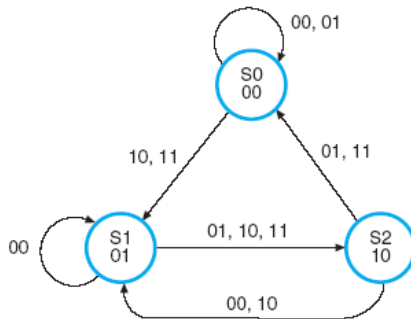
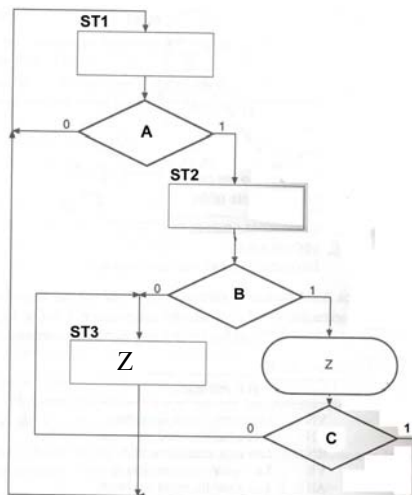




1. En la figura aparece el diagrama de estados de un circuito secuencial síncrono. Dibujar su correspondiente diagrama ASM. Utilizar el mínimo número de cajas de decisión. Las entradas al circuito son X_2 y X_1 y las salidas Z_1 y Z_2 .



2. Encontrar el valor de la salida y los estados del circuito síncrono cuyo diagrama ASM es:



ante la siguiente secuencia de entradas (suponer que el estado inicial es ST1):

A:	0	1	1	0	1	1
B:	1	1	1	1	0	0
C:	0	1	0	1	0	1

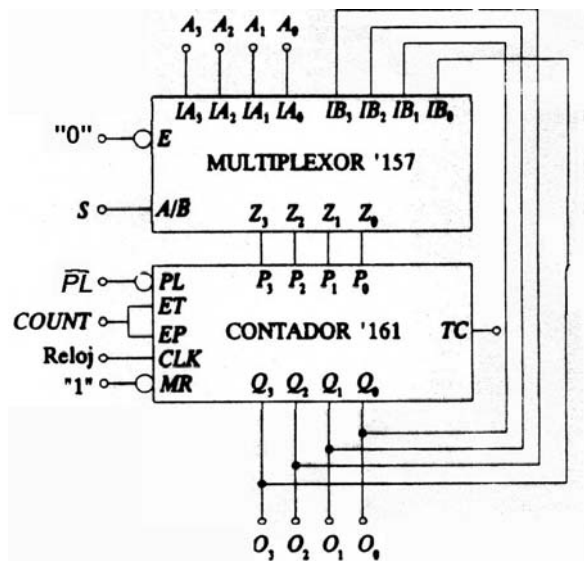
3. Encontrar el diagrama de flujo y la tabla de estados correspondiente al diagrama ASM del problema anterior.
4. Implementar el diagrama ASM del ejercicio 2 utilizando el método de 1 f-f por estado.
5. Implementar el diagrama del problema 2 utilizando un registro de secuencia y un decodificador.
6. Deducir el diagrama ASM correspondiente al siguiente autómata: Consta de 2 estados: A y B y de dos entradas: X e Y. Si nos encontramos en el estado A y la entrada X vale 0 entonces el siguiente estado es A. Si, por el contrario X=1, entonces el siguiente estado es B. Si estamos en el estado B y la entrada Y vale 1, el siguiente estado es B pero si Y=0, el siguiente estado es A. La salida Z=1 cuando el circuito se encuentra en el estado B.
7. Deducir el diagrama ASM de un circuito que detecta la diferencia entre los valores que toma la entrada X en dos ciclos consecutivos de reloj. Si X toma valores diferentes, entonces la salida Z vale 1 en el siguiente ciclo de reloj. En caso contrario Z=0.
8. Implementar el diagrama ASM del problema 7 utilizando un registro de secuencia y un decodificador.
9. Dibujar el diagrama ASM de un controlador de semáforos que funciona como sigue: una señal temporal T es la entrada al controlador. T define el intervalo de luz amarilla así como los cambios de las luces roja y verde. Las salidas de las señales se definen según la siguiente tabla:

Salida	Encendida
GN	Verde, semáforo Norte/Sur
YN	Amarilla, semáforo Norte/Sur
RN	Roja, semáforo Norte/Sur
GE	Verde, semáforo Este/Oeste
YE	Amarilla, semáforo Este/Oeste
RE	Roja, semáforo Este/Oeste

Mientras T=0 la señal verde está encendida para uno de los semáforos y en rojo para el otro. Cuando T=1, la luz amarilla se enciende en el semáforo que previamente estaba en verde mientras que el semáforo en rojo sigue en rojo. Cuando T vuelve a

0, el semáforo en amarillo se pone en rojo y el rojo en verde. Este patrón de colores continúa indefinidamente. Suponer que el controlador es síncrono con un reloj con una frecuencia mucho mayor que la de la señal T.

10. Simular en función del tiempo el proceso de multiplicar los dos números sin signo siguientes: 1001 (multiplicando) y 1010 (multiplicador) utilizando la ruta de datos vista en la teoría con $n=4$ y el control cableado también estudiado en teoría. Indicar los contenidos que van tomando los registros A, Q, P y C y los estados de control.
11. Determinar el tiempo que tarda la operación multiplicación en el sistema digital descrito en la teoría suponiendo que el registro Q tiene n bits y que el periodo de la señal de reloj es T nanosegundos (transparencia 4.5.)
12. Demostrar que la multiplicación de 2 números de n bits es un número de no más de $2n$ bits. Obsérvese que esta condición implica que no se puede producir un rebasamiento en el resultado final en el circuito multiplicador definido en la teoría.
13. Tomando como unidad de proceso el circuito de la figura, diseñar un controlador con una entrada C tal que:
 - a) Si $C=0$ el contador incrementa 2 veces la palabra $A_3 A_2 A_1 A_0$ y el resultado aparece en la salida $O_3 O_2 O_1 O_0$.
 - b) Si $C=1$ el contador rota hacia la izquierda la palabra $A_3 A_2 A_1 A_0$ y el resultado aparece en la salida $O_3 O_2 O_1 O_0$.



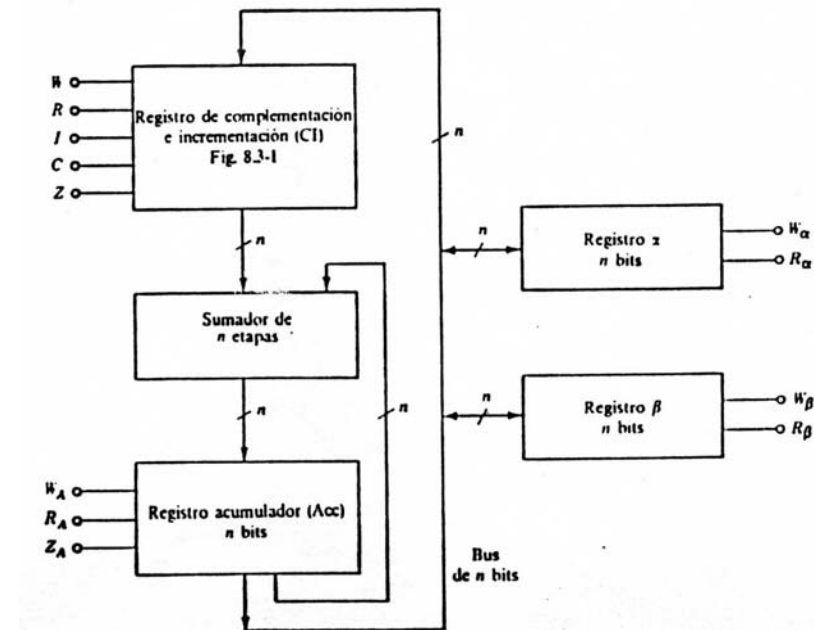
El sistema permanecerá en un estado de espera que permita la introducción de datos $A_3 A_2 A_1 A_0$, la lectura de resultados en $O_3 O_2 O_1 O_0$ y la selección de la

operación C . Al accionar un pulsador X , se pondrá en marcha la secuencia de microoperaciones hasta volver otra vez al estado de espera durante el que aparecerá en la salida $O_3 O_2 O_1 O_0$ el nuevo resultado.

