Introducción a los Circuitos Integrados

Circuito Integrado

- **Circuito: Conjunto de conductores que recorre una corriente eléctrica, y en el cual hay generalmente intercalados aparatos productores o consumidores de esta corriente.**
- **Circuito integrado: Combinación de elementos de circuito miniaturizados que se alojan en un único soporte o chip, generalmente de silicio.**

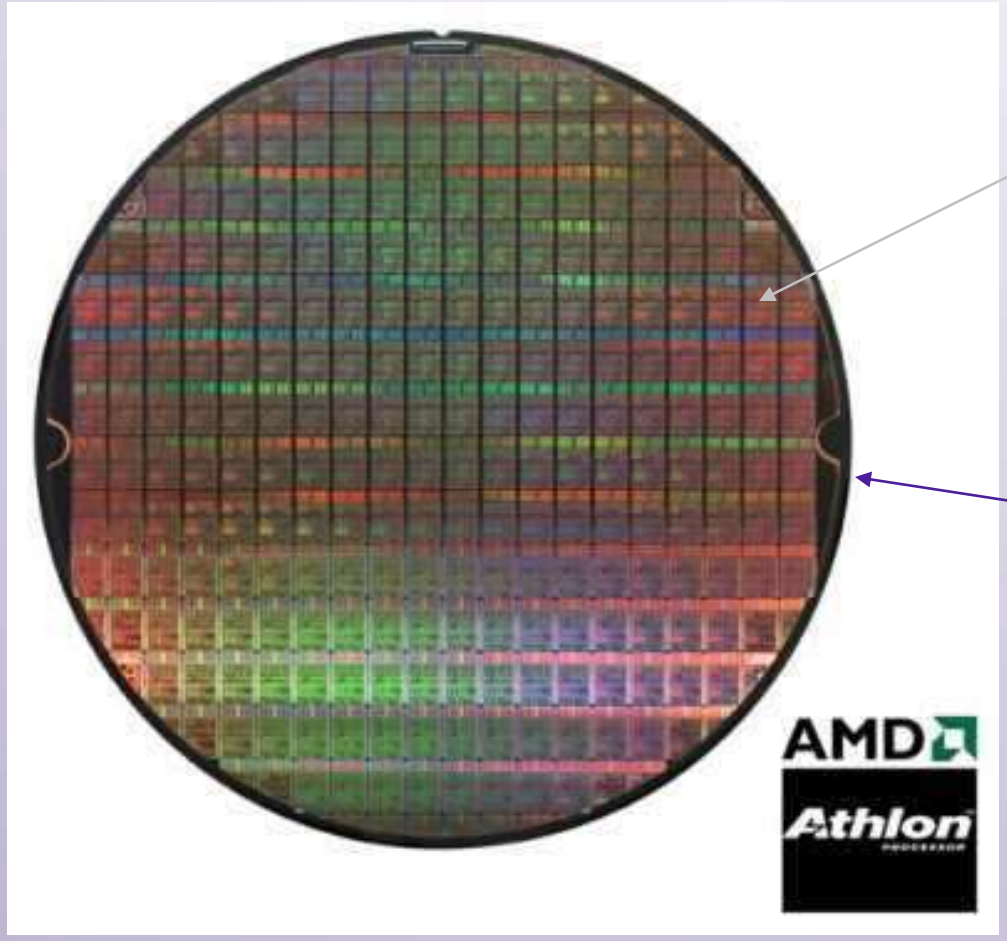
Circuito Integrado

- **Circuito integrado: Conjunto de transistores y circuitos eléctricos construidos sobre un mismo cristal. Los circuitos integrados actuales no miden más de un centímetro de largo y pueden contener millones de transistores.**

Algunas Definiciones

- **MSI: Medium Scale Integration, tipo de integración de chip capaz de albergar entre 10 y 500 transistores.**
- **LSI: Large Scale Integration, tipo de integración de chip capaz de albergar entre 1.000 y 10.000 transistores.**
- **VLSI: Very Large Scale Integration, tipo de integración de chip capaz de albergar sobre 100.000 transistores.**
- **ULSI: Ultra Large Scale Integration, tipo de integración de chip capaz de albergar sobre 10.000 circuitos.**
- **Hoy en día VLSI y ULSI se confunden**

Die



die

wafer

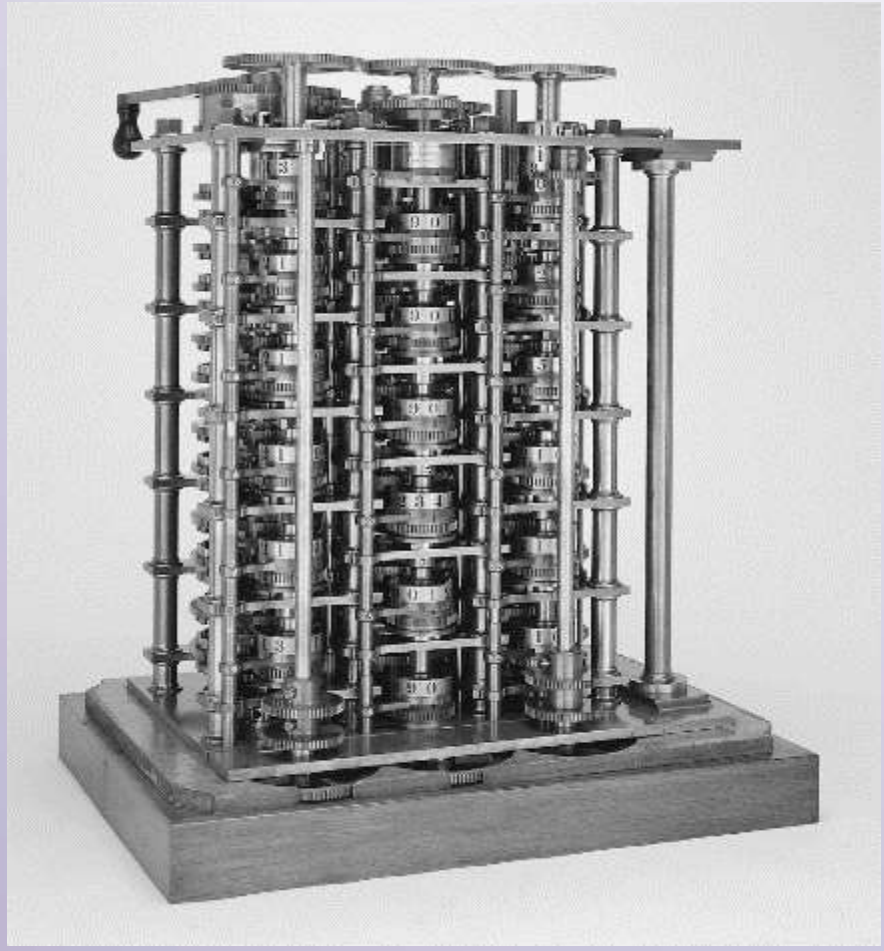
Algunas Definiciones

- **Die Size: Describe erróneamente el tamaño menor de los transistores en el chip. Corresponde al largo y ancho del circuito en la oblea de silicio.**
- **ASIC: Application Specific Integrated Circuit, circuito diseñado para una aplicación específica en oposición a los circuitos de propósito general como los microprocesadores. El uso de ASICs como componentes en los dispositivos electrónicos permite mejorar el rendimiento, reducir el consumo de potencia, mejorar la seguridad y reducir los costos .**

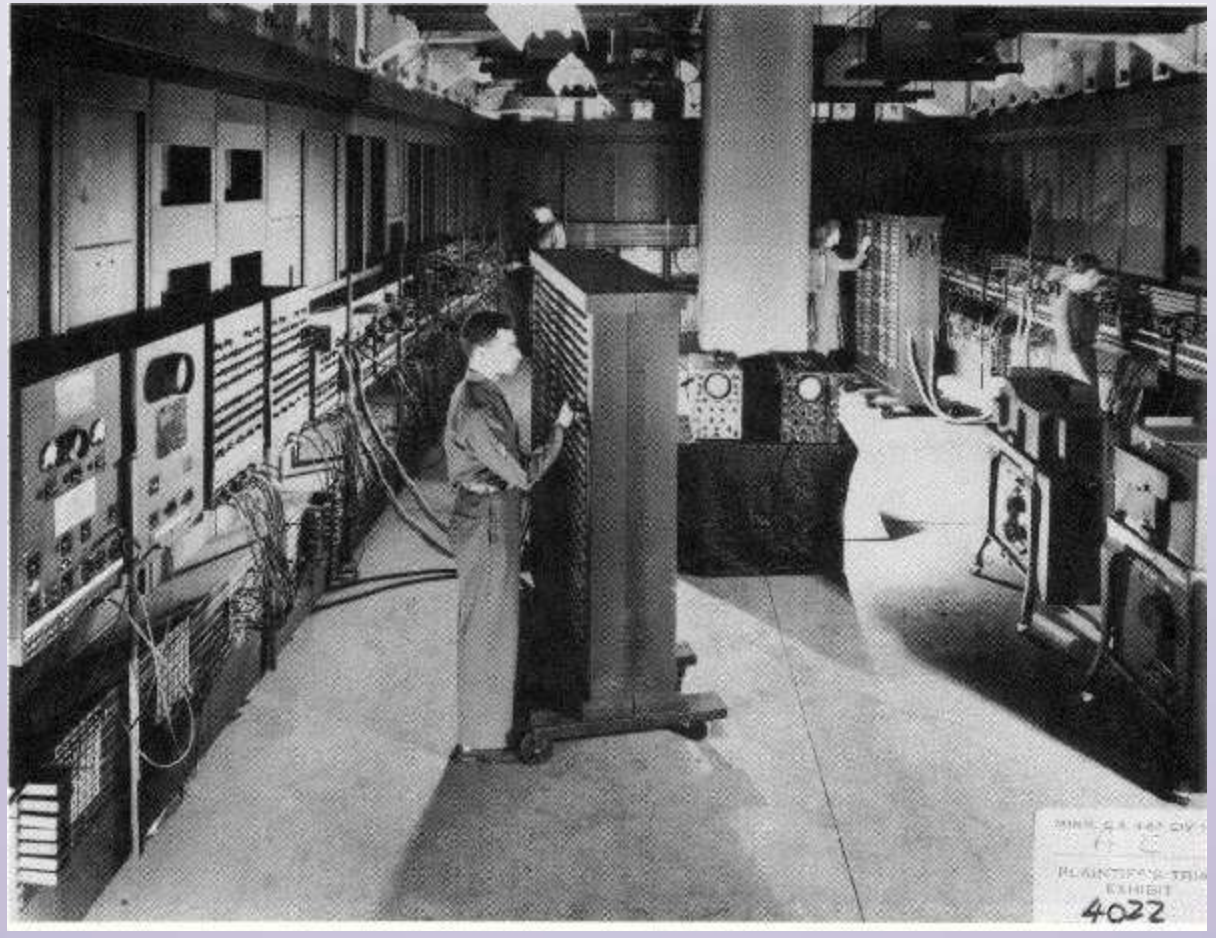
Algunas Definiciones

- **Síntesis lógica es el procesos por el cual las descripciones algorítmicas de circuitos son convertidas en un diseño de hardware. Ejemplos de este proceso incluyen la síntesis de Lenguajes de Descripción de Hardware (HDL) tales como VHDL y Verilog. El resultado de un proceso de síntesis puede ser un PAL, un FPGA o un ASIC.**
- **Compilador de silicio es un software que a partir de una especificación del usuario genera un circuito integrado.**

