

Ejercicios sistemas algorítmicos

10

Ej. 1 — Diseñar un sistema algorítmico que calcule el máximo común divisor de dos números, N_1 y N_2 , basándose en el siguiente algoritmo.

```
A ← N1, B ← N2, MCD ← 0, Fin ← 0;  
while A ≠ 0 y B ≠ 0 do  
    if A > B then  
        | A ← A - B;  
    else  
        | B ← B - A;  
    end  
end  
Fin ← 1, MCD ← A;
```

Ej. 2 — Diseñar un sistema algorítmico que calcule la raíz cuadrada entera, r , de un número entero positivo, $x > 1$, basándose en el siguiente algoritmo.

```
r ← 1, d ← 2, s ← 4, Fin ← 0;  
while x ≥ s do  
    | r ← r + 1;  
    | d ← d + 2;  
    | s ← s + d + 1;  
end  
Fin ← 1;
```

Sólo se dispone de un sumador/restador para realizar todas las sumas y también para realizar la operación de comparación. Usar los componentes combinacionales que se consideren necesarios.

Ej. 3 — Diseñar un sistema algorítmico que controle la apertura de una cerradura. Las especificaciones son:

- Spec 1. Hay dos pulsadores, A y B, y la cerradura se abrirá únicamente si la secuencia de 8 pulsaciones coincide con la que tiene almacenada en un registro.
- Spec 2. La secuencia de control se programa de la siguiente forma: con la cerradura abierta, la secuencia AAAABBBB indica que las 8 pulsaciones siguientes son la nueva secuencia de control.
- Spec 3. Para cerrar se ha de aplicar la secuencia BBBBAAAA.

Se pueden definir las señales y condiciones de funcionamiento que se consideren necesarias.

Ej. 4 — Diseñar un sistema algorítmico que traduzca un número BCD de 4 dígitos al formato binario puro.

Ej. 5 — Diseñar un sistema algorítmico de reconocimiento de identidad en una sucursal bancaria. Las especificaciones son:

- Spec 1. El sistema permite reconocer a 256 usuarios.
- Spec 2. Cada usuario tiene asociado un código de identificación de usuario de 8 bits y una clave de 6 bits.

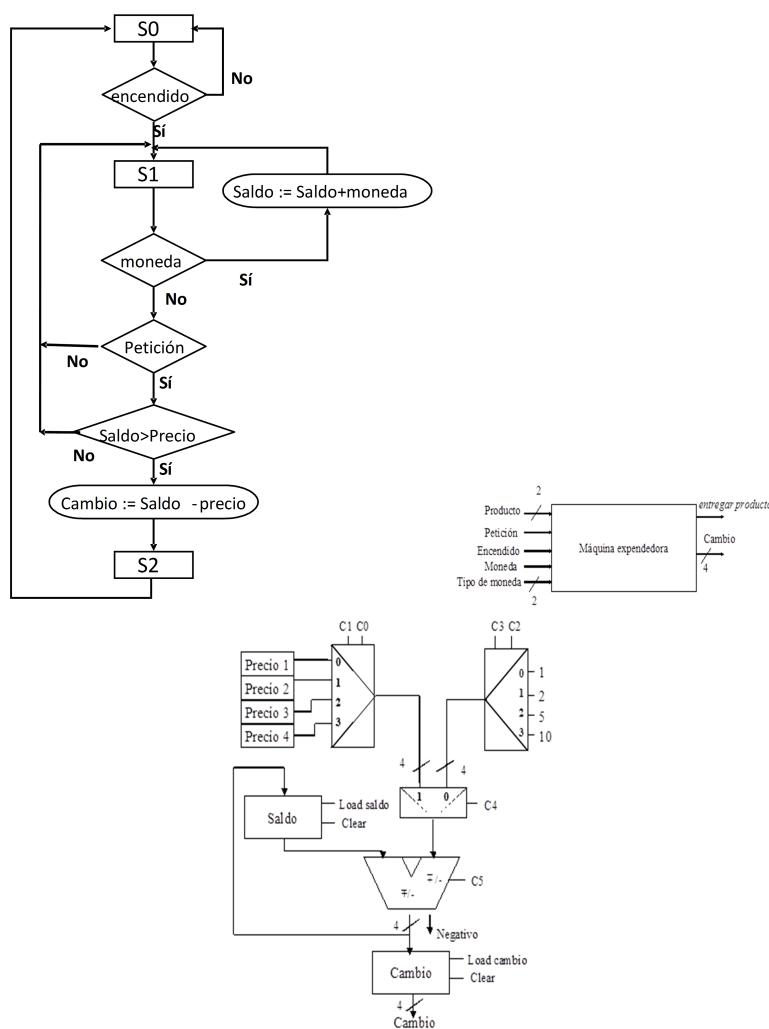
- Spec 3. Cuando un usuario del banco activa la señal reconocer, el sistema almacenará, primero, su código de identificación de usuario y, en el siguiente ciclo, su clave de acceso.
- Spec 4. La clave de acceso se compara con la que tiene almacenado el sistema, si coinciden la identificación es correcta y se activa la señal correcto. Si no coinciden, el sistema volverá a preguntar por el código de identificación de usuario.
- Spec 5. Si la identificación ha sido correcta entonces el usuario tendrá que pulsar el botón entrar para acceder a la entidad y el sistema activará la señal abrir durante un ciclo de reloj.
- Spec 6. Durante ese ciclo el usuario puede decidir cambiar la clave, y para ello tiene que pulsar el botón de cambio_clave.
- Spec 7. Si desea cambiar la clave, el sistema pregunta por la nueva clave y la almacena, activándose la señal de fin cuando el proceso concluye.
- Spec 8. Si no se desea cambiar la clave, el sistema espera 5 ciclos de reloj, transcurridos los cuales vuelve al estado inicial.

