

Arquitecturas y Organización de Computadoras I

2: Análisis de circuitos digitales

Rafael Ignacio Zurita

Depto. Ingeniería de Computadoras

September 16, 2020

Diseño Lógico - Diseño Digital

- * **Señales digitales**
- * **Análisis de Circuitos Digitales**
- * **Metodología para realizar un diseño lógico/digital**
- * **Circuitos combinacionales comunes**
- * **Relojes**
- * **Circuitos secuenciales**
- * **Máquinas de Estado Finito (FSM)**

Diseño Lógico - Diseño Digital

- * Señales digitales
- * Análisis de Circuitos Digitales
- * Metodología para realizar un diseño lógico/digital
- * Circuitos combinacionales comunes
- * Relojes
- * Circuitos secuenciales
- * Máquinas de Estado Finito (FSM)

» Señales digitales

- * La electrónica interna de un computador actual utiliza componentes analógicos (como los transistores en un circuito cmos).
- * De todas maneras se modela de manera digital para operar con dos niveles de voltaje: un voltaje alto y un voltaje bajo
- * El resto de los valores de voltaje son temporales y ocurren durante la transición entre los valores alto y bajo o bajo y alto

» Señales digitales

- * Esta es una razón clave por la que los computadores utilizan números binarios, ya que un sistema binario se corresponde directamente con la abstracción subyacente a la electrónica
- * En las diferentes implementaciones electrónicas los voltajes y sus relaciones difieren,
- * Por lo que no se utiliza el valor del voltage sino una indicación de si la señal esta activada o no (verdadera o no, 1 o 0, etc).

» Señales digitales

- * Por eso hablamos de señales que son:
 - * (lógicamente) ciertas, ó 1, ó afirmadas, asertadas
 - * (lógicamente) falsas, ó 0, ó negadas
- * Los valores 0 y 1 reciben el nombre de complementarios o inversos el uno del otro

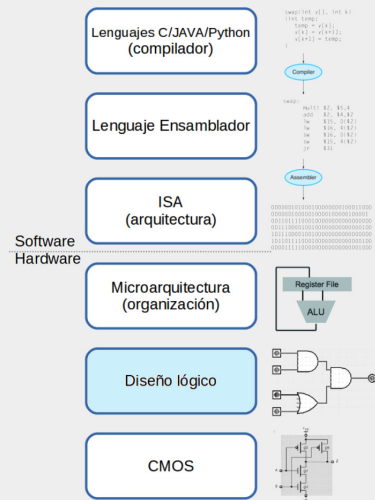
Diseño Lógico - Diseño Digital

- * Señales digitales
- * **Análisis de Circuitos Digitales**
- * Metodología para realizar un diseño lógico/digital
- * Circuitos combinacionales comunes
- * Relojes
- * Circuitos secuenciales
- * Máquinas de Estado Finito (FSM)

» Diseño lógico o digital

Los bloques lógicos (circuitos digitales o lógicos) se clasifican en dos tipos:

- * **circuitos combinacionales**
 - * Sus salidas dependen sólo de las entradas
- * **circuitos secuenciales**
 - * Mantienen un estado interno. Sus salidas pueden depender tanto de las entradas actuales como del valor almacenado en memoria, conocido como estado del bloque
 - * Permiten modelar memorias y **máquinas de estados finitos**
 - * Las máquinas de estado finito permiten modelar **máquinas algorítmicas** (por ej. una CPU)

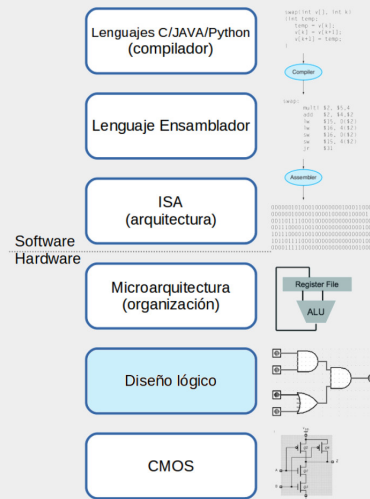
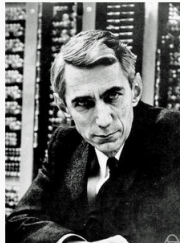


Diseño lógico o digital

Booleano En 1854 George Boole define un algebra que utiliza sólo dos valores.

