

Práctica 3. Codiseño Hardware/Software. SystemC – FSM

24 de Marzo de 2020. Entregar antes del 3 de Abril

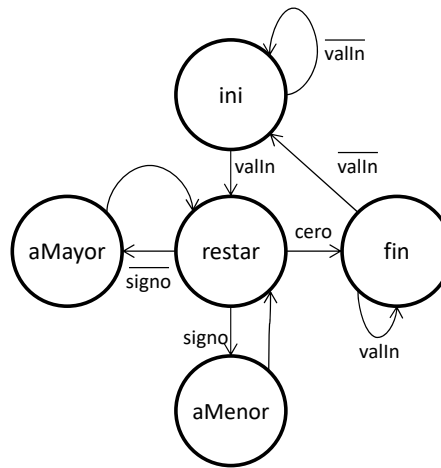
Calcular el máximo común divisor mediante una versión simplificada del algoritmo de Euclides

Es posible calcular el MCD utilizando únicamente restas. Dados 2 números A y B, se inicia un proceso iterativo, en el que el resultado de cada iteración es: el menor de los 2 números, y la diferencia entre el mayor y el menor. El resultado final se obtiene cuando A y B son iguales. El algoritmo de Euclides es más eficiente si existe gran diferencia entre los números.

<u>A</u>	<u>B</u>
15	12
3	12
3	9
3	6
3	3

En esta práctica, siempre se calculará A-B, así que cuando A sea menor que B, el resultado será negativo. En tal caso, se intercambiarán los valores de A y B y se volverá a intentar. En las siguientes figuras vemos el mismo ejemplo tal y como los debemos implementar, y el diagrama de estados:

<u>A</u>	<u>B</u>
15	12
3	12
12	3
9	3
6	3
3	3



- El archivo que debéis completar se llama mcd.cpp
- A y B son registros, y en el código SystemC están implementadas como sc_signal.
- El código que calcula la resta, signo y cero ya está hecho.
- La máquina de estados debe ser de Moore. Así que las salidas (listo y resultado) deben ser función del estado, exclusivamente
- En cada estado, se deben hacer las siguientes tareas:
 - ini: leer los valores entrantes entA y entB
 - restar: dejar que reste (nada en realidad)
 - aMayor: actualizar A
 - aMenor: intercambiar A y B
 - fin: activar la señal de listo

Comprender por qué funciona así, y solucionar los problemas que surjan es vuestra tarea.

Existe un único *productorConsumidor* que es totalmente funcional. El motivo por el que están fusionados es porque así es mucho más sencillo implementar el mecanismo de handshaking con vuestro módulo.

Los datos para simular son parte del *productorConsumidor*, así que no existen archivos de datos adicionales.