

RESUMEN DEL SEMINARIO 5: INTRODUCCIÓN AL MANEJO DE UN SIMULADOR Y UN ENTRENADOR LÓGICO

Tecnología y Organización de Computadores

Grado en Ingeniería Informática

1. Introducción al Simulador Lógico: Logic Works

1.1. Funcionamiento y Entorno de Trabajo

[cite_sstart]Logic Worksesunentornodetrabajodigitaldiseadoparaeldiseño y simulación de circuitos digitales. [cite_sstart]Elinterfazdetrabajosedivideenvariasreasclave[cite : 85, 90, 101] :

[cite_sstart]

- 0. **Ventana de Diseño (Ventana Principal):** Es el área de trabajo donde el usuario introduce el esquemático del circuito digital a diseñar y simular[cite: 86]. [cite_sstart]
- 0. **Paleta de Componentes (Parts Palette):** Ubicada a la derecha, contiene la biblioteca de componentes ordenados por funcionalidad[cite: 90, 92]. [cite_sstart]Desdeaqueselecciona los componentes. [cite_sstart]
- 0. **Ventana de Visualización de Formas de Onda (Timing Window):** Situada en la parte inferior, se utiliza para la simulación temporal, permitiendo comprobar el funcionamiento del circuito a lo largo del tiempo mediante cronogramas[cite: 101].

1.2. Selección y Conexión de Componentes

[cite_sstart]

Bibliotecas Esenciales [cite: 96, 97]

Logic Works organiza sus componentes en bibliotecas. Las más importantes son:

- **Simulation Gates:** Contiene puertas lógicas básicas (AND, OR, NOT, XOR, etc.) con diverso número de entradas.
- **Simulation Logic:** Incluye circuitos digitales sencillos, como Latches, biestables, registros, contadores y sumadores.
- **Simulation IO (Input/Output):** Proporciona dispositivos para la interacción con el circuito durante la simulación, como **BINARY SWITCH** (conmutadores binarios), **HEX KEYBOARD**, **BINARY PROBE** (sondas binarias) y **HEX DISPLAY**.

- **Connect:** Contiene elementos como **PORT IN** y **PORT OUT**, esenciales para la definición de subcircuitos.

Métodos de Conexionado

[cite_start] *El conexionado entre componentes se puede realizar de dos maneras* [cite : 187] :
[cite_start]

Conexión por Arrastre: Consiste en situar el cursor cerca del pin de origen y arrastrar el ratón hasta el pin de destino para simular una conexión eléctrica directa [cite: 188, 189]. [cite_start]

Conexión por Etiquetado (*Labeling*): Se etiquetan los pines utilizando la opción **Edit** → **Text** [cite: 191]. [cite_start] *Do pines con la misma etiqueta (que aparece en color rosado)* 190, 192, 193].

2. Simulación y Análisis Temporal en Logic Works

[cite_start] *La simulación en Logic Works se puede abordar en dos planos distintos* [cite : 203] :

2.1. Simulación Lógica Funcional

[cite_start] *Esta simulación se realiza directamente en la Ventana de Diseño utilizando componentes de E/* 204]. [cite_start] *El usuario interactúa haciendo clic en los conmutadores de entrada y observa inmediatamente* 205]. *Este método se usa para comprobar la tabla de verdad del circuito.*

2.2. Simulación Temporal (Cronogramas)

[cite_start] *Es una herramienta muy potente que permite representar gráficamente las formas de onda de las* 207].

Definición de Estímulos

[cite_start] *Los estímulos (entradas) se definen temporalmente a través de un fichero de estímulos con extensión* 210]. [cite_start] *El formato del fichero debe ser tabulado y constar de las siguientes columnas esenciales* [cite : 211] :

T: Instante inicial de tiempo.

D: Duración del periodo de tiempo.

I Nombre_ñal_entrada: Valor lógico (0 o 1) de cada señal de entrada en ese periodo.