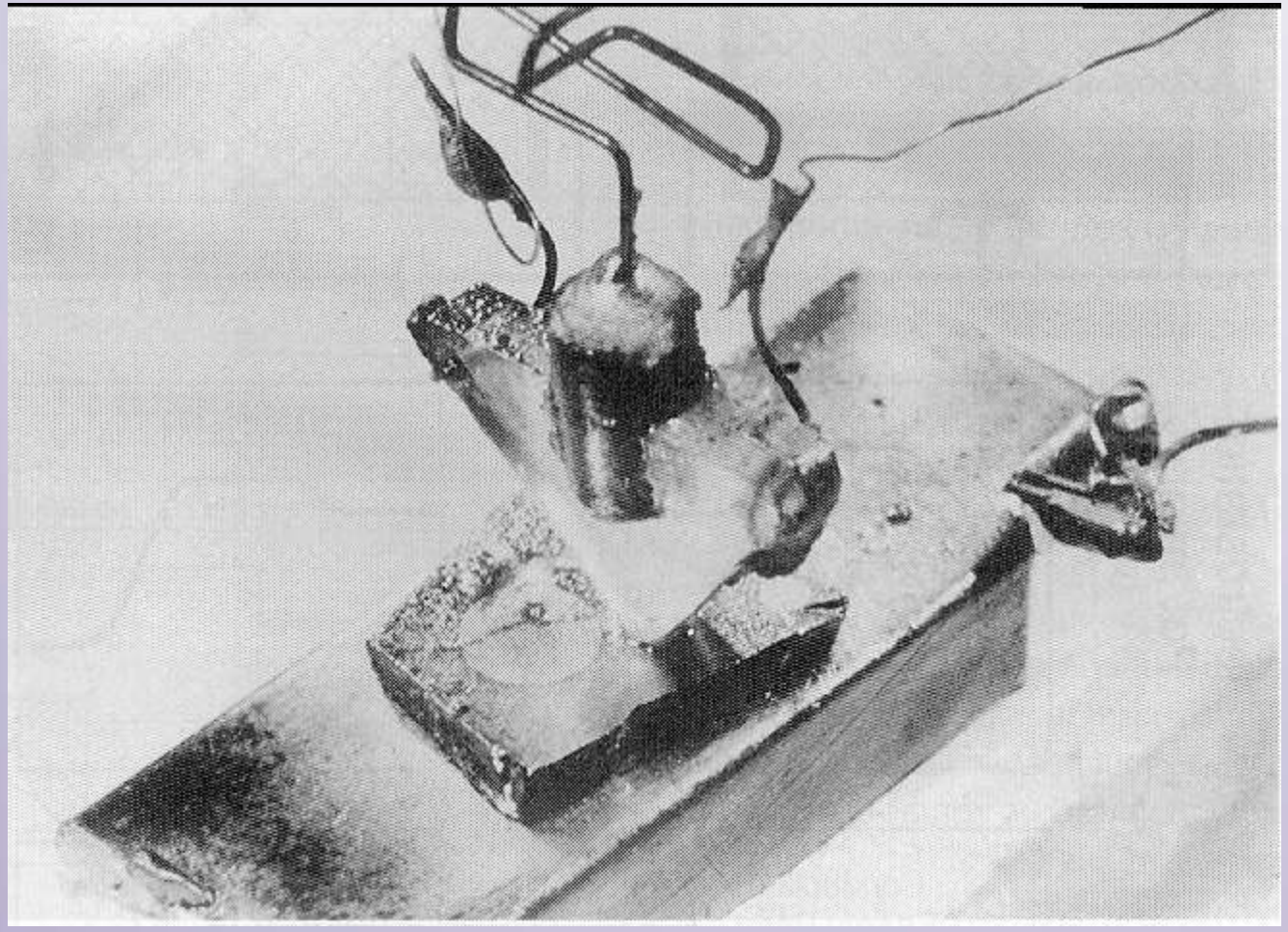
El Primer Computador



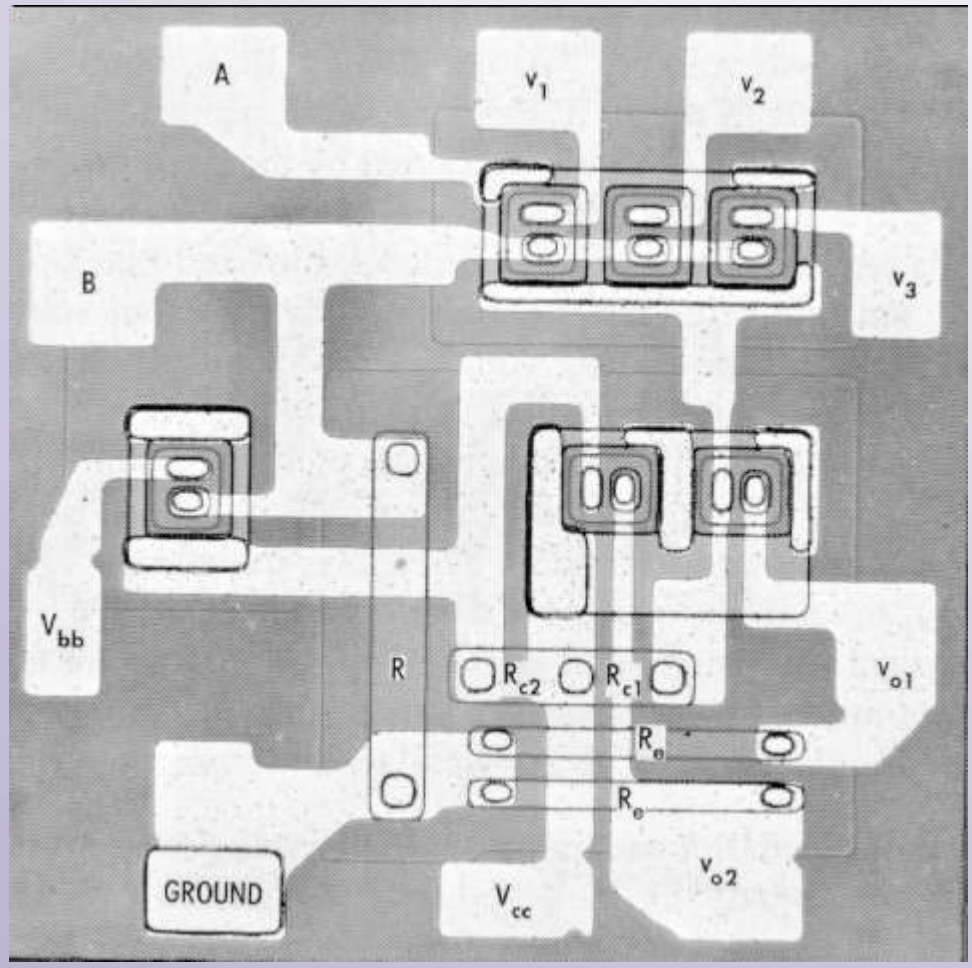
ENIAC – El Primer Computador Electrónico (1946)



