntesis de sistemas combinacionales. Tablas de verdad resumidas y con indiferencias. Puertas lógicas

CRITERIO DE ÉXITO:	Se ha comprendido el funcionamiento de los subsistemas digitales presentados.

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

1. Introducción. Subsistemas combinacionales

Los subsistemas digitales son sistemas digitales simples que pueden ser utilizados como bloques funcionales para diseñar sistemas más complejos. La utilización de subsistemas es una de las técnicas que permiten abordar el diseño de sistemas digitales complejos. Aunque el diseño de este tipo de sistemas complejos está fuera del ámbito de Electrónica 2, en la asignatura se estudian los subsistemas combinacionales más utilizados: sumadores, multiplexores, codificadores, decodificadores y comparadores.

En este texto se realizará una breve descripción de los subsistemas combinacionales antes mencionados. Para cada subsistema se pondrá un ejemplo de diseño y se propondrá un ejercicio. De esta forma se cubrirán dos objetivos: la iniciación en el aprendizaje de los subsistemas combinacionales y la adquisición de destrezas en la metodología de diseño de sistemas combinacionales.

2. Sumadores

Descripción: Un sumador es un sistema combinacional cuya salida responde a la suma de sus dos entradas. En la Fig. 1 puede verse el símbolo de un sumador de N bits, esto es, un sumador que es capaz de sumar datos de N bits. La salida del sumador tendrá N bits, más un bit de acarreo (carry), tal y como se muestra en la figura.

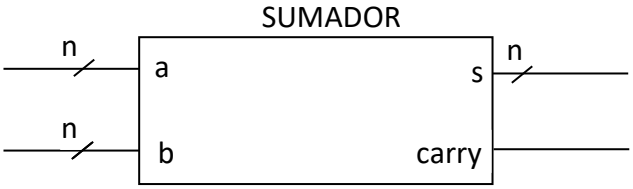


Fig. 1. Símbolo de un sumador de N bits

Normalmente las entradas y la salida del sumador estarán codificadas en binario natural, de forma que el sumador realizará la suma binaria de ‘a’ y ‘b’ (s=a+b). En el caso de que la suma no pueda representarse con los ‘n’ bits de la salida, se activará el bit de ‘carry’, indicando que se ha rebasado la capacidad del sumador. Por ejemplo, para un sumador de 4 bits:

a	b	carry	s
0011	0101	0	1000
1010	0111	1	0001

Ejemplo de diseño: En este ejemplo se diseñará un sumador de un bit.

El diseño se realiza aplicando la metodología general. En primer lugar se identifica la entrada/salida del sistema y su funcionalidad. Un sumador de un bit tiene dos entradas de un bit y una salida de un bit más acarreo, tal y como puede verse en el símbolo de la Fig. 2.

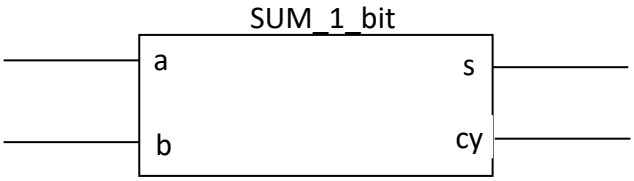


Fig. 2. Sumador de un bit

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

El sumador obtiene en sus salidas la suma binaria de 'a' y 'b' (s=a+b) más el acarreo (cy). El siguiente paso es la obtención de la tabla de verdad. Con dos entradas y dos salidas la tabla debe tener 4 filas y 4 columnas. Para cada una de las combinaciones de las entradas a y b se obtienen los valores de s y cy. Por ejemplo, si 'a=b=1', 's=0' y 'cy=1'.

a	b	cy	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

A partir de la tabla de verdad se obtienen las funciones booleanas correspondientes a las salidas. Suele utilizarse la forma canónica disyuntiva (suma de minitérminos), a no ser que la conjuntiva dé lugar a un número inferior de términos:

$$s = \bar{a}b + a\bar{b}$$

$$cy = ab$$

En este caso las expresiones que se obtienen no son simplificables. El diseño con puertas lógicas es muy simple (recuerde que las conexiones en los esquemas de los circuitos pueden realizarse mediante etiquetas):

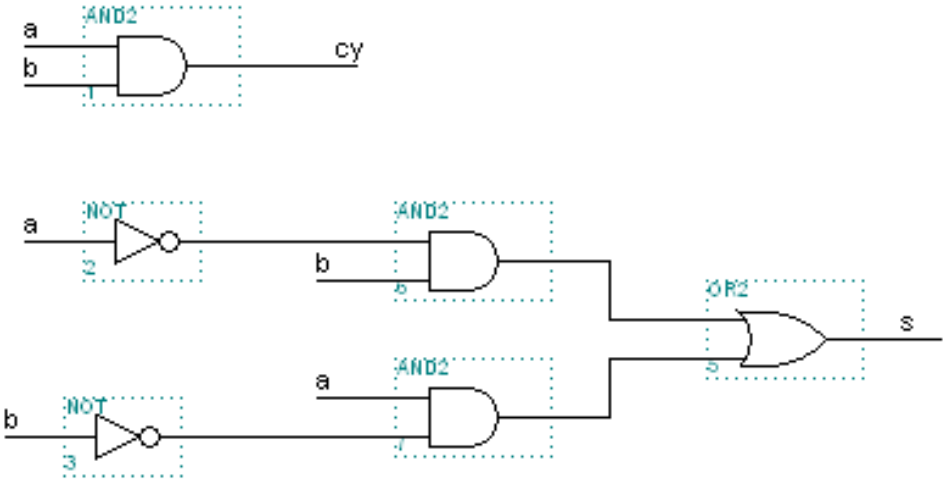


Fig. 3. Circuito sumador de un bit.

Utilizando la puerta XOR puede obtenerse una representación más compacta de este circuito, como se muestra a continuación:

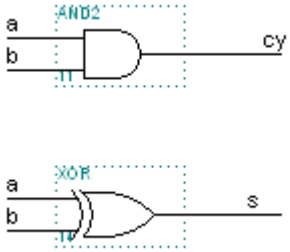


Fig. 4. Circuito sumador de un bit utilizando una puerta XOR.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 1: Diseñe un circuito sumador de dos bits.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

3. Multiplexores

Descripción: Un multiplexor es un sistema combinacional en cuya salida se obtiene el nivel lógico de una de sus entradas, que se selecciona mediante el código aplicado en una o varias entradas de selección. La Fig. 5 muestra un multiplexor de 8 entradas. El sistema posee 3 entradas de selección (sel[2:0]), que permiten seleccionar una de entre las 8 entradas de datos (e_0,..., e_7). Normalmente, las entradas de datos llevan asociado un número (en este caso es el subíndice) que permite conocer qué entrada es seleccionada con una determinada combinación de las entradas de selección. Por ejemplo, si sel[2:0]=100, la salida del multiplexor tomaría el mismo nivel lógico que la entrada e_4 (dado que $100_b = 4_d$).

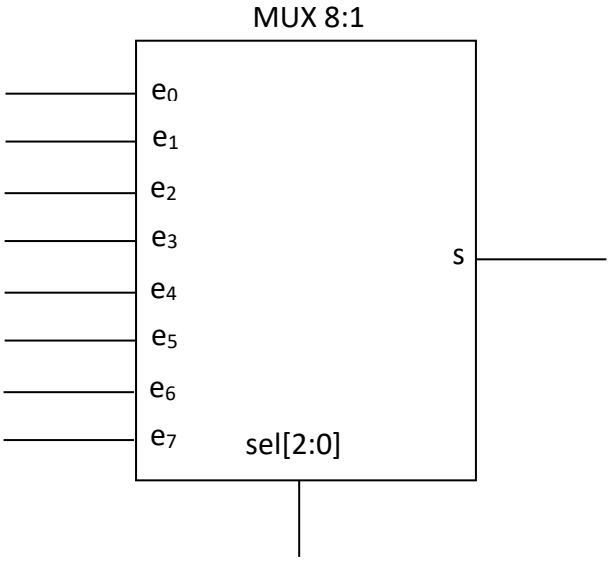


Fig. 5. Multiplexor de 8 entradas (mux 8:1).

Observe que las entradas de selección se encuentran en la parte inferior del símbolo en lugar de estar a la izquierda, como es lo normal en la representación de una entrada. En ocasiones, en los sistemas digitales, las entradas que controlan a otras entradas pueden ubicarse en la parte inferior o superior de los símbolos para diferenciarlas de las entradas de datos.

A continuación se muestra la tabla de verdad resumida del multiplexor de la Fig. 5:

sel_2	sel_1	sel_0	s
0	0	0	e_0
0	0	1	e_1
0	1	0	e_2
0	1	1	e_3
1	0	0	e_4
1	0	1	e_5
1	1	0	e_6
1	1	1	e_7

Aunque la tabla de verdad anterior representa perfectamente el funcionamiento del multiplexor y permite la extracción de las expresiones algebraicas que relacionan la salida con las entradas de selección y datos, en ocasiones se emplea una versión más explícita de esa tabla, que es la que se muestra a continuación:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

sel_2	sel_1	sel_0	e_7	e_6	e_5	e_4	e_3	e_2	e_1	e_0	s
0	0	0	X	X	X	X	X	X	X	0	0
0	0	0	X	X	X	X	X	X	X	1	1
0	0	1	X	X	X	X	X	X	0	X	0
0	0	1	X	X	X	X	X	X	1	X	1
0	1	0	X	X	X	X	X	0	X	X	0
0	1	0	X	X	X	X	X	1	X	X	1
0	1	1	X	X	X	X	0	X	X	X	0
0	1	1	X	X	X	X	1	X	X	X	1
1	0	0	X	X	X	0	X	X	X	X	0
1	0	0	X	X	X	1	X	X	X	X	1
1	0	1	X	X	0	X	X	X	X	X	0
1	0	1	X	X	1	X	X	X	X	X	1
1	1	0	X	0	X	X	X	X	X	X	0
1	1	0	X	1	X	X	X	X	X	X	1
1	1	1	0	X	X	X	X	X	X	X	0
1	1	1	1	X	X	X	X	X	X	X	1

En general, los multiplexores pueden tener N entradas, siendo N habitualmente una potencia de 2, pero sólo tendrán una salida. Un multiplexor de N entradas requiere $\log_2 N$ entradas de selección. Por ejemplo, un multiplexor de 16:1 necesitaría 4 entradas de selección.

En otro orden de cosas, también puede hablarse de multiplexores cuyas entradas sean buses en lugar de bits. Por ejemplo, en la Fig. 6 se muestra un multiplexor de 8:1 con entradas de 8 bits. El símbolo es el mismo que el del multiplexor de la Fig. 5 y posee también 3 entradas de selección. La diferencia es que tanto las entradas como la salida son buses de 8 bits. En general, puede hablarse de multiplexores de N entradas de M bits, con $\log_2 N$ entradas de selección y una salida de M bits.

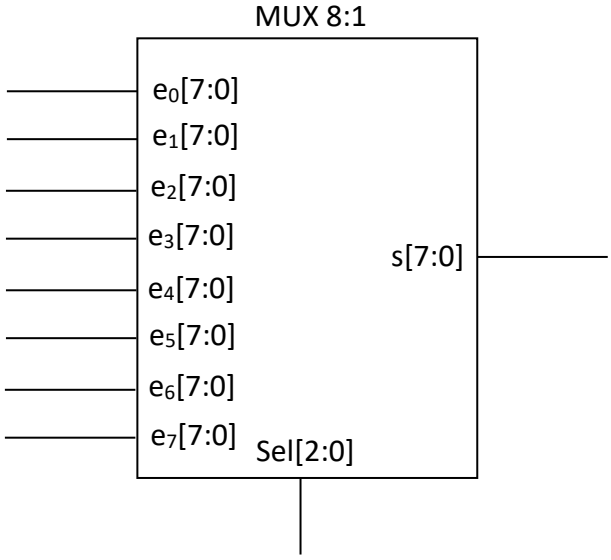


Fig. 6. Multiplexor de 8 entradas de 8 bits cada una

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

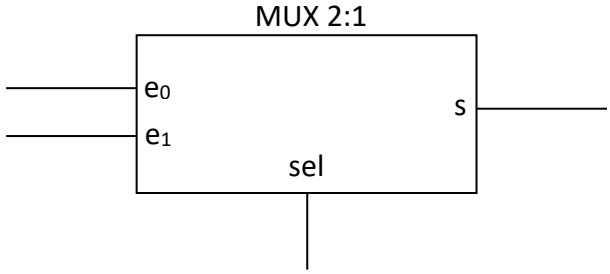
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejemplo de diseño: En este ejemplo se diseñará un multiplexor de dos entradas (2:1).

Un multiplexor de dos entradas tiene dos entradas de datos, una entrada de selección y una salida de datos. A continuación se muestra el símbolo del multiplexor 2:1 y su tabla de verdad resumida (en sus dos formas alternativas):



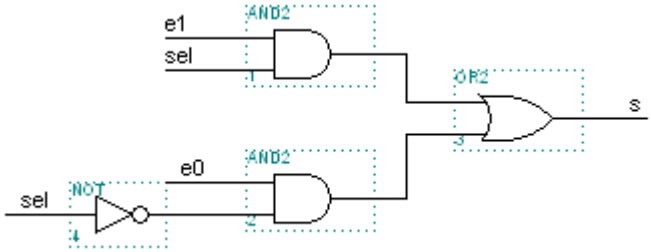
sel	s
0	e_0
1	e_1

sel	e_1	e_0	s
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1

La función booleana para la salida del multiplexor es función de las entradas de datos y de la entrada de selección:

$$s = e_0\,\overline{sel} + e_1\,sel$$

Finalmente, el circuito multiplexor con puertas lógicas se implementa de la siguiente manera:



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 2: Diseñe un circuito multiplexor de 4 entradas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

4. Codificadores de 1 entre N

Descripción: Un codificador de 1 entre N es un sistema combinacional que genera a su salida un código, generalmente binario, cada vez que se activa una de sus N entradas. En general, un codificador binario de 1 entre N tendrá N entradas, siendo habitualmente N una potencia de 2, y un número de salidas igual a $\log_2 N$, que es el número de bits necesario para codificar las N entradas. La Fig. 7 muestra un codificador binario de 1 entre 8, que posee 8 entradas y 3 salidas.

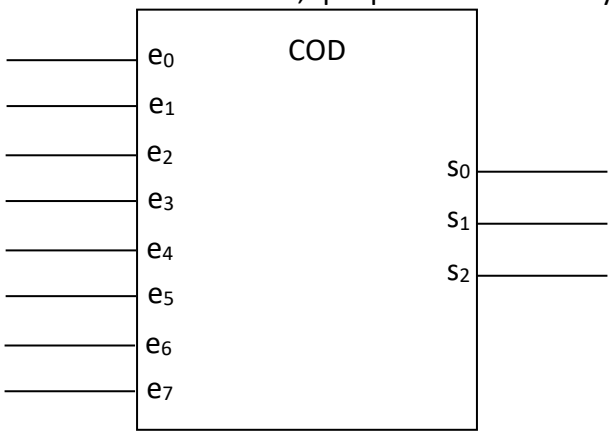


Fig. 7. Codificador binario de 1 entre 8

Este tipo de codificadores pueden diseñarse con o sin prioridad. A continuación se muestra la tabla de verdad resumida de un codificador binario de 1 entre 8 sin prioridad:

e_7	e_6	e_5	e_4	e_3	e_2	e_1	e_0	s_2	s_1	s_0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1
otras								X	X	X

Observe que:

- Cuando todas las entradas están a '0' (inactivas) la salida presenta el código '000_b'
- Cuando se activa la entrada 'e_i' la salida presenta el código binario correspondiente a la entrada, que coincide con el del subíndice.
- El codificador no está preparado para codificar las combinaciones en las que se activan simultáneamente dos o más entradas. En la tabla de verdad, estas combinaciones están todas en la última fila (otras) y cuando se da alguna de ellas el comportamiento del codificador es indiferente.
- No hay forma de distinguir la activación de la entrada E_0 de la no activación de ninguna entrada. En ambos casos el codificador obtiene a su salida el código '000_b'. Naturalmente, podría diseñarse un codificador que distinguiera estos dos casos, pero sería a costa de añadir una cuarta salida.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

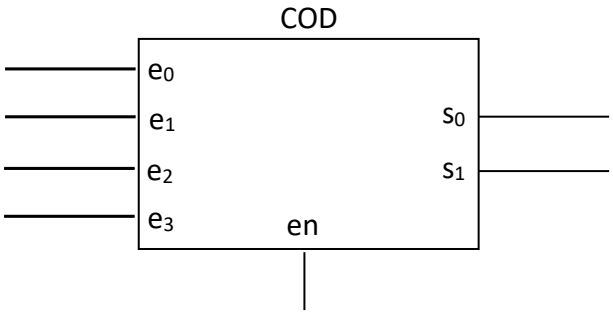
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Los codificadores con prioridad sí están preparados para tener en cuenta los casos en los que dos o más salidas se activan de manera simultánea. En esos casos, el codificador genera a su salida el código correspondiente a la entrada de mayor prioridad. A continuación se muestra la tabla de verdad resumida de un codificador de 1 entre 8 binario con prioridad, en el que se ha establecido que las entradas con subíndices mayores tienen mayor prioridad que las entradas con subíndices más bajos:

e_7	e_6	e_5	e_4	e_3	e_2	e_1	e_0	s_2	s_1	s_0
0	0	0	0	0	0	0	X	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

Observe que de nuevo en este caso el código correspondiente a la activación de la entrada E_0 es indistinguible del código correspondiente a la no activación de ninguna entrada.

Por otra parte, cualquier circuito digital puede tener lo que se denomina una *entrada de habilitación*. Esta es una entrada de control que, cuando se activa, permite que el circuito realice la función para la que ha sido diseñado mientras que, si permanece inactiva, el circuito normalmente deja sus salidas a un nivel lógico predeterminado. En la Fig. 8 se muestra un codificador binario de 1 entre 4 con prioridad y con entrada de habilitación (en). En la tabla de verdad se observa que las salidas del circuito se corresponden con el código binario de la entrada más prioritaria que esté activada mientras la entrada 'en' está activa (a 1). Si la entrada 'en' está a '0', entonces las salidas permanecen fijas a '00_b'.



en	e_3	e_2	e_1	e_0	s_1	s_0
0	X	X	X	X	0	0
1	0	0	0	X	0	0
1	0	0	1	X	0	1
1	0	1	X	X	1	0
1	1	X	X	X	1	1

Fig. 8. Codificador de 1 entre 4 con prioridad y entrada de habilitación.

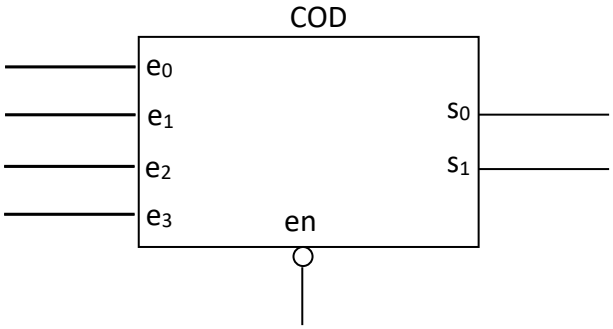
La entrada de habilitación del circuito de la Fig. 8 es activa a nivel alto, es decir, su función dentro del circuito, que es habilitar el funcionamiento del codificador, se lleva a cabo cuando se aplica un '1' en esa entrada. Cuando ocurre lo contrario, como en el codificador de la Fig. 9, entonces se dice que la entrada es activa a nivel bajo. Aunque no es obligatorio, esta circunstancia puede señalizarse en el símbolo del circuito con un círculo, tal y como se ve en la Fig. 9.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

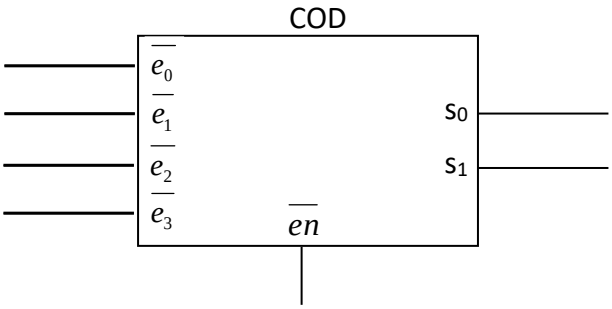
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019



en	e_3	e_2	e_1	e_0	s_1	s_0
1	X	X	X	X	0	0
0	0	0	0	X	0	0
0	0	0	1	X	0	1
0	0	1	X	X	1	0
0	1	X	X	X	1	1

Fig. 9. Codificador de 1 entre 4 con prioridad y entrada de habilitación activa a nivel bajo.

Las entradas del codificador también pueden ser activas a nivel bajo. En este caso una entrada se considera activada cuando se aplica en ella un '0' en lugar de un '1', a diferencia de lo que sucede en los anteriores ejemplos. En la Fig. 10 se muestra un codificador en el que las entradas de habilitación y datos son activas a nivel bajo. En esta ocasión se ha optado por señalizar esta circunstancia por un procedimiento alternativo: las señales que son activas a nivel bajo se denotan con una barra inversora. Esta barra es parte del nombre de la señal; por ejemplo, la entrada de habilitación se denomina '$\overline{en}$', lo cual implica que en ningún caso puede sustituirse en una expresión booleana el término '$\overline{\overline{en}}$' por el término 'en'.



$\overline{en}$	$\overline{e_3}$	$\overline{e_2}$	$\overline{e_1}$	$\overline{e_0}$	s_1	s_0
1	X	X	X	X	0	0
0	1	1	1	X	0	0
0	1	1	0	X	0	1
0	1	0	X	X	1	0
0	0	X	X	X	1	1

Fig. 10. Codificador de 1 entre 4 con prioridad y entradas de habilitación y datos activas a nivel bajo.

También las salidas del codificador pueden ser activas a nivel bajo. Cuando esto sucede, las salidas se generan invertidas con respecto al estado que tendrían si fuesen activas a nivel alto. Por ejemplo, el codificador de la Fig. 11 tiene salidas activas a nivel bajo.

e_3	e_2	e_1	e_0	$\overline{s_1}$	$\overline{s_0}$
X	X	X	X	1	1
0	0	0	X	1	1
0	0	1	X	1	0
0	1	1	X	0	1
1	X	X	X	0	0

Fig. 11. Codificador de 1 entre 4 con prioridad y salidas activas a nivel bajo.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

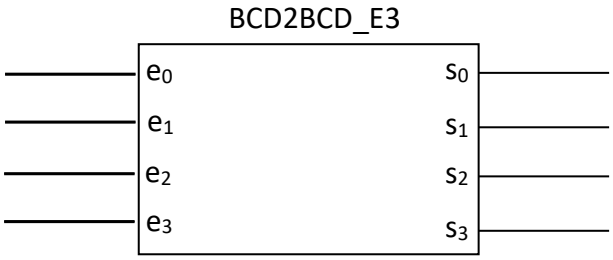
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Es importante señalar que, aunque los conceptos de entradas y salidas activas a nivel alto o bajo se han ejemplificado sobre un codificador con prioridad, son válidos (y se utilizan) en cualquier circuito digital.

Para terminar, el término *codificador* no se aplica solo a los codificadores de 1 a N. Por un lado, el código de salida no tiene porqué ser binario natural. Por otro lado, las entradas pueden estar codificadas también con cualquier tipo de código. A modo de ejemplo, en la Fig. 12 se muestra un codificador de 'BCD' a 'BCD exceso 3'. Los circuitos cambiadores de código como el de este ejemplo reciben también el nombre de *transcodificadores*.

