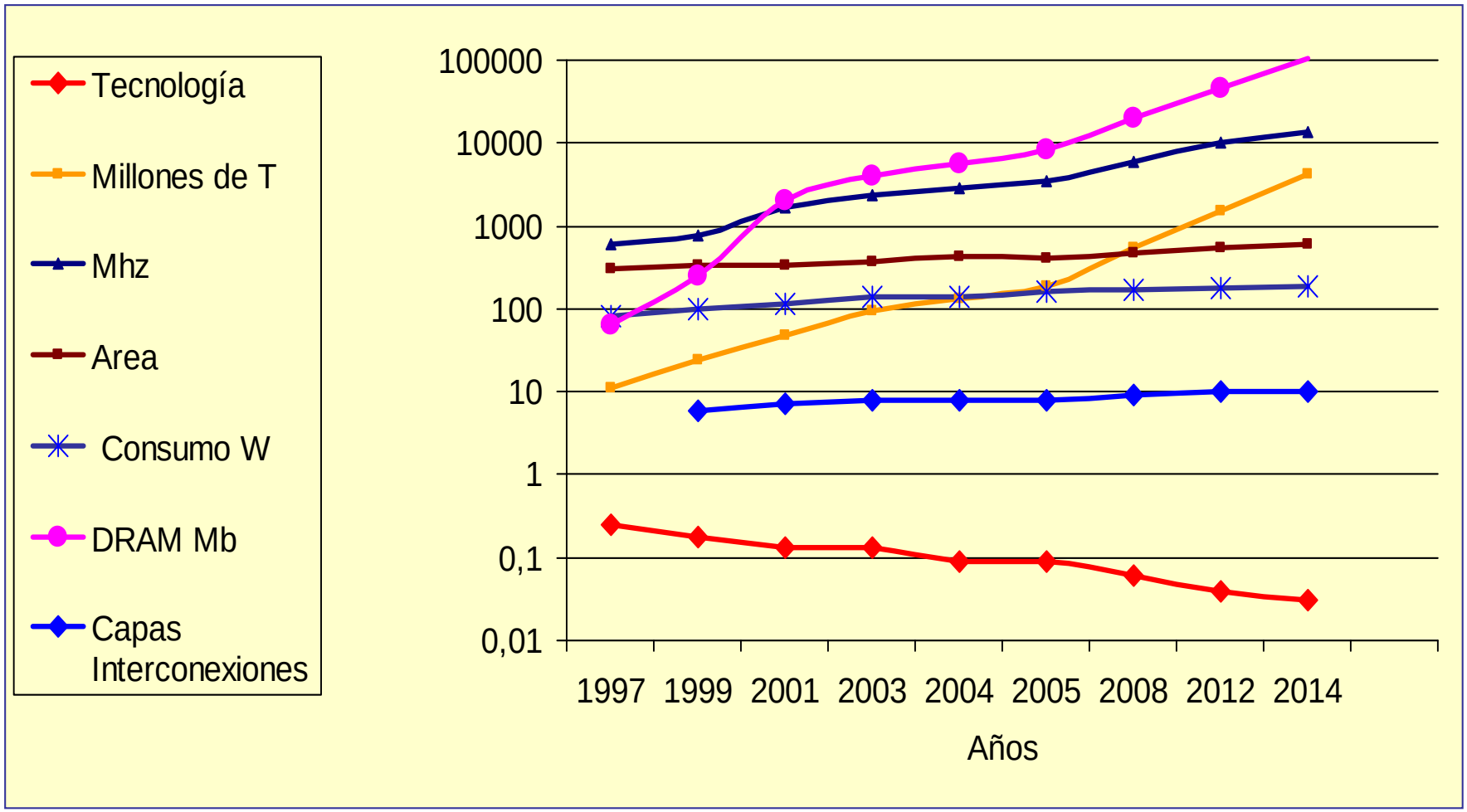
Síntesis arquitectónica y de alto nivel

Módulo 1. Concepto y fases de la Síntesis de Alto Nivel

Diseño de circuitos: la complejidad

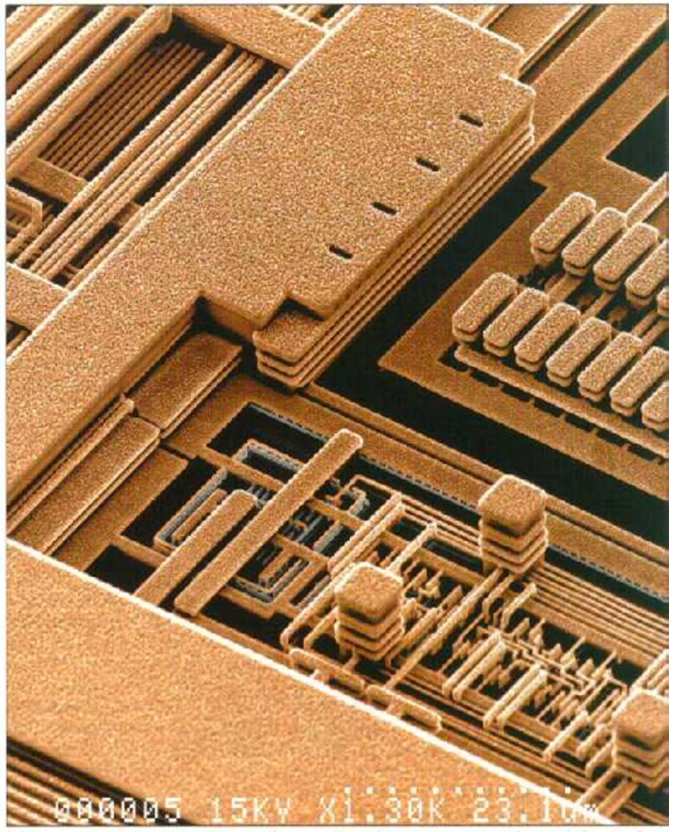
- Tratamiento de problemas de complejidad creciente
  - Rápido desarrollo Tecnología Electrónica
  - Avances en Arquitectura de Computadores
- Dificultades
  - Enorme complejidad de los chips
  - Acortamiento de la vida activa del chip
- Consecuencias
  - Acortamiento del ciclo de diseño
  - Necesidad de herramientas CAD
    - Place&routing, esquemas, lenguajes
  - Aumento del nivel de abstracción
    - Layout, transistor, puerta, esquema RT, algoritmos, …

Evolución tecnológica



Evolución tecnológica

- Nuevos procesos tecnológicos
  - Conexiones de **cobre** (IBM, Motorola, Xilinx)
    - Menor resistividad
    - Menor anchura
    - Menor capacidad
    - Mayor velocidad
  - **Once capas de metal** para interconexiones
  - Tamaño tecnología: 45 nm.
  - Tensión de alimentación: ~1.5 V (menos consumo)
  - 1000 millones de transistores en un chip.