El Primer Transistor



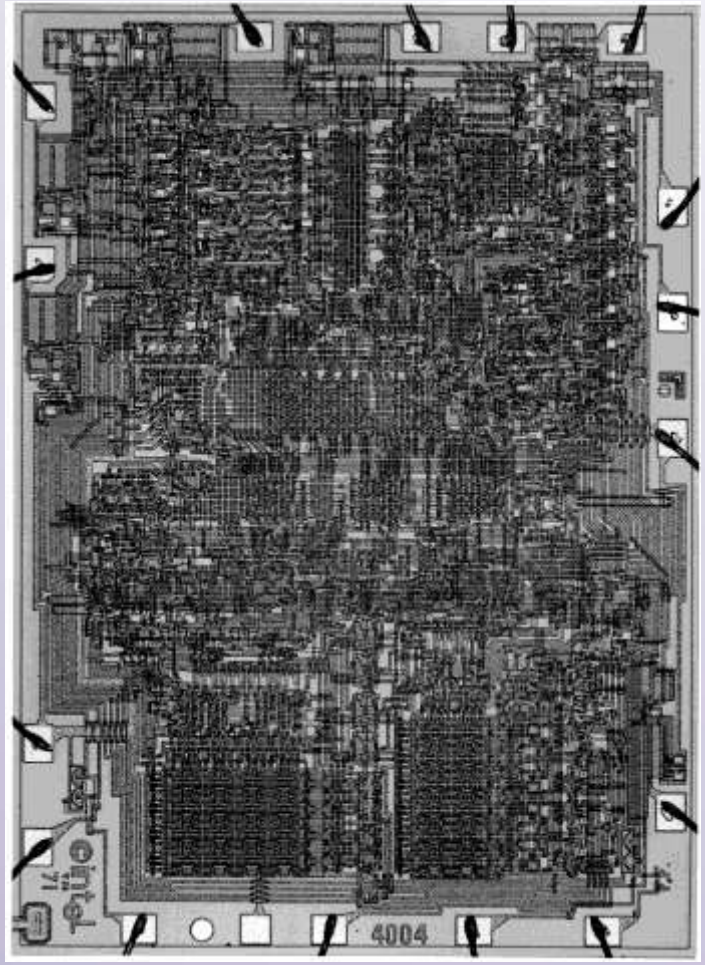
El Primer Circuito Integrado



Lógica bipolar
1960

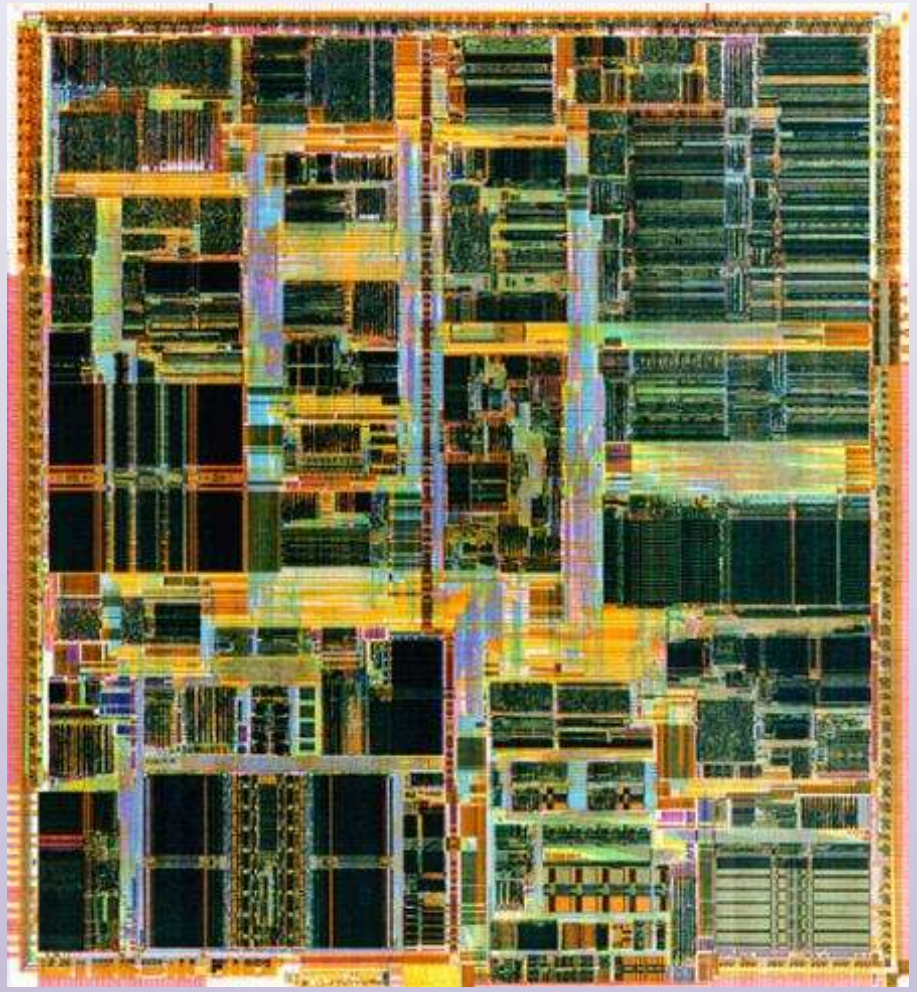
ECL 3-input Gate
Motorola 1966

Microprocesador 4004 - Intel



1971
1000 transistores
1 MHz operación

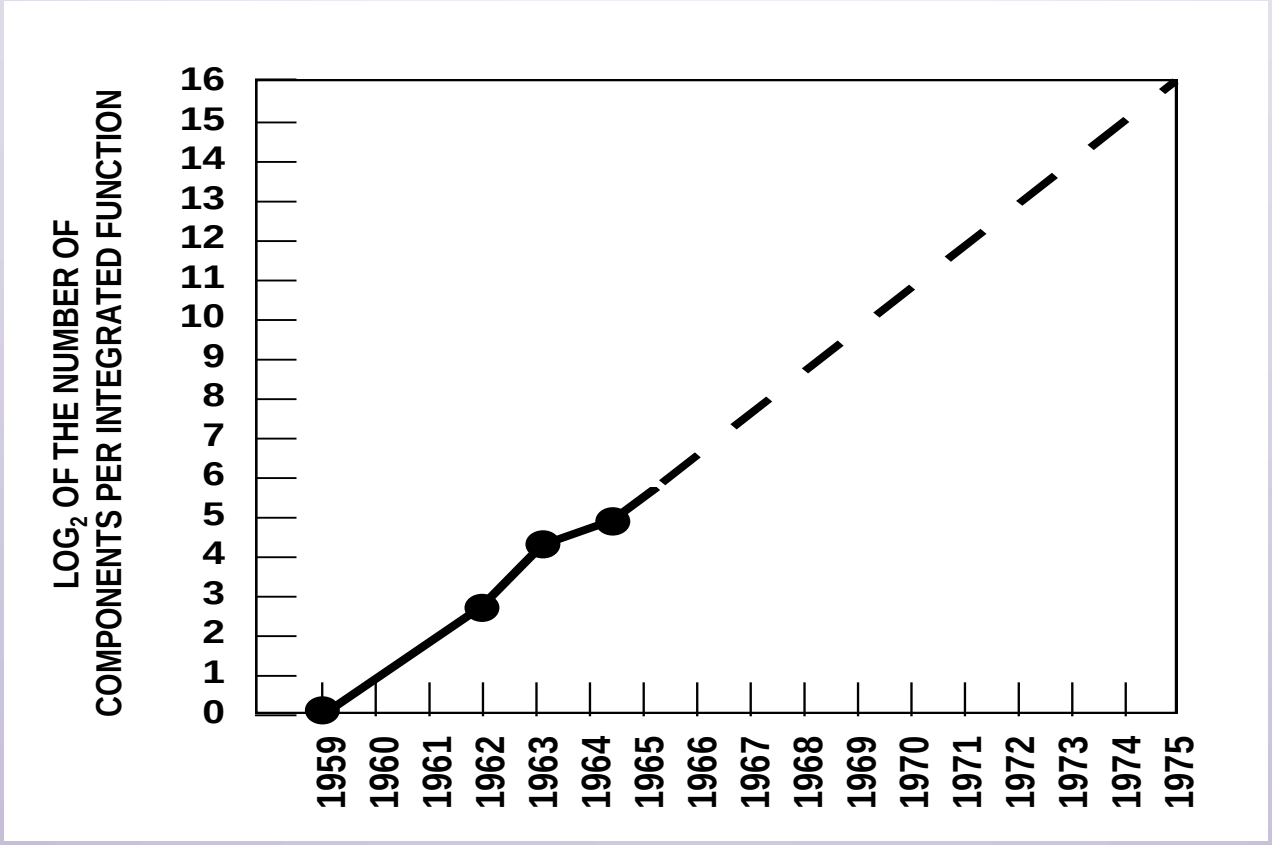
Microprocesador Pentium IV - Intel



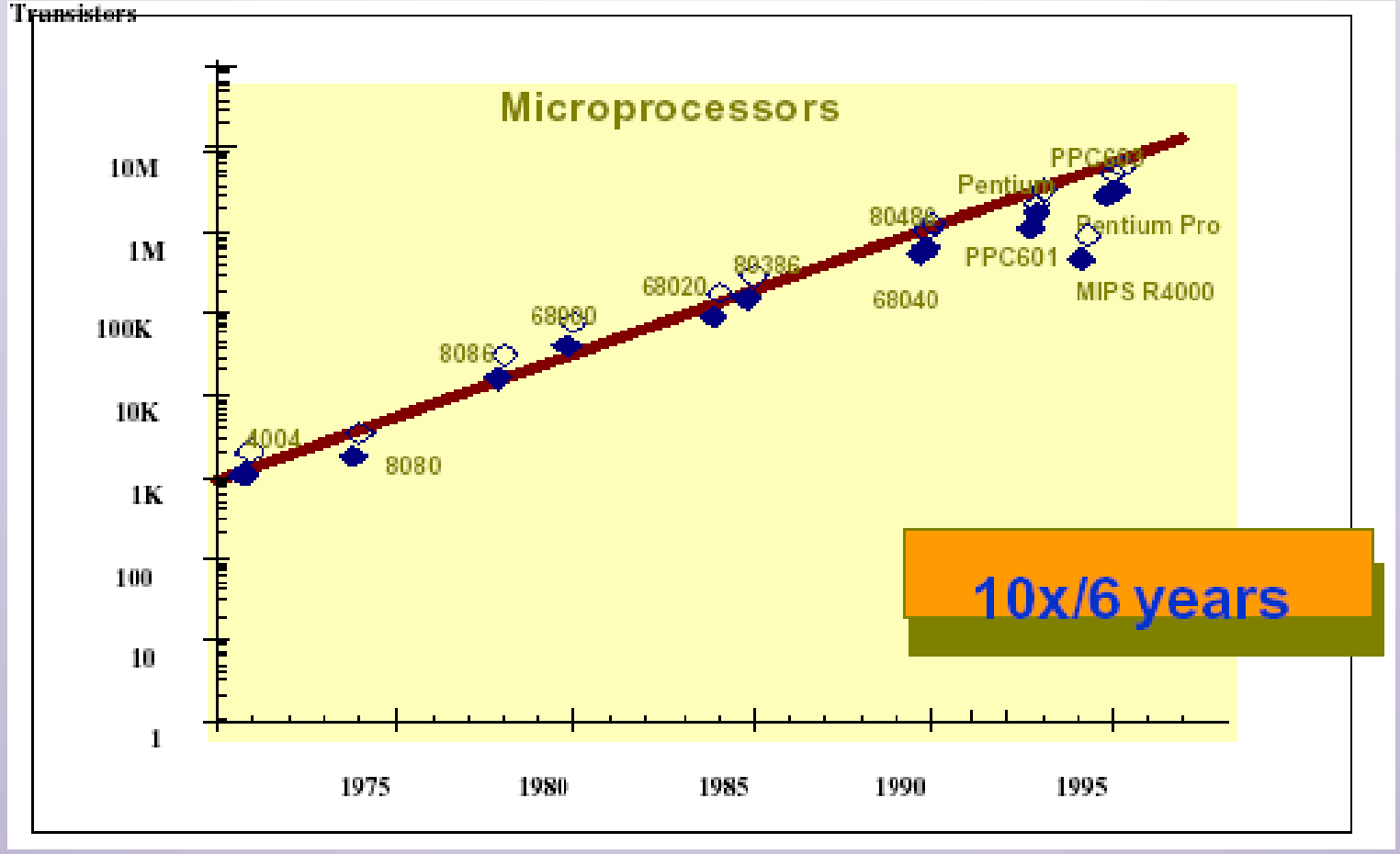
Ley de Moore

- **En 1965, Gordon Moore, co-fundador de Intel observó que el número de transistores en un chip se duplicaba cada 18 a 24 meses.**
- **A partir de esta observación predijo que la tecnología de semiconductores duplicaría su efectividad cada 18 meses.**

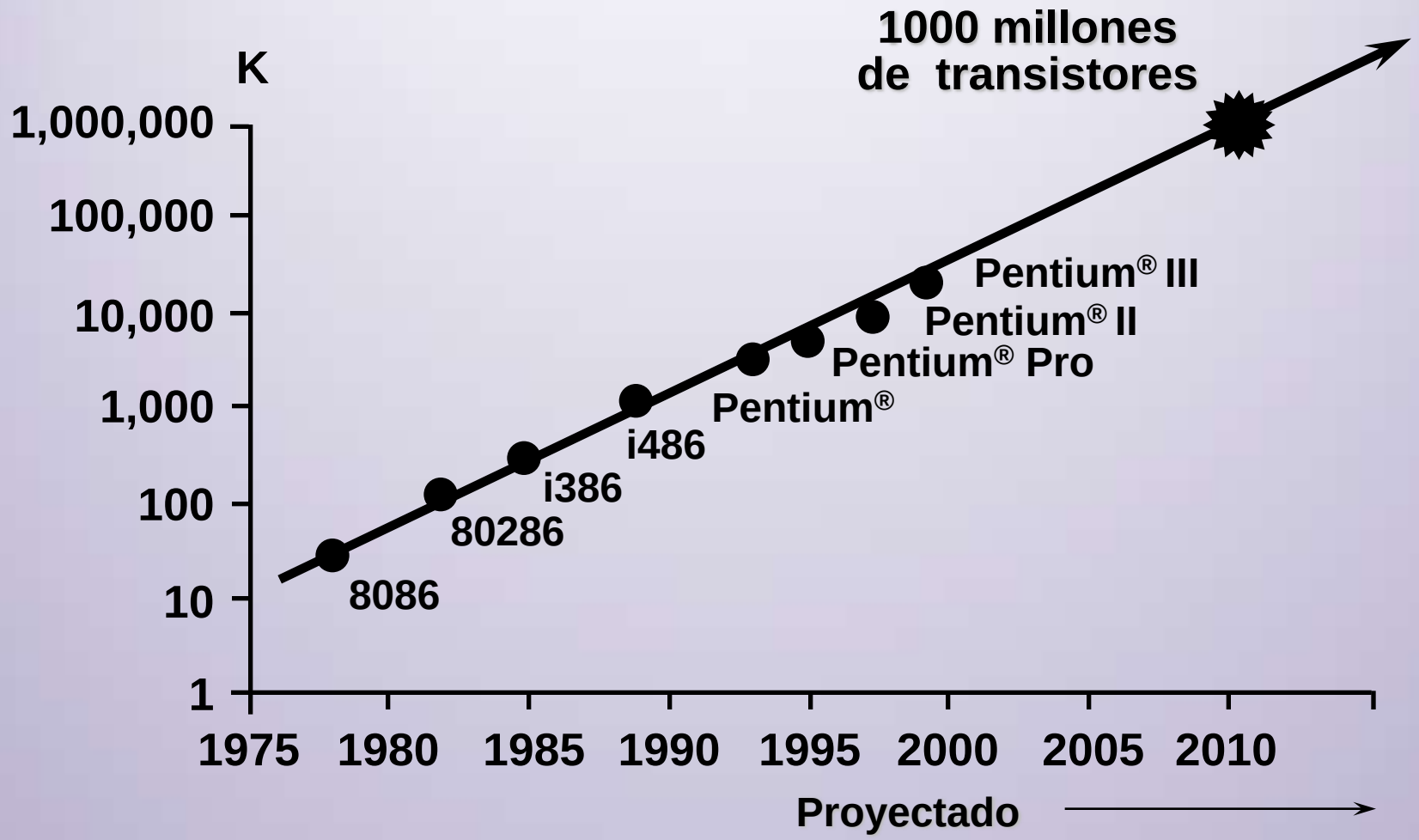
Ley de Moore


LOG2 OF THE NUMBER OF COMPONENTS PER INTEGRATED FUNCTION
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975

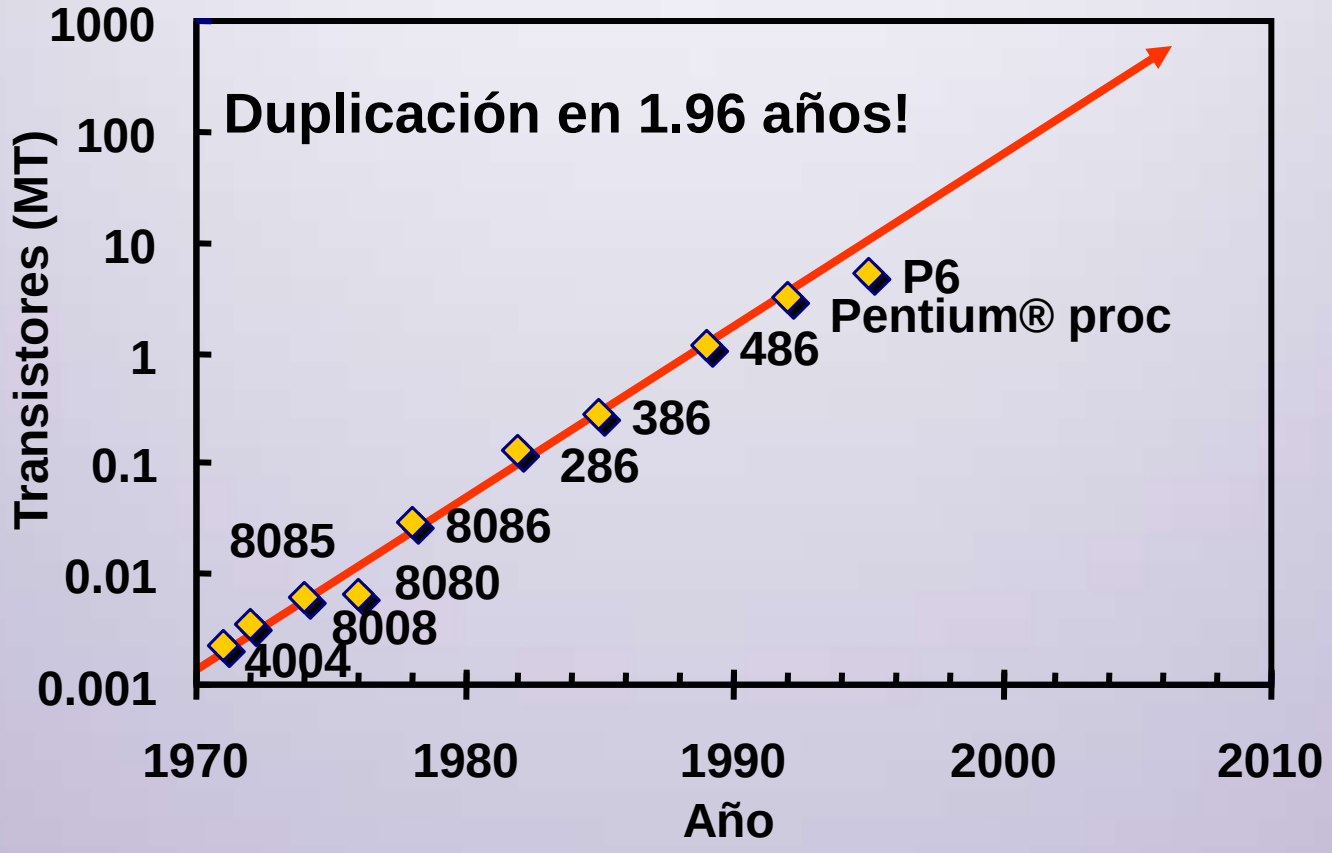
Ley de Moore


Transistors
Microprocessors
10M
1M
100K
10K
1K
100
10
1
1975
1980
1985
1990
1995
4004
8080
8086
68000
68020
80386
80486
68040
Pentium
PPC601
Pentium Pro
MIPS R4000
10x/6 years

