

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

1	2	3	4

Duración 2:00 h.

1.- [2 puntos]

- a) Describa qué es y cómo funciona una memoria LIFO.
- b) Describa el funcionamiento de la memoria de pila del CS3. Que funcionalidad añade, que modificaciones hay que introducir sobre la arquitectura del CS2 qué instrucciones son las involucradas, cómo se accede a la pila y similitudes y diferencias con una LIFO.

SOLUCIÓN

a) Memoria FIFO: (Last Input, First Output). El último dato que se escribe es el primero que se lee. Por lo tanto no se puede acceder a la dirección de memoria que se desee.

b) Memoria de pila del CS3 (puntos más significativos, pero hay que desarrollarlos):

- Añade la posibilidad de hacer saltos en la ejecución del programa.
- Hay que introducir un registro de pila (SP) con capacidad de incremento y decremento que apunta a la primera dirección libre de la memoria de pila. Para la memoria de pila se utiliza la memoria de datos, comenzando por la dirección más alta (255).
- Para acceder que pasar la dirección de acceso al MAR y leer o escribir el dato e incrementar o decrementar el SP.
- La memoria de pila funciona como una memoria LIFO.

CRITERIO DE CORRECCIÓN

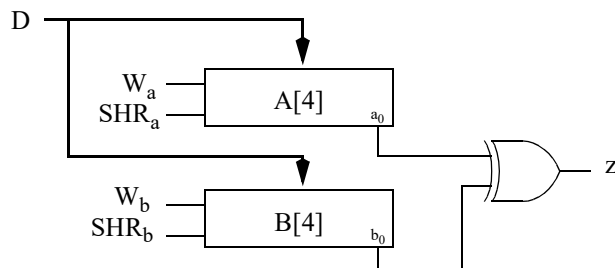
Apartado a) 4 puntos, apartado b, 6 puntos.

- 2.- [2 puntos]** Se tiene la unidad de datos de la figura, donde A y B son registros de 4 bits disparados por flanco de subida del reloj (clk) cuyo funcionamiento a nivel RT se describe en la tabla adjunta. Dibuje la

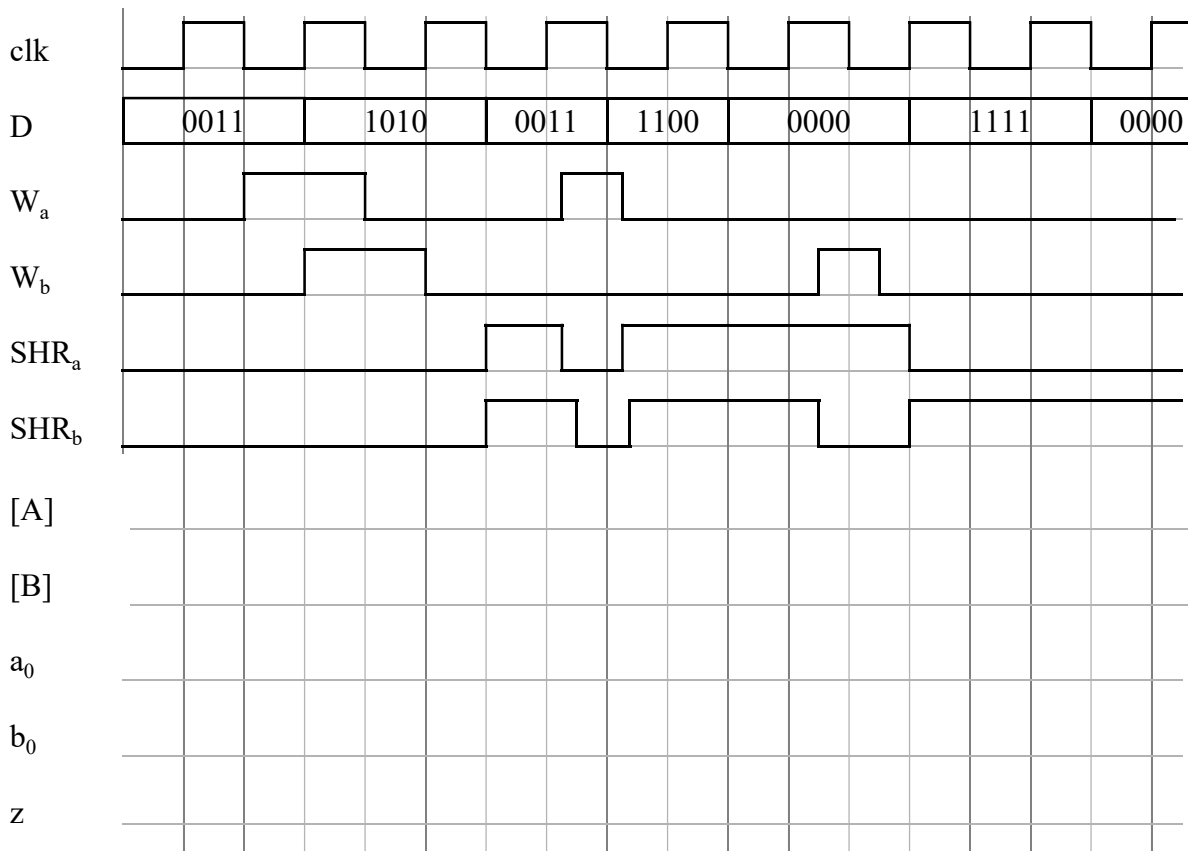
Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

