

ELECTRÓNICA DIGITAL

Tema 16

LENGUAJES DE DESCRIPCIÓN DE LOS SISTEMAS DIGITALES (PARTE 1)

FUNDAMENTOS

IMPORTANCIA DE LOS ESQUEMAS

Los circuitos y sistemas electrónicos digitales cuya complejidad es limitada, se pueden describir mediante símbolos lógicos interconectados que dan lugar a lo que en ingeniería se conoce como un esquema (*Schematic*).

Los esquemas, en especial cuando la representación de sus elementos se normaliza, constituyen una herramienta de gran utilidad para transferir los conceptos básicos de cualquier tecnología a la memoria de larga duración del que trata de aprenderla.

INCONVENIENTES DE LOS ESQUEMAS

La realización de un esquema es cada vez más difícil, e incluso llega a hacerse imposible, conforme se eleva la complejidad de los sistemas digitales. Ello incentivó el desarrollo de lenguajes que facilitasen la descripción de un sistema digital mediante un texto, su edición mediante un computador y su conversión en la información adecuada para implementarlo en un circuito integrado. Reciben el nombre de lenguajes de descripción de sistemas digitales y se les conoce como HDL (acrónimo de *Hardware Description Languages*).

FUNDAMENTOS

CUANDO COMENZARON LOS HDL

La necesidad de desarrollar lenguajes que facilitasen la descripción de los sistemas digitales complejos comenzó a hacerse patente a principios de la década de 1970 cuando el sistema físico (*Hardware*) de los procesadores digitales programables alcanzó un grado de complejidad elevado que hizo prácticamente imposible su descripción mediante un esquema.

COMO COMENZARON LOS HDL

Debido a ello nació un tipo de lenguaje que describe de forma simbólica y sencilla las acciones que es capaz de realizar un sistema digital. Dicho lenguaje recibió el nombre de lenguaje de transferencias entre registros, conocido como RTL (acrónimo de *Register Transfer Language*), debido a que la mayoría de las acciones que describe son transferencias de información entre registros.

FUNDAMENTOS

LENGUAJE RTL

Conjunto de símbolos y de reglas que describen la relación entre las acciones que ejecuta un sistema digital y su sistema físico (*Hardware*).

DESARROLLO DE LOS LENGUAJES RTL

Existen infinitas formas de concebir para definir los elementos de memoria y las transferencias de información un lenguaje RTL que se diferencian solo por las reglas utilizadas entre ellos.

No existe un único lenguaje RTL sino que se desarrollaron tantos como grupos de I + D de universidades y empresas [ZAIN 98] (en especial en USA) fueron conscientes de las ventajas que su utilización les proporcionaría para elevar su capacidad de diseño de sistemas digitales complejos.

•Dos de los lenguajes RTL más utilizados fueron:

**AHPL (acrónimo de *A Hardware Programming Language*) [HILL 73] |
DDL (acrónimo de *Digital Design Language*) [BREU 75].**

FUNDAMENTOS

LENGUAJES DE TRANSFERENCIA ENTRE REGISTROS

RTL (acrónimo de *Register Transfer Language*)

Principales elementos de un lenguaje RTL:

- **Identificadores (*Identifiers*)**
- **Operadores (*Operators*)**
- **Declaraciones o definiciones (*Statements*)**

FUNDAMENTOS

LENGUAJES RTL

Identificadores (*Identifiers*)

Se utilizan para denominar a un recurso físico y están constituidos por un conjunto de símbolos, una o más letras y números decimales, el primero de los cuales es siempre una letra.

Elemento	Lenguaje RTL
Valor numérico	K
Registros	X, Y
Contenido del registro X	(X)
Biestables	u,w
Biestable u del registro X	(X<u>)
Biestables u a w del registro X	(X<u:w>)
Asignación	⇒

Tabla 9.1 Identificadores utilizados en el lenguaje RTL de los microcontroladores PIC.