Marco conceptual del diseño

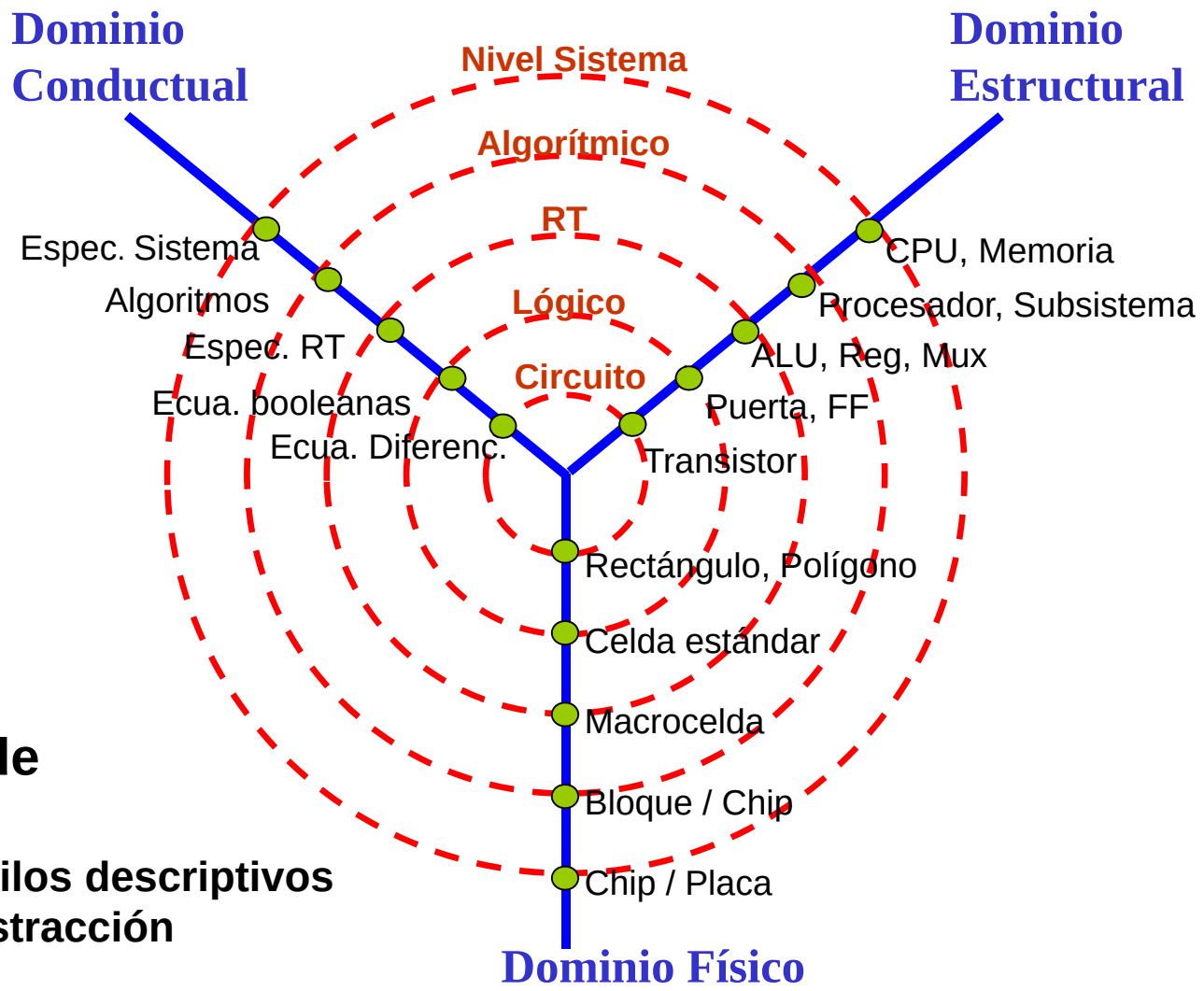


Diagrama en Y de Gajski-Kuhn

- **Dominios: Estilos descriptivos**
- **Niveles de abstracción**

Lenguajes de descripción de HW (LDH)

- Notación utilizada para describir un sistema digital con un cierto rango de niveles de abstracción en al menos uno de los dominios de descripción
- Evolución
  - Abundancia de lenguajes
  - Tendencia a la estandarización: Verilog, VHDL
- Aplicaciones
  - Diseño automático: descripción inicial del sistema
  - Reducción de la complejidad notacional
    - LDH de alto nivel: Descripción precisa y concisa
    - Vehículo de comunicación fabricantes usuarios
  - Simulación y verificación.
    - Comprobación de las especificaciones
    - Detección precoz de errores
    - Predicción de rendimiento

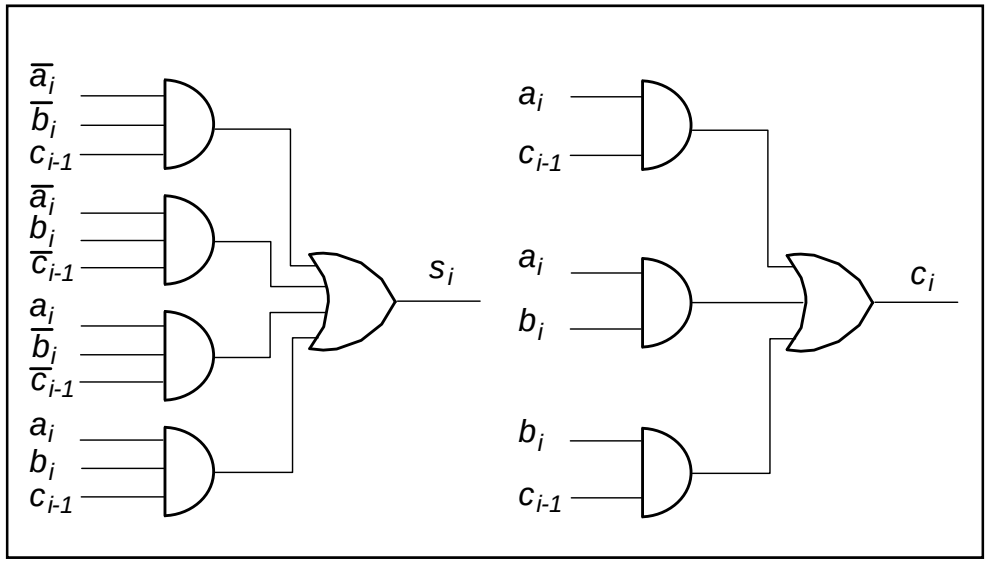
Dominios: estilos descriptivos

Conductual

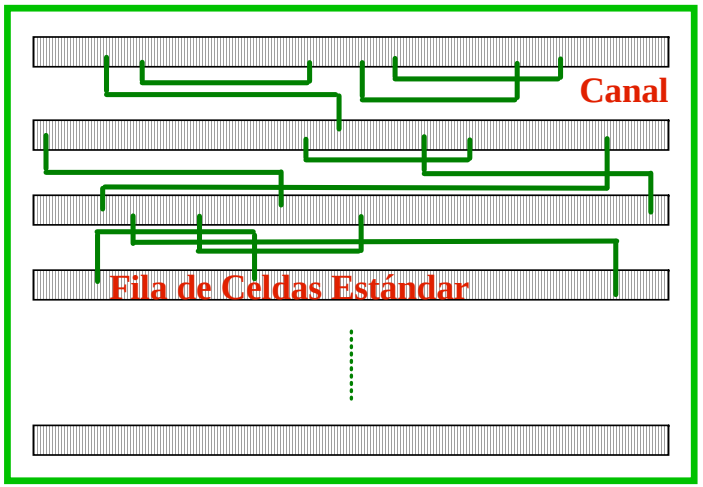
$$s_i(a_i,b_i,c_{i-1}) = \bar{a}_i\bar{b}_i c_{i-1} + \bar{a}_i b_i \bar{c}_{i-1} + a_i\bar{b}_i\bar{c}_{i-1} + a_i b_i c_{i-1}$$