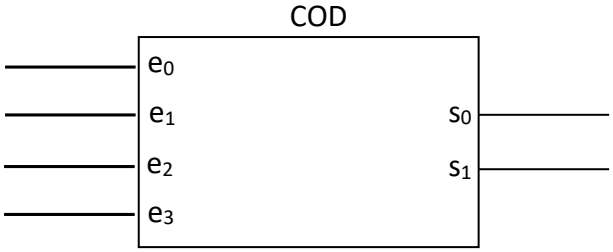


e_3	e_2	e_1	e_0	s_3	s_2	s_1	s_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
resto				X	X	X	X

Fig. 12. Codificador de BCD a BCD exceso 3.

Ejemplo de diseño: En este ejemplo se diseñará un codificador de 1 entre 4 sin prioridad.

El circuito tiene 4 entradas y dos salidas. Las salidas generan el código binario correspondiente a cada entrada cuando únicamente una entrada está activa. Si no se activa ninguna entrada el código de salida será el correspondiente a la entrada e_0. Para el resto de casos (cuando dos o más entradas se activan al mismo tiempo) la salida es indiferente, esto es, el comportamiento del circuito para estos casos puede ser cualquiera (el que elija el diseñador).



e_3	e_2	e_1	e_0	s_1	s_0
0	0	0	X	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
resto				X	X

En casos como este, si no hay otras consideraciones, las salidas indiferentes se fuerzan a '0' o a '1' con el objetivo de reducir el número de términos de las expresiones booleanas. Por ejemplo, forzando las salidas indiferentes a '0', se tiene:

$$s_0 = \bar{e}_3\,\bar{e}_2\,e_1\,\bar{e}_0 + e_3\,\bar{e}_2\,\bar{e}_1\,\bar{e}_0$$

$$s_1 = \bar{e}_3\,e_2\,\bar{e}_1\,\bar{e}_0 + e_3\,\bar{e}_2\,\bar{e}_1\,\bar{e}_0$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

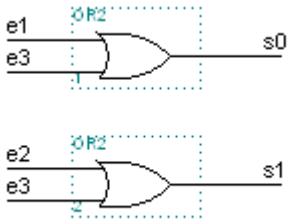
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Las expresiones anteriores pueden simplificarse si tenemos en cuenta que, en realidad, el requisito para la activación de s_0 es que se activen e_1 o e_3, independientemente del estado de las otras entradas. De la misma forma, el requisito para la activación de s_1 es que se activen s_2 o s_3, sin tener en cuenta al resto de entradas. Esto nos lleva a un conjunto de ecuaciones muy simple:

$$s_0 = e_1 + e_3$$

$$s_1 = e_2 + e_3$$

El circuito resultante es también de una gran simplicidad:



La simplificación que se ha llevado a cabo en este ejercicio resulta bastante evidente (una vez se ha visto cómo hacerlo), y además la idea funciona para codificadores sin prioridad con cualquier número de entradas. No obstante, desde el punto de vista de la metodología de diseño, no resulta tan evidente, si tenemos en cuenta que, en realidad, las ecuaciones obtenidas están representando la siguiente tabla de verdad (en la que las filas que antes estaban marcadas con indiferencias se han oscurecido):

e_3	e_2	e_1	e_0	s_1	s_0
0	0	0	X	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 3: Diseñe un circuito codificador de 1 entre 8 sin prioridad.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

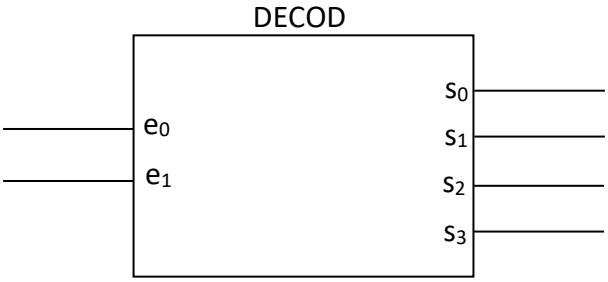
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

5. Decodificadores de M a N

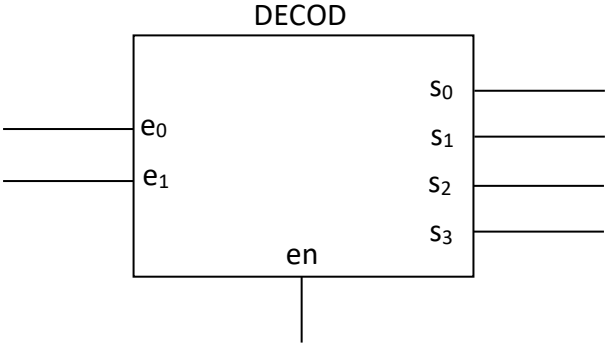
Descripción: Un decodificador de M a N es un sistema combinacional que activa una salida de entre N posibles en función del código aplicado en las M entradas. En general, en un decodificador de M a N, el número de salidas, N, será habitualmente una potencia de 2 y el número de entradas, M, será igual a $\log_2 N$. La Fig. 13 muestra un decodificador binario de 2 a 4, junto con su tabla de verdad.



e_1	e_0	s_3	s_2	s_1	s_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Fig. 13. Decodificador binario de 2 a 4.

Observe que en este tipo de decodificadores no existe el concepto de prioridad. Sí pueden tener, en cambio, una entrada de habilitación, como en el ejemplo de la Fig**. 14**.



en	e_1	e_0	s_3	s_2	s_1	s_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Fig. 14. Decodificador binario de 2 a 4 con entrada de habilitación.

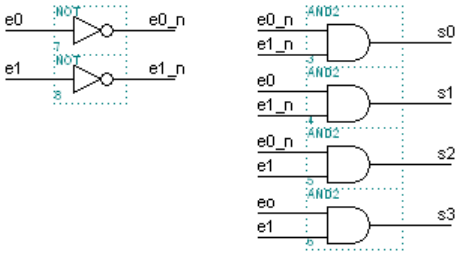
Ejemplo de diseño: En este ejemplo se diseñará un decodificador de 2 a 4.

El símbolo y la tabla de verdad son las que se han mostrado en la Fig. 13. Teniendo en cuenta la tabla de verdad, las expresiones booleanas de las salidas serían las siguientes:

$$s_0 = \bar{e}_1 \bar{e}_0 \qquad s_1 = \bar{e}_1 e_0$$

$$s_2 = e_1 \bar{e}_0 \qquad s_3 = e_1 e_0$$

las cuales se corresponden con este circuito:



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 4: Diseñe un circuito decodificador de 3 a 8.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

6. Comparadores

Descripción: Un comparador de N bits es un sistema combinacional que compara dos números, 'A' y 'B', de N bits cada uno, generalmente codificados en binario natural, y activa una salida si se da una determinada condición, que puede ser cualquiera de las siguientes:

A=B	A>B	A<B	A≥B	A≤B	A≠B
aeb	agb	alb	ageb	aleb	aneb

Por ejemplo, la Fig. 15 muestra un comparador de 2 bits que activa una salida cuando A≥B.

b_1	b_0	a_1	a_0	ageb
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Fig. 15. Comparador de 2 bits tipo a≥b

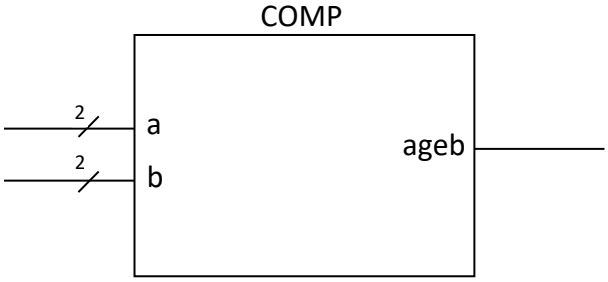
Puede ocurrir que el comparador trabaje con códigos diferentes del binario natural, por ejemplo, Complemento a 2 o Valor absoluto y signo. La Fig. 16 muestra un comparador de 2 bits que detecta si A≥B, en el que las entradas 'A' y 'B' están codificadas en Complemento a dos.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019



b_1	b_0	a_1	a_0	ageb
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Fig. 16. Comparador de 2 bits en Complemento a 2 tipo a≥b

Ejemplo de diseño: En este ejemplo se diseñará un comparador binario de 2 bits tipo A=B.

El circuito tiene dos entradas de 2 bits y una salida. La siguiente figura muestra el símbolo y la tabla de verdad:

b_1	b_0	a_1	a_0	aeb
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

A partir de la tabla de verdad, puede obtenerse la siguiente expresión utilizando la forma canónica disyuntiva (suma de productos):

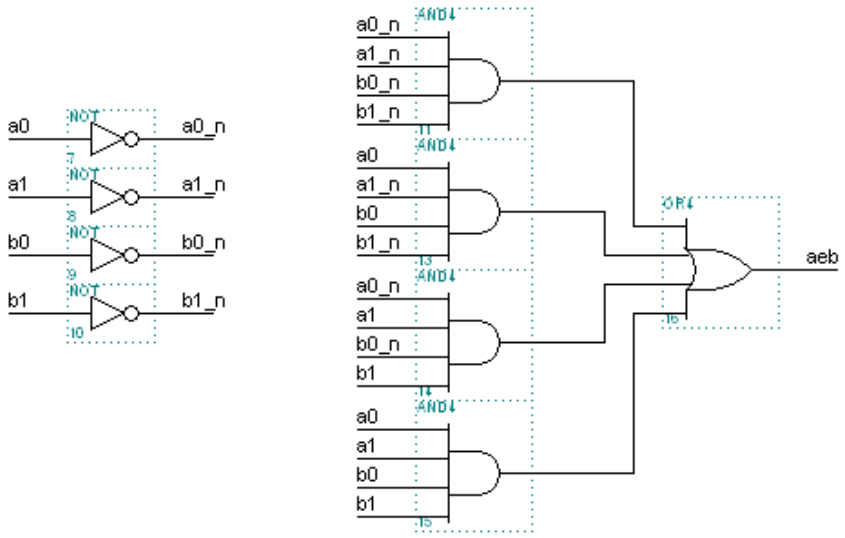
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

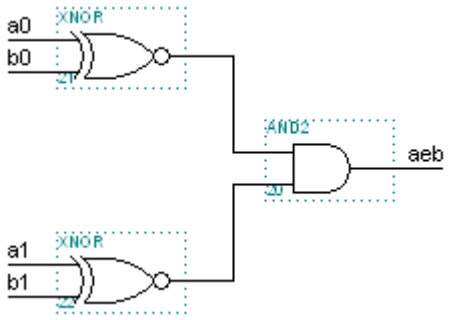
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

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

la cual puede implementarse con el siguiente circuito:



Una implementación alternativa puede obtenerse recordando una interesante propiedad de las puertas XNOR: su salida es '1' si sus dos entradas son iguales y '0' si son diferentes. Observando la tabla de verdad del comparador que se está diseñando, puede verse que su salida se activa siempre que 'a_0=b_0' y 'a_1=b_1' de manera que el circuito también podría realizarse de esta forma:



Este resultado es generalizable para comparadores tipo 'A=B' de cualquier número de bits, lo cual hace que su diseño sea inmediato, sin necesidad de llevar a cabo los pasos de la metodología de diseño. Para comparadores tipo 'A≠B' puede negarse la salida con un inversor o bien puede utilizarse la misma estructura del circuito anterior con puertas XOR en lugar de las puertas XNOR y una OR en lugar de una AND. Para otros tipos de comparadores no existen estos atajos.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 5: Diseñe un circuito comparador de 2 bits del tipo ‘A>B’.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

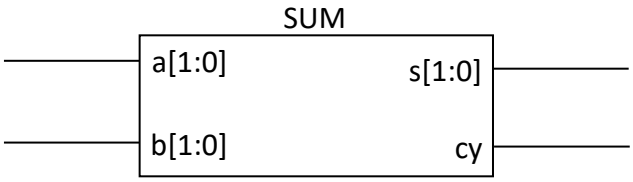
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones de los ejercicios.

A continuación se detallan las soluciones a los 5 ejercicios propuestos en esta actividad. No mire las soluciones hasta que no haya resuelto los ejercicios.

Ejercicio 1: Diseñe un circuito sumador de dos bits.

El circuito sumador de dos bits posee dos entradas de dos bits y una salida de dos bits más acarreo. A continuación se muestran su símbolo y su tabla de verdad:



a_1	a_0	b_1	b_0	cy	s_1	s_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

A partir de la tabla de verdad se obtienen las siguientes expresiones booleanas:

$$s_0 = \sum m(1,3,4,6,9,11,12,14) = \bar{a}_1\bar{a}_0\bar{b}_1b_0 + \bar{a}_1\bar{a}_0b_1b_0 + \bar{a}_1a_0\bar{b}_1\bar{b}_0 + \bar{a}_1a_0b_1\bar{b}_0 + a_1\bar{a}_0\bar{b}_1b_0 + a_1\bar{a}_0b_1b_0 + a_1a_0\bar{b}_1\bar{b}_0 + a_1a_0b_1\bar{b}_0$$

$$s_1 = \sum m(2,3,5,6,8,9,12,15) = \bar{a}_1\bar{a}_0b_1\bar{b}_0 + \bar{a}_1\bar{a}_0b_1b_0 + \bar{a}_1a_0\bar{b}_1b_0 + \bar{a}_1a_0b_1\bar{b}_0 + a_1\bar{a}_0\bar{b}_1\bar{b}_0 + a_1\bar{a}_0\bar{b}_1b_0 + a_1a_0\bar{b}_1\bar{b}_0 + a_1a_0b_1b_0$$

$$cy = \sum m(7,10,11,13,14,15) = \bar{a}_1a_0b_1b_0 + a_1\bar{a}_0b_1\bar{b}_0 + a_1\bar{a}_0b_1b_0 + a_1a_0\bar{b}_1b_0 + a_1a_0b_1\bar{b}_0 + a_1a_0b_1b_0$$

Antes de representar el circuito con puertas lógicas resulta conveniente simplificar las expresiones anteriores:

$$s_0 = \bar{a}_0b_0 + a_0\bar{b}_0$$

$$s_1 = \bar{a}_1\bar{a}_0b_1 + \bar{a}_1b_1\bar{b}_0 + a_1\bar{b}_1\bar{b}_0 + a_1\bar{a}_0\bar{b}_1 + \bar{a}_1a_0\bar{b}_1b_0 + a_1a_0b_1b_0$$

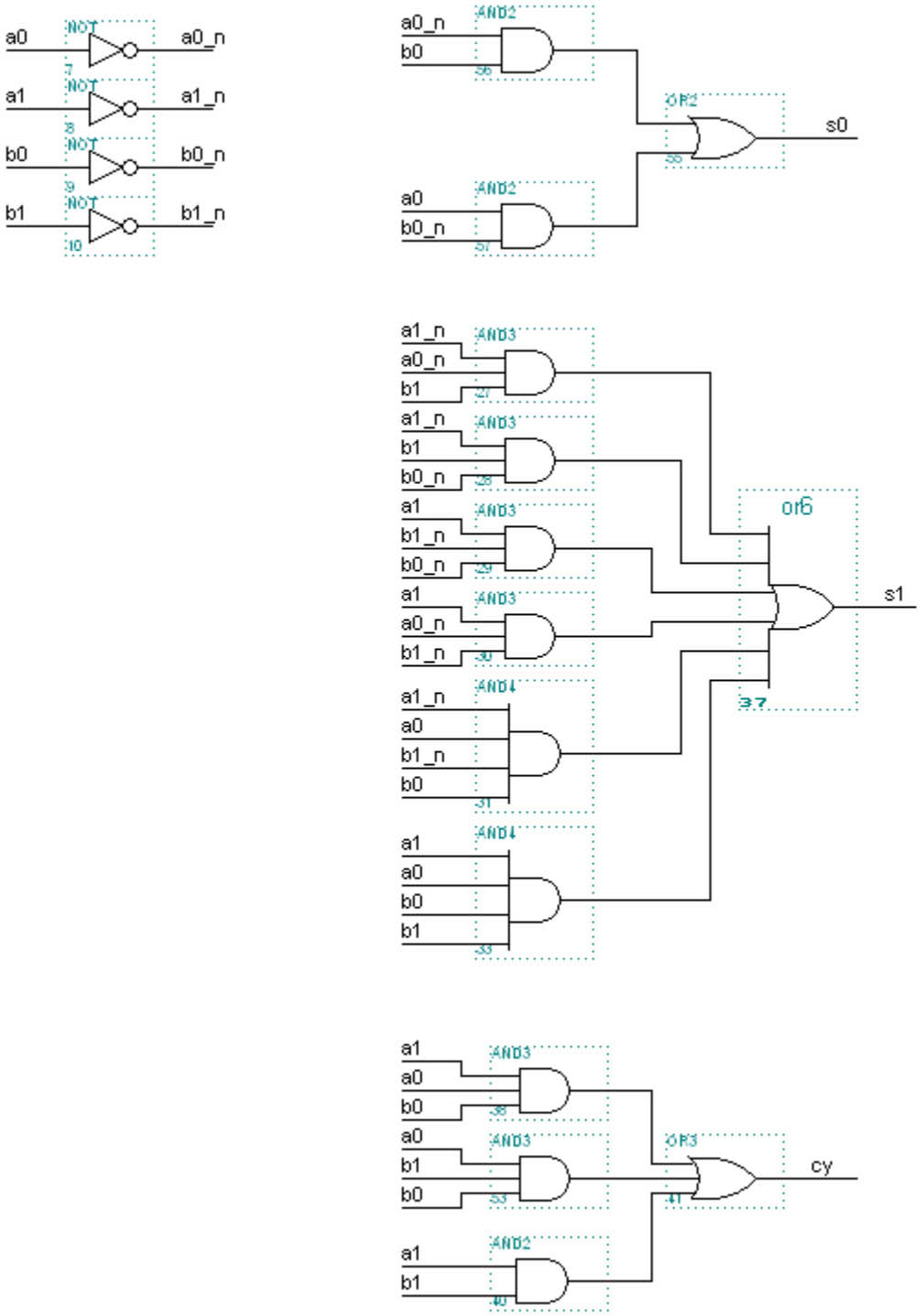
$$cy = a_1a_0b_0 + a_0b_1b_0 + a_1b_1$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Finalmente, el circuito sumador de dos bits sería el siguiente:



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

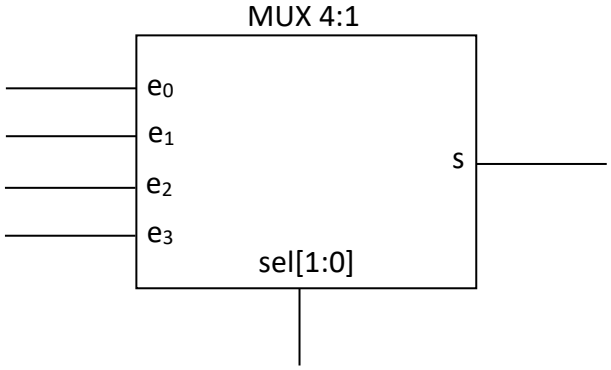
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 2: Diseñe un circuito multiplexor de 4 entradas.

El circuito tiene cuatro entradas de datos, 'e0,…, e3', dos entradas de selección, 'sel0' y 'sel1', y una salida de datos, 's'. La tabla de verdad resumida muestra que la salida del multiplexor toma el valor de la entrada cuyo subíndice corresponde a la combinación binaria que se introduce en las entradas de selección.

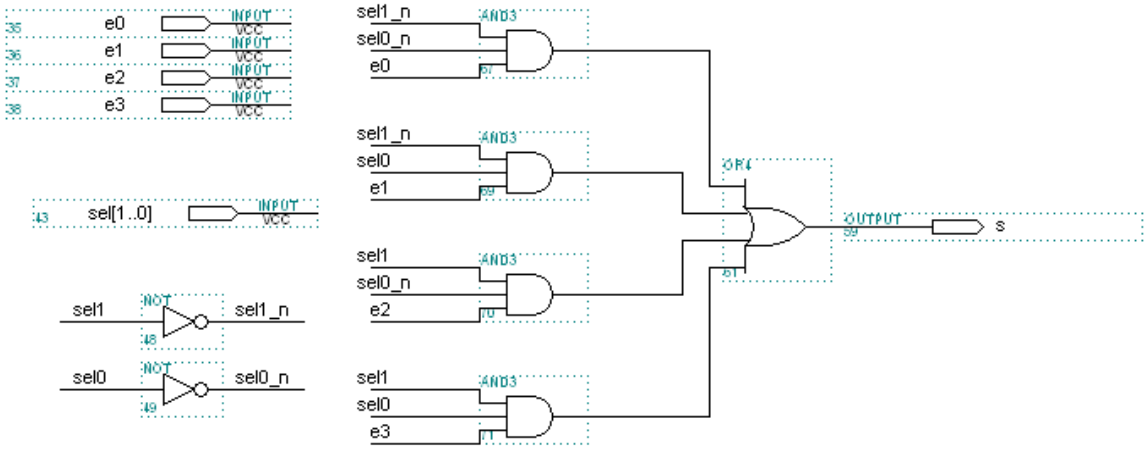


sel_1	sel_0	s
0	0	e_0
0	1	e_1
1	0	e_2
1	1	e_3

A partir de la tabla de verdad se obtiene la ecuación de la salida en función de las entradas:

$$s = e_0\ \overline{sel_1}\ \overline{sel_0} + e_1\ \overline{sel_1}\ sel_0 + e_2\ sel_1\ \overline{sel_0} + e_3\ sel_1\ sel_0$$

A partir de esta ecuación se obtiene la siguiente representación del multiplexor con puertas lógicas:



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

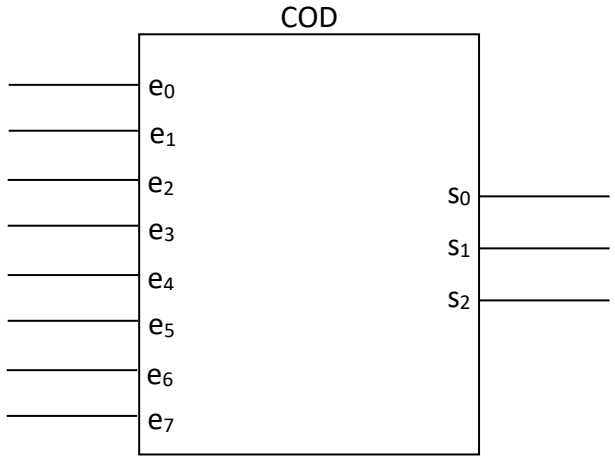
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 3: Diseñe un circuito codificador de 1 entre 8 sin prioridad.

El símbolo del codificador de 1 entre 8 sin prioridad y su tabla de verdad resumida ya se mostraron en el apartado 4 de este texto:



e_7	e_6	e_5	e_4	e_3	e_2	e_1	e_0	s_2	s_1	s_0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1
otras								X	X	X

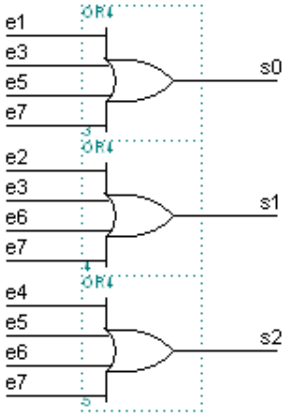
Tal y como se explicó en el ejemplo de diseño del apartado 4, la implementación de un codificador de estas características es muy sencilla, teniendo en cuenta que:

- La salida s_0 se pone a '1' cuando se activa la entrada e_1 o la e_3 o la e_5 o la e_7
- La salida s_1 se pone a '1' cuando se activa la entrada e_2 o la e_3 o la e_6 o la e_7
- La salida s_2 se pone a '1' cuando se activa la entrada e_4 o la e_5 o la e_6 o la e_7

Las ecuaciones booleanas del codificador son:

$$s_0 = e_1 + e_3 + e_5 + e_7$$
$$s_1 = e_2 + e_3 + e_6 + e_7$$
$$s_2 = e_4 + e_5 + e_6 + e_7$$

y se corresponden con la siguiente representación con puertas lógicas:



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

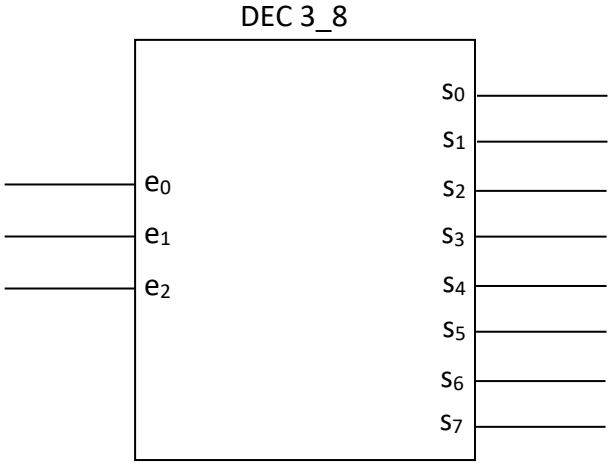
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 4: Diseñe un circuito decodificador de 3 a 8.