FUNDAMENTOS LENGUAJES RTL

Operadores (*Operators*)

Establecen las operaciones que ejecuta la unidad operativa con los contenidos de los diferentes elementos de memoria.

Operación	Operador
Operación aritmética en general	*
Suma Aritmética	+
Resta	-
Operación lógica en general	LOG
Comparación	==
Y lógica	AND
O lógica	OR
O exclusiva	XOR
Complemento a uno del contenido del registro X	(/X)

FUNDAMENTOS LENGUAJES RTL

Declaraciones o definiciones (*Statements*)

Especifican las acciones que realiza un sistema digital.

Acción	Lenguaje RTL
Carga el valor A en el registro X	$A \Rightarrow (X)$
Transfiere el contenido del registro X al registro Y	$(X) \Rightarrow (Y)$
Realiza la operación aritmética * entre el valor A y el contenido del registro X y almacena el resultado en el registro X o Y	$(X) * A \Rightarrow (X)$ $(X) * A \Rightarrow (Y)$
Realiza la operación aritmética * entre los contenidos de los registros X e Y y almacena el resultado en el registro X o Y	$(X) * (Y) \Rightarrow (X)$ $(X) * (Y) \Rightarrow (Y)$
Realiza la operación lógica LOG entre A y el contenido del registro X y almacena el resultado en el registro X o Y	$(X) \text{ LOG } A \Rightarrow (X)$ $(X) \text{ LOG } A \Rightarrow (Y)$
Realiza la operación lógica LOG entre los contenidos de los registros X e Y y almacena el resultado en el registro X o Y	$(X) \text{ LOG } (Y) \Rightarrow (X)$ $(X) \text{ LOG } (Y) \Rightarrow (Y)$
Pone a cero el biestable u	$0 \Rightarrow u$
Pone a uno el biestable u	$1 \Rightarrow u$
Pone a cero el biestable u del registro X	$0 \Rightarrow (X \langle u \rangle)$
Pone a uno el biestable u del registro X	$1 \Rightarrow (X \langle u \rangle)$
Compara el contenido del registro X y el valor A	$[(X) == A]$
Compara con cero el biestable u del registro X	$[(X \langle u \rangle) == 0]$
Compara con uno el biestable u del registro X	$[(X \langle u \rangle) == 1]$

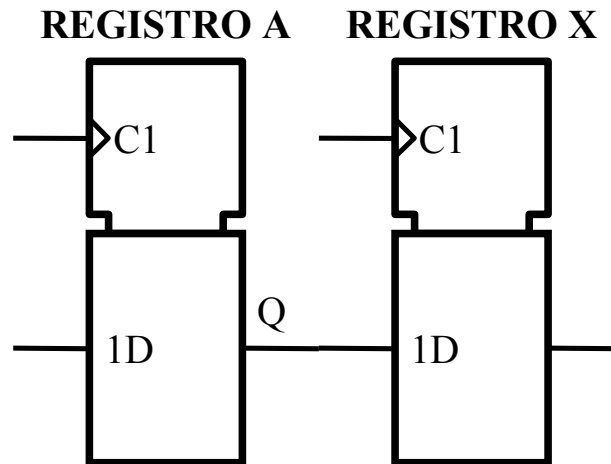
FUNDAMENTOS

LENGUAJES RTL

Los lenguajes RTL permiten llevar a cabo la descripción del sistema físico (*Hardware*) [SHIV 00] de un sistema digital porque establecen los diferentes conceptos asociados con su funcionamiento, que son los siguientes:

- El conjunto de elementos de memoria.**
- Las operaciones entre el contenido de los elementos de memoria.**
- Las tomas de decisión que establecen la secuencia de operaciones que ejecuta.**

