

# Introducción al diseño en sistemas embebidos

## Lección 5

Prof.Ing. Jeferson González G.

CE-5303 Sistemas Embebidos

*Área de Ingeniería en Computadores*

*Instituto Tecnológico de Costa Rica*

- 1 Introducción
- 2 Niveles de abstracción
- 3 Nivel de Procesador
- 4 Nivel de Sistema

# Diseño

El diseño moderno de sistemas embebidos tiene características particulares:

- Sistemas complejos.
- Se cuenta con herramientas para diseño asistido.
- Sigue metodología de co-diseño (Hw y Sw se diseñan simultáneamente).
- Se diseña en diferentes niveles de abstracción con diferentes modelos.

# Niveles de abstracción

Actualmente, el diseño de sistemas embebidos se realiza en diferentes **niveles de abstracción**, cada uno con características, herramientas y metodologías diferentes.

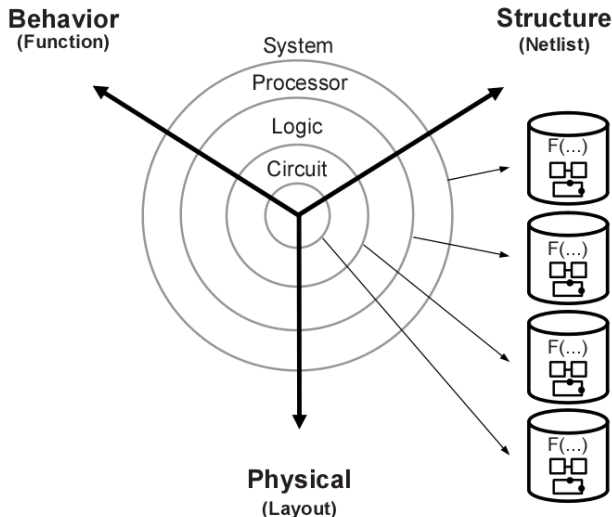
# Y-Chart

Se utiliza para explicar las relaciones en diferentes niveles de abstracción, herramientas y metodologías de diseño en sistemas embebidos.

- Cada diseño puede ser modelado de tres formas básicas:
  - **Comportamiento:** funcionalidad, especificación.
  - **Estructura:** netlist, diagrama de bloques.
  - **Diseño físico:** Layout, diseño de tarjeta
- Identifica 4 niveles de abstracción:
  - Sistema
  - Procesador
  - Lógica
  - Circuito

## Y-Chart

## Y-Chart



# Y-Chart - Comportamiento

Diseño es descrito como **caja negra**.

- Las salidas se describen en función de las entradas en el tiempo.
- La especificación no indica cómo construir la caja negra

# Y-Chart - Estructura

Especifica el diseño como un conjunto de **componentes** y **conexiones**.

- El comportamiento de la caja negra viene dado por sus componentes y la interacción de los mismos.
- Los detalles de implementación de cada componente no se muestran en este nivel.



# Y-Chart - Diseño Físico

El diseño físico agrega **dimensionalidad** a la estructura.

- Especifica el tamaño de cada componente, puerto, conexión, chip, PCB, etc.
- Muestra a mayor detalle la funcionalidad de los componentes.

# Y-Chart - Niveles de abstracción

Cada nivel se nombra por el tipo de componente que se genera:

- **Circuitos:** Se generan celdas estándar, conformadas por transistores (NMOS/PMOS)
- **Lógica:** Se usan compuertas lógicas y flip flops para generar RTL (multiplicadores, registros, ALU's, FSMs)
- **Procesador:** Se generan procesadores estándar o a la medida así como hardware especializado (controladores de memoria, arbitros de bus, interfaces, NoC, etc)
- **Sistema:** Se generan sistemas embebidos con procesadores, memorias, buses, coprocesadores, etc.

# Síntesis

Cada nivel de abstracción requiere componentes que serán usados en la estructura del diseño.