El circuito tiene 3 entradas y 8 salidas. Tal y como se ve en la tabla de verdad, la salida 's_i' se activa cuando en la entrada se aplica el código binario 'i'.

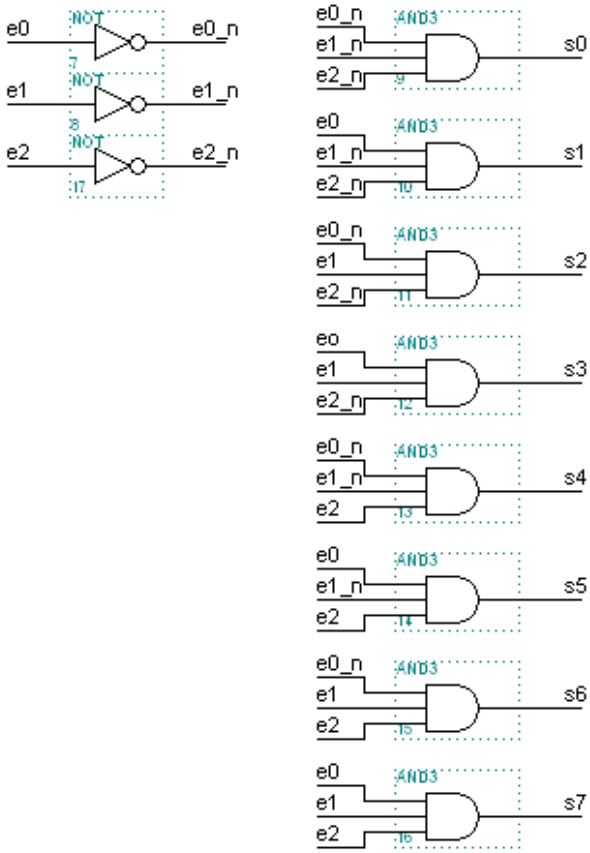


e_2	e_1	e_0	s_7	s_6	s_5	s_4	s_3	s_2	s_1	s_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Las ecuaciones booleanas de las salidas en función de las entradas se obtienen a partir de la tabla de verdad:

$$s_0 = \bar{e}_2\,\bar{e}_1\,\bar{e}_0 \qquad s_1 = \bar{e}_2\,\bar{e}_1\,e_0 \qquad s_2 = \bar{e}_2\,e_1\,\bar{e}_0 \qquad s_3 = \bar{e}_2\,e_1\,e_0$$

$$s_4 = e_2\,\bar{e}_1\,\bar{e}_0 \qquad s_5 = e_2\,\bar{e}_1\,e_0 \qquad s_6 = e_2\,e_1\,\bar{e}_0 \qquad s_7 = e_2\,e_1\,e_0$$



Ejercicio 5: Diseñe un circuito comparador de 2 bits del tipo 'A>B'.

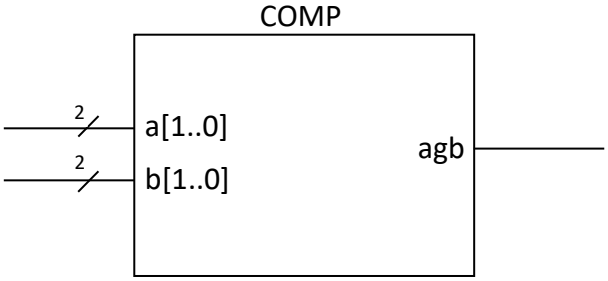
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

El circuito posee dos entradas de dos bits, 'a' y 'b', y una salida. 'agb'. La salida se activa cuando el valor binario de la entraba 'a' es mayor que el de la entrada 'b'. Este comportamiento está descrito en la siguiente tabla de verdad:



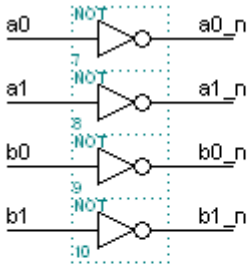
b_1	b_0	a_1	a_0	agb
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

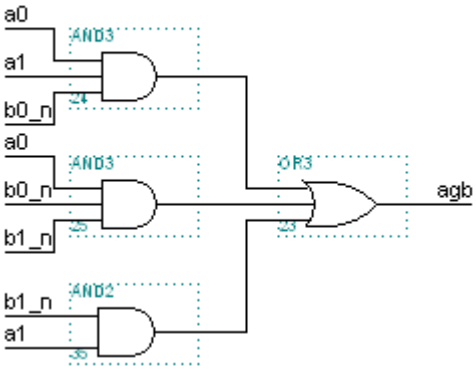
La ecuación de la salida en función de las entradas es:

$$agb = \overline{b}_1\overline{b}_0\overline{a}_1a_0 + \overline{b}_1\overline{b}_0a_1\overline{a}_0 + \overline{b}_1\overline{b}_0a_1a_0 + \overline{b}_1b_0a_1\overline{a}_0 + \overline{b}_1b_0a_1a_0 + b_1\overline{b}_0a_1a_0$$

$$= \overline{b}_1a_1 + \overline{b}_1\overline{b}_0a_0 + \overline{b}_0a_1a_0$$

Y, finalmente, la representación del circuito con puertas lógicas:





© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre diseño de circuitos combinacionales.**	**CÓDIGO:** **BT1_ATGP4**

GRUPO		FECHA:	

MODALIDAD:	Aprendizaje en Grupo	TIPO:	Presencial	DURACIÓN:	50 minutos
CALENDARIO:	Novena Sesión.	REQUISITOS:	Metodología de síntesis de circuitos combinacionales. Puertas lógicas.		

CRITERIO DE ÉXITO:	Los dos ejercicios bien hechos
INTERDEPENDENCIA POSITIVA:	No procede
EXIGIBILIDAD INDIVIDUAL:	No procede
ROLES:	No procede
HABILIDADES SOCIALES:	No procede
REFLEXIÓN DE GRUPO:	No procede

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	No procede

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE I. Descripción de la Actividad.

El profesor irá presentando diversos ejemplos de diseño de circuitos combinacionales, correspondientes a algunos ejemplos de subsistemas combinacionales típicos: codificadores, decodificadores y multiplexores. Los estudiantes, mediante trabajo en grupo, resolverán los ejercicios que se plantean en este documento, que suponen pequeñas variantes sobre el ejemplo expuesto por el profesor.

El orden recomendado para abordar los ejercicios es el siguiente: Ejemplo A, Ejercicio 1, Ejemplo B, Ejercicio 2 y Ejemplo C.

PARTE II. Enunciados de los Ejercicios.

EJERCICIO 1) Codificador con prioridad de 4 a 2.

Diseñe un circuito combinacional con 4 entradas que obtenga a su salida el código binario correspondiente a la entrada activada. Si dos o más entradas se activan al mismo tiempo la salida codifica la más prioritaria, que en este caso va a ser la que tenga un subíndice mayor. Si no se activa ninguna entrada la salida debe permanecer a 0. Indique un diagrama de bloque con las entradas y salidas del circuito y obtenga su tabla de verdad, funciones booleanas de las salidas y el correspondiente circuito con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 2) Decodificador de 3 a 8 con entradas y salidas activas a nivel alto.

Diseñe un decodificador de 3 a 8 con entradas y salidas activas a nivel alto. Dibuje un diagrama de bloques con las entradas y salidas del circuito y obtenga su tabla de verdad, funciones booleanas de las salidas y el correspondiente circuito con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios de caracterización, síntesis, simplificación y representación con puertas lógicas de sistemas combinacionales.**	**CÓDIGO:** **BT1_AINP17**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Individual.	TIPO:	No presencial	DURACIÓN:	60 minutos

CALENDARIO:	Novena Sesión. Antes de la Decima sesión	REQUISITOS:	Metodología de síntesis de sistemas combinacionales. Simplificación. Tablas de verdad resumidas y con indiferencias. Puertas lógicas

CRITERIO DE ÉXITO:	Se han realizado bien dos de los tres ejercicios.

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

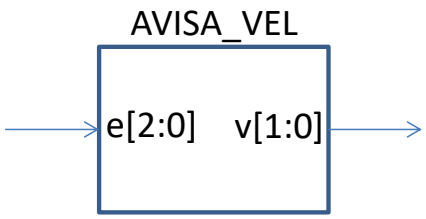
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 1: Avisador velocidad. Un avisador de velocidad recibe como entrada el tipo de carretera por la que circula un vehículo (autopista, autovía, nacional, comarcal, local o camino) y genera a partir de este dato la información sobre la velocidad máxima que puede alcanzar el vehículo en esa vía (120 en autopista y autovía, 110 en nacional, 90 en comarcal y local y 60 Km/h en camino). Suponga que se utiliza el siguiente esquema de codificación para las entradas y las salidas:



Tipo de carretera	e_2	e_1	e_0
autopista	0	0	0
autovía	0	0	1
nacional	0	1	0
comarcal	0	1	1
local	1	0	0
camino	1	0	1

Velocidad máxima	v_1	v_0
120	0	0
110	0	1
90	1	0
60	1	1

Obtenga la tabla de verdad del circuito. Tenga en cuenta que, para aquellas combinaciones de las entradas para las cuales los valores de las salidas sean indiferentes, éstas deberán estar a 'X'.

e_2	e_1	e_0	v_1	v_0

En la tabla de verdad que ha obtenido anteriormente, las salidas que están a 'X' pueden valer '0' o '1' sin alterar el funcionamiento relevante del circuito combinacional. De esta manera, la antedicha tabla de verdad representa en realidad varias posibles funciones v_0 diferentes (también varias v_1). ¿Cuántas? Razone la respuesta.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Para v_0: obtenga todas las posibles funciones que representa la tabla de verdad con indiferencias.

Para v_1: obtenga todas las posibles funciones que representa la tabla de verdad con indiferencias.

Defina nuevamente la tabla de verdad escogiendo para las salidas indiferentes los valores que hacen que las funciones que se obtienen sean las más simples.

e_2	e_1	e_0	v_1	v_0

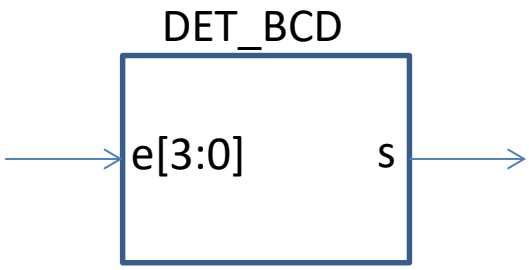
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 2: Obtenga la tabla de verdad de un sistema que detecte, con un '1' lógico, si determinado número BCD es mayor o igual a 6. El comportamiento del circuito para números que no sean BCD es indiferente; en estos casos la salida debe ser 'X'.



e_3	e_2	e_1	e_0	s

Obtenga razonadamente el número de posibles funciones que representa la tabla de verdad que ha obtenido anteriormente.

Obtenga una función booleana para la salida del circuito. Para ello, seleccione previamente un conjunto de valores para las combinaciones indiferentes y anote cada valor al lado de su correspondiente 'X' en la tabla de verdad. Realice esta selección de valores buscando que la función de salida sea simple, pero limitando el esfuerzo empleado en esa búsqueda a un máximo de cinco minutos.

Indique cuál es el criterio que se ha utilizado para seleccionar los valores para las combinaciones indiferentes.

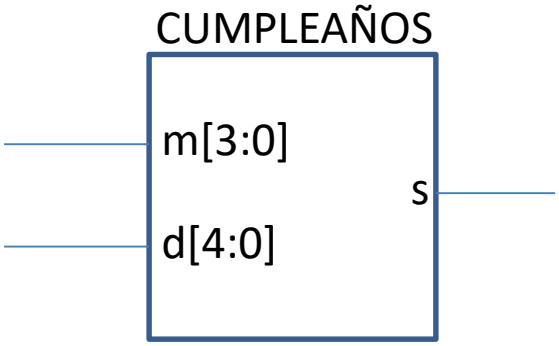
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 3: Realice un sistema que detecte cuándo es el cumpleaños de cada miembro de una familia de 3. Las fechas de los cumpleaños son: 03 de mayo de 1968, 11 de julio de 1972 y 25 de octubre de 2001.



Obtenga, de forma razonada, el número de filas que tiene la tabla de verdad completa correspondiente a este circuito.

Obtenga la tabla de verdad resumida.

Obtenga la función booleana correspondiente a la salida del circuito.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

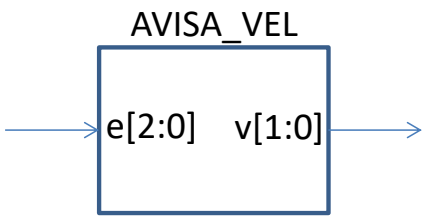
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones de los ejercicios.

A continuación se detallan las soluciones a los 8 ejercicios propuestos en esta actividad. No mire las soluciones hasta que no haya resuelto los ejercicios.

Ejercicio 1: Avisador velocidad. Un avisador de velocidad recibe como entrada el tipo de carretera por la que circula un vehículo (autopista, autovía, nacional, comarcal, local o camino) y genera a partir de este dato la información sobre la velocidad máxima que puede alcanzar el vehículo en esa vía (120 en autopista y autovía, 110 en nacional, 90 en comarcal y local y 60 Km/h en camino). Suponga que se utiliza el siguiente esquema de codificación para las entradas y las salidas:



Tipo de carretera	e_2	e_1	e_0
autopista	0	0	0
autovía	0	0	1
nacional	0	1	0
comarcal	0	1	1
local	1	0	0
camino	1	0	1

Velocidad máxima	v_1	v_0
120	0	0
110	0	1
90	1	0
60	1	1

Obtenga la tabla de verdad del circuito. Tenga en cuenta que, para aquellas combinaciones de las entradas para las cuales los valores de las salidas sean indiferentes, éstas deberán estar a 'X'.

e_2	e_1	e_0	v_1	v_0
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	X	X
1	1	1	X	X

En la tabla de verdad que ha obtenido anteriormente, las salidas que están a 'X' pueden valer '0' o '1' sin alterar el funcionamiento práctico del circuito combinacional. De esta manera, la antedicha tabla de verdad representa en realidad varias posibles funciones v_0 diferentes (también varias v_1). ¿Cuántas? Razone la respuesta.

4 para v_0 y otras tantas para v_1. El número de posibles funciones definidas por una tabla de verdad con N indiferencias es 2^N. En este caso, cada función tiene N=2 indiferencias.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Para v_0: obtenga todas las posibles funciones que representa la tabla de verdad con indiferencias.

A continuación se representa la tabla de verdad con todas las posibles funciones v_0, que se han denotado como v_{0a}, v_{0b}, v_{0c} y v_{0d}.

e_2	e_1	e_0	v_{0d}	v_{0c}	v_{0b}	v_{0a}
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	1	1	1	1
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	1	1	1	1
1	1	0	1	1	0	0
1	1	1	1	0	1	0

Las funciones lógicas pueden obtenerse, por ejemplo, como suma de minitérminos:

$$v_{0a} = \bar{e}_2 e_1 \bar{e}_0 + e_2 \bar{e}_1 e_0$$
$$v_{0b} = \bar{e}_2 e_1 \bar{e}_0 + e_2 \bar{e}_1 e_0 + e_2 e_1 e_0 = \bar{e}_2 e_1 \bar{e}_0 + e_2 e_0$$
$$v_{0c} = \bar{e}_2 e_1 \bar{e}_0 + e_2 \bar{e}_1 e_0 + e_2 e_1 \bar{e}_0 = e_2 \bar{e}_1 e_0 + e_1 \bar{e}_0$$
$$v_{0d} = \bar{e}_2 e_1 \bar{e}_0 + e_2 \bar{e}_1 e_0 + e_2 e_1 \bar{e}_0 + e_2 e_1 e_0 = e_2 e_0 + e_1 \bar{e}_0$$

Para v_1: obtenga todas las posibles funciones que representa la tabla de verdad con indiferencias.

e_2	e_1	e_0	v_{1d}	v_{1c}	v_{1b}	v_{1a}
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	0	0
1	1	1	1	0	1	0

$$v_{1a} = \bar{e}_2 e_1 e_0 + e_2 \bar{e}_1 \bar{e}_0 + e_2 \bar{e}_1 e_0 = \bar{e}_2 e_1 e_0 + e_2 \bar{e}_1$$
$$v_{1b} = \bar{e}_2 e_1 e_0 + e_2 \bar{e}_1 \bar{e}_0 + e_2 \bar{e}_1 e_0 + e_2 e_1 e_0 = e_1 e_0 + e_2 \bar{e}_1$$
$$v_{1c} = \bar{e}_2 e_1 e_0 + e_2 \bar{e}_1 \bar{e}_0 + e_2 \bar{e}_1 e_0 + e_2 e_1 \bar{e}_0 = e_2 \bar{e}_1 + \bar{e}_2 e_1 e_0 + e_2 \bar{e}_0$$
$$v_{1d} = \bar{e}_2 e_1 e_0 + e_2 \bar{e}_1 \bar{e}_0 + e_2 \bar{e}_1 e_0 + e_2 e_1 \bar{e}_0 + e_2 e_1 e_0 = e_2 + e_1 e_0$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

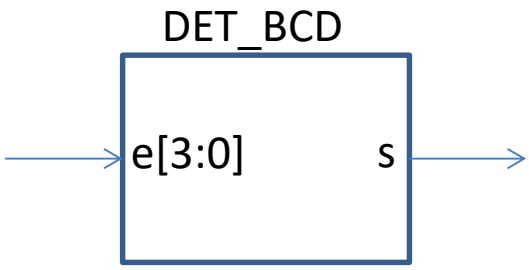
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Defina nuevamente la tabla de verdad escogiendo para las salidas indiferentes los valores que hacen que las funciones que se obtienen sean las más simples.

e_2	e_1	e_0	v_1	v_0
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Se ha escogido la función v_{0d} para v_0.

Ejercicio 2: Obtenga la tabla de verdad de un sistema que detecte, con un '1' lógico, si determinado número BCD es mayor o igual a 6. El comportamiento del circuito para números que no sean BCD es indiferente; en estos casos la salida debe ser 'X'.



e_3	e_2	e_1	e_0	s
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Obtenga razonadamente el número de posibles funciones que representa la tabla de verdad que ha obtenido anteriormente.

El número de posibles funciones es 64 (2^6).

Obtenga una función booleana para la salida del circuito. Para ello, seleccione previamente un conjunto de valores para las combinaciones indiferentes y anote cada valor al lado de su correspondiente 'X' en la tabla de verdad. Realice esta selección de valores buscando que la función de salida sea simple, pero limitando el esfuerzo empleado en esa búsqueda a un máximo de cinco minutos.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

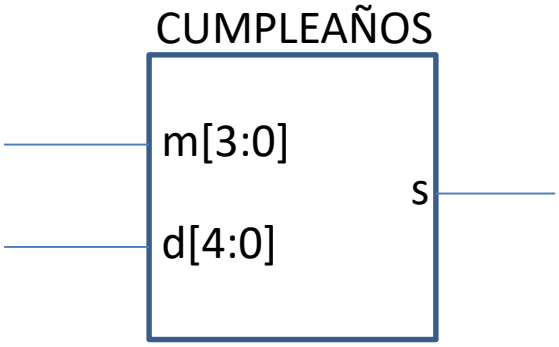
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

$$s = \bar{e}_3 e_2 e_1 \bar{e}_0 + \bar{e}_3 e_2 e_1 e_0 + e_3 \bar{e}_2 \bar{e}_1 \bar{e}_0 + e_3 \bar{e}_2 \bar{e}_1 e_0 = \bar{e}_3 e_2 e_1 + e_3 \bar{e}_2 \bar{e}_1$$

Indique cuál es el criterio que se ha utilizado para seleccionar los valores para las combinaciones indiferentes.

Obtener el mínimo número de minitérminos.

Ejercicio 3: Realice un sistema que detecte cuándo es el cumpleaños de cada miembro de una familia de 3. Las fechas de los cumpleaños son: 03 de mayo de 1968, 11 de julio de 1972 y 25 de octubre de 2001.



Obtenga, de forma razonada, el número de filas que tiene la tabla de verdad completa correspondiente a este circuito.

El número de filas es 512. El número de filas de una tabla de verdad de N entradas es 2^N. En este caso N=9.

Obtenga la tabla de verdad resumida.

d_4	d_3	d_2	d_1	d_0	m_3	m_2	m_1	m_0	s
0	0	0	1	1	0	1	0	1	1
0	1	0	1	1	0	1	1	1	1
1	1	0	0	1	1	0	1	0	1
Otras combinaciones									0

Obtenga la función booleana correspondiente a la salida del circuito.

$$s = \bar{d}_4 \bar{d}_3 \bar{d}_2 d_1 d_0 \bar{m}_3 m_2 \bar{m}_1 m_0 + \bar{d}_4 d_3 \bar{d}_2 d_1 d_0 \bar{m}_3 m_2 m_1 m_0 + d_4 d_3 \bar{d}_2 \bar{d}_1 d_0 m_3 \bar{m}_2 m_1 \bar{m}_0$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios de caracterización, síntesis, simplificación y representación con puertas lógicas de sistemas combinacionales.**	**CÓDIGO:** **BT1_AINP18**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	60 minutos

CALENDARIO:	Novena Sesión. Antes de la Decima sesión	REQUISITOS:	Metodología de síntesis de sistemas combinacionales. Simplificación. Tablas de verdad resumidas y con indiferencias. Puertas lógicas

CRITERIO DE ÉXITO:	Se han realizado bien tres de los cinco ejercicios

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 1: Luis, Irene, Ana y Miguel están dudando si ir al cine o al teatro. Para tomar una decisión acuerdan votar y que, en caso de empate, el voto de Miguel sea decisivo. Se utiliza la siguiente codificación para los valores de las variables: si se vota por ir al cine, la variable toma valor el 0; en caso contrario, la variable toma el valor 1. Análogamente, la función Cine/Teatro valdrá 0 si la decisión es ir al cine y 1 si la decisión es ir al teatro.

Realice la tabla de verdad que define el problema anterior.

Obtenga la salida como función booleana de las entradas.

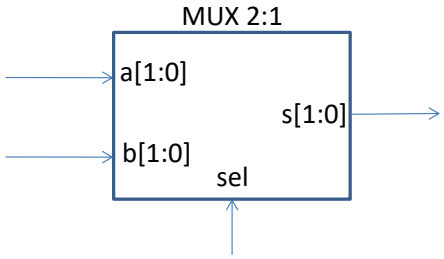
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 2: Diseñe un multiplexor de 2 canales (es decir, de 2 entradas de datos), en el cual cada entrada esté representada por 2 bits. Cuando 'sel' es 0 la salida 's' toma el valor de la entrada 'a'. Obtenga en primer lugar la tabla de verdad resumida:



sel	s_1	s_0
0		
1		

Obtenga las funciones booleanas para s1 y s0.

