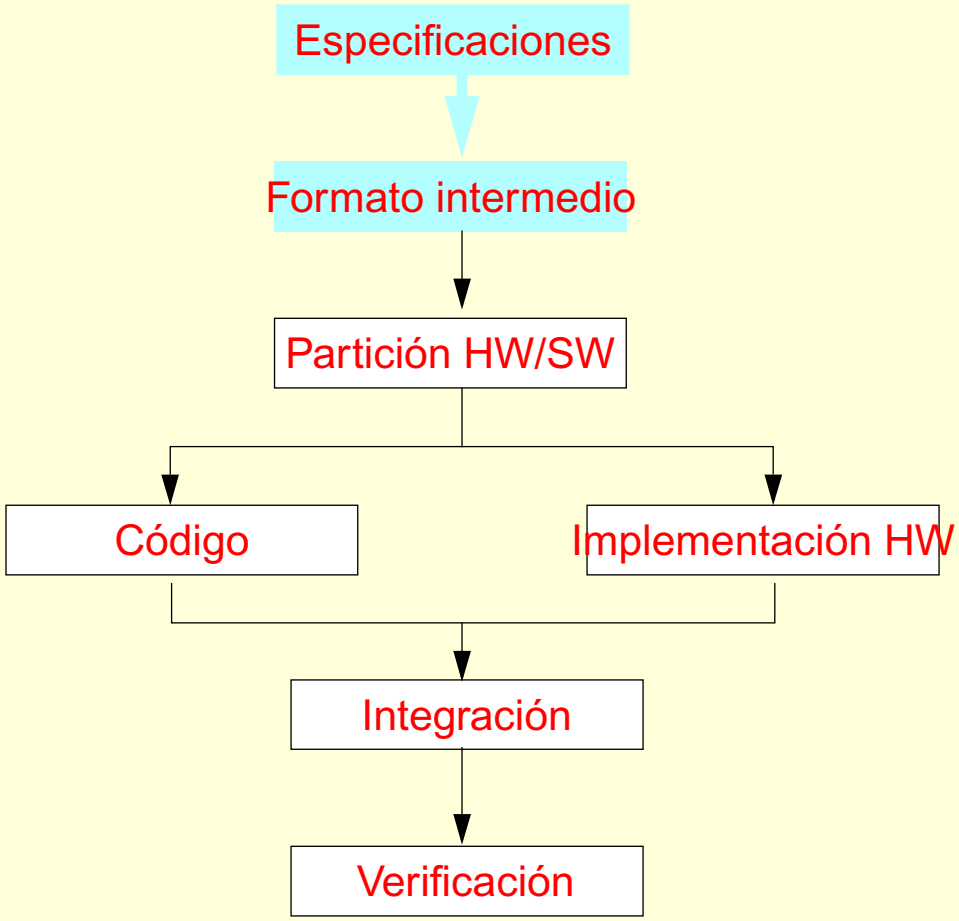
TEMA II: MODELOS DE ESPECIFICACIÓN Y LENGUAJES



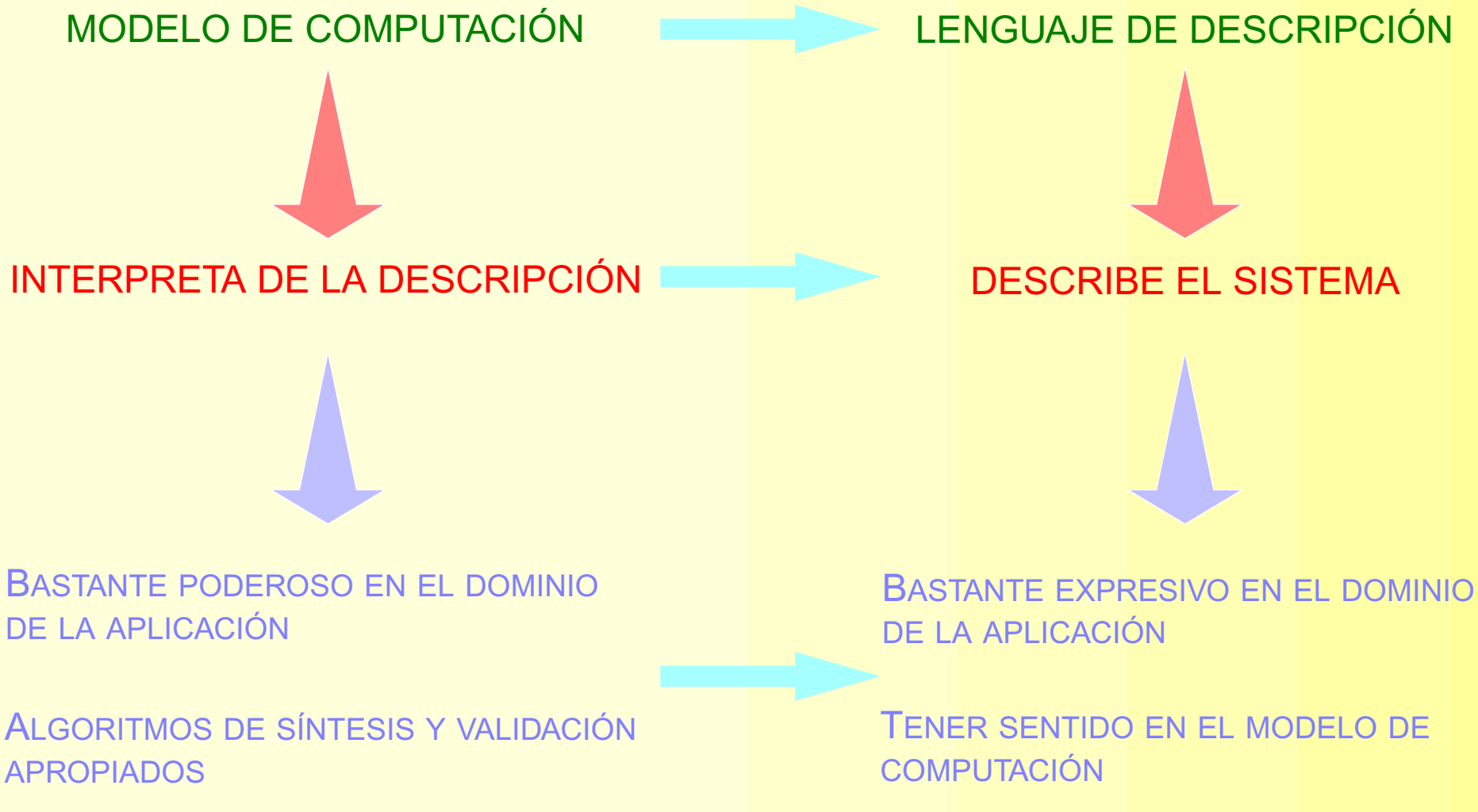
UN MODELO DE COMPUTACIÓN ES UN ENTORNO DE TRABAJO EN EL QUE EXPRESAR LA SECUENCIA DE ACCIONES QUE DEBEN SER EJECUTADAS PARA COMPLETAR UNA OPERACIÓN

MODELO DE COMPUTACIÓN

INTERPRETA DE LA DESCRIPCIÓN

BASTANTE PODEROSO EN EL DOMINIO DE LA APLICACIÓN

ALGORITMOS DE SÍNTESIS Y VALIDACIÓN APROPIADOS


MODELO DE COMPUTACIÓN
LENGUAJE DE DESCRIPCIÓN
INTERPRETA DE LA DESCRIPCIÓN
DESCRIBE EL SISTEMA
BASTANTE PODEROSO EN EL DOMINIO DE LA APLICACIÓN
ALGORITMOS DE SÍNTESIS Y VALIDACIÓN APROPIADOS
BASTANTE EXPRESIVO EN EL DOMINIO DE LA APLICACIÓN
TENER SENTIDO EN EL MODELO DE COMPUTACIÓN

Señales

Llegada en bloques regulares

Independencia de datos

Llegada en tiempos diferentes

	Señales de dato	Señales de control
Especificación	Dependencia funcional entre entradas y salidas	Relación causa/efecto
Síntesis	Eficiencia memoria/tiempo (pipeline)	Tiempo de respuesta
Validación	Todos los procesos y eventos son iguales	Prioridad entre eventos y procesos.