Diseñar el camino de datos, la unidad de control como una máquina Moore (especificando el diagrama de estados y la tabla de salidas) y la interconexión entre los dos módulos.

Ej. 6 — Una máquina expendedora vende cuatro productos de distintos precios y acepta monedas de 1, 2, 5 y de 10 céntimos. Su comportamiento es el siguiente:

- Spec 1. La máquina permanece inactiva hasta que se pulsa el botón encendido.
- Spec 2. A partir de ahí, la máquina incrementa el saldo (que inicialmente es cero) cada vez que se introduzca una moneda.
- Spec 3. Si el cliente presiona el botón de petición de uno de los productos (la entrada producto indica cual se ha solicitado), se comprueba si hay saldo suficiente y de ser así se activa la señal entregar_producto y se devuelve el cambio.
- Spec 4. En caso contrario se ignora la solicitud.

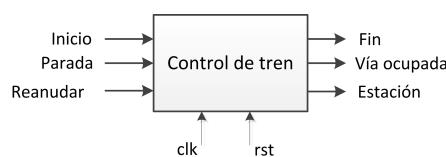
En las figuras aparecen el diagrama de ASM, el camino de datos y las entradas y salidas del sistema algorítmico. Diseñar la unidad de control para que el sistema funcione tal y como se describe en el diagrama de estados. La lógica para calcular el estado siguiente se debe implementar con un contador módulo 4 y puertas lógicas. Las señales de control se deben implementar con puertas lógicas.



Dibujar el cronograma del estado, las salidas y las señales de control para la siguiente secuencia de eventos.

1. En el ciclo de reloj 0, el usuario pulsa el botón encendido.
2. En el ciclo de reloj 1, el usuario introduce una moneda de 10 céntimos.
3. En el ciclo de reloj 2, el usuario solicita el producto 1. Supóngase que el precio del producto sea siete céntimos.
4. Finalmente transcurren dos ciclos más sin que se accione ningún botón.

Ej. 7 — Diseñar un sistema algorítmico para controlar un tren que viaja desde la estación Origen a la estación Destino, situada a 50 km, deteniéndose en una estación intermedia, que está a 15 km de Origen. El sistema tiene las siguientes entradas y salidas.



Las especificaciones son:

Spec 1. El tren comienza su viaje cuando el maquinista pulsa inicio.

- Spec 2. En ese momento el tren se pone en marcha (avanza 1 km por ciclo de reloj) y activa la salida `tren_circulando`, que permanece a 1 hasta que el tren llegue a su destino.
- Spec 3. El tren sigue su marcha hasta que llega a la estación intermedia. En ella para 10 ciclos, durante los cuales está activa la salida `estación`.
- Spec 4. Cuando el tren llegue a su destino activa `fin`.
- Spec 5. Si en cualquier momento el maquinista pulsa `parada_emergencia` el tren se detiene hasta que el maquinista pulse `reanudar`.