[cite_start] *Es crucial eliminar las entradas lógicas (BINARY SWITCH) del diseño antes de importarlo* 218].

Control de la Simulación

[cite_start]Una vez insertados los estmulos, la simulación se controla desde la **Simulator Toolbar** [cite : 220]. Los botones principales son :

[cite_start]

Reset: Borra las formas de onda para empezar una nueva simulación [cite: 221]. [cite_start]

Run: Ejecuta la simulación de forma continua [cite: 222]. [cite_start]

Step: Ejecuta la simulación paso a paso [cite: 222]. [cite_start]

Clear X: Fuerza al simulador a asignar valores iniciales a las señales, reemplazando el estado *desconocido* ('X') [cite: 223].

[cite_start] *LogicWorks maneja 5 estados lógicos posibles para las señales* [cite : 227] :

Valor	Significado
0	Valor lógico 0
1	Valor lógico 1
X	Desconocido (0 ó 1)
Z	Alta impedancia (desconexión eléctrica)
C	Conflicto (valores contradictorios en la señal)

[cite_start] La simulación temporal permite observar los **retardos** en las señales de salida frente a las de entrada [cite: 232, 233].

3. Generación de Símbolos para Subcircuitos

Un subcircuito permite encapsular un diseño complejo en un único componente, reutilizable en otros esquemas.

[cite_start] *El proceso consta de dos pasos principales* [cite : 236] :

3.1. Paso 1: Definición de Puertos de E/S

[cite_start]

0. **Reemplazo de E/S:** Los conmutadores (BINARY SWITCH) se reemplazan por **PORT IN** y los visualizadores (BINARY PROBE) por **PORT OUT** (ambos de la biblioteca **Connect**) [cite: 238]. [cite_start]

0. **Etiquetado de Puertos:** Se etiquetan los PORT IN/OUT con los nombres de las variables de entrada y salida del subcircuito (ej. A0, B0, S0, C1) [cite: 240, 241].

3.2. Paso 2: Creación y Guardado del Símbolo

[cite_start]

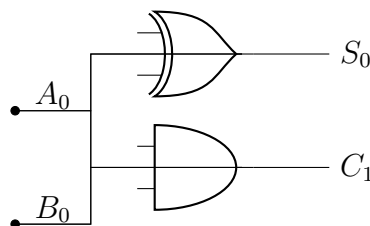
0. **Archivo de Diseño:** El circuito con los puertos definidos se guarda en un archivo (ej., `circuito2.cct`)[cite: 242]. [cite_start]
0. **Creación del Símbolo:** Se accede al editor de dispositivos (`File` → `New` → `Device Symbol`)[cite: 243]. [cite_start] *Se selecciona la opción para crear un nuevo símbolo*[cite: 245]. [cite_start]
0. **Generación Automática:** Mediante `Options` → `Autocreate symbol` y `Extract Pin List`, Logic Works genera automáticamente el símbolo, colocando las entradas a la izquierda y las salidas a la derecha[cite: 247, 248]. [cite_start] *Se asigna un nombre (Part Name) al componente*[cite: 249].
0. **Guardado en Biblioteca:** El nuevo símbolo se guarda en un fichero de biblioteca con extensión `.clf`. [cite_start] *Se recomienda crear una biblioteca para los componentes*[cite: 253, 254].

3.3. Ejemplo: Implementación de un Semisumador

El circuito de ejemplo utilizado en el seminario es un **Semisumador** (Half-Adder), que suma dos bits A_0 y B_0 y produce una suma S_0 y un acarreo C_1 .

$$S_0 = A_0 \oplus B_0$$

$$C_1 = A_0 \cdot B_0$$



4. Introducción al Entrenador Lógico SIDAC DET 2020

[cite_start] *El SIDAC DET 2020 es un entrenador de prácticas diseñado para la realización de circuitos lógicos*[cite: 270]. [cite_start] *Contiene las principales funciones lógicas (puertas, biestables, contadores) ya alimentadas*[cite: 271, 272].

