ELECTRÓNICA DIGITAL **Primera convocatoria 2024-2025**

1	2	3	4

Apellidos:.....SOLUCIÓN.....

Nombre:.....

TEORÍA (Cada pregunta vale 1 punto)

1.- Atendiendo a los datos de la tabla siguiente:

HCT		AC	
V_{OH} = 2,4V	V_{IH} = 2,0V	V_{OH} = 4,4V	V_{IH} = 3,5V
V_{OL} = 0,4V	V_{IL} = 0,8V	V_{OL} = 0,5V	V_{IL} = 1,5V
$\|I_{OH}\|$ = 15 mA	$\|I_{IH}\|$ = 20 uA	$\|I_{OH}\|$ = 24 mA	$\|I_{IH}\|$ = 1 uA
$\|I_{OL}\|$ = 24 mA	$\|I_{IL}\|$ = 0,1 mA	$\|I_{OL}\|$ = 24 mA	$\|I_{IL}\|$ = 1 mA

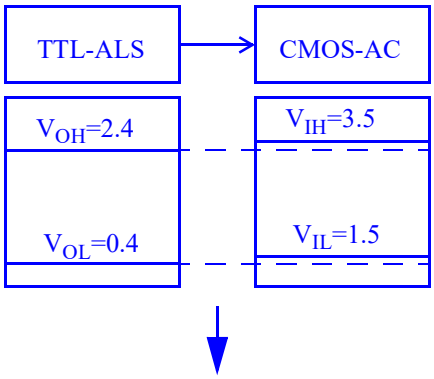
a) Defina brevemente V_{OH}, V_{OL}, V_{IH}, V_{IL}, I_{OH}, I_{OL}, I_{IH}, I_{IL}.
b) Explique las condiciones para que dos familias lógicas sean compatibles.
c) Indique si las familias HCT y AC lo son.

SOLUCIÓN

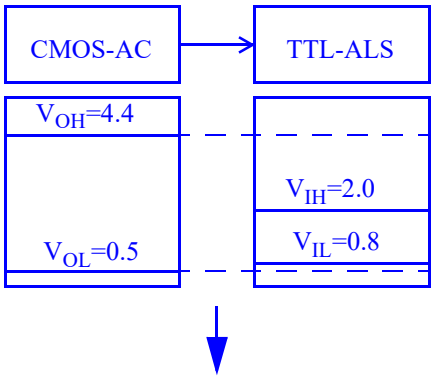
Para que sean compatibles se tiene que cumplir que: $V_{OH} > V_{IH}$, $V_{OL} < V_{IL}$, $I_{OH} > I_{IH}$, $I_{OL} > I_{IL}$.

La comprabación debe ser bidireccional. Es decir, cuando las puertas HCT controlan puertas AC y viceversa. Esto nos da que hay que comprobar 8 condiciones. Si alguna de las 8 condiciones falla entonces no son compatibles.

Compatibilidad en tensión



NO COMPATIBLE



COMPATIBLE

Para el caso de salidas HCT conectadas a entradas AC no se cumple que $V_{OH} > V_{IH}$. Esto significa que los unos lógicos generados por las puertas de la familia TTL-ALS no son interpretados correctamente por las de la familia CMOS-AC. Por lo tanto ambas familias no son compatibles en tensión y ello conlleva que no sean compatibles.

Ya no es necesario analizar la compatibilidad en intensidad.

CRITERIO DE CORRECCIÓN

- Definiciones 20%
- Condiciones 40%
- Comprobación HCT-AC 40%

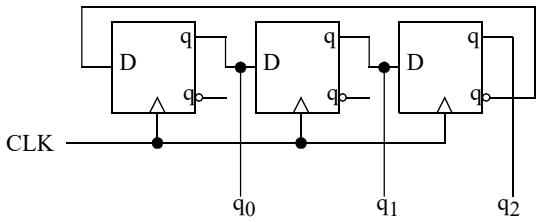
ELECTRÓNICA DIGITAL **Primera convocatoria 2024-2025**

1	2	3	4

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

2.- Analice el circuito de la figura e indique qué hace. Suponga que inicialmente los biestables tienen el estado 0.



