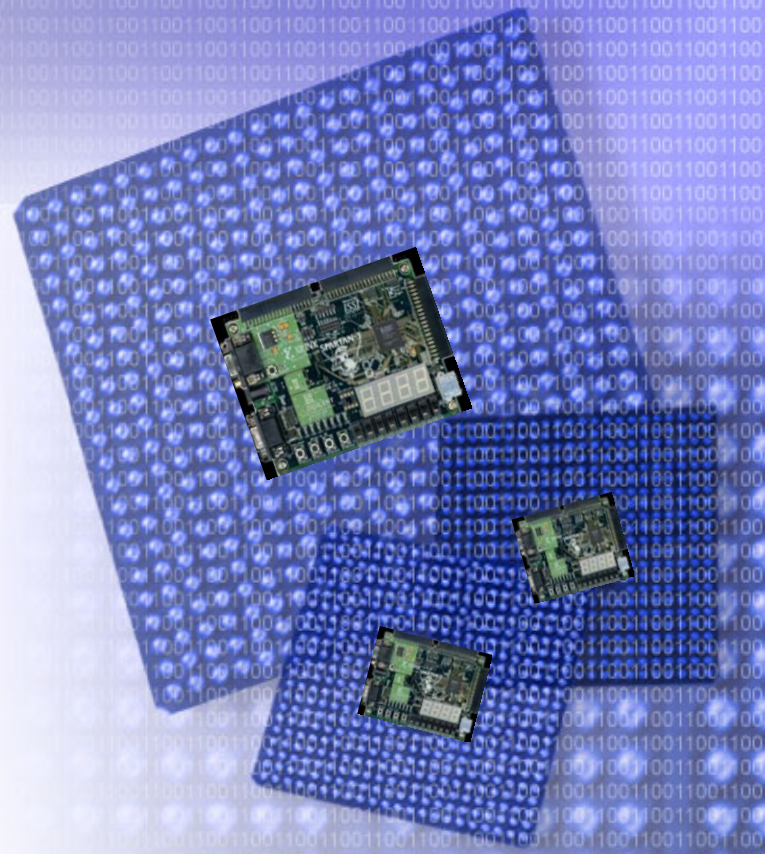
Universitat d'Alacant
Universidad de Alicante



VHDL. Lenguaje de descripción hardware

Introducción e historia

Dominios descriptivos y niveles de abstracción

Dominios descriptivos:

- **Comportamiento**
  - Se realiza la función sin información de cómo se hace
- **Estructural**
  - Los bloques se conectan mediante interconexiones (*netlist* o esquemas)
- **Físico**
  - Localización y propiedades físicas reales