Finalmente, dibuje el circuito del multiplexor con puertas lógicas.

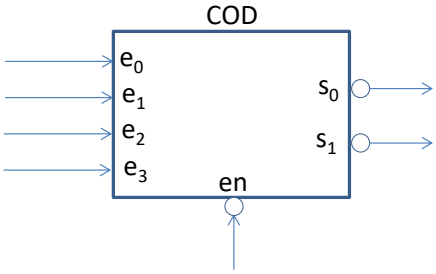
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 3: Diseñe un codificador de 4:2 con prioridad con entrada de habilitación y salidas activas a nivel bajo. Las entradas más prioritarias son las de subíndice más alto.



Obtenga en primer lugar la tabla de verdad resumida del circuito:

Obtenga ahora las expresiones booleanas de las salidas en función de las entradas.

Finalmente, dibuje el circuito con puertas lógicas.

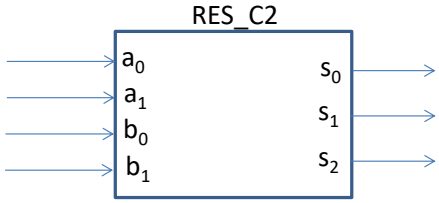
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 4: Diseñe un circuito restador de 2 bits en complemento a 2 (s=a-b, tanto las entradas como la salida codificadas en C2).



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 5: Diseñe un circuito comparador de 2 entradas, ‘a’ y ‘b’, de 2 bits en binario natural, que active, a nivel alto, su salida ‘aleb’ cuando la entrada ‘a’ sea menor o igual que la entrada ‘b’.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones de los ejercicios.

A continuación se detallan las soluciones a los 8 ejercicios propuestos en esta actividad. No mire las soluciones hasta que no haya resuelto los ejercicios.

Ejercicio 1: Luis, Irene, Ana y Miguel están dudando si ir al cine o al teatro. Para tomar una decisión acuerdan votar y que, en caso de empate, el voto de Miguel sea decisivo. Se utiliza la siguiente codificación para los valores de las variables: si se vota por ir al cine, la variable toma valor el 0; en caso contrario, la variable toma el valor 1. Análogamente, la función Cine/Teatro valdrá 0 si la decisión es ir al cine y 1 si la decisión es ir al teatro.

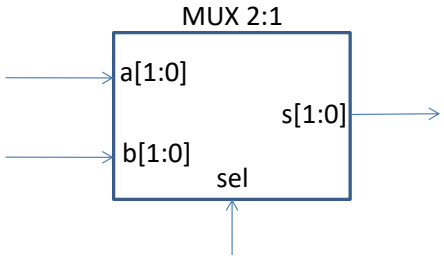
Realice la tabla de verdad que define el problema anterior.

L	*I*	*A*	*M*	*S*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Obtenga la salida como función booleana de las entradas.

$$s = AM + I\overline{A}M + L\overline{A}M + LIA$$

Ejercicio 2: Diseñe un multiplexor de 2 canales (es decir, de 2 entradas de datos), en el cual cada entrada esté representada por 2 bits. Obtenga en primer lugar la tabla de verdad resumida:



sel	s_1	s_0
0	a_1	a_0
1	b_1	b_0

Obtenga las funciones booleanas para s1 y s0.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

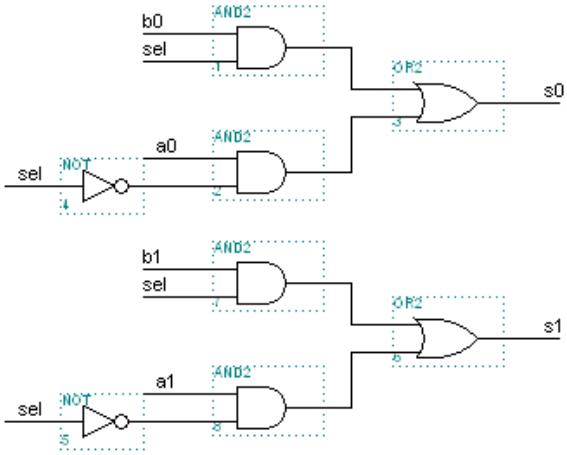
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

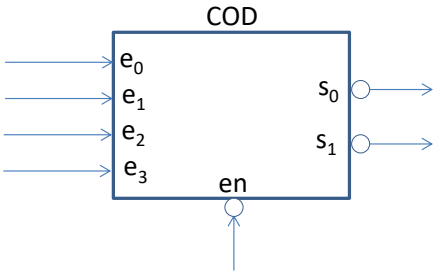
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

$$s_0 = \overline{sel}\, a_0 + sel\, b_0$$
$$s_1 = \overline{sel}\, a_1 + sel\, b_1$$

Finalmente, dibuje el circuito del multiplexor con puertas lógicas.



Ejercicio 3: Diseñe un codificador de 4:2 con prioridad con entrada de habilitación y salidas activas a nivel bajo. Las entradas más prioritarias son las de subíndice más alto.



Obtenga en primer lugar la tabla de verdad resumida del circuito:

e_3	e_2	e_1	e_0	en	s_1	s_0
X	X	X	X	1	1	1
0	0	0	1	0	1	1
0	0	1	X	0	1	0
0	1	X	X	0	0	1
1	X	X	X	0	0	0

Obtenga ahora las expresiones booleanas de las salidas en función de las entradas.

$$s_0 = en + \overline{en}\, e_0\, \overline{e_1}\, \overline{e_2}\, \overline{e_3} + \overline{en}\, e_2\, \overline{e_3}$$
$$s_1 = en + \overline{en}\, e_0\, \overline{e_1}\, \overline{e_2}\, \overline{e_3} + \overline{en}\, e_1\, \overline{e_2}\, \overline{e_3}$$

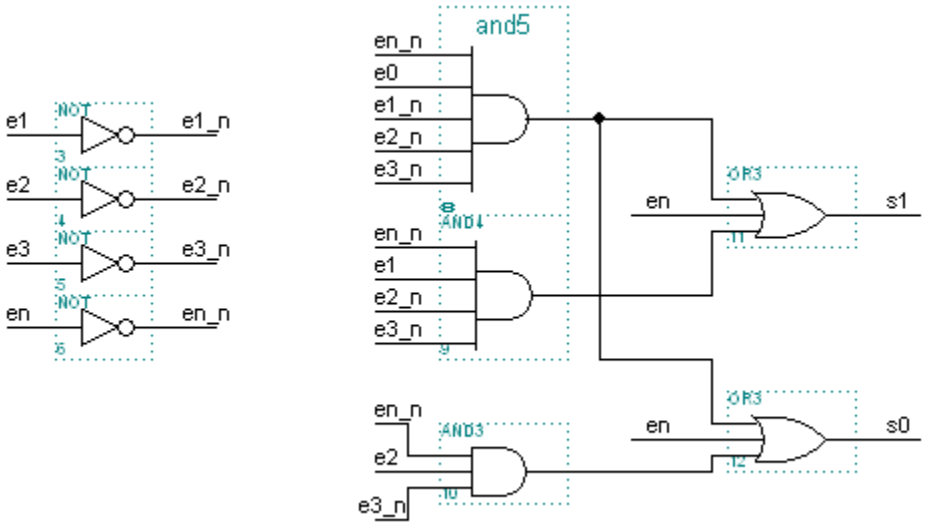
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

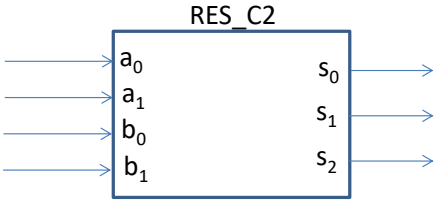
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Finalmente, dibuje el circuito con puertas lógicas.



Ejercicio 4: Diseñe un circuito restador de 2 bits en complemento a 2 (s=a-b, tanto las entradas como la salida codificadas en C2).



La tabla de verdad del circuito es la siguiente:

b_1	b_0	a_1	a_0	s_2	s_1	s_0	explicación
0	0	0	0	0	0	0	0-0=0
0	0	0	1	0	0	1	1-0=1
0	0	1	0	1	1	0	-2-0=-2
0	0	1	1	1	1	1	-1-0=-1
0	1	0	0	1	1	1	0-1=-1
0	1	0	1	0	0	0	1-1=0
0	1	1	0	1	0	1	-2-1=-3
0	1	1	1	1	1	0	-1-1=-2
1	0	0	0	0	1	0	0-(-2)=2
1	0	0	1	0	1	1	1-(-2)=3
1	0	1	0	0	0	0	-2-(-2)=0
1	0	1	1	0	0	1	-1-(-2)=1
1	1	0	0	0	0	1	0-(-1)=1
1	1	0	1	0	1	0	1-(-1)=2
1	1	1	0	1	1	1	-2-(-1)=-1
1	1	1	1	0	0	0	-1-(-1)=0

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

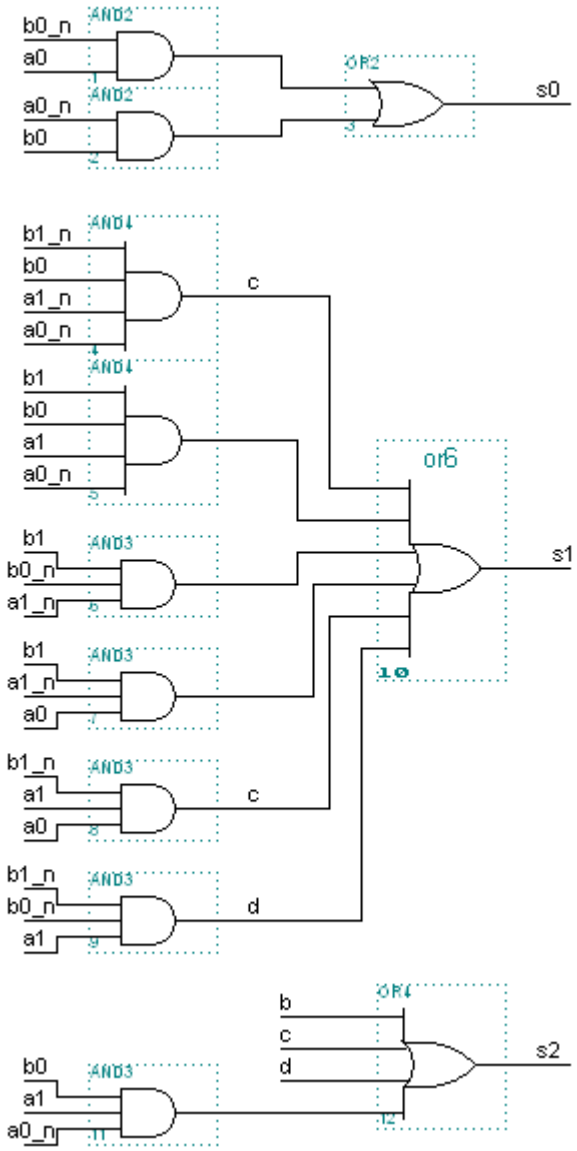
A partir de la tabla de verdad se obtienen las siguientes funciones como suma de minitérminos:

$$s_0 = \sum m(1,3,4,6,9,11,12,14) = \bar{b}_1\,\bar{b}_0\,\bar{a}_1\,a_0 + \bar{b}_1\,\bar{b}_0\,a_1\,a_0 + \bar{b}_1\,b_0\,\bar{a}_1\,\bar{a}_0 + \bar{b}_1\,b_0\,a_1\,\bar{a}_0 + b_1\,\bar{b}_0\,\bar{a}_1\,a_0 + b_1\,\bar{b}_0\,a_1\,a_0$$
$$+ b_1\,b_0\,\bar{a}_1\,\bar{a}_0 + b_1\,b_0\,a_1\,\bar{a}_0 = \bar{b}_0\;a_0 + b_0\,\bar{a}_0$$

$$s_1 = \sum m(2,3,4,7,8,9,13,14) = \bar{b}_1\,\bar{b}_0\,a_1\,\bar{a}_0 + \bar{b}_1\,\bar{b}_0\,a_1\,a_0 + \bar{b}_1\,b_0\,\bar{a}_1\,\bar{a}_0 + \bar{b}_1\,b_0\,a_1\,a_0 + b_1\,\bar{b}_0\,\bar{a}_1\,\bar{a}_0 + b_1\,\bar{b}_0\,\bar{a}_1\,a_0$$
$$+ b_1\,b_0\,\bar{a}_1\,a_0 + b_1\,b_0\,a_1\,\bar{a}_0 = \bar{b}_1\,b_0\,\bar{a}_1\,\bar{a}_0 + b_1\,b_0\,a_1\,\bar{a}_0 + b_1\,\bar{b}_0\,\bar{a}_1 + b_1\;\bar{a}_1\,a_0 + \bar{b}_1\;a_1\,a_0 + \bar{b}_1\,\bar{b}_0\,a_1$$

$$s_2 = \sum m(2,3,4,6,7,14) = \bar{b}_1\,\bar{b}_0\,a_1\,\bar{a}_0 + \bar{b}_1\,\bar{b}_0\,a_1\,a_0 + \bar{b}_1\,b_0\,\bar{a}_1\,\bar{a}_0 + \bar{b}_1\,b_0\,a_1\,\bar{a}_0 + \bar{b}_1\,b_0\,a_1\,a_0 + b_1\,b_0\,a_1\,\bar{a}_0$$
$$= \bar{b}_1\,b_0\,\bar{a}_1\,\bar{a}_0 + \bar{b}_1\;a_1\,a_0 + \bar{b}_1\,\bar{b}_0\,a_1 + b_0\,a_1\,\bar{a}_0$$

A partir de las funciones puede obtenerse una representación con puertas lógicas:



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

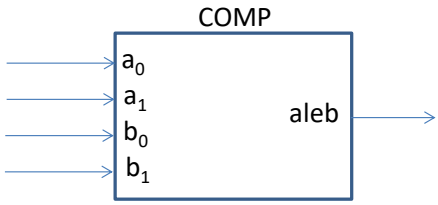
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 5: Diseñe un circuito comparador de 2 entradas, 'a' y 'b', de 2 bits, que active su salida 'aleb' cuando la entrada 'a' sea menor o igual que la entrada 'b'.

El símbolo y la tabla de verdad del circuito son los siguientes:

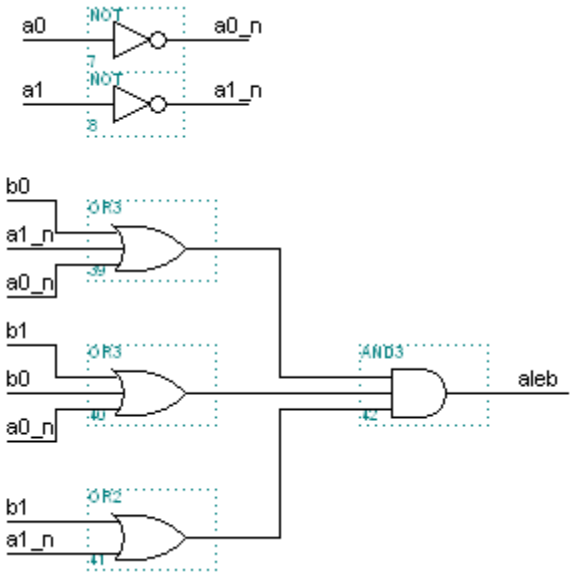


b_1	b_0	a_1	a_0	aleb
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

A partir de la tabla de verdad puede obtenerse la siguiente expresión booleana, como producto de maxitérminos:

$$aleb = \prod M(1,2,3,6,7,11) = (b_1 + b_0 + a_1 + \bar{a}_0)(b_1 + b_0 + \bar{a}_1 + a_0)(b_1 + b_0 + \bar{a}_1 + \bar{a}_0)(b_1 + \bar{b}_0 + \bar{a}_1 + a_0)$$
$$(b_1 + \bar{b}_0 + \bar{a}_1 + \bar{a}_0)(\bar{b}_1 + b_0 + \bar{a}_1 + \bar{a}_0) = (b_1 + \bar{a}_1)(b_0 + \bar{a}_1 + \bar{a}_0)(b_1 + b_0 + \bar{a}_0)$$

A partir de la expresión booleana puede obtenerse una representación con puertas lógicas:



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

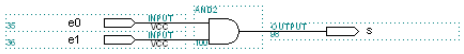
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

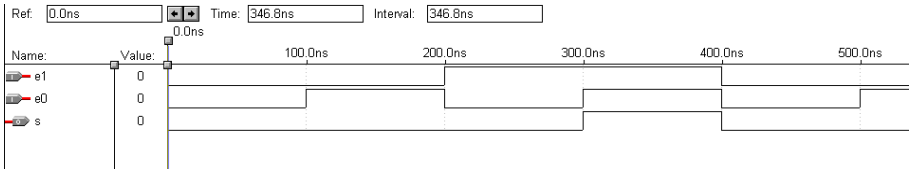
Electrónica 2
Bloque Temático 1

Representación mediante cronogramas del funcionamiento de un sistema combinacional. Representación de buses, indiferencias e indeterminaciones. Realización y análisis de cronogramas

Cronogramas

- Un cronograma es un diagrama en el que se representa la evolución con el tiempo de entradas, salidas o nodos intermedios de un circuito





BT1_S10 BT1_P19_2

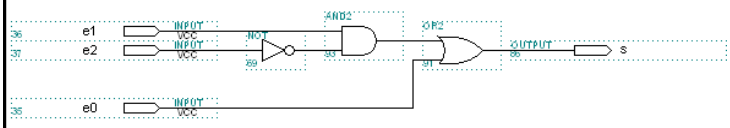
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Realización de cronogramas



e_2	e_1	e_0	s
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

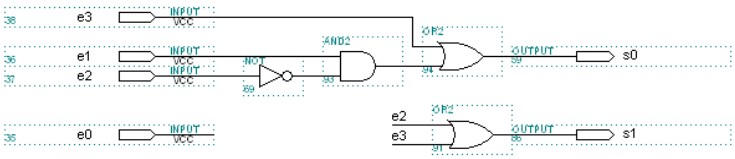
Estos son los nodos que se van a visualizar (en este caso, solo las entradas y la salida del circuito)

Esto son los estímulos. Son los estados por los que van evolucionando las entradas del circuito en función del tiempo

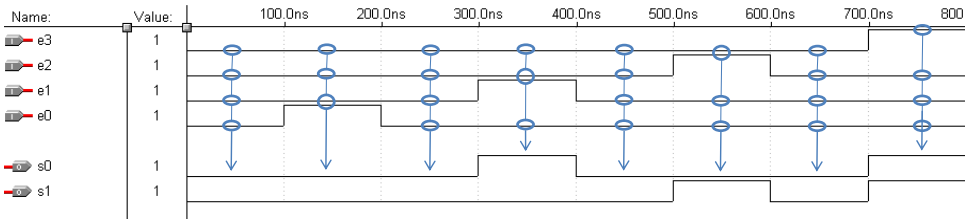
Y esta es la salida del circuito, cuya evolución en el tiempo depende de cómo evolucionen las entradas

BT1_S10 BT1_P19_3

Realización de cronogramas



Codificador 4:2 con prioridad



BT1_S10 BT1_P19_4

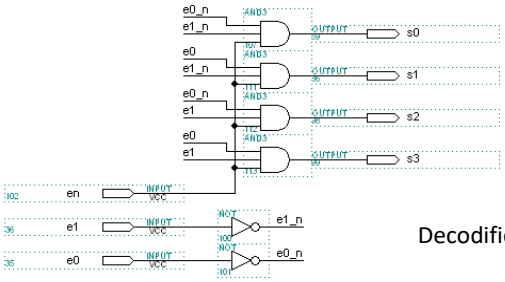
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

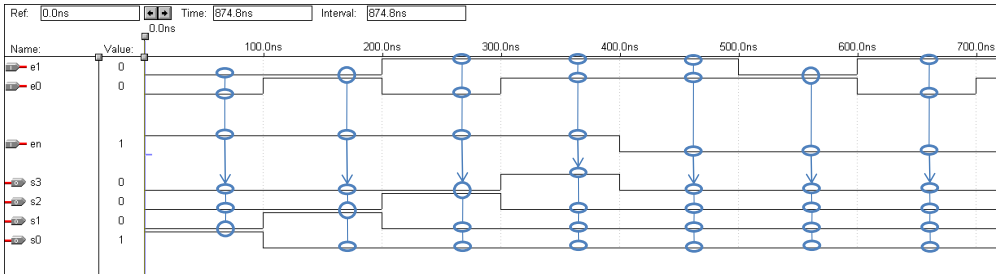
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Realización de cronogramas



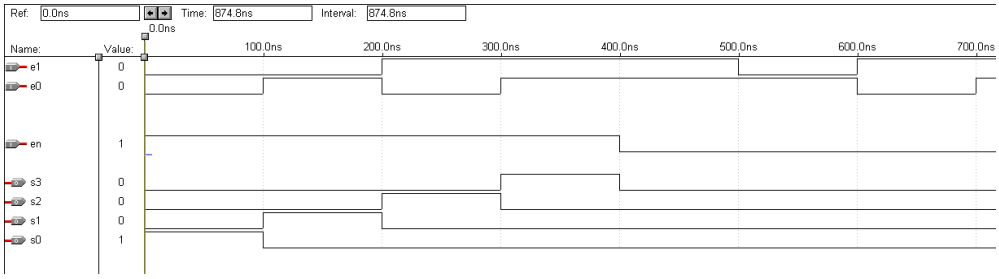
Decodificador 2:4 con *enable*

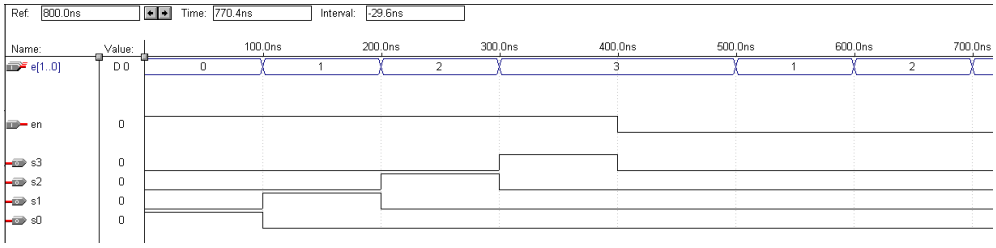


BT1_S10 BT1_P19_5

Buses en cronogramas

Decodificador 2:4 con *enable*





BT1_S10 BT1_P19_6

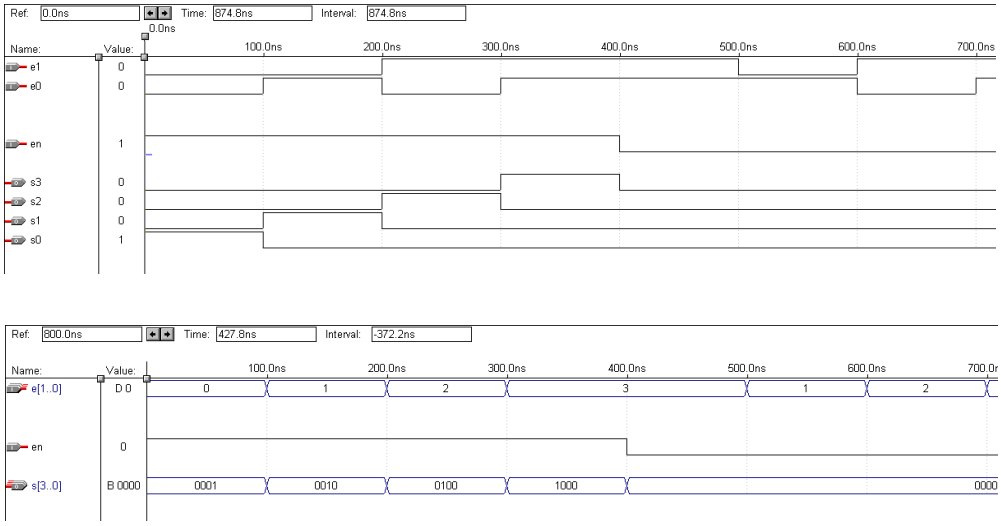
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Buses en cronogramas

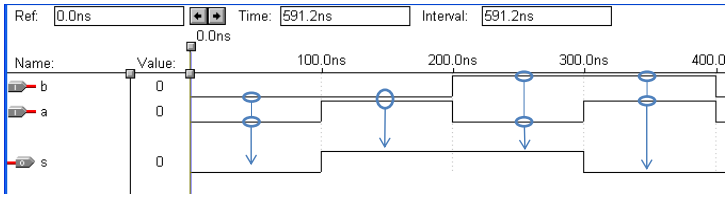
Decodificador 2:4 con *enable*



BT1_S10 BT1_P19_7

Análisis de cronogramas

- A partir de un cronograma es posible obtener la tabla de verdad del circuito siempre que estén representadas todas las posibles combinaciones de entrada:



b	a	s
0	0	0
0	1	1
1	0	1
1	1	0

BT1_S10 BT1_P19_8

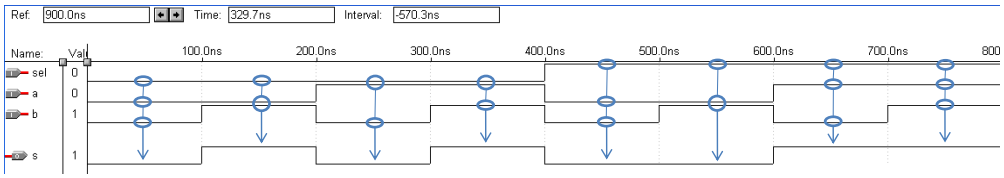
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Análisis de cronogramas

- Ejemplo: A partir del siguiente cronograma, construya la tabla de verdad del circuito.