SOLUCIÓN

Análisis 1:

Al tratarse de un circuito tan sencillo, podemos hacer un análisis modular. La evolución de los estados ($q_{2:0}$) sería: 000, 001, 011, 111, 110, 100 y vuelta a empezar. Es una especie de registro de desplazamiento circular en el que la realimentación se realiza con la salida invertida. Lo que se obtiene es un contador módulo 6 que se llama contador Jhonson.

Análisis 2: aplicando la metodología genérica vista en clase, tenemos

- Ecuaciones de Excitación/Salida:

$$D_0 = \overline{q_2} \qquad D_1 = q_0 \qquad D_2 = q_1$$

- Tabla de Excitación/Salida:

q_0 \ q_2q_1	00	01	11	10
0	001	101	100	000
1	011	111	110	010

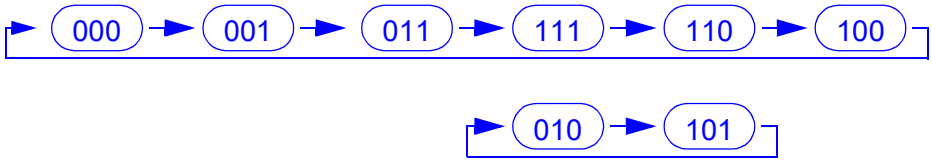
$D_{2:0}$

- Tabla de Transición/Salida:

q_0 \ q_2q_1	00	01	11	10
0	001	101	100	000
1	011	111	110	010

$Q_{2:0}$

- Diagrama de Estados/Salida:



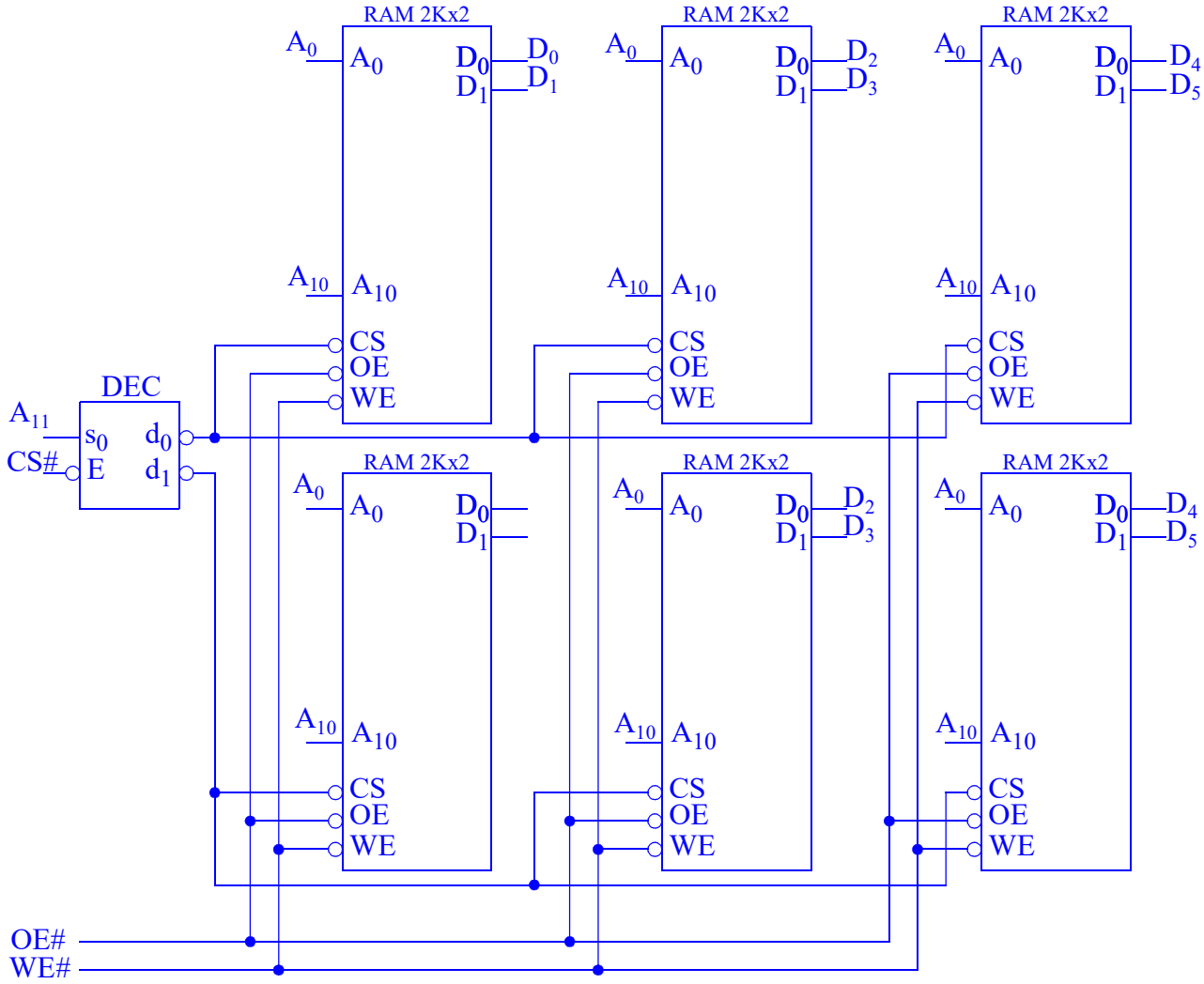
Se observa que salen dos ciclos de estados. Uno de 6 y otro de 2. Para el valor inicial que nos dan, 000, sería de aplicación el primer ciclo.