Número de Transistores

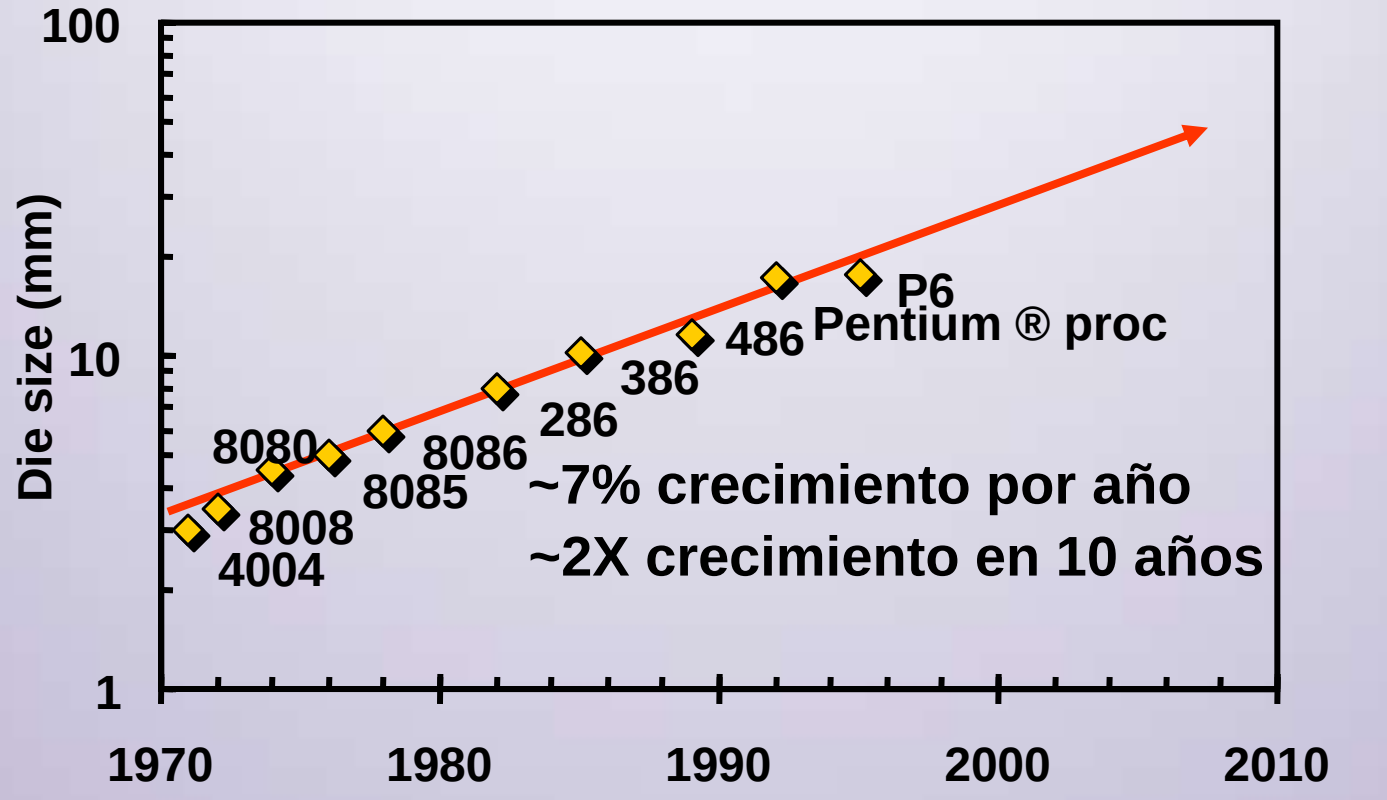


Ley de Moore en Microprocesadores



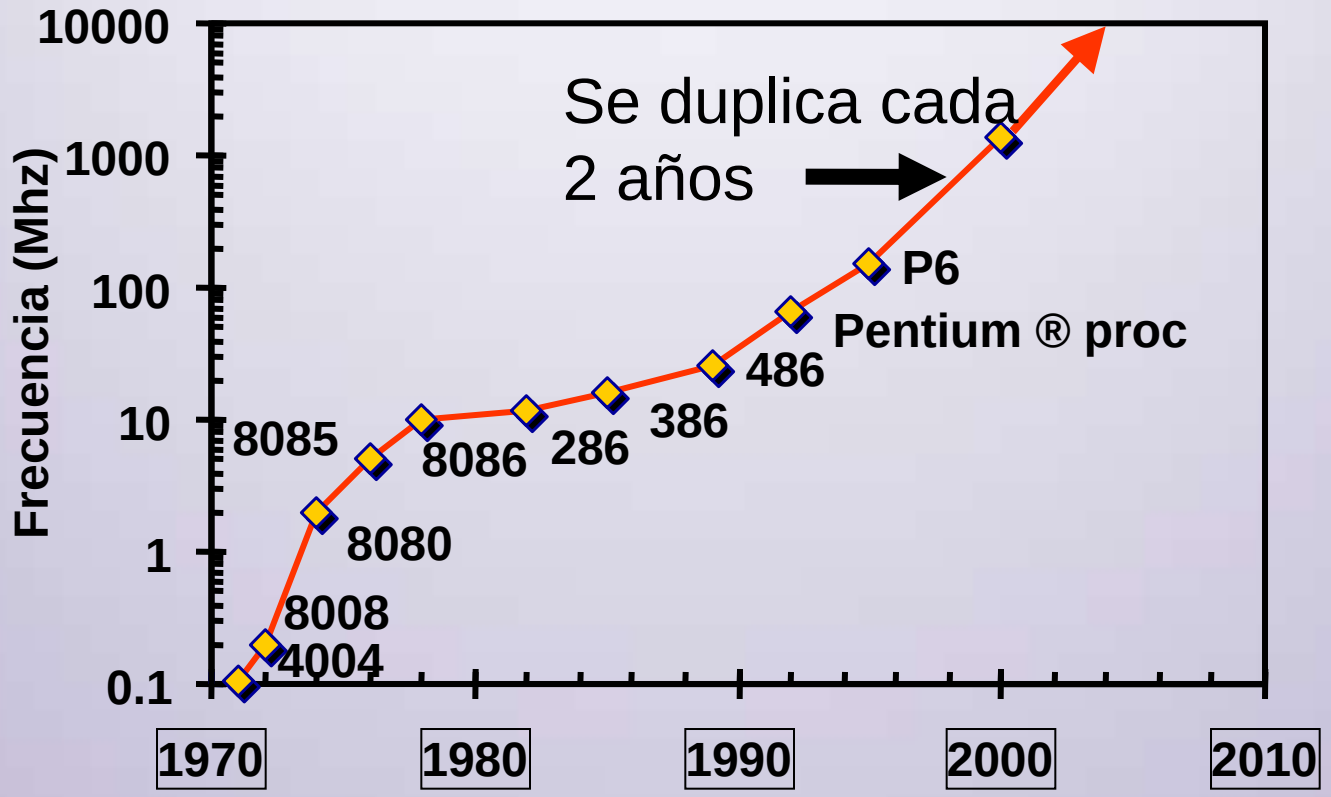
El número de transistores en microprocesadores se duplica cada dos años

Crecimiento del Die Size



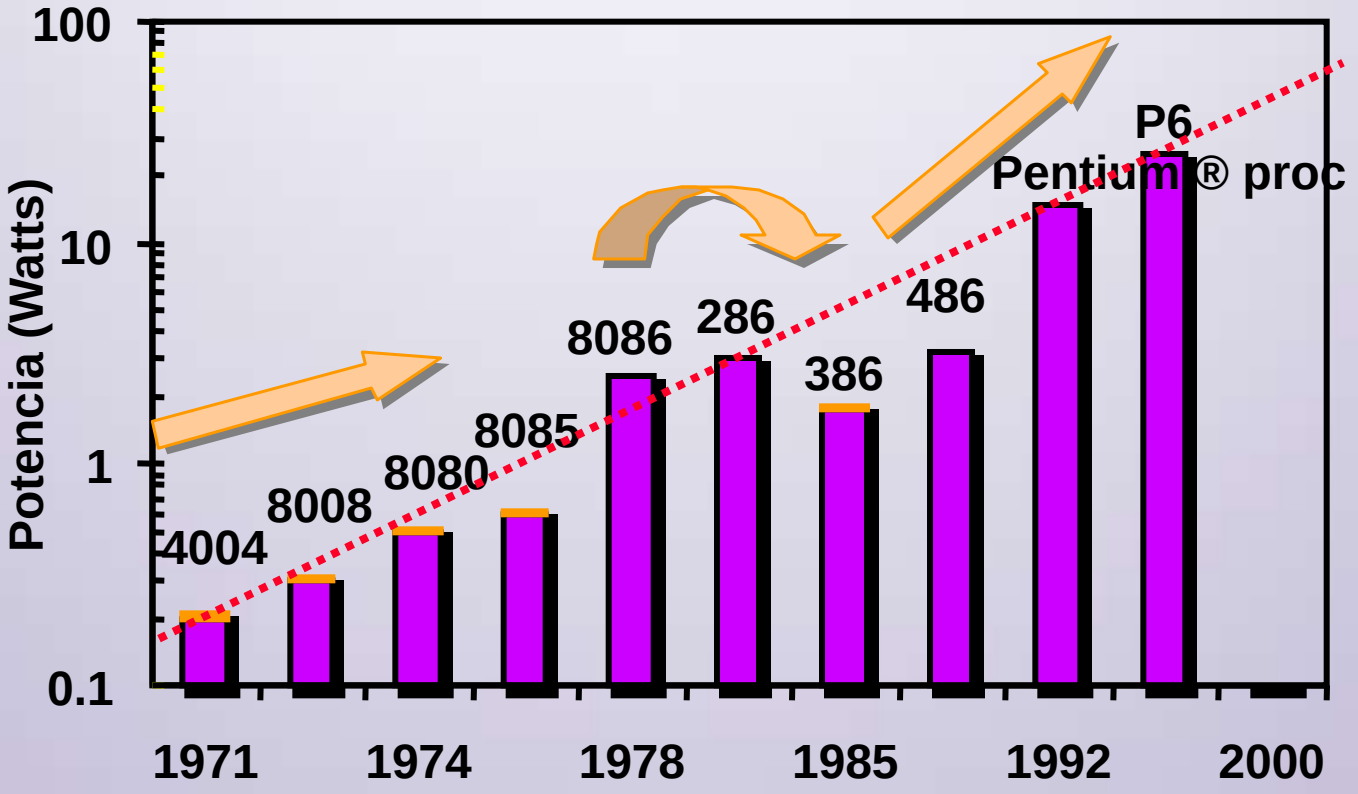
Die size crece 14% para satisfacer la ley de Moore

