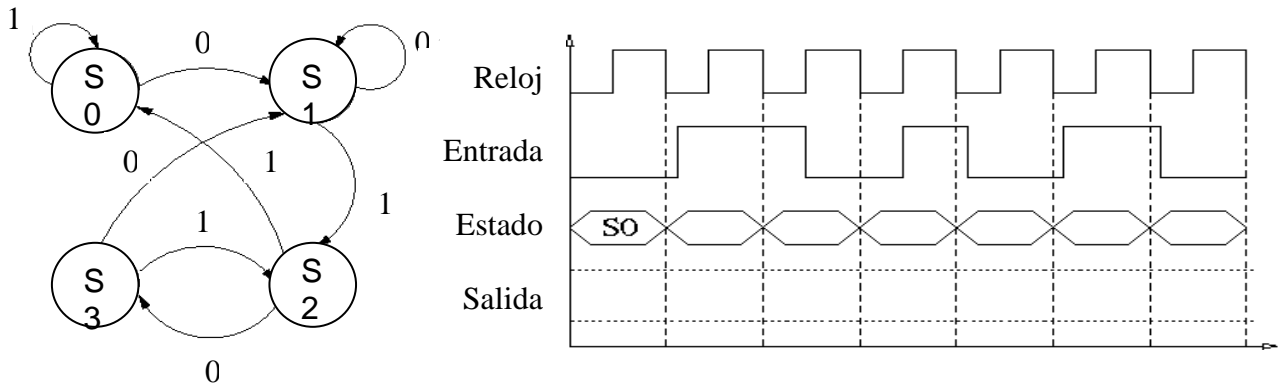
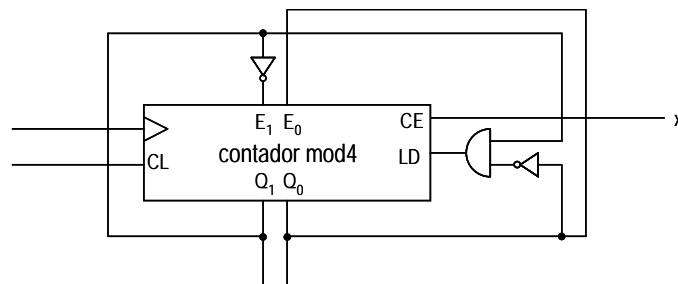


Problemas del tema 5

1. Considere el siguiente sistema secuencial especificado mediante su diagrama de estados y complete el siguiente cronograma. Considere despreciables los retardos de cambio de estado y de salida.



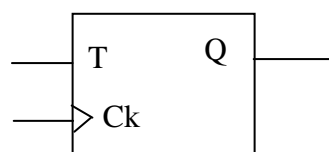
2. Usando la hoja adjunta, simule el comportamiento del circuito secuencial mostrado en la figura. En la simulación tenga en cuenta los retardos combinatoriales conociendo que el retardo de una puerta se corresponde con una división vertical.



3. Usando un biestable D disparado por flanco y las puertas necesarias, implemente un circuito secuencial cuyo comportamiento coincida con el de la tabla siguiente

J(t)	K(t)	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

4. El denominado biestable T es un tipo de biestable cuyo esquema y comportamiento viene dado por la siguiente figura y la siguiente tabla, respectivamente:



T(t)	Q(t+1)
0	Q(t)
1	$\overline{Q(t)}$

Usando un biestable D disparado por flanco y las puertas necesarias, implemente un circuito secuencial cuyo comportamiento coincida con el de un biestable T.

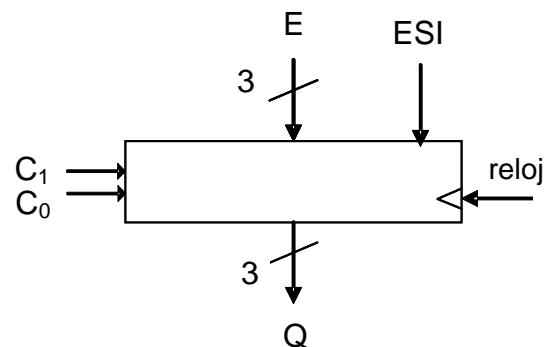
5. Utilizando biestables y la menor lógica combinacional posible, implementa un sistema secuencial tal que cuando la señal de entrada C valga 1 siga la secuencia de estados: 0, 1, 3, 2, 6, 7, 5, 4, 0... Cuando la señal C vale 0, el sistema permanece en el mismo estado.

6. Implemente usando biestables y un único módulo ROM el sistema secuencial cuyas funciones de transición de estados y de salida vienen dadas por la siguiente tabla:

Estado actual	Salida z1 z0	Entrada x1 x0			
		00	01	10	11
S0	01	S0	S0	S1	S2
S1	11	S0	S1	S2	S1
S2	10	S2	S2	S1	S3
S3	11	S0	S0	S0	S3

7. Usando biestables D y módulos combinacionales (multiplexores, sumadores ...) implementa un sistema secuencial síncrono que sea capaz de realizar las siguientes funciones:

C ₁ C ₀	FUNCIÓN	Q(t+1)
00	Carga en paralelo	E(t)
01	Rota a la derecha	(Q ₀ , Q ₂ , Q ₁)(t)
10	Desplaza a la izquierda	(Q ₁ , Q ₀ , ESI)(t)
11	Incrementa en uno	Q(t) + 1



8. Diseña un registro con capacidad para cargar en paralelo desde dos entradas diferentes, así como rotar una posición a la derecha su contenido. El registro tendrá dos señales de control, S1 y S0, que gobiernan su funcionamiento según indica la siguiente tabla

S1	S0	Operación
0	0	Nada
0	1	Carga la entrada 1
1	0	Carga la entrada 2
1	1	Rota el contenido una posición a la derecha

Nota: la operación de rotar a la derecha es equivalente a un desplazamiento donde el bit menos significativo que se pierde por la derecha vuelve a entrar por la izquierda del registro.

9. Diseñe un registro de desplazamiento de 3 bits. El registro tendrá dos señales de control, S1 y S0, que gobiernan su funcionamiento según indica la siguiente tabla:

S1	S0	Operación
0	0	Carga paralelo
0	1	División entre 2
1	0	Multiplicación por 2 (módulo 8)
1	1	Complemento a 1

10. Diseñe mediante biestables y multiplexores un registro de desplazamiento (3 bits) con entrada serie con las siguientes características:

control	
00	No desplaza
01	Desplaza derecha
10	Desplaza izquierda
11	Clear

11. Diseñe un registro de 5 bits que puede almacenar valores procedentes de 3 entradas diferentes. El registro tendrá 2 señales de control, S1 y S0 que gobiernan su funcionamiento según la tabla siguiente:

S1	S0	Operación
0	0	Nada
0	1	Carga la entrada 1
1	0	Carga la entrada 2
1	1	Carga la entrada 3

12. Diseñar un registro de desplazamiento bidireccional de 3 bits con dos entradas en paralelo E1 y E2. El registro tendrá una entrada de control S que indica la función a realizar según la siguiente tabla:

S	FUNCIÓN
0	$Q(t+1)=Q(t)$
1	$Q(t+1)=Q(t)$ desplazada 1 bit a la izquierda
2	$Q(t+1)=Q(t)$ desplazada 1 bit a la derecha
3	$Q(t+1)=\text{not } Q(t)$
4	$Q(t+1)=E1(t)$
5	$Q(t+1)=E2(t)$
6	$Q(t+1)=Q(t)\text{and}E2(t)$

13. Diseñe un registro de 3 bits con capacidad de desplazamiento a la derecha, desplazamiento a la izquierda, complemento lógico y carga en paralelo usando multiplexores y biestables. El registro tendrá una entrada de control de 2 bits que indicará la función a realizar. Además tendrá entrada para la carga paralela y entradas serie.

14. Usando puertas lógicas y contadores módulo 16 con capacitación de cuenta y carga paralela conectados a un reloj de 60 Hz, diseñe un temporizador de 1 segundo (su salida vale 1 durante un ciclo de cada 60).

15. Se desea diseñar un sistema secuencial síncrono que en cada ciclo de reloj recibe una entrada $X \in \{\text{Norte, Sur, Este, Oeste}\}$. El sistema posee, además, una salida $Z \in \{0,1\}$. La salida Z tomará el valor 1 si tras 3 ciclos de reloj se ha detectado una de las siguientes subsecuencias:

$$X(t-2), X(t-1), X(t) = \begin{cases} \text{Norte, Este, Este} \\ \text{ó} \\ \text{Sur, Este, Este} \end{cases}.$$

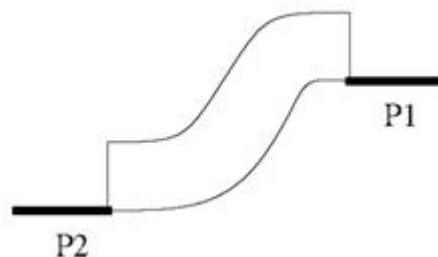
En todos los demás casos el valor de Z será 0. Se pide:

- Construir el diagrama de estados del sistema en la forma de una máquina Mealy. Explique el significado de cada estado.
- Implemente el sistema con el menor número posible de biestables D y puertas lógicas.

16. Se desea diseñar el sistema de control de una escalera mecánica bidireccional. Para ello, disponemos de dos sensores de presión, P1 y P2, como se muestra en la figura. Cuando se activa un sensor, la escalera empieza a andar en dirección al otro sensor y no para hasta que dicho sensor se active. No se considera la situación de que se coloquen de forma simultánea dos personas sobre P1 y P2 porque una vez activado el sensor P1, éste se apaga y no vuelve a encenderse hasta que se ha activado P2, y viceversa. Las situaciones que nos podemos encontrar son las siguientes:

- La escalera está parada.
- La escalera está subiendo.
- La escalera está bajando.

El sistema tiene dos salidas: encendido E ($E=1$ cuando la escalera se mueve) y dirección S ($S=1$ cuando la escalera sube y $S=0$ cuando la escalera baja).



- Diseñe el sistema secuencial Moore que muestre el comportamiento de la dirección de los motores, así como su estado (encendido o apagado).
- Implemente el sistema mediante biestables y una memoria ROM.

17. Un sistema secuencial tiene una entrada X de 2 bits y dos salidas, Z y M. La salida $Z = 1$ indica que el valor actual de la entrada, $X(t)$, es igual al valor anterior de la entrada, $X(t-1)$. Cuando el valor actual de X no coincide con el valor anterior de X (es decir, cuando $Z = 0$) la salida $M = 1$ si el valor actual de X es mayor que el anterior y $M = 0$ si es menor.

- Obtenga una especificación del sistema como máquina de Mealy (en forma de diagrama de estados o de tabla).
- Implemente el sistema usando biestables D y puertas lógicas.