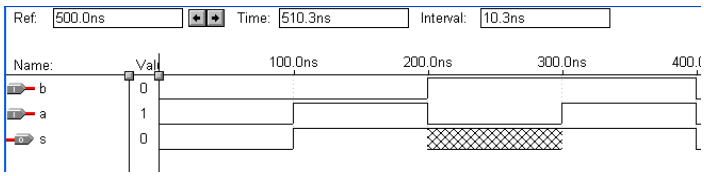


sel	a	b	s
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

BT1_S10 BT1_P19_9

Análisis de cronogramas

- Análogamente a lo que ocurre en las tablas de verdad, en los cronogramas también pueden aparecer condiciones de indiferencia:



b	a	s
0	0	0
0	1	1
1	0	X
1	1	1

BT1_S10 BT1_P19_10

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre cronogramas de circuitos combinacionales.**	**CÓDIGO:** **BT1_ATGP5**

GRUPO		FECHA:	

MODALIDAD:	Aprendizaje Cooperativo	TIPO:	Presencial	DURACIÓN:	35 minutos
CALENDARIO:	Decima Sesión	REQUISITOS:	Metodología de síntesis de circuitos combinacionales. Puertas lógicas. Cronogramas		

CRITERIO DE ÉXITO:	Al menos cinco de los seis ejercicios bien hechos.
INTERDEPENDENCIA POSITIVA:	En el tiempo disponible no pueden completar todos los ejercicios si no trabajan todos los miembros del grupo.
EXIGIBILIDAD INDIVIDUAL:	Al finalizar la actividad uno de los miembros del grupo, elegido al azar, deberá realizar una pequeña prueba acerca de los contenidos de la actividad, y esta nota será asignada a todos los miembros del grupo.
ROLES:	*Secretarios*: dos miembros toman nota de las decisiones alcanzadas y rellenan las hojas de resultados. *Controlador de Tiempo*: controla el tiempo que queda para la finalización de la tarea. *Moderador*: dirige la discusión, controla los turnos de palabra y garantiza la participación por igual de todos los miembros.
HABILIDADES SOCIALES:	Capacidad para expresar y sintetizar ideas, capacidad para escuchar, asimilar y organizar la información presentada por otra persona. Capacidad para razonar y analizar.
REFLEXIÓN DE GRUPO:	Se deberá analizar el grado de participación de cada miembro del grupo y su capacidad como grupo para mantener un debate fluido que lleve a una toma de decisiones eficaz y compartida por todos

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	No procede

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

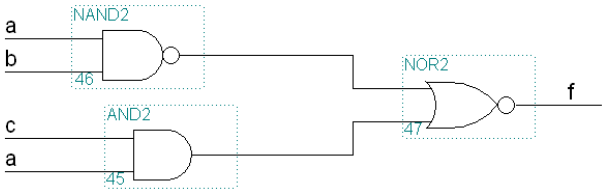
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

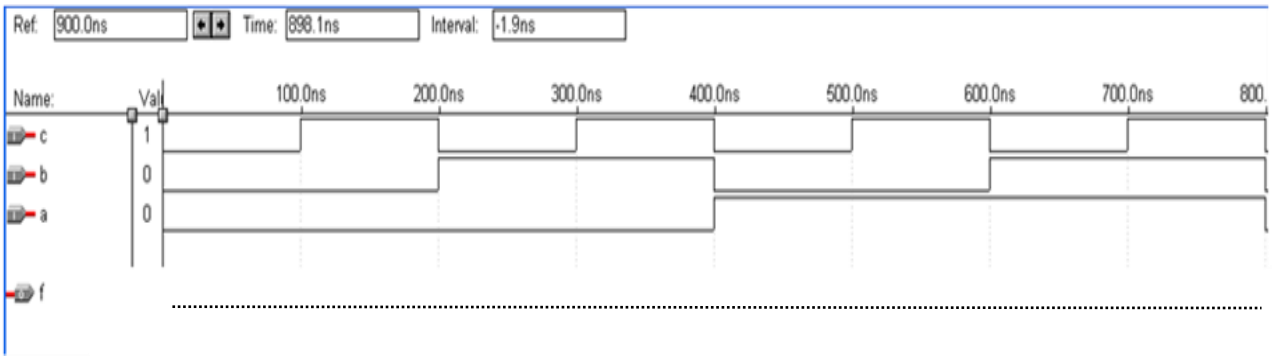
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

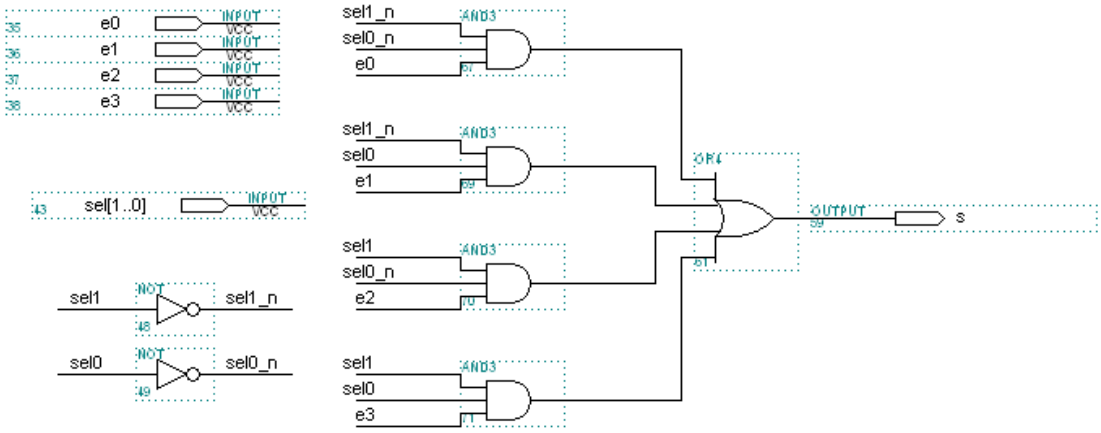
Los miembros del grupo, trabajando de manera conjunta, resolverán los seis ejercicios que se proponen. Finalizada la actividad, uno de los miembros del grupo (el que el profesor indique) realizará una prueba breve acerca de los contenidos trabajados. La nota, que se aplicará a todos los miembros del grupo, servirá para calcular la calificación global de la actividad, junto con el entregable realizado por el grupo.

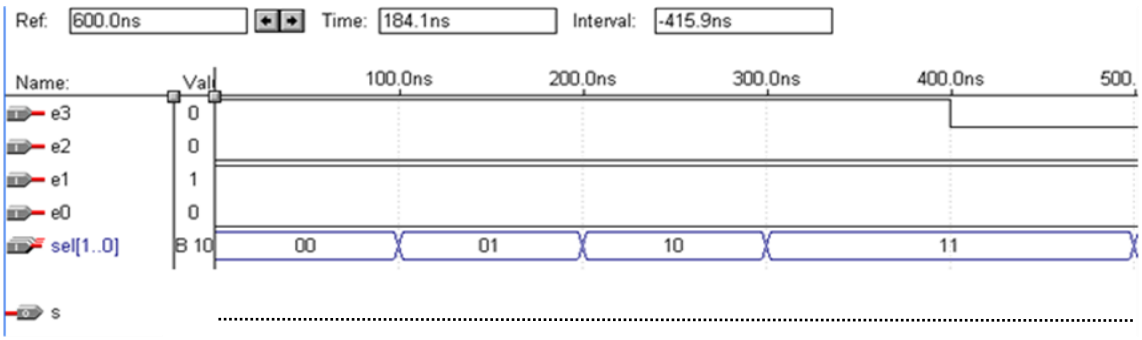
EJERCICIO 1) Complete el siguiente cronograma, que corresponde al circuito de la figura.





EJERCICIO 2) Complete el siguiente cronograma, que corresponde al circuito de la figura.





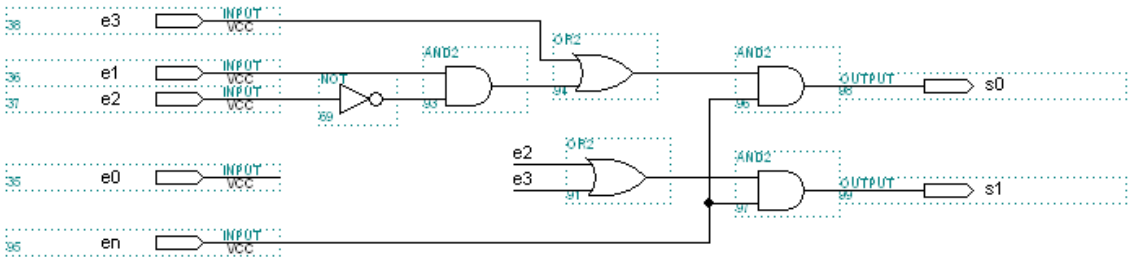
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

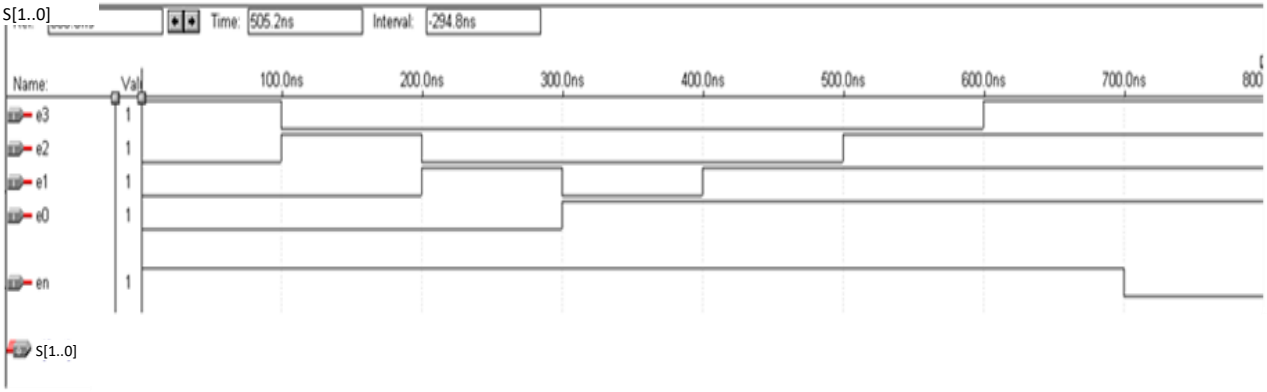
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

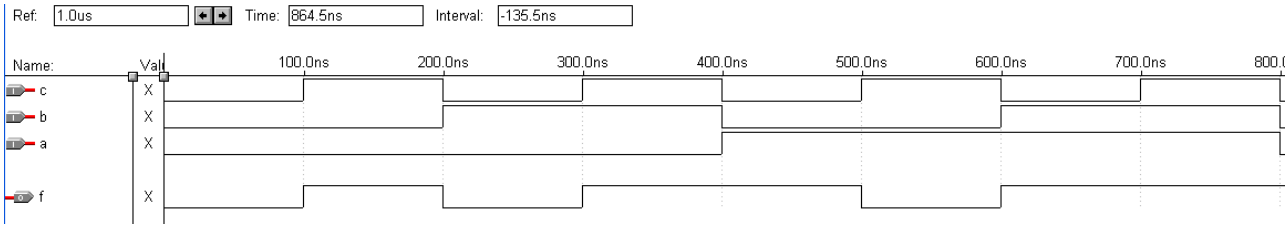
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 3) Complete el siguiente cronograma, que corresponde al circuito de la figura.





EJERCICIO 4) Obtenga la tabla de verdad del circuito que responde al siguiente cronograma.



a	b	c	f

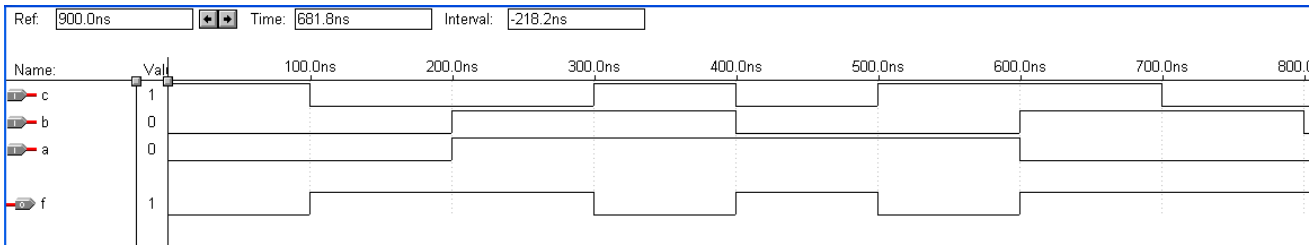
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

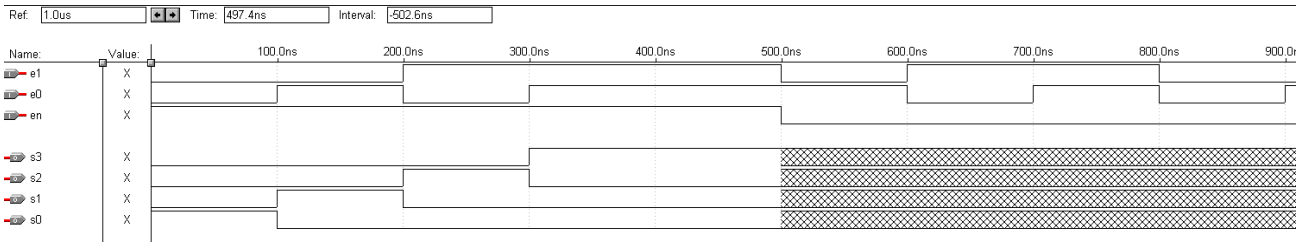
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 5) Obtenga la tabla de verdad del circuito que responde al siguiente cronograma.



a	b	c	f

EJERCICIO 6) Obtenga la tabla de verdad del circuito que responde al siguiente cronograma.



en	e_1	e_0	s_3	s_2	s_1	s_0

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

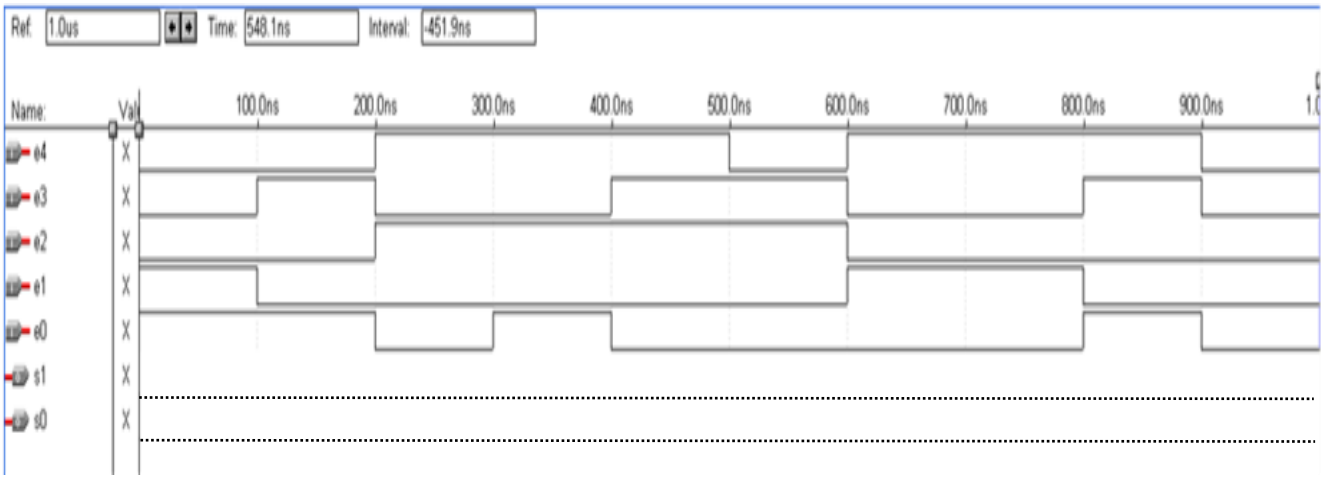
BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Test final sobre síntesis de sistemas combinacionales simples.**	**CÓDIGO:** **BT1_ATGP5**

FECHA:			
NOMBRE:		APELLIDOS:	
GRUPO		CALIFICACIÓN: [entre 0 y 10 puntos]	puntos

Complete el cronograma de la figura, supuesto que el circuito tiene la siguiente tabla de verdad.

e_4	e_3	e_2	e_1	e_0	s_1	s_0
0	X	X	X	X	0	0
1	0	0	X	X	1	0
1	0	1	X	X	0	1
1	1	0	X	X	1	1
1	1	1	X	X	X	X



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre análisis y diseño de circuitos combinacionales.**	**CÓDIGO:** **BT1_ATGP6**

GRUPO		FECHA:	

MODALIDAD:	Aprendizaje en Grupo	TIPO:	Presencial	DURACIÓN:	50 minutos
CALENDARIO:	Decima Sesión.	REQUISITOS:	Metodología de síntesis de circuitos combinacionales. Puertas lógicas.		

CRITERIO DE ÉXITO:	Al menos tres de los cuatro ejercicios bien hechos.
INTERDEPENDENCIA POSITIVA:	En el tiempo disponible no pueden completar todos los ejercicios si no trabajan todos los miembros del grupo.
EXIGIBILIDAD INDIVIDUAL:	No procede.
ROLES:	*Secretarios*: dos miembros toman nota de las decisiones alcanzadas y rellenan las hojas de resultados. *Controlador de Tiempo*: controla el tiempo que queda para la finalización de la tarea. *Moderador*: dirige la discusión, controla los turnos de palabra y garantiza la participación por igual de todos los miembros.
HABILIDADES SOCIALES:	Capacidad para expresar y sintetizar ideas, capacidad para escuchar, asimilar y organizar la información presentada por otra persona. Capacidad para razonar y analizar.
REFLEXIÓN DE GRUPO:	Se deberá analizar el grado de participación de cada miembro del grupo y su capacidad como grupo para mantener un debate fluido que lleve a una toma de decisiones eficaz y compartida por todos

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	No procede

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 1)

Un sistema combinacional dispone de una entrada E de 6 bits. El circuito debe detectar cuando, en la combinación de entrada, existen dos o más unos adyacentes. Diseñe el sistema incluyendo su representación con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 2)

Un circuito combinacional se encarga de informar del consumo de un sistema en función de la información que le llega por seis señales C5, C4, C3, C2, C1 y C0, de un bit cada una. Cada señal informa si existe o no existe consumo en distintas partes del sistema. La activación de C0 supone un consumo de 5 mA, la de C1 de 10 mA, la de C2 de 15 mA, la de C3 de 20 mA, la de C4 de 25mA y la de C5 de 30mA. El circuito tendrá una salida activa a nivel alto: ALERTA, que se activará cuando el consumo total del sistema sea mayor que 80mA. Obtenga su tabla de verdad. Diseñe el sistema incluyendo su representación con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 3)

Un sistema recibe un número en binario natural de 8 bits X[7..0] y debe activar su salida XLK cuando ese número sea estrictamente menor que 1010 0010. Diseñe el sistema incluyendo su representación con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 4)

Un sistema selecciona, mediante una señal de selección C de 3 bits (C[2..0]) la información que llega por 8 canales de un bit (E[7..0]). Dicha información se lleva a una salida de un bit, S. La información se selecciona de la siguiente forma:

- S = 'Ei', si i es par.
- S = '/Ei', si i es impar.

Diseñe el sistema incluyendo su representación con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre síntesis, representación con puertas lógicas y obtención de cronogramas de sistemas combinacionales.**	**CÓDIGO:** **BT1_AINP19**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Ejercicios. Individual	TIPO:	No presencial	DURACIÓN:	120 minutos

CALENDARIO:	Decima Sesión. Previo a la Undécima Sesión.	REQUISITOS:	Metodología de síntesis de sistemas combinacionales. Puertas lógicas. Cronogramas.

CRITERIO DE ÉXITO:	Realización correcta de cuatro de los seis ejercicios

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	

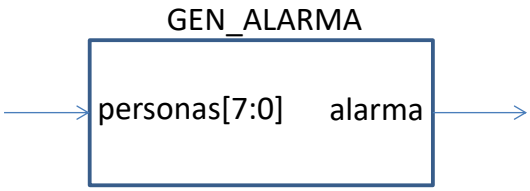
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 1: Alarma. El sistema tiene como entrada el número de personas en un local (binario natural, 8 bits) y genera una salida de alarma (a 1 lógico) cuando este número supera los 150.

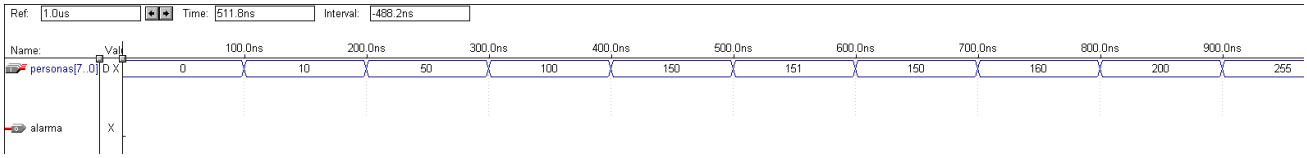


Obtenga la tabla de verdad del sistema.

Obtenga la expresión booleana que relaciona la salida con las entradas.

Dibuje el circuito con puertas lógicas.