Propiedades

Afirmación del comportamiento

Concurrencia

Ejecución simultánea de varias tareas

Transiciones de estado

División del comportamiento en estados

Jerarquía

División del sistema en componentes más pequeños

Construcciones de programación

Operaciones implementadas vía software

Acabado

Cada componente acaba su operación

Comunicación

Desincronizada

Lectura-modificación-escritura

FIFO ilimitada

FIFO limitada

Rendezvous

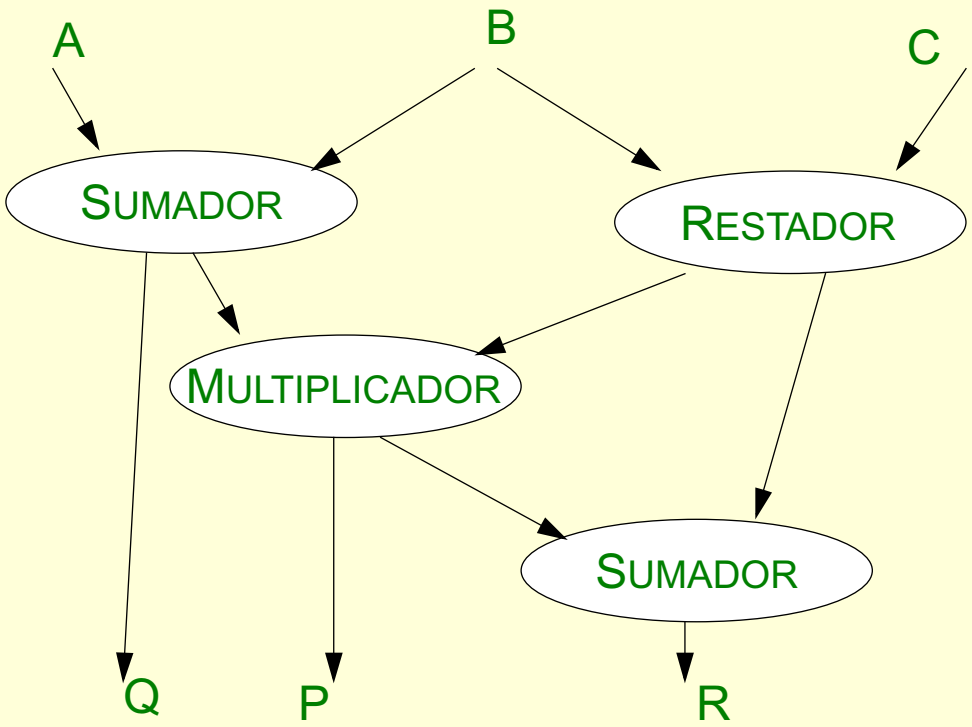
Sincronización

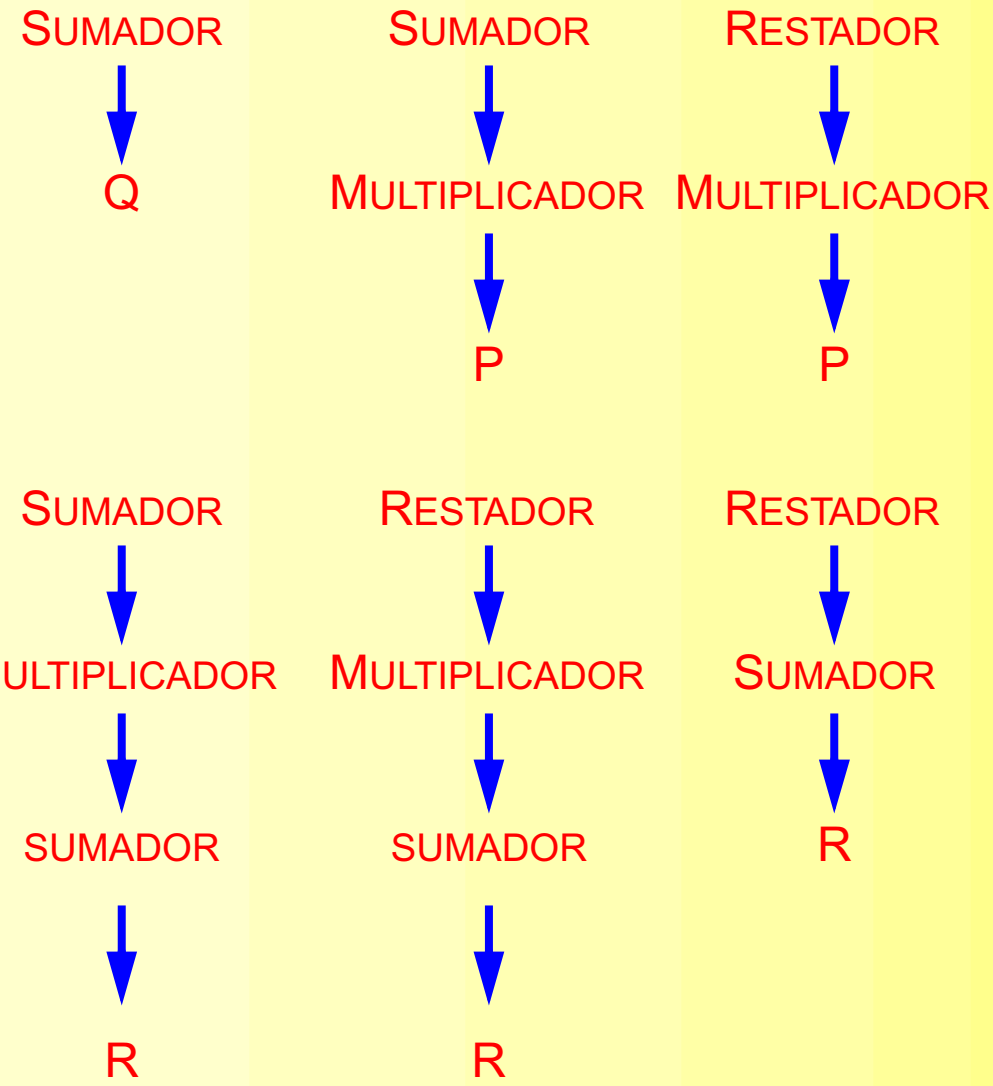
Muy relacionada con la comunicación

Temporización

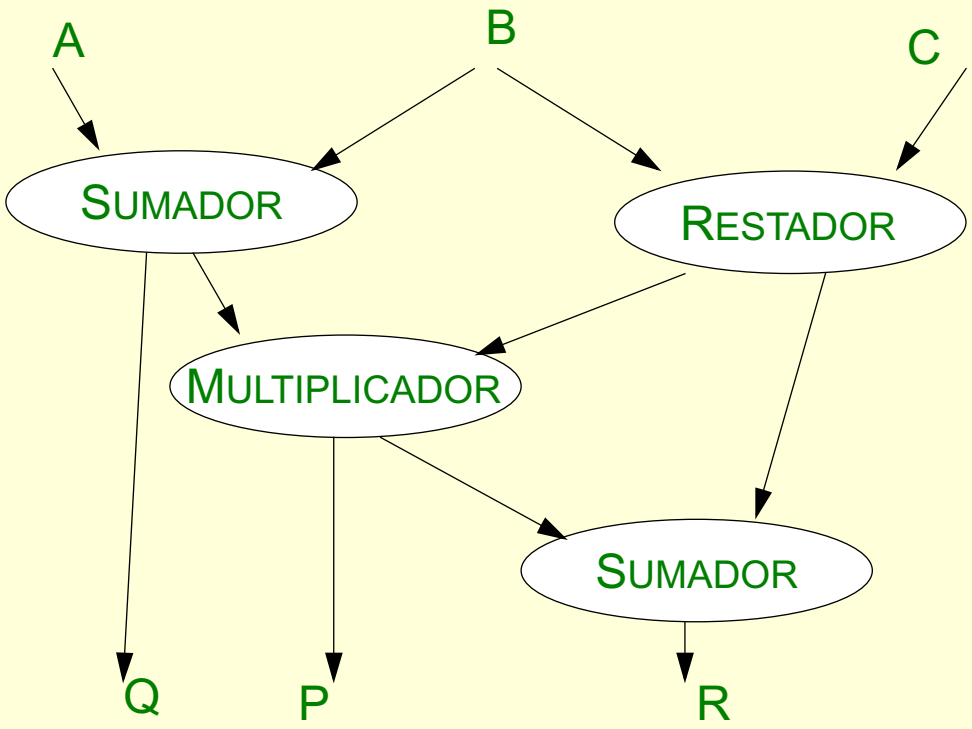
Requerida para la implementación en el mundo real

Ejemplo de Concurrencia





Ejemplo de Transiciones de estado y Jerarquía



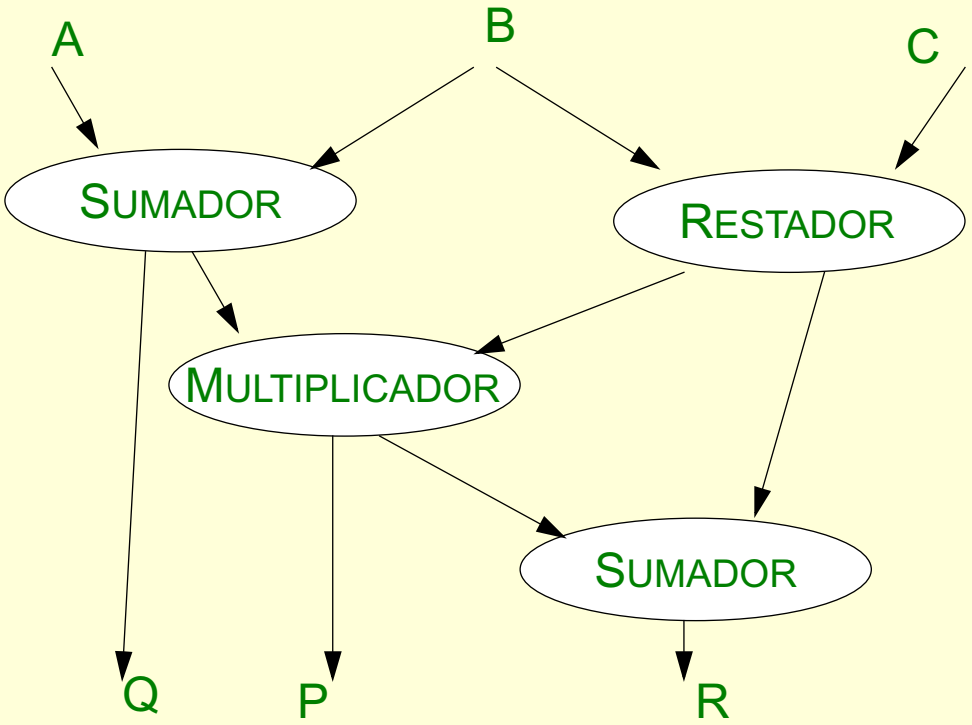
Estados

Sumador1
Multiplicador
Restador
Sumador2

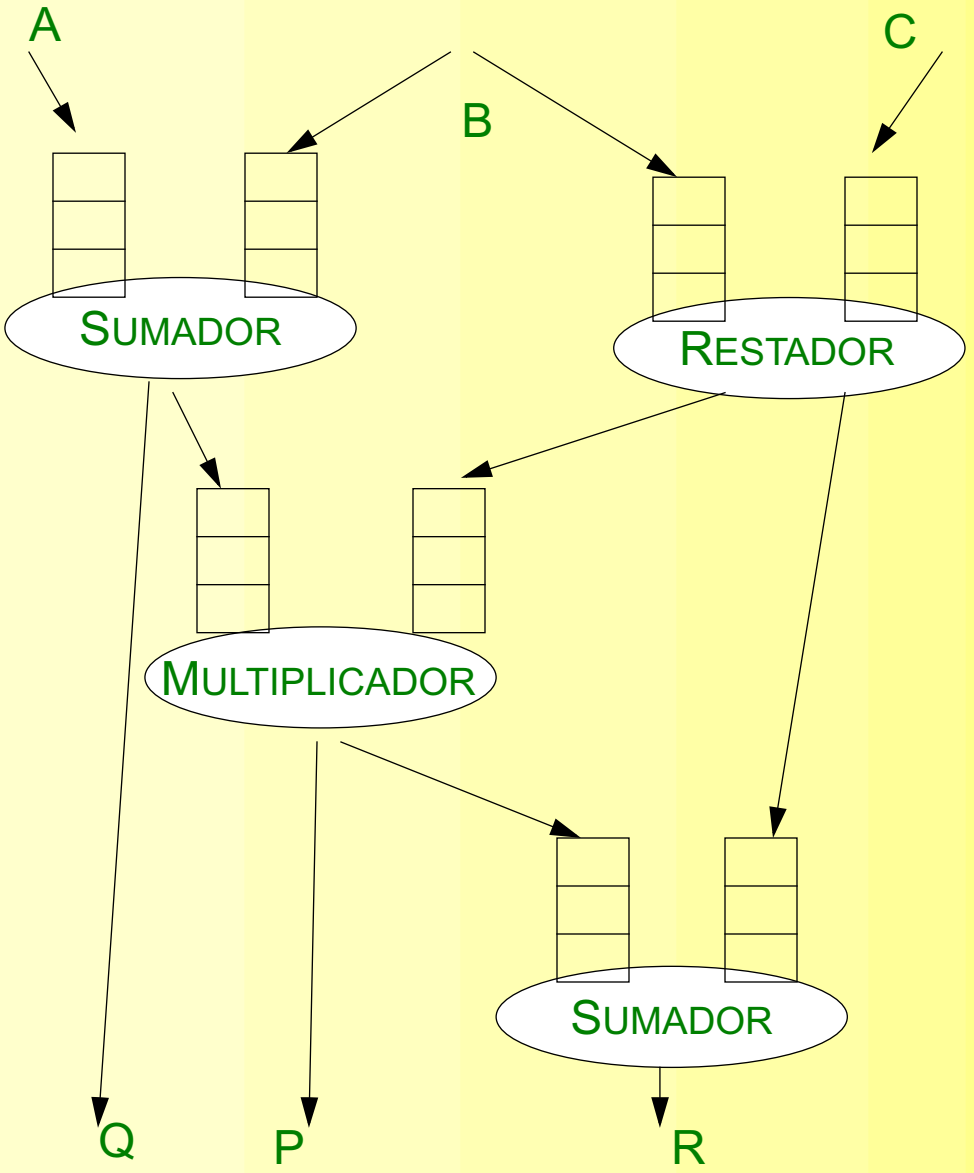
Componentes

Sumador
Multiplicador
Restador

Ejemplo de Comunicación



RENDEZVOUS



FIFO

MODELO TAGGED-SIGNAL

TAG = MARCA TEMPORAL DE UN SISTEMA FÍSICO

EVENTO = PAR VALOR/TAG DENTRO DE UN CONJUNTO DE VALORES Y TAGES

SEÑAL = CONJUNTO DE EVENTOS

$$\forall \text{evento} \rightarrow e_1 = (t, v_1), e_2 = (t, v_2) \Rightarrow v_1 = v_2$$

PROCESO = CONJUNTO DE POSIBLES COMPORTAMIENTOS O UNA RELACIÓN ENTRE SEÑALES DE ENTRADA Y SALIDA

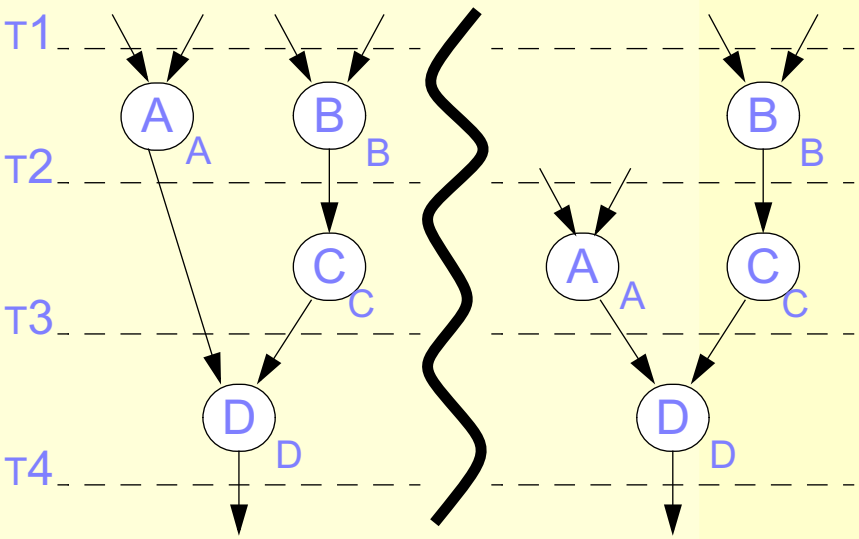
SISTEMA TEMPORIZADO => TODOS LOS TAGS ORDENADOS

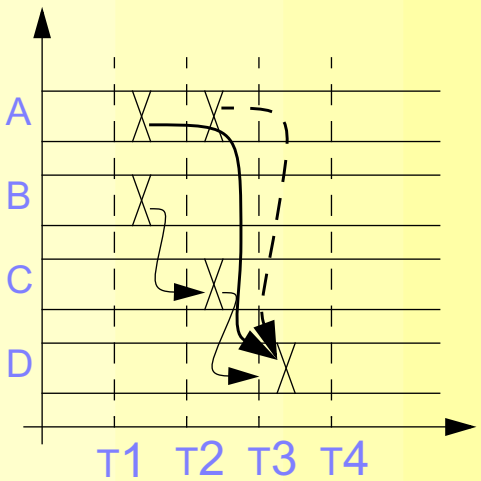
SISTEMA NO TEMPORIZADO => NO TODOS TAGS ORDENADOS

DOS EVENTOS SÍNCRONOS => TIENEN EL MISMO TAG

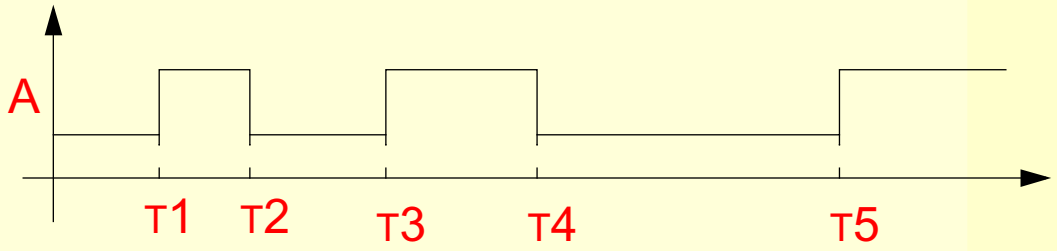
DOS SEÑALES SÍNCRONAS => TIENEN LOS MISMOS TAGS

SISTEMA SÍNCRONO => TODAS LAS SEÑALES SÍNCRONAS



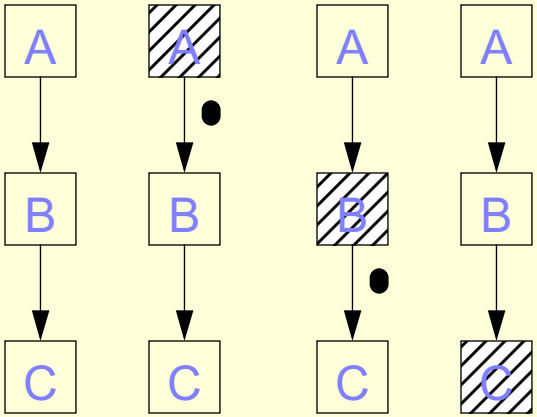


MODELO DE EVENTOS DISCRETOS (DE)