18. Diseñar un circuito tipo Mealy que genere como salida dos dígitos BCD que siguen indefinidamente la siguiente secuencia: 10, 12, 11, 12, 10, 12, 11, 12 ...

El circuito cuenta con una entrada A/D tal que, si A/D vale cero la secuencia se recorre en orden ascendente, mientras que si A/D vale uno, se recorre en orden descendente. Se debe utilizar el mínimo número de biestables. El circuito combinacional que genera la salida debe implementarse utilizando únicamente multiplexores 2 a 1 y 4 a 1. El circuito combinacional que calcula el estado siguiente debe implementarse con una memoria ROM.

19. Diseñe un reconocedor de secuencias en la forma de una máquina de Mealy que responda a las siguientes especificaciones:

- Tiene una entrada X de tres bits (x_2, x_1, x_0) por la que cada ciclo de reloj llega un dígito decimal del conjunto $\{0,1,\dots,7\}$.
- La salida toma el valor 1 siempre que los últimos 4 dígitos recibidos ($X(t-3), X(t-2), X(t-1), X(t)$) formen la secuencia (0,3,4,7).

Se pueden usar biestables y puertas lógicas.

20. Diseñe un reconocedor de secuencias en la forma de una máquina de Mealy que posee una entrada $X \in \{0,1,2,3\}$ y una salida $Z \in \{0,1\}$. La salida Z(t) tomará el valor:

- | | |
|---|---|
| 1 | si $(X(t-3), X(t-2), X(t-1), X(t)) = (3,2,3,0)$ |
| 1 | si $(X(t-3), X(t-2), X(t-1), X(t)) = (3,0,2,1)$ |
| 0 | en cualquier otro caso |

21. Considere el diseño de un sistema secuencial con una entrada binaria X y una salida binaria Z. La salida en el tiempo t es 1 si $X(t-3,t) = 0101$ ó 0110 . Implemente el sistema utilizando:

- Biestables D y el mínimo número de puertas lógicas.
- Biestables D y una ROM.
- Un registro de desplazamiento de 3 bits con entrada serie/salida paralelo y puertas lógicas.

22. Un sistema secuencial tiene una entrada X y una salida Z, ambas de un bit. Tiene el siguiente comportamiento:

$$Z = \begin{cases} 1 & \text{si } X(t-3,\dots,t) = 0111 \\ 0 & \text{en caso contrario} \end{cases}$$

Se pide:

- Diseñe el sistema como máquina de Mealy usando biestables D y puertas lógicas.
- Diseñe el sistema usando un contador y puertas lógicas.

23. Un sistema secuencial posee una entrada $X \in \{0,1\}$ y una salida $Z \in \{0,1\}$. La salida tomará el valor 1 si y solo si la secuencia de entradas contiene un número impar de ceros y un número par de unos. Se pide:

- Especifique el sistema como una máquina de Moore, mediante un diagrama de estados.
- Implemente dicha especificación mediante biestables D y una ROM.

24. Se desea realizar un reconocedor de secuencias que sea capaz de reconocer de manera solapada las siguientes secuencias "101" y "100".

- c) Diseñar el circuito como máquina de Mealy utilizando multiplexores 2 a 1 y los biestables y puertas lógicas necesarias.
- d) Diseñar el circuito como máquina de Moore utilizando memorias ROM de tamaño mínimo para la salida y el cambio de estado, y los biestables necesarios.

25. Un sistema secuencial síncrono tiene una entrada $X \in \{a,b,c\}$ y una salida $Z \in \{s,n\}$. La salida Z viene dada por:

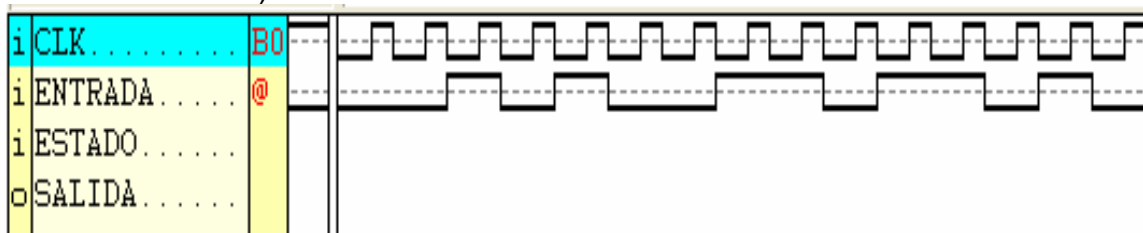
$$Z(t) = \begin{cases} s & \text{si } X(t-3) = b, X(t-2) = b, X(t-1) = c \\ n & \text{en caso contrario} \end{cases}$$

Se pide:

- a) Dibujar el diagrama de estados del sistema
- b) Implementar el sistema utilizando el menor número posible de biestables D y puertas lógicas

26. Para un reconocedor de secuencias con solapamiento cuya salida valga 1 cuando reconozca la secuencia 010:

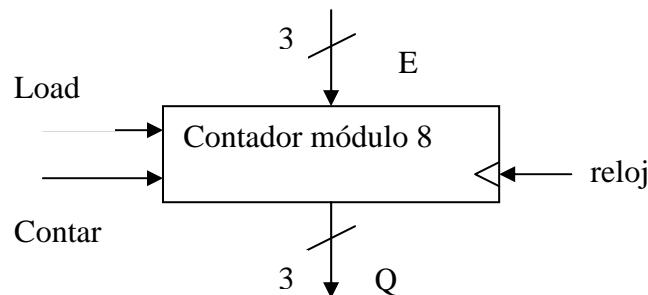
- Definir las entradas y salidas del sistema
- Dibujar el diagrama de estados como máquina de Mealy
- Escribir la codificación de estados
- Escribir las tablas de transición de estados y de salida
- Completar el siguiente cronograma (debe ser coherente con la máquina de estados):



27. Utilizando un contador módulo-16 y la menor cantidad posible de lógica combinacional, implemente:

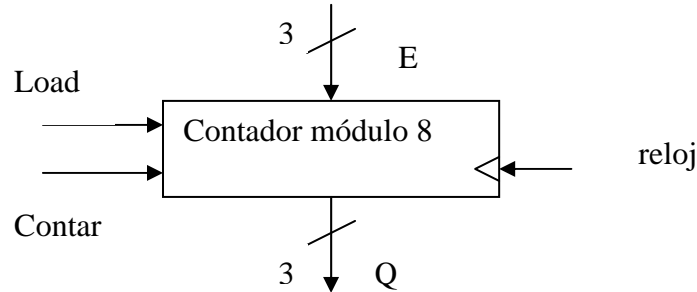
- a) Un sistema secuencial cuya salida sigue la secuencia: 4, 5, 6, 7, 8, 9, 10, 11, 4, 5...
- b) Un sistema secuencial cuya salida sigue la secuencia: 0, 1, 2, 2, 3, 3, 4, 5, 6, 7, 0, 1...

28. Utilizando un contador módulo 8, como el que se muestra en la figura, y el mínimo número de puertas lógicas, implemente un sistema secuencial cuya salida siga la secuencia: 0, 2, 2, 4, 6, 6, 0, 0,

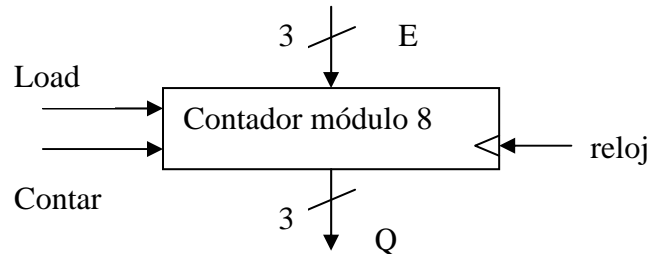


29. Utilizando un contador módulo 8, como el que se muestra en la figura, y el mínimo número de puertas lógicas, implemente:

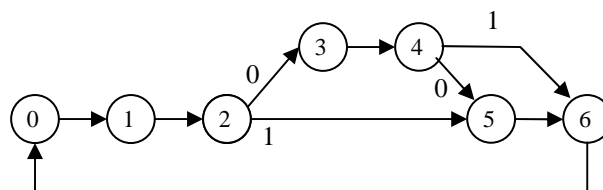
- a) Un sistema secuencial cuya salida siga la secuencia: 0,1, 4, 5, 6, 0, ...
- b) Un sistema secuencial cuya salida siga la secuencia: 0, 1, 4, 5, 6, 6, 0, ...



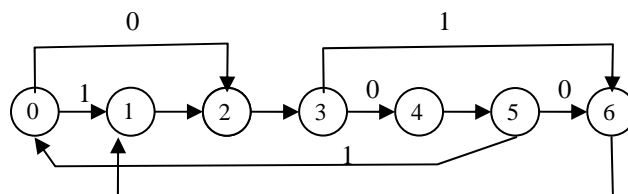
30. Utilizando un contador módulo 8, como el que se muestra en la figura, y el mínimo número de puertas lógicas, implemente un sistema secuencial cuya salida siga la secuencia: 0, 1, 4, 4, 7, 7, 0...



31. Utilizando un contador con carga en paralelo modulo 8 y puertas lógicas implementar un sistema secuencial que tiene una entrada X de 1 bit y que se comporta como el diagrama de la figura. Utilizar el mínimo de lógica combinacional posible.



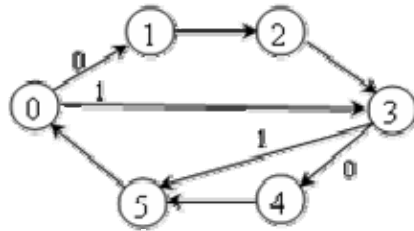
32. Utilizando un contador con carga en paralelo modulo 8 para implementar un sistema secuencial que tiene una entrada x de 1 bit y que se comporta como el diagrama de la figura.