FUNDAMENTOS LENGUAJES RTL

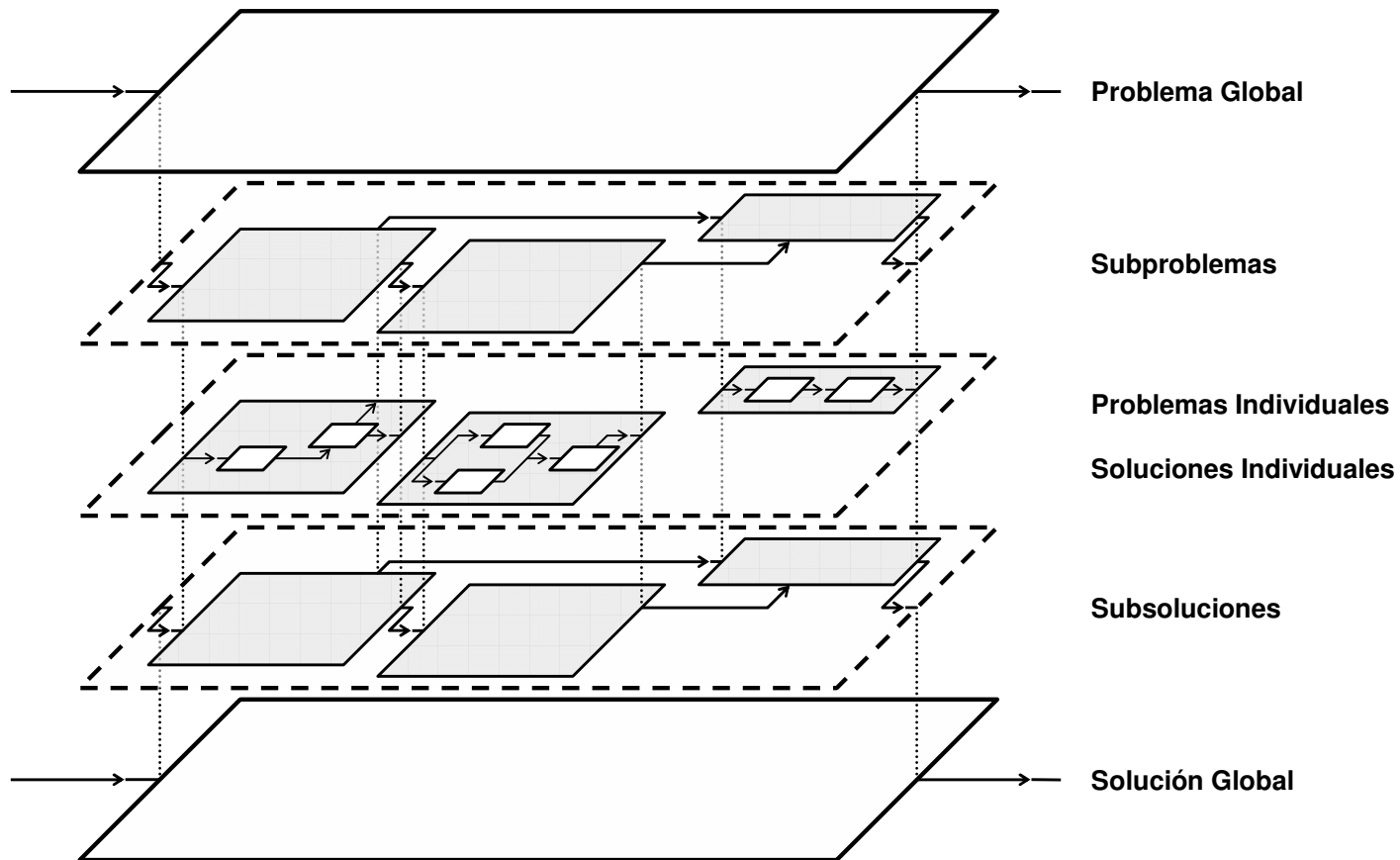


Por ejemplo la acción $A \Rightarrow (X)$ equivale al circuito representado en la figura que establece una conexión entre la salida del registro A y la entrada del registro X para que el contenido del primero pueda ser transferido al segundo.