3.- Obtenga un módulo de RAM de 4Kx6 usando chips de 2Kx2 y los elementos adicionales que pueda necesitar.

SOLUCIÓN

Se necesita ampliar tanto en anchura de palabra como en longitud (número de palabras). Para pasar de una anchura de 2 a 6 bits se necesitarán 3 chips obteniendo

un módulo de 1Kx6. Como, además, hay que duplicar el número de palabras, se necesitará otro módulo como el anterior, todo ello controlado por un decodificador gobernado por la señal de direcciones de mayor peso.



4.- Compare las dos arquitecturas de referencia de computador estudiadas.

ELECTRÓNICA DIGITAL **Primera convocatoria 2024-2025**

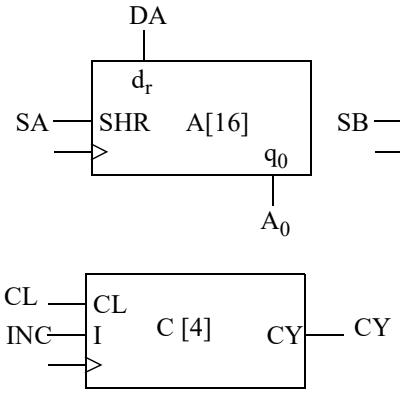
1	2	3	4

Apellidos:.....SOLUCIÓN.....

Nombre:.....

PROBLEMAS (Cada pregunta vale 3 puntos)

1.- Para la unidad de datos de la figura:



Restador completo

Ci A B	Co R
0 0 0	0 0
0 0 1	1 1
0 1 0	0 1
0 1 1	0 0
1 0 0	1 1
1 0 1	1 0
1 1 0	0 0
1 1 1	1 1

a) Describa a nivel RT los componentes de la UD.

b) Obtenga las cartas ASM de datos y control de una UC que, dados dos números sin signo en A y B, deja en A el mayor y en B el menor.

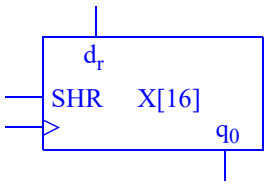
c) Realice la implementación de la UC usando la técnica de un biestable por estado.

NOTA: Se incluye la tabla de verdad del restador completo.

SOLUCIÓN

a) Descripción RT:

Registro X (A y B):



SHR	X←	q_0=
0	X	$[X_0]$
1	SHR (X, d_r)	$[X_0]$

Registro CONT:

CL I	CONT←	CY=
1 -	0	1 sii [CONT]=15
0 1	CONT+1	
0 0	CONT	

b) Es necesario comparar A y B para saber si los números están bien ubicados o no. Si no lo están se hará el intercambio A↔B. En caso contrario se ha terminado.