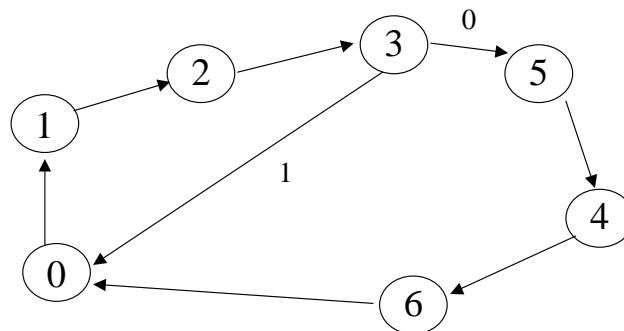
El contador tiene una señal load de carga en paralelo y una señal de cont que capacita la cuenta. Cuando la señal load está a 1 se realiza la carga paralela independientemente del valor de la señal cont. Sabiendo que E2,E1, E0 son los bits que codifican el siguiente estado:

- a) Hallar la Tabla de verdad del sistema
- b) Implementar la señal de load y cont simplificando el circuito aplicando mapas de Karnaugh
- c) Implementar E2 utilizando mux de 4 señales de control
- d) Implementar E1 utilizando decodificadores de tamaño mínimo y puertas or
- e) Implementar E0 utilizando rom

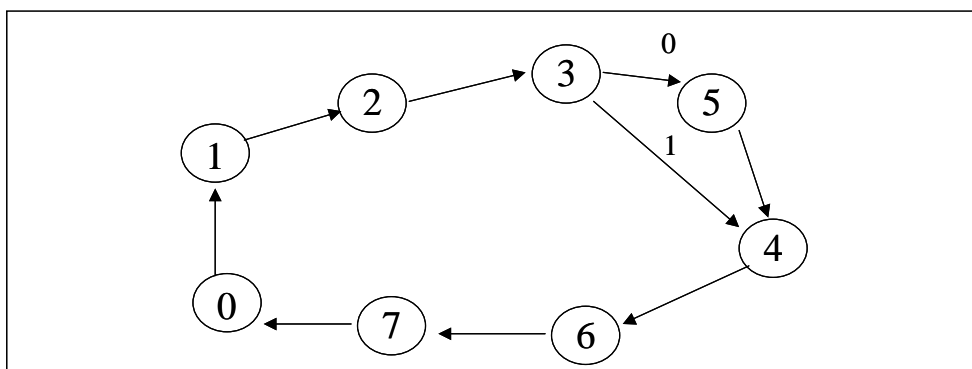
33. Utilizando un contador con carga en paralelo implemente un sistema secuencial con una entrada X que se comporte de acuerdo con el diagrama de la figura. Además del contador, utilice el menor número posible de puertas lógicas.



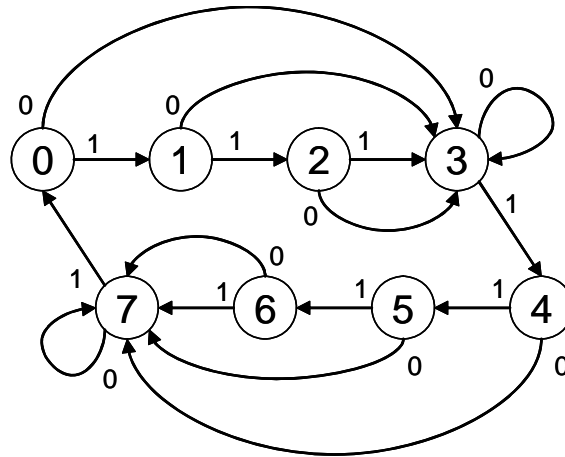
34. Utilizando un contador con carga en paralelo implemente un sistema secuencial con una entrada X que se comporte de acuerdo con el diagrama de la figura siguiente. Además del contador, se desea usar la mínima cantidad de lógica combinacional posible.



35. Utilizando un contador con carga en paralelo implemente un sistema secuencial con una entrada X que se comporte de acuerdo con el diagrama de la figura siguiente. Además del contador, se desea usar la mínima cantidad de lógica combinacional posible.



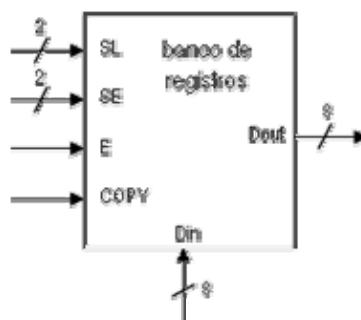
36. Usando la mínima cantidad de lógica combinacional y un contador módulo 8 con señales activas a 1 para capacitación de cuenta ascendente, CE, carga paralela, LD, e inicialización asíncrona, CL (supóngase que la señal de carga es prioritaria a la de cuenta); implementa un sistema secuencial que según el valor de una entrada, X, se comporte según el siguiente diagrama de estados:



37. Diseña un contador módulo 12 a partir de uno módulo 16, con el menor número de puertas posibles.

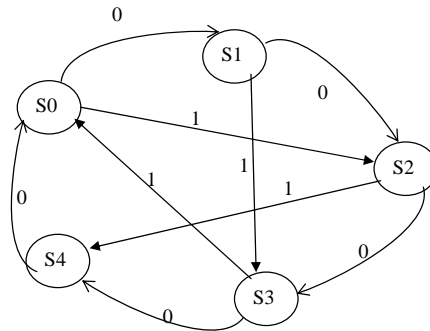
38. A partir de contadores módulo 8 con carga en paralelo y la lógica combinacional que considere necesaria, diseñe un contador módulo 12.

39. Se desea ampliar la funcionalidad de un banco de registros mostrado en la figura añadiéndole una nueva señal de control que llamaremos copy. Cuando copy vale 0 el circuito se comporta normalmente, y cuando vale 1 copia el contenido del registro indicado por SL en el registro indicado por SE ignorando el valor presente en Din. Diseñese el circuito propuesto partiendo del diseño de un banco de registros convencional.



40. Imaginemos una máquina que introduce golosinas en bolsas de tal forma que cada bolsa tiene que incluir 5 golosinas y estas pueden ser de dos tipos. Una de ellas más simples y baratas y otras más elaboradas que cuentan como dos simples (ya que duplican el precio de las primeras). Una entrada externa nos dirá si la golosina introducida es simple o elaborada. Según esta señal, nuestra máquina de estados tendrá que pasar al estado siguiente o saltar al estado siguiente +1 (como se ha dicho, las golosinas complejas cuentan como dos simples y la bolsa puede llevar un máximo de 5 golosinas). Cuando la bolsa contiene 5 golosinas se cierra y avanza la cadena teniendo una nueva

bolsa vacía por lo que empezamos de nuevo todo el proceso. Un contador módulo 8 con carga en paralelo y puertas lógicas nos servirá para implementar el comportamiento de este sistema secuencial. A partir del diagrama de transición de estados que se indica, implementar la tabla de verdad con la función de transición de estados con Mealy, la función de salida es 1 cuando la bolsa está llena y 0 en otro caso, y el contador con la señal Load, Count y el estado a cargar cuando se realiza la carga en paralelo.



NOTA: los contadores pueden cargarse de forma serie o paralela dependiendo de si la señal de Load está a 0 o a 1 (respectivamente). La señal Count aumenta secuencialmente el contador con cada pulso de reloj.

41. Sea un sistema secuencial que implementa el siguiente algoritmo, donde A, B, C son números de 8 bits.

```

for I = 0 to B
  if A > B then C = A + B;
  else C = A - B;
  A = C/2;
end for;

```

- Describe brevemente el hardware necesario para su implementación (número de registros, contadores, sumadores ...)
- Si además de los módulos descritos anteriormente se necesitan multiplexores, indique dónde se necesitan y por qué.
- Diseñe y codifique la máquina de estados.

42. Sea un sistema secuencial que implementa el siguiente algoritmo, donde A, B, C son números de 8 bits.

```

for I = 0 to B
  if A > B then C = A/2;
  else C = A*2;
  A = A + C;
end for;

```

- Describe brevemente el hardware necesario para su implementación (número de registros, contadores, sumadores ...)
- Si además de los módulos descritos anteriormente se necesitan multiplexores, indique dónde se necesitan y por qué.
- Diseñe y codifique la máquina de estados.

Problemas del tema 6

43. Indique cuál es el resultado de ejecutar las siguientes instrucciones, dando el contenido final de los registros, el valor de los indicadores de condición y el valor de R@. Se supone para cada caso que el contenido inicial de los registros y posiciones de memoria es el siguiente:

R1=2454h, R2=5656h, R3=FFFFh, R4=0000h, R5=0002h

Memoria:	Dirección	contenido
	00h	0339h
	01h	63AFh
	02h	EA00h
	03h	3304h
	04h	7834h
	05h	54AAh

- a) ADD R1, R2, R5
- b) SUBI R3, #1, R1
- c) LOAD 2(R4), R1
- d) STORE R5, 3(R3)

44. Indique cuál es el resultado de ejecutar las siguientes instrucciones, dando el contenido final de los registros y posiciones de memoria que se modifican y el valor de los indicadores de condición. Se supone para cada caso que el contenido inicial de los registros y posiciones de memoria es el siguiente:

R1=2454h, R2=5656h, R3=FFFFh, R4=0000h, R5=0002h

Memoria:	Dirección	contenido
	00h	0339h
	01h	63AFh
	02h	EA00h
	03h	3304h
	04h	7834h
	05h	54AAh

- a) ADD R1, R2, R5
- b) SUBI R3, #1, R1
- c) LOAD 2(R4), R1
- d) STORE R5, 3(R3)

45. Dadas las siguientes instrucciones y el contenido de los registros especificado y sabiendo que el contenido inicial de la posición de memoria 0003h es 0101h, Indicar el contenido de los registros después de la ejecución de las 3 instrucciones:

SUB R1,R3,R4	R1= 2454 h
ADDI R3, #2, R4	R2= 0002 h
LOAD 1(R2), R5	R3= 5885 h

R4= 0000 h
R5= 47AB h

46. Traduzca a código máquina el siguiente programa de la máquina rudimentaria:

0Ah LOAD 9(R0), R2
0Bh BEQ 14
0Ch SUBI R2, #1, R2
0Dh SUB R2, R1, R0
0Eh BR 11
0Fh STORE R2, 9(R0)

47. En un computador de tamaño palabra de 32 bits se tiene el siguiente esquema para las instrucciones aritméticas (COP indica el tipo de instrucción aritmética). Teniendo en cuenta los bits que sobran contestar las siguientes preguntas.

- a) Si las instrucciones tienen 3 operandos en registro ($R1 = R2 + R3$). ¿Cuál puede ser el tamaño del banco de registros?
- b) Si las instrucciones tuvieran un operando en registro más inmediato ($R1 = R1 + \text{inmd.}$) ¿qué valores podría alcanzar el inmediato? (suponiendo que tenemos un banco de 32 registros).

1111	COP (6 bits)	
------	--------------	--

48. Suponiendo que en cada uno de los cuatro casos siguientes el contenido del PC de la máquina rudimentaria es 50, y teniendo en cuenta la información que se proporciona sobre el estado de los indicadores de condición, determine razonadamente (no se otorgará puntuación por una respuesta no razonada) cuál será la dirección de la instrucción que se ejecutará inmediatamente después de ejecutar cada una de ellas:

Caso	Instrucción	biestable N	biestable Z
a)	BEQ 100	0	0
b)	BLE 100	0	0
c)	BNE 100	1	0
d)	BGE 100	0	0

49. Dados dos vectores A y B de 10 componentes cada uno se desea construir otro vector C tal que

$$C(i) = |A(i) + B(9-i)| \quad i = 0, \dots, 9.$$

Se pide: escriba un programa en lenguaje de alto nivel que construya el vector C. Traduzca el programa al lenguaje ensamblador de la MR. Los datos se ubicarán a partir de la posición de memoria 20 y el programa a partir de la posición de memoria 60.

