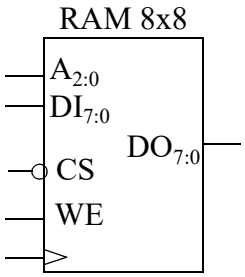
ELECTRÓNICA DIGITAL. **CONTROL 2. Curso 2024-2025**

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

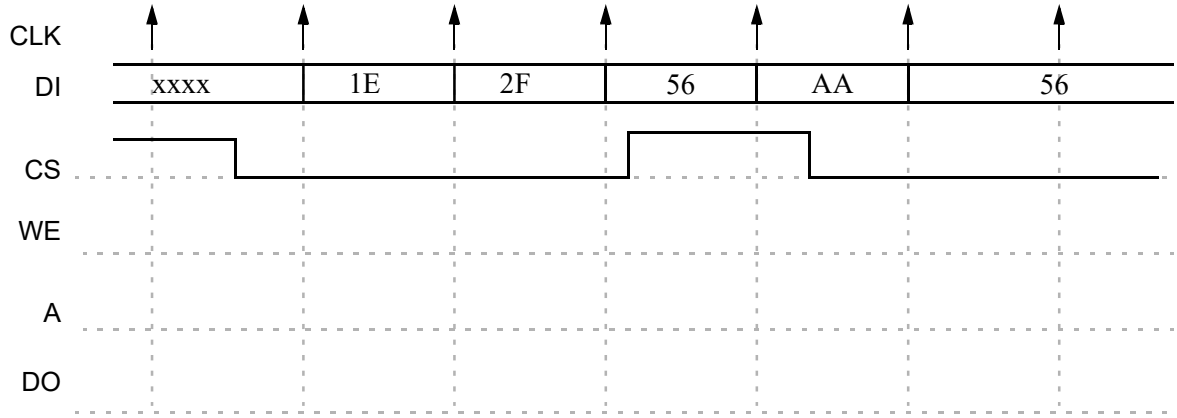
1	2	3	4

1.- **[2 puntos]** Dispone de una RAM de 8 filas y 8 bits cada fila, cuyo funcionamiento se muestra en la tabla adjunta. Se sabe además que la escritura es síncrona (flanco de subida del reloj) pero la lectura es asíncrona (cuando se pone la dirección A, se lee el valor contenido en la dirección de memoria).

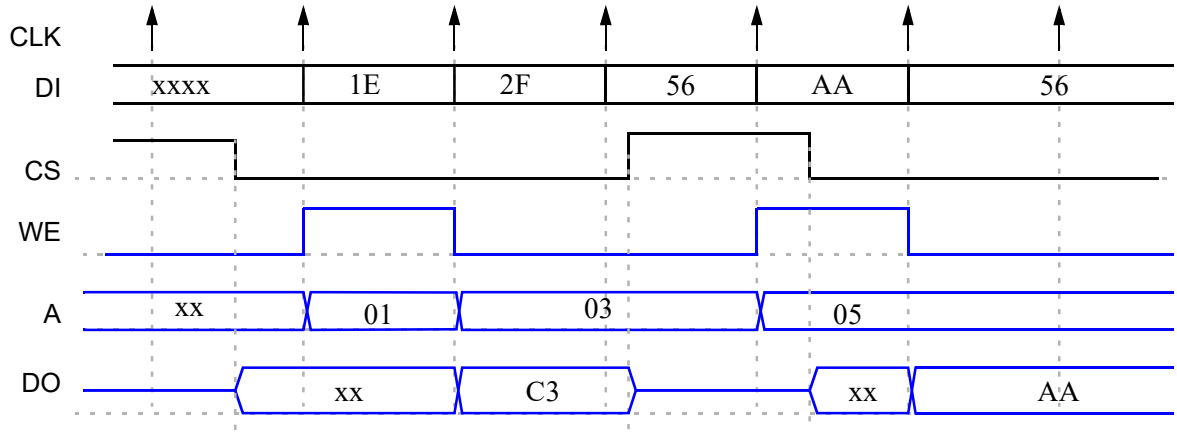


CS#	WE	DO=	RAM ←
1	X	HiZ	RAM
0	0	RAM(A)	RAM
0	1	RAM(A)	RAM(A) ← DI

En el cronograma adjunto cumplimente los valores de las señales WE y A para que el valor $1E_{16}$ se escriba en la dirección 1 y el valor AA_{16} se escriba en la dirección 5 de memoria. No escriba en ninguna otra posición de memoria. Suponiendo que la dirección 3 contiene el valor $C3_{16}$, ponga los valores para leerla. Se desconoce el contenido del resto de posiciones de la memoria. Rellene también el valor de DO en todos los instantes. Las señales WE y A sólo pueden cambiar con el flanco activo del reloj.