Dadas las entradas y salidas del diagrama anterior se pide diseñar la unidad de control como una máquina de Moore, usando contadores módulo 4 y puertas lógicas.

Ej. 8 — Diseñar el sistema algorítmico que controla el funcionamiento de un túnel de lavado de coches. Las especificaciones son:

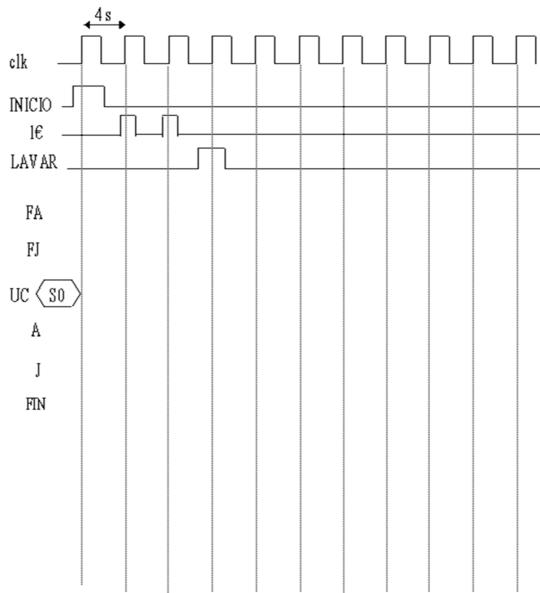


- Spec 1. El sistema tiene cinco entradas: `inicio`, monedas (monedas de 1€), `lavar`, `fa` (fin del agua) y `fj` (fin del jabón).
- Spec 2. El sistema tiene tres salidas: `a` (agua), `j` (jabón), `fin` (saca el coche del túnel).
- Spec 3. El usuario pulsa el botón `inicio` y después introduce 1 ó 2 monedas de 1€ en función del tipo de lavado que desee. Después pulsa el botón `lavar`. El sistema empieza a funcionar y el usuario simplemente espera a que el coche salga del túnel de lavado.
- Spec 4. Si el número de monedas introducidas es 0, la máquina no hace nada y vuelve al estado inicial.
- Spec 5. Si el número de monedas es 1 entonces el sistema activa la señal `a` ($A = 1$) para que caiga agua sobre el coche. Cuando se activa la señal `fa` ($FA = 1$) entonces el sistema activa la señal `fin` y vuelve al estado inicial.
- Spec 6. Si el usuario ha introducido 2 o más monedas entonces (1) el sistema primero mojará el coche (activa señal `a`), (2) cuando se activa la señal `fa` finaliza el aclarado y lo enjabona activando la señal `j` ($j = 1$) y (3) cuando se activa la señal `fj` y lo aclara activando de nuevo la señal `a`. El aclarado finaliza cuando $fa = 1$, se activa la señal `fin` y vuelve al estado inicial.
- Spec 7. Esta máquina no devuelve cambio ni detecta cantidades mayores de 3€.

La solución del problema consiste en:

- 1.Dibujar el diagrama ASM.
- 2.Diseñar el camino de datos utilizando los módulos combinacionales y secuenciales estudiados en la asignatura, indicando claramente sus entradas, salidas y la función que realizan (en caso de señales de control, función que realizan para cada posible valor o combinación de valores) y en el caso de los registros, y líneas de datos, su anchura en bits.
- 3.Dibujar la unidad de control y el camino de datos como dos cajas negras. Sobre dicho esquema, indicar todas las señales que entran y salen de las cajas (incluidas las que van de UC a CD y viceversa) y la función que realiza cada una de ellas y momentos en que se activa en relación al diagrama de flujo.

4. Diseñar la UC como máquina de Moore: (a) diagrama de estados en que se refleje la relación de los estados con el diagrama de flujo del primer apartado, (b) tabla de estados y salidas Añadir las señales de control internas del sistema diseñado y completar el cronograma siguiente (incluyendo el estado siguiente y el valor de las señales después de activar la señal fin):

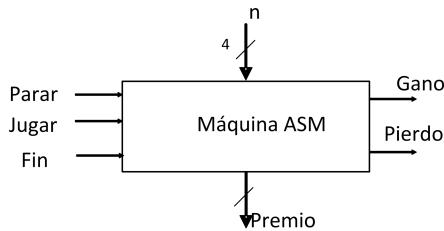


Nota: El tiempo programado para que caiga agua sobre el coche es 6 s (contados a partir de la activación de la señal a), y lo mismo para el enjabonado: son 6 s a partir de la activación de las correspondientes señales.