Switching En 1938 Claude Shannon demuestra que el algebra de boole puede ser utilizado para analizar y modelar circuitos digitales.

Claude Elwood Shannon



» Puertas, Tablas de Verdad, Ecuaciones Lógicas

Tablas de Verdad

- * Debido a que un bloque de lógica combinatoria no contiene memoria, puede especificarse completamente definiendo los valores de las salidas para cada posible conjunto de valores de entrada
- * Dicha descripción se da normalmente en forma de tabla de verdad

- * Para un bloque lógico con n entradas, existen 2^n posiciones en la tabla de verdad, puesto que este es el número de combinaciones posible de los valores de entrada
- * Cada posición en la tabla especifica el valor de todas las salidas para una combinación particular de las entrada
- * Las tablas de verdad pueden **describir completamente cualquier función lógica** combinatoria
- * Sin embargo, su tamaño **crece rápidamente** y puede dificultar su comprensión

» Puertas, Tablas de Verdad, Ecuaciones Lógicas

Álgebra de Boole - Leyes y Teoremas

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\bar{A} + \bar{B} = \overline{AB}$

» Puertas, Tablas de Verdad, Ecuaciones Lógicas

Álgebra de Boole - Leyes y Teoremas

- * Cualquier función lógica puede ser reescrita como una ecuación, con la salida en la parte izquierda de la igualdad, y una función de las variables de entrada utilizando las operaciones del álgebra a la derecha.
- * Ejemplo:

$$E = ((A \cdot B) + (A \cdot C) + (B \cdot C)) \cdot (\overline{A \cdot B \cdot C})$$

We can also derive E by realizing that E is true only if exactly two of the inputs are true. Then we can write E as an OR of the three possible terms that have two true inputs and one false input:

$$E = (A \cdot B \cdot \overline{C}) + (A \cdot C \cdot \overline{B}) + (B \cdot C \cdot \overline{A})$$

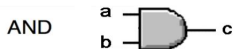
» **Diseño digital**

Compuertas/Puertas (Gates)

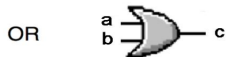
- * Los bloques lógicos se construyen a partir de compuertas (puertas) lógicas que realizan las funciones lógicas básicas como AND, OR y NOT
- * Una puerta AND o una OR pueden tener múltiples entradas, con la salida igual a la AND o la OR de todas ellas
- * La función lógica NOT se realiza mediante un inversor que siempre tiene una entrada

» Diseño digital

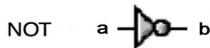
Compuertas/Puertas (Gates)



ab	c
00	0
01	0
10	0
11	1



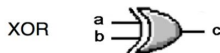
ab	c
00	0
01	1
10	1
11	1



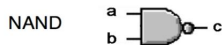
a	b
0	1
1	0

» **Diseño digital**

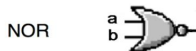
Compuertas/Puertas (Gates)



ab	c
00	0
01	1
10	1
11	0



ab	c
00	1
01	1
10	1
11	0



ab	c
00	1
01	0
10	0
11	0

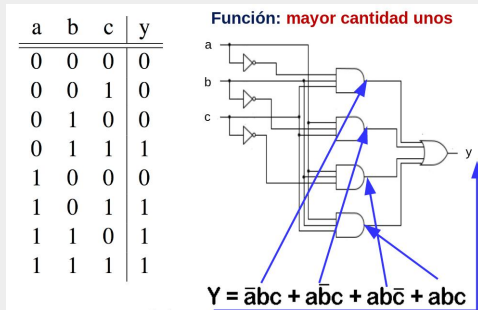
Diseño Lógico - Diseño Digital

- * Señales digitales
- * Análisis de Circuitos Digitales
- * Metodología para realizar un diseño lógico/digital
- * Circuitos combinacionales comunes
- * Relojes
- * Circuitos secuenciales
- * Máquinas de Estado Finito (FSM)

» Diseño digital

Metodología para realizar un diseño lógico/digital de un circuito combinacional

1. Definir coloquialmente todas las entradas y todas las salidas
2. Confeccionar una tabla de verdad para todas las combinaciones de valores de entrada y todas las salidas
3. Describir para cada salida una función lógica en base a la tabla
4. Confeccionar un diagrama del circuito digital resultante