Complete el siguiente cronograma de funcionamiento:



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 2: Recomendador de operador de telefonía. Un circuito recomendador de operador de telefonía dispone de la información acerca de la duración esperada de su próxima llamada (menos de 1 minuto, entre uno y 5 minutos, más de 5 minutos), del momento del día (mañana, tarde o noche) y del tipo de llamada (local, nacional o internacional). A partir de esa información, recomienda el operador A para todas las llamadas locales, así como para las nacionales, siempre que la llamada no exceda de 1 minuto o bien supere los 5 minutos y (solo en este último caso) sea por la tarde. El operador B se recomienda para todas las llamadas internacionales, así como para las nacionales superiores a 1 minuto por la mañana o superiores a 5 minutos por la noche. Para el resto de llamadas el sistema recomienda el operador C.

Obtenga el símbolo del sistema y realice la codificación de las entradas. Utilice una salida para cada operador.

Obtenga la tabla de verdad del sistema y las ecuaciones de las salidas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

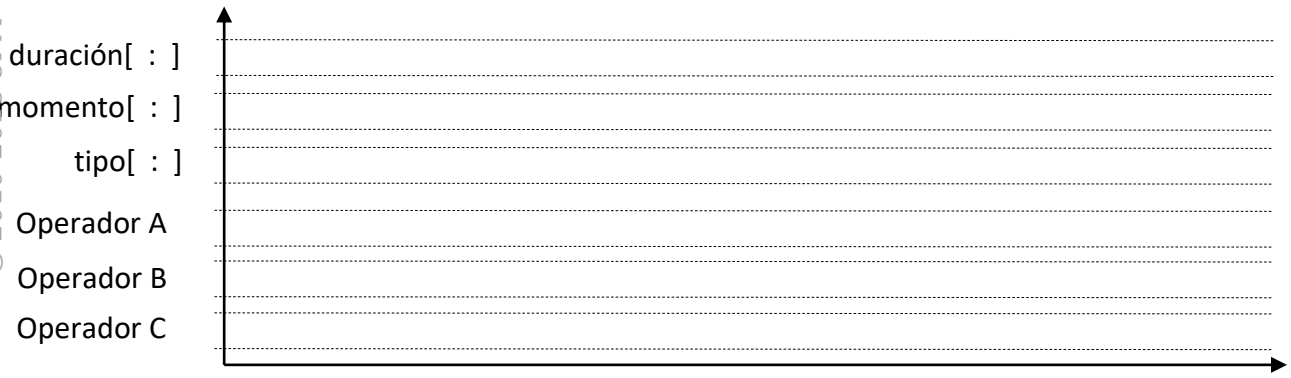
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Represente el circuito del recomendador con puertas lógicas.

Dibuje un cronograma de funcionamiento en el que se comprueben todas las combinaciones posibles de duración de la llamada y momento del día sólo para las llamadas locales. Utilice buses para representar cada tipo de entrada y señales individuales para las salidas. Complete los tamaños de los buses en el cronograma.



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

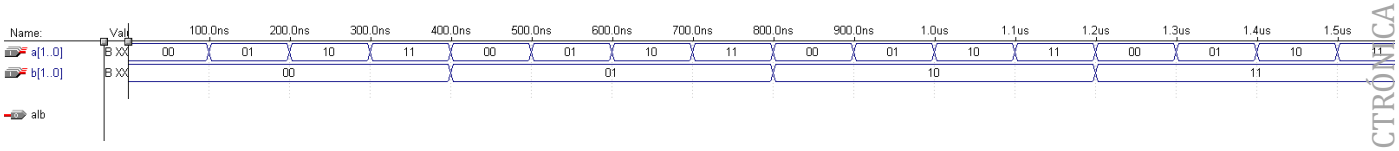
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 3: Comparador de 2 bits en C2. Diseñe un comparador de 2 bits en código complemento a 2. Este circuito comparará dos entradas, ‘a’ y ‘b’, codificadas en C2 y activará su salida, ‘alb’, si la entrada ‘a’ es menor que la entrada ‘b’.

Dibuje el símbolo del sistema y obtenga su tabla de verdad.

Complete el siguiente cronograma:



Obtenga la expresión booleana de la salida ‘alb’ en función de las entradas.

Dibuje una representación del circuito con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

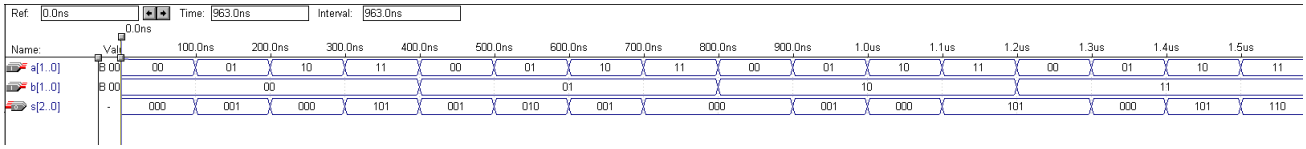
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 4: Sumador de 2 bits en código valor absoluto y signo (VAS).

En el código VAS, cada número se codifica con N bits para la magnitud y un bit de signo. Los números positivos siempre tienen como bit de signo el '0' mientras que los negativos tienen como bit de signo el '1'. Por ejemplo, con 5 bits pueden codificarse números entre el 15_d (01111) y el -15_d (11111). El 8_d se codificaría como '01000' y el -2_d se codificaría como '10010'. El cero siempre puede codificarse de dos maneras diferentes (con 5 bits, '00000' y '10000'). Si solo dispusiéramos, por ejemplo, de 3 bits, todavía podríamos codificar el cero ('000' o '100') y el -2_d ('110'), pero no el 15_d, el -15_d o el 8_d.

Diseñe un sumador de 2 bits en el que las entradas y la salida estén codificadas en VAS. El circuito debe comportarse según el siguiente cronograma:



Rellene en primer lugar las siguientes tablas con los códigos VAS de los números enteros que pueden codificarse con 2 y 3 bits:

número	Código VAS	
	signo	magnitud
	b	a
+0		
-0		
+1		
-1		

número	Código VAS		
	signo	magnitud	
	c	b	a
+0			
-0			
+1			
-1			
+2			
-2			
+3			
-3			

Dibuje el símbolo del circuito y obtenga la tabla de verdad. Añada a la derecha de la tabla de verdad una columna donde se describa la operación que realiza el sumador para cada una de las filas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Obtenga la expresión booleana de las salidas y dibuje una representación del circuito con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

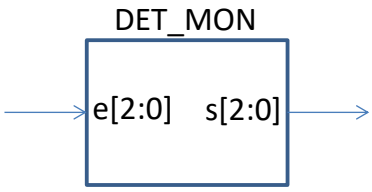
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 5: Detección de monedas. El sistema recibe la información del peso de una moneda en el rango entre 0 y 30 g. con una resolución de 5 g. y decide el tipo de moneda en función del peso. Si pesa menos de 5 g. es de 5 c., entre 5 g. y 9 g. es de 10 c., entre 10 g. y 19 g. es de 50 c., entre 20 g. Y 24 g. es de 1 €, a partir de 25 g. es de 2 €. Este sistema se ha diseñado de la siguiente forma:

En primer lugar, se ha establecido la codificación de las entradas y de las salidas:

Peso de la moneda	Entradas		
	e2	e1	e0
0 g.	0	0	0
5 g.	0	0	1
10 g.	0	1	0
15 g.	0	1	1
20 g.	1	0	0
25 g.	1	0	1
30 g.	1	1	0

Valor de la moneda	Salidas		
	s2	s1	s0
5 c.	0	0	0
10 c.	0	0	1
50 c.	0	1	0
1 €.	0	1	1
2 €.	1	0	0



Luego se ha obtenido la siguiente tabla de verdad:

e_2	e_1	e_0	s_2	s_1	s_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	X	X	X

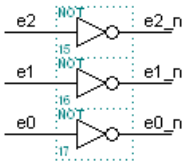
A partir de la tabla de verdad se han obtenido las expresiones de las salidas en función de las entradas (considerando que las indiferencias son '0') y una representación con puertas lógicas:

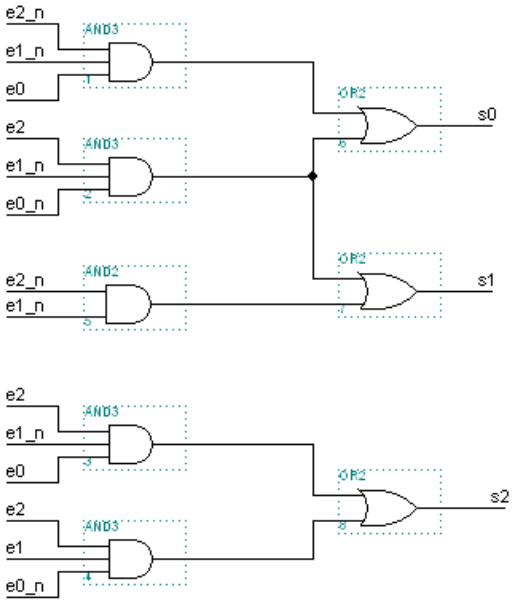
$$s_0 = \bar{e}_2\bar{e}_1e_0 + e_2\bar{e}_1\bar{e}_0$$

$$s_1 = \bar{e}_2e_1\bar{e}_0 + \bar{e}_2e_1e_0 + e_2\bar{e}_1\bar{e}_0$$

$$= \bar{e}_2\bar{e}_1 + e_2\bar{e}_1\bar{e}_0$$

$$s_2 = e_2\bar{e}_1e_0 + e_2e_1\bar{e}_0$$



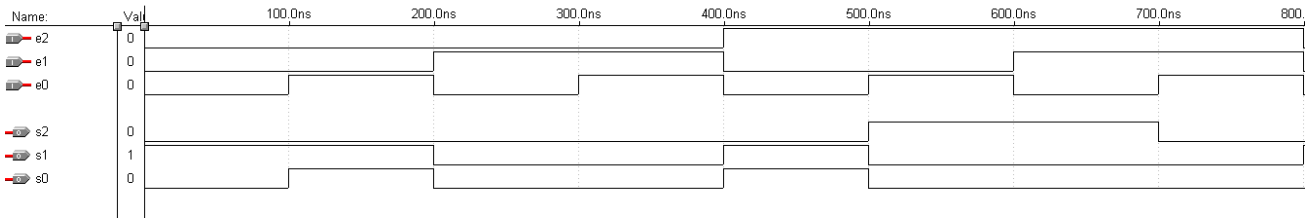


© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Finalmente, se ha obtenido el siguiente cronograma de funcionamiento mediante el procedimiento de dar valores a las entradas y obtener los valores de salida que generaría el circuito representado con puertas lógicas.



Revise con detalle el cronograma y compruebe si el funcionamiento se corresponde con el que se define en la tabla de verdad. Si observa algún error en el diseño, corríjalo.

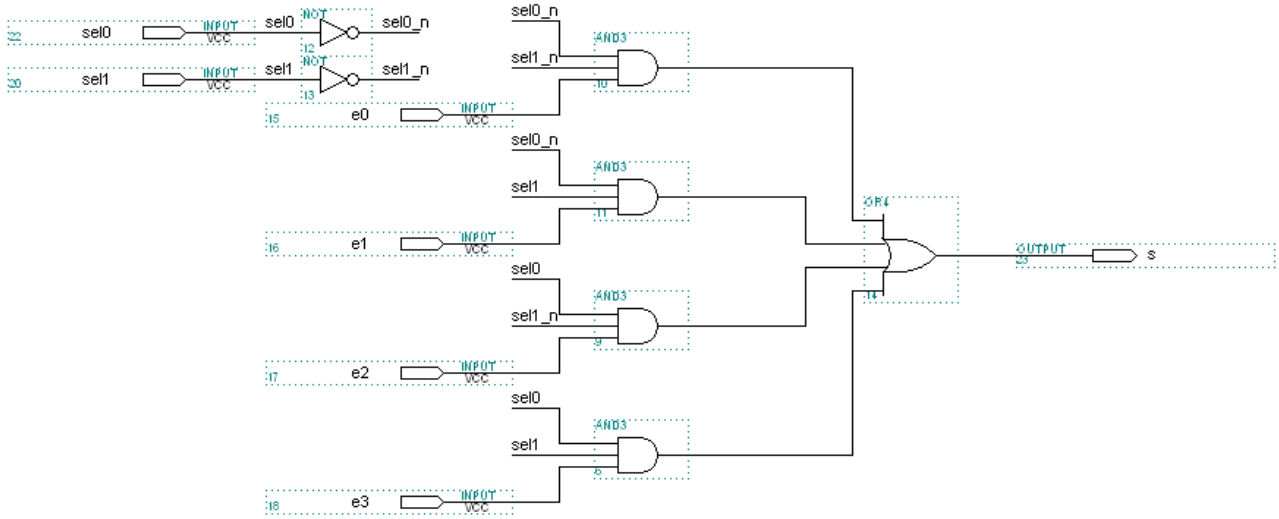
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

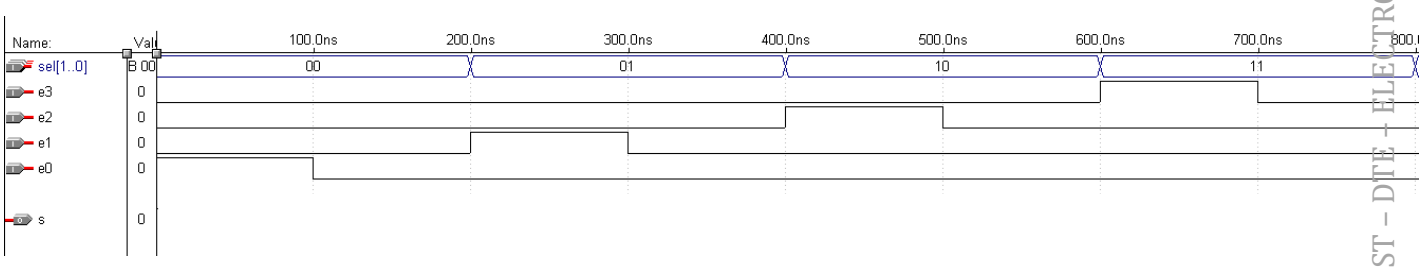
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 6: Multiplexor de 4 a 1. Se ha diseñado un circuito multiplexor de 4 canales que debe obtener a su salida la entrada 'e_i' cuando en las entradas de selección aparezca el valor 'i'.



Complete el cronograma de la figura. Para cada combinación de las entradas, calcule en primer lugar las salidas de las 4 puertas AND y luego obtenga la salida del circuito calculada con la OR.



A la vista del cronograma obtenido, verá que el circuito no se comporta como un multiplexor de 4:1. Explique cuáles son las diferencias.

Averigüe dónde está el error y corrija el diseño para que el circuito se comporte correctamente.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

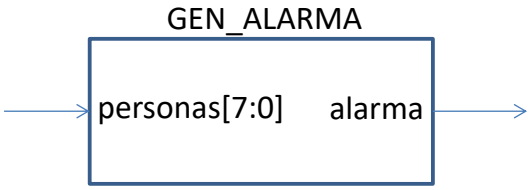
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Soluciones de los ejercicios.

A continuación se detallan las soluciones a los seis ejercicios propuestos en esta actividad. No mire las soluciones hasta que no haya resuelto los ejercicios.

Ejercicio 1: Alarma. El sistema tiene como entrada el número de personas en un local (binario natural, 8 bits) y genera una salida de alarma (a 1 lógico) cuando este número supera los 150.



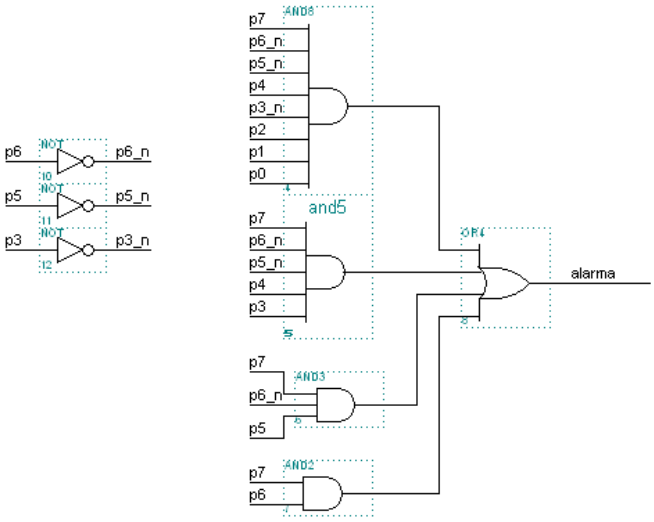
Obtenga la tabla de verdad del sistema.

p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0	alarma
1	0	0	1	0	1	1	1	1
1	0	0	1	1	X	X	X	1
1	0	1	X	X	X	X	X	1
1	1	X	X	X	X	X	X	1
Otras combinaciones								0

Obtenga la expresión booleana que relaciona la salida con las entradas.

$$alarma = p_7\,\overline{p}_6\,\overline{p}_5\,p_4\,\overline{p}_3\,p_2\,p_1\,p_0 + p_7\,\overline{p}_6\,\overline{p}_5\,p_4\,p_3 + p_7\,\overline{p}_6\,p_5 + p_7\,p_6$$

Dibuje el circuito con puertas lógicas.

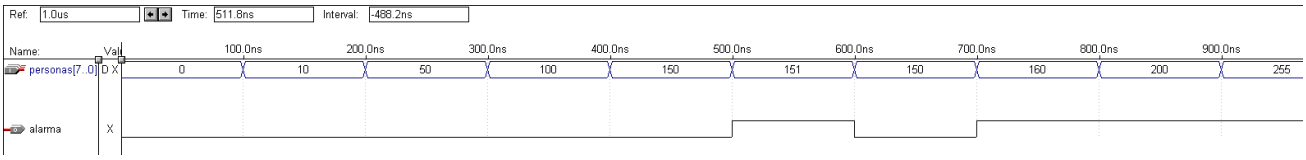


Complete el siguiente cronograma de funcionamiento:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

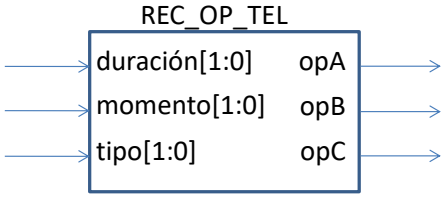
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019



Ejercicio 2: Recomendador de operador de telefonía. Un circuito recomendador de operador de telefonía dispone de la información acerca de la duración esperada de su próxima llamada (menos de 1 minuto, entre uno y 5 minutos, más de 5 minutos), del momento del día (mañana, tarde o noche) y del tipo de llamada (local, nacional o internacional). A partir de esa información, recomienda el operador A para todas las llamadas locales, así como para las nacionales, siempre que la llamada no exceda de 1 minuto o bien supere los 5 minutos y (solo en este último caso) sea por la tarde. El operador B se recomienda para todas las llamadas internacionales, así como para las nacionales superiores a 1 minuto por la mañana o superiores a 5 minutos por la noche. Para el resto de llamadas el sistema recomienda el operador C.

Obtenga el símbolo del sistema y realice la codificación de las entradas. Utilice una salida para cada operador.



duración[1:0] d_1	d_0	Duración de la llamada
0	0	menos de 1 ‘
0	1	entre 1’ y 5’
1	0	más de 5’
momento[1:0] m_1	**m_0**	**Momento de la llamada**
0	0	mañana
0	1	tarde
1	0	noche
tipo[1:0] t_1	**t_0**	**Tipo de llamada**
0	0	local
0	1	nacional
1	0	internacional

Obtenga la tabla de verdad del sistema y las ecuaciones de las salidas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

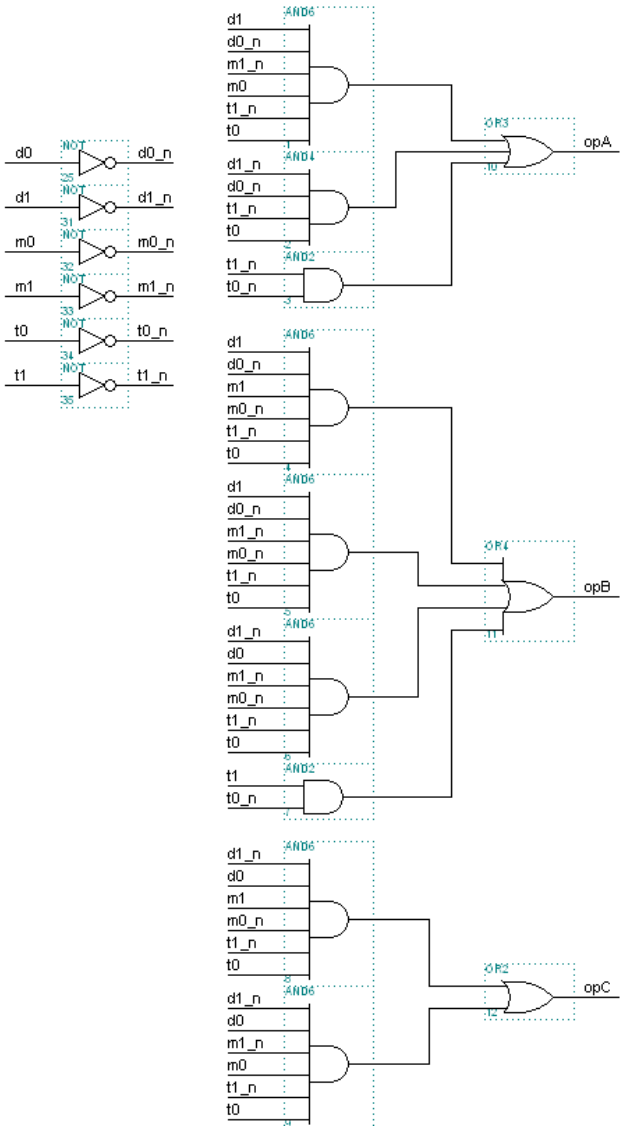
d_1	d_0	m_1	m_0	t_1	t_0	opC	opB	opA	Tipo de llamada
X	X	X	X	0	0	0	0	1	Todas las locales
X	X	X	X	1	0	0	1	0	Todas las internacionales
0	0	X	X	0	1	0	0	1	Nacionales, inf. 1’
1	0	0	1	0	1	0	0	1	Nacionales, sup. 5’, tarde
0	1	0	0	0	1	0	1	0	Nacionales, sup. 1’, mañana
1	0	0	0	0	1	0	1	0	
1	0	1	0	0	1	0	1	0	Nacionales, sup. 5’, noche
0	1	0	1	0	1	1	0	0	Nacionales, entre 1’ y 5’, tarde o noche
0	1	1	0	0	1	1	0	0	
resto						X	X	X	Opciones no válidas

$$opA = \bar{t}_1\bar{t}_0 + \bar{d}_1\bar{d}_0\bar{t}_1t_0 + d_1\bar{d}_0\bar{m}_1m_0\bar{t}_1t_0$$

$$opB = t_1\bar{t}_0 + \bar{d}_1d_0\bar{m}_1\bar{m}_0\bar{t}_1t_0 + d_1\bar{d}_0\bar{m}_1\bar{m}_0\bar{t}_1t_0 + d_1\bar{d}_0m_1\bar{m}_0\bar{t}_1t_0$$

$$opC = \bar{d}_1d_0\bar{m}_1m_0\bar{t}_1t_0 + \bar{d}_1d_0m_1\bar{m}_0\bar{t}_1t_0$$

Represente el circuito del recomendador con puertas lógicas.