Los lenguajes RTL constituyen actualmente, bajo la denominación de modelado mediante el flujo de datos (*Data Flow Modeling*), una de las formas de describir los sistemas digitales incluidas en los lenguajes estructurados de descripción de sistemas digitales.

FUNDAMENTOS

DESCRIPCIÓN GRÁFICA DEL MODELO DE DISEÑO DE SISTEMAS COMPLEJOS VDI 2221



FUNDAMENTOS

SISTEMAS DIGITALES COMPLEJOS

Antes de realizar un prototipo de un sistema digital complejo hay que comprobar que funciona de acuerdo con las especificaciones (*Requirements*). Esto da lugar a que su proceso de diseño se pueda dividir de forma simplificada en tres fases:

- La descripción del sistema**
- La verificación del funcionamiento del sistema**
- La implementación del sistema**

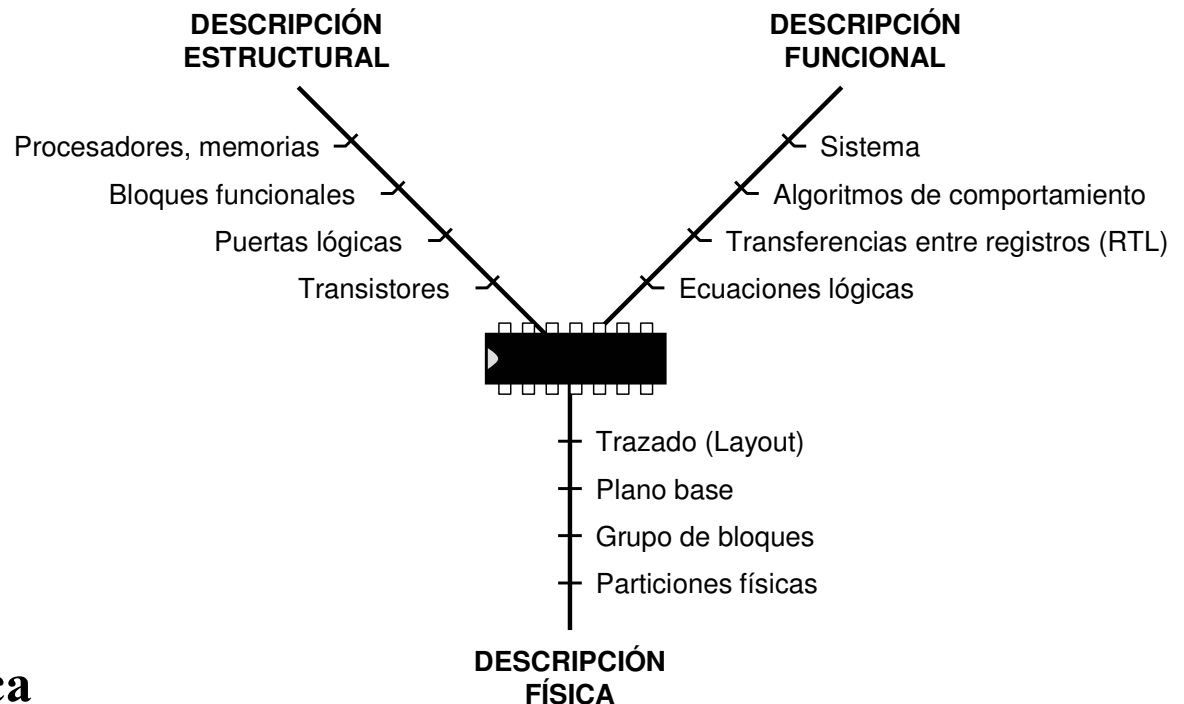
FUNDAMENTOS

DESCRIPCIÓN DE LOS SISTEMAS DIGITALES COMPLEJOS

- Descripción estructural
(*Structural modeling*)
Especifica los elementos y sus interconexiones

- Descripción funcional o de comportamiento
(*Behavioural modeling*)
Describe el funcionamiento

- Descripción física o geométrica
(*Physical design*)
Genera los datos para la realización física



FUNDAMENTOS

LENGUAJES DE DESCRIPCIÓN DE LOS SISTEMAS DIGITALES

HDL (acrónimo de *Hardware Description Language*)

Existen múltiples formas de describir los sistemas digitales complejos, que tienen como objetivo facilitar su implementación y la comprobación de su funcionamiento mediante herramientas (*Tools*) de diseño asistido por computador conocido como CAD (*Computer Aided Design*) o EDA (*Electronic Design Automation*).