14. En la figura aparece representada la ruta de datos de un sistema digital sumador/restador basado en acumulador. Inicialmente los operandos se encuentran almacenados en los registros R_α , R_β , y el resultado al final de la operación debe aparecer en R_β . El sistema digital se ha diseñado para que permita la ejecución de las cuatro operaciones siguientes en función de los valores de los bits de control $C_1 C_0$ que aparecen en la siguiente tabla:

$C_1 C_0$	Operación
00	$A+B$
01	$A-B$
10	$-A+B$
11	$-A-B$

En respuesta a los bits de operación $C_1 C_0$ y a una señal de Inicio X , la unidad de control realizará una de las operaciones de la tabla almacenándose al final el resultado en el registro R_β y quedando el sistema a la espera de otro pulso por la señal de inicio X que le indique la ejecución de una nueva operación.



El registro CI es un registro sobre el que se pueden realizar las siguientes operaciones elementales en respuesta a sus correspondientes bits de control:

Control	Operación
W	CI ← Bus
R	Sumador ← CI
I	CI ← CI + 1
C	CI ← \overline{CI}
Z	CI ← 0

El resto de los registros tienen estructuras más sencillas que sólo permiten la lectura, escritura y puesta a 0 en respuesta a sus entradas de control. Las salidas del registro acumulador A_{CC} están directamente conectadas a una de las entradas del sumador y su terminal de lectura sirve para que su contenido acceda al bus de n bits al que pueden acceder también el resto de los registros.

Realizar el diseño completo de este sistema digital siguiendo las siguientes pautas:

- Describir a nivel de puertas lógicas la estructura de cada uno de los registros: R_α , R_β , A_{CC} , CI.
- Dibujar el diagrama ASM de la unidad de control del sistema.
- Implementar la unidad de control del sistema utilizando el diseño de 1 f-f por estado.

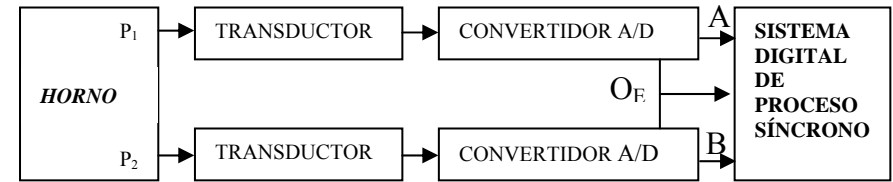
15. Diseñar un sistema secuencial de proceso que reciba en sus entradas A3 A2 A1 A0 información de un dígito en BCD y una orden de operación constituido por un cambio de "0" a "1" de la señal O_e (inicializador). Al producirse dicha orden, el sistema ha de realizar simultáneamente o en secuencia, según convenga las operaciones elementales siguientes:

- 1) Almacenamiento en paralelo de la información presente en las entradas A0 a A3.
- 2) Comparación del número almacenado con otro número preprogramado en el sistema e indicación mediante una salida llamada Z1 del resultado de dicha comparación. $Z1 = "1"$ si $A >$ número preprogramado. Z1 permanecerá a "1" hasta que se realice una nueva operación (indicada por un nuevo cambio en O_e de "0" a "1").
- 3) Salida en serie del número almacenado A3 A2 A1 A0 a través de otra salida, llamada Z2 si el número almacenado es menor que el preprogramado e inhibición de dicha salida en caso contrario. Realizadas estas operaciones el sistema quedará preparado para recibir una nueva orden de operación.

Diseñar la unidad operativa utilizando un registro de desplazamiento con entradas en paralelo y salida en serie, un comparador '85, un biestable R-S para implementar la salida Z1. ¿Cuál será la estructura interna del registro?

Diseñar la unidad de control utilizando el mínimo número de flip-flops tipo D y un decodificador y las puertas OR que sean necesarias para implementar la parte combinacional.