Ej. 9 — Diseñar un sistema algorítmico que controla un juego electrónico consistente en adivinar un número entre 0 y 15. Debe satisfacer las siguientes especificaciones:

- Spec 1. La máquina proporciona 3 oportunidades para acertar el número, y dispone de 3 premios diferentes codificados según la tabla 1.
- Spec 2. El juego dispone de un contador funcionando a una frecuencia de 100MHz que genera números del 0 al 15.
- Spec 3. El funcionamiento del juego es el siguiente: En primer lugar el juego está parado hasta que se pulsa el botón de Jugar.
- Spec 4. En ese momento, el usuario introduce el número que cree va a salir, y el sistema comienza a generar números a una frecuencia de 100MHz, de forma que el usuario desconoce qué salida tiene el contador en cada momento.
- Spec 5. El sistema continúa generando números hasta que el jugador decide parar el juego y adivinar el número, pulsando el botón Parar.
- Spec 6. Si el número introducido por el usuario no coincide con el generado, el sistema ofrece al usuario otra oportunidad de juego (así hasta un total de 3 oportunidades), de modo que el usuario introduce un nuevo número y el sistema vuelve a generar más números aleatoriamente.
- Spec 7. Si tras 3 fallos el usuario no ha acertado el número, se activará la salida Pierdo durante un ciclo de reloj, volviendo a continuación el sistema al estado inicial.
- Spec 8. Si en alguna de las oportunidades permitidas el usuario acierta el número, se activa la señal Gano hasta que el sistema ha terminado de dar el premio. Cuando

se ha dado el premio, la máquina que suministra las monedas activa la señal de Fin (entrada de control de la máquina ASM) y el sistema vuelve al estado inicial.



Oportunidad	Premio
Primera	100
Segundo	50
Tercera	25

Dibujar el diagrama ASM, el camino de datos –especificando el tamaño de registros, las entradas y salidas de los módulos y las señales de control– y la unidad de control –diagrama de estados y tabla de salidas–.

Ej. 10 — Un sistema algorítmico suma todos los elementos de un vector que se encuentra almacenado en una memoria SRAM síncrona de un solo puerto. Las especificaciones del sistema son:

- Spec 1. El sistema posee dos entradas: la dirección inicial del vector, dir de 8 bits, y su longitud, n de 4 bits.
- Spec 2. Estas señales son almacenadas cuando se activa la señal ini.
- Spec 3. Los elementos del vector son de tamaño 6 bits.
- Spec 4. Los elementos del vector se leen uno a uno y se calcula su suma.
- Spec 5. Cuando el sistema termina de operar activa la señal fin que indica que la salida resultado –de 8 bits– tiene el valor correcto y vuelve al estado inicial.

Diseñar el camino de datos utilizando un sumador, dos contadores, un comparador y el número mínimo de registros del tamaño adecuado. Diseñar la unidad de control que se especificará con el diagrama de estados y la tabla de salidas.

Ej. 11 — Diseñar un sistema algorítmico que sume dos operandos en formato BCD situados en una memoria SRAM síncrona de un solo puerto y exprese el resultado en formato BCD. Las especificaciones del sistema son:

- Spec 1. El sistema posee dos entradas: dirA y dirB.
- Spec 2. El sistema posee una salida: suma.
- Spec 3. Las entradas son almacenadas cuando se activa la señal empezar.
- Spec 4. En 1 ciclo de reloj sólo se puede acceder a un dato almacenado en memoria.
- Spec 5. Los datos almacenados en la memoria están en formato BCD.
- Spec 6. Primero se debe leer el dato almacenado en la posición dirA. Despues el dato almacenado en la posición dirB.
- Spec 7. Los dos datos deben sumarse y el resultado debe representarse de forma correcta con dos dígitos BCD. Así, si el resultado es menor que 10 no hace nada más, pero si es mayor se debe sumar 6 para que el resultado esté correctamente representado.
- Spec 8. Cuando haya finalizado la operación, la señal fin se pone a 1 y el sistema vuelve al estado inicial.