Frecuencia



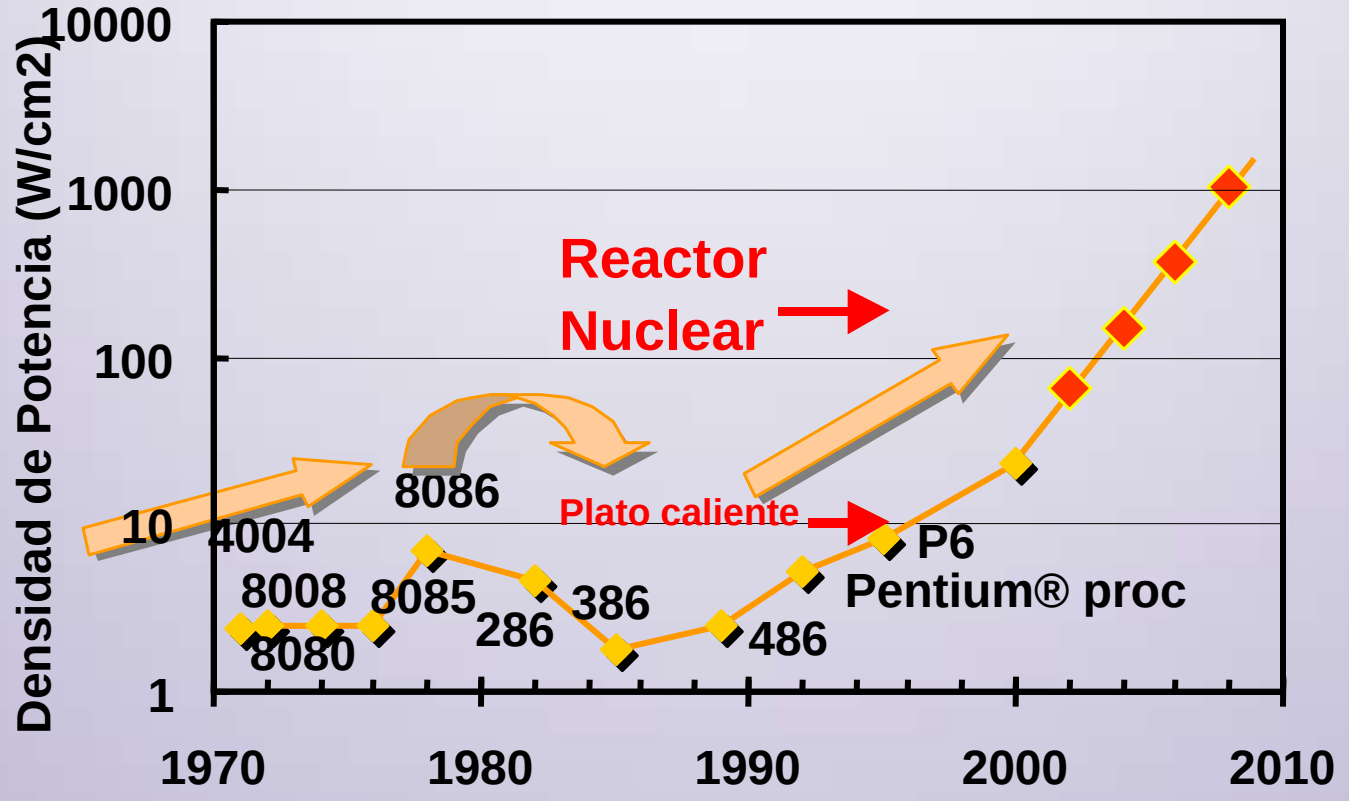
La frecuencia en microprocesadores se duplica cada 2 años

Disipación de Potencia



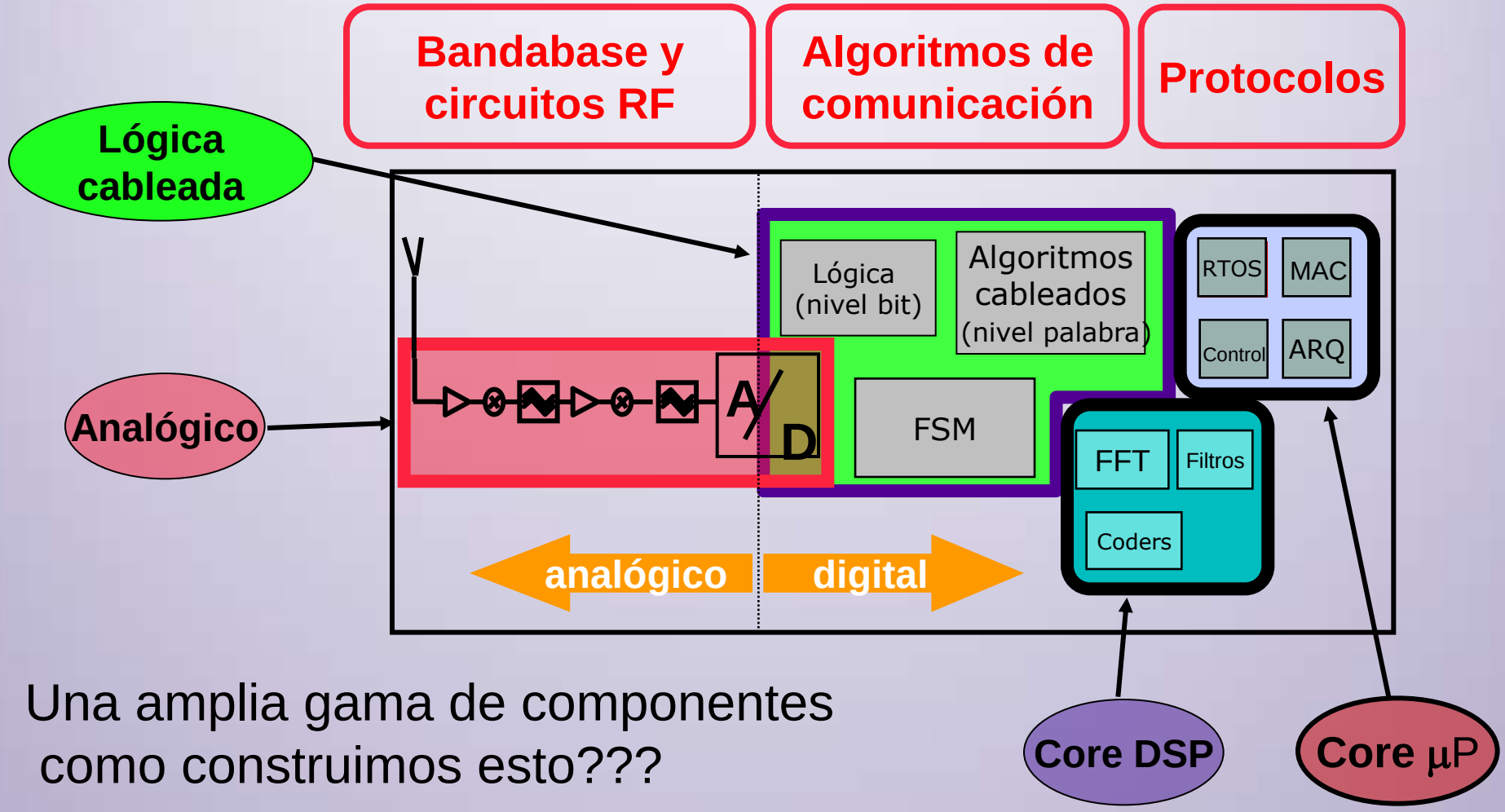
La potencia de los microprocesadores continua creciendo

Densidad de Potencia



La densidad de potencia es muy alta para mantener la juntura a baja Tº

Sistema Inalámbrico



Una amplia gama de componentes como construimos esto???

¿Qué es un SoC?

SoC es un estilo de diseño y un tipo de producto

¿Qué es un SoC?

Un chip diseñado con la funcionalidad “completa” de un sistema que incorpora una mezcla heterogénea de arquitecturas de proceso y de computación

¿Qué es un SoC?

- **Mezcla de CPUs, memoria, y periféricos en un chip**
- **Mezcla de bloques sintetizados y bloques custom (macros hechas por hardware)**
- **Para productos con restricciones de costo y time-to-market**

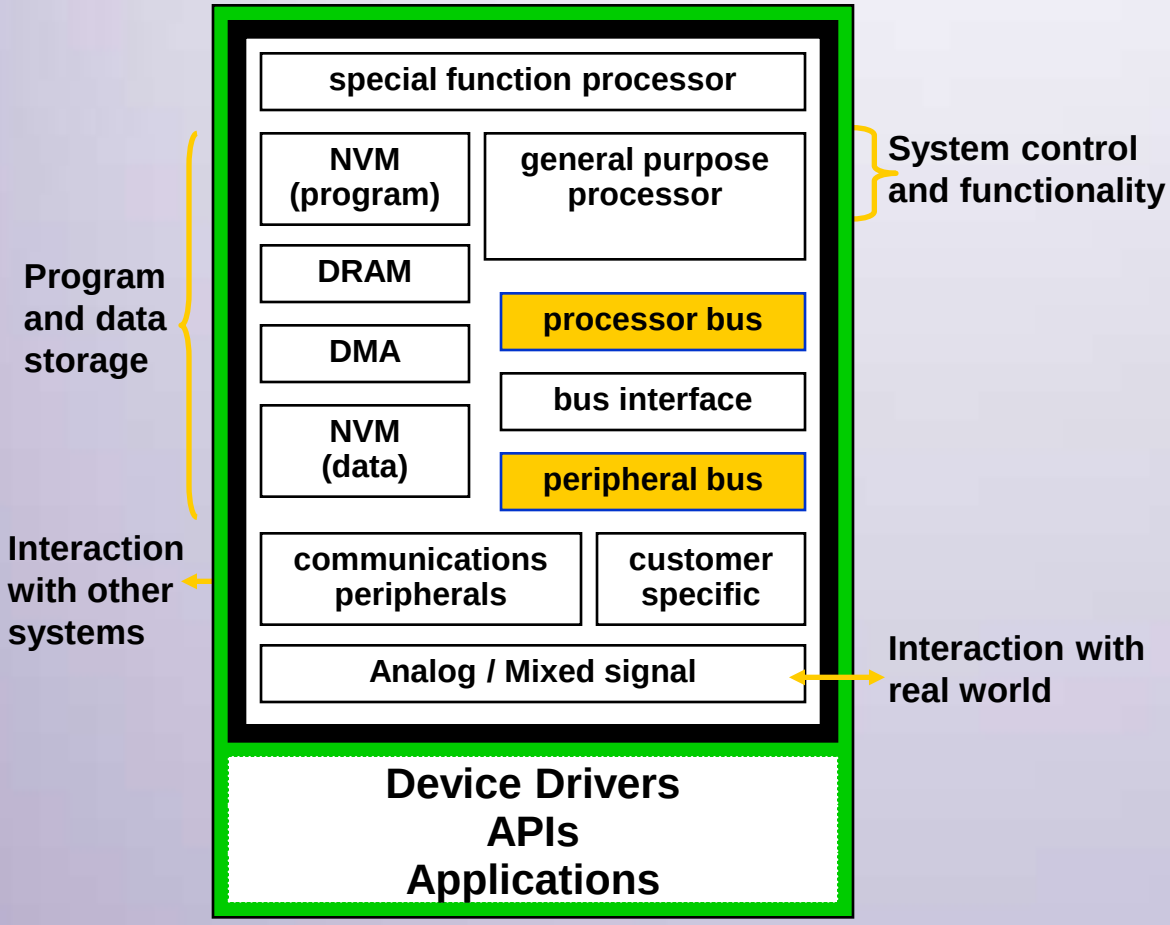
¿Qué es un SoC?

- **Implicancias metodológicas:**
  - **Diseño de bloques IP usando estándares estrictos para creación y reusabilidad**
  - **Uso de definiciones estándares de interfaz**
  - **Combinación de alto nivel – “estilo ASIC” – usando flujos y herramientas estándares**

SoC es

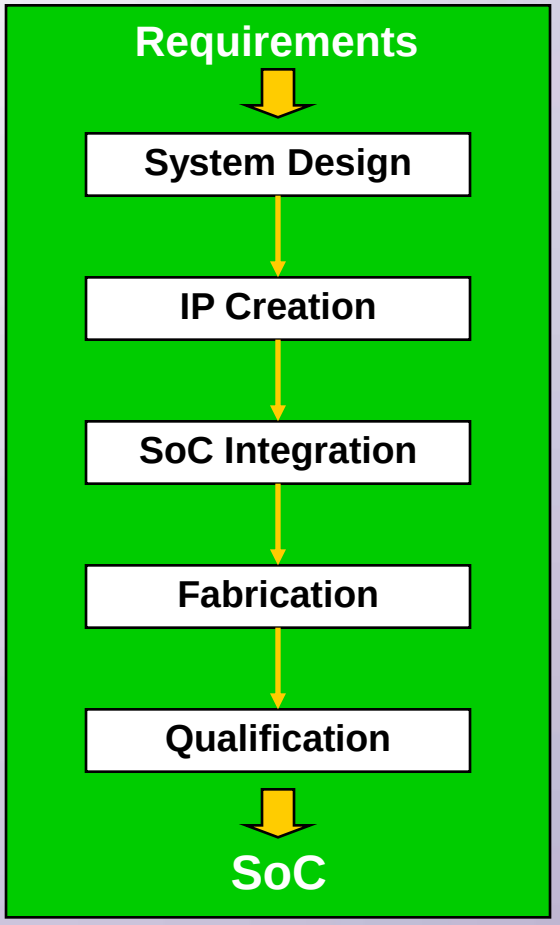
… un producto ...