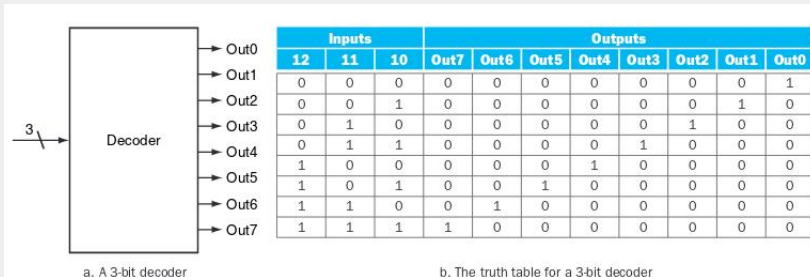


Diseño Lógico - Diseño Digital

- * Señales digitales
- * Análisis de Circuitos Digitales
- * Metodología para realizar un diseño lógico/digital
- * **Circuitos combinacionales comunes**
- * Relojes
- * Circuitos secuenciales
- * Máquinas de Estado Finito (FSM)

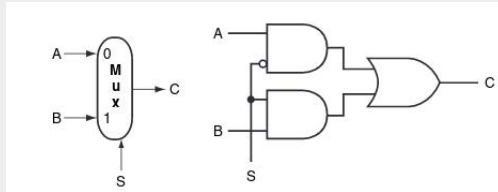
» Diseño digital

Construcción de diseño lógico/digital: Decoder



» **Diseño digital**

Construcción de diseño lógico/digital: Multiplexor



Construcción de diseño lógico/digital: Multiplexor 32 bits



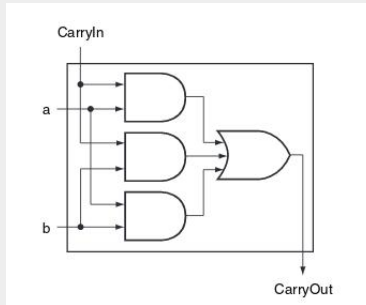
» Diseño digital

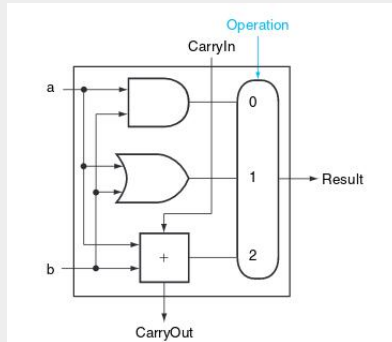
Diseño de ALU de un bit

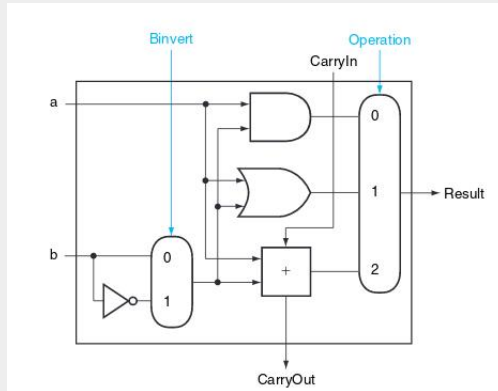
Inputs			Outputs		Comments
a	b	CarryIn	CarryOut	Sum	
0	0	0	0	0	$0 + 0 + 0 = 00_{\text{two}}$
0	0	1	0	1	$0 + 0 + 1 = 01_{\text{two}}$
0	1	0	0	1	$0 + 1 + 0 = 01_{\text{two}}$
0	1	1	1	0	$0 + 1 + 1 = 10_{\text{two}}$
1	0	0	0	1	$1 + 0 + 0 = 01_{\text{two}}$
1	0	1	1	0	$1 + 0 + 1 = 10_{\text{two}}$
1	1	0	1	0	$1 + 1 + 0 = 10_{\text{two}}$
1	1	1	1	1	$1 + 1 + 1 = 11_{\text{two}}$