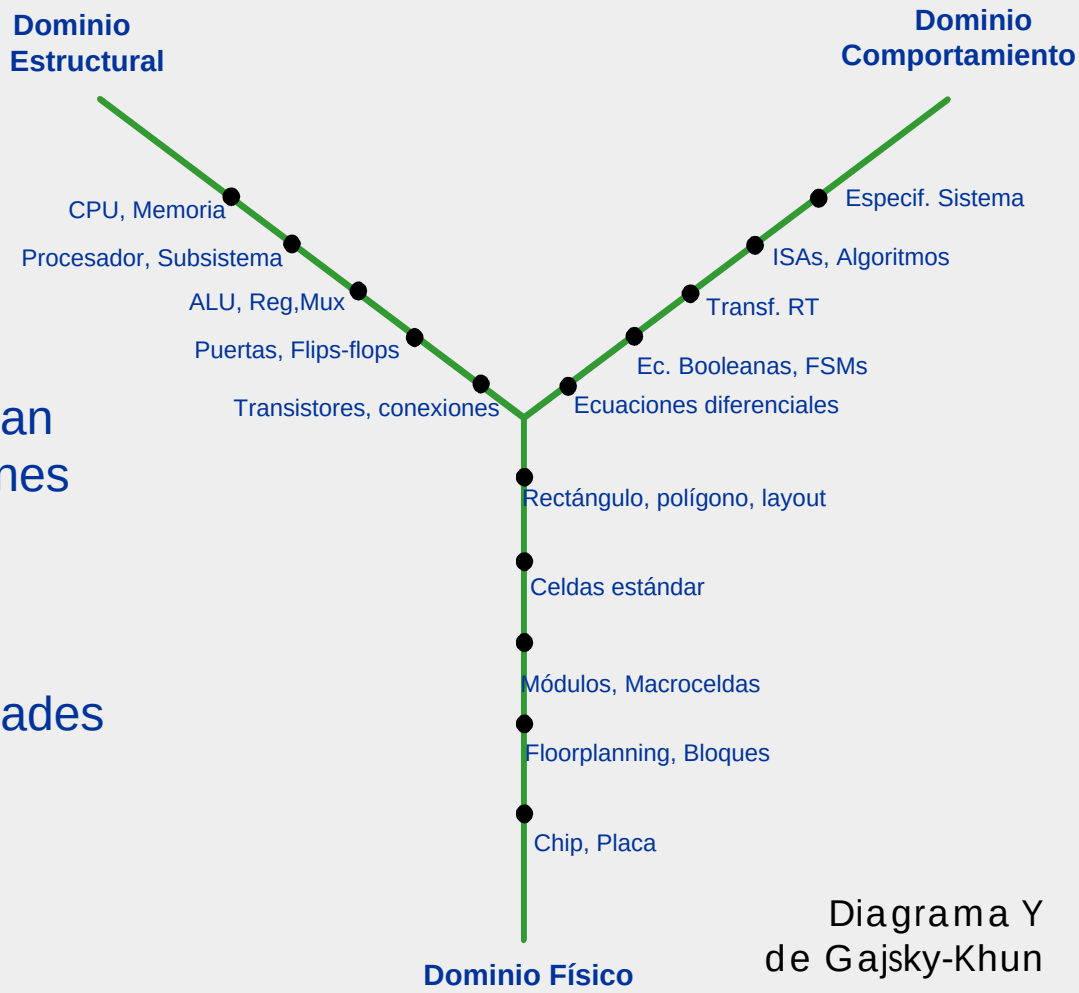


Diagrama Y de Gajsky-Khun

Dominios descriptivos y niveles de abstracción

Niveles de abstracción:

- **Circuito**
  - Valores continuos, Todo es electrónica, tiempo continuo
- **Lógico**
  - Valores lógicos (T,F), sólo computación, tiempo continuo
- **RT** (*Register Transfer*)
  - Palabras con valores discretos, control y procesamiento, tiempo discreto
- **Algorítmico**
  - Estructuras abstractas, dependencias en lugar de tiempo
- **Sistema**
  - Relaciones entre subsistemas, sincro. y protocolos