16. En la figura se representa el diagrama de bloques de un procesador industrial:



Mediante sendos transductores, la presión en los puntos P_1 y P_2 del horno se convierte en una magnitud eléctrica analógica que es transformada en digital por medio de dos convertidores analógico-digitales. La salida de estos convertidores es un número positivo codificado en binario natural de 8 bits. Una vez que la información procedente de los dos convertidores está disponible, se produce un pulso de sincronismo en la línea O_E . Esta señal indica al sistema digital de proceso que comience a realizar las siguientes operaciones:

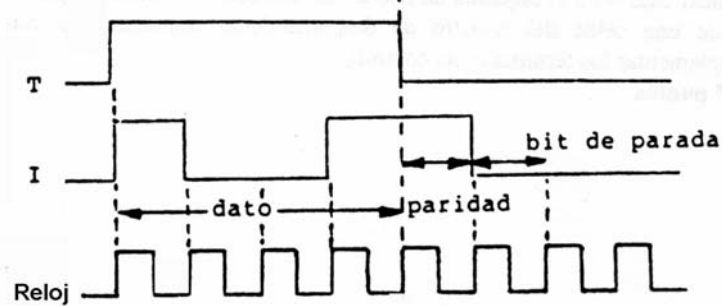
- a) Memorizar la información procedente de ambos convertidores en sendos registros R_1 y R_2 .
- b) Comparar con un número preprogramado N la información mayor de ambos convertidores. Si el resultado de esta comparación indica que el número comparado con N es mayor que él, se activará una alarma y se dará por terminado el proceso. La alarma permanecerá activa hasta que se verifique que $N > \max(R_1, R_2)$.
- c) Si se comparó el contenido de R_1 con N, se realizará la resta del contenido de R_2 del de R_1 ($R_1 - R_2$) y, si se comparó el contenido de R_2 con N, se realizará la resta del contenido de R_1 del de R_2 ($R_2 - R_1$). El resultado de ambas operaciones se memorizará.
- d) La diferencia obtenida en el caso anterior se comparará con el valor máximo de todas las diferencias obtenidas en sucesivos procesos y se reemplazará por ella en el caso de que sea mayor. A continuación se dará por terminada la secuencia y el sistema quedará a la espera de recibir nuevas órdenes de proceso.

Diseñar el sistema digital de proceso. La unidad operativa utilizará un único restador de 8 bits con el convenio de complemento a dos.

Nota: Considerar, como es habitual en estos problemas que el tiempo que transcurre entre pulsos sucesivos en la señal O_E es mucho mayor que el periodo de la señal de reloj con la que funciona el sistema.

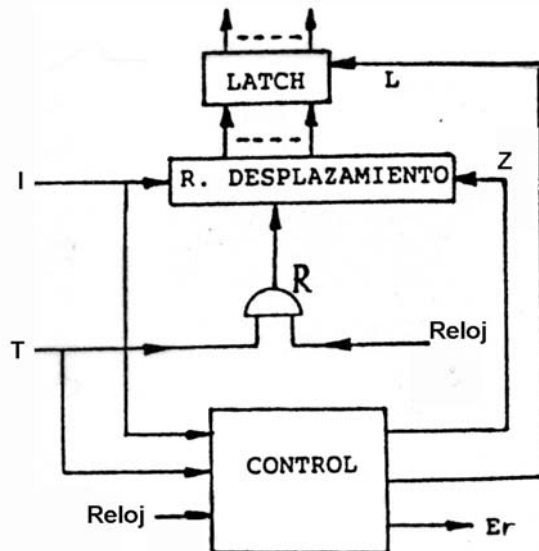
17. En cierto dispositivo se reciben datos en modo serie, de longitud variable. Una señal de control indica si se está enviando un dato ($T=1$) o no ($T=0$). Inmediatamente después del último bit del dato y con la señal de control puesta ya a 0, se recibe un bit adicional para control de paridad (paridad IMPAR, bit $p=1$ si el número de unos es par) y por último inmediatamente después del bit de paridad tanto la línea T como

la de datos I se mantienen a 0 durante al menos un ciclo de reloj más (bit de parada) hasta que aparezca un nuevo dato (T=1). Ver figura



La recepción de los datos se realiza mediante el circuito de la figura siguiente. Está formado por:

- Un registro de desplazamiento con entrada de datos serie y salida en paralelo. Posee un terminal para la señal de reloj (R) y otro síncrono para la puesta a cero (Z). La longitud de dicho registro es mayor que la máxima longitud de palabra de los datos que se transmiten en serie.
- Un registro constituido por biestables (Latches) para recoger los datos una vez que se ha completado su recepción. El terminal de control asíncrono (L) nos permite escribir una palabra en el Latch.
- Un circuito secuencial síncrono (Moore) que controla el proceso. Todos los terminales son activos en alta.