» Diseño digital

Diseño de ALU de un bit

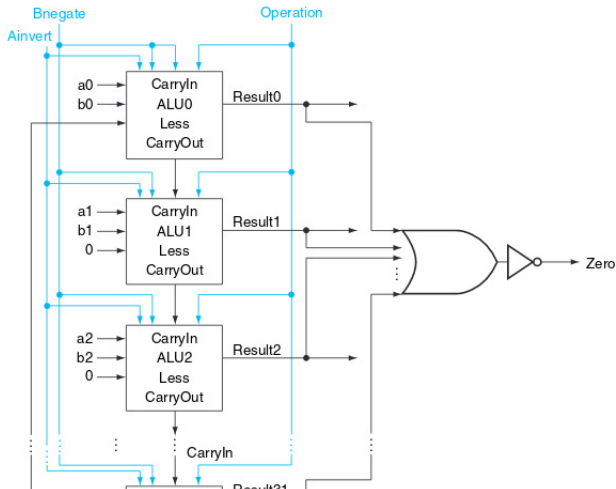


» **Diseño digital****Diseño de ALU de un bit**

» **Diseño digital****Diseño de ALU de un bit**

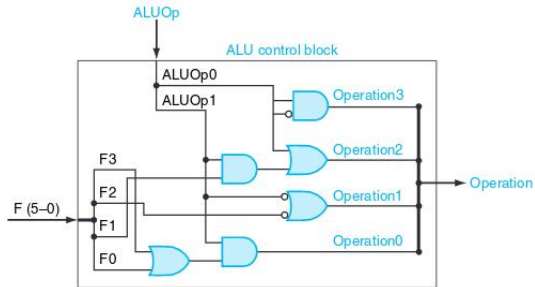
» Diseño digital

Diseño de ALU de 32 bits



» Diseño digital

Diseño de una unidad de control



Diseño Lógico - Diseño Digital

- * Señales digitales
- * Análisis de Circuitos Digitales
- * Metodología para realizar un diseño lógico/digital
- * Circuitos combinacionales comunes
- * Relojes
- * Circuitos secuenciales
- * Máquinas de Estado Finito (FSM)

» Diseño digital

Relojes

- * Los relojes son necesarios en la lógica secuencial, para decidir cuando un elemento que contiene un estado debe ser actualizado.
- * Terminología: tiempo del ciclo (período), frecuencia del reloj.
- * Metodología : edge-triggered clocking (reloj disparado por flanco)

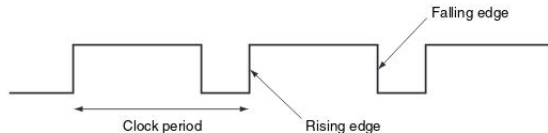


FIGURE C.7.1 A clock signal oscillates between high and low values. The clock period is the time for one full cycle. In an edge-triggered design, either the rising or falling edge of the clock is active and causes state to be changed.

Relojes

-
- Figure C.7.2 illustrates the operation of a sequential logic circuit. The top part is a block diagram showing a feedback loop: State element 1 feeds into Combinational logic, which feeds into State element 2, which then feeds back into State element 1. The bottom part is a timing diagram. It shows a clock cycle (a square wave) and the output of State element 2. The output of State element 2 changes only at the rising edges of the clock cycle, demonstrating that the clock edge determines when the contents of the state elements are updated.

Diseño Lógico - Diseño Digital

- * Señales digitales
- * Análisis de Circuitos Digitales
- * Metodología para realizar un diseño lógico/digital
- * Circuitos combinacionales comunes
- * Relojes
- * Circuitos secuenciales
- * Máquinas de Estado Finito (FSM)

» Diseño digital

Elementos de memoria (circuitos secuenciales)

- * Set-Reset Latch
- * Elemento básico para almacenar un bit. Contiene un Loop en su diseño.

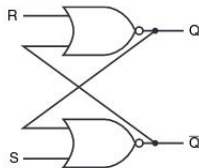


FIGURE C.8.1 A pair of cross-coupled NOR gates can store an internal value. The value stored on the output Q is recycled by inverting it to obtain \bar{Q} and then inverting \bar{Q} to obtain Q . If either R or \bar{Q} is asserted, Q will be deasserted and vice versa.

» **Diseño digital**

Elementos de memoria (circuitos secuenciales)

- * D-Latch
- * Elemento básico para almacenar un bit. Contiene un Loop en su diseño
- * Sus entradas son el bit a almacenar y la señal de reloj

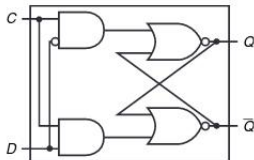


FIGURE C.8.2 A D latch implemented with NOR gates. A NOR gate acts as an inverter if the other input is 0. Thus, the cross-coupled pair of NOR gates acts to store the state value unless the clock input, *C*, is asserted, in which case the value of input *D* replaces the value of *Q* and is stored. The value of input *D* must be stable when the clock signal *C* changes from asserted to deasserted.