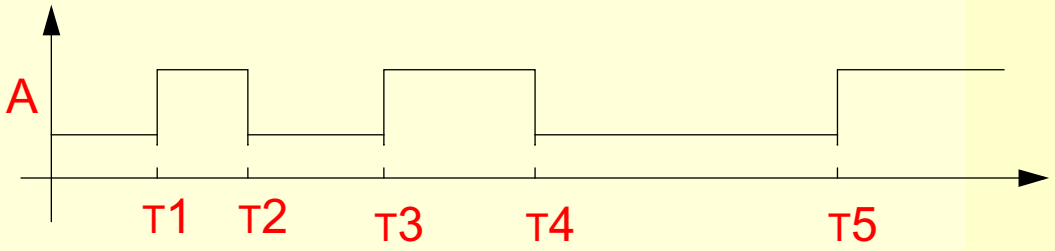


A = {(0,0), (1,τ1), (0,τ2), (1,τ3), (0,τ4), (1, τ5)}

PUEDE EXISTIR UN CAMBIO EN EL SISTEMA CUANDO CAMBIA ALGUNA DE SUS SEÑALES

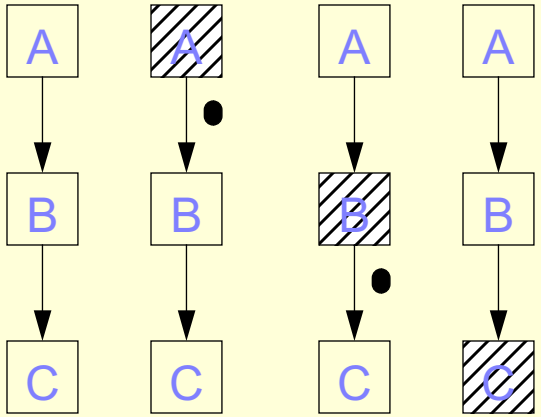


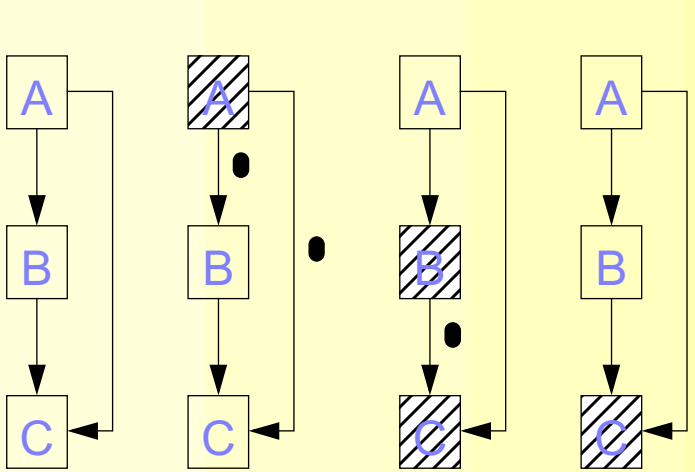
MODELO DE EVENTOS DISCRETOS (DE)

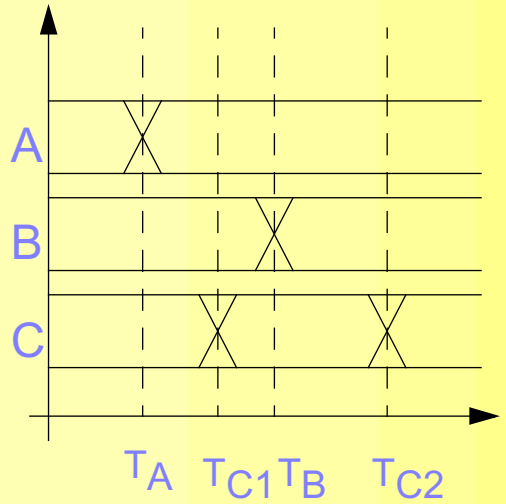


A = {(0,0), (1,τ1), (0,τ2), (1,τ3), (0,τ4), (1, τ5)}

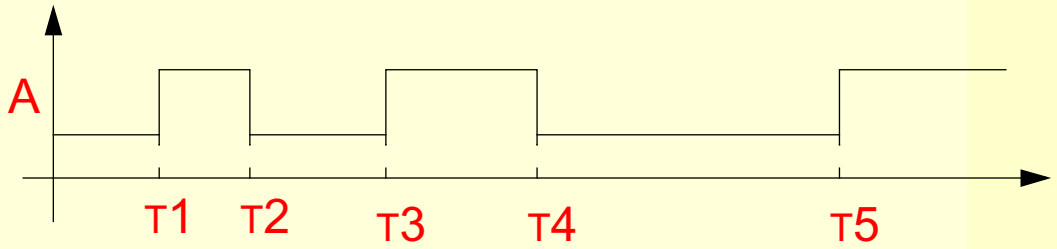
PUEDE EXISTIR UN CAMBIO EN EL SISTEMA CUANDO CAMBIA ALGUNA DE SUS SEÑALES





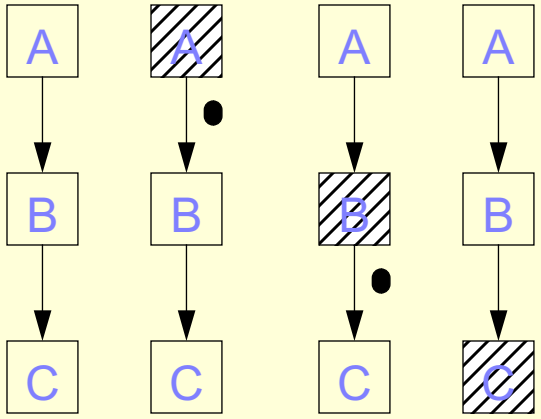


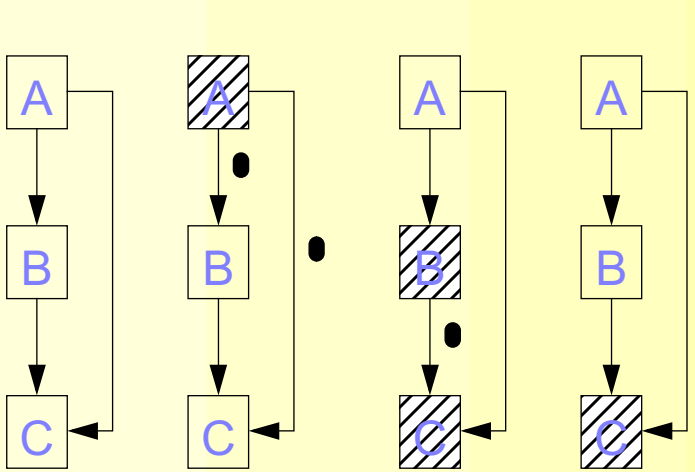
MODELO DE EVENTOS DISCRETOS (DE)

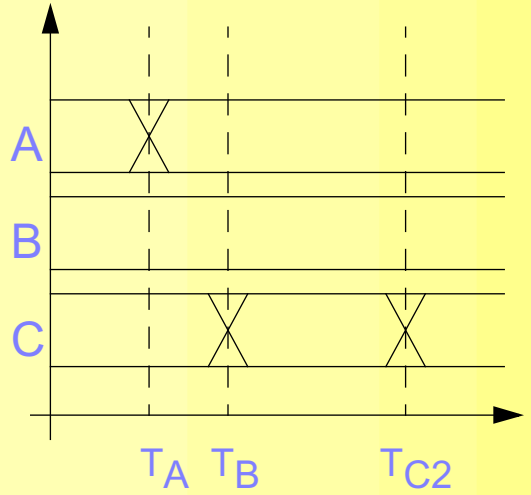


A = {(0,0), (1,τ1), (0,τ2), (1,τ3), (0,τ4), (1, τ5)}

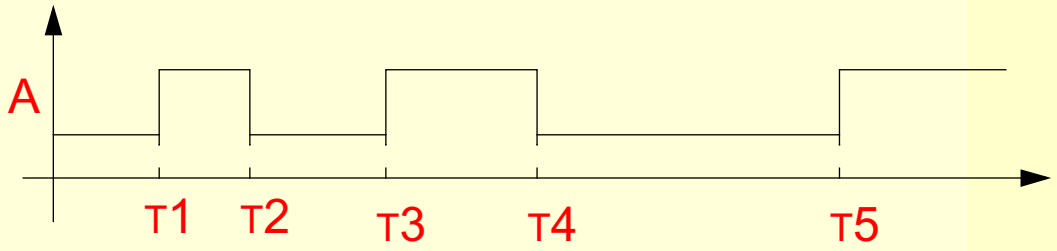
PUEDE EXISTIR UN CAMBIO EN EL SISTEMA CUANDO CAMBIA ALGUNA DE SUS SEÑALES





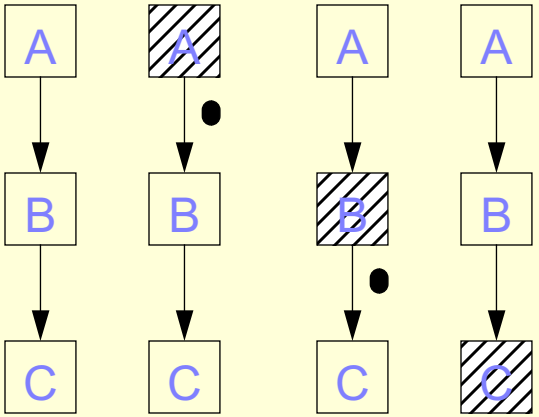


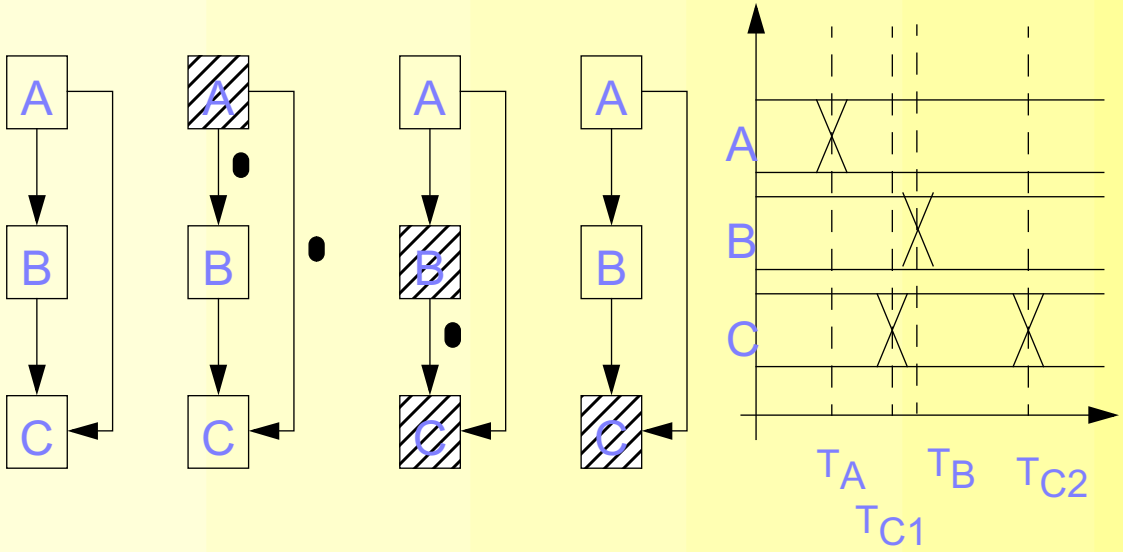
MODELO DE EVENTOS DISCRETOS (DE)