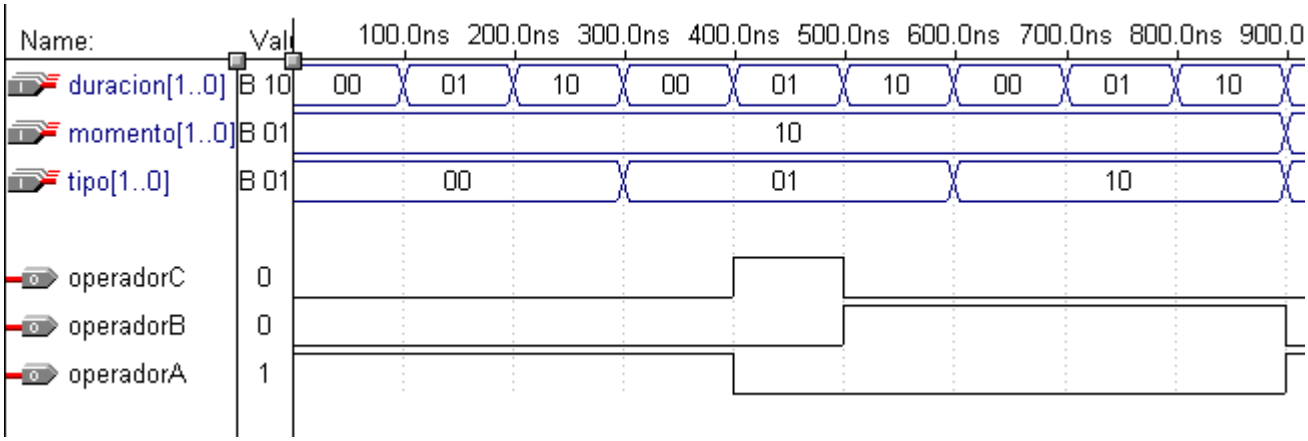
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Dibuje un cronograma de funcionamiento en el que se comprueben todas las combinaciones posibles de duración y tipo de la llamada sólo para las llamadas nocturnas. Utilice buses para representar cada tipo de entrada y señales individuales para las salidas. Complete los tamaños de los buses en el cronograma.



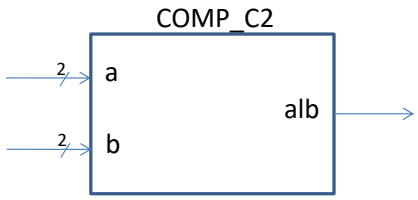
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

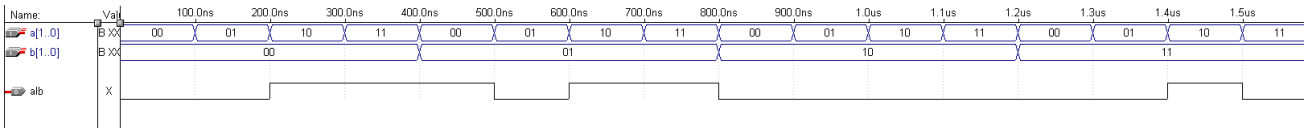
Ejercicio 3: Comparador de 2 bits en C2. Diseñe un comparador de 2 bits en código complemento a 2. Este circuito comparará dos entradas, 'a' y 'b', codificadas en C2 y activará su salida, 'alb', si la entrada 'a' es menor que la entrada 'b'.

Dibuje el símbolo del sistema y obtenga su tabla de verdad.



b_1	b_0	a_1	a_0	alb
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

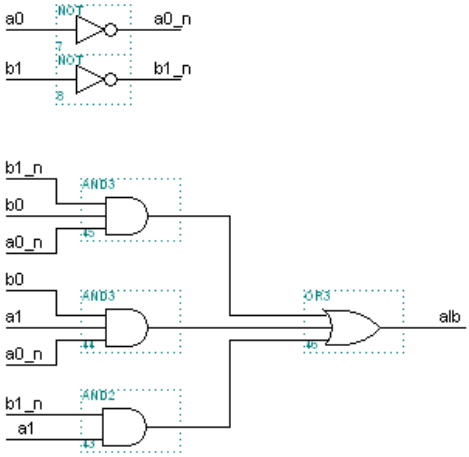
Complete el siguiente cronograma:



Obtenga la expresión booleana de la salida 'alb' en función de las entradas.

$$alb = \overline{b}_1\overline{b}_0 a_1\overline{a}_0 + \overline{b}_1\overline{b}_0 a_1 a_0 + \overline{b}_1 b_0\overline{a}_1\overline{a}_0 + \overline{b}_1 b_0 a_1\overline{a}_0 + \overline{b}_1 b_0 a_1 a_0 + b_1 b_0 a_1\overline{a}_0 = \overline{b}_1 a_1 + \overline{b}_1 b_0\overline{a}_0 + b_0 a_1\overline{a}_0$$

Dibuje una representación del circuito con puertas lógicas.



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

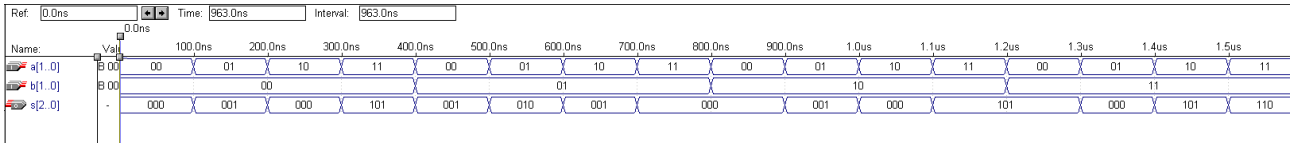
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 4: Sumador de 2 bits en código valor absoluto y signo (VAS).

En el código VAS, cada número se codifica con N bits para la magnitud y un bit de signo. Los números positivos siempre tienen como bit de signo el '0' mientras que los negativos tienen como bit de signo el '1'. Por ejemplo, con 5 bits pueden codificarse números entre el 15_d (01111) y el -15_d (11111). El 8_d se codificaría como '01000' y el -2_d se codificaría como '10010'. El cero siempre puede codificarse de dos maneras diferentes (con 5 bits, '00000' y '10000'). Si solo dispusiéramos, por ejemplo, de 3 bits, todavía podríamos codificar el cero ('000' o '100') y el -2d ('110'), pero no el 15_d, el -15_d o el 8_d.

Diseñe un sumador de 2 bits en el que las entradas y la salida estén codificadas en VAS. El circuito debe comportarse según el siguiente cronograma:

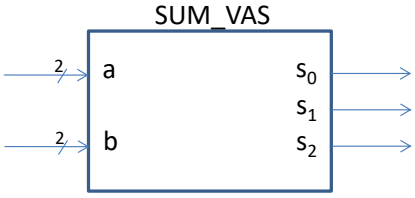


Rellene en primer lugar las siguientes tablas con los códigos VAS de los números enteros que pueden codificarse con 2 y 3 bits:

número	Código VAS	
	signo	magnitud
	b	a
+0	0	0
-0	1	0
+1	0	1
-1	1	1

número	Código VAS		
	signo	magnitud	
	c	b	a
+0	0	0	0
-0	1	0	0
+1	0	0	1
-1	1	0	1
+2	0	1	0
-2	1	1	0
+3	0	1	1
-3	1	1	1

Dibuje el símbolo del circuito y obtenga la tabla de verdad. Añada a la derecha de la tabla de verdad una columna donde se describa la operación que realiza el sumador para cada una de las filas.



b_1	b_0	a_1	a_0	s_2	s_1	s_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	1	0	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	1
1	0	1	0	0	0	0
1	0	1	1	1	0	1
1	1	0	0	1	0	1
1	1	0	1	0	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

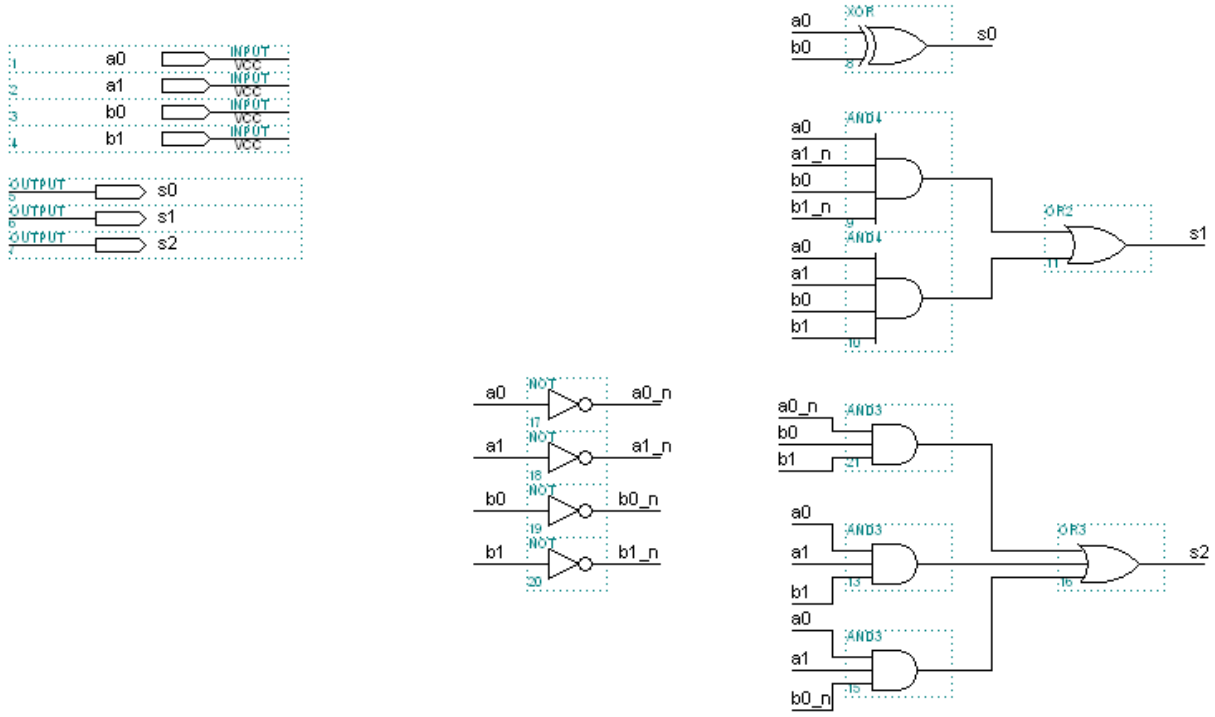
Obtenga la expresión booleana de las salidas y dibuje una representación del circuito con puertas lógicas.

$$s_0 = \bar{b}_1\bar{b}_0\bar{a}_1a_0 + \bar{b}_1\bar{b}_0a_1a_0 + \bar{b}_1b_0\bar{a}_1\bar{a}_0 + \bar{b}_1b_0a_1\bar{a}_0 + b_1\bar{b}_0\bar{a}_1a_0 + b_1\bar{b}_0a_1a_0 + b_1b_0\bar{a}_1\bar{a}_0 + b_1b_0a_1\bar{a}_0$$

$$= \bar{b}_0a_0 + b_0\bar{a}_0$$

$$s_1 = \bar{b}_1b_0\bar{a}_1a_0 + b_1b_0a_1a_0$$

$$s_2 = \bar{b}_1\bar{b}_0a_1a_0 + b_1\bar{b}_0a_1a_0 + b_1b_0\bar{a}_1\bar{a}_0 + b_1b_0a_1a_0 + b_1b_0a_1\bar{a}_0 = \bar{b}_0a_1a_0 + b_1a_1a_0 + b_1b_0\bar{a}_0$$



© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

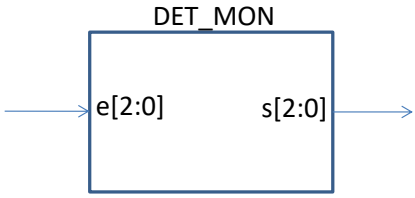
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 5: Detección de monedas. El sistema recibe la información del peso de una moneda en el rango entre 0 y 30 g. con una resolución de 5 g. y decide el tipo de moneda en función del peso. Si pesa menos de 5 g. es de 5 c., entre 5 g. y 9 g. es de 10 c., entre 10 g. y 19 g. es de 50 c., entre 20 g. Y 24 g. es de 1 €, a partir de 25 g. es de 2 €. Este sistema se ha diseñado de la siguiente forma:

En primer lugar, se ha establecido la codificación de las entradas y de las salidas:

Peso de la moneda	Entradas		
	e2	e1	e0
0 g.	0	0	0
5 g.	0	0	1
10 g.	0	1	0
15 g.	0	1	1
20 g.	1	0	0
25 g.	1	0	1
30 g.	1	1	0

Valor de la moneda	Salidas		
	s2	s1	s0
5 c.	0	0	0
10 c.	0	0	1
50 c.	0	1	0
1 €.	0	1	1
2 €.	1	0	0



Luego se ha obtenido la siguiente tabla de verdad:

e_2	e_1	e_0	s_2	s_1	s_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	X	X	X

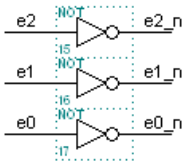
A partir de la tabla de verdad se han obtenido las expresiones de las salidas en función de las entradas (considerando que las indiferencias son '0') y una representación con puertas lógicas:

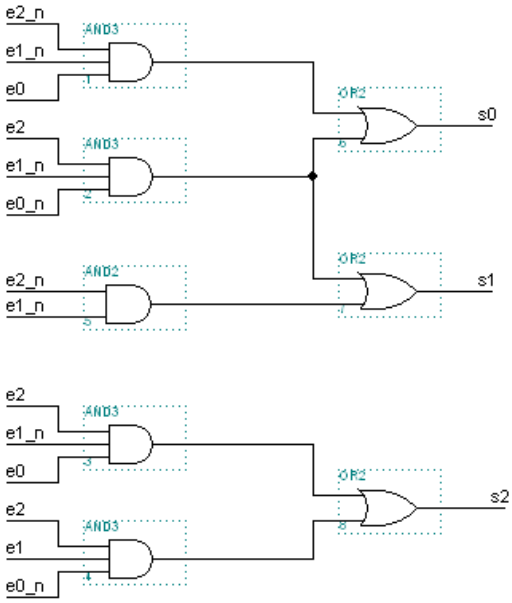
$$s_0 = \bar{e}_2\bar{e}_1e_0 + e_2\bar{e}_1\bar{e}_0$$

$$s_1 = \bar{e}_2e_1\bar{e}_0 + \bar{e}_2e_1e_0 + e_2\bar{e}_1\bar{e}_0$$

$$= \bar{e}_2\bar{e}_1 + e_2\bar{e}_1\bar{e}_0$$

$$s_2 = e_2\bar{e}_1e_0 + e_2e_1\bar{e}_0$$



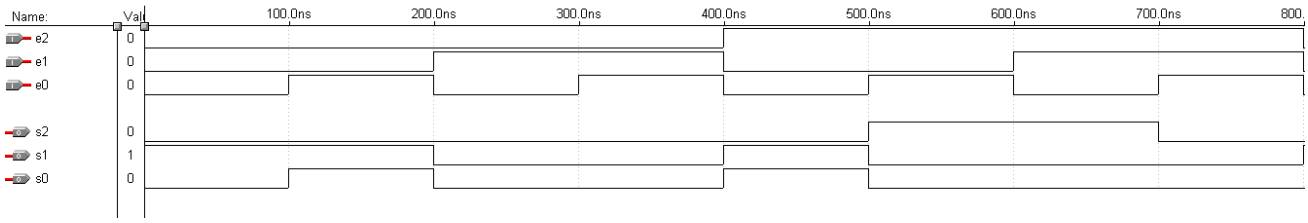


© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Finalmente, se ha obtenido el siguiente cronograma de funcionamiento mediante el procedimiento de dar valores a las entradas y obtener los valores de salida que generaría el circuito representado con puertas lógicas.



Revise con detalle el cronograma y compruebe si el funcionamiento se corresponde con el que se define en la tabla de verdad. Si observa algún error en el diseño, corríjalo.

A partir del cronograma del circuito se puede obtener la tabla de verdad que está implementando:

e_2	e_1	e_0	s_2	s_1	s_0
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	0
1	1	1	0	0	0

Comparando esta tabla de verdad con la que debería tener el circuito, se observa que hay errores en la función s1 (que se han resaltado en color rojo). Revisando la expresión booleana de la función s1, se observa que:

$$s_1 = \bar{e}_2 e_1 \bar{e}_0 + \bar{e}_2 e_1 e_0 + e_2 \bar{e}_1 \bar{e}_0 = \sum m(2,3,4)$$

Lo cual es correcto. Sin embargo, esta expresión se ha simplificado como:

$$s_1 = \bar{e}_2 \bar{e}_1 + e_2 \bar{e}_1 \bar{e}_0$$

en lugar de

$$s_1 = \bar{e}_2 e_1 + e_2 \bar{e}_1 \bar{e}_0$$

Con esta última expresión, puede comprobarse que la tabla de verdad coincide con la que debería tener el circuito.

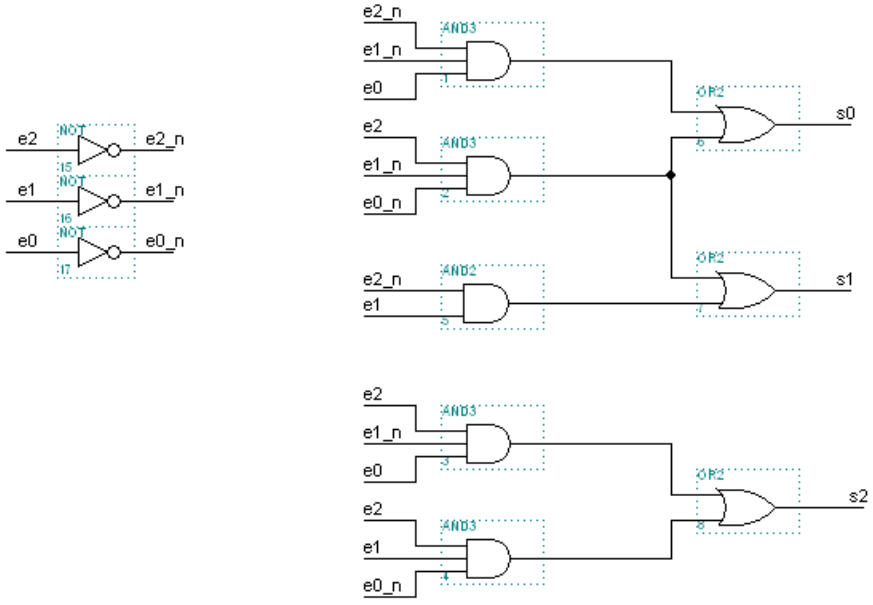
El error está también en la representación con puertas lógicas del circuito. La versión corregida es la siguiente:

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019



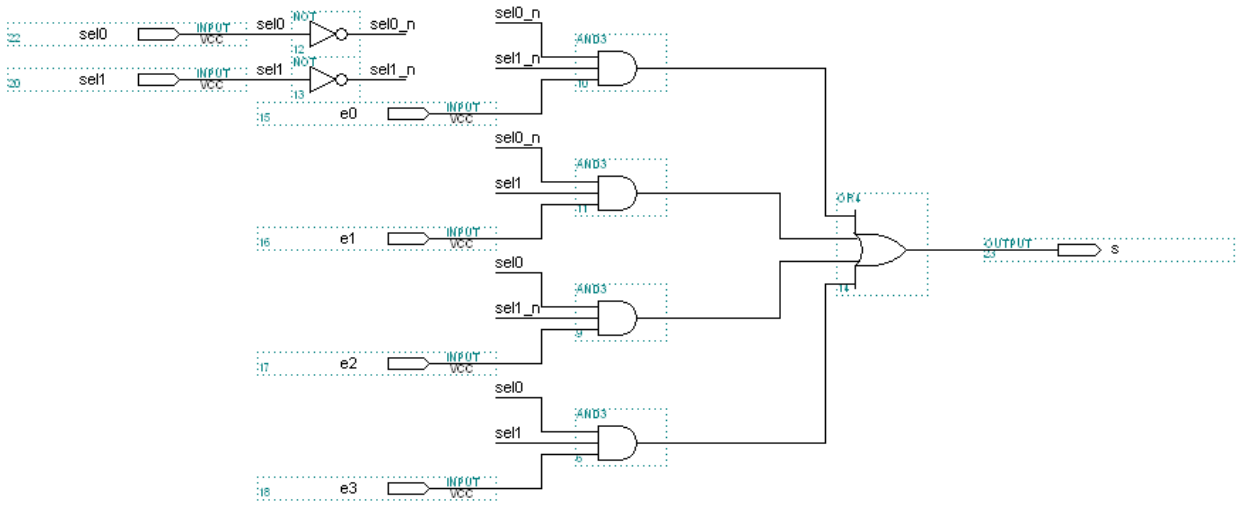
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

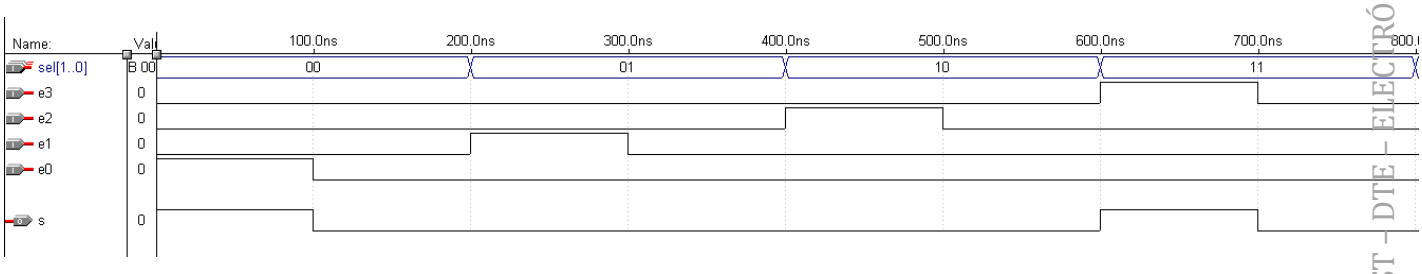
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Ejercicio 6: Multiplexor de 4 a 1. Se ha diseñado un circuito multiplexor de 4 canales que debe obtener a su salida la entrada 'e_i' cuando en las entradas de selección aparezca el valor 'i'.



Complete el cronograma de la figura. Para cada combinación de las entradas, calcule en primer lugar las salidas de las 4 puertas AND y luego obtenga la salida del circuito calculada con la OR.



A la vista del cronograma obtenido, verá que el circuito no se comporta como un multiplexor de 4:1. Explique cuáles son las diferencias.

Cuando las entradas de selección son '01' o '10' la salida permanece a cero aún cuando las entradas e_1 y e_2 cambian a '1'.

Averigüe dónde está el error y corrija el diseño para que el circuito se comporte correctamente.