A continuación dos ejemplos de funcionamiento:

Ejemplo Si $A = 1000$ y $B = 0001$ entonces $A + B = 00001001$. Si se interpreta el resultado 1. como dos dígitos BCD es lo mismo que decir que $08 + 01 = 09$.

Ejemplo Si $A = 1000$ y $B = 1001$ entonces $A + B = 00010001|_2 = 11|_{BCD}$ que es claramente 2. incorrecto. Para obtener el resultado correcto se debe sumar 6. Así, $08 + 09 = 11$ y para convertirlo a BCD $11 + 06 = 17$.

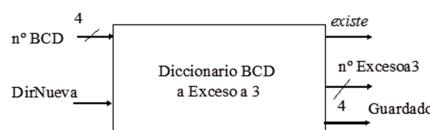
El camino de datos debe realizarse con los siguientes módulos: una memoria SRAM síncrona con un único puerto, tres registros de 8 bits, un sumador y un módulo combinacional que indique si un número de 8 bits es menor que 10. Aparte de estos módulos únicamente se pueden utilizar multiplexores. La unidad de control se especificará con el diagrama de estados y la tabla de salidas.

Ej. 12 — Diseñar un diccionario de dígitos BCD a exceso a 3. Las especificaciones del sistema son:

- Spec 1. El sistema tiene dos entradas: un dígito BCD, bcd , y la señal que marca la presencia de un dígito BCD nuevo, $DirNueva$.
- Spec 2. El sistema tiene dos salidas: $existe$, guardado y bcd_out .
- Spec 3. El valor de bcd se almacena cuando la señal $DirNueva$ vale 1.
- Spec 4. A partir del valor de bcd el sistema genera la traducción a exceso a 3 y busca la palabra almacenada en el diccionario asociada con el dígito BCD introducido.
- Spec 5. Si la palabra almacenada es igual al dígito codificado en exceso a 3 la señal $existe$ se pondrá a 1 indicando que la traducción ya existía en el diccionario, y a continuación el sistema volverá al estado inicial.
- Spec 6. De no existir, la traducción se almacena en el diccionario y cuando el proceso ha terminado se activará la señal $guardado$, indicando que la traducción de BCD a exceso a 3 del dígito de entrada se ha almacenado correctamente, y el sistema volverá al estado inicial.
- Spec 7. En cualquier caso, el dígito codificado en exceso a 3 se visualizará en la salida bcd_out .

Diseñar el camino de datos utilizando una SRAM síncrona 16×4 , un único sumador/restador, 2 registros y el mínimo número de multiplexores necesarios. El sumador/restador cuenta con una salida $Rcero$ que cuando vale uno indica que la salida de datos del sumador/restador está a cero. Diseñar la unidad de control presentando su diagrama de estados y la tabla de salidas.

Nota: Un dígito en exceso a 3 es igual al dígito BCD más 3.



Ej. 13 — Diseñar un juego electrónico de cartas similar a las 7 y media. Las especificaciones del sistema son:

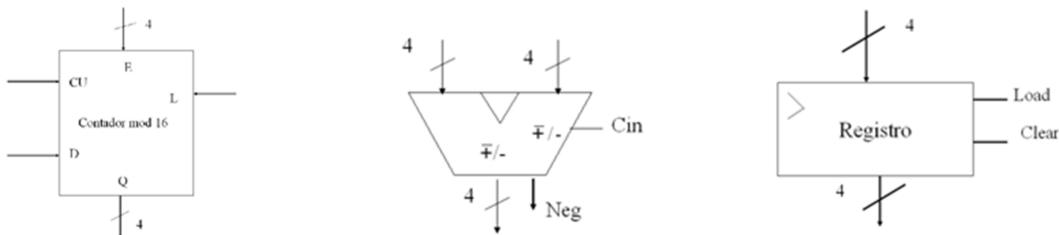
- Spec 1. El juego dispone de un contador que genera cartas con valores de 1 a 10.
- Spec 2. Inicialmente el juego está detenido hasta que se pulsa el botón *jugar*.
- Spec 3. Al pulsar el botón *jugar*, el sistema comienza a barajar las cartas hasta que el jugador activa la señal *pedir_carta*.
- Spec 4. La suma de todas las cartas pedidas se almacena en el registro marcador.