$$c_i(a_i,b_i,c_{i-1}) = a_i c_{i-1} + a_i b_i + b_i c_{i-1}$$

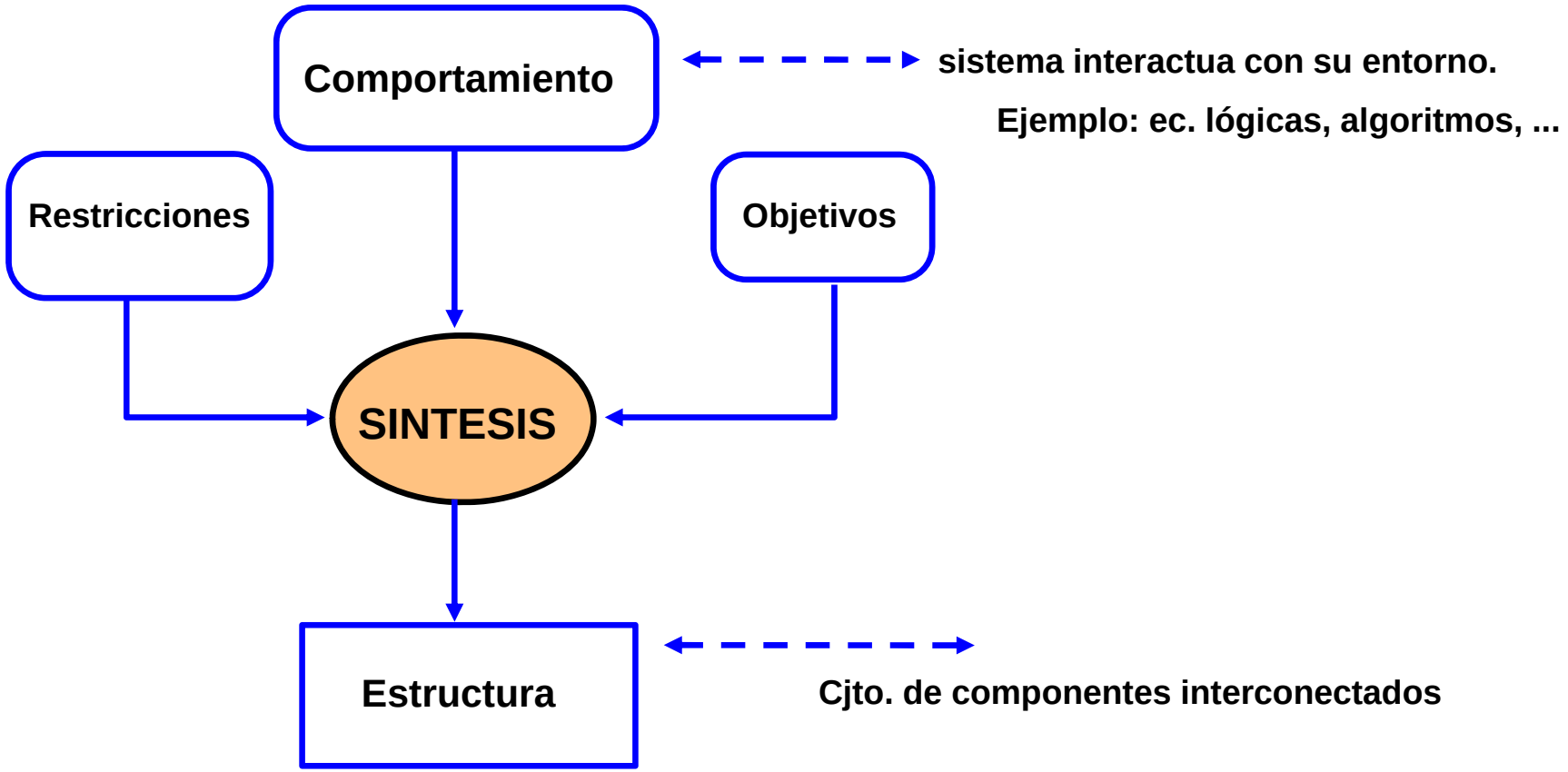
Estructural



Físico / Geométrico



Concepto de síntesis



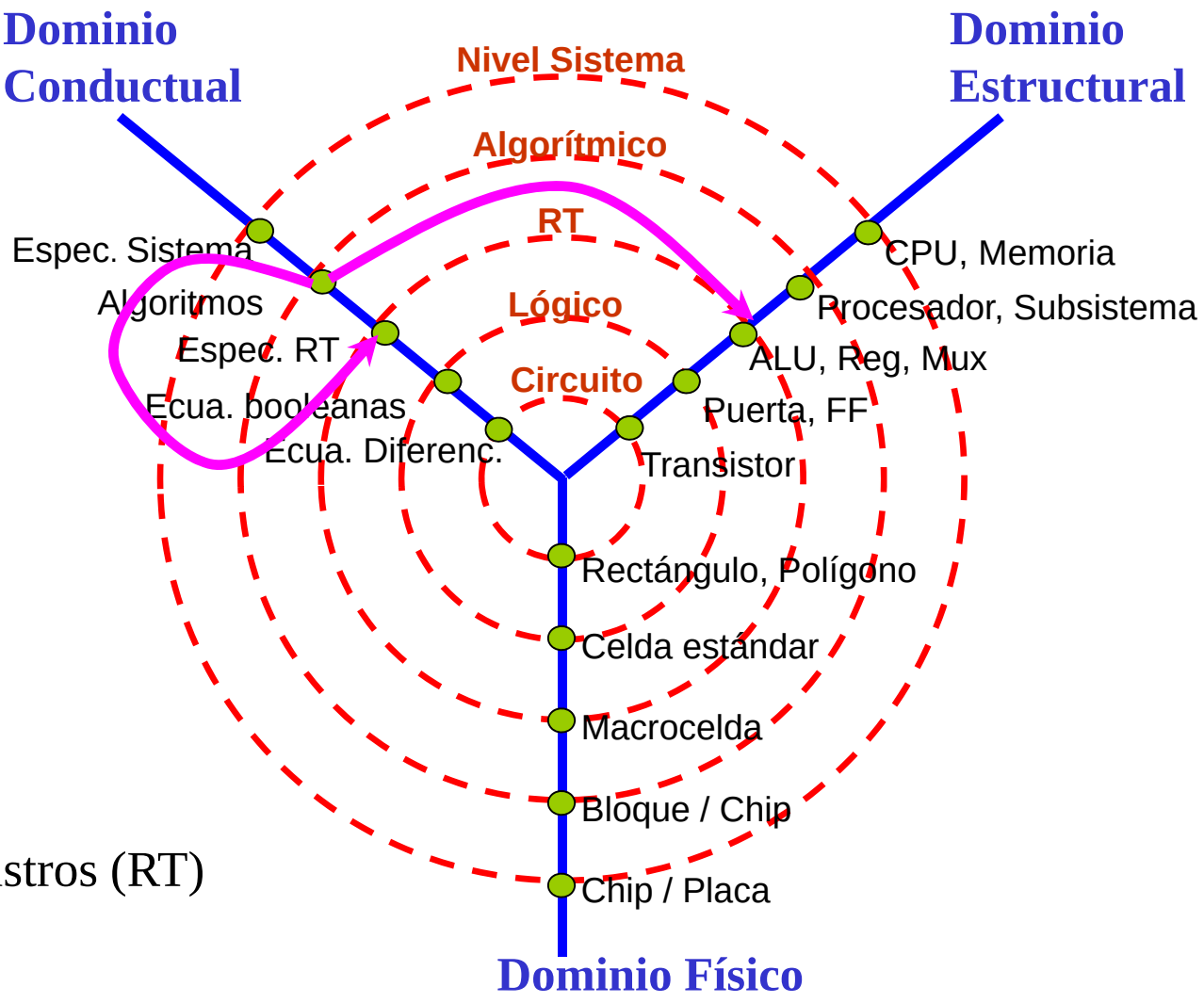
Ejemplos:

Algoritmos de síntesis lógica (espresso)

Síntesis RTL (Synopsis, Actel, ...)

Síntesis de Alto Nivel (SAN)

- Entrada:
  - Conductual
  - Algorítmico
- Salida:
  - Estructural
  - Transf. Registros (RT)



Síntesis de Alto Nivel

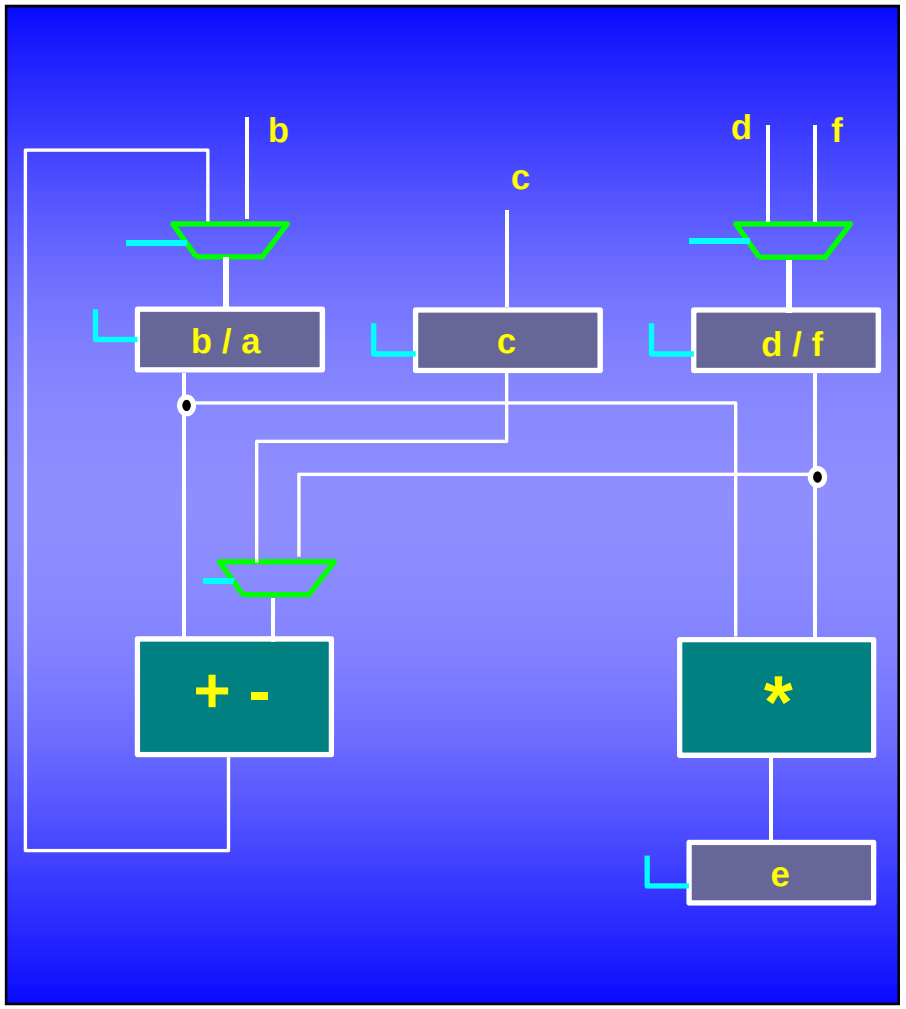
Proceso de síntesis donde

- Descripción comportamiento
  - A nivel algorítmico
  - Tipos de datos habituales: enteros, arrays, etc.
  - Escasa mención a estructura