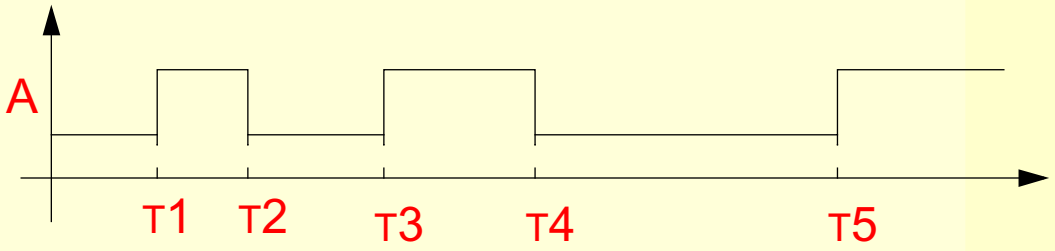
A = {(0,0), (1,τ1), (0,τ2), (1,τ3), (0,τ4), (1, τ5)}

PUEDE EXISTIR UN CAMBIO EN EL SISTEMA CUANDO CAMBIA ALGUNA DE SUS SEÑALES



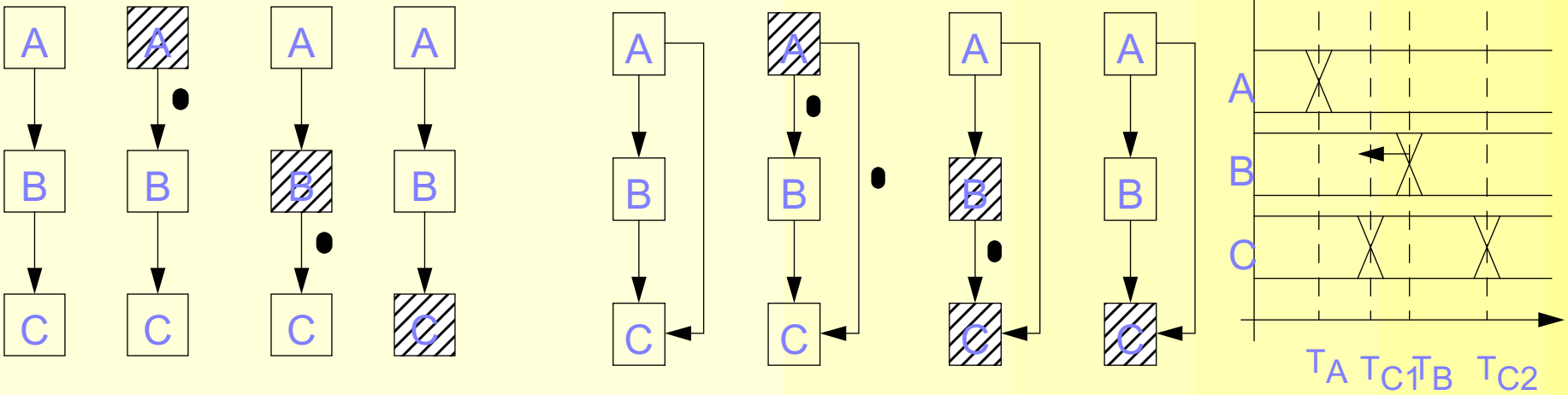


MODELO DE EVENTOS DISCRETOS (DE)



A = {(0,0), (1,τ1), (0,τ2), (1,τ3), (0,τ4), (1, τ5)}

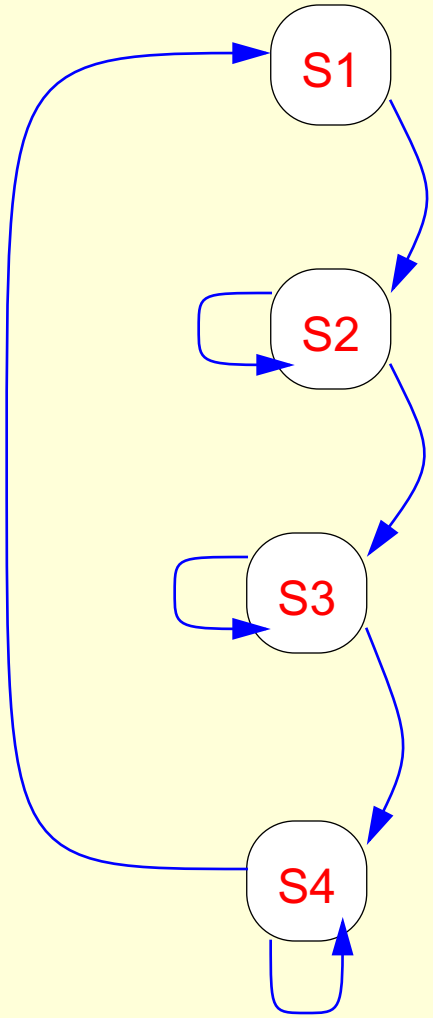
PUEDE EXISTIR UN CAMBIO EN EL SISTEMA CUANDO CAMBIA ALGUNA DE SUS SEÑALES



MODELO UTILIZADO EN SIMULADORES DE VHDL, VERILOG Y PTOLEMY

Modelo de máquinas de estado finito (FSM)

(I, O, S, δ, λ)



Buena para especificar comportamiento secuencial

No forman un modelo completo (Turing)

Dominios típicos: control intensivo y protocolos

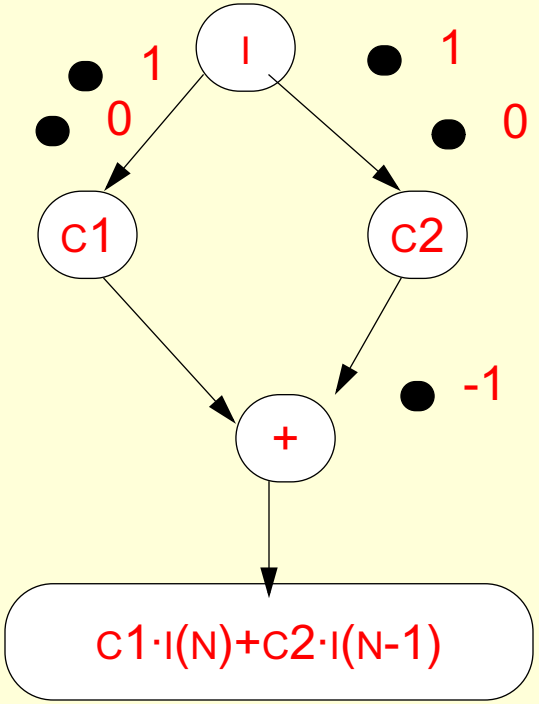
Problema con un grandes necesidades de memoria

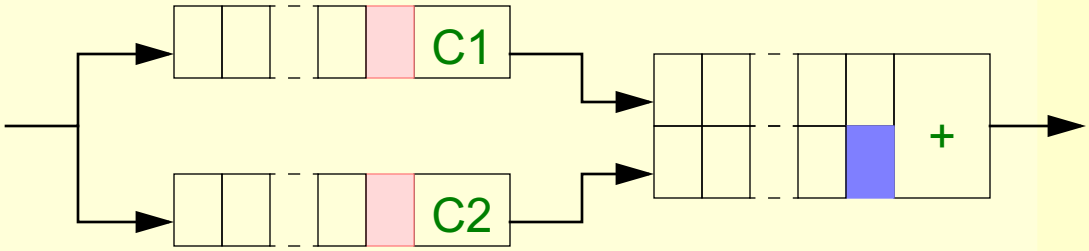
Jerarquía

Concurrencia

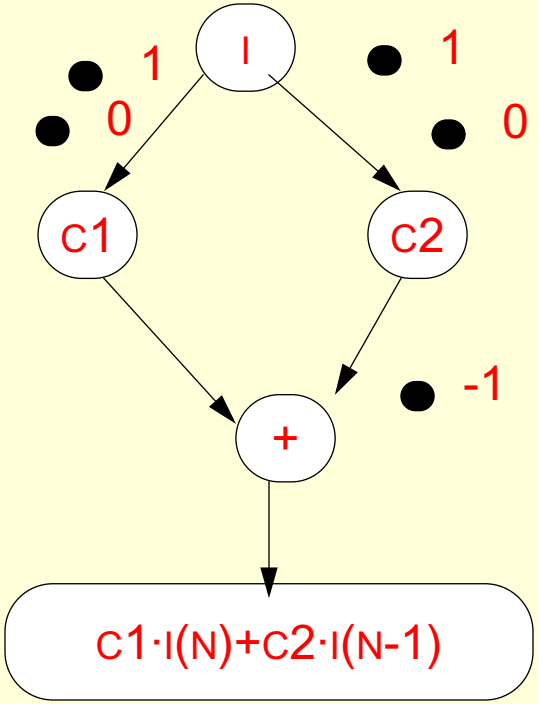
No determinismos

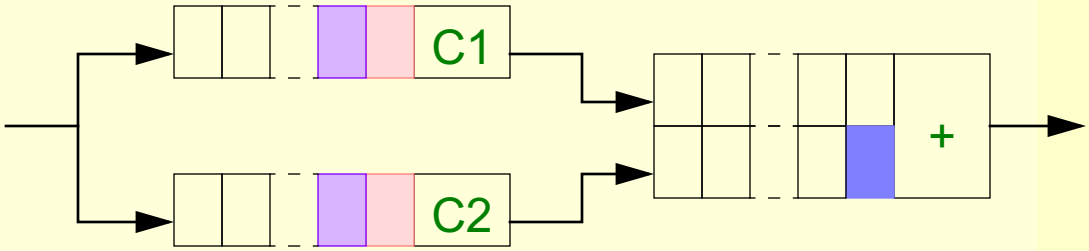
Modelo de Redes de Proceso de Flujo de datos (DF)



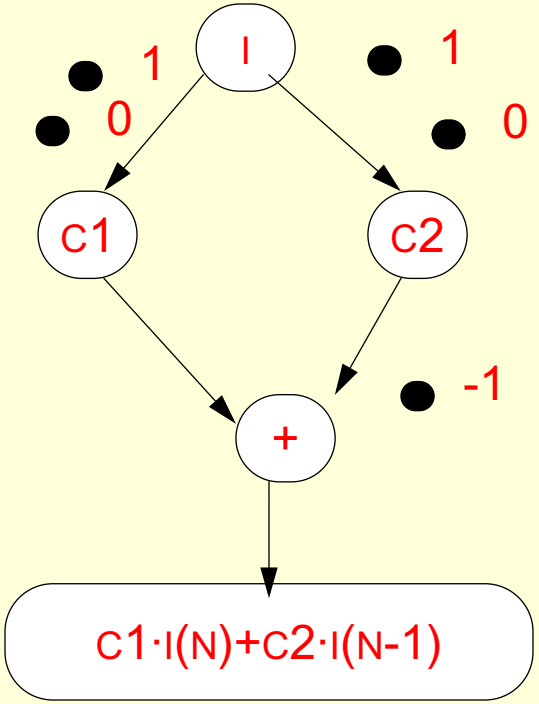


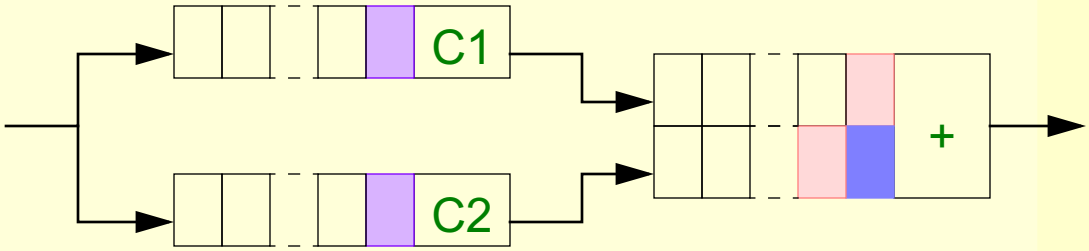
Modelo de Redes de Proceso de Flujo de datos (DF)