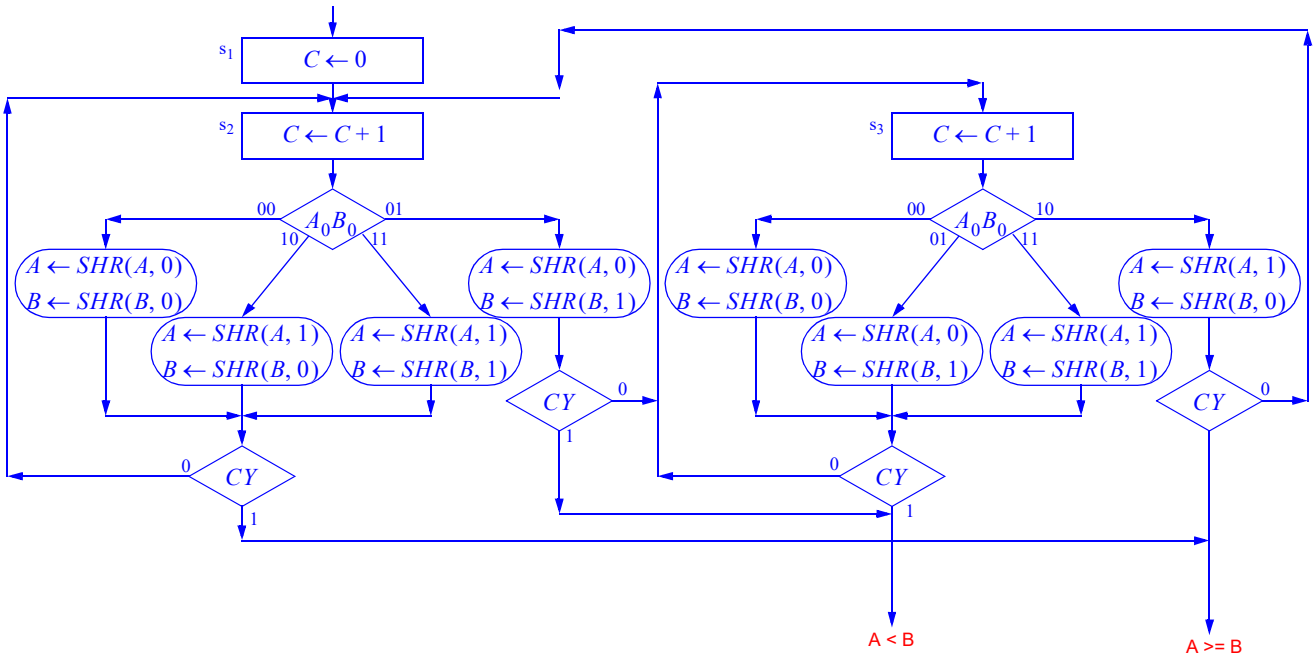
La comparación se puede hacer de dos formas:

- Empezando por el msb, la UC compara los bits de A y B. Si son distintos, se tiene un veredicto. Si no, se procesan los siguientes bits (en orden del msb al lsb). Si se termina el procesamiento y todos los bits han sido iguales, los números son iguales y, en nuestro caso, estarían bien ordenados. Esta opción no se puede usar porque se requiere procesar los bits empezando por el msb y en la UD sólo se accede a

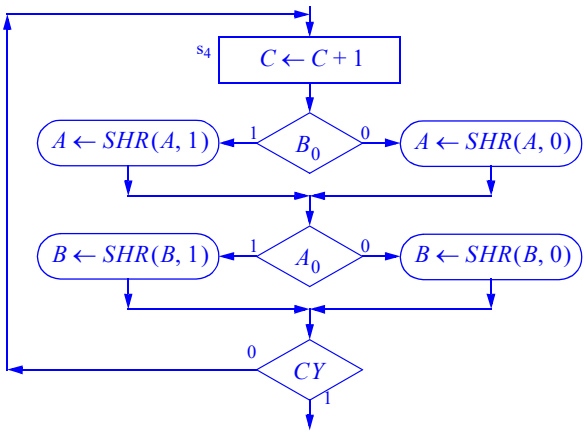
los lsb.

- Realizar la resta A-B, que procede por los lsb, con lo cual es implementable en nuestra UD. No es necesario almacenar el resultado. Todo lo contrario, hay que conservar los valores de A y B, por lo que habrá que reinsertar los bits que vayan saliendo para dejarlo todo como estaba. Lo que sí que hay que obervar es el carry de salida. Si es 0, $A-B\geq 0$, lo que implica que $A\geq B$. En caso contrario, $A-B<0$ y $A<B$.