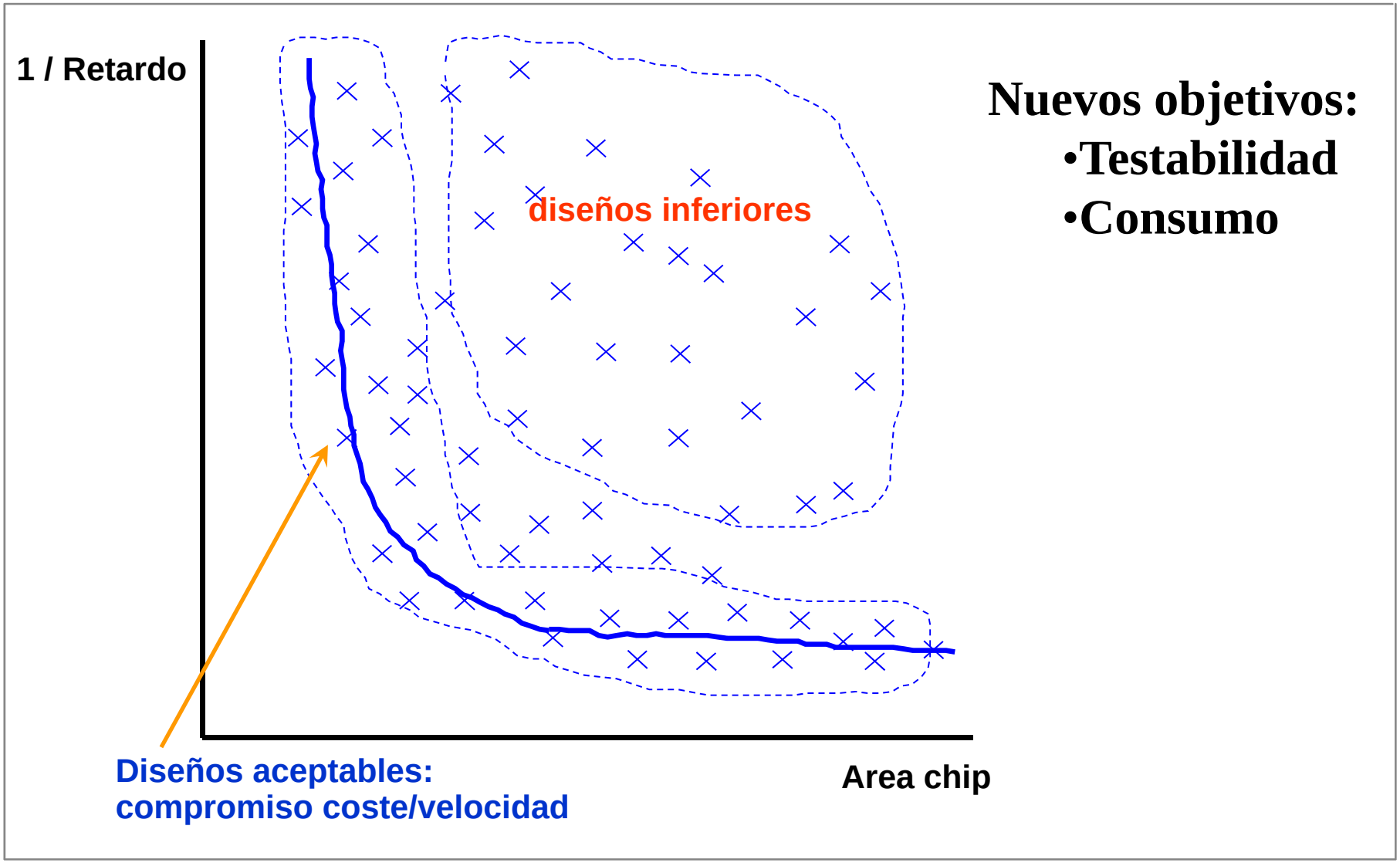
    a = b + c - d

    e = a * f

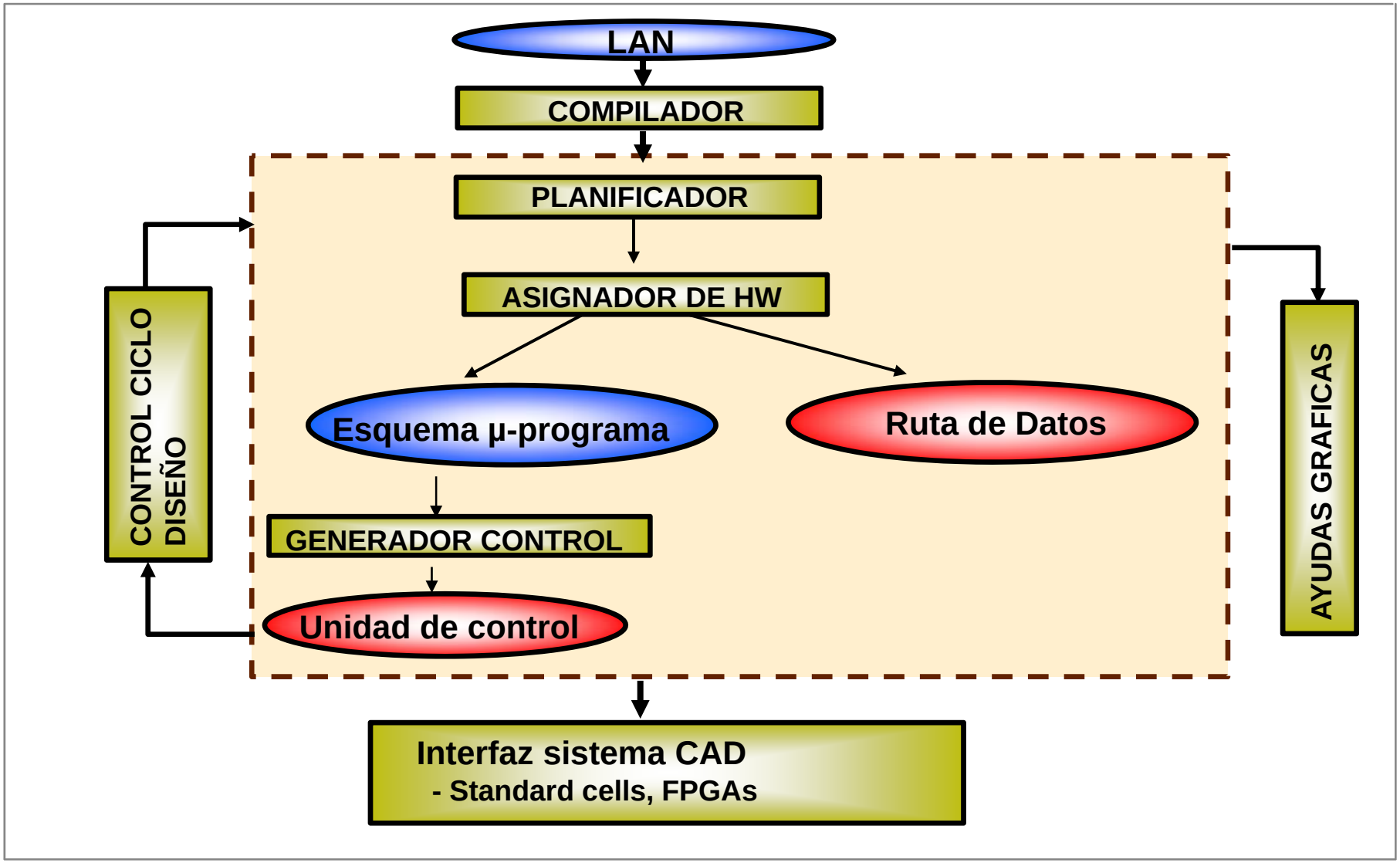
- Estructura generada
  - A nivel RT
  - Ruta de datos: Registros, buses, MUX, UFs, …
  - Control: autómata finito. Activa los puntos de control de la RD