El funcionamiento es el siguiente: una vez que se ha completado la carga del dato en el registro de desplazamiento, si la paridad es correcta se transfiere al Latch, en el caso contrario, se envía un "1" por la salida de Error (Er) durante un ciclo de reloj. A continuación, en cualquier caso, se limpia el registro de desplazamiento (todas las celdas a cero) y se espera la recepción del siguiente dato.

- Diseñar el controlador utilizando una memoria ROM y el mínimo número de Flip-flops D. Indicar detalladamente cuál debe ser el tamaño de la memoria así como sus contenidos.
 - Indicar cual sería el esquema de puertas de una celda del Latch implementada con biestables R-S y de una celda del registro de desplazamiento implementada con Flip-flops R-S (como implementar los terminales de control).
18. Diseñar un sistema digital con 3 registros de 16 bits que llamaremos AR, BR y CR y una línea de entrada IN de 16 bits que realice secuencialmente las siguientes operaciones sobre números escritos en representación de complemento a dos. Ignorar las situaciones de rebasamiento. Operaciones:
- Transferir 2 números de 16 bits del bus de entrada IN a AR y BR en ciclos de reloj sucesivos tras poner una señal G a 1.
 - Si el número en AR es positivo pero distinto de cero, multiplicar dicho número por 2 y transferir el resultado al registro CR.
 - Si el número en AR es negativo, multiplicar el contenido de BR por 2 y transferir el contenido a CR.
 - Si el número en AR es cero, borrar el contenido de CR.

19. Diseñar un sistema digital que multiplique dos números escritos en binario natural por el método de sumas sucesivas. Por ejemplo, para multiplicar 5 por 4 el sistema digital sumaría el multiplicando (5) las veces que le diga el multiplicador (4): $5+5+5+5=20$. Suponer que el multiplicando se encuentra en un registro BR, el multiplicador en AR y el producto en PR. Un circuito sumador sumará el contenido de BR y lo almacenará en PR. Un circuito de detección de cero Z comprueba el valor de AR cada vez que éste se decremента. Diseñar la unidad de control por el método de un f-f por estado.

20. Deducir el diagrama ASM de un circuito síncrono con un reloj CLK que sirve para controlar el funcionamiento de una lavadora. El circuito consta de 3 señales de entrada externas que son START, FULL y EMPTY que verifican que tienen una duración en 1 de al menos un ciclo de reloj y que son mutuamente excluyentes (no pueden valer 1 a la vez), y 4 salidas externas HOT; COLD; DRAIN y TURN.

La ruta de datos del sistema que controla el funcionamiento de la lavadora consiste en un contador decreciente con 3 entradas de control RESET, DEC y LOAD. El contador se decremента cada minuto cuando DEC=1 pero además se puede poner en reset síncronamente en cualquier flanco activo del reloj CLK en respuesta a su

entrada RESET. El contador tiene una única salida ZERO que vale 1 siempre que el contador esté en el estado 0 y vale 0 en caso contrario.

El circuito funciona como sigue: la lavadora pasa por 4 ciclos distintos: WASH (lavado), SPIN (centrifugado), RINSE (aclarado) y SPIN de nuevo. A continuación detallaremos cada uno de los ciclos:

- WASH – Suponemos que al encender la lavadora, el autómata se encuentra en un estado IDLE. Al colocar START a 1 durante 1 ciclo de reloj, HOT se pone a 1 y ahí permanece hasta que FULL=1 lo que indica que la lavadora está llena de agua. A continuación, activando LOAD, el contador se carga con un valor que proviene del panel de control que indica cuantos minutos queremos que dure el ciclo de lavado. DEC y TURN se ponen a 1 y la lavadora lava. Cuando la salida del contador ZERO = 1 el lavado ha terminado y las señales TURN y DEC pasan a valer 0.
- SPIN – A continuación, DRAIN=1 y comienza el desagüe. La señal EMPTY=1 indica que ya no queda agua en la lavadora y entonces se carga el contador con un 7. DEC y TURN se ponen a 1 y la lavadora empieza a centrifugar. Cuando Zero se pone a 1, DRAIN, DEC y TURN vuelven a 0.
- RINSE – Después, COLD=1 y en este valor permanece hasta que FULL=1 que indica de nuevo que la lavadora se ha llenado de agua, esta vez fría para aclarar. Utilizando LOAD se carga el contador con un 10. DEC y TURN se ponen a 1 y la lavadora aclara. Cuando ZERO=1 el aclarado está completo y TURN y DEC vuelven a 0.
- SPIN – Por último, DRAIN=1 y se desagua el agua del aclarado. Después al tomar EMPTY el valor 1 se carga el contador con 8. DEC y TURN se ponen a 1 y comienza el centrifugado. Cuando ZERO=1, DRAIN, DEC y TURN vuelven a 0 y el circuito vuelve al estado IDLE.

- a) Dibujar el diagrama ASM del circuito controlador de la lavadora.
- b) Modificar el diseño del apartado a) suponiendo que hay 2 entradas adicionales llamadas PAUSE y STOP. PAUSE provoca que el circuito, incluido el contador se pare y que todas las salidas se pongan a 0. Cuando se pulsa START de nuevo, la lavadora reinicia la operación en el punto que lo dejó. Por otro lado, cuando se pulsa STOP todas las salidas se colocan a 0 excepto DRAIN que se pone a 1. Cuando EMPTY pasa a 1 la lavadora vuelve a IDLE.

21. Suponer que disponemos de una unidad de control microprogramado similar a la estudiada en teoría para el multiplicador (figura 4.18, transparencia 4.19) (con un registro CAR + memoria control + 2 mux) excepto que el multiplexor MUX2 que selecciona una entrada de control condicional (viene de la ruta de datos) tiene 15 entradas de los bits de estado en vez de sólo 2; Además la memoria de control consta

de 1024 palabras. Rediseñar en este caso el formato de microinstrucción y especificar el número de bits en cada campo. ¿Cuántos bits se necesitan en cada palabra de la memoria de control? ¿Cuál será la capacidad total de la memoria de control?

22. Un ordenador maneja instrucciones de 32 bits divididas en los siguientes campos:

- Código de operación – 6 bits
- campo de registros – 2 de 6 bits cada uno
- campo de registro u operando inmediato 14 bits.

Contestar a las siguientes preguntas:

- a) ¿Cuál es el máximo número de operaciones que se pueden especificar?
- b) ¿Cuántos registros se pueden direccionar?
- c) ¿Cuál es el rango de operandos inmediatos sin signo que se pueden manejar?
- d) ¿Cuál es el rango de operandos con signo que se pueden manejar suponiendo que el bit 13 es el bit de signo?

23. Un sistema digital consta de una unidad de memoria con 24 bits por palabra. El conjunto de instrucciones consiste en 150 operaciones diferentes. Sólo hay un tipo de formato de instrucción con un código de operación y un operando inmediato. Cada instrucción se almacena en una única palabra de memoria.

- a) ¿Cuántos bits se necesitan para el código de instrucción?
- b) ¿Cuántos bits se dejan para el campo de operando inmediato?
- c) Si el operando inmediato se interpreta como una dirección de memoria sin signo ¿Cuál es el máximo número de palabras que se puede direccionar?
- d) ¿Cuáles son el mayor y menor valores algebraicos en complemento a dos que se pueden representar en el operando inmediato?

24. Se quieren ejecutar las 5 instrucciones que aparecen en la tabla descritas como transferencias entre registros en el ordenador de ciclo sencillo que hemos visto en la teoría.

- a) Completar la tabla indicando la microinstrucción o palabra de control (salidas del decodificador de instrucciones) que ejecutan cada una de las instrucciones. Utilizar para ello las tablas del tema anterior en las que aparecen los códigos de operación de la ruta de datos. Si algún campo no se utiliza dejarlo en blanco.