La resta se hace con un par de estados (s_2 y s_3) que implementan cada uno una mitad de la tabla de verdad del restador completo. s_2 recuerda que el carry de entrada es 0 y s_3 que es 1. En cada estado se calcula el carry de salida (que determina el próximo estado) y se reinsertan los A_0 y B_0 para conservar los registros. Después de 16 iteraciones el último carry indica si $A\geq B$ (c_O=0) o $A<B$ (c_O=1) y el contador vuelve a estar a 0, listo para hacer el intercambio si hace falta.



Para intercambiar el contenido de los dos registros, basta preguntar por lsb del otro registro e introducirlo por la izquierda al hacer el desplazamiento a la derecha. Dicho desplazamiento deja expuesto el siguiente bit en el lsb para la siguiente iteración.



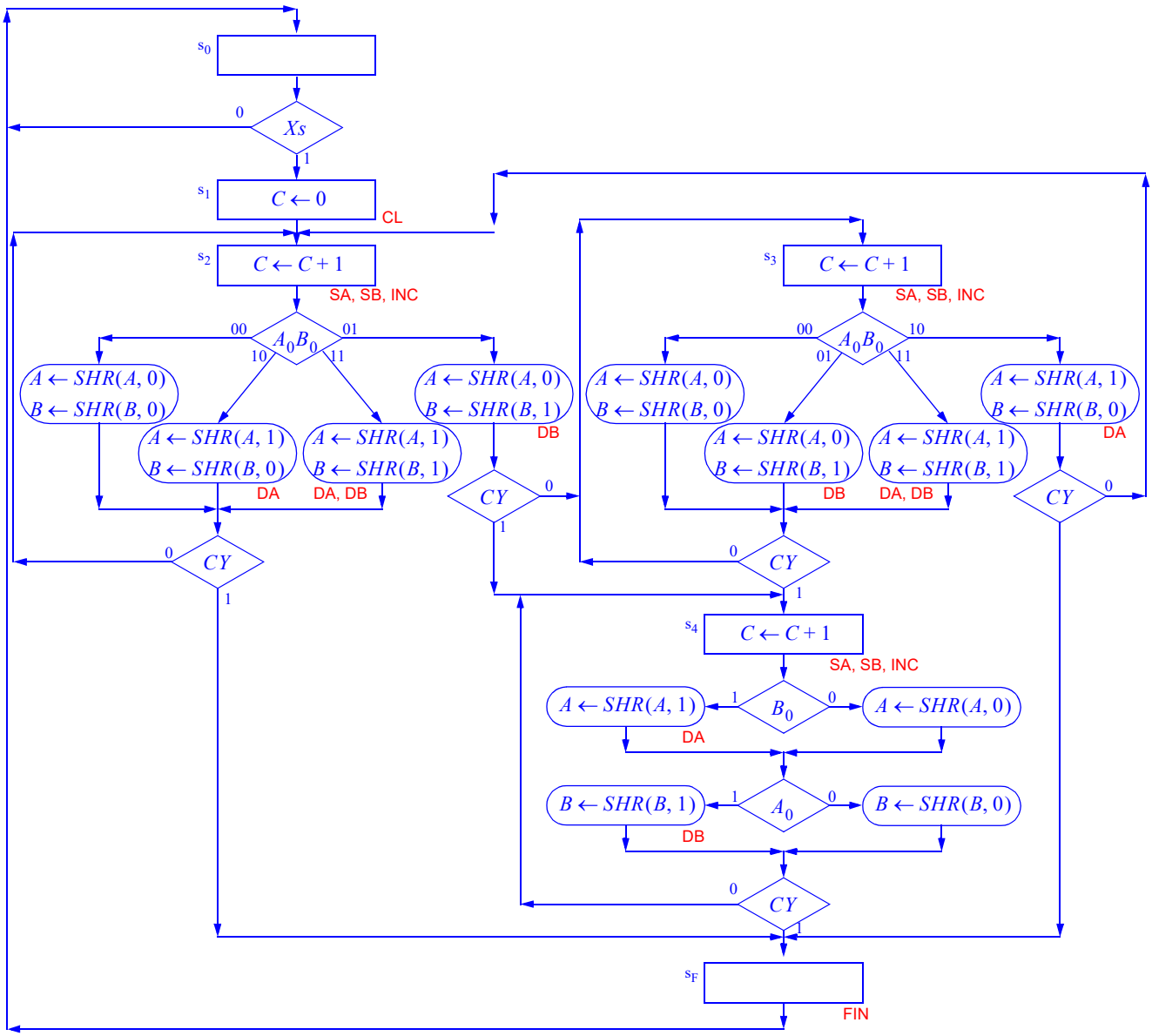
ELECTRÓNICA DIGITAL **Primera convocatoria 2024-2025**