LENGUAJES HDL

- HDL sencillos no estructurados orientados a la realización de un único circuito o módulo por fichero.**
- HDL estructurados o complejos que permiten definir submódulos y enlazarlos jerárquicamente en un único fichero.**

FUNDAMENTOS

LENGUAJES HDL ESTRUCTURADOS

Infinidad de maneras de concebir un lenguaje HDL estructurado: necesidad de normalización.

El punto de partida de los primeros HDL estructurados lo constituyeron, tal como se indica en el apartado 9.2, los lenguajes de transferencia de información entre registros denominados RTL (*Register Transfer Language*) [BREU 75] [HILL 73], que se extendieron por un lado hacia el nivel de algoritmo y por otro hacia el de una estructura realizada con puertas lógicas y bloques o módulos funcionales.

El Departamento de Defensa (DOD) de USA inició en 1980 un proyecto denominado "*Very High Speed Integrated Circuit*" (VHSIC), con el principal objetivo de armonizar el desarrollo de circuitos integrados de muy altas prestaciones para aplicaciones militares. Para llevar a cabo esta tarea fue necesario crear un lenguaje HDL que facilitara el flujo de información entre diseñadores, fabricantes y usuarios. De esta forma, en otoño de 1980, se inició el desarrollo del VHSIC (acrónimo de *Very High Speed Integrated Circuit Hardware Design/ Description Language*), que se denominó inicialmente VHD2L, y pasó posteriormente a denominarse simplemente VHDL.

FUNDAMENTOS

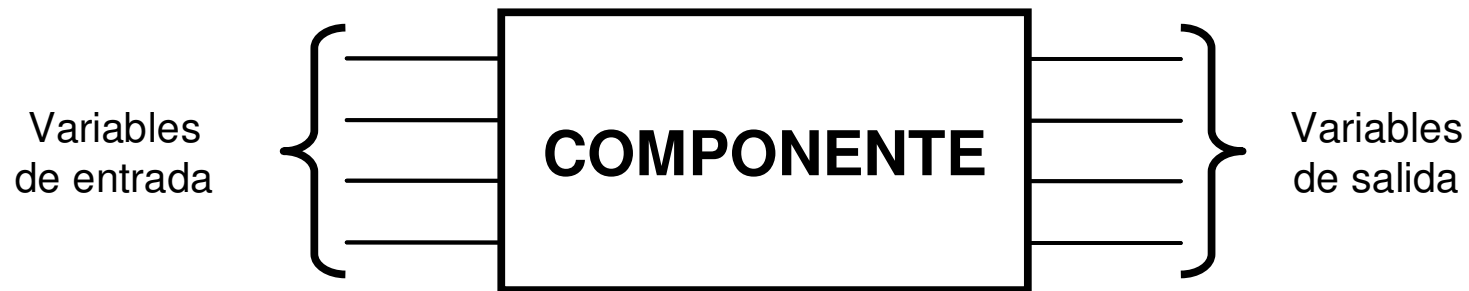
CARACTERÍSTICAS DE LOS LENGUAJES HDL ESTRUCTURADOS

- **Capacidad multinivel para poder definir un amplio rango de niveles de jerarquía.**
- **Capacidad para combinar la descripción estructural y las de comportamiento.**
- **Un conjunto de instrucciones que definen una sintaxis independiente del nivel de descripción.**
- **Independencia tecnológica que permite diseñar circuitos cuya descripción no depende de la forma de realización física posterior.**
- **Universalidad, es decir, compatibilidad con un gran número de herramientas de diseño microelectrónico asistido por computador (*Microelectronic CAD*), como por ejemplo simuladores (*Simulators*), generadores de vectores de prueba, etc.**
- **Facilidad de comprensión y lectura que simplifica la documentación.**