10. EJERCICIOS SISTEMAS ALGORÍTMICOS

- Spec 5. Cada vez que se pide una carta se actualiza marcador y se compara con 11. Si $marcador > 11$, el jugador pierde, el sistema activa la señal *pierdo* y regresa al estado inicial.
- Spec 6. Si por el contrario, $marcador \leq 11$, el sistema vuelve a barajar y el jugador decide si pide o no una carta nueva.
- Spec 7. Si decide no pedir carta, puede ser debido a que quiere seguir barajando o se quiere plantar. En este último caso, se activa la señal *fin* y el sistema vuelve al estado inicial.

Diseñar el camino de datos –en su implementación sólo se pueden usar sumadores, contadores, registros (como los de las figuras) y puertas lógicas– y la Unidad de control –diagrama de estados y la tabla de salidas–. Dibujar un cronograma que describa el estado, las salidas y las señales de control para la siguiente secuencia de eventos:

- 1.Ciclo 1: *jugar* = 1.
- 2.Ciclo 2: *pedir_carta* = 1 y sale un 4.
- 3.Ciclo 4: *pedir_carta* = 1 y sale un 5.
- 4.Ciclo 6: *pedir_carta* = 0, *plantar* = 1.



Ej. 14 — Diseñar el sistema algorítmico para una máquina expendedora de tickets de aparcamiento. Las especificaciones del sistema son:

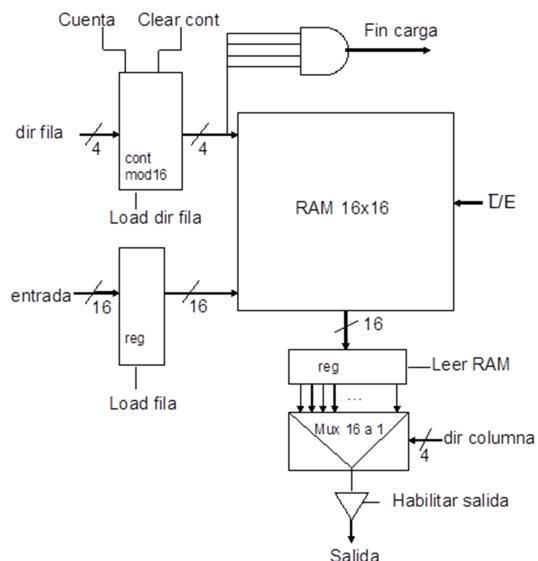
- Spec 1. La máquina tiene 2 entradas: *inicio* (desbloquea la ranura y permite introducir monedas de 1€ solamente) y *IMPR_TICKET* que ordena la impresión del ticket.
- Spec 2. La máquina imprime 3 tipos de tickets: 20 min. de aparcamiento (1€); 40 min. (2€) y 60 min. (3€).
- Spec 3. Si se pulsa al botón *inicio* y luego *IMPR_TICKET* y no se han introducido monedas, la máquina no hace nada y vuelve al estado inicial.
- Spec 4. El procedimiento para sacar un ticket es:
 - 1.Pulsar *inicio*.
 - 2.Introducir 1, 2 o 3 monedas de 1€ según el tiempo deseado.
 - 3.Pulsar *IMPR_TICKET*.

Diseñar el camino de datos –indicando claramente las entradas, salidas y señales de control del sistema, así como la función que realizan los módulos utilizados para el camino de datos– y la unidad de control.

Ej. 15 — Diseñar un multiplicador de números de 4 bits usando un contador. Describir el diagrama ASM y diseñar el camino de datos –indicando el tamaño de los registros, las entradas, las salidas y las señales de control así como la función que realizan los módulos utilizados– y la unidad de control.

Ej. 16 — Diseñar un sumador de números binarios de 8 bits en magnitud y signo –1 bit de signo y 7 de magnitud–. Describir el diagrama ASM y diseñar el camino de datos –indicando el tamaño de los registros, las entradas, las salidas y las señales de control así como la función que realizan los módulos utilizados– y la unidad de control.