Soluciones para aplicaciones específicas que implementan sistemas enteros



...y un proceso.

Del sistema al silicio en un time-to-market rápido.



Impulso de SoC

Dos fuerzas trabajan en conjunto en la industria electrónica:

Los proveedores de sistemas deben **diferenciar productos** a través de **aplicaciones de software**.

El dinamismo del mercado requiere:

- time-to-market rápido
- Bajo costo
- Curva de aprendizaje rápida

Geometrias pequeñas permiten:

Integración de alto rendimiento

Fabricantes de semiconductores deben **cubrir los costos de fabricación** a través de **sistemas de valor agregado**.

Desafíos del Diseño

Funcionalidad + Testabilidad

Retraso en cableado

Gestión de potencia

Software embebido

Integridad de las señales

Efectos RF

Chip híbridos

Packaging

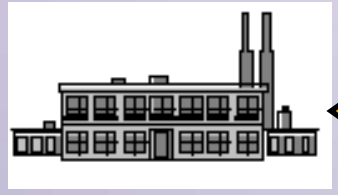
Limites físicos

Diseño de Chips – CAD

Sistemas electrónicos

Industria EDA

Mundo real



Foundries

Industria de semiconductores

Mayor Complejidad de Dispositivos y Contexto

- **Crecimiento exponencial de la complejidad de los dispositivos – ley de Moore.**
- **Crecimiento de la complejidad de los sistemas en los cuales se utilizan los dispositivos (ej. celular).**
- **Crecimiento de la productividad en diseño**

Hay exponencialmente más transistores

Efectos Submicrón

- **Las geometrías pequeñas causan diversos efectos que eran ignorados en el pasado**
  - **Capacitancias de acoplamiento**
  - **Integridad de señales**
  - **Resistencia**
  - **Inductancia**

El diseño de cada transistor es más difícil

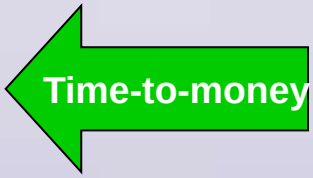
Heterogeneidad en el Chip

- **Gran diversidad de elementos en el chip**
  - **Procesadores**
  - **Software**
  - **Memoria**
  - **Análogo**

Más transistores hacen cosas diferentes

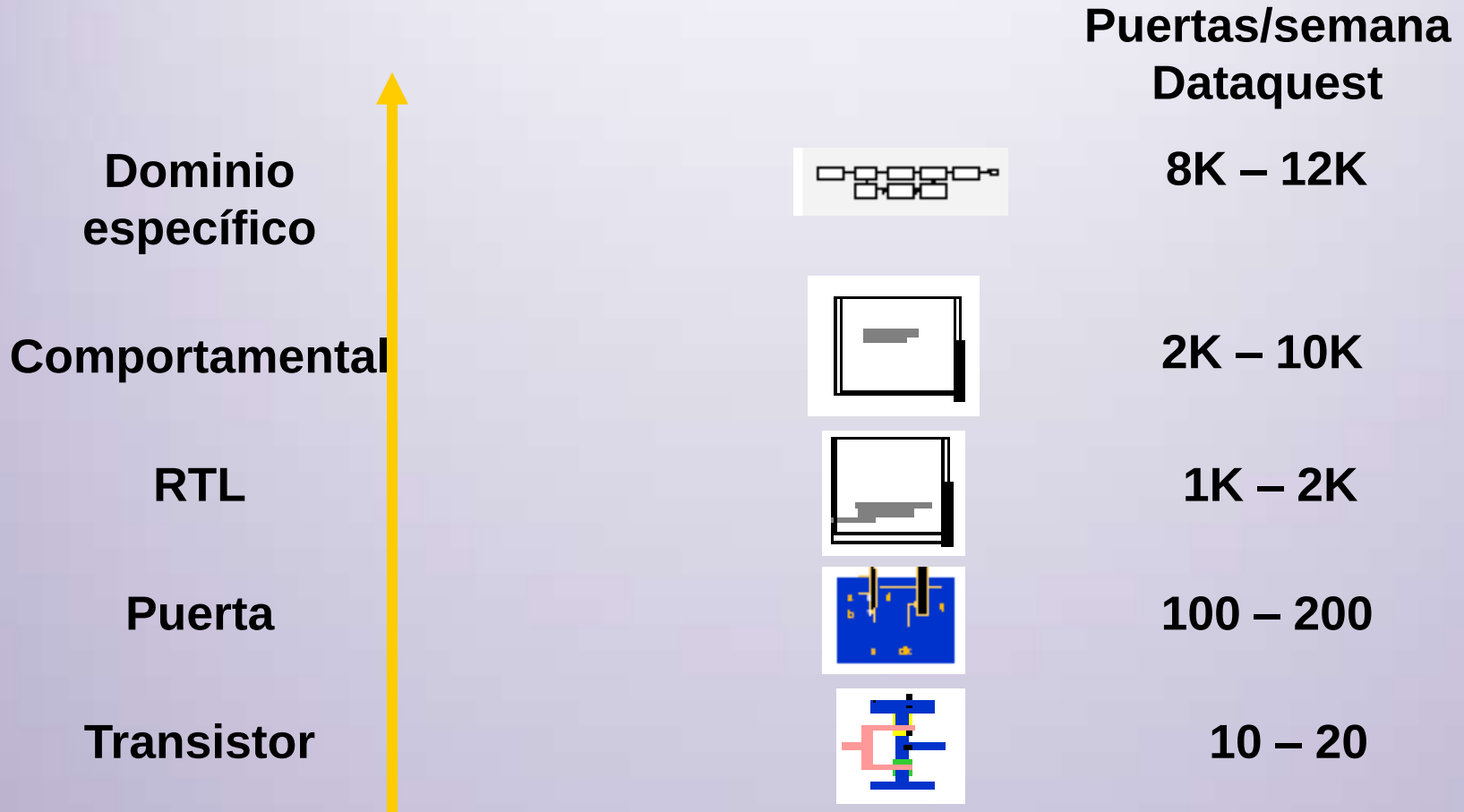
Fuerte Presión del Mercado

- **Ventana de diseño más pequeña**
- **Menor tolerancia a revisiones**

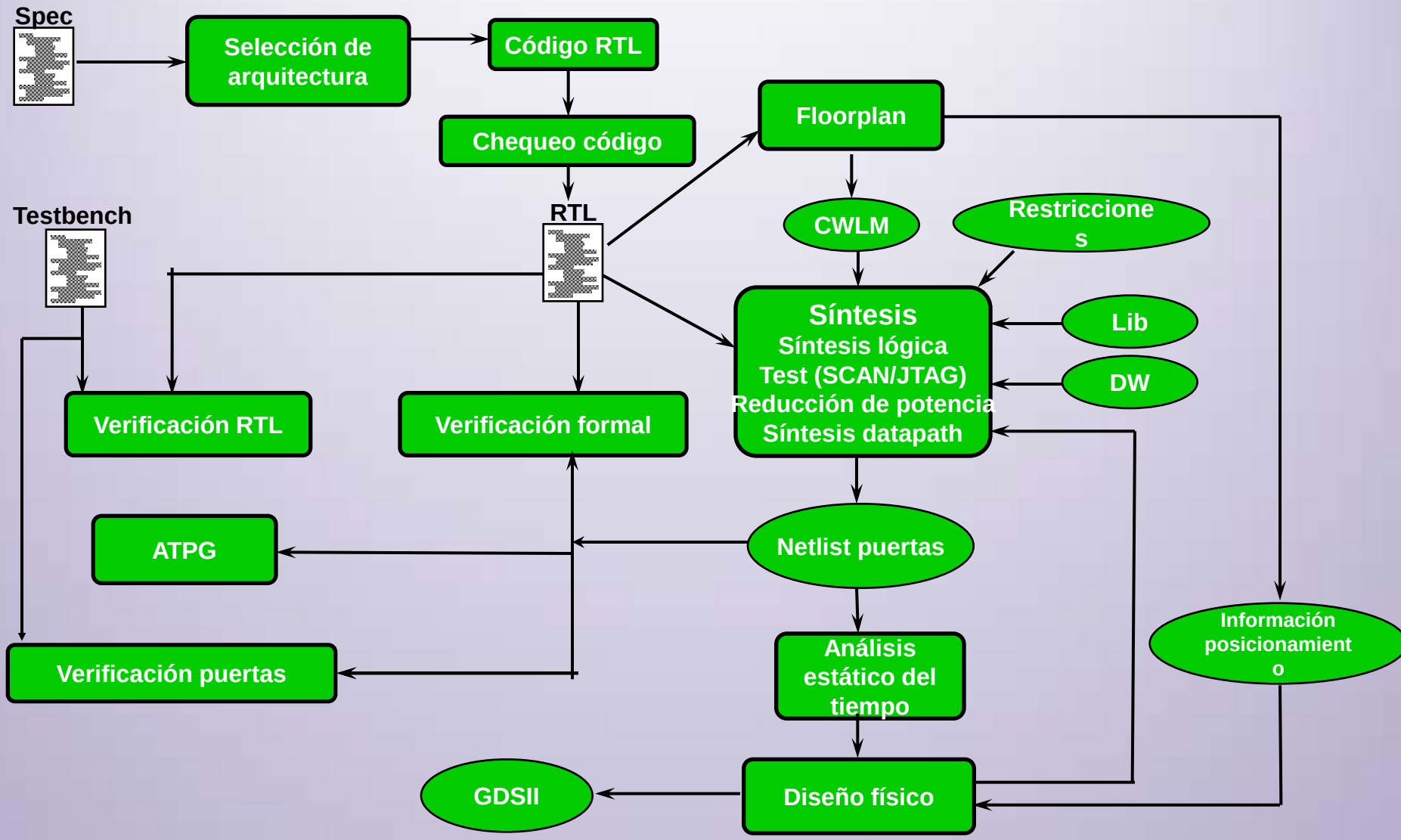


Mayor complejidad, mayor riesgo, mayor variedad, ventana más pequeña

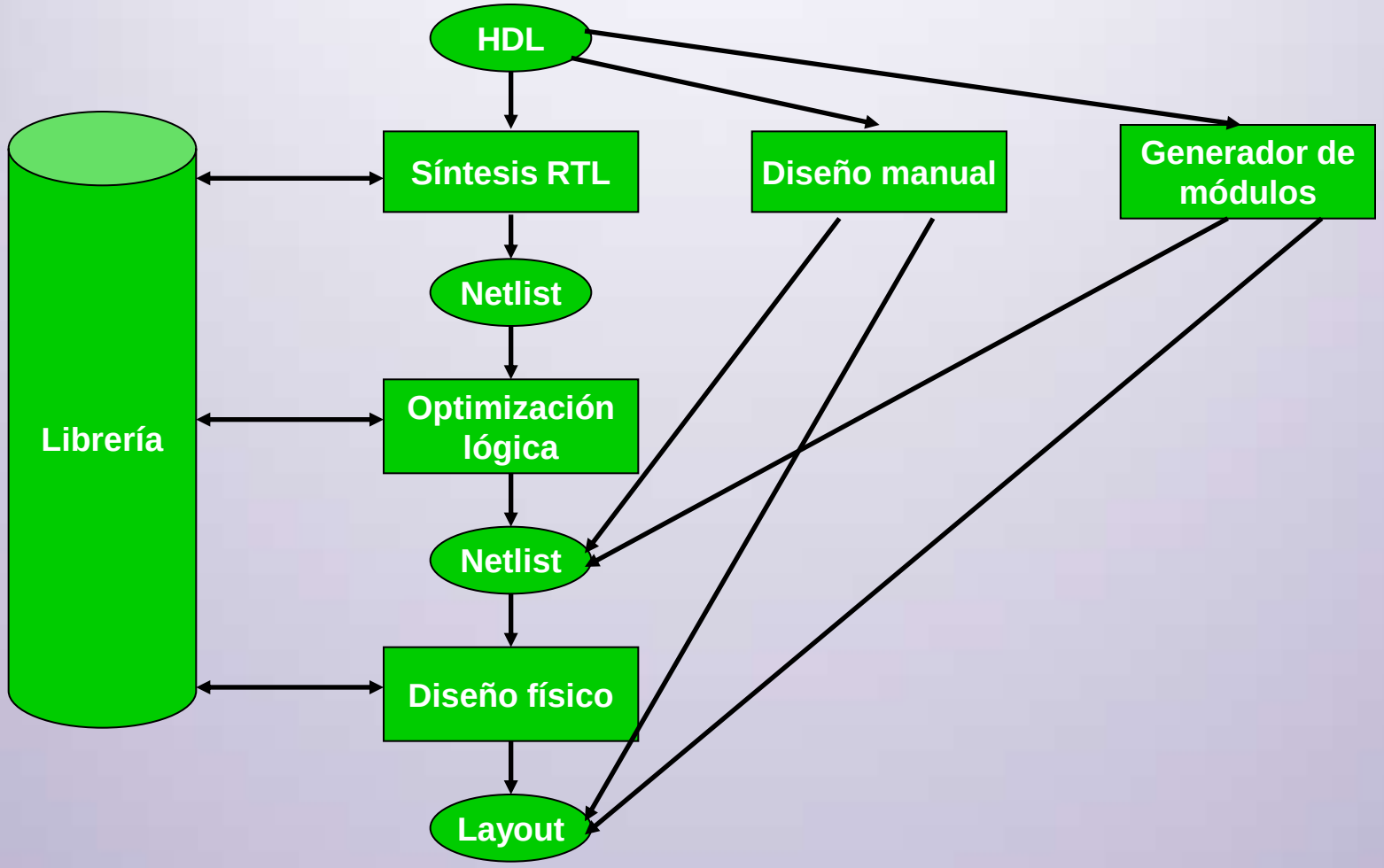
Productividad del Diseño



Flujo de Diseño



Flujo Simplificado



Diseño Manual

- Nivel compuerta (100 compuertas / semana)
- Nivel transistor (10 – 20 compuertas / semana)
- Excesivamente caro (costo y tiempo)
- Usado para
  - Analógico
  - Biblioteca de compuertas
  - Datapath en diseños de alto rendimiento

Generador de Módulos

- Generadores parametrizables de layout
- Generalmente usados en
  - Memorias
  - PLA
  - Register files
- Ocasionalmente usados para
  - Multiplicadores
  - Datapath de propósito general
  - Datapaths en diseños de alto rendimiento

Biblioteca

- **Contiene por cada celda**
  - **Información funcional**
  - **Información temporal**
  - **Información física (área)**
  - **Características de potencia**
  - **Modelos de simulación**

HDL a Nivel RTL

```
module foobar (q,clk,s,a,b);
   input clk, s, a, b;
   output q;req q; reg d;
always @(a or b or s) // mux
   begin
      if(!s)
         d = a;
      else if(s)
         d = b;
      else
         d = 'bx
   end //always
```

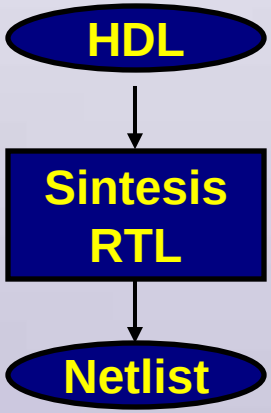
```
always @(clk) // latch
   begin
      if(clk == 1)
         q = d;
      else if(clk !== 0))
         q = 'bxb;

   end //always
End module
```

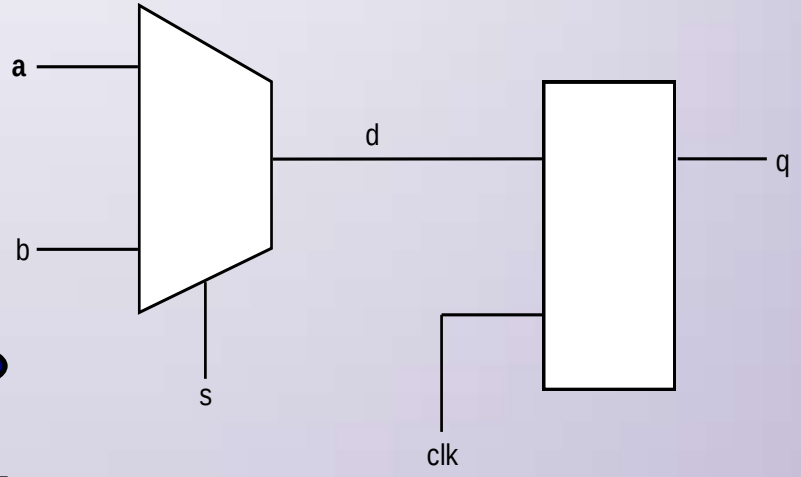
RTL

- **Implícitamente estructural**
  - **Los registros y su interconectividad están definidos**
  - **El comportamiento clock-to-clock está definido**
  - **Solo la lógica de control de transferencia es sintetizada**
- **Mejoras posibles**
  - **Asignación automática de recursos**

Sintesis RTL

```
module foobar (q,clk,s,a,b);
   input clk, s, a, b;
   output q;req q; reg d;
always @(a or b or s) // mux
   begin
      if(!s)
         d = a;
      else if(s)
         d = b;
      else
         d = 'bx
   end //always
```





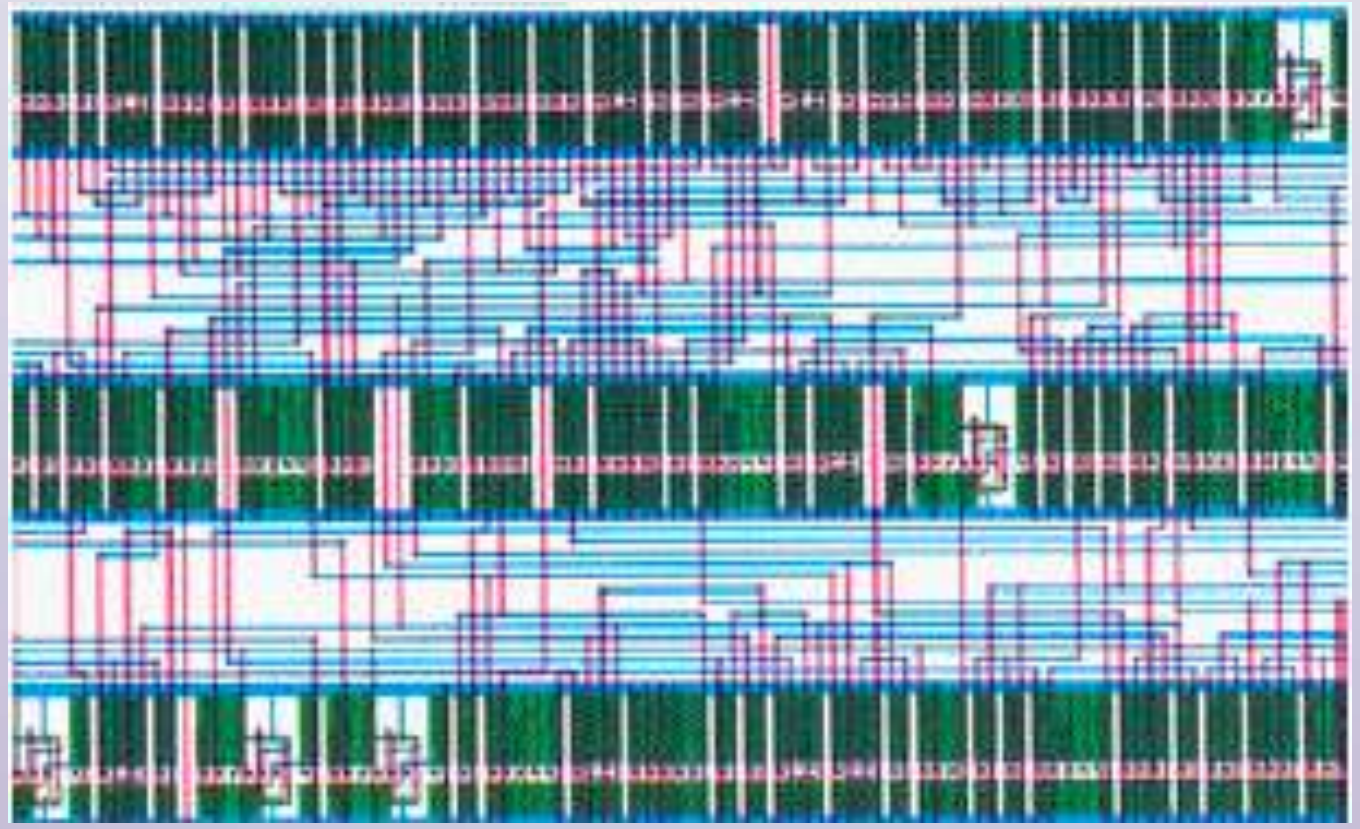
Optimización Lógica

- **Realiza transformaciones y optimizaciones**
  - **Transformación grafos estructurados**
  - **Transformaciones booleanas**
  - **Mapeo en una librería física**

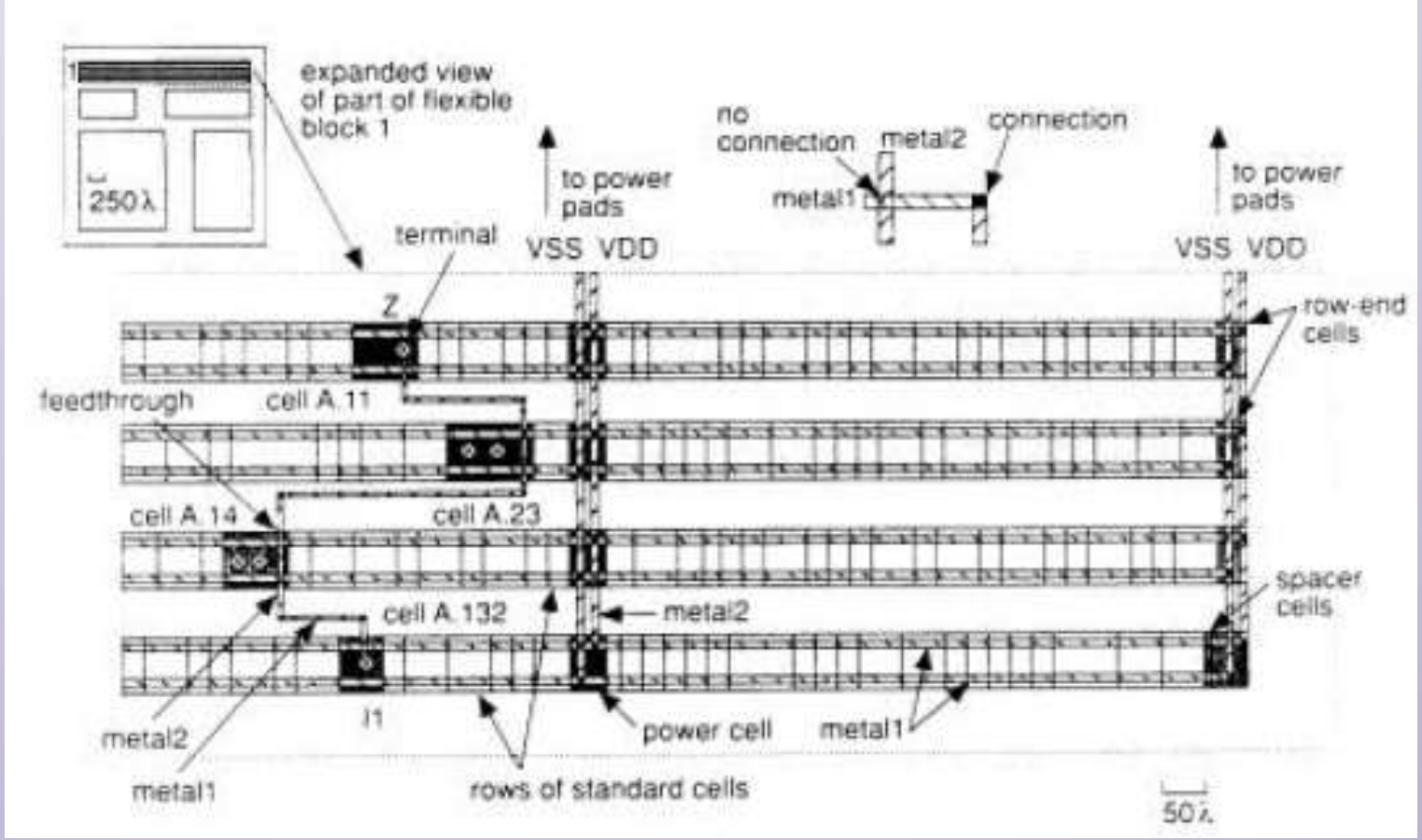
Diseño Físico

- Transforma circuitos secuenciales en circuitos físicos
  - Posiciona componentes
  - Rutea
  - Transforma en mascaras
- O FPGA
  - Posiciona tablas look-up
  - Rutea

Layout en Celdas Estándares



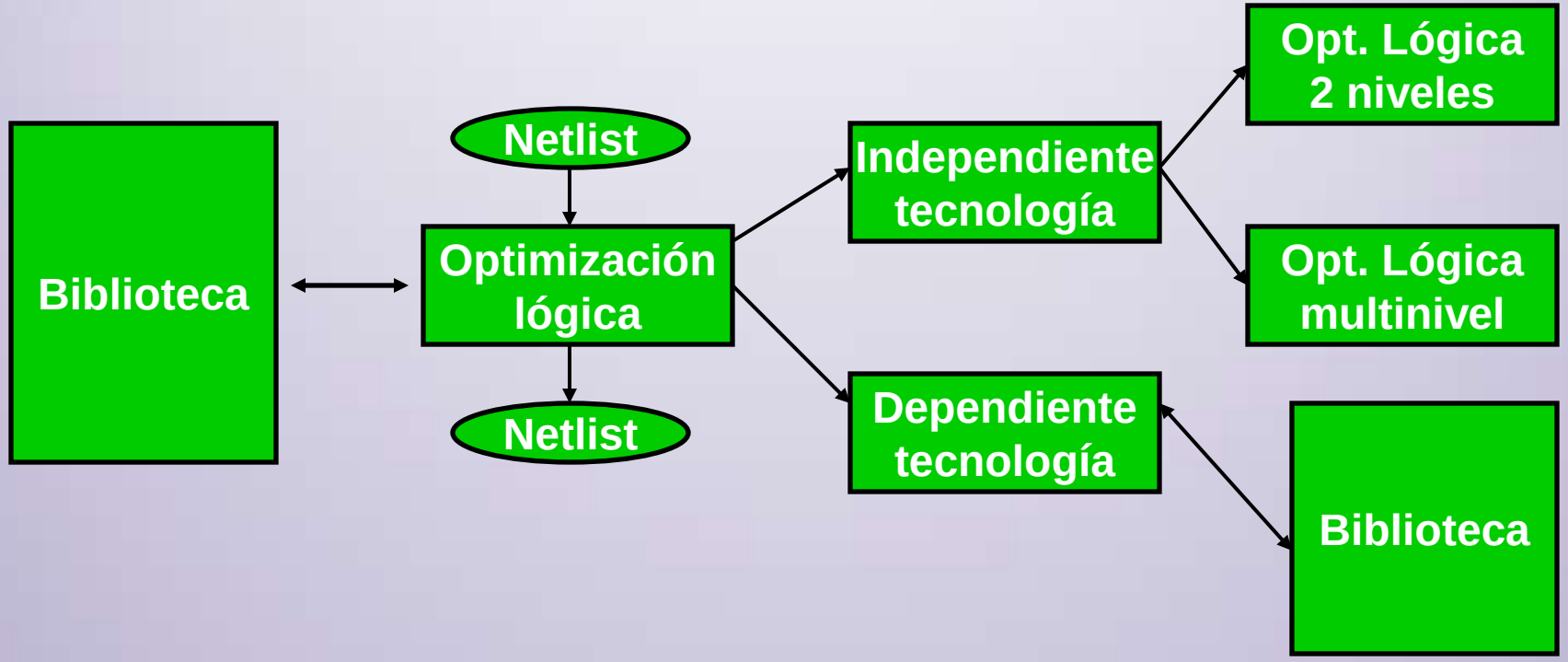
Gate Array


expanded view of part of flexible block 1
250λ
terminal
to power pads
VSS VDD
no connection
metal2
connection
metal1
to power pads
VSS VDD
Z
row-end cells
feedthrough
cell A.11
cell A.14
cell A.23
cell A.132
metal2
spacer cells
I1
metal2
metal1
power cell
metal1
rows of standard cells
50λ

Optimización Lógica Combinatoria

- **Entradas**
  - **Red booleana inicial**
  - **Caracterización temporal del módulo**
    - **Tiempo de llegada de entradas**
    - **Factores de carga**
  - **Objetivos de optimización**
    - **Tiempos requeridos**
    - **Superficie**
  - **Descripción librería a usar**
- **Salida**
  - **Netlist con área mínima que cumple con los tiempos requeridos**

Flujo de Diseño RTL



Optimización 2 Niveles

- **Eficiente y madura**
- **Fundamentos teóricos para la optimización lógica multinivel**
- **Usada directamente para PLA y PLD**
- **Usada como subrutina en optimización multinivel**
- **“Logic Minimization Algorithms for VLSi Synthesis”, Robert King Brayton, Alberto L. Sangiovanni-Vincentelli, Curtis T. McMullen, Gary D. Hachtel, Agosto 1984**

Nueva Metodología

- **Divide la optimización lógica en dos problemas**
  - **Optimización independiente de la tecnología**
    - **Determina la estructura lógica general**
    - **Estima costos independientes de la tecnología**
  - **Optimización dependiente de la tecnología**
    - **Mapea en puertas de la biblioteca**

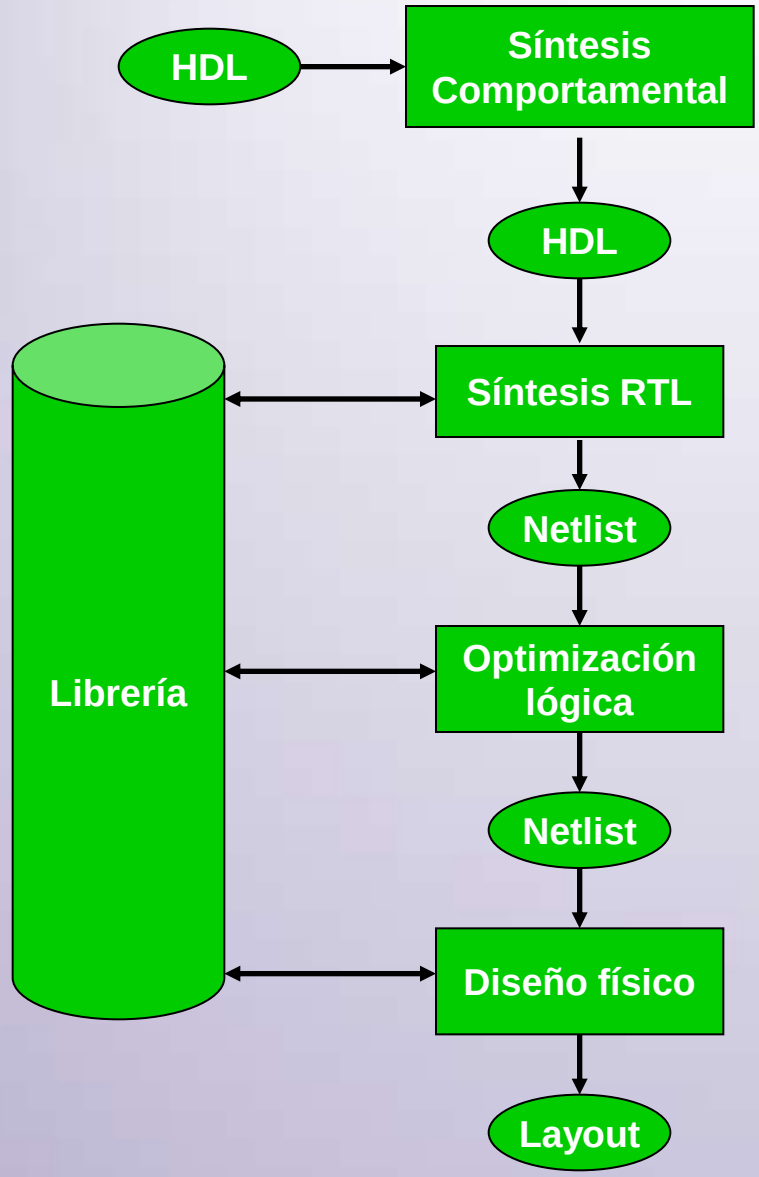
Optimización Independiente de la Tecnología

- Minimiza las funciones lógicas (2 niveles)
- Busca subexpresiones comunes
- Sustituye una expresión dentro de la otra
- Factoriza funciones simples
- `f=ac+ad+bc+bd+a!e` (suma de productos)

  `=(a+b)(c+d)+a!e` (forma factorizada)

Técnicas de Optimización

- Independientes
  - Two-level minimization
  - Selective collapsing
  - Algebraic decomposition
  - Restructuring for timing
  - Redundancy removal
  - Transduction
  - Global-flow
- Dependientes
  - Tree covering
  - Load buffering
  - Rule-based mapping
  - Signature analysis
  - Inverter phase assignment
  - Discrete sizing



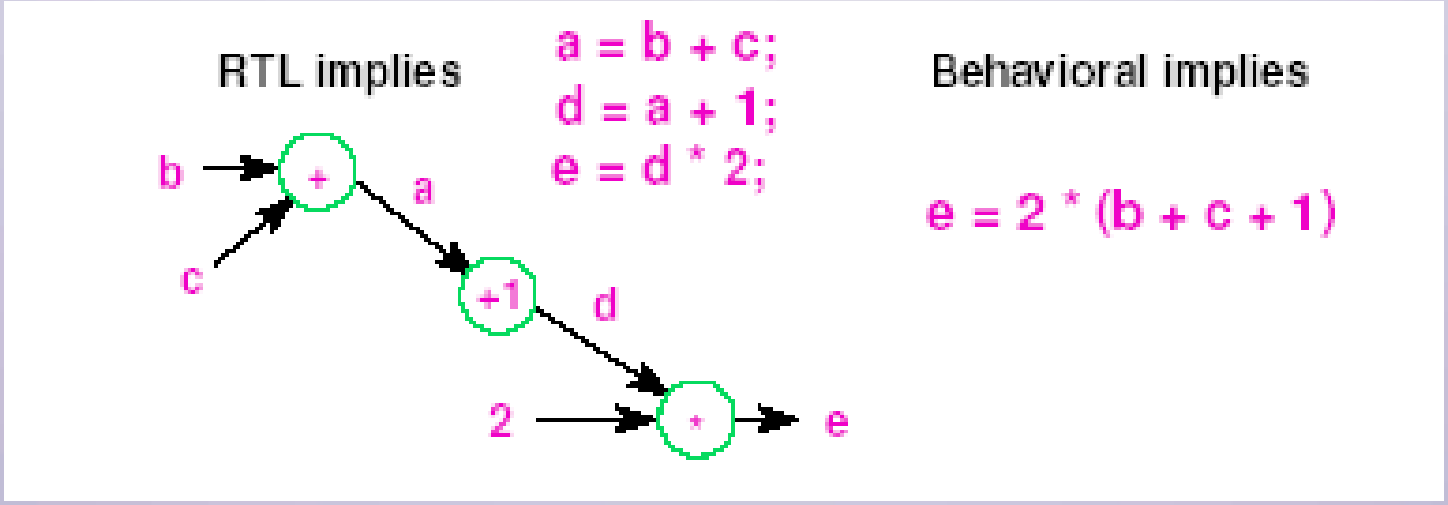
Síntesis Comportamental

Nivel Comportamental

- **Una descripción comportamental es siempre funcional**
- **Relaciones temporales son expresadas como precedencias**
- **Una micro arquitectura completa es sintetizada a partir de una descripción comportamental**

Elementos Claves

- **Asignación automática de recursos**
- **Ordenamiento cronológico (scheduling)**



Características Sintesis Comportamental

- **Ordenamiento de operaciones (scheduling)**
- **Inferencia de memoria**
- **Asignación de recursos**
- **Uso de componentes pipeline**
- **Lazos de pipeline**
- **Generación automática de autómatas de estado finito para control**

Beneficios Diseño Comportamental

- **Abstracción**
  - **Especifica funcionalidad en vez de implementación**
  - **Simulación rápida**
  - **Diseño a nivel sistema**
  - **Mejor calidad de resultado**
  - **Generación automática de FSM**

Estado del Arte

- **Síntesis RTL madura y usada para diseño de chips**
- **Síntesis comportamental menos madura**
  - **Usada originalmente en diseño de DSP**
  - **Creciente uso en video, networking, y diseño ASIC**
  - **No ha crecido lo suficiente para desplazar síntesis RTL**

Síntesis de Sistemas

Diseños son heterogéneos y atraviesan los dominios del control y flujo de datos en forma arbitraria

Diseños deben ser modelados en lenguajes estándares y gráficos con consistencia entre dominios y niveles de abstracción

SW
Integrado antes en el proceso de diseño
Evaluación rápida de partición HW/SW
Reuso de código debe ser considerada

HW
Diseño de altos niveles de abstracción
Reuso debe ser considerado a altos niveles de abstracción
Necesita mezclar C, C++, Verilog y VHDL

Estado de la Sintesis de Sistemas

- **Ha fallado aun más que la síntesis comportamental**
  - **Más inversión que para comportamental**
  - **Menos retorno que comportamental**
- **Problemas**
  - **Cual es el lenguaje de diseño?**
  - **Partición HW/SW**
  - **Generación automática de HW/SW a partir de la descripción**