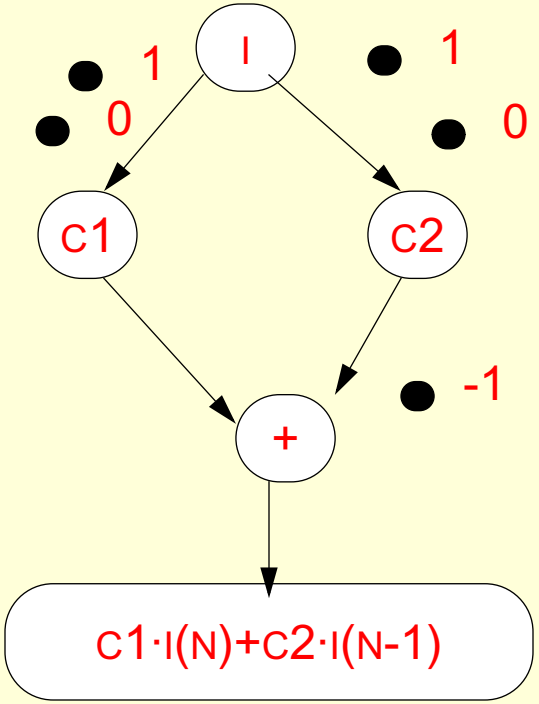


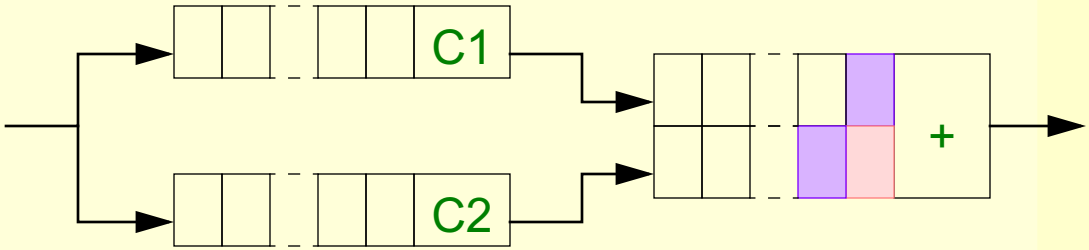
Modelo de Redes de Proceso de Flujo de datos (DF)



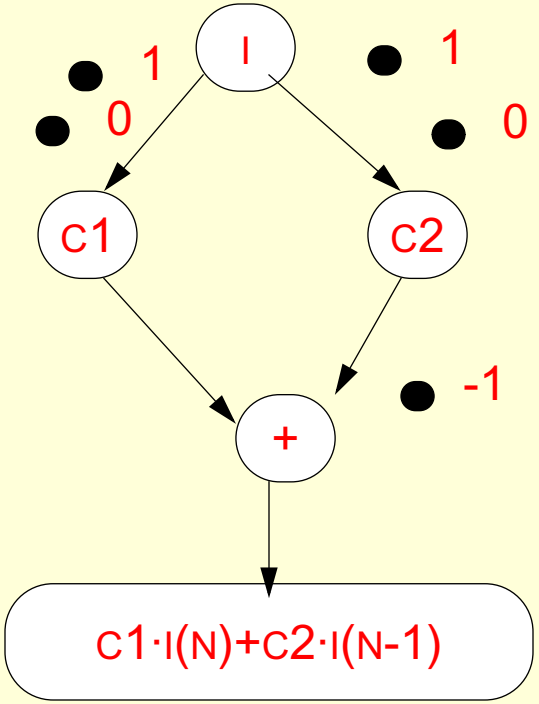


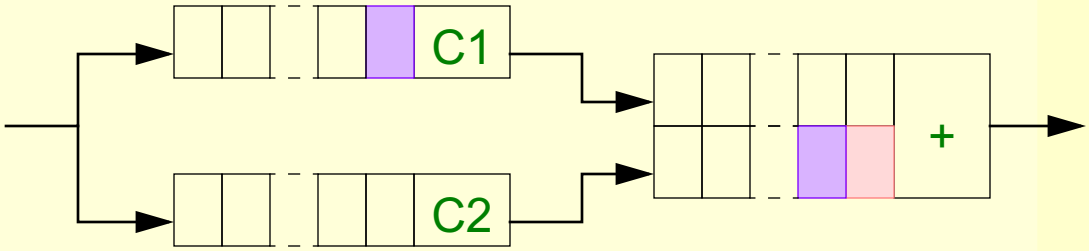
Modelo de Redes de Proceso de Flujo de datos (DF)



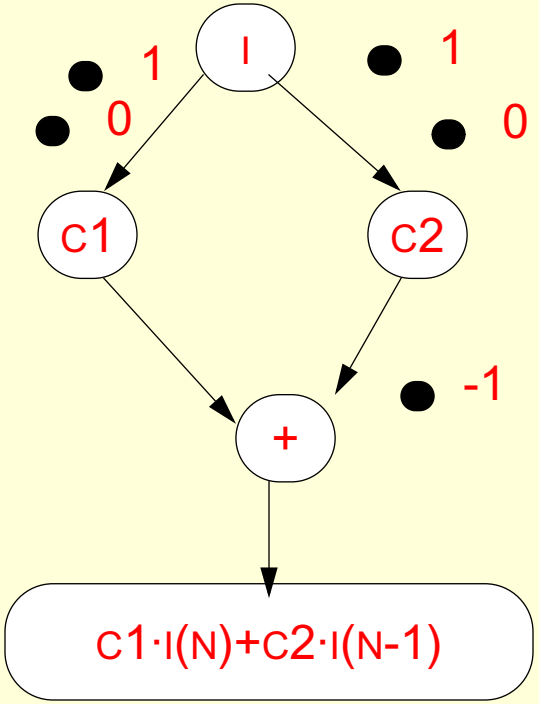


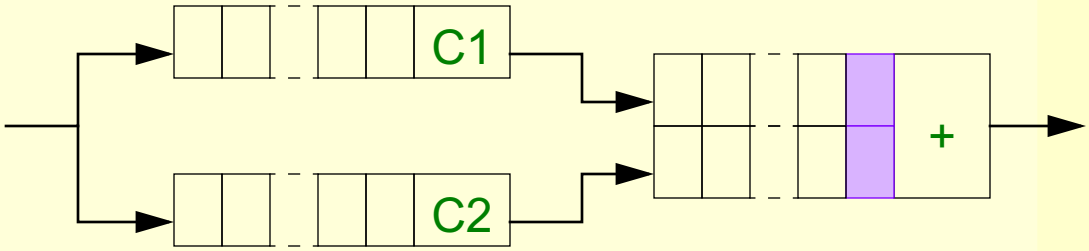
Modelo de Redes de Proceso de Flujo de datos (DF)



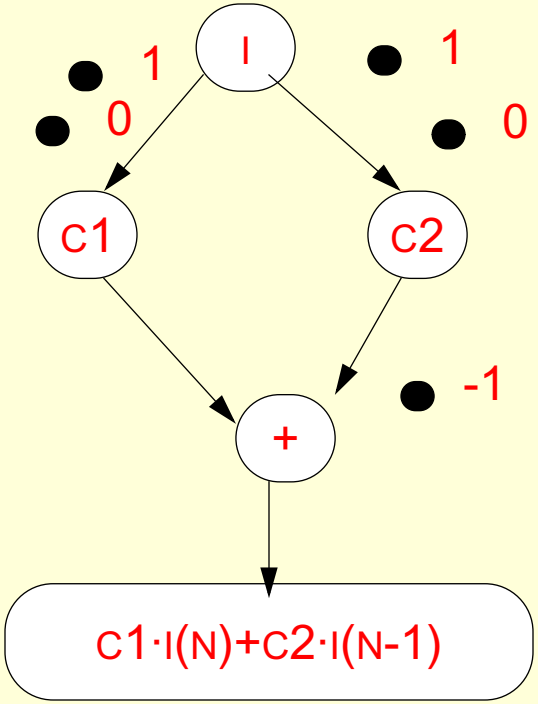


Modelo de Redes de Proceso de Flujo de datos (DF)





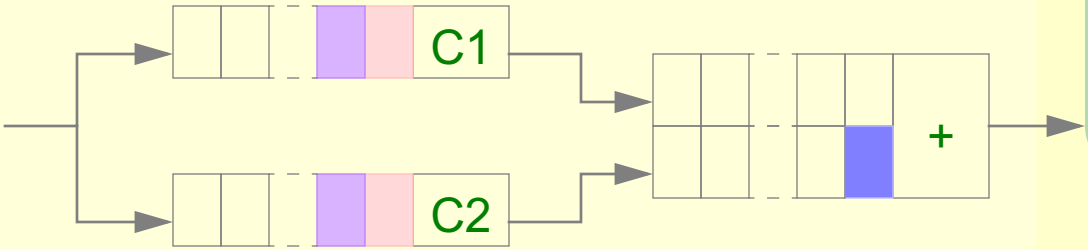
Modelo de Redes de Proceso de Flujo de datos (DF)



Fácil de usar

Algoritmos potentes para verificaciones y síntesis

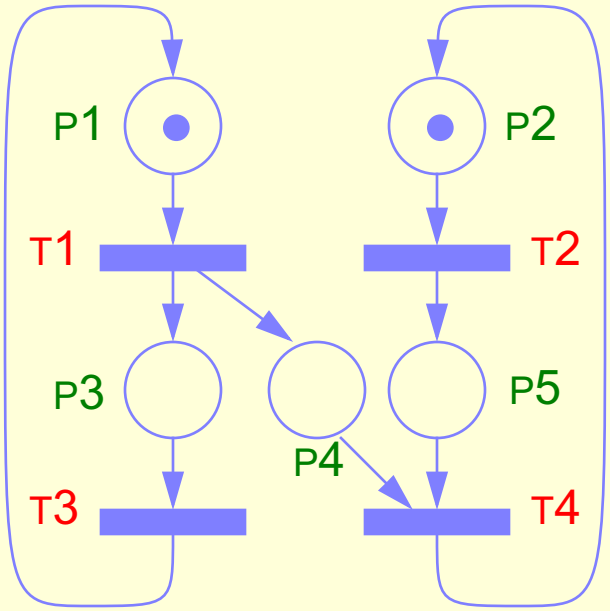
Concurrencia explícita



Eficaz sólo para modelos renstringidos

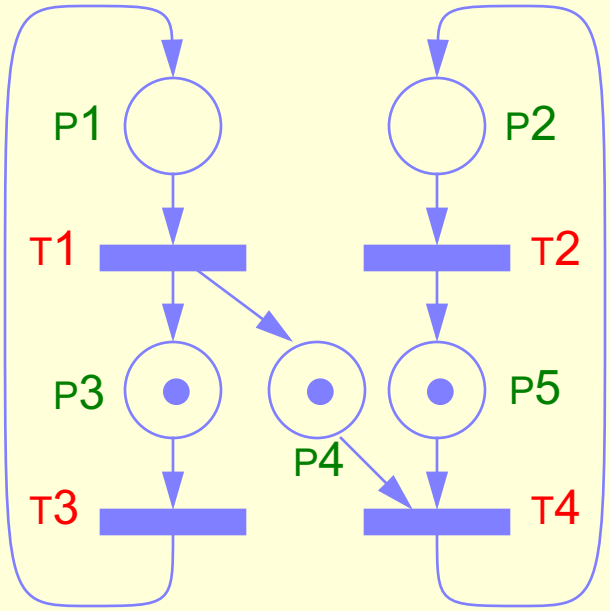
No describe control reactivo

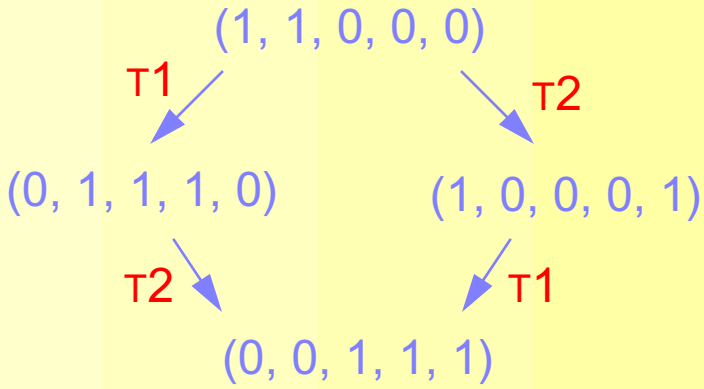
MODELO DE REDES DE PETRI (PN)



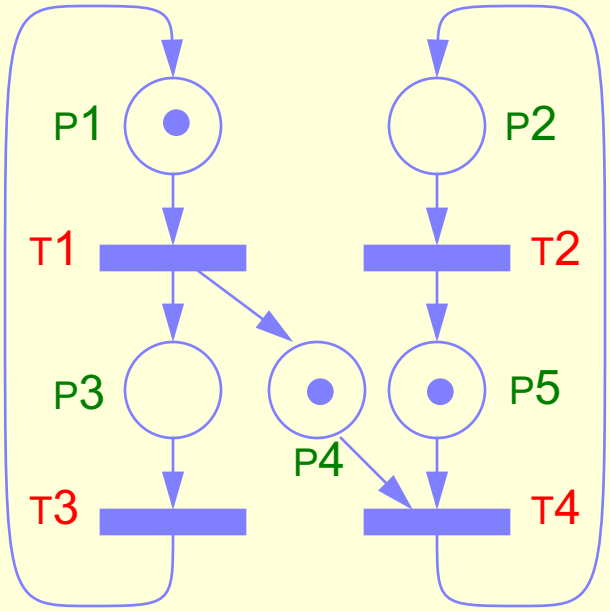
(1, 1, 0, 0, 0)