1	2	3	4

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

Uniéndolo todo nos queda (en rojo los datos de la carta ASM de la UC):



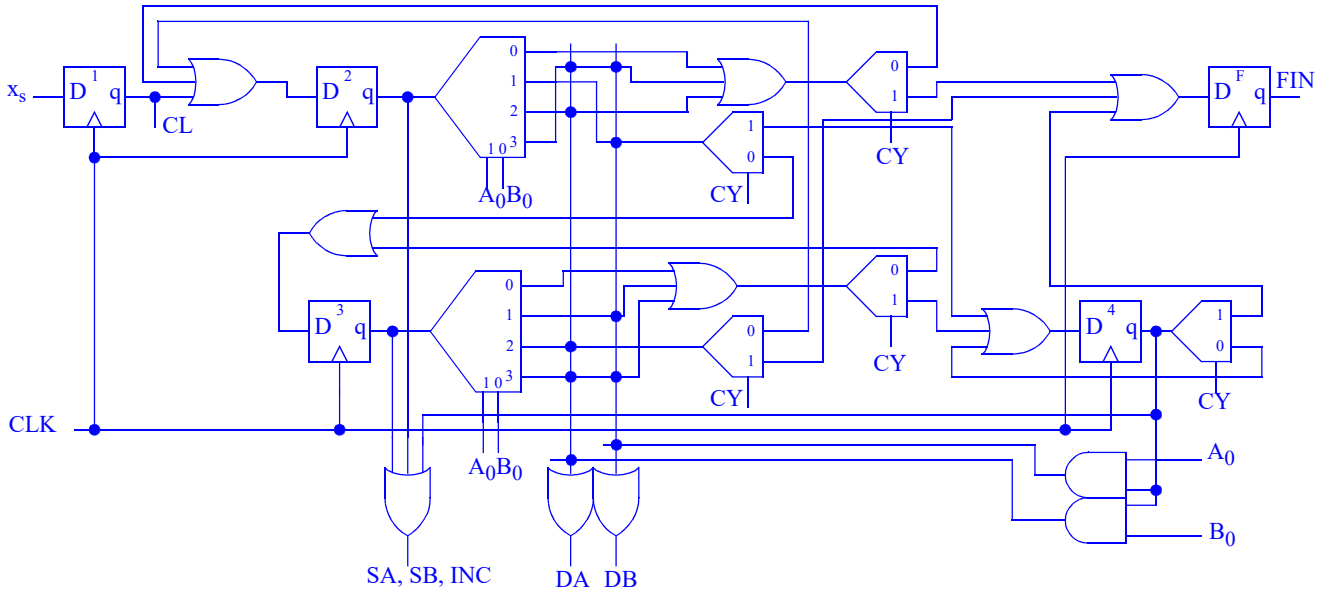
ELECTRÓNICA DIGITAL **Primera convocatoria 2024-2025**

1	2	3	4

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

Implementación de la unidad de control:



CRITERIO DE CORRECCIÓN

a) 10% descripción RT
b) 40% comparación
c) 20% intercambio
d) 30% implementación

2.- Se desea incluir en el conjunto de instrucciones del CS3 la instrucción `DBNZ Rf, dir` que decrementa el registro `Rf` en 1 y salta si no es cero a la dirección dir (*D*ecrement and *B*ranch if *N*ot *Z*ero). En caso contrario, se ejecuta la siguiente instrucción. Se pide (justifique las respuestas):

a) Indique si es necesario o no modificar la UD del CS3. En caso afirmativo, detalle los cambios.
b) Proponga un código de operación para la misma, así como un formato de codificación.
c) Indique la secuencia de microoperaciones del ciclo de ejecución, así como las señales que se activarían en cada caso.
d) Use la instrucción en la subrutina `TestMem` que comprueba un bloque de la memoria de datos del sistema. Recibe en R0 la dirección de inicio y en R1 el número de palabras a comprobar. Para cada posición de memoria, realiza lo siguiente:

- Salva el contenido de la posición actual en un registro temporalmente.
- Escribe en la memoria el número $55.
- Comprueba que se lee de la posición el dato escrito.
- Escribe en la memoria el número $AA.
- Comprueba que se lee de la posición el dato escrito.
- Restaura el valor que había en dicha posición de memoria.
- Comprueba que se lee de la posición el dato escrito.