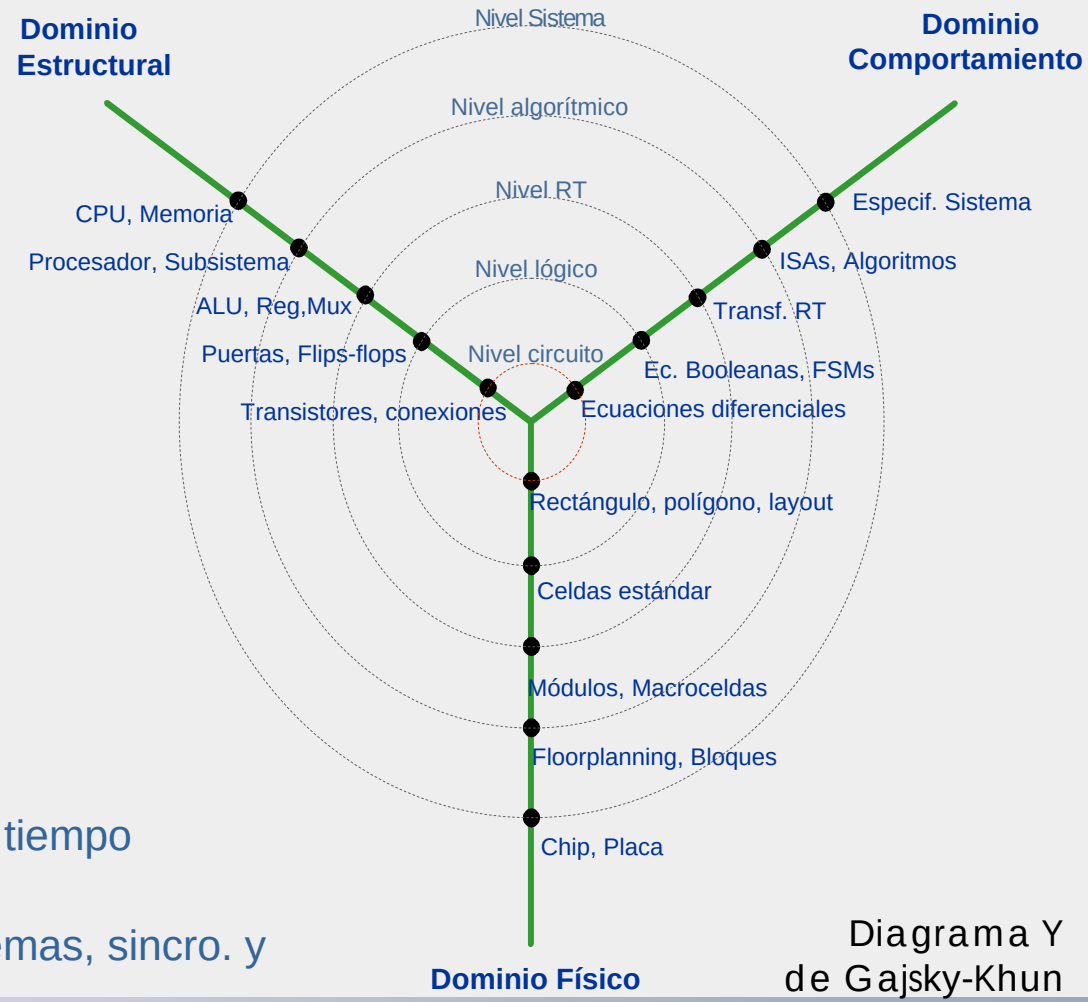


Diagrama Y de Gajsky-Khun

Dominios descriptivos y niveles de abstracción

- Medidas:
  - **Circuito**
    - Tiempo de subida y bajada, consumos área
  - **Lógico**
    - Tiempo de conmutación, *skew*, área equivalente
  - **RT**
    - Tiempo de ciclo, márgenes, puertas equivalentes
  - **Algorítmico**
    - Latencia, cadencia de datos, número de módulos
  - **Sistema**
    - Ancho de banda, MIPS.