» **Diseño digital**

Elementos de memoria (circuitos secuenciales)

- * D Flip-Flop : utilizados en la construcción de REGISTROS
- * Elemento básico para almacenar un bit. Contiene un Loop en su diseño
- * Sus entradas son el bit a almacenar y la señal de reloj
- * Se sincroniza (actualiza) en el flanco de reloj

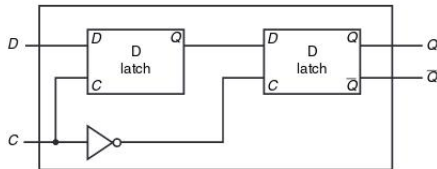


FIGURE C.8.4 A D flip-flop with a falling-edge trigger. The first latch, called the master, is open and follows the input D when the clock input, C , is asserted. When the clock input, C , falls, the first latch is closed, but the second latch, called the slave, is open and gets its input from the output of the master latch.

Elementos de memoria (circuitos secuenciales)

-
- FIGURE C.8.5 Operation of a D flip-flop with a falling-edge trigger, assuming the output is initially deasserted.** When the clock input (C) changes from asserted to deasserted, the Q output stores the value of the D input. Compare this behavior to that of the clocked D latch shown in Figure C.8.3. In a clocked latch, the stored value and the output, Q , both change whenever C is high, as opposed to only when C transitions.

» **Diseño digital**

Elementos de memoria (circuitos secuenciales)

- * D Flip-Flop
- * Si la señal D cambia cuando la señal de reloj está baja (desacertada) el estado interno del flip-flop no cambia. Eso lo distingue de un latch.

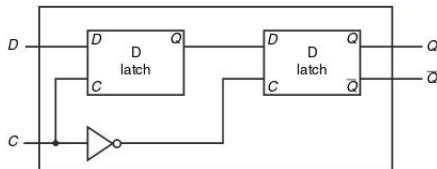
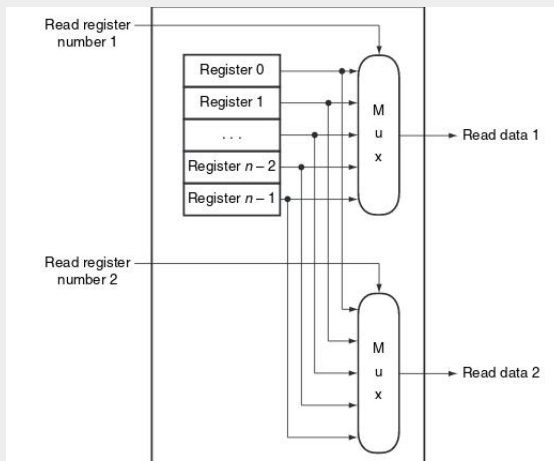


FIGURE C.8.4 A D flip-flop with a falling-edge trigger. The first latch, called the master, is open and follows the input D when the clock input, C , is asserted. When the clock input, C , falls, the first latch is closed, but the second latch, called the slave, is open and gets its input from the output of the master latch.

» **Diseño digital**

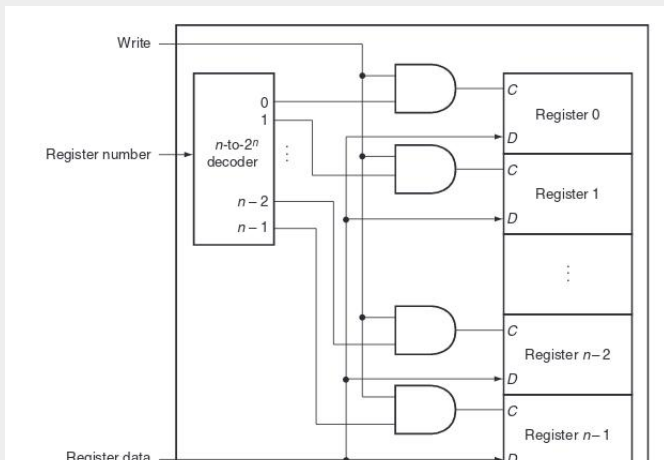
Diseño de ALU y Registros

- * Elementos de estado, multiplexores, decodificadores, ALU



Diseño de ALU y Registros

- * Elementos de estado, multiplexores, decodificadores, ALU



Diseño Lógico - Diseño Digital

- * Señales digitales
- * Análisis de Circuitos Digitales
- * Metodología para realizar un diseño lógico/digital
- * Circuitos combinacionales comunes
- * Relojes
- * Circuitos secuenciales
- * Máquinas de Estado Finito (FSM)