En caso de que falle alguna de las comprobaciones, se sale devolviendo en R0 el código -1. En caso contrario, devuelve 0.

ELECTRÓNICA DIGITAL **Primera convocatoria 2024-2025**

1	2	3	4

Apellidos:.....SOLUCIÓN.....

Nombre:.....

SOLUCIÓN

a) Sí es necesario modificar la UD para poder hacer la resta del registro. Diversas soluciones se pueden aplicar: por una parte se puede modificar la ALU para poder hacer la operación decremento. También se podría modificar el MUX del canal B de la ALU para que se pudiera seleccionar la constante 1 como una de sus opciones. Estimamos la primera como más barata, ya que sólo se necesita hacer la operación A-B con B=0 y Ci=1. El código para hacer la resta es $OP_{3:0}$=101-. Se podría dejar el código 1010 para A-B y el código 1011 para A-1.

b) El código para la instrucción debe ser uno de los disponibles y, si es posible, diferenciarse en un sólo bit de las instrucciones más parecidas (`SUBI` 11010 y `BRcc` 00110). Los valores posibles son 11110, 10110, 01110. Escojamos, por ejemplo, la última. El formato de instrucción sería el C, pero almacenando en los bits del 8 al 10 el número del registro en lugar de la condición.

c)

1.- AC ← REG [$IR_{10:8}$]-1	OP_3, OP_1, OP_0, W_{AC}, W_S
2.- AC ← $IR_{7:0}$	OP_3, OP_2, W_{AC}, INM
3.- $\overline{Z}$: PC ← AC	$\overline{Z}$: W_{PC}, R_{AC}

d)

```
;-------------------------------------------------------------------------------
; TestMem                                                                 v1.0
;
; Comprueba un bloque de la memoria de datos del sistema.
; Para ello escribe y comprueba que se puede leer los contenidos $55 y $AA.
; El contenido de la memoria no es modificado.
; Entradas:
;   R0: dirección de inicio
;   R1: número de palabras
; Salidas:
;   R0: -1 en caso de error. 0 si todo fue bien.
;-------------------------------------------------------------------------------
TestMem:   ldi       r3, $55    ;R3 = constante 1
           ldi       r4, $AA    ;R4 = constante 2
TestMBuc:  ld        r2, (r0)   ;Leer byte y guardar en R2
           st        (r0), r3   ;Grabar primera constante
           ld        r5, (r0)   ;Leerla
           cp        r5, r3     ;Son iguales?
           breq      TestMSig1  ;...sí, seguir
           jmp       TestMErr   ;...no, error
TestMSig1: st        (r0), r4   ;Grabar segunda constante
           ld        r5, (r0)   ;Leerla
           cp        r5, r4     ;Son iguales?
           breq      TestMSig2  ;...sí, seguir
           jmp       TestMErr   ;...no, error
TestMSig2: st        (r0), r2   ;Grabar valor inicial
           ld        r5, (r0)   ;Leerlo
           cp        r5, r2     ;Son iguales?
           breq      TestMSig3  ;...sí, seguir
           jmp       TestMErr   ;...no, error
TestMSig3: addi      r0, 1      ;Avanzar puntero
           dbnz      r1, TestMBuc;Descontar palabra y seguir si quedan
           ldi       r0, 0      ;R0=código OK
           jmp       TestMFin   ;Salir
TestMErr:  ldi       r0, -1     ;R0=código Error
TestMFin:  ret
```

CRITERIO DE CORRECCIÓN

ELECTRÓNICA DIGITAL **Primera convocatoria 2024-2025**

1	2	3	4

Apellidos:.....SOLUCIÓN.....

Nombre:.....

a) 20%
b) 10%
c) 30%
d) 40%