Espacio de diseño de la SAN



Subtareas de la SAN

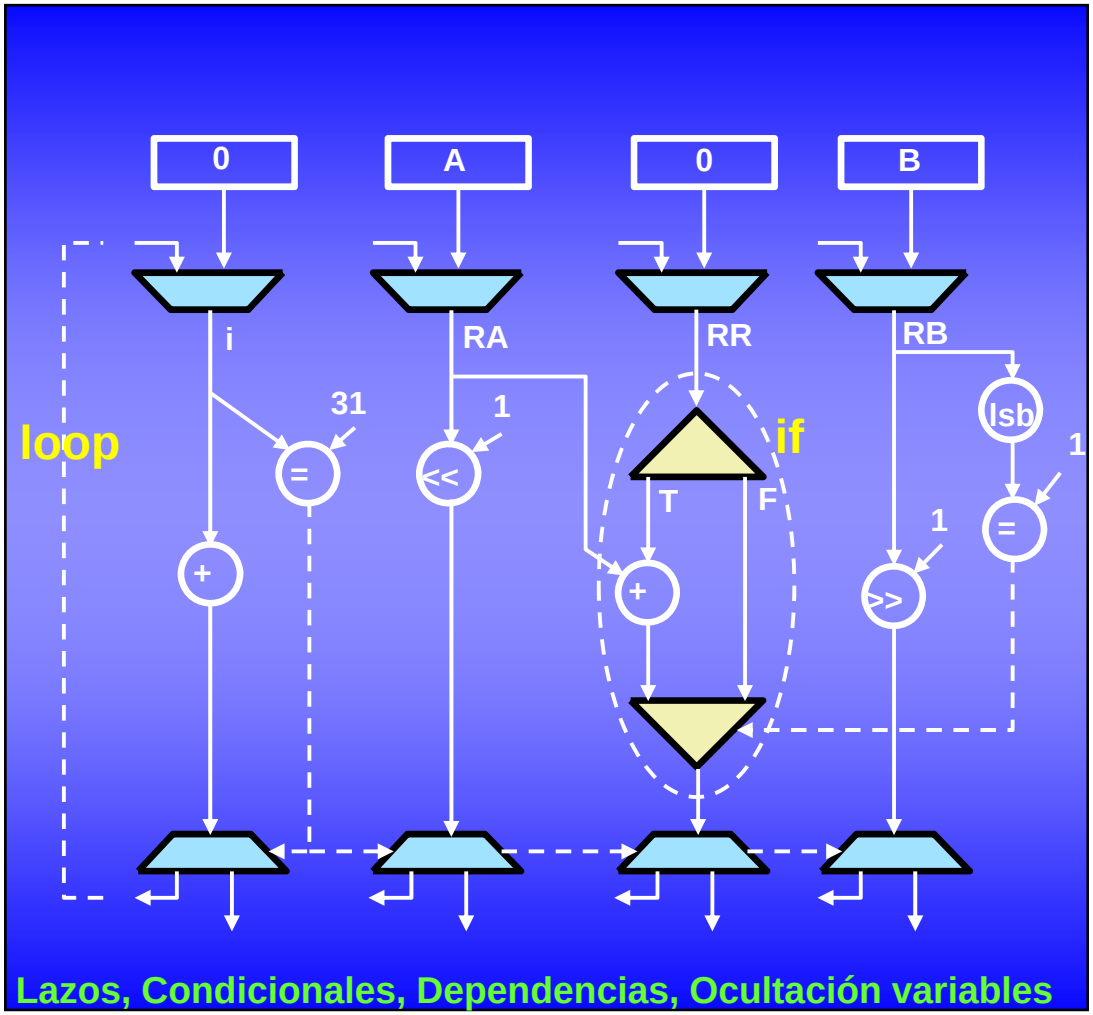


Compilación / Representación interna

Ejemplo

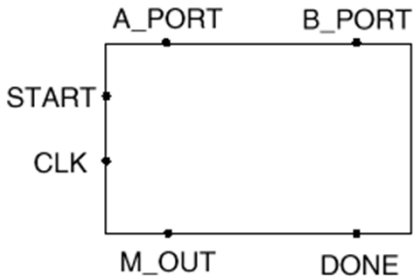
Algoritmo:

```
RA := A   /* multiplicando */
RB := B  /* multiplicador */
RR := 0  /* resultado */
for i:=0 to 31 do
     if RB(0)=1 then RR=RR+RA end;
          RA << RA,1
          RB >> RB,1
endfor;
```



Flujo de diseño en SAN y representación de datos

- Ejemplo
  - Descripción VHDL de un multiplicador para números de 4 bits

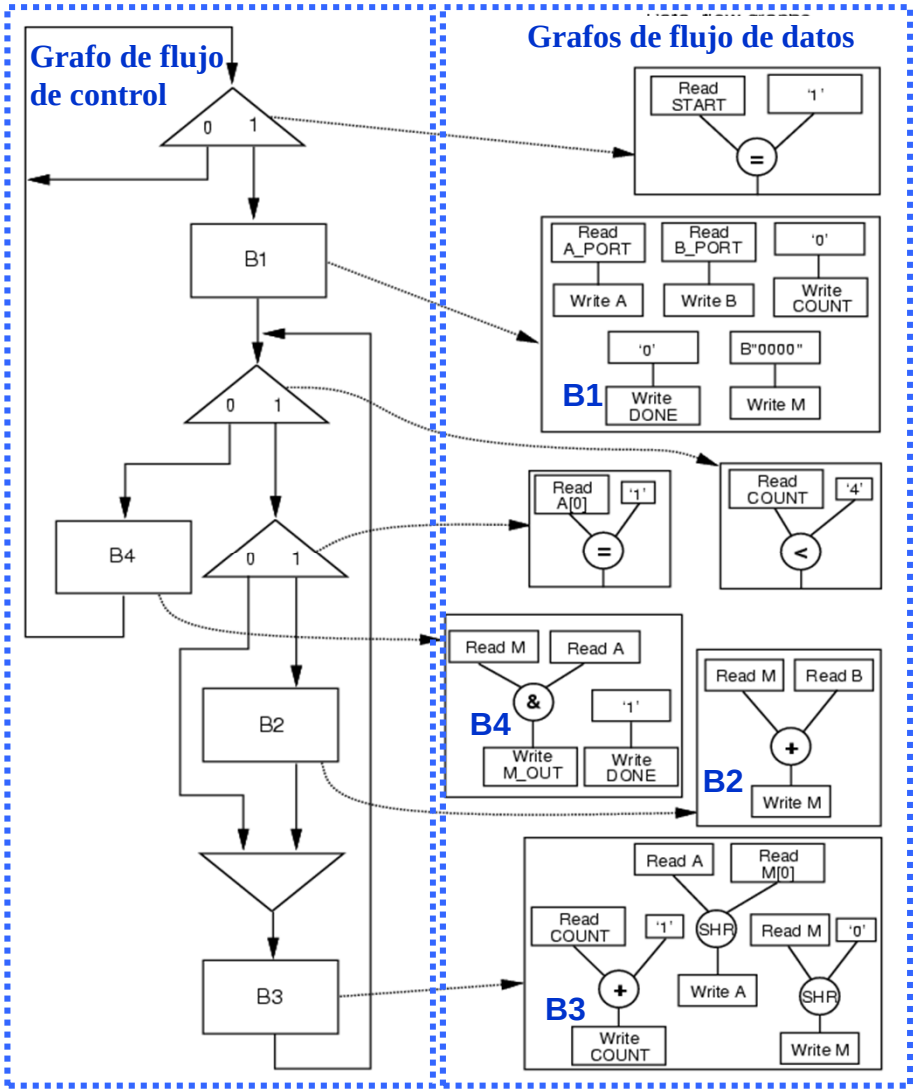



```
entity MULT is
 port ( A_PORT,
          B_PORT:  in bit_vector(3 downto 0);
          M_OUT:  out bit_vector(7 downto 0);
          CLK:  in CLOCK;
          START:  in BIT;
          DONE:  out BIT;
        );
end MULT;
```

```
architecture SHIFT_MULT  of  MULT  is
begin
      process
          variable A, B, M  :  BIT_VECTOR;
          variable COUNT  :  INTEGER;
      begin
          wait until (START = 1);
          A := A_PORT;   COUNT := 0;
          B := B_PORT;   DONE <= '0';          B1
          M := B"0000";
          while  (COUNT < 4)  loop
              if (A(0) = '1')  then
                    M  :=  M + B;              B2
              end if;
              A := SHR(A, M(0));
              M := SHR(M, '0');                B3
              COUNT := COUNT + 1;
          end loop;
          M_OUT  <=  M & A;
          DONE <= '1';                         B4
      end process;
end SHIFT_MULT;
```