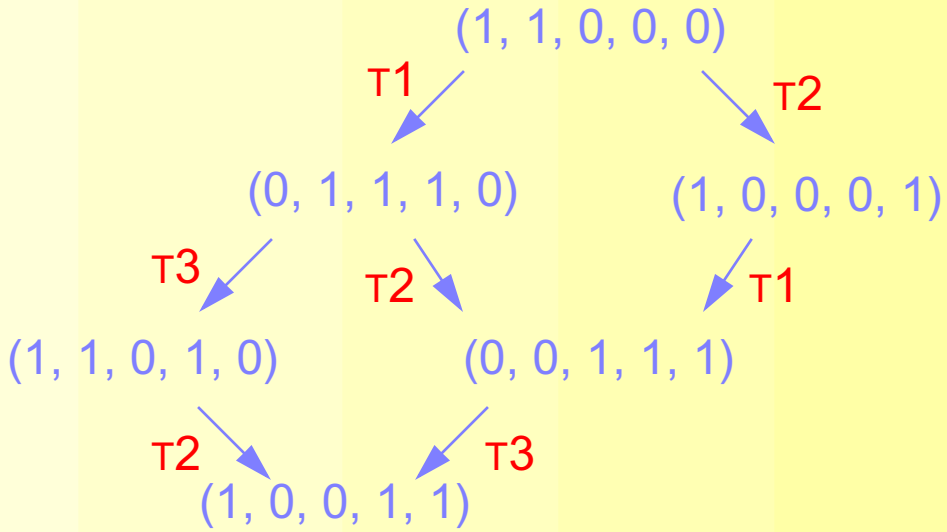
MODELO DE REDES DE PETRI (PN)



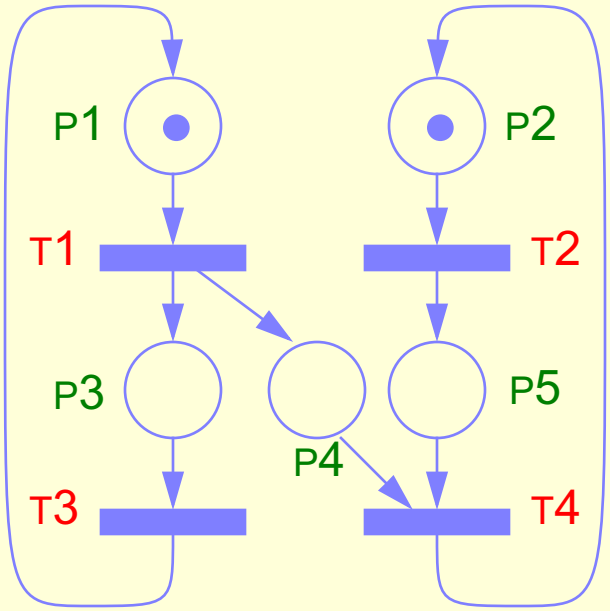


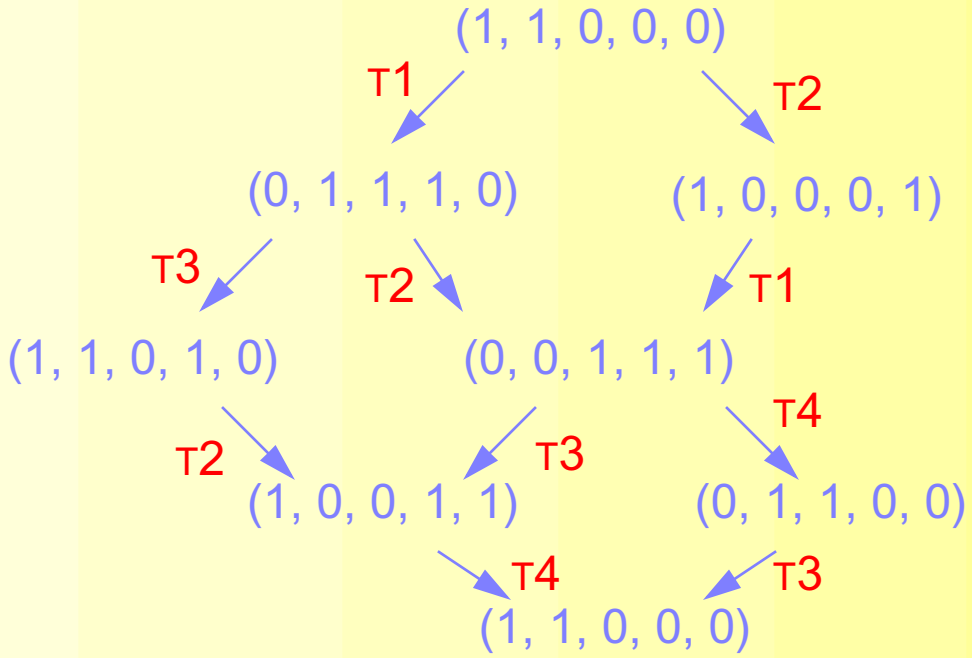
MODELO DE REDES DE PETRI (PN)



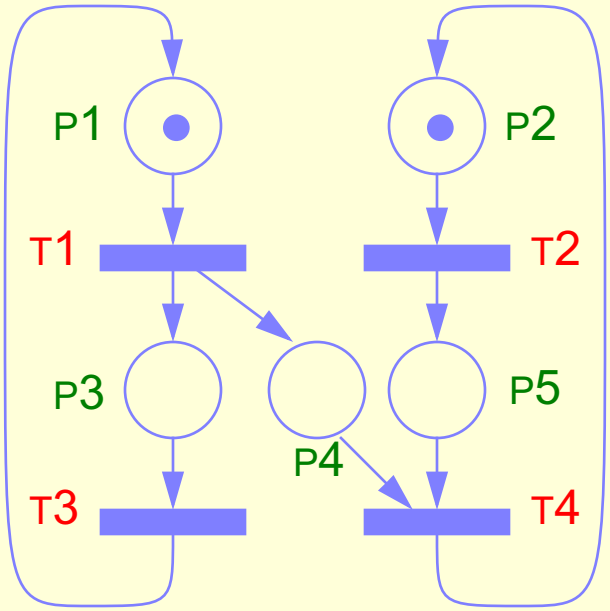


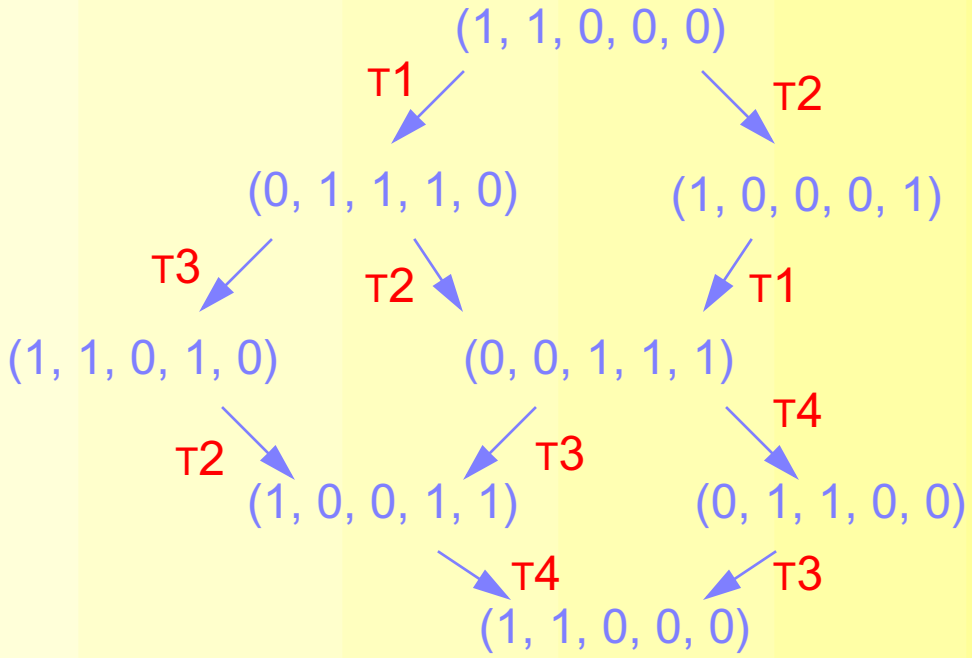
MODELO DE REDES DE PETRI (PN)





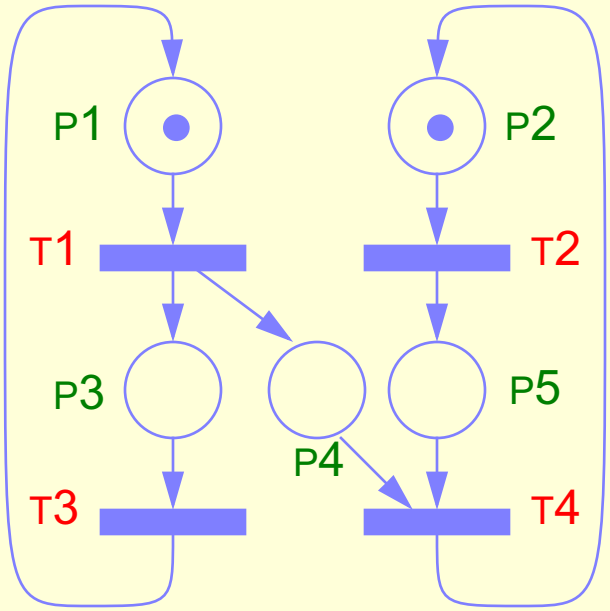
MODELO DE REDES DE PETRI (PN)

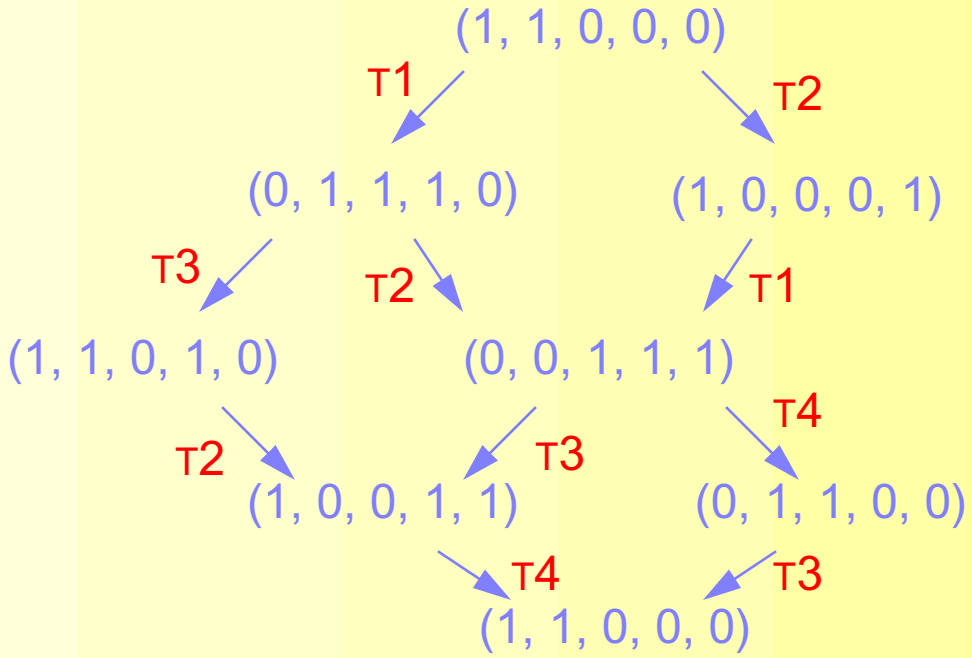




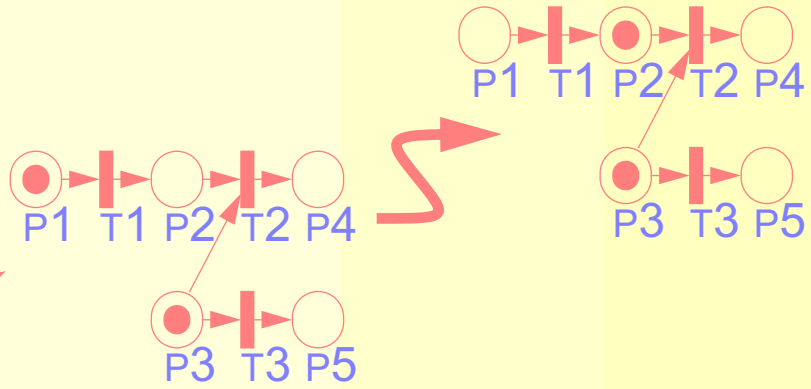
- LIVENESS
- BOUNDNESS
- FREECHOICE

MODELO DE REDES DE PETRI (PN)