50. Dado un vector A de 10 componentes se desea generar otro vector B, tal que B sólo contiene las componentes de A que son mayores que 5. Ejemplo: si

$A = (0,1,7,8,4,5,12,11,6,3)$ entonces $B = (7,8,12,11,6)$. Se pide: Escriba un programa en lenguaje de alto nivel que implemente el procesamiento descrito y calcule el número de componentes del vector B. Traduzca el programa al lenguaje ensamblador de la MR. Los datos se ubicarán a partir de la posición de memoria 10 y el programa a partir de la posición de memoria 50.

51. Dado un vector A de 10 componentes se desea generar otro vector B, tal que B contenga las componentes de A en orden inverso.

Ejemplo: si $A = (0,1,7,8,-4,5,-12,11,-6,3)$ entonces $B = (3,-6,11,-12,5,-4,8,7,1,0)$.

- a) Escriba un programa en lenguaje de alto nivel que implemente el procesamiento descrito. Traduzca el programa al lenguaje ensamblador de la MR.
- b) Incluye un bucle al final del programa anterior para que en el vector B se almacenen los valores absolutos de A en orden inverso.

Problemas del tema 7

52. En relación al diseño de la máquina rudimentaria, indique (razonadamente) cuáles de las siguientes afirmaciones son falsas:

- Todas las instrucciones tardan en ejecutarse el mismo número de ciclos.
- Toda instrucción tarda en ejecutarse un número de ciclos fijo.
- La duración del ciclo de reloj depende fundamentalmente de los retardos de los módulos combinacionales de la unidad de proceso.
- La duración del ciclo de reloj no es fija y es diferente en función del retardo de los módulos de la unidad de proceso implicados en la instrucción a ejecutar.

53. Indique el conjunto de acciones que tiene lugar en la MR en la fase DECO. Indique los puntos de control que se deben activar para implementar dichas acciones.

54. Si la unidad de control de la Máquina Rudimentaria envía a la unidad de Proceso la señal PC/@=1 y a la memoria L/E=1:

- ¿Qué acciones realizará?
- ¿En qué estados del diagrama simplificado toman simultáneamente dichos valores? Complete la siguiente tabla con el resto de señales de salida.

Estado	Ld_RA	Ld_IR	Ld_PC	Ld_R@	Ld_RZ	Ld_RN	ERd	CRf	OPERAR

55. Si la Unidad de Control de la MR envía a la Unidad de Proceso las siguientes señales: Ld_RN = 1 y Ld_RZ = 1

- ¿Qué le está indicando?
- ¿En que estado/s del diagrama de estados simplificado de la MR toman las señales de control estos valores?. Completa la tabla con el valor que toman las señales de control en dichos estados.

Salidas Unidad Control											
Estado	Ld_RA	Ld_IR	Ld_PC	Ld_R@	Ld_RZ	Ld_RN	ERd	L/E	PC/@	CRf	OPERAR

56. Indique el conjunto de acciones que tiene lugar en la MR en la fase BRANCH. Teniendo en cuenta el valor de las señales de control de dicha fase, indique cuáles son las señales que integran cada uno de los siguientes grupos:

- Grupo 0: señales que valen 0
- Grupo 1: señales que valen 1
- Grupo 2: señales cuyo valor es irrelevante

57. Indique el conjunto de acciones que tiene lugar en la MR en la fase ARIT. Teniendo en cuenta el valor de las señales de control de dicha fase, indique cuáles son las señales que integran cada uno de los siguientes grupos:

- Grupo 0: señales que valen 0
- Grupo 1: señales que valen 1
- Grupo 2: señales cuyo valor es irrelevante

58. Indique cómo afectaría a las siguientes componentes de la máquina rudimentaria el aumentar de tamaño la memoria hasta tener 16384 palabras de 32 bits en lugar de 256 palabras de 16 bits.

- a) Registro contador de programa (PC) y registro de direcciones (R@)
- b) Registro de datos (RA) y banco de registros

59. Indique cómo afectaría a las siguientes componentes de la máquina rudimentaria el aumentar de tamaño la memoria hasta tener 4096 palabras en lugar de 256.

- a) Registro contador de programa (PC) y registro de direcciones (R@)
- b) Registro de datos (RA) y banco de registros
- c) Unidad aritmético lógica
- d) Diagrama de estados de la unidad de control

60. Suponer que queremos implementar el repertorio de instrucciones de la máquina rudimentaria con un banco de registros que solo permita realizar en un ciclo de reloj o una lectura o una escritura, pero nunca las dos operaciones simultáneamente.

- a) Como afectaría este cambio al formato de las instrucciones. Justifícalo
- b) Como afectaría al camino de datos. dibuja el esquema. Justifícalo.
- c) Como afectaría a la rama del diagrama de estados de las instrucciones aritmético - lógicas. Indica para cada estado las señales activas.

61. Indique cómo afectaría a los elementos de la ruta de datos y al formato de instrucciones de la MR las siguientes modificaciones:

- a) Cambiar la memoria de 256 palabras por una de 512 y con anchura de 32 bits
- b) un Banco de Registros de 32 bits y 32 posiciones
- c) Las modificaciones expresadas en a) y en b) conjuntamente.

62. Indique cómo afectaría a las siguientes componentes de la máquina rudimentaria el aumentar de tamaño la memoria hasta tener 16384 palabras de 32 bits en lugar de 256 palabras de 16 bits.

- a) Registro contador de programa (PC) y registro de direcciones (R@)
- b) Registro de datos (RA) y banco de registros
- c) Unidad aritmético lógica
- d) Diagrama de estados de la unidad de control

63. Indique cómo afectaría al registro contador de programa (PC), registro de direcciones (R@) y registro de datos (RA) de la MR el aumentar de tamaño la memoria hasta tener una memoria de 1024 palabras en lugar de 256 palabras.

- a) ¿Sería imprescindible cambiar el formato de las instrucciones LOAD y STORE para poder acceder a cualquier posición de la memoria? Razone la respuesta. En caso afirmativo explique el cambio.
- b) ¿Sería imprescindible cambiar el formato de las instrucciones de salto para poder saltar a cualquier posición de la memoria? Razone la respuesta. En caso afirmativo explique el cambio.

64. Suponiendo que la palabra de procesador de la Máquina rudimentaria es de 64 bits y que el banco de registros tiene 64 registros, indicar:

- a) Máximo tamaño de memoria que se podría direccionar.
- b) Rango de operandos inmediatos representables en complemento a dos.
- c) Nuevo formato de las instrucciones de load, bifurcación y suma con inmediato.

65. Indique cómo afectarían al formato de instrucción y los elementos de la unidad de proceso de la máquina rudimentaria las siguientes modificaciones:

- a) Ampliar la memoria para que tenga 512 palabras de 32 bits cada una
- b) Ampliar el Banco de Registros para que tenga 32 registros de 32 bits cada uno
- c) Las modificaciones expresadas en a) y en b) conjuntamente.

66. Indique cómo afectaría al registro de instrucción (IR), registro de direcciones (R@) y tamaño de palabra de memoria de la MR el aumentar a 32 el número de registros. ¿Sería imprescindible cambiar el formato de las instrucciones? Razone la respuesta. En caso afirmativo explique el cambio.

67. En un computador de tamaño palabra de 32 bits se tiene el siguiente esquema para las instrucciones aritméticas, en donde hay 16 operaciones que se pueden ejecutar tanto con inmediato como sin inmediato (Ej ADD y ADDI en la MR) y 8 operaciones que tienen siempre dos operandos (Ej AND en la MR).

- a) Decidir la codificación de estas instrucciones.
- b) Diseñar la ALU.
- c) Cuantas operaciones aritméticas diferentes se podrían codificar

1111	Rd (5 bits)	Rf1(5 bits)	Rf2(5 bits)	
------	-------------	-------------	-------------	--

1111	Rd (5 bits)	Rf1(5 bits)	Inmediato (12 bits)	
------	-------------	-------------	---------------------	--

68. Se desea añadir al juego de instrucciones de la MR las siguientes instrucciones aritméticas: 1) una instrucción de desplazamiento aritmético a la izquierda con un operando fuente almacenado en un registro y el resultado en un registro destino (ASL Rf, Rd); 2) una instrucción de test que activa los indicadores de condición según el valor almacenado en un registro fuente y escribe ese mismo valor en un registro destino (TST Rf, Rd).

- a) ¿Qué implicaciones tiene en el formato de las instrucciones el añadir estas instrucciones?
- b) Diseñe la nueva ALU y rediseñe el hardware que rodea la ALU.

OP	000	001	100	101	110	111
	ADDI	SUBI	ADD	SUB	ASR	AND

69. Dada una máquina que posee el mismo juego de instrucciones de la máquina rudimentaria y:

- Tiene 16 registros de propósito general
- Trabaja con números en complemento a 2 de 16 bits
- Tiene una memoria de 2048 palabras

Se pide:

- Propón una codificación de instrucciones para dicha máquina y especifica el tamaño mínimo de las palabras de memoria, el tamaño mínimo de las instrucciones y el tamaño máximo del operando inmediato.
- ¿Cómo afectaría el nuevo formato al tamaño de los registros RA, R@, IR y PC? Razone la respuesta.
- Esta máquina trabaja con el repertorio de instrucciones de la máquina rudimentaria, pero dicho repertorio sólo permite una operación lógica (la operación AND). Se quiere completar el conjunto de instrucciones aritmético-lógicas añadiendo las siguientes operaciones OR, NAND y NOR, manteniendo el mismo formato que la AND. Además se quiere que estas nuevas instrucciones, junto a la AND, tengan también una versión con operando inmediato. Define el formato de las nuevas instrucciones y propón una codificación para estas instrucciones de tal forma que no sea necesario modificar el diagrama de estados de la MR ni el multiplexor SELDAT de la ruta de datos. Ignorar las posibles modificaciones en el interior de la ALU.

70. Dada una máquina que posee el mismo juego de instrucciones de la máquina rudimentaria y que tiene 64 registros de propósito general, trabaja con números en complemento a 2 de 16 bits y tiene una memoria de 256K palabras

- Propón una codificación de instrucciones para dicha máquina y especifica el tamaño mínimo de las palabras de memoria, el tamaño mínimo de las instrucciones y el tamaño máximo del operando inmediato.
- ¿Qué modificaciones habría que hacer en el resto de la unidad de proceso?
- El repertorio de instrucciones de la máquina rudimentaria sólo permite almacenar en memoria datos desde un registro (instrucción STORE Rf, A(Ri)). Se quiere añadir una nueva instrucción, STOREI #n, A(Ri), que permita almacenar en memoria datos desde la instrucción (operando inmediato). Redefine el formato de las instrucciones teniendo en cuenta la nueva instrucción de tal forma que no sea necesario modificar el diagrama de estados de la MR. ¿Se conserva el tamaño de la instrucción?
- ¿Qué modificaciones hay que hacer en la unidad de proceso (ruta de datos) de forma que sea posible ejecutar la nueva instrucción? ¿Afecta al tiempo de la fase STORE? Razona tus respuestas.