FUNDAMENTOS

CONCEPTOS BÁSICOS DE LOS LENGUAJES HDL ESTRUCTURADOS

Parten del concepto de "Componente" (*Component*), que es un circuito digital especificado mediante su descripción funcional, de comportamiento o estructural. Cada componente se identifica mediante un nombre y tiene asociado un conjunto de terminales de entrada/salida (señales de entrada/salida) que permiten establecer conexiones externas.

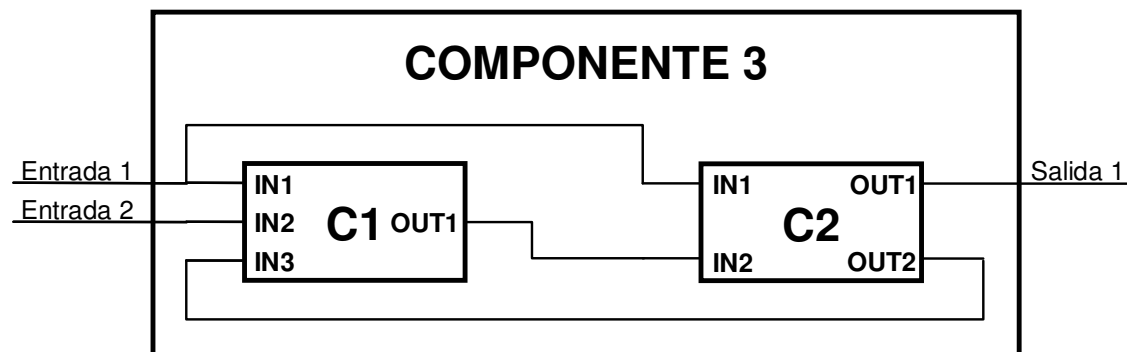


FUNDAMENTOS

CONCEPTOS BÁSICOS DE LOS LENGUAJES HDL ESTRUCTURADOS

Para llevar a cabo un diseño jerárquico, un componente de nivel superior hace referencia a los de nivel inferior a través de la definición de sus nombres y de sus conexiones en un proceso que se puede denominar repetición o reutilización (*Instantiation process*).

Por ejemplo, en la figura se representa un componente (Componente 3) que tiene como entradas las señales "Entrada 1" y "Entrada 2" y como salida la señal "Salida 1". Dicho componente está formado por otros dos componentes (C1 y C2) que pertenecen a una jerarquía inferior.



FUNDAMENTOS

CONCEPTOS BÁSICOS DE LOS LENGUAJES HDL ESTRUCTURADOS

Definición de los componentes

Incluye la definición del nombre y de las conexiones externas. Para definir el componente cada lenguaje utiliza denominaciones diferentes. Por ejemplo, en VHDL el componente se llama entidad (*Entity*), y en Verilog módulo (*Module*)

La forma de definir las conexiones externas y el tipo de conexión también difiere de un lenguaje a otro.

	<pre>ENTITY componente3 IS PORT(entrada1, entrada2: IN bit; salida1: OUT bit); END componente3;</pre>	
--	---	--

Definición de un componente en VHDL

FUNDAMENTOS

CONCEPTOS BÁSICOS DE LOS LENGUAJES HDL ESTRUCTURADOS

Descripción lógica de los componentes

Se puede realizar en el nivel de flujo de datos o en el algorítmico (comportamiento), en alguna de las formas permitidas por el lenguaje (ecuaciones lógicas, tablas de verdad, diagrama de estados en formato de texto, etc.), o en el nivel estructural, mediante la reutilización (*Instantiation*) e interconexión de componentes previamente descritos.