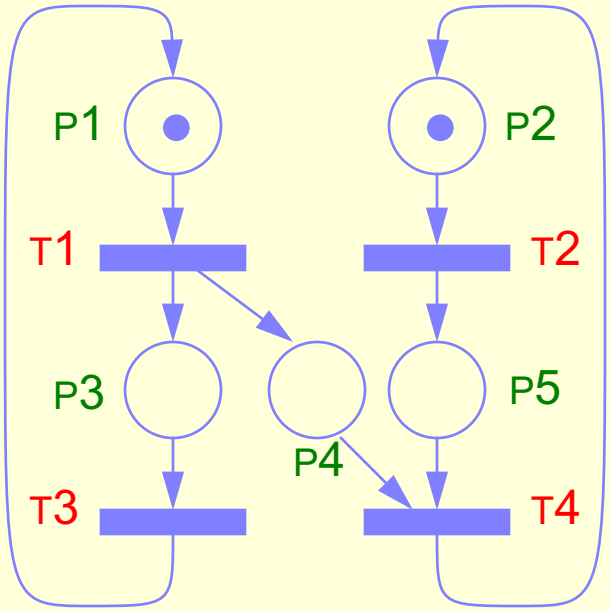


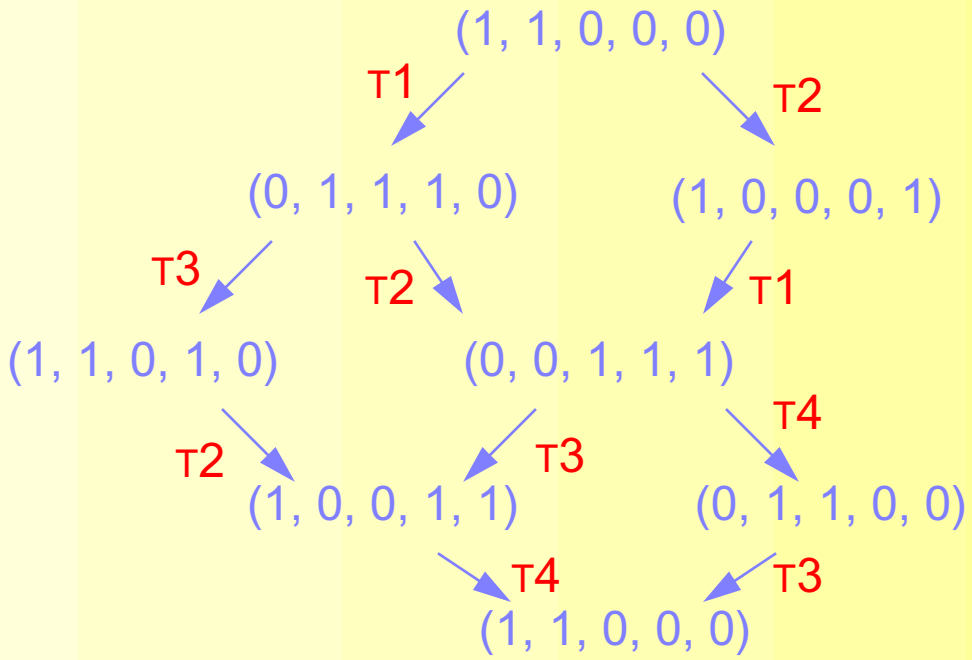


- LIVENESS
- BOUNDNESS
- FREECHOICE

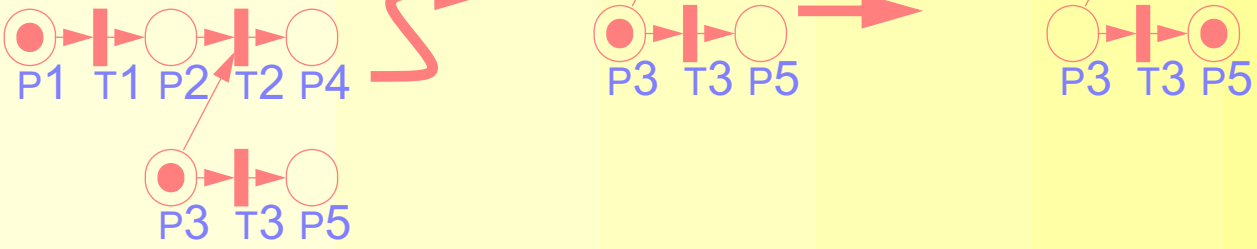


MODELO DE REDES DE PETRI (PN)

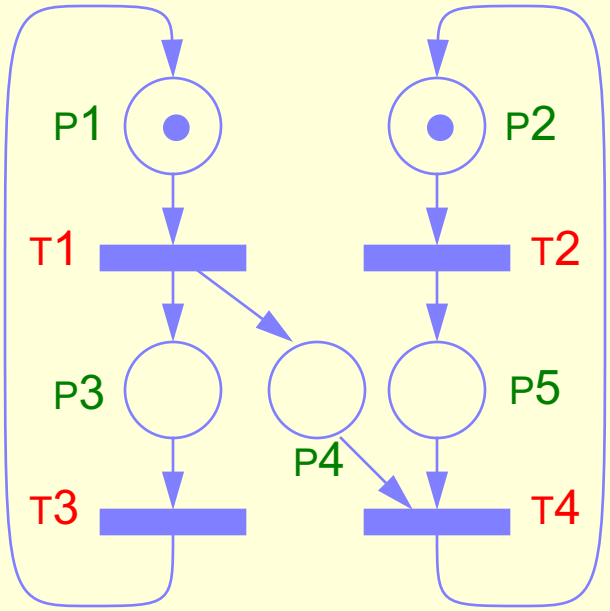


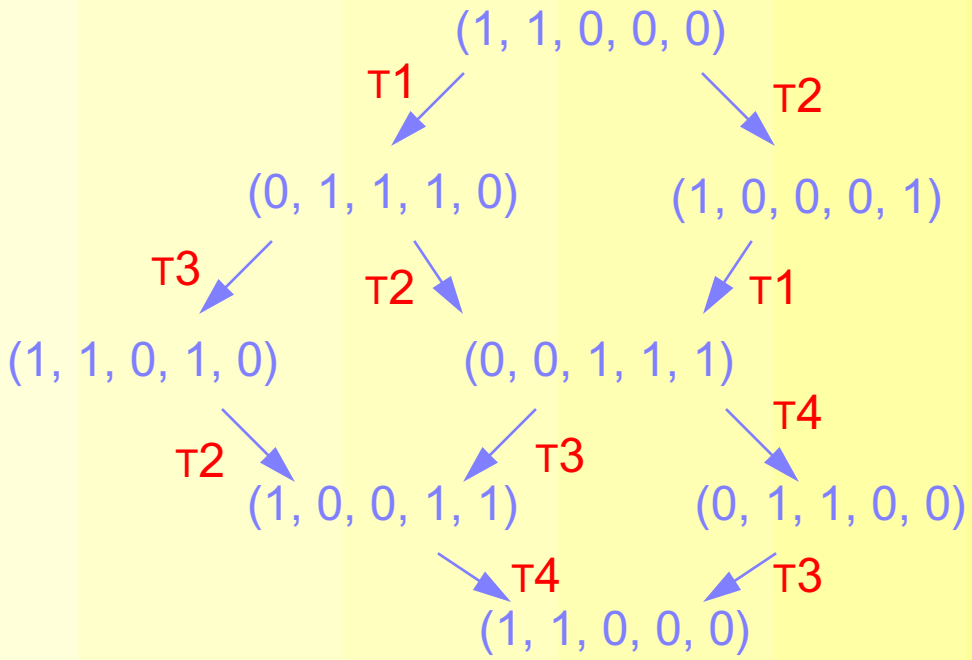


- LIVENESS
- BOUNDNESS
- FREECHOICE

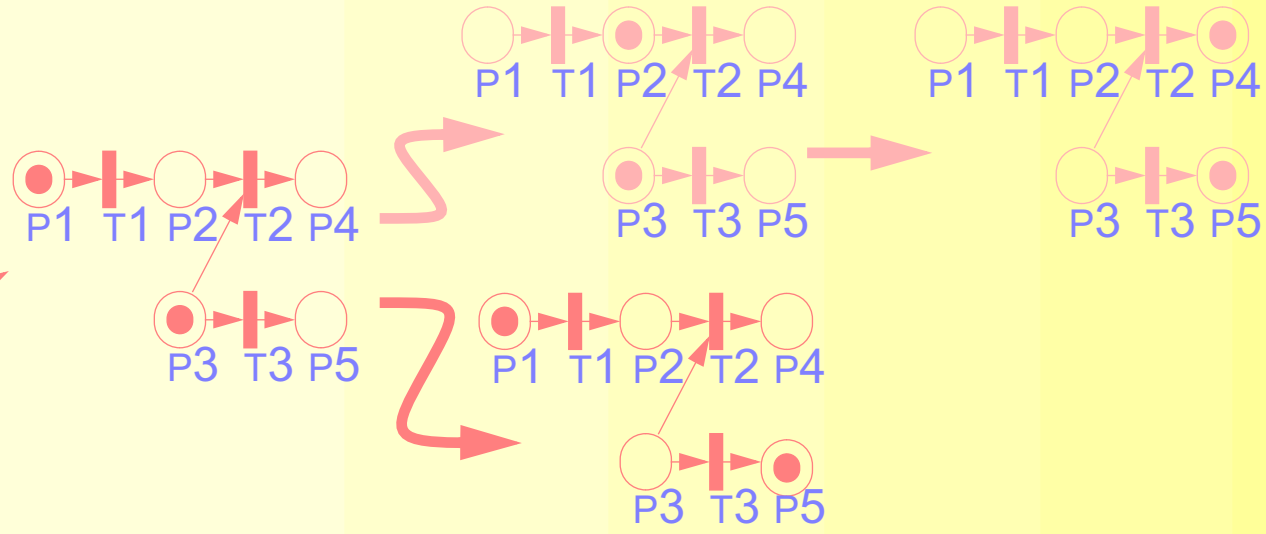


MODELO DE REDES DE PETRI (PN)

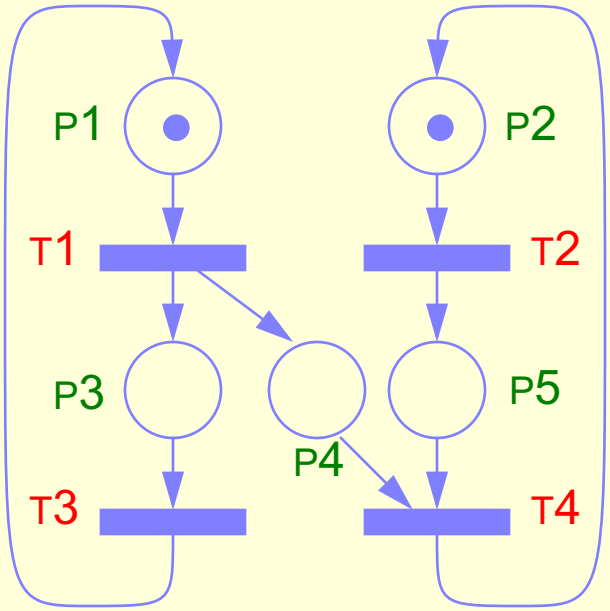


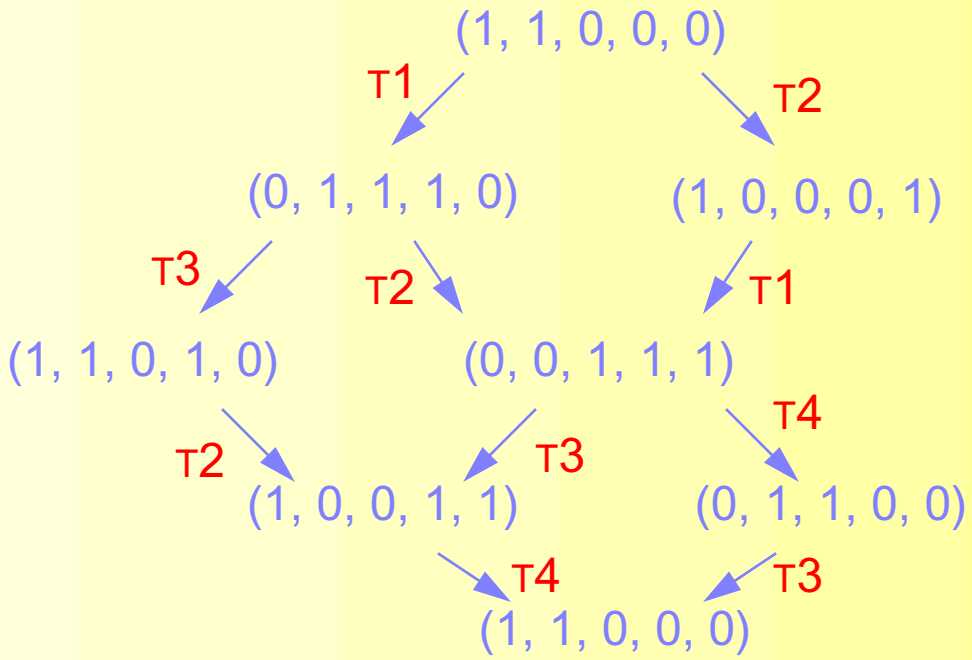


- LIVENESS
- BOUNDNESS
- FREECHOICE

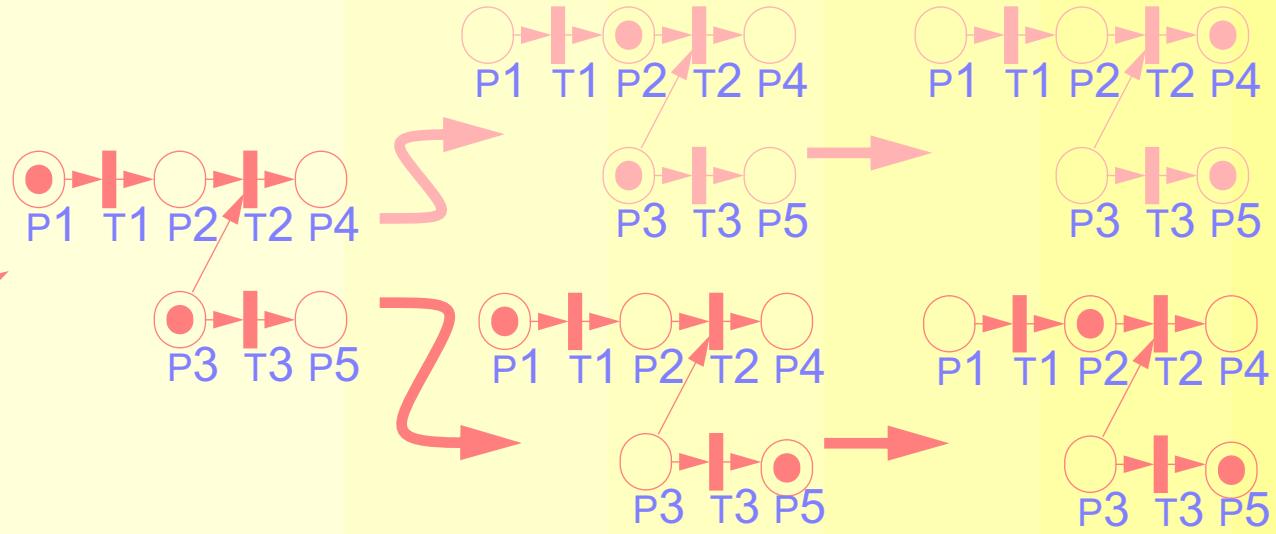


MODELO DE REDES DE PETRI (PN)