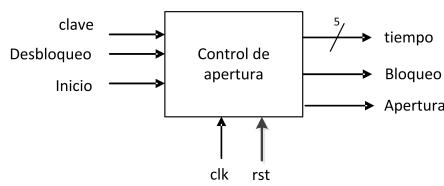
Ej. 17 — Diseñar un sistema algorítmico que controle el funcionamiento de un juego de barcos. En la figura se muestra el camino de datos utilizada para implementar una parte del juego. La señal de reloj no aparece para simplificar la figura. El juego comienza cuando el jugador pulsa el botón de encendido. Si en esos momentos también se ha pulsado el botón Nuevo_mapa, el sistema entra en el modo grabar una configuración nueva. En la memoria RAM se almacenan secuencialmente las 16 palabras de 16 bits que definen el nuevo mapa, en el que se han situado los barcos correspondientes. Si por el contrario, no se pulsa el botón Nuevo_mapa, el jugador suministra las coordenadas (fila, columna) y el circuito debe informarle si en esa posición hay un barco o hay agua (1 quiere decir tocado y 0 agua). Hacer el diagrama de flujo que dirige el funcionamiento del juego. Diseñar la unidad de control, incluyendo el diagrama de estados y la tabla de salidas.



Ej. 18 — El laboratorio de Rendimiento Deportivo de la Facultad de Medicina cuenta con un equipo digital para la medida de velocidad instantánea en atletas. Para ello, los atletas se disponen al inicio de una plataforma horizontal y, cuando desean comenzar el experimento, activan el pulsador de Inicio. El sistema de medida detecta al atleta en cuestión gracias a que su código ID (de 4 bits) se encuentra codificado en una tarjeta de identificación por radiofrecuencia, la cual porta el atleta en su bolsillo. Inmediatamente después de activar el pulsador de Inicio, el atleta comienza a correr a lo largo de la plataforma, siendo su movimiento detectado gracias a un sensor de presión. Mientras dura el movimiento del atleta, el tiempo transcurrido (siempre menor de 12 ciclos para los atletas considerados) es medido por un contador. Si a la finalización del movimiento el atleta ha mejorado su marca, la cual está almacenada en la memoria RAM del sistema de medida, ésta se actualiza con la nueva medida y la nueva marca se presenta por el display del sistema. Al final del ciclo de medida, se activa la señal fin que informa a los médicos deportivos de la finalización del experimento.

Ej. 19 — Diseñar un sistema de control de apertura de una puerta. El sistema funcionará del siguiente modo: Inicialmente, la puerta está cerrada. Cuando la señal externa Inicio se pone a 1 se activa un reloj (señal tiempo, que cuenta 16 ciclos del reloj del sistema), de forma que el usuario

tiene 16 ciclos para introducir la clave correcta. La clave consta de 4 bits que se introducen en serie (señal clave) comenzando por el bit más significativo. El sistema comienza a leer los bits de la clave cuando se activa la señal Inicio. Cuando los 4 bits se han introducido, esta clave se compara con la almacenada en la puerta, y si las dos claves son iguales la puerta se abrirá, sino el sistema permite al usuario una nueva oportunidad, así hasta 2 veces. Si tras dos intentos, el usuario no acierta la clave la puerta se bloquea (señal Bloqueo a '1'). La puerta, por lo tanto, se bloqueará si no se acierta la clave en dos intentos o si una vez introducida la clave correcta se han superado 16 ciclos de reloj del sistema. Una vez que la puerta se bloquea permanecerá bloqueada hasta que un empleado de seguridad active la señal Desbloqueo, volviendo en este caso el sistema al estado inicial. Si la puerta se abre permanecerá abierta un ciclo de reloj y después el sistema vuelve al estado inicial.



Ej. 20 — Diseñar el camino de datos y la unidad de control de un sistema algorítmico que dado dos números enteros positivos en C2, n y m con $n \geq m$, permita calcular la suma de los números comprendidos entre n y m , ambos inclusive. Las entradas del sistema son: inicio (1 bit), n (10 bits) y m (10 bits). Las salidas son: fin (1 bit) y resultado. En el camino de datos sólo pueden usarse un sumador, un restador y un comparador con cero. ¿Qué tamaño deben tener los registros del sistema algorítmico para poder calcular todos los posibles resultados con los anchos de los datos de entrada?

```

A ← n, B ← m, T ← 0 ;
while A ≥ B do
    T ← T + A;
    A ← A - 1;
end
resultado ← T;
    
```

Ej. 21 — Un número triangular, $T(n)$, es un número obtenido sumando todos los enteros positivos de valor menor o igual que un entero dado, n . Los primeros números triangulares son por tanto 1, 1+2, 1+2+3,... Diseñar el camino de datos y la unidad de control del sistema algorítmico que dado n permita calcular el número triangular correspondiente. Las entradas del sistema son: inicio (1 bit) y n (10 bits). Las salidas son: fin (1 bit) y resultado. En el camino de datos sólo puede usarse un único sumador/restador. El número triangular $T(1023)=523776$, ¿qué tamaño deben tener los registros del sistema algorítmico?