Instrucción	DA	AA	BA	MB	FS	MD	RW	MW	PL	JB
$R[1] \leftarrow R[7] \oplus R[4]$										
$R[5] \leftarrow M[R[6]]$										
$R[3] \leftarrow R[4] + 2$										
$R[5] \leftarrow sR[6]$										
$R[4] \leftarrow R[7]$										

- b) Completar la siguiente tabla, dando la instrucción para la ejecución de cada una de las microinstrucciones definidas como transferencia de registros. Utilizar las tablas correspondientes para determinar el código de operación. Si algún campo no se utiliza, dejarlo con el valor 0.

Instrucción	Código Operación	DR	SA	SB u Op. inmediato
$R[0] \leftarrow R[6] \oplus R[7]$				
$R[7] \leftarrow M[R[6]]$				
$R[7] \leftarrow R[7] + 2$				
$R[7] \leftarrow s!R[7]$				
$R[6] \leftarrow R[7]$				

25. Simular manualmente el ordenador de ciclo sencillo que hemos estudiado en la teoría, ejecutando la siguiente secuencia de instrucciones, suponiendo que en cada registro inicialmente se encuentra contenido un número igual a su subíndice (por ejemplo R0 contiene 0, R1 contiene 1, etc...):

```

ADD R0, R1, R2
ADD R3, R4, R5
ADD R6, R7, R0
ADD R0, R0, R3
ADD R0, R0, R6
ST R7, R0
LD R6, R7
ADI R0, R6, 0
ADI R3, R6, 3

```

26. Una microinstrucción almacenada en la dirección 35 de la memoria de control del micro de ciclos múltiples realiza la operación:

$$R1 \leftarrow R1 + R2, CAR \leftarrow CAR + 1$$

Suponiendo que DR = SA = 1, SB = 2, indicar cómo es la microinstrucción escrita en forma simbólica.

27. Escribir las microinstrucciones en modo simbólico y binario de las siguientes instrucciones entre registros (las direcciones del CAR están escritas en hexadecimal):

- $R3 \leftarrow R1 - R2, CAR \leftarrow 17$. Suponiendo que en la instrucción DR=3, SA=1, SB=2
- $R5 \leftarrow sr R5, CAR \leftarrow CAR + 1$ suponiendo que DR=5, SB=5
- If (Z=0) entonces (CAR←21) else (CAR←CAR+1)
- $R6 \leftarrow R6, C=0, CAR \leftarrow CAR + 1$ suponiendo DR=SA=6

28. Escribir un programa para la CPU de ciclos múltiples que calcule el valor entero promedio de los 4 números binarios de 8 bits sin signo almacenados en los registros R1, R2, R3 y R4. El valor promedio debe almacenarse en el registro R5. No alterar el contenido de ningún otro registro.

29. Escribir un programa para la CPU de ciclos múltiples que compare dos números binarios sin signo almacenados en los registros R[SA] y R[SB]. El registro que contiene el número pequeño se resetea. Si los dos números son iguales, entonces debe resetearse tanto un registro como el otro. ¿Es posible implementar el programa con una única instrucción?

30. Especificar un formato de instrucción para el ordenador microprogramado que realiza la operación:

$$R[DR] \leftarrow R[SB] + M[SA]$$

Dibujar el diagrama ASM que implementa esta instrucción, suponiendo que su código de operación es 0000111. Indicar la secuencia de microoperaciones que ejecutan esta instrucción, incluyendo los valores de CAR.

31. En el ordenador de ciclo sencillo estudiado en la teoría se pueden ejecutar bifurcaciones de acuerdo con los valores de \bar{V} y \bar{N} . Sin embargo, en el control microprogramado de ciclo múltiple no están contemplados estos casos. Modificar el control microprogramado para permitir la toma de decisiones según el valor de estos dos bits de estado. Incrementar para ello el tamaño del multiplexor MUXS y hacer los cambios adicionales que consideres oportunos.

32. Diseñar una unidad de control cableada basada en decodificadores que incluya la instrucción de desplazamiento a la derecha múltiple siguiendo los pasos que se describen a continuación:

- Modificar el esquema ASM de la unidad de control cableada vista en teoría para incluir la ejecución de SRM.
- Modificar el diagrama de bloques de la unidad de control basada en decodificadores teniendo en cuenta el esquema ASM diseñado en el apartado anterior.
- Escribir todas las expresiones lógicas de control para el nuevo diseño.