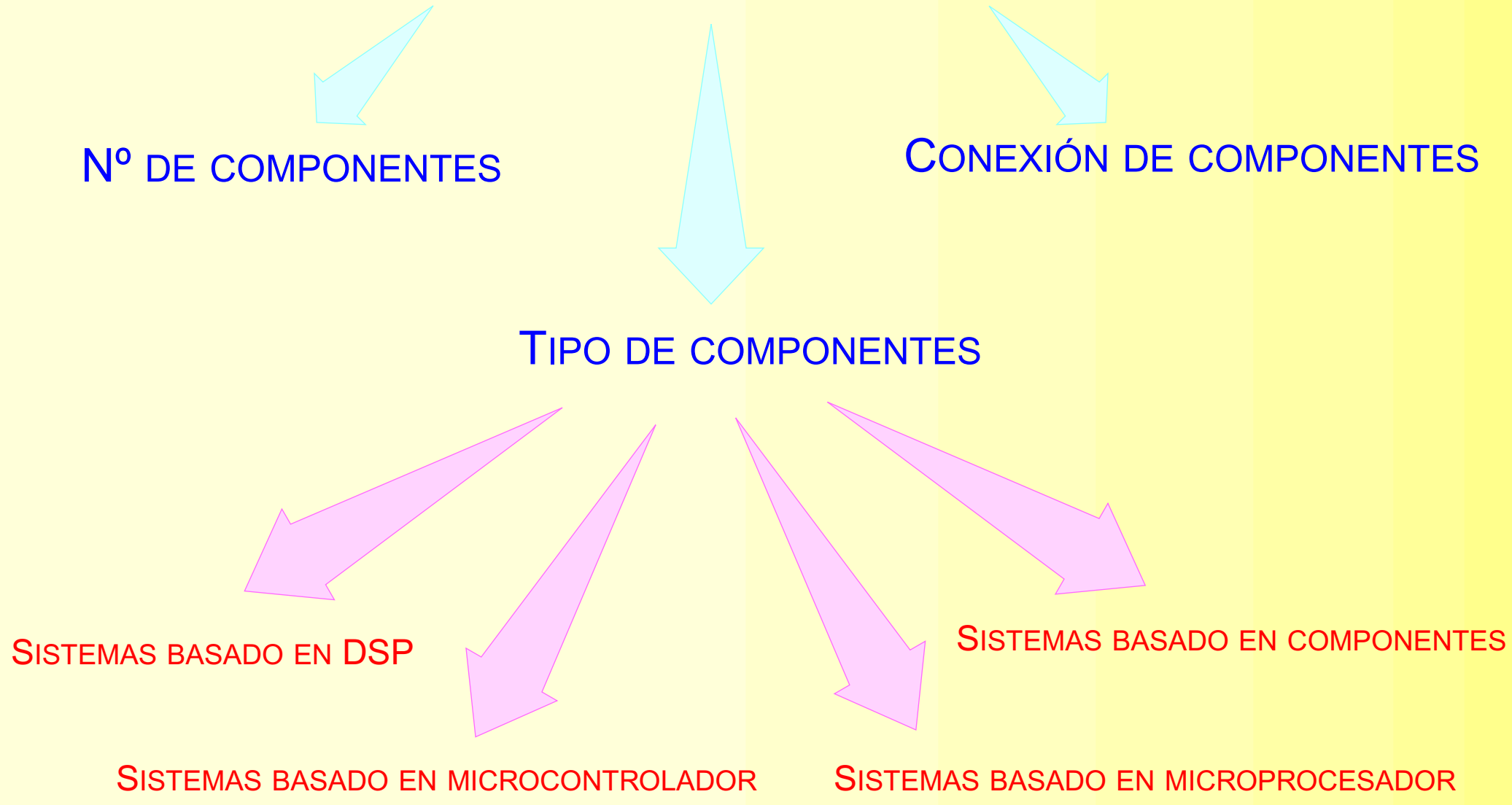




- LIVENESS
- BOUNDNESS
- FREECHOICE



MODELADO DE LA ARQUITECTURA



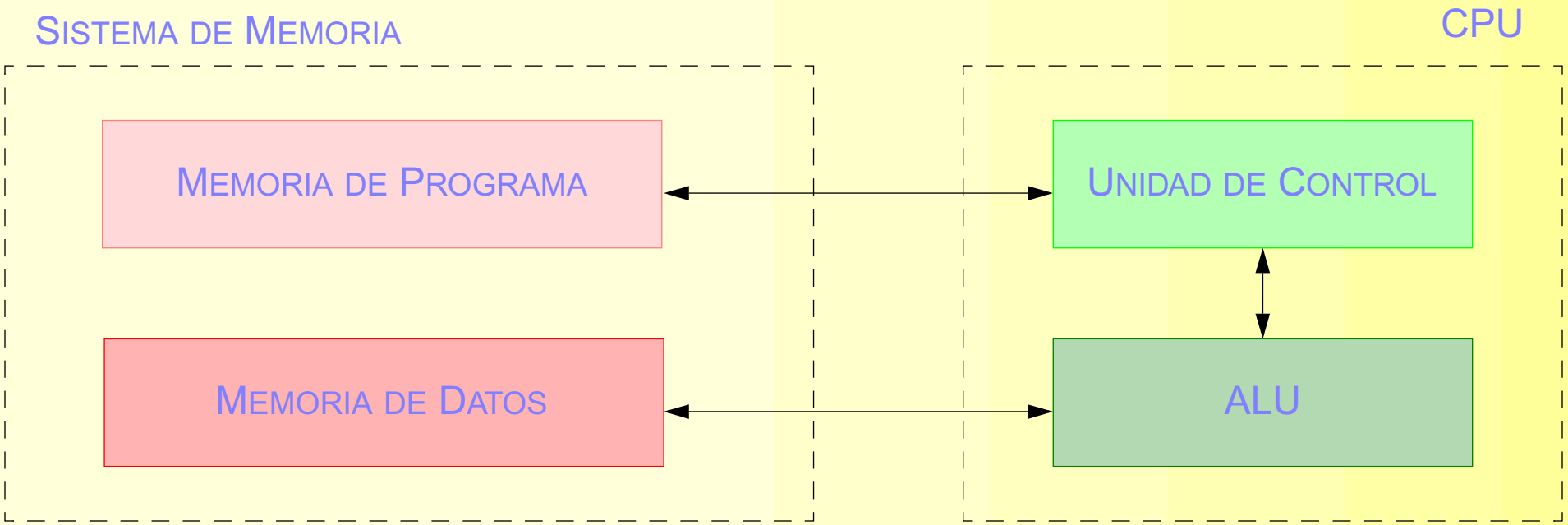
Sistemas basado en DSP

UN DSP ES UN DISPOSITIVO QUE EJECUTA UN ALGORITMO SOBRE SEÑALES DIGITALES, Y DEVUELVE LA SEÑAL DIGITAL TRATADA

- Algoritmos. (de)codificación de voz
- Tasas de muestreo. DSP multitasa
- Tasa de reloj
- Representación de nº. punto fijo/flotante

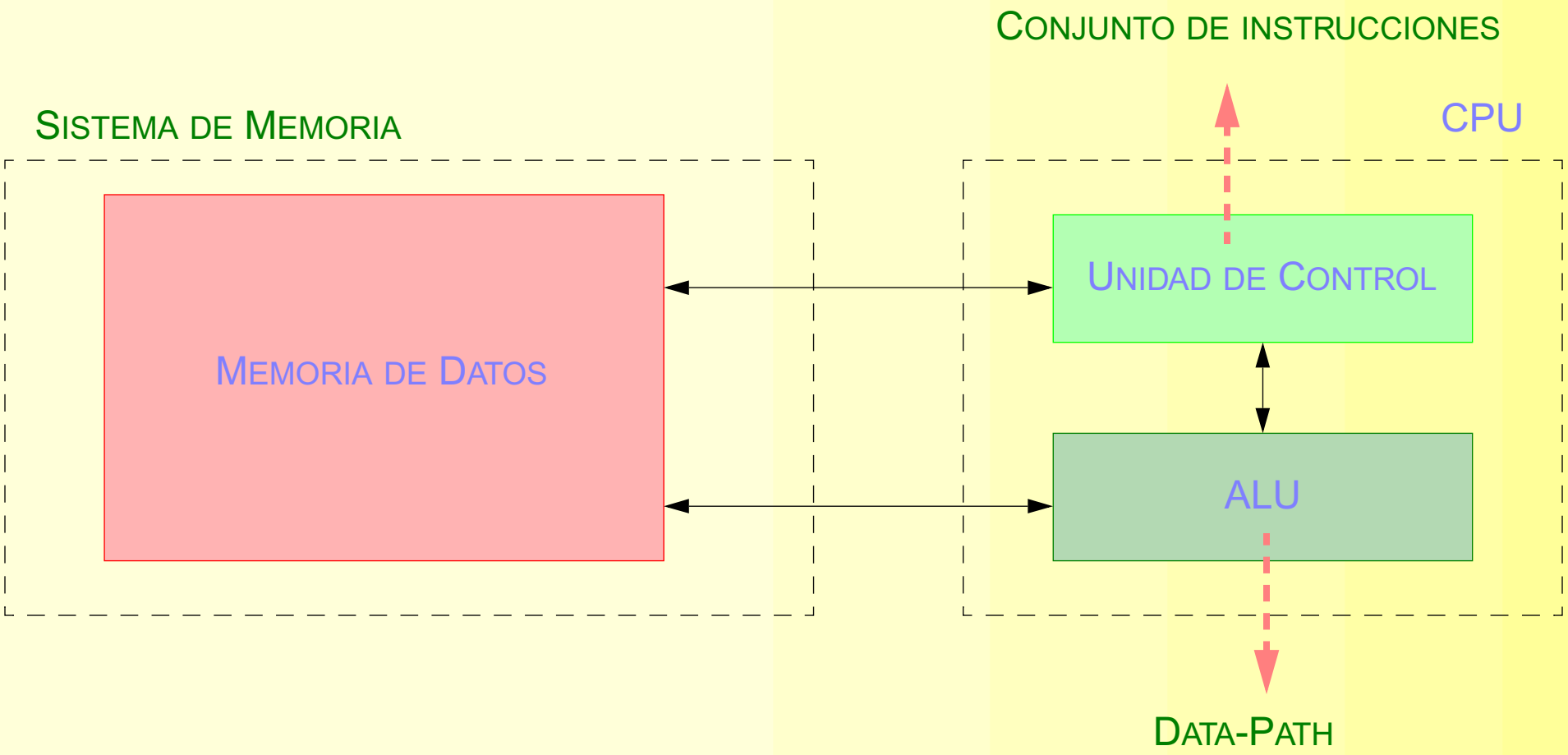
- Sistemas de bajo coste
- Sistemas de altas prestaciones
- Sistemas multimedia basados en PC

Sistemas basado en Microcontrolador (μC)



- Alto volumen de producción
- Alta fiabilidad y calidad
- Bajo coste y mantenimiento
- Tiempo de salida al mercado reducido
- Soporte para diversas versiones del sistema
- Arquitectura seleccionada previamente

Sistemas basado en Microprocesador (μP)



Sistemas basado en componentes

UN COMPONENTE ES UN SUBSISTEMA AUTOCONTENIDO QUE PUEDE SER UTILIZADO EN UN SISTEMA MAYOR