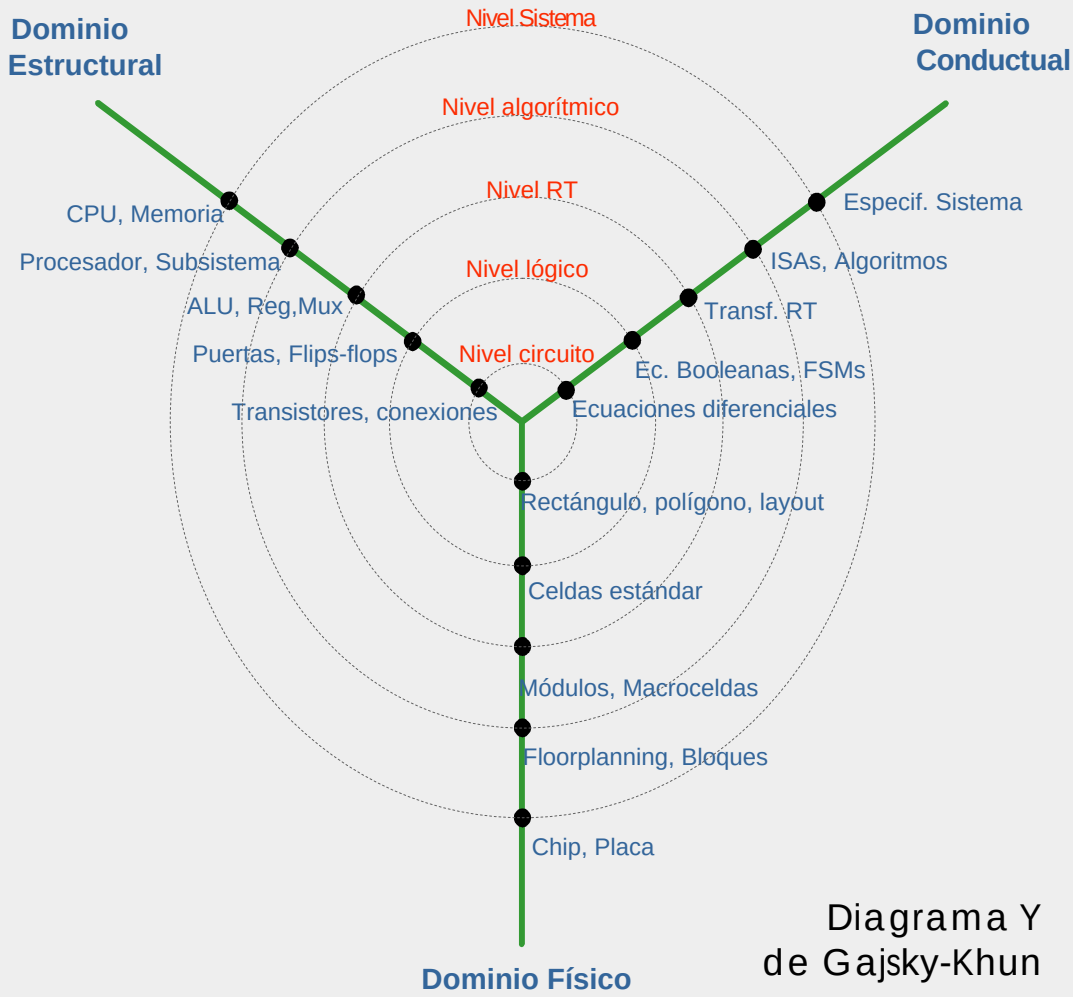


Diagrama Y de Gajsky-Khun

Dominios descriptivos y niveles de abstracción

- Transiciones:
  - **Síntesis/Análisis**
    comportamiento ⇔ estructural
  - **Optimización**
    Mejora de una descripción sin variar el nivel de abstracción
  - **Generación/ Extracción**
    estructural ⇔ físico
  - **Refinamiento/Abstracción**
    Bajar/Subir el nivel de abstracción en el mismo dominio