71. El repertorio de instrucciones de la máquina rudimentaria sólo permite cargar en registro datos desde la memoria (instrucción LOAD A(Ri), Rd). Se quiere añadir una nueva instrucción, LOADI A(Ri), #n, que permita almacenar en registro datos desde la instrucción (operando inmediato).

- Redefine el formato de las instrucciones teniendo en cuenta la nueva instrucción de tal forma que no sea necesario modificar el diagrama de

estados de la MR. ¿Se conserva el tamaño mínimo de la instrucción? ¿y el tamaño máximo del operando?

- b) ¿Qué modificaciones hay que hacer en la unidad de proceso (ruta de datos) de forma que sea posible ejecutar la nueva instrucción? ¿Afecta al tiempo de la fase LOAD? Razona tus respuestas.
- c) Se quiere modificar el diagrama de estados de la MR de manera que la nueva instrucción se ejecute en el menor número de ciclos posibles, para ello añadimos un nuevo estado llamado LOADI. Dibuja el nuevo diagrama de estados ¿Cuántos ciclos tardan las instrucciones LOADI?
- d) Señala en el dibujo de la unidad de proceso las componentes, señales de control y buses internos que se usan en dicho estado.

72. Dada una máquina que posee el mismo juego de instrucciones de la máquina rudimentaria:

- a) Propón una codificación de instrucciones en el supuesto de que dicha máquina:
 - Tenga 32 registros de propósito general
 - Trabaje con números en complemento a 2 de 20 bits
 - Tenga una memoria de 1024 palabras

Especificar el tamaño máximo del operando inmediato.

- b) ¿Qué modificaciones habría que hacer en el resto de la unidad de proceso?
- c) ¿Cuántas instrucciones aritmético-lógicas podríamos añadir como máximo a las que ya existen en la máquina rudimentaria? Añadir al repertorio de instrucciones de la máquina rudimentaria una nueva instrucción: INI #n,Rd

Esta instrucción inicializa el registro Rd con el valor n. Propón una codificación para esta instrucción, conservando el tamaño del operando inmediato.

- d) Especificar el valor de las señales de control CRf, Erd y Operar en todos los estados de la unidad de control para la nueva instrucción. ¿Cuál será el valor de la expresión de conmutación de la señal Operar?

73. Indique cuantos ciclos tarda en ejecutarse el siguiente programa, suponiendo el grafo simplificado para la unidad de control.

```
00h ADDI R0,#5,R1
01h SUB R0,R1,R0
02h BL 4
03h BR 7
04h STORE R1,3(R0)
05h ADD R0,R0,R2
06h BE 3
```

74. Calcule el número de ciclos y el tiempo que tarda en ejecutar la máquina rudimentaria el siguiente programa, suponiendo que la frecuencia de reloj es 10 MHz:

```
00h BR 4
01h 0005h
02h 0003h
03h 0000h
```

```

04h  LOAD 1(R0),R1
05h  LOAD 2(R0),R2
06h  ADD R0,R0,R3
07h  ADD R1,R3,R3
08h  SUBI R2,#1,R2
09h  BNE 7
0Ah  STORE R3,3(R0)

```

75. Dado el siguiente programa:

```

00h ADDI R0,#5,R1
01h SUB R0,R1,R0
02h BL 4
03h BR 6
04h ADD R0,R0,R2
05h BE 3

```

Contesta las siguientes preguntas:

- Valor del PC en cada una de las instrucciones.
- Cuantos ciclos tarda en ejecutarse el programa, suponiendo el grafo simplificado para la unidad de control.

76. Considere el siguiente programa, escrito en código ensamblador de la MR:

```

03h  LOAD 0(R0),R1
04h  SUBI R1,#6,R1
05h  BLE 7
06h  BR 9
07h  LOAD 1(R0),R2
08h  STORE R1,2(R0)
09h  ADD R1,R2,R3

```

Las primeras posiciones de memoria están inicializadas a los siguientes valores:

Dirección	Contenido
00h	0003h
01h	FFFFh
02h	EA00h

- Indique el orden de ejecución del código. ¿Cuántos ciclos tarda en ejecutarse usando el diagrama simplificado de la MR? (1 punto)
- Indique el contenido de RZ y RN tras la ejecución de la instrucción LOAD 1(R0),R2 ¿Y tras ejecutar la instrucción STORE R1,2(R0)? (1 punto)
- Calcule el tamaño del código del programa. ¿Cuál sería el tamaño de este mismo código si la máquina rudimentaria dispusiera de 32 registros de propósito general en lugar de solamente 8? ¿Y si además aumentamos el tamaño de la memoria hasta 1024 palabras? Justifique brevemente tus respuestas (1,5 puntos)

77. Traducir a código ensamblador de la máquina rudimentaria el siguiente programa, intentando minimizar el número de instrucciones. Calcular el valor de PC, P@ y los registros de condición para cada una de las instrucciones.

a = 7;
b = 5;

```

c:= a-b;
if c>0 then
    d:= c + b;
else
    d:= c + a;
end if;
while d>0 then
    c:= d + c;
    d:= d - a;
end while;

```

78. Calcule el número de ciclos y el tiempo que tarda en ejecutar la máquina rudimentaria la siguiente sección de código suponiendo que la frecuencia de reloj es 13 MHz:

```

01h  BR 5
02h  0001h
03h  000ch
04h  000ah
05h  LOAD 2(R0),R1
06h  LOAD 2(R1),R2
07h  LOAD 3(R1),R3
08h  ADD R2,R3,R4
09h  SUB R2,R3,R5
0Ah  SUB R4 R5 R6
0Bh  BE 0F
0Ch  STORE R4,3(R1)

```

79. Sea el siguiente programa escrito en pseudocódigo.

```

a, b ,c : word;
a := 5;
b := 7;
for i = 0 to b do
    if a>b then a := a-1;
    else
        if a > 0 then a := a+1;
    end for;
c := (a+b)/2;

```

- Traduce este programa a lenguaje ensamblador de la Máquina Rudimentaria.
- Calcule el tiempo de ejecución del programa completo, se supone que cada estado del diagrama simplificado de la máquina de control de la MR se ejecuta en un ciclo de reloj.
- Señale las instrucciones en donde el PC señalará siempre a la siguiente instrucción que se va a ejecutar.

80. Usando la hoja adjunta, partiendo del estado del computador que se describe en la primera columna y suponiendo que la dirección 09h de memoria contiene 0007h, realice una simulación ciclo a ciclo de la ejecución en la

máquina rudimentaria del siguiente programa (en cada columna debe indicarse el estado del controlador y valor de los registros en cada ciclo):

```

0Ah LOAD 9(R0), R2
0Bh BEQ 15
0Ch SUBI R2, #1, R2
0Dh SUB R2, R1, R0
0Eh BR 11
0Fh STORE R2, 9(R0)

```

81. Considere el siguiente programa, escrito en código ensamblador de la MR:

```

02h LOAD 0(R0),R1
03h ADDI R1,#6,R1
04h BLE 6
05h STORE R1,2(R0)
06h LOAD 1(R0),R2
07h SUB R1,R2,R3

```

Las primeras posiciones de memoria están inicializadas a los siguientes valores:

Dirección	Contenido
00h	0001h
01h	FFFFh

- Indique el orden de ejecución del código. (0.5 puntos)
- ¿Cuánto tiempo tardaría en ejecutarse en la máquina del apartado a) si ésta sigue el diagrama simplificado de la MR? (1 punto)
- Calcule el tamaño del código (nº de bits) del programa. ¿Cuál sería el tamaño de este mismo código si la máquina rudimentaria dispusiera de 32 registros de propósito general en lugar de solamente 8? (1 punto)

82. Dado el siguiente programa expresado en lenguaje ensamblador de la Máquina Rudimentaria y el contenido inicial y de la memoria y registros:

04h: LOAD 0(R0),R1

05h: ADD R1,R0,R2

06h: BEQ 9

07h: ADDI R0,#1,R3

08h: STORE R3,1(R0)

09h: STORE R0,2(R0)

Registros	Indicadores	Memoria
R1: A57Bh	RN: 0	00h: 0002h
R2: 0B00h	RZ: 0	01h: 000Ch
R3: 0001h		02h: 00FFh
R4: 0000h		03h: 8004h
R5: FF00h		
R6: 011Bh		
R7: 0C88h		

- Expresa en hexadecimal el contenido de la memoria desde la posición 04h a la 09h.
- Indica el orden de ejecución de las instrucciones.
- Indica el contenido de los registros (R0-R7), de los indicadores de condición (RN y RZ) y de la memoria (posiciones 00h-03h), al finalizar la ejecución.
- Indica la secuencia de estados seguida por el controlador para ejecutar el programa completo suponiendo que el estado inicial del mismo al comienzo de la instrucción 04h es FETCH, ¿cuántos ciclos de reloj tarda en ejecutarse el programa?

83. Dado el contenido inicial de los registros y de la memoria y sabiendo que el programa está cargado entre la posición 04h y la 06h (ambas inclusive). Contestar las siguientes preguntas:

Registros

R0: 0000h
 R1: 0001h
 R2: 0004h
 R3: FE04h
 R4: BA00h
 R5: 001Bh
 R6: 0300h
 R7: 0000h
 RZ: 0
 RN: 0

Memoria:

Dirección	Contenido
00h:	0003h
01h:	0005h
02h:	000Ah
03h:	C804h
04h:	EC09h
05h:	B009h
06h:	6902h

- a) Cual es el programa que hay cargado en la memoria?. Razona tu respuesta.
- b) ¿Cuál es el orden de ejecución de las instrucciones? Razona tu respuesta.
- c) ¿Cuáles son los valores de la memoria, los registros del banco de registros (R0-R7) y los indicadores de condición (RN y RZ) al finalizar la ejecución?.
- d) ¿Cuántos ciclos de reloj tarda en ejecutarse el programa completo? Razona tu respuesta.

84. Dado el siguiente código en lenguaje ensamblador de la máquina rudimentaria:

- a) Discute razonadamente el orden de ejecución del siguiente código en lenguaje ensamblador de la máquina rudimentaria. Muestra el contenido de los registros y de la memoria al final de la ejecución del programa.