El diseño implementa la siguiente función:

$$s = \overline{sel_1}\,\overline{sel_0}e_0 + sel_1\overline{sel_0}e_1 + \overline{sel_1}sel_0e_2 + sel_1sel_0e_3$$

que se corresponde con esta tabla de verdad:

sel_1	sel_0	s
0	0	e_0
0	1	e_2
1	0	e_1
1	1	e_3

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

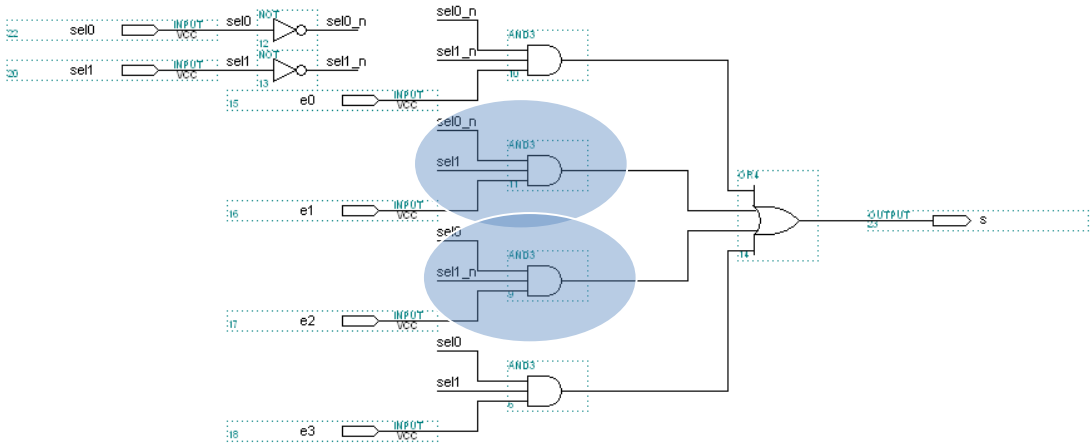
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

la cual, efectivamente, no se corresponde con la de un multiplexor de 4 canales. La correcta sería la siguiente:

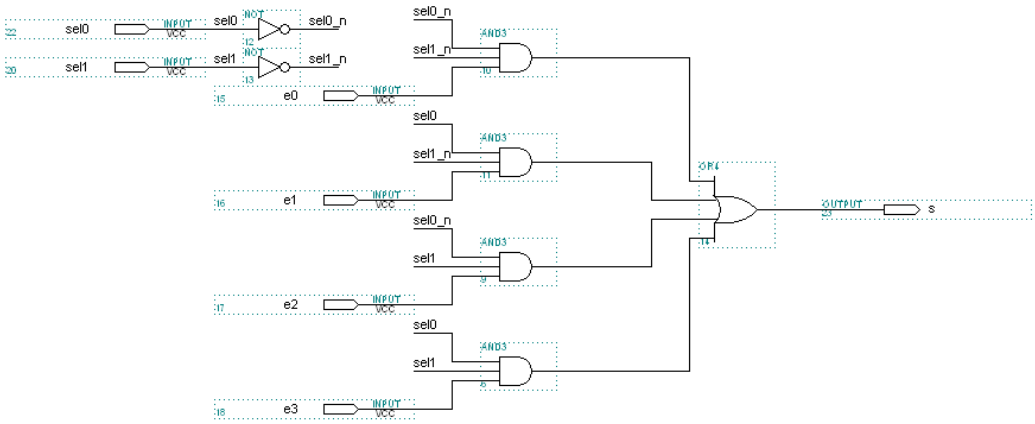
sel_1	sel_0	s
0	0	e_0
0	1	e_1
1	0	e_2
1	1	e_3

El fallo del circuito está en la implementación de los términos de dos de las puertas AND:



En este circuito, cuando las entradas de selección son '00' la salida es e0 y cuando son '11' la salida es e1, lo cual en ambos casos es correcto. Pero cuando las entradas de selección son '01' la salida es e2 en lugar de e1. Así mismo, cuando las entradas de selección son '10' la salida es e1 en lugar de e2.

El circuito correcto es el siguiente:



En este circuito puede verse que, cuando las entradas de selección valen '01' la salida es e_1 y cuando valen '10' la salida es e_2.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Ejercicios sobre síntesis de subsistemas combinacionales.**	**CÓDIGO:** **BT1_ATGP7**

GRUPO		FECHA:	

MODALIDAD:	Aprendizaje Cooperativo	TIPO:	Presencial	DURACIÓN:	100 minutos
CALENDARIO:	Undécima Sesión	REQUISITOS:	Metodología de síntesis de circuitos combinacionales. Puertas lógicas.		

CRITERIO DE ÉXITO:	Cualquier miembro del grupo deberá ser capaz de explicar todos y cada uno de los cuatro ejercicios que se han trabajado.
INTERDEPENDENCIA POSITIVA:	Cada miembro del grupo debe aportar su capacidad de razonamiento para llegar a la solución del problema que le ha correspondido. Los demás miembros del grupo dependen de los resultados de uno para tener el problema resuelto de manera completa.
EXIGIBILIDAD INDIVIDUAL:	Cada miembro del grupo es responsable de su parte del trabajo.
ROLES:	*Secretarios*: toman nota de las decisiones alcanzadas y rellenan las hojas de resultados. *Controlador de Tiempo*: controla el tiempo que queda para la finalización de la tarea. *Moderador*: dirige la discusión, controla los turnos de palabra y garantiza la participación por igual de todos los miembros.
HABILIDADES SOCIALES:	Capacidad para expresar y sintetizar ideas, capacidad para escuchar, asimilar y organizar la información presentada por otra persona. Capacidad para razonar y analizar.
REFLEXIÓN DE GRUPO:	Se deberá analizar el grado de participación de cada miembro del grupo y su capacidad como grupo para mantener un debate fluido que lleve a una toma de decisiones eficaz y compartida por todos

TIEMPO DEDICADO:	No procede	CALIFICACIÓN: [entre 0 y 10 puntos]	No procede

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

PARTE I. Descripción de la Actividad.

a) Cada miembro del grupo se queda con uno de los ejercicios. Pensará su solución y la pondrá por escrito en una hoja. (15 minutos)
b) Se reunirán en "grupos de expertos" de cuatro en cuatro, los miembros de distintos grupos que han trabajado un mismo ejercicio, con el fin de contrastar soluciones, resolver dudas y garantizar que los cuatro integrantes han llegado a una solución correcta. (15 minutos)
c) Cada uno de los miembros de los grupos de trabajo originales retorna a su grupo y expone a sus compañeros la solución del ejercicio que ha realizado. La exposición deberá ser interactiva, en el sentido de que los compañeros que atienden a la explicación lo harán con una actitud crítica que permita detectar errores y subsanarlos. El objetivo es que todos los integrantes de cada grupo comprendan la solución de todos los ejercicios. (40 minutos)
d) Revisión del documento final a entregar al profesor. (10 minutos)
e) Test final a uno de los miembros del grupo. (20 minutos)

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 1) Codificador con prioridad.

Diseñe un circuito codificador con 4 entradas de datos (e_0, e_1, e_2, e_3), una entrada de habilitación (en) y dos salidas (s_0, s_1). Mientras la entrada de habilitación esté activa (a '1'), las salidas codifican en binario la entrada activada, con el mismo código que el subíndice de la entrada. Si dos o más entradas se activan a un tiempo, la salida codificará la de mayor prioridad, que será la que tenga el subíndice **más bajo**. Si la entrada de habilitación se desactiva (a '0'), las salidas deberán estar a "00", independientemente de la entrada o entradas que se activen.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 2) Multiplexor.

Diseñe un multiplexor de cuatro entradas, con las entradas de selección activas a nivel bajo.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 3) Comparador.

Diseñe un circuito comparador de 2 bits del tipo ‘A<=B’.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

EJERCICIO 4) Decodificador.

Diseñe un circuito decodificador de 3 a 8, con las salidas activas a nivel bajo.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Test final sobre síntesis de subsistemas combinacionales.**	**CÓDIGO:** **BT1_ATGP7**

FECHA:	

NOMBRE:		APELLIDOS:	

GRUPO		CALIFICACIÓN: [entre 0 y 10 puntos]	puntos

EJERCICIO 5) Codificador con prioridad de 4 a 2 con entradas y salidas activas a nivel bajo.

Diseñe un circuito combinacional con 4 entradas que obtenga a su salida el código binario correspondiente a la entrada activada. Las entradas y las salidas deberán ser activas a nivel bajo (ej: la entrada e0 se activa con un '0' y produce a la salida el código "11"). Si dos o más entradas se activan al mismo tiempo la salida codifica la más prioritaria, que en este caso va a ser la que tenga un subíndice mayor. Si no se activa ninguna entrada la salida debe ser "11". Dibuje el símbolo del codificador y obtenga su tabla de verdad, funciones booleanas de las salidas y el correspondiente circuito con puertas lógicas.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta página se ha dejado intencionadamente en blanco

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Electrónica 2
Bloque Temático 1

Sistemas combinacionales complejos.
Ejemplos de diseño: sumadores serie, multiplexores y restadores.
Lenguajes de descripción *hardware* (HDLs)

© 2010-2015 UPM

Circuitos combinacionales complejos

- La complejidad de los circuitos combinacionales depende, fundamentalmente, del número de entradas (con N entradas la tabla de verdad tiene 2^N filas).
- También depende del número de salidas (M salidas = M funciones), pero esa complejidad es abordable.
- Finalmente, la complejidad depende de la funcionalidad. Por ejemplo, un codificador de 3 entre 8 puede abordarse con las metodologías que conoce, un sumador de 4 bits, no.

© 2010-2015 UPM BT1_S12 BT1_P20_2

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Cómo abordar la complejidad

- Los circuitos con un número grande de entradas, pero cuya tabla de verdad posee pocas filas en las que la función vale '1' (o '0'), pueden resolverse con facilidad aplicando la metodología de diseño basada en tablas de verdad (p. ej, el circuito detector del cumpleaños de los miembros de una familia de 3 tiene 9 entradas).
- También los circuitos con un número grande de entradas, pero en los que la tabla de verdad puede ponerse de forma resumida pueden abordarse más fácilmente con esa metodología (p. ej, el multiplexor de 8 canales tiene 11 entradas).
- Para el resto de circuitos, a partir de 5 entradas, no resulta factible su diseño utilizando la metodología de diseño que conoce. Es necesario, pues, utilizar otra.

Divide et vinces

- La estrategia 'dividir para vencer' se utiliza en electrónica digital para el diseño de todo tipo de circuitos (no solo los combinacionales).
- Consiste en dividir el problema complejo en partes más simples que sean abordables directamente con las metodologías ya conocidas y, luego, interconectar esas partes para resolver el problema completo.
- En las próximas dispositivas se verán 3 ejemplos. El objetivo no es ser capaz de realizar diseños complejos sino tomar conciencia de que esos diseños complejos existen y de que hay soluciones para abordar esa complejidad.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sumador de N bits

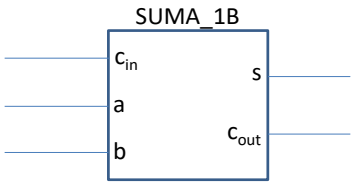
- Un sumador de N bits tiene 2N entradas. Para N>2 es un circuito complejo (p. ej, con N=3 la tabla de verdad tendría 64 filas).
- Pero, si nos fijamos en el algoritmo de la suma, por ejemplo:

```
  1 1
  1 0 1 1
+ 0 1 1 0
1 0 0 0 1
```

Para sumar dos bits solo es necesario conocer el acarreo del bit anterior y los dos bits que hay que sumar y hay que generar como salidas la suma y el acarreo de salida para el bit siguiente

Sumador de 1 bits

- Un sumador de 1 bit con acarreo de entrada y salida es muy sencillo de diseñar:



c_{in}	b	a	c_{out}	s
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

$$s = \overline{c_{in}}\,\bar{b}\,a + \overline{c_{in}}\,b\,\bar{a} + c_{in}\,\bar{b}\,\bar{a} + c_{in}\,b\,a$$

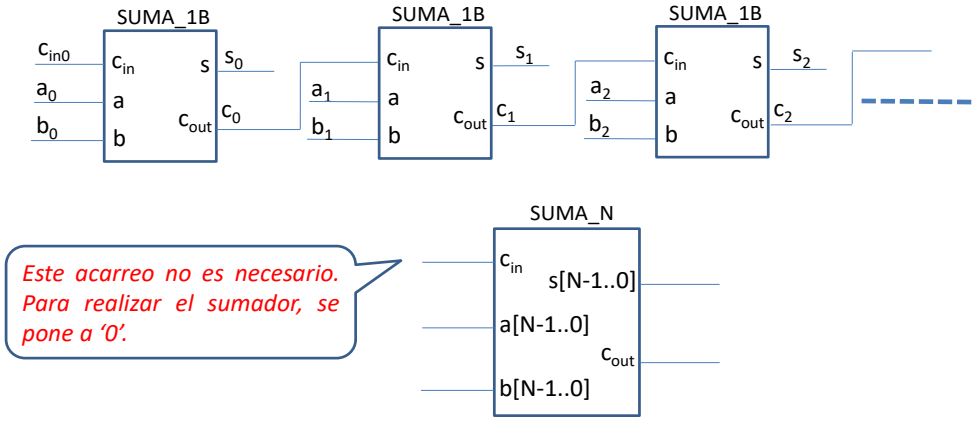
$$c_{out} = \overline{c_{in}}\,b\,a + c_{in}\,\bar{b}\,a + c_{in}\,b\,\bar{a} + c_{in}\,b\,a$$

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Sumador de N bits

- Interconectando N sumadores de 1 bit pueden diseñarse un sumador de N bits:



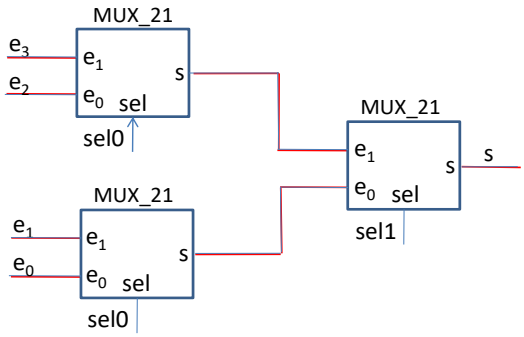
Multiplexor de N canales

- Un multiplexor de N canales tiene N entradas de datos y $\log_2 N$ entradas de selección. Para N>16 es un circuito complejo (p. ej, con N=32 la tabla de verdad resumida tendría 32 filas).

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Multiplexor de N canales

- Con varios multiplexores sencillos se puede construir uno más complejo, por ejemplo:



sel_1	sel_0	s	e_2	e_1	e_0	s
0	0	e_0	X	X	e_0	e_0
0	1	e_1	X	e_1	X	e_1
1	0	e_2	e_2	X	X	e_2
1	1	e_3	X	X	X	e_3

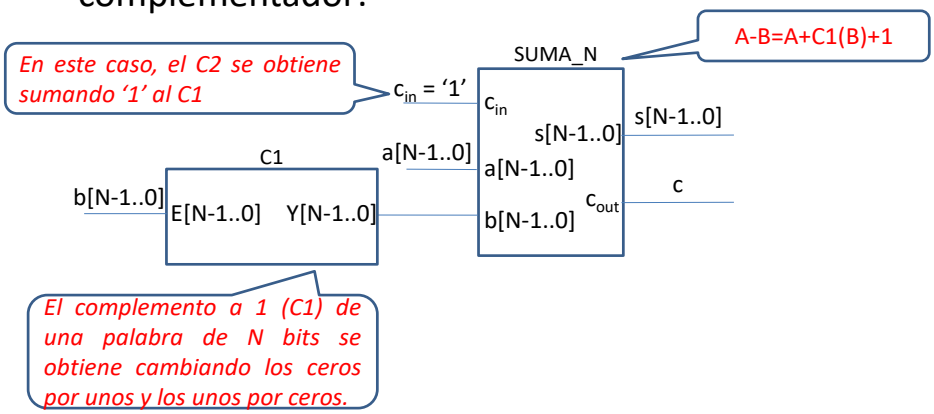
Multiplexor de N canales

- Si tiene curiosidad, el multiplexor de N canales puede diseñarse utilizando multiplexores de 2 canales, con $\log_2 N$ etapas:
  - La primera etapa con N/2 multiplexores
  - La segunda etapa con N/4 multiplexores
  - …
  - La última etapa con un multiplexor
- También pueden utilizarse otras estructuras basadas en combinaciones de multiplexores de más de 2 canales

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Restador de N bits en C2

- Codificando los datos de entrada y de salida en C2, un restador de N bits puede realizarse (y habitualmente se realiza) utilizando un sumador de N bits y un complementador:



Lenguajes de Descripción *Hardware* (HDLs) (I)

- El problema de la metodología conocida hasta ahora se centra en la imposibilidad material de:
  - escribir la tabla de verdad cuando esta es muy grande
  - realizar la obtención de las funciones lógicas cuando estas tiene muchos términos
  - obtener el circuito cuando éste contiene muchas puertas
- Sin embargo, supuesto que la tabla de verdad ha sido escrita, los pasos de la metodología son completamente mecánicos y susceptibles de ser realizados automáticamente por herramientas informáticas adecuadas

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Lenguajes de Descripción *Hardware* (HDLs) (II)

- El problema de diseño se centra, por tanto, en describir la tabla de verdad (la función) a una herramienta de diseño capaz de, a partir de ella, obtener las funciones lógicas y el circuito que la implementa
- Para poder describir tablas de verdad complejas interesa emplear una representación algorítmica de las mismas:
  - una descripción algorítmica concisa puede representar una tabla muy extensa
  - es fácilmente procesable por una herramienta informática

Lenguajes de Descripción *Hardware* (HDLs) (III)

- Existen lenguajes estandarizados que permiten representar textualmente la función de un circuito digital. Se conocen como lenguajes de descripción *hardware* (*Hardware Description Languages)*:
  - VHDL
  - Verilog y SystemVerilog
  - SystemC
- En estos lenguajes se define tanto la interfaz como la función del circuito

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

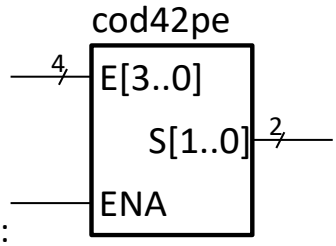
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Lenguajes de Descripción *Hardware* (HDLs) (IV)

- Por ejemplo, la interfaz de un codificador 4 a 2 con prioridad y *enable*:



- Puede describirse en VHDL como:

```
entity cod42pe is
  port(
    ENA : in    bit;
    E   : in    bit_vector(3 downto 0);
    S   : out   bit_vector(1 downto 0)
  );
end entity cod42pe;
```


Lenguajes de Descripción *Hardware* (HDLs) (V)

- Y su función podría describirse en VHDL como:

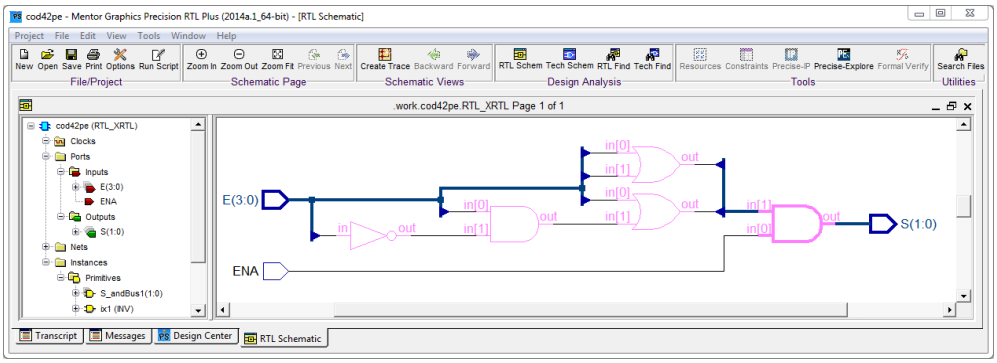
```
architecture RTL of cod42pe is
  S <= "00" when ENA = '0' else
       "11" when E(3) = '1' else
       "10" when E(2) = '1' else
       "01" when E(1) = '1' else
       "00";
end architecture RTL;
```


© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

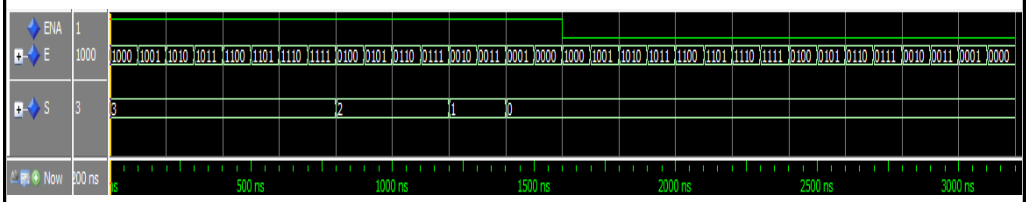
Lenguajes de Descripción *Hardware* (HDLs) (VI)

- A partir del anterior código VHDL, una herramienta CAD denominada *sintetizador lógico* puede obtener la implementación del circuito con puertas o empleando otras tecnologías



Lenguajes de Descripción *Hardware* (HDLs) (VII)

- Del mismo modo, otra herramienta CAD denominada *simulador lógico* puede obtener las salidas que el circuito generará a partir de unos estímulos dados (VHDL, Verilog/SystemVerilog y SystemC también permiten describir cómo son esos estímulos)



- Estas técnicas se describen en las asignaturas «Diseño Digital I», «Diseño Digital II» y «Diseño Digital Avanzado»

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

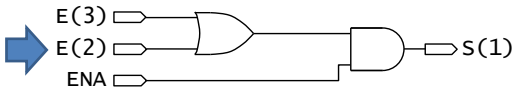
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

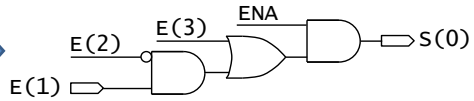
Lenguajes de Descripción *Hardware* (HDLs) (VIII)

- Otro modelo VHDL válido para el codificador podría ser:

```
architecture RTL of cod42pe is
 S(1)<='0' when ENA = '0' else
       '1' when E(3) = '1' else
       '1' when E(2) = '1' else
       '0' when E(1) = '1' else
       '0';

 S(0)<='0' when ENA = '0' else
       '1' when E(3) = '1' else
       '0' when E(2) = '1' else
       '1' when E(1) = '1' else
       '0';
end architecture RTL;
```





- Los dos subcircuitos trabajan simultáneamente y, por tanto, los HDL deben ser *lenguajes concurrentes*

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019
© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Preparación en grupo del examen.**	**CÓDIGO:** **BT1_ATGNP1**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Repaso. Trabajo en grupo.	TIPO:	No presencial	DURACIÓN:	60 minutos

CALENDARIO:	Duodécima Sesión.	REQUISITOS:	

CRITERIO DE ÉXITO:	

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	puntos

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Esta última sesión presencial del primer bloque temático sólo ocupa una de las dos horas habilitadas en el horario de clases. La segunda hora se dedicará en la propia aula a preparar en grupo el examen del bloque.

Consejos para la preparación en grupo del examen

- Haga una puesta en común con los demás miembros de su grupo de todas las actividades realizadas hasta el momento. Si se encuentran discrepancias entre los miembros del grupo o actividades que alguno de los miembros no haya sido capaz de realizar correctamente intente resolver las dudas junto con sus compañeros. Si queda alguna duda que no sea posible resolver solicite una tutoría, preferentemente en grupo.
- Revise el modelo de examen que tiene disponible en Moodle. Vea, junto con sus compañeros, si sabe cómo abordar cada uno de los ejercicios que en él se plantean.
- Revise la lista de indicadores en la Guía de la Asignatura. Los indicadores proporcionan información detallada y concreta sobre lo que debe ser capaz de hacer tras realizar las actividades programadas del bloque temático.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

BLOQUE TEMÁTICO 1

TÍTULO DE LA ACTIVIDAD: **Preparación individual del examen.**	**CÓDIGO:** **BT1_AINP20**

FECHA:			
NOMBRE:		APELLIDOS:	

MODALIDAD:	Repaso. Trabajo individual.	TIPO:	No presencial	DURACIÓN:	160 minutos

CALENDARIO:	Duodécima Sesión. Previo a la realización del examen	REQUISITOS:	

CRITERIO DE ÉXITO:	

COMENTARIOS E INCIDENCIAS:

TIEMPO DEDICADO:	minutos	AUTOEVALUACIÓN: [entre 0 y 10 puntos]	puntos

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

Consejos para la preparación del examen

- Revise la lista de indicadores en la Guía de la Asignatura. Los indicadores proporcionan información detallada y concreta sobre lo que debe ser capaz de hacer tras realizar las actividades programadas del bloque temático.
- Revise todas las actividades realizadas hasta el momento. Si hay actividades que no ha realizado correctamente, revise el material disponible sobre la temática de esas actividades e intente realizarlas de nuevo.

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019

© 2010-2018 UPM – ETSIST – DTE – ELECTRÓNICA II – Curso 2018-2019