SOLUCIÓN



CRITERIO DE CORRECCIÓN

- Escritura en la dirección 1: 25%
- Escritura en la dirección 5: 25%
- Lectura de la posición 3: 25%

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

3	4

- Resto de ciclos: 25%

2.- **[2 puntos]** Enumere y explique los modos de direccionamiento disponibles en el CS3. Indique un ejemplo de instrucción que use cada uno de ellos y descomponga en microoperaciones el ciclo de ejecución de una de ellas (sólo transferencias).

SOLUCIÓN

- Directo a registro: el operando está en un registro del procesador. MOV R0, R1
- Absoluto o directo de memoria: el operando está en la memoria de datos. En la instrucción se especifica la dirección efectiva de dicha posición de memoria. LDS R0, N
- Indirecto de registro: el operando está en la memoria de datos. En la instrucción se especifica el registro que contiene la dirección efectiva de dicha posición de memoria. LD R0, (R1)
- Inmediato: el operando está en los 8 bits menos significativos de la instrucción. LDI R0, N

Microoperaciones de MOV R0, R1:

AC ← REG[$IR_{2:0}$]

REG[IR10:8] ← AC

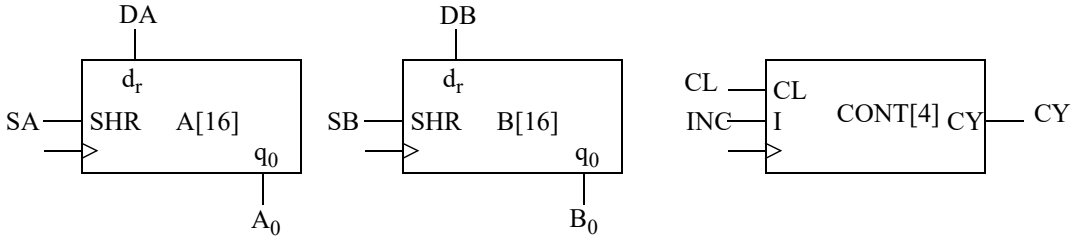
CRITERIO DE CORRECCIÓN

- Modos de direccionamiento, 80%. Cada modo de direccionamiento, 25%. 70% para la descripción y 30% para el ejemplo.
- Microoperaciones, 20%

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

3.- **[4 puntos]** Para la unidad de datos de la figura:



a) Obtenga las cartas ASM de datos y control de una UC que haga la instrucción A ← SHUFFLE (A, B) y B ← SHUFFLE (B, A) cuando I=0 y A ← A AND B y B ← A OR B cuando I=1.

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

NOTA: La macro SHUFFLE (x, y) devuelve un número que contiene los bits impares de x y los pares de y alternados. Suponiendo 4 bits, si x = $x_3x_2x_1x_0$ e y = $y_3y_2y_1y_0$, SUFFLE (x, y) = $x_3y_2x_1y_0$.