03h: SUBI R5, #9, R6
 04h: LOAD 1(R6), R5
 05h: BNE 7
 06h: BEQ 8
 07h: AND R5, R1, R2
 08h: STORE R2, 1(R0)

Registros:

R0: 0000h
 R1: 0FF0h
 R2: 0A09h
 R3: FF04h
 R4: AA00h
 R5: 000Ah
 R6: 0000h
 R7: 0000h
 RZ: 0
 RN: 0

Memoria:

00h: 0003h
 01h: 0005h
 02h: B00Bh
 03h: F549h
 04h: 2E01h
 05h: A807h
 06h: 8808h
 07h: D527h
 08h: 5001h

- b) ¿Cuántos ciclos tarda en ejecutarse el código anterior usando el diagrama simplificado de la máquina rudimentaria? Razona tu respuesta.
- c) Indica los diferentes valores (en hexadecimal) que toman los registros IR, PC y RN en cada una de las fases de ejecución de las tres primeras instrucciones del programa. NOTA: inicialmente, PC= 03h e IR=0000h.

Fase	FETCH							
PC								
RN								
IR								

- d) Indica el valor que tienen cada una de las componentes, señales de control y buses internos de la unidad de proceso de la máquina rudimentaria en la fase ARIT de la primera instrucción. (marcarlo en el dibujo)

85. Utilizando una tabla como la mostrada (use valores en hexadecimal) y partiendo de la situación inicial en ella indicada, realice una simulación ciclo a ciclo del comportamiento de la máquina rudimentaria durante la ejecución del programa siguiente:

```

00h  BR 2
01h  FFF5h
02h  LOAD 1(R0),R1
03h  SUBI R1, #11, R1

```

Estado	FETCH							
IR	0000							
PC	00							
R@	00							
RA	0000							
RZ	0							
RN	0							
R1	0000							

86. Dado el siguiente código en lenguaje ensamblador de la máquina rudimentaria:

- a) Discuta razonadamente el orden de ejecución del siguiente código en lenguaje ensamblador de la máquina rudimentaria. Muestra el contenido de los registros que se han modificado, incluyendo el RZ y el RN, y el contenido de la memoria al final de la ejecución del programa

```

03h  LOAD 1(R2), R3
04h  SUBI R3, #12, R1
05h  BEQ 7
06h  BR 8
07h  ADD R1, R5, R6
08h  STORE R1, 1(R6)

```

Registros

R0: 0000h
 R1: 8003h
 R2: 0001h
 R3: FF04h
 R4: AA00h
 R5: 000Bh
 R6: 0000h
 R7: 0000h
 RZ: 0
 RN: 0

Memoria:

Dirección	Contenido
00h:	0003h
01h:	0005h
02h:	000Ah
03h:	1A01h
04h:	CB19h
05h:	8807h
06h:	8008h
07h:	F1A4h
08h:	4E01h

- b) ¿Cuántos ciclos tarda en ejecutarse el código anterior usando el diagrama simplificado de la máquina rudimentaria?. Razona tu respuesta.
- c) Indica los diferentes valores (en hexadecimal) que toman los registros IR, PC y R@ en cada una de las fases de ejecución de las tres primeras instrucciones del programa.

Fase	FETCH							
PC								
R@								
IR								

- d) Indica razonadamente el valor que tienen cada una de las componentes, señales de control y buses internos de la unidad de proceso de la máquina rudimentaria en la fase DECO de la primera instrucción. (señalar estos valores en el dibujo de la unidad de proceso, resaltando los que van a ser relevantes para la instrucción de LOAD)

NOTA: el contador de programa cuando empieza la ejecución contiene la dirección 03h, el R@ contiene 00h y el IR 0000h.

87. En la siguiente tabla se puede ver el contenido inicial de los registros y de las primeras posiciones de memoria de la Máquina Rudimentaria

Contenido inicial de registros:	Contenido inicial de memoria:
R1: A57Bh RN: 0	00h: 0002h
R2: 0001h RZ: 0	01h: 000Ch
R3: 0005h	02h: 00FFh
R4: 0000h	03h: 8004h
R5: FF00h	04h: 0000h
R6: 011Bh	05h: 0000h
R7: 0C88h	06h: 0A02h

- a) Expresa en lenguaje ensamblador de la Máquina Rudimentaria la instrucción que se encuentra en la posición de memoria 06h
- b) Completa la siguiente tabla para cada uno de los estados por los que pasa dicha instrucción hasta que se habiliten las señales de carga de la siguiente

instrucción. Suponer que los registros RA, R@ e IR están reseteados y PC=06h

	Estados						
Registros							
IR							
PC							
R@							
RA							
RZ							
RN							
Ri Banco-Reg							

Nota: Ri Banco-Reg sólo hay que rellenarlo si la instrucción escribe en el Ri

88. El contenido inicial de la memoria de la Máquina Rudimentaria es el siguiente:

00h 0030h
 01h C0F2h
 02h 0800h
 03h D110h
 04h 5001h

- Sabiendo que la primera instrucción está almacenada en la posición 02h, escriba el programa correspondiente en Lenguaje Ensamblador.
- Indique el número de ciclos necesarios para la ejecución del programa. Razone su respuesta.
- Complete la siguiente tabla con los valores de los registros indicados durante toda la ejecución.

	FETCH								
IR	x								
PC	02h								
RA	x								
RN	x								

89. Suponiendo que la memoria contiene el siguiente programa expresado en lenguaje ensamblador y que el resto de posiciones de memoria (a partir de 05h, inclusive) contienen FFFFh:

00h: LOAD 10(R0),R1
 01h: BEQ 4
 02h: ADDI R1,#1,R1
 03h: BR 1
 04h: STORE R1,11(R0)

- Expresa en hexadecimal el contenido de la memoria desde la posición 00h a la 04h.
- Partiendo de la situación inicial mostrada, completa la siguiente tabla indicando los estados y los valores en hexadecimal que toman los registros durante la ejecución del programa.

Estado	IR	PC	R@	RA	RZ	RN	R1
FETCH	0000	00	00	00000	0	0	0000

90. Dado el siguiente programa expresado en lenguaje ensamblador de la Máquina Rudimentaria, y el contenido inicial de los registros y de la memoria de la figura:

```

04h: LOAD 0(R0),R1
05h: ADD R1,R2,R2
06h: BEQ 9
07h: SUBI R0,#1,R3
08h: STORE R3,1(R0)
09h: STORE R0,2(R0)

```

Contenido inicial de registros:	Contenido inicial de memoria:
R1: A57Bh	00h: 0002h
R2: 000Bh	01h: 000Ch
R3: 0001h	02h: 00FFh
R4: 0000h	03h: 8004h
R5: FF00h	
R6: 011Bh	
R7: 0C88h	
RN: 0	
RZ: 0	

- a) Completa el contenido de la memoria (posiciones 04h a 09h). Razona tu respuesta (0,5 puntos)
- b) ¿Cuál es el orden de ejecución de las instrucciones? Razona tu respuesta (0,5 puntos)
- c) ¿Cuáles son los valores de la memoria, los registros del banco de registros (R0-R7) y los indicadores de condición (RN y RZ) al finalizar la ejecución? (0,5 puntos)
- d) ¿Cuántos ciclos de reloj tarda en ejecutarse el programa completo? Razona tu respuesta (1 punto)

e) Completa en la siguiente tabla los valores (expresados en hexadecimal) de los registros IR, PC, R@ y RZ durante las tres primeras instrucciones ejecutadas. Considera que inicialmente PC=04h (1,5 puntos).

Fases	FETCH													
IR														
PC														
R@														
RZ														

91. Dado el siguiente código en lenguaje ensamblador de la máquina rudimentaria:

a) Discuta razonadamente el orden de ejecución. Muestre el contenido en hexadecimal de los registros que se han modificado, incluyendo el RZ y el RN, y el contenido de la memoria al final de la ejecución del programa.

```

03h  ADD R0, R0, R1
04h  ADDI R0, #2, R2
05h  STORE R3, 0 (R2)
06h  BG 08h
07h  AND R3, R4, R4
08h  LOAD 1(R1), R6
  
```

Registros

R0: 0000h
R1: 4010h
R2: 0004h
R3: FE04h
R4: BA00h
R5: 001Bh
R6: 0300h
R7: 0000h
RZ: 0
RN: 1

Memoria:

Dirección	Contenido
00h:	0003h
01h:	0005h
02h:	000Ah
03h:	C804h
04h:	D010h
05h:	5A00h
06h:	B808h
07h:	E387h
08h:	3101h

b) Discuta el contenido de los registros PC, RA y R@ durante las distintas fases de la primera instrucción suponiendo que para los tres registros el valor inicial es 3.

Fases					
PC					
RA					
R@					

c) Considere que el tiempo de respuesta de los diferentes elementos de la ruta de datos de la máquina rudimentaria es el siguiente:

- 10ns para los multiplexores
- 20ns para el incrementador del PC
- 25ns para el sumador de direcciones

- 80ns para el extensor de signo
- 5ns para la evaluación de la condición de salto
- 30ns para leer un registro del banco de registros
- 100ns el tiempo de respuesta de la ALU
- 60ns para acceder a la memoria en operaciones de lectura y 120ns en operaciones de escritura

Calcule el tiempo de ciclo mínimo necesario para que funcione correctamente la fase de ARIT de la máquina rudimentaria. Razone la respuesta.

92. Considere que el tiempo de respuesta de los diferentes elementos de la ruta de datos de la máquina rudimentaria es el siguiente:

- 10ns para el multiplexor SELDIR
- 20ns para los multiplexores SELREG y SELDAT
- 15ns para el incrementador del PC y para el sumador de direcciones
- 80ns para el extensor de signo
- 20ns para la evaluación de la condición de salto
- 40ns para leer un registro del banco de registros
- 120ns el tiempo de respuesta de la ALU
- 40ns para acceder a la memoria en operaciones de lectura y 100ns en operaciones de escritura

- a) Calcule la frecuencia máxima de reloj a la que puede funcionar la máquina rudimentaria.
- b) Discuta razonadamente el orden de ejecución del siguiente código en lenguaje ensamblador de la máquina rudimentaria. ¿Cuántos ciclos tarda en ejecutarse el siguiente código usando el diagrama simplificado de la máquina rudimentaria?

```

00h  LOAD 1(R0), R1
01h  BR 4
02h  ADDI R0, #2, R2
03h  BR 6
04h  SUB R0, R0, R0
05h  BEQ 2

```

- c) Si se añade un registro RI entre el extensor de signo y el multiplexor SELDAT, ¿será necesario modificar el diagrama de estados simplificado de la máquina rudimentaria? Razonar la respuesta. Indique las modificaciones que se producen en la tabla de salidas.
- d) Calcule de nuevo la frecuencia máxima de reloj a la que funciona la máquina rudimentaria después de añadir el registro RI.