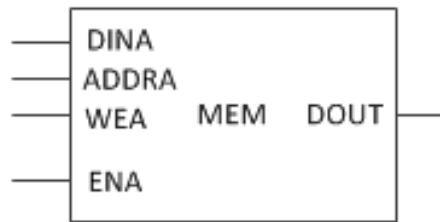
```

A ← n, T ← 0 ;
while A ≠ 0 do
    T ← T + A;
    A ← A - 1;
end
resultado ← T;
    
```

Ej. 22 — Diseñar la ruta de datos y la unidad de control (diagrama ASM y tabla de salidas) de un sistema algorítmico que dados dos números, A y B (8 bits en C2), almacenados en una memoria RAM, calcule $A-B$, almacene el resultado en la memoria y lo visualice en el puerto de salida Resultado. Finalmente se activa la señal Fin a 1 y el sistema vuelve al estado inicial. La dirección de memoria donde tiene que guardarse el resultado será la primera dirección a partir de la dirección

de B que tenga almacenado un cero. Suponer que siempre hay alguna posición libre. Las entradas del sistema son: Inicio, DirA, DirB (direcciones de memoria de A y B, respectivamente). Las salidas del sistema son: Fin y Resultado. En la ruta de datos se debe usar una memoria SRAM^(*) de 128 palabras con un único puerto de lectura/escritura, un único sumador/restador, registros y elementos combinacionales que se consideren necesarios. Se valorará el uso del hardware mínimo.

(*) La memoria SRAM será síncrona y los puertos funcionales de la memoria son los que se muestran en la siguiente figura.



Ej. 23 — Diseñar la ruta de datos y la unidad de control (diagrama ASM y tabla de salidas) de un sistema algorítmico que dados n números ($n > 1$) de 4 bits en binario puro los ordene de mayor a menor según el algoritmo de ordenamiento de burbuja.

```

 $i \leftarrow 1, Fin \leftarrow 0;$ 
while  $i \leq n$  do
     $j \leftarrow 1;$ 
    while  $j \leq n - 2$  do
        if  $M(j) \leq M(j + 1)$  then
             $M(j) \leftarrow M(j + 1);$ 
             $M(j + 1) \leftarrow M(j);$ 
        end
         $j \leftarrow j + 1;$ 
    end
     $i \leftarrow i + 1;$ 
end
 $Fin \leftarrow 1;$ 

```

Los números estarán almacenados a partir de la posición 0x00 de una memoria SRAM⁽¹⁾. El sistema comenzará a funcionar cuando se active la señal **inicio** y una vez ordenados los n números se activará la señal de salida **fin**. Las entradas del sistema son **inicio** (1 bit), **n** (8 bits, binario puro). La salida es **fin** (1 bit). En la ruta de datos se debe usar una memoria SRAM⁽¹⁾, comparadores del ancho necesario, contadores ascendentes/descendentes⁽²⁾, sumadores, registros y multiplexores.

⁽¹⁾: El alumno puede escoger el tipo de SRAM: síncrona o asíncrona y debe indicarlo de forma explícita. En cualquier caso, la memoria es de doble puerto: en cada puerto se puede realizar de forma independiente una operación de lectura o escritura. Los puertos de salida conservan el valor leído hasta que se realice una nueva lectura. La definición de puertos es: **din{a,b}**, bus de datos de entrada; **addr{a,b}**, bus de direcciones; **we{a,b}**, write enable; **en{a,b}**, señal de habilitación del puerto; **dout{a,b}**, bus de datos de salida.

⁽²⁾: La definición de puertos del contador ascendente/descendente: **din**, bus dato de entrada; **ce**, habilitación del contador; **load**, señal de carga; **ud**, cuenta arriba ($ud=1$) o cuenta abajo ($ud=0$); **dout**, bus de salida.