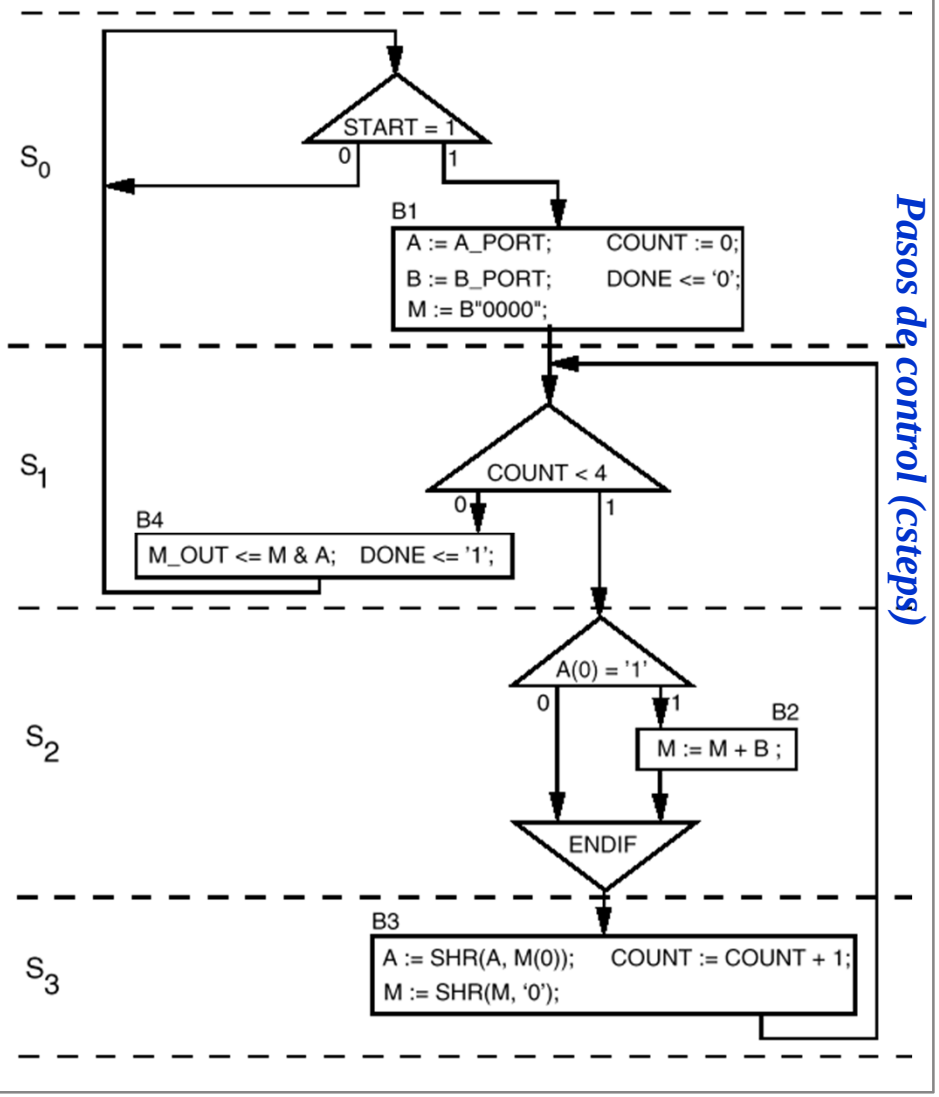
Flujo de diseño en SAN y representación de datos

- Ejemplo (cont.)
  - Paso 1. Compilación
    - Obtención de una representación que muestre el flujo de datos y de control
    - Sin alusión a la estructura hardware
    - Extracción del paralelismo implícito



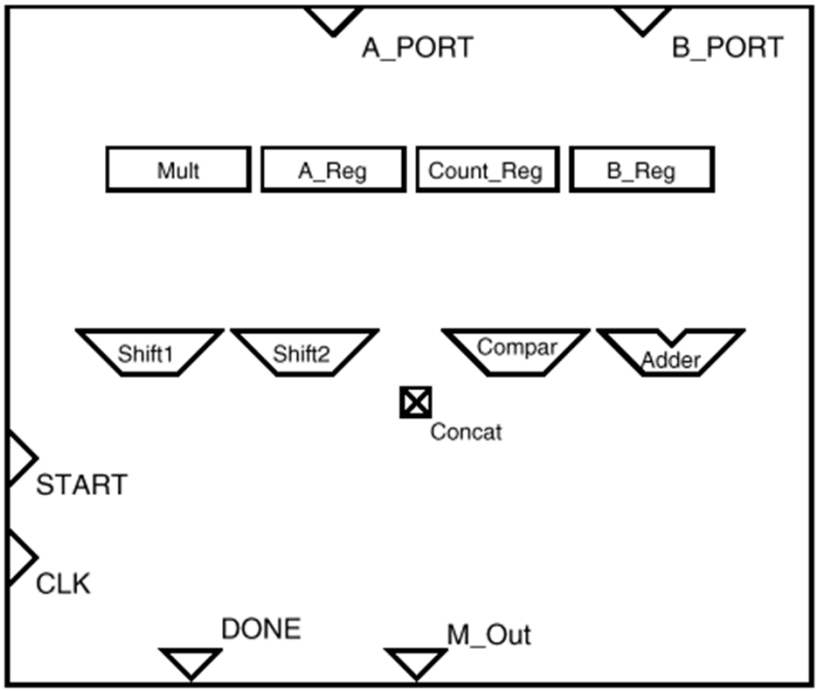
Flujo de diseño en SAN y representación de datos

- Ejemplo (cont.)
  - Paso 2. Planificación temporal (scheduling)
    - Determinar en qué estado o paso de control se ejecuta cada una de las operaciones de la descripción
    - Muchas alternativas: repercusión sobre coste y tiempo
    - Retroanotación (“back annotation”)