93. Calcula el tiempo de ciclo mínimo necesario para que funcione correctamente el estado ARIT sabiendo que el tiempo de respuesta de los diferentes elementos de la ruta de datos de la máquina rudimentaria es el siguiente:

- 10ns para los multiplexores
- 20ns para el incrementador del PC
- 25ns para el sumador de direcciones
- 2ns para el extensor de signo
- 5ns para la evaluación de la condición de salto
- 30ns para leer un registro del banco de registros

- 100ns el tiempo de respuesta de la ALU
- 60ns para acceder a la memoria en operaciones de lectura y 120ns en operaciones de escritura

94. Suponiendo que se desee que la máquina rudimentaria funcione a una frecuencia de reloj de 6,25 MHz. ¿Cuál sería el tiempo de respuesta máximo que podría tener la ALU, sabiendo que los tiempos de respuesta de los restantes elementos de la ruta de datos son los siguientes?:

- 10ns para los multiplexores
- 20ns para el incrementador del PC
- 25ns para el sumador de direcciones
- 5ns para la evaluación de la condición de salto
- 30ns para leer un registro del banco de registros
- 60ns para acceder a la memoria en operaciones de lectura y 120ns en operaciones de escritura

95. Considere que el tiempo de respuesta de los diferentes elementos de la ruta de datos de la máquina rudimentaria es el siguiente:

- 25ns para el multiplexor SELDIR
- 35ns para los multiplexores SELREG y SELDAT
- 15ns para el incrementador del PC
- 20ns para el sumador de direcciones
- 40ns para el extensor de signo
- 20ns para la evaluación de la condición de salto
- 40ns para leer un registro del banco de registros
- 120ns el tiempo de respuesta de la ALU con operaciones aritmético-lógicas
- 25ns el tiempo de respuesta de la ALU cuando la información pasa sin operar.
- 60ns para una lectura de memoria y 100ns para una escritura.

Sabiendo que fase de instrucción que más tarda en ejecutarse es la fase de ARIT ¿Cuál sería el período mínimo de la señal de reloj para poder ejecutar correctamente una instrucción en la MR siguiendo el diagrama simplificado?. Justifique su respuesta.

96. Sabiendo que los tiempos de respuesta de los elementos de la ruta de datos son los siguientes:

- 12ns para los multiplexores
- 22ns para el incrementador del PC
- 30ns para el sumador de direcciones
- 7ns para la evaluación de la condición de salto
- 40ns para leer un registro del banco de registros
- 70ns para acceder a la memoria en operaciones de lectura y 140ns en operaciones de escritura
- 45ns para la UAL

Cuáles serían las duraciones de las fases deco y load. Razonar la respuesta.

97. Usando la hoja adjunta y partiendo del estado que se alcanza tras el reset del computador (descrito en la primera columna), realice una simulación ciclo a

ciclo del comportamiento de la máquina rudimentaria suponiendo el siguiente contenido de la memoria (en cada columna debe indicarse el estado del controlador y valor de los registros en cada ciclo, cuando el valor no cambie respecto del ciclo anterior puede usarse el símbolo de igualdad):

dir. dato

00h 8003h
01h 0014h
02h 0000h
03h 1101h
04h D259h
05h B004h
06h 5002h

98. Supongamos que los tiempos de respuesta de los distintos elementos de la unidad de proceso de la máquina rudimentaria son los siguientes

- Multiplexores: 5ns
- Sumador de direcciones e incrementador conectado a la entrada del PC: 20ns
- Circuito de evaluación de la condición de salto: 10ns
- Lectura de un registro del banco de registros: 20ns
- Tiempo de respuesta de la ALU: 50 ns
- Tiempo de acceso a memoria: 60 ns

Sabiendo que el estado load es el que más tiempo consume:

a) Calcula la frecuencia máxima de reloj a la que puede funcionar el computador

b) Supongamos el siguiente bucle

```
bucle: load ...  
      add ...  
      sub ...  
      store ...  
      addi ...  
      bne bucle
```

que se ejecuta 1000 veces. En las condiciones del apartado a) determinar el tiempo que tardará en ejecutarse las 1000 iteraciones.

99. ¿Qué modificaciones habría que hacer en la ruta de datos y el controlador de la máquina rudimentaria para que la instrucción STORE también modificara flags? Razone la respuesta. Tendrán más valor aquellas soluciones que requieran menos modificaciones.

100. Suponiendo que el tiempo de respuesta de los diferentes elementos de la Unidad de Proceso de la MR es el siguiente:

- 10 ns para cada multiplexor
- 30 ns para el sumador de direcciones y el incrementador conectado al PC
- 20 ns para el bloque realiza la evaluación de la condición de salto
- 30 ns para leer un registro del banco de registros

- 80 ns el tiempo de respuesta de la ALU en caso de que realice una operación (si únicamente deja pasar la entrada B consume 10 ns)
- 110 ns el tiempo requerido para leer o escribir un dato en memoria RAM

Responda a las siguientes preguntas despreciando el tiempo que puede tardar en reaccionar la unidad de control.

- Indique la frecuencia máxima de reloj a la que puede funcionar la MR.
- El estado STORE se puede dividir en dos nuevos estados si incluimos un registro RD entre la salida del banco de registros y la entrada Min de memoria. Indique las acciones que se desarrollarían en cada uno de estos dos nuevos estados y el tiempo que consumirían.
- Indique si la inclusión del registro RD en el diseño para dividir en dos el estado STORE implica la necesidad de dividir algún otro estado del diagrama de estados simplificado.
- Indique la nueva frecuencia de trabajo.

101. Considere que el tiempo de respuesta de los diferentes elementos de la ruta de datos de la máquina rudimentaria es el siguiente:

- 10ns para los multiplexores
 - 20ns para el incrementador del PC y para el sumador de direcciones
 - 15ns para el extensor de signo
 - 5ns para la evaluación de la condición de salto
 - 30ns para leer un registro del banco de registros
 - 100ns el tiempo de respuesta de la ALU
 - 50ns para acceder a la memoria en operaciones de lectura y 110ns en operaciones de escritura
- Calcule el tiempo de ciclo mínimo necesario para que funcionen correctamente los estados ARIT y STORE de la máquina rudimentaria. Razone la respuesta.
 - Si se sustituye la memoria de la MR por una memoria con dos buses de direcciones, uno de lectura (M@L) y otro de escritura (M@E), esta nueva memoria permite leer y escribir sobre dos direcciones distintas simultáneamente. Dibuje la parte de la unidad de proceso que queda modificada por esta incorporación.
 - Tras la modificación anterior, ¿se puede introducir alguna nueva optimización a la máquina de estados de la MR? Razonar la respuesta. Indique las modificaciones que se producen en la tabla de salidas en la etapa afectada.

102. Suponiendo que el tiempo de respuesta de los diferentes elementos de la U.P. de MR es el siguiente:

- 15 ns. Para cada multiplexor
- 20 ns. Para el bloque que realiza la evaluación de la condición de salto
- 40 ns. Para el sumador de dir
- 30 ns. Para el incrementador del PC
- 20 ns. Para acceder a un dato del BR
- 50 ns. Para acceder a la memoria en lectura o escritura
- 70 ns. Para hacer una operación con la ALU

- Indique la frecuencia máxima de reloj a la que puede trabajar la Máquina Rudimentaria.
- El Registro RA se podría eliminar si tuviéramos un banco de registros con dos salidas para la lectura, ¿cómo afectaría esto al diagrama de estados de la Máquina Rudimentaria?.
- ¿Qué otros elementos se podrían modificar y/o eliminar con la acción propuesta en b)?
- Dibuje el diagrama de estados completo para la nueva situación.

103. Suponiendo que el tiempo de respuesta de los diferentes elementos de la Unidad de Proceso de la MR es el siguiente:

- 10 ns para cada multiplexor
- 40 ns para el sumador de direcciones y el incrementador conectado al PC
- 20 ns para el bloque realiza la evaluación de la condición de salto
- 30 ns para leer un registro del banco de registros
- 100 ns el tiempo de respuesta de la ALU en caso de que realice una operación (si únicamente deja pasar la entrada B consume 10 ns)
- 50 ns el tiempo requerido para leer o escribir un dato en memoria RAM

Responda a las siguientes preguntas despreciando el tiempo que puede tardar en reaccionar la unidad de control.

- Indique la frecuencia máxima de reloj a la que puede funcionar la MR.
- El estado ARIT se puede dividir en dos nuevos estados si incluimos un registro RB entre la salida del multiplexor SELDAT y la entrada B de la ALU. Indique las acciones que se desarrollarían en cada uno de estos dos nuevos estados y el tiempo que consumirían.
- Indique si la inclusión del registro RB en el diseño para dividir en dos el estado ARIT implica la necesidad de dividir algún otro estado del diagrama de estados simplificado.
- Indique la nueva frecuencia de trabajo.

104. En la siguiente tabla se puede ver el contenido inicial de los registros y de las primeras posiciones de memoria de la Máquina Rudimentaria:

Registros	Memoria
R1: A57Bh	00h: 0002h
R2: 0001h	01h: 000Ch
R3: 0005h	02h: 00FFh
R4: 0000h	03h: 8004h
R5: FF00h	04h: 0000h
R6: 011Bh	05h: 0000h
R7: 0C88h	06h: 0A02h

- Expresa en lenguaje ensamblador de la Máquina Rudimentaria la instrucción que se encuentra en la posición de memoria 06h.
- Si ejecutamos dicha instrucción, indica qué registros y/o posiciones de memoria se modifican y qué nuevo valor toman.
- Partiendo de la situación inicial mostrada, completa la siguiente tabla indicando los estados y los valores en hexadecimal que toman los registros

durante la ejecución de la instrucción (incluye también el primer estado de la ejecución de la instrucción siguiente).

Estado	FETCH						
IR	0000						
PC	06						
R@	0000						
RA	00						
RZ	0						
RN	0						

- c) Calcula el tiempo de ciclo mínimo necesario para que puedan ejecutarse los estados LOAD y ARIT, sabiendo que el tiempo de respuesta de los diferentes elementos de la ruta de datos de la máquina rudimentaria es el siguiente:
- 10ns para los multiplexores
 - 20ns para el incrementador del PC
 - 25ns para el sumador de direcciones
 - 5ns para la evaluación de la condición de salto
 - 30ns para leer un registro del banco de registros
 - 100ns el tiempo de respuesta de la ALU
 - 60ns para acceder a la memoria en operaciones de lectura y 120ns en operaciones de escritura

105. Considere que el tiempo de respuesta de los diferentes elementos de la ruta de datos de la máquina rudimentaria es el siguiente:

- 10 ns para el multiplexor SELDIR
- 20ns para los multiplexores SELREG y SELDAT
- 15ns para el incrementador del PC y para el sumador de direcciones
- 80ns para el extensor de signo
- 20ns para la evaluación de la condición de salto
- 40ns para leer un registro del banco de registros
- 120ns el tiempo de respuesta de la ALU
- 40 ns para una lectura de memoria y 100 ns para una escritura.