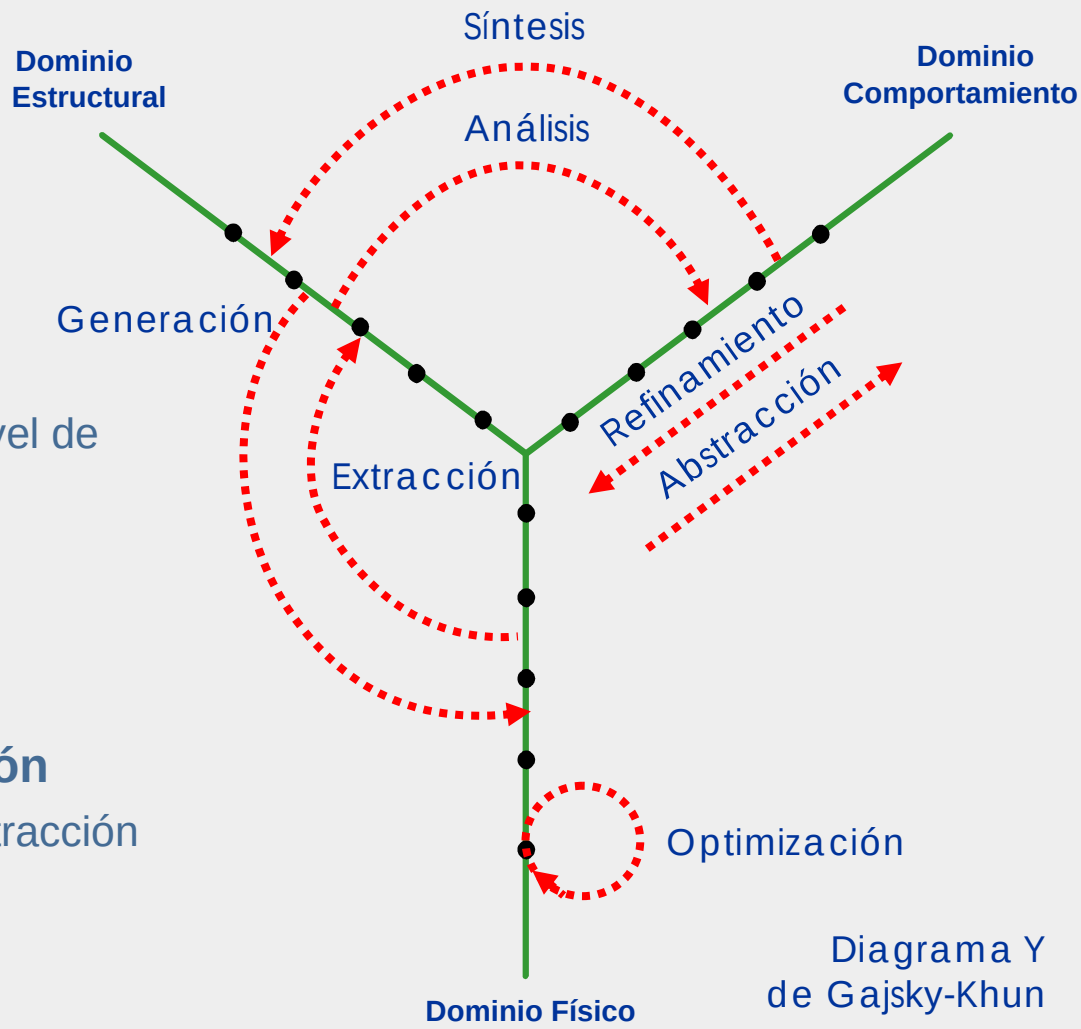
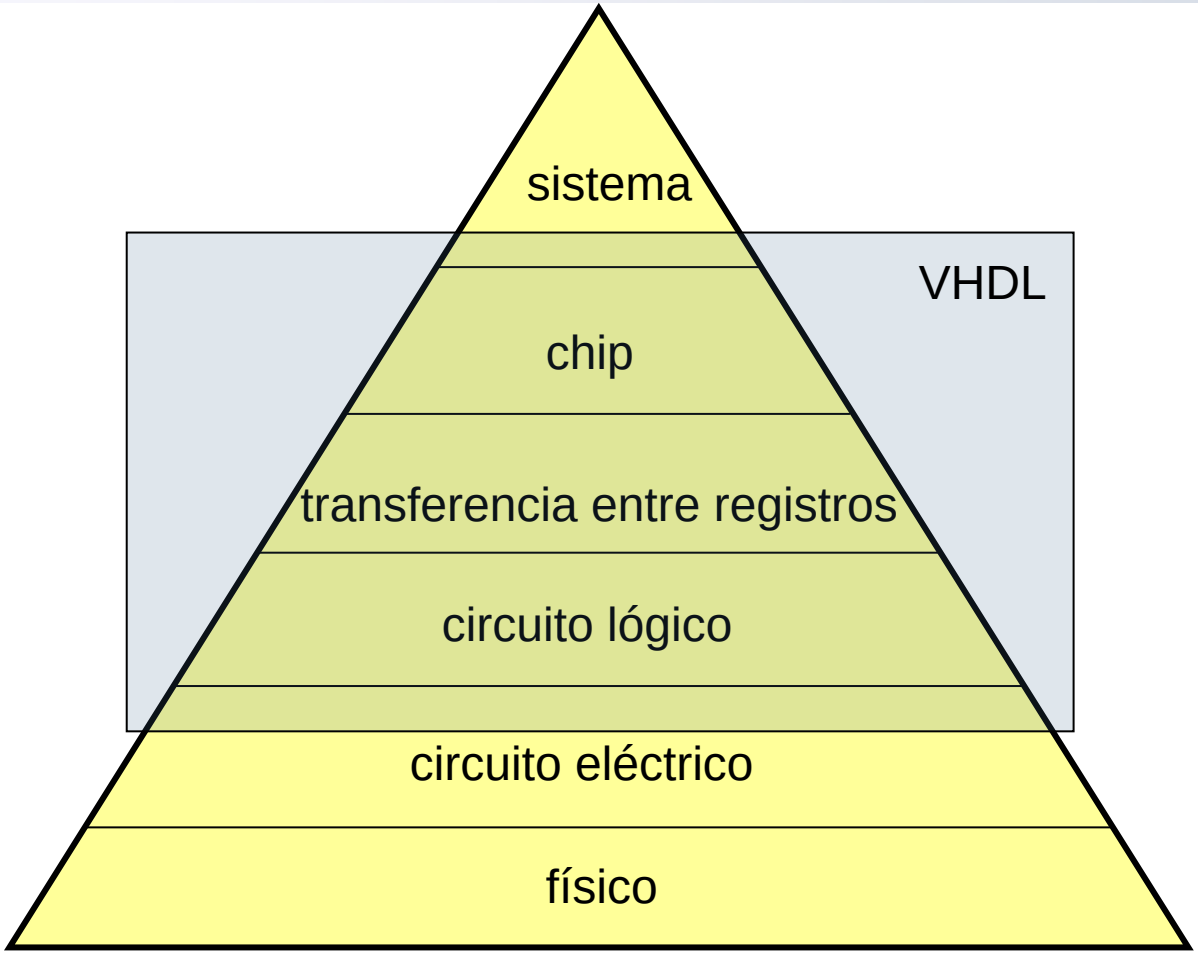