SOLUCIÓN

a) $\bar{I}$: A ← SHUFFLE (A, B) y B ← SHUFFLE (B, A); I: A ← A AND B y B ← A OR B

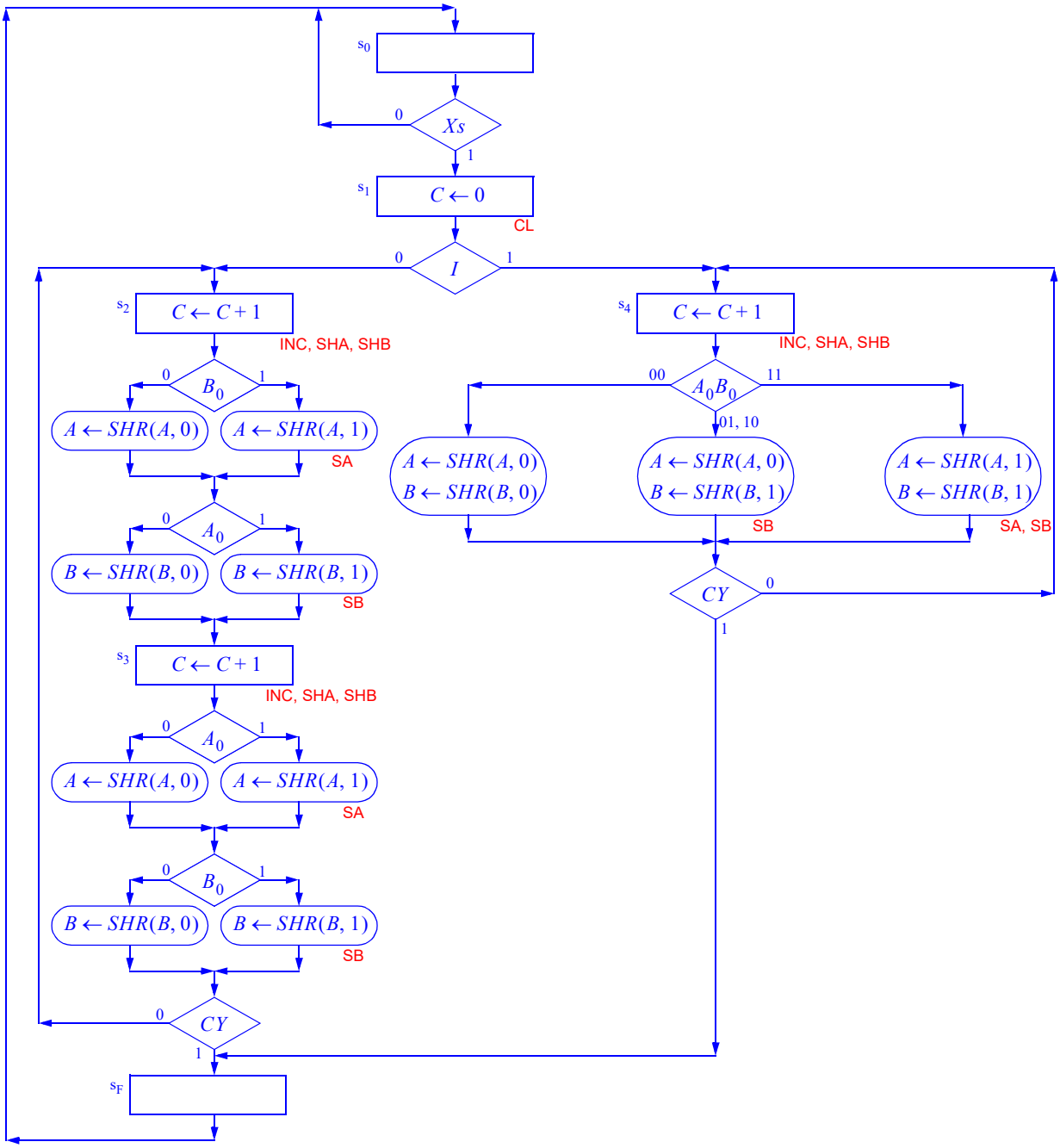
El procedimiento para las dos instrucciones es el mismo: desplazar A y B en cada ciclo para acceder al siguiente bit menos significativo y hacer hueco para el nuevo bit que se inserta. La Unidad de Control será la encargada de calcular los valores de SA y SB para rellenar los registros con el valor apropiado. En cada estado se incrementará el contador que se ha puesto a 0 en la inicialización y se inspeccionará el carry en cada vuelta del bucle.

En el caso de la mezcla, se necesitarán 2 estados en cada vuelta del bucle para coger, en el primero de ellos el bit del segundo operando (estado s2) y en el segundo el primer operando (s3).

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

En el caso de las operaciones lógicas se procede directamente: el estado s4 escanea A0 y B0 y calcula directamente Sa y SB según convenga.



CRITERIO DE CORRECCIÓN

a) 70%. I=0 60% e I=1 40%

b) 30%

4.- **[2 Puntos]** Escriba la subrutina `FiltraVector` en ensamblador del CS3 que recibe en R0 la dirección de un vector de bytes sin signo, en R1 el número de elementos del vector, en R2 el valor mínimo y en R3 el máximo. La subrutina recorre el vector y sustituye todos los valores menores al mínimo por el mínimo y los mayores al máximo por el máximo, dejando el resto inalterados.

ELECTRÓNICA DIGITAL. **CONTROL 2. Curso 2024-2025**

Apellidos:.....SOLUCIÓN.....

Nombre:.....

SOLUCIÓN

```
;-------------------------------------------------------------------------------
; FiltraVector                                                          v1.0
;
; Sustituye todos los valores menores al mínimo por el mínimo y los mayores
; al máximo por el máximo.
; Entradas:
;    R0: Puntero al vector
;    R1: Longitud del vector
;    R2: Valor mínimo
;    R3: Valor máximo
; Salidas:
;    Nada
;-------------------------------------------------------------------------------
FiltraVector: ld   r4, (r0)    ;Leer elemento
              cp   r4, r2      ;Elemento < min?
              brcs FiltraMin   ;...sí, filtrar valor
              cp   r3, r4      ;...no. max < elemento?
              brcs FiltraMax   ;...sí, filtrar valor
FiltraSig:    addi r0, 1       ;...no, Avanzar puntero
              subi r1, 1       ;Contabilizar elemento. Quedan?
              brzs FiltraFin   ;...no, salir
              jmp  FiltraVector;...sí, siguiente elemento
MaximoMin:    st   (r0), r2    ;Sustituir por mínimo
              jmp  FiltraSig   ;seguir proceso
MaximoMax:    st   (r0), r3    ;Sustituir por máximo
              jmp  FiltraSig   ;seguir proceso
FiltraFin:    ret
```

CRITERIO DE CORRECCIÓN

No poner comentarios: -2