- a) ¿Cuál sería el período mínimo de la señal de reloj para poder ejecutar correctamente una instrucción de STORE ? (detalle el tiempo mínimo para cada etapa que puede atravesar una instrucción de STORE). Justifique su respuesta.
- b) Se ha decidido dotar a la MR de un segundo módulo de memoria. Este segundo módulo dispone de las mismas señales (con la misma funcionalidad) que la memoria original de la MR: Min, M@, Mout y L/E. Así mismo, se decide que todas las instrucciones se almacenarán en una de las memorias (MEM1 a partir de ahora), mientras que los datos se almacenarán en la otra (MEM2). Dibuje la parte de la unidad de proceso que queda modificada por esta incorporación.
- c) Tras la modificación anterior, ¿se puede introducir alguna nueva optimización a la máquina de estados de la MR? Indique, utilizando la tabla adjunta el valor de la señales L/E1 (de la MEM1) y L/E2 (de la MEM2),

Ld_R@, Ld_PC y Ld_IR para cada uno de los estados de la nueva máquina de estados.

- d) Suponiendo que una lectura a la memoria de datos tarde 70 ns y una escritura tarde 150 ns, ¿cuál será ahora el período mínimo de la señal de reloj para poder ejecutar una instrucción de STORE?

	FETCH	DECO	ARIT	BRANCH	LOAD	STORE	
L/E ₁							
L/E ₂							
Ld_R@							
Ld_PC							
Ld_IR							

106. ¿Cómo podríamos reducir el número de ciclos necesario para ejecutar la instrucción de LOAD en la MR? (Aunque haya que modificar otras cosas).

107. Incluimos en el diseño de la Máquina Rudimentaria una nueva memoria de tal modo que el almacenamiento de datos e instrucciones se realiza en memorias diferentes. Teniendo en cuenta esta mejora de diseño, ¿se podría reducir el número de ciclos necesarios para ejecutar cada tipo de instrucción (LOAD, STORE, BRANCH, ARIT)? Razona la respuesta.

108. Se quiere eliminar del repertorio de la MR la instrucción ASR e insertar una nueva instrucción que compare dos operandos fuente (CMP Rf1, Rf2). La nueva instrucción debe tener también una versión con operando inmediato (CMPI Rf1, #inmed). Cuando se ejecute la instrucción el resultado será el siguiente:

- El bit Z toma el valor 1 si los dos operandos son iguales, y toma el valor 0 si los dos operandos son distintos.
- El valor que tome el bit N es irrelevante.
- No se modifica ningún registro del Banco de Registros.

- a) Define el formato de las nuevas instrucciones de tal forma que no sea necesario modificar el diagrama de estados de la MR. Razona las decisiones tomadas.
- b) Diseña la ALU de forma que sea posible ejecutar las nuevas instrucciones. Diseña la conexión de la ALU con el resto de la MR.

109. Se desea añadir al juego de instrucciones de la MR las siguientes instrucciones aritméticas: 1) una instrucción de desplazamiento aritmético a la izquierda con un operando fuente almacenado en un registro y el resultado en un registro destino (ASL Rf, Rd); 2) una instrucción de test que activa los indicadores de condición según el valor almacenado en un registro fuente y escribe ese mismo valor en un registro destino (TST Rf, Rd).

- a) ¿Qué implicaciones tiene en el formato de las instrucciones el añadir estas instrucciones?
- b) Diseña la nueva ALU y rediseña el hardware que rodea la ALU.

110. Sea el siguiente programa en pseudocódigo, donde la instrucción *swap(x, y)* intercambia los valores de las variables *x* e *y*.

a) Traducir el siguiente programa a lenguaje ensamblador de la Máquina Rudimentaria, suponiendo que se almacena a partir de la dirección 00h.

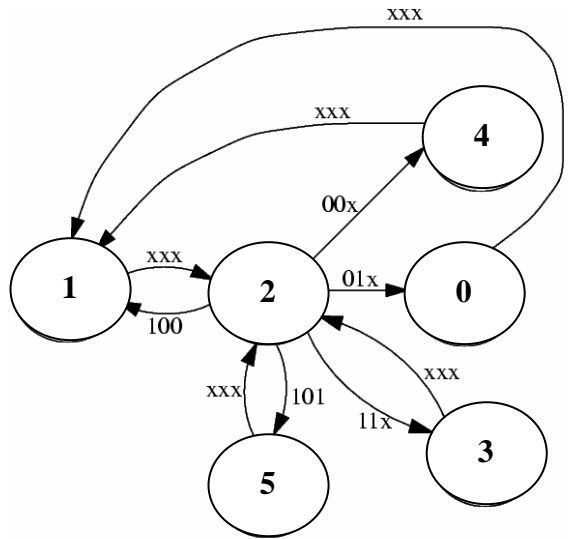
```

a:= 7;
b:= 5;
while b < a do
    c:= (b + a)/2;
    if c > a then
        swap
(c,a);
else

```

b) Supongamos que un estudio de los programas ejecutados en la MR nos aconseja implementar la instrucción swap, para lo cual se utiliza una de las codificaciones libres de las instrucciones aritméticas y se añaden dos registros a la ALU. Indicar como afectaría esto al diagrama de estados simplificado, a las entradas del banco de registros, al registro RA ...

111. El diagrama siguiente representa una posible codificación de la unidad de control de la máquina rudimentaria. Implemente la unidad de control mediante un contador módulo 8 como los vistos en clase (con señal de Load prioritaria sobre la de Contar y Reset asíncrono). La máquina secuencial sólo debe representar las transiciones de estados (no es necesario implementar las salidas de la UC en cada estado) en función de las señales Cond, OP15 y OP14.



112. Se desea añadir al juego de instrucciones de la máquina rudimentaria las siguientes 2 instrucciones DIV2 Rf,Rd en donde $Rd = Rf/2$ y MUL2 Rf,Rd en donde $Rd = Rf*2$.

- Codificar estas dos nuevas instrucciones.
- Diseñar la nueva ALU.
- ¿Se puede utilizar alguna codificación ya existente para alguna de las dos instrucciones?

113. En la MR cuando se tenía que realizar una operación aritmético-lógica se accedía al segundo registro o al inmediato y se realizaba la operación, a la vez que se cargaba el resultado en el banco de registros. Supongamos que la ALU tarda demasiado tiempo en obtener el resultado y no se puede cargar el resultado en ese ciclo de reloj y ha de hacerse en el siguiente. Conteste razonando a las siguientes preguntas:

- a) ¿Se necesita añadir un estado más para cargar el resultado en el banco de registros?
- b) ¿Se necesita añadir un registro a la salida de la ALU?
- c) ¿Se puede encadenar la carga del registro en el banco de registros con la lectura del registro de la siguiente instrucción?

114. En los computadores comerciales el acceso a memoria es siempre superior a 1 ciclo debido al tamaño de memoria. Suponiendo que en la Máquina Rudimentaria el acceso a memoria supusiese 2 ciclos de reloj ¿cómo afectaría esto al diagrama de estados simplificado?

115. Se desea diseñar un procesador a partir del creado para la Máquina Rudimentaria con el mismo repertorio de instrucciones, en el que la instrucción LOAD no actualice los indicadores de condición N, y Z y el tiempo de ejecución de los estados afectados sea mínimo.

- a) ¿Qué componentes y buses de la Unidad de Proceso se verían afectados? Indique las modificaciones necesarias.
- b) Indique qué estado/s del diagrama simplificado se verían afectados por dicha modificación ¿Qué salidas de la Unidad de Control cambiarían en dichos estados?

116. El repertorio de instrucciones de la máquina rudimentaria sólo permite cargar en registro datos desde la memoria (instrucción LOAD A(Ri), Rd). Se quiere añadir una nueva instrucción, LOADI #n, Rd que permita almacenar en registro datos desde la instrucción (operando inmediato).

- a) Redefine el formato de las instrucciones teniendo en cuenta la nueva instrucción de tal forma que no sea necesario modificar ni el diagrama de estados de la MR, ni el tamaño de la memoria ni el tamaño del banco de registros. ¿Se conserva el tamaño de la instrucción?. (1 puntos)
- b) ¿Qué modificaciones hay que hacer en la unidad de proceso (ruta de datos) de forma que sea posible ejecutar la nueva instrucción?. ¿Afecta al tiempo de la fase LOAD?. Razona tus respuestas. (1 puntos)
- c) Se quiere modificar el diagrama de estados de la MR de manera que la nueva instrucción se ejecute en el menor número de ciclos posibles. Para ello ponemos un nuevo estado llamado LOADI.
 - Señala en el dibujo de la unidad de proceso las componentes, señales de control y buses internos que se usan en dicho estado. (1 puntos)
 - Dibuja como queda el nuevo diagrama de estados (0,5 puntos)
 - Cuantos ciclos tarda ahora la instrucción LOADI (0,5 puntos)

117. En el diseño de la máquina rudimentaria se desea eliminar el módulo incrementador (+1) que antecede al registro PC y, en su lugar, reutilizar para

este propósito el sumador (+) de direcciones que antecede al registro R@. De esta manera se tendría un único sumado que podría realizar tanto la suma $R_{x7..0} + IR_{7..0}$ como la suma (salida de SELDIR) + 1. Indique las modificaciones que sería necesario hacer (en unidad de proceso y en la unidad de control) para que esto sea posible.

118. Se quiere cambiar la memoria de la maquina rudimentaria por dos memorias, una de datos y otra de instrucciones.

- a) Dibuja las modificaciones que hay que hacer en la unidad de proceso (ruta de datos).
- b) ¿Se puede hacer alguna optimización en el diagrama de estados de la MR? Justifica la respuesta y si es que SI, dibuja como queda el nuevo diagrama de estados

119. Las instrucciones de acceso a memoria de la máquina rudimentaria utilizan, para indicar la dirección de memoria, una dirección base codificada explícitamente en la instrucción y una referencia a un registro del banco de registros. Esquemáticamente la funcionalidad de dichas instrucciones es:

- LOAD $n(R_i), R_d$: $M[n+R_i] \rightarrow R_d$, $PC+1 \rightarrow PC$, modifica flags según el valor cargado
- STORE $R_f, n(R_i)$: $R_f \rightarrow M[n+R_i]$, $PC+1 \rightarrow PC$

Se desea modificar ambas instrucciones de manera que la dirección base explícita se reemplace por otra referencia a un registro del banco de registros, quedando la funcionalidad como sigue:

- LOAD $(R_b, R_i), R_d$: $M[R_b+R_i] \rightarrow R_d$, $PC+1 \rightarrow PC$, modifica flags según el valor cargado
- STORE $R_f, (R_b, R_i)$: $R_f \rightarrow M[R_b+R_i]$, $PC+1 \rightarrow PC$

Se pide:

- a) Modifique adecuadamente el formato de instrucción de ambas instrucciones.
- b) Modifique adecuadamente la ruta de datos para que se puedan realizar las nuevas funcionalidades. Tendrán más valor aquellas soluciones que requieran menos modificaciones.
- c) Sin modificar los estados actuales del controlador, añada los estados que crea necesarios (indicando los valores de las señales de control) de manera que se puedan incorporar las nuevas funcionalidades.