Diagrama Y de Gajsky-Khun

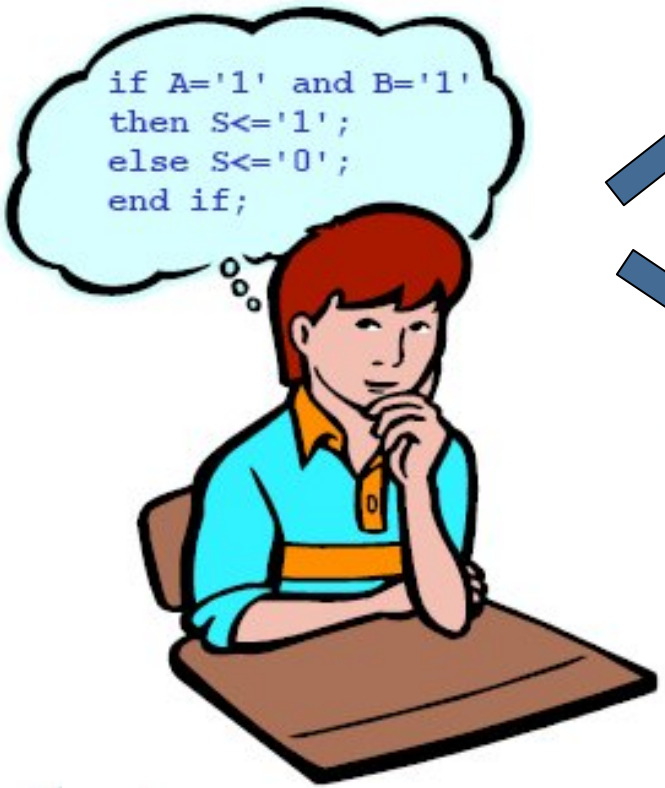
Niveles de diseño y dominios de representación



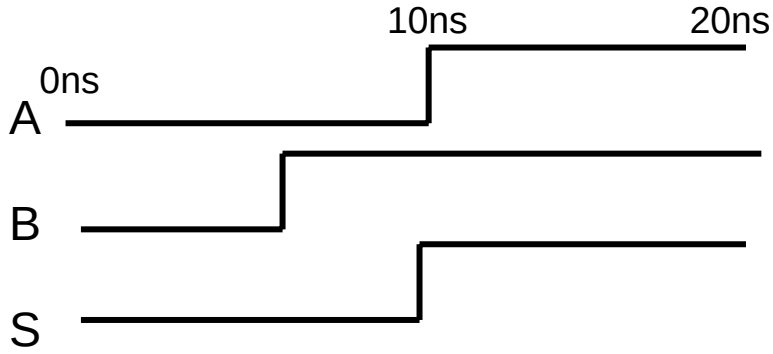
VHDL: VHSIC, Hardware Description Language
VHSIC: Very High Speed Integrated Circuits

¿Para qué sirven los HDL?