» **Diseño digital**

Máquinas de Estado Finito (FSM)

- * Permiten especificar circuitos secuenciales
- * Ejemplo: control de la calefacción de un coche

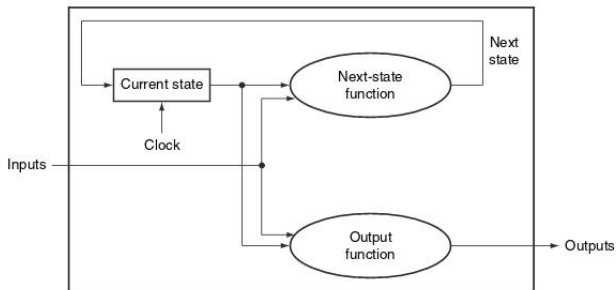


FIGURE C.10.1 A state machine consists of internal storage that contains the state and two combinational functions: the next-state function and the output function. Often, the output function is restricted to take only the current state as its input; this does not change the capability of a sequential machine, but does affect its internals.

Máquinas de Estado Finito (FSM)

- * Autómata de Mealy
- * Autómata de Moore

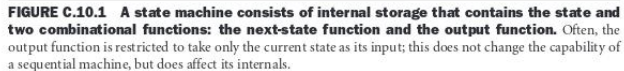
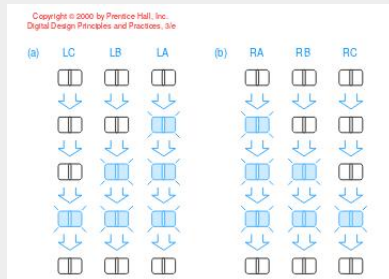


Diagram illustrating the rear view of a 1966 Ford Thunderbird, focusing on the tail light assembly. The car is shown from the rear, with the license plate reading "CALIFORNIA ZOTTFFS". The tail light assembly consists of two main sections, each containing six rectangular light segments. Labels LC, LB, and LA point to the left section, while RA, RB, and RC point to the right section. The diagram is a line drawing with blue lines indicating the specific light segments.

» Diseño digital

Máquinas de Estado Finito (FSM)

- * Ejemplo: Luces de giro del Ford Thunderbird



» Diseño digital

Máquinas de Estado Finito (FSM)

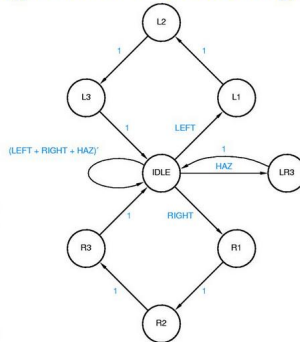
- * Ejemplo: Luces de giro del Ford Thunderbird
- * Diagrama de estados

Initial state diagram for T-bird tail lights

Copyright © 2000 by Prentice Hall, Inc.
Digital Design Principles and Practices, 3/e

Output Table

State	LC	LB	LA	RA	RB	RC
IDLE	0	0	0	0	0	0
L1	0	0	1	0	0	0
L2	0	1	1	0	0	0
L3	1	1	1	0	0	0
R1	0	0	0	1	0	0
R2	0	0	0	1	1	0
R3	0	0	0	1	1	1
LR3	1	1	1	1	1	1



» Diseño digital

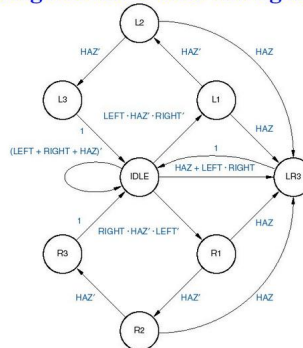
Máquinas de Estado Finito (FSM)

- * Ejemplo: Luces de giro del Ford Thunderbird
- * Diagrama de estados (segunda versión)

Enhanced state diagram for T-bird tail lights

Inputs:
LEFT, RIGHT, HAZ

Outputs:
Six lamps
(function of state only)



» Consejos y preguntas

* ¿Preguntas?

» Bibliografia

Libros

- * David. Patterson John L. Hennessy (1995), ORGANIZACIÓN Y DISEÑO DE COMPUTADORES La interfaz hardware/software, McGraw-Hill (8 copias en biblioteca).