4.1. Recursos y Componentes Integrados

[cite_start]Elentrenadorofreceunaampliavariedaddecomponentesyrecursosdecontrol[cite : 298, 299] :

Funciones Lógicas Integradas

[cite_start]

- **Puertas Lógicas:** 6 Inversores, 8 NAND-2, 3 NAND-3, 4 NAND-4, 4 EXOR-2, 4 AND-2, 4 OR-2, 4 NOR-2[cite: 298]. [cite_start]
- **Circuitos MSI/LSI:** 8 Flip-Flops JK, 2 Circuitos AND-OR-INVERT (AOI), 2 Decodificadores BCD-7 Segmentos, 2 Contadores Bidireccionales[cite: 298].

Recursos de Control y Señalización

[cite_start]

- **Indicadores de Estado Lógico:** Consisten en **LEDs** que se encienden para un nivel lógico alto (Hi) y permanecen apagados para un nivel bajo (Lo) o desconexión[cite: 278, 279].
- **Reloj (Clock):** Genera pulsos de reloj. [cite_start]Su frecuencia es controlable mediante [cite: 280, 281]. [cite_start]
- **Generadores de Pulsos:** Incluye 2 generadores de pulsos **sin rebote**[cite: 299]. [cite_start]
- **Conmutadores Programables:** Permiten obtener hasta ocho salidas independientes con niveles lógicos Hi (1) o Lo (0) fijos, actuando como entradas de usuario[cite: 282, 289].

4.2. Recursos de Interconexión y Alimentación

Islas de Conexión

[cite_start]Elentrenadorutilizadiferentestiposdeislaspara facilitar la interconexión y la alimentación[cite : 294, 296] :

Islas de Nivel Fijo: Marcadas como Hi (para nivel alto fijo) y Lo (para nivel bajo fijo/tierra).

Islas Libres: No están conectadas a ningún potencial fijo y se utilizan para **distribuir una señal** a varios puntos del circuito.

Zócalos Auxiliares

[cite_{start}]El entrenador dispone de cuatro zócalos de 16 patillas para la utilización de [cite_{end} 276, 290].

[cite_{start}]

Advertencia de Conexión: Las patillas de estos zócalos **no están conectadas** a alimentación o tierra por defecto [cite: 291]. [cite_{start}]

Conexión Requerida: Es necesario conectarlos manualmente utilizando las islas de alimentación [cite: 292]:

- +5- para circuitos **TTL**.
- +VA- para tensiones distintas, conectadas a alimentaciones externas.

5. Realizaciones Prácticas Comunes

Las prácticas en el entrenador y el simulador se centran en la implementación de funciones lógicas minimizadas y en la creación de equivalencias entre puertas.

5.1. Implementación de Funciones a partir de Expresiones Mínimas

Una práctica fundamental es la implementación de la **Forma Mínima Disyuntiva (AND/OR)** de una función de conmutación.

5.2. Realización de Funciones con Puertas Universales

Implementación de Funciones Básicas con Puertas Universales (Seminario 5, p. 32)

Función	Usando solo puertas NAND-2	Usando solo puertas NOR-2
NOT (\bar{A})	Uniendo las entradas de una puerta NAND-2. $\overline{A \cdot A} = \bar{A}$.	Uniendo las entradas de una puerta NOR-2. $\overline{A + A} = \bar{A}$.
AND-2 ($A \cdot B$)	$\overline{\overline{A \cdot B}}$ de una NAND-2. $\overline{\overline{A \cdot B}} = A \cdot B$.	Usando el teorema de De Morgan. $\overline{\overline{A} + \overline{B}} = A \cdot B$.
OR-2 ($A + B$)	Usando el teorema de De Morgan. $\overline{\overline{A} \cdot \overline{B}} = A + B$.	$\overline{\overline{A + B}}$ de una NOR-2. $\overline{\overline{A + B}} = A + B$.