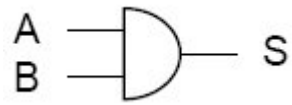
- Sirven para **modelar** circuitos, para **expresar** ideas

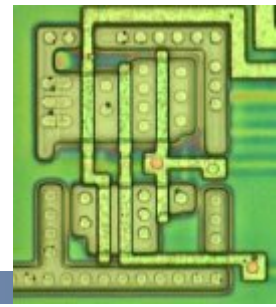


- Los modelos se pueden **simular** para comprobar que se corresponden con la **funcionalidad** deseada



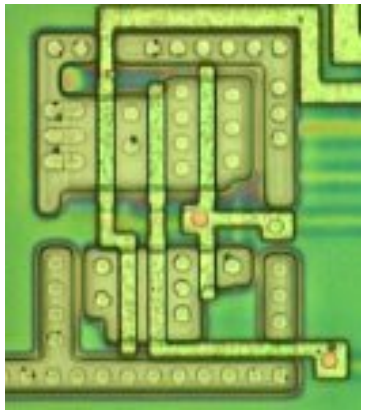
- O se pueden **sintetizar** para crear un circuito que funcione como el modelo





¿Para qué sirven los HDL?

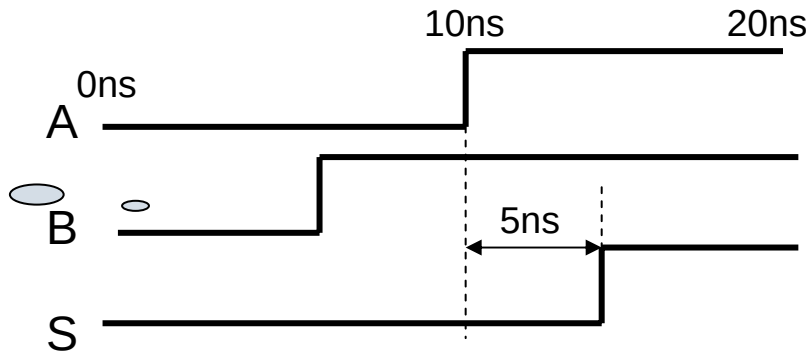
- Se puede construir el modelo de un circuito que ya exista, que ya esté implementado, es decir, **sintetizado**



```
if A =‘1’ and B=‘1’ then
   S <= ‘1’ after 5 ns;
else
   s <= ‘0’ after 5 ns;
end if;
```

- En este caso el objetivo es simular el circuito para comprobar que su funcionalidad se ajusta a las especificaciones iniciales

simulación post-layout



¿Para qué sirven los HDL?

- Para generar documentación, siempre y cuando los modelos de los circuitos estén bien comentados.

```
entity bit8tobcd is
    Port ( clk    : in std_logic;  -- reloj del sistema, para dividir
           rst    : in std_logic;
           cuenta : out std_logic_vector(1 downto 0);-- para depurar
           bitin  : in std_logic_vector(7 downto 0); -- 8 bits de entrada
           seg    : out std_logic_vector(6 downto 0);-- tiempo a iluminar
           ctrlD  : out std_logic_vector(3 downto 0; -- ctrl de los ánados
           bcd    : out std_logic_vector(15 downto 0)-- para depurar
         );
end bit8tobcd;
```

- Para crear bancos de prueba (*test-bench*), es decir, crear los estímulos y ver los resultados durante la simulación.