evolución del contenido de los registros y de las salidas para las entradas propuestas.

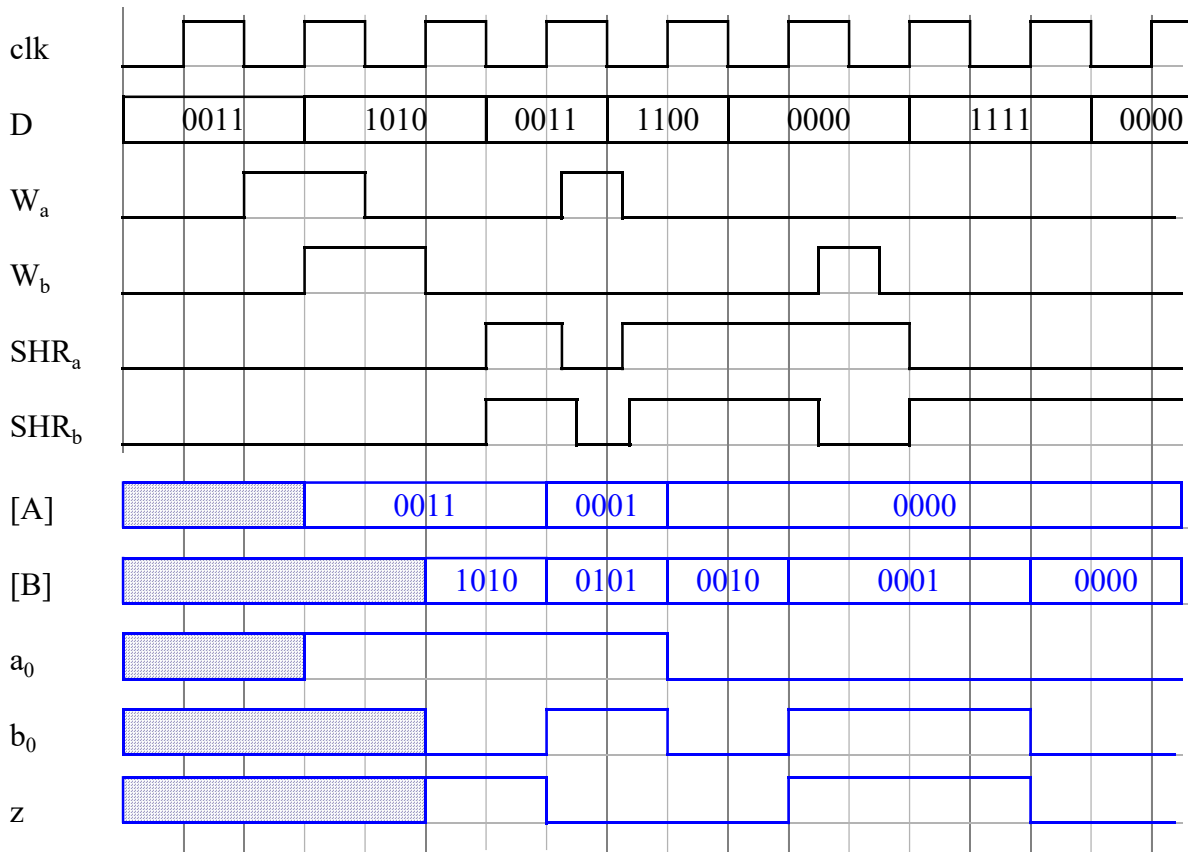


W SHR	$X \leftarrow$	$x_0 =$
00	[X]	X(0)
01	SHR (X, 0)	(bit menos significativo)
10	D	
11	No permitida	



Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

SOLUCIÓN:

Comentario: La escritura se produce sólo cuando hay un flanco activo de reloj (tanto la carga en paralelo como el desplazamiento), la lectura se produce de forma inmediata. La puerta XOR funciona de forma combinacional.