Flujo de diseño en SAN y representación de datos

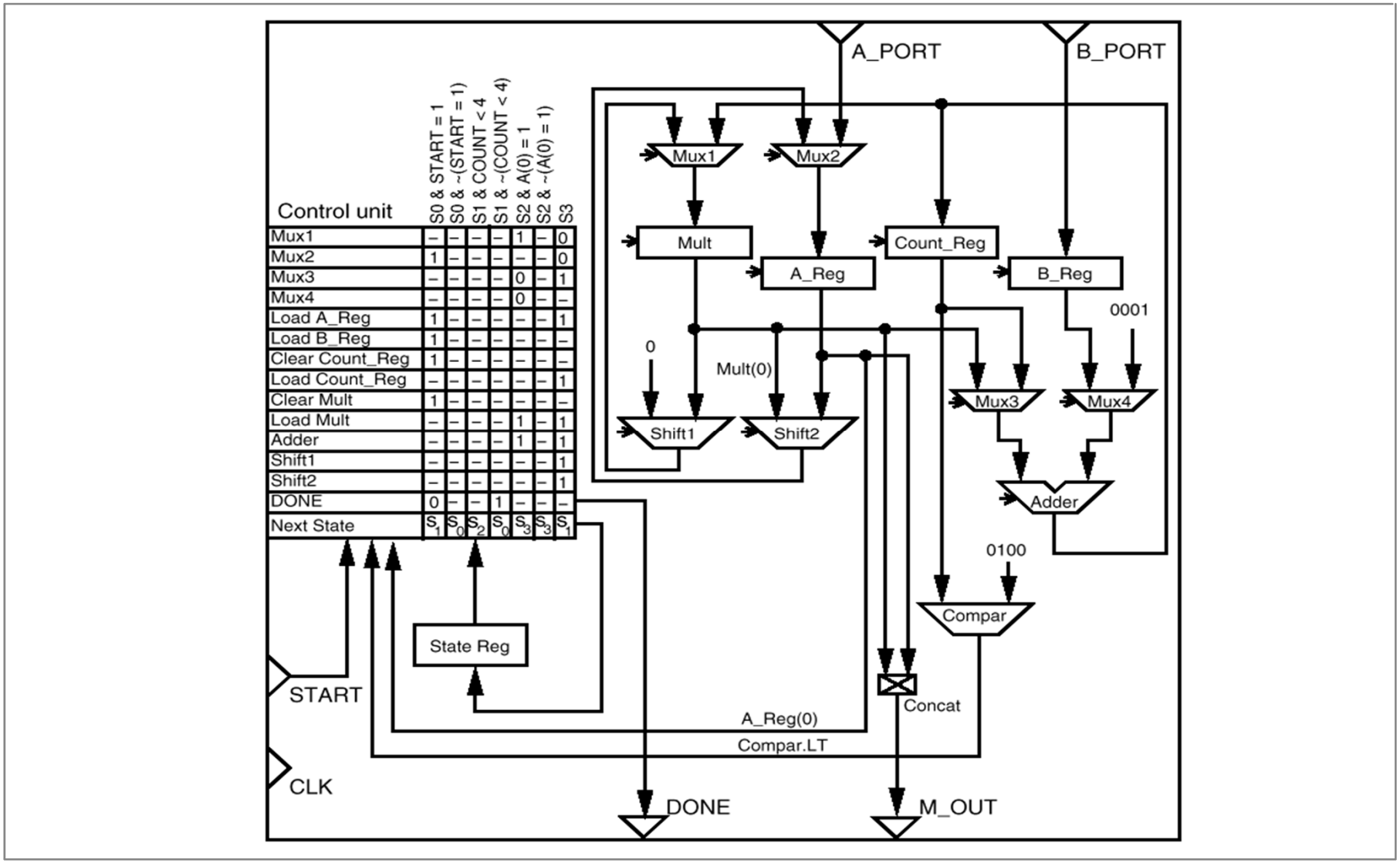
- Ejemplo (cont.)
  - Paso 3. Selección de módulos
    - Determinar tipos y número de módulos hardware de nivel RT que se usarán en el diseño
    - Selección condicionada por el scheduling



Flujo de diseño en SAN y representación de datos

- Ejemplo (cont.)
  - Paso 4. Asignación de módulos e interconexiones
    - Determinar con qué módulo se implementa cada operación
    - Determinar en qué registro se almacena cada dato
    - Proveer las interconexiones necesarias (modelos)
    - Generación del control simbólica
    - Retroanotación
    - Pasos 3+4: Asignación de hardware (“Allocation”)

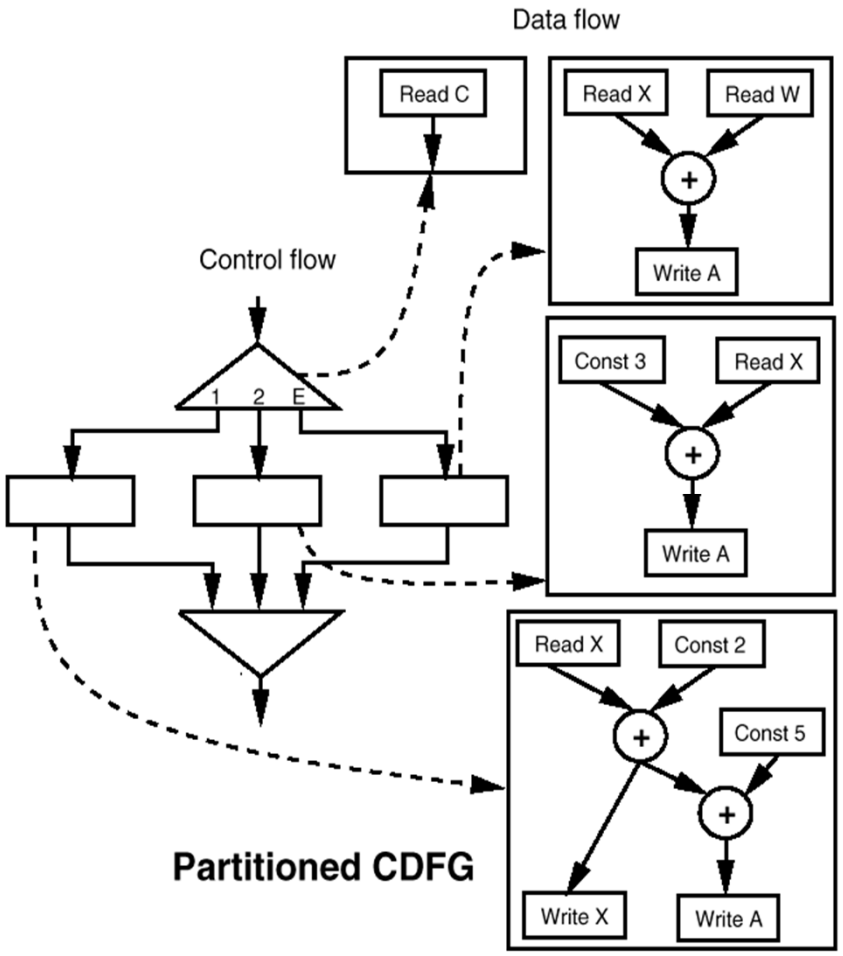
Flujo de diseño en SAN y representación de datos: Diseño resultante



Técnicas de representación del comportamiento

- Representación separada del flujo de datos y de control
  - Mantiene las construcciones de control originales de forma muy explícita
  - Nodos operativos del CFG apuntan a grafos parciales de flujo de datos. Se debe analizar el flujo de datos global
- Ejemplo

```
1. case C is
2.     when 1 =>  X := X + 2;
3.                A := X + 5;
4.     when 2 =>  A := X + 3;
5.     when others => A := X + W;
6. end case;
```



Técnicas de representación del comportamiento

- Representación híbrida
  - Incorpora en un solo grafo la información relevante de los grafos de flujo de datos y de control
  - Ocultación de nombres de variables (traza de valores)
- Ejemplo

```
1. case C is
2.     when 1 =>  X := X + 2;
3.                A := X + 5;
4.     when 2 =>  A := X + 3;
5.     when others => A := X + W;
6. end case;
```