Estado actual y alternativa

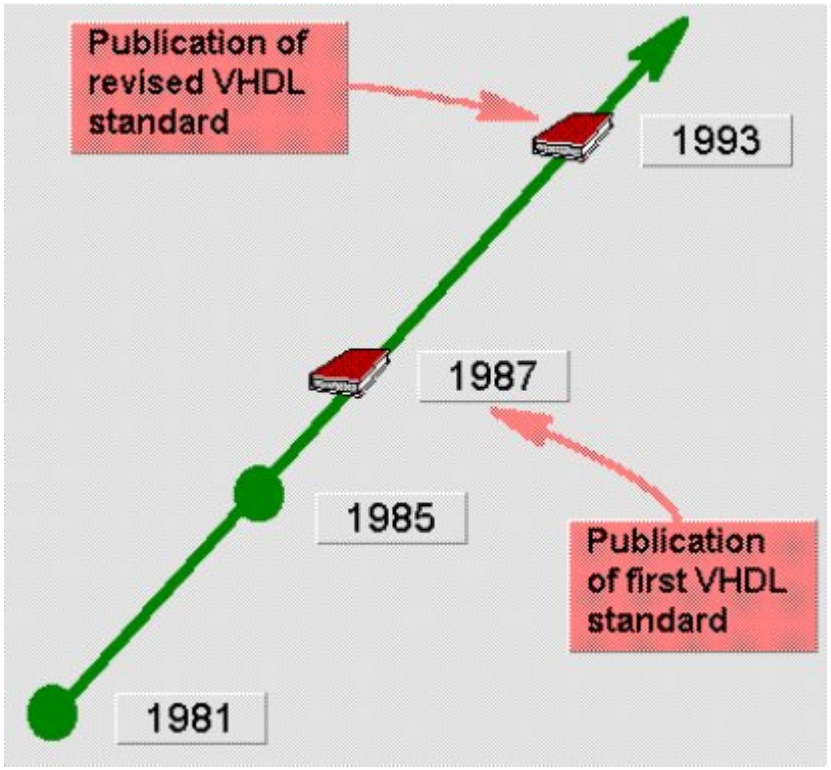
- En la actualidad el diseño mediante esquemas no es una alternativa realista en ningún proyecto, por ejemplo GForce4 tiene 65 Mtransistores y 800.000 líneas de código Verilog.
- La alternativa estándar es usar un HDL
  - Verilog: Costa Oeste de EEUU, *para ASICs, menos verboso, más parecido a C, menos expresivo.*
  - VHDL: Costa Este y Europa, *para FPGAs, más verboso, más parecido a PASCAL y ADA, más expresivo.*
- El diseño se sintetiza a partir de un HDL, pero gran parte del diseño y la verificación se realiza con lenguajes estándares
  - C y Matlab
- VHDL es el estándar para FPGAs en proyectos industriales de moderada complejidad en España.

VHDL. Historia

- Surge a principios de los ’80s del proyecto DARPA (del Departamento de Defensa de los EE.UU.) llamado VHSIC (Very High Speed Integrated Circuits)
- Se utiliza como forma de describir circuitos integrados
  - Crisis del ciclo de vida del HW: *Cada vez los circuitos integrados eran más complejos, y el coste de reponerlos cada vez era mayor, básicamente porque no estaban correctamente documentados. VHDL nació como una manera estándar de documentar circuitos.*
  - El uso de VHDL permitió comprobar que el tiempo de diseño de los circuitos se reducía, porque se podían crear directamente de su descripción: *utilidad de la síntesis.*
  - En 1987 el trabajo se cedió al IEEE, y a partir de ese momento es un estándar abierto.
- VHDL: VHSIC Hardware Description Language

  VHSIC: Very High Speed Integrated Circuits

VHDL. Evolución



1980: El departamento de defensa de los EEUU funda el proyecto para crear un HDL estándar dentro del programa VHSIC

1981: Woods Hole Workshop, reunión inicial entre el Gobierno, Universidades e Industria

1983: Se concedió a Intermetrics, IBM y Texas Instruments el contrato para desarrollar VHDL

1985: Versión 7.2 de dominio público.

1987: El IEEE lo ratifica como su estándar 1076 (VHDL-87)

1993: El lenguaje VHDL fue revisado y ampliado, pasando a ser estándar 1076 ‘93 (VHDL-93)

2000: Última modificación de VHDL

VHDL. ¿Futuro?

- Los lenguajes de descripción de hardware también tienen limitaciones:
  - Metodología de diseño nueva, exige un cambio de mentalidad con respecto al SW
  - No permiten reusar código SW para HW.
  - Poseen enorme reusabilidad intrínsica.
  - La decisión HW/SW se debe hacer antes de la codificación
  - La simulación es lenta, siempre hay que recurrir a una simulación algorítmica usando lenguajes SW
- Como respuesta a esto, hay varias iniciativas para describir HW usando lenguajes de alto nivel, tipo SW
  - Handel-C, System-C
  - Forge (Java)
  - Superlog
- **VHDL es un lenguaje de presente, en el futuro ya se verá ...**