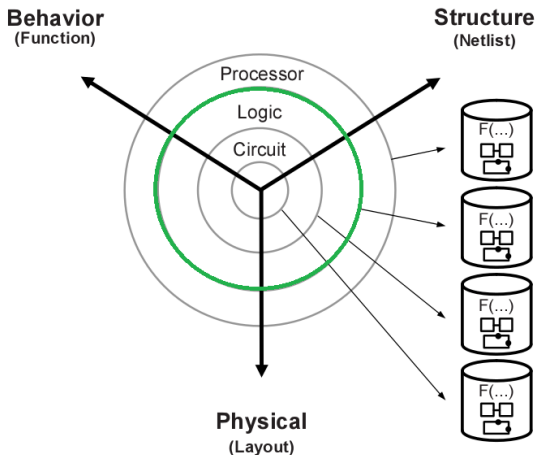
El proceso de pasar de comportamiento a estructura en cada nivel de abstracción se conoce como **síntesis**.

La síntesis permite pasar de un nivel de abstracción mayor a uno menor (p.e RTL – > circuitos)

- Cada componente puede tener un modelo diferente en cada eje del Y-Chart.

# Nivel: Procesador

## Diseño a nivel de procesador



# Procesador: Diseño de modelo de comportamiento

En el nivel de procesador, cada elemento diseñado será un elemento de procesamiento (**PE**).

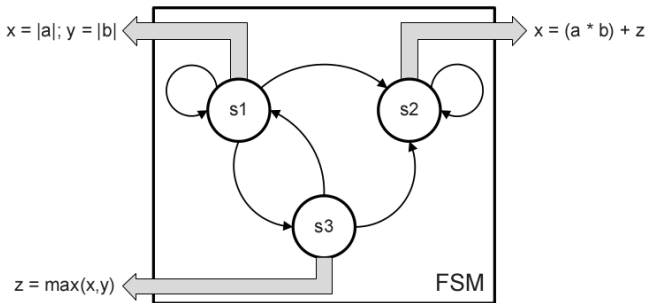
Un PE ejecuta funciones específicas (a la medida) o estándar (procesadores comunes) que ejecutan instrucciones en algún lenguaje de programación.

Pueden ser descritos de varias formas:

- Máquinas de estados finitos (FSM)
- Máquinas de estados finitos con datos (FSMD)
- Grafo de flujo de control y datos (CDFG)
- Flujo de set de instrucción (ISF)

# Comportamiento: FSM

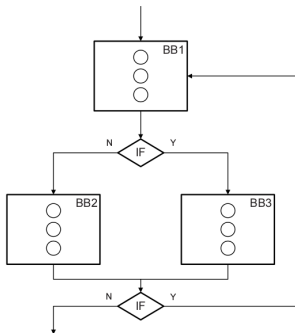
Estados representan operaciones aritméticas o expresiones de programación.



- No es adecuado para representar algoritmos expresados en lenguajes de programación estándar (C)

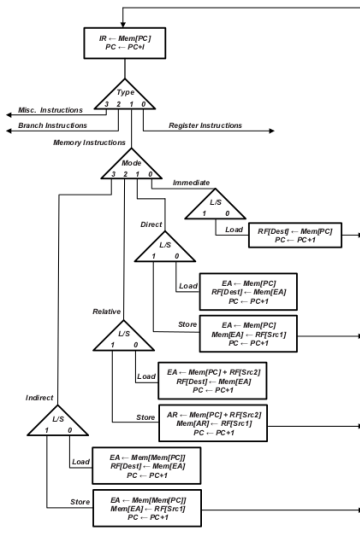
# Comportamiento: CDFG

Compuesto por bloques básicos (BB) y condiciones.



- Natural para lenguajes de programación secuenciales.
- Muestra dependencias de control entre BB.
- Puede convertirse en FSMd con *super-estados*.

# Comportamiento: IS flow





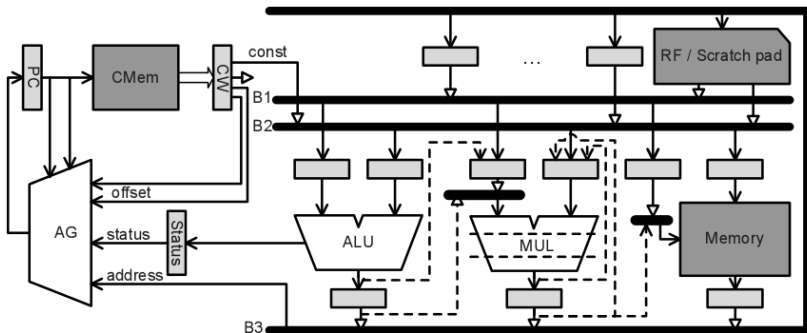
# Modelo estructural

La implementación de un elemento de procesamiento se realiza por medio de componentes RTL. La estructura interna consiste de:

- Datapath: registros, memoria, unidades funcionales, buses.
- Controlador (FSM, combinacional)

La tarea de selección de componentes y estructura de un PE y la definición de operaciones de RTL en cada ciclo de reloj corresponde a la síntesis.

# Modelo estructural de un procesador



# Síntesis a nivel de procesador

Proceso de generación de la estructura del procesador a partir de la descripción de comportamiento.

La síntesis a nivel de procesador toma como entrada el set de instrucciones (IS) del procesador.

- IS es ingresado manualmente -> alto desempeño.
- Diferentes etapas:
  - Selección (Allocation)
  - Calendarización (Scheduling)
  - Asignación (Binding)
  - Síntesis
  - Refinamiento

# Allocation

Corresponde a la selección de componentes de un biblioteca RTL.

- Seleccionar al menos un componente por operación del modelo de comportamiento.
- Biblioteca debe incluir características de componentes y métricas.

# Scheduling

Se debe calendarizar las operaciones del modo del comportamiento en ciclos.

- Lectura de operandos de almacenamiento (BR) - CC1
- Ejecutar operación - CC2
- Escritura de operandos - CC5

Cada bloque básico debe calendarizarse en varios ciclos de reloj y algunas operaciones podrán calendarizarse en el mismo ciclo de reloj.

# Binding

Asignación de variables, operaciones y transferencias.

- Cada variable se asigna a una unidad de almacenamiento.  
Múltiples variables pueden ser asignadas a la misma unidad de almacenamiento si no se traslapan en el tiempo.
- Las operaciones deben ser asignadas a unidades funcionales capaces de ejecutarlas.
- Debe existir conexión entre unidades funcionales y elementos de almacenamiento.

# Síntesis de controlador

Generación del controlador por compuestas lógicas o controles de funciones, a partir de pasos anteriores.

- El controlador puede tener memoria de programa estática o programable.

# Refinamiento

Cada procesador puede generarse con diferentes estilos de asignación de variables, operaciones y transferencias.

(1) without any binding:

$$a = b + c;$$

(2) with storage binding of  $a$  to RF(1),  $b$  to RF(3), and  $c$  to RF(4):

$$\text{RF}(1) = \text{RF}(3) + \text{RF}(4);$$

(3) with storage and functional unit binding with  $+$  bound to ALU1:

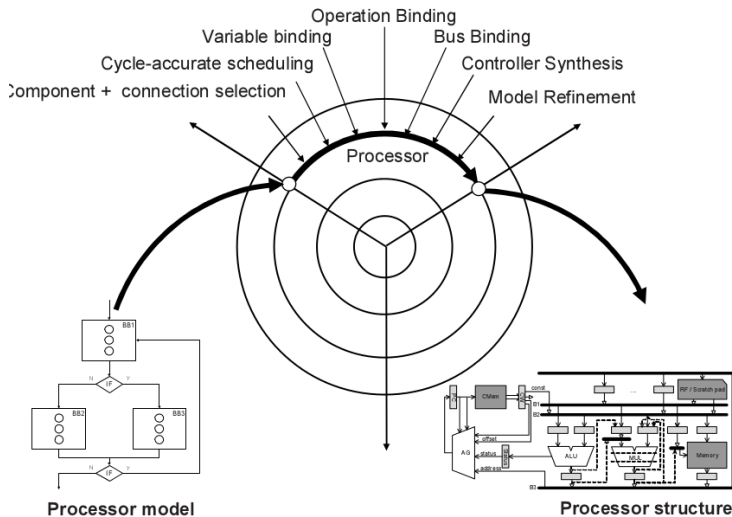
$$\text{RF}(1) = \text{ALU1}(+, \text{RF}(3), \text{RF}(4));$$

(4) or with storage, functional unit, and connectivity binding:

$$\begin{aligned} \text{Bus1} &= \text{RF}(3); \text{Bus2} = \text{RF}(4); \text{Bus3} = \text{ALU1} \\ (+, \text{Bus1}, \text{Bus2}); \text{RF}(1) &= \text{Bus3}; \end{aligned}$$

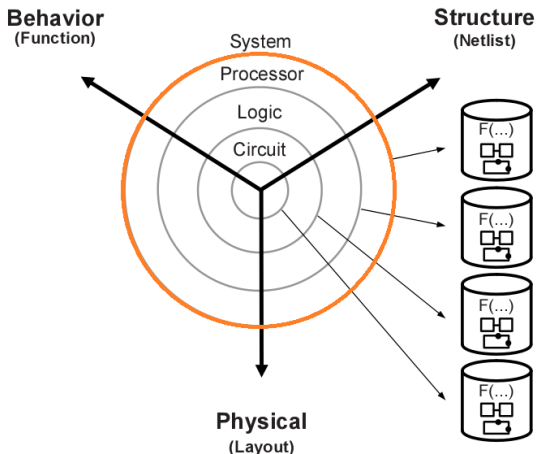


# Síntesis en nivel de procesador



# Nivel: Sistema

## Diseño a nivel de sistema



# Sistema: Diseño de modelo de comportamiento

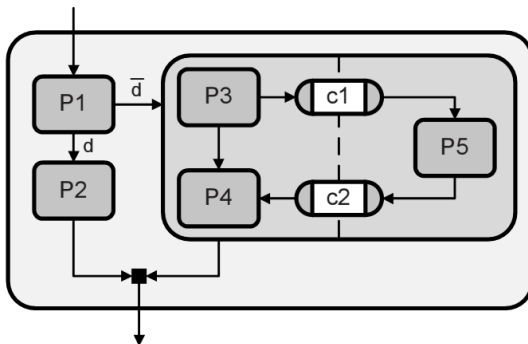
Los modelos de comportamiento, como CDFG, funcionan bien para un procesador pero no para un sistema completo.

Un modelo de sistema debe contemplar:

- Múltiples procesos (en uno o varios PE)
- Interacción SW y HW
- Contempla concurrencia y segmentación
- Sincronización y canales
- Modelo Común: Máquina de estados de procesos (PSM)

# Máquina de estados de procesos - PSM

Máquina de estados que relaciona procesos secuenciales y paralelos por señales y canales de comunicación.

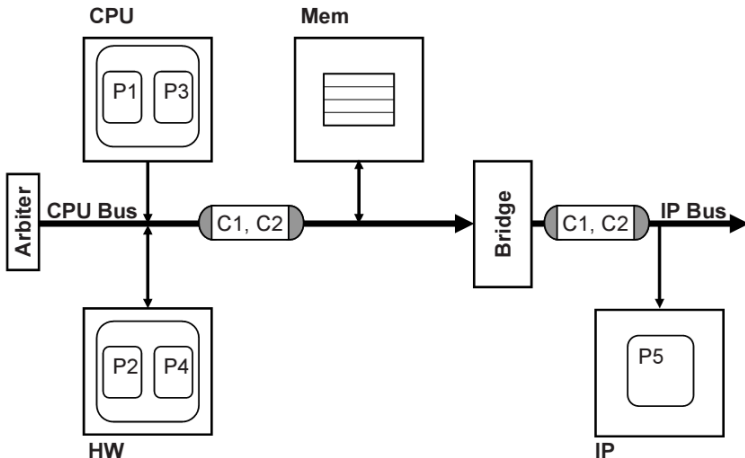


# Modelo estructural

Es un diagrama de bloques de los componentes del sistema, generado a partir de la descripción de comportamiento.

- Elementos de procesamiento (PE)
- Componentes de almacenamiento (M)
- Elementos de comunicación (CE)
- Interfaces (IF)
- Bloques de propiedad intelectual (IP cores)

# Diagrama de bloques de sistema



# Síntesis

Proceso de generación de diagrama de bloques (estructura) del sistema a partir de la especificación (modelo de comportamiento).

Realiza las tareas:

- Perfilado y estimación
- Selección de componentes y conexiones (allocation)
- Asignación de procesos y canales (binding)
- Calendarización de procesos
- Inserción de interfaces
- Refinamiento del modelo

# Perfilado y estimación

Perfilado de código.

- Tipo y frecuencia de operaciones
- Transferencias de bus
- Llamadas a funciones
- Accesos a memoria

Estimación de métricas:

- Desempeño
- Costo
- Consumo de potencia
- Ancho de banda
- Tolerancia a fallos



# Allocation

Selección de componentes de bibliotecas de: procesadores estándar, memorias, IP, buses, interfaces, etc.

- Creación de plataforma
- Selección en plataforma definida.

# Binding

- Asignación de procesos a elementos de procesamiento :  
 $P \rightarrow PE$
- Asignación de canales a buses o NoC:  $C \rightarrow B$
- Asignación de variables a elementos de memoria:  $v \rightarrow M$

# Scheduling

Procesos paralelos corriendo en el mismo PE deben ser calendarizados (estática o dinámicamente).

- Requiere la generación de un sistema operativo de tiempo real para calendarización dinámica.

# Inserción de interfaz

Los módulos de interfaz (hardware) deben seleccionarse de la biblioteca. Puede que requieran agregarse algunos módulos de interfaz de software (firmware) como:

- Drivers de dispositivos
- Tablas de routing
- Rutinas de interrupción
- Mensajes (IPC)

En hardware:

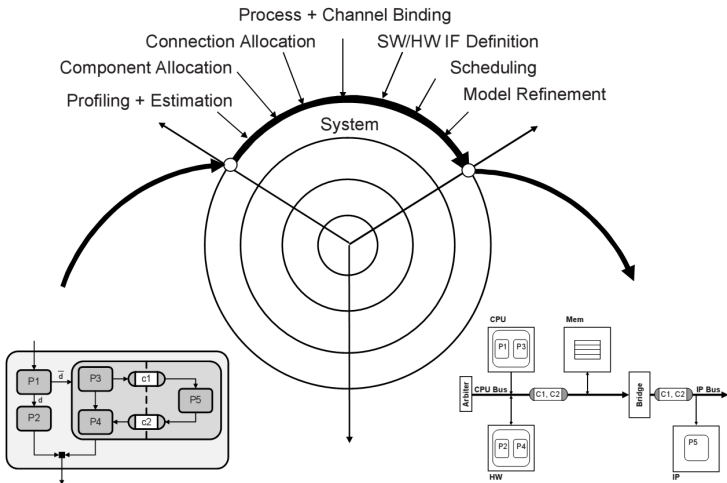
- Controladores de interrupción
- Controladores de memoria

# Refinamiento

Síntesis suele ser una tarea iterativa.

Luego de la síntesis deben medirse variables (simulación) y refinar en los diferentes pasos para lograr los resultados esperados.

# Síntesis en nivel de sistema





Gajski, D.D., Abdi, S., Gerstlauer, A., Schirner, G (2009)  
Embedded System Design - Modeling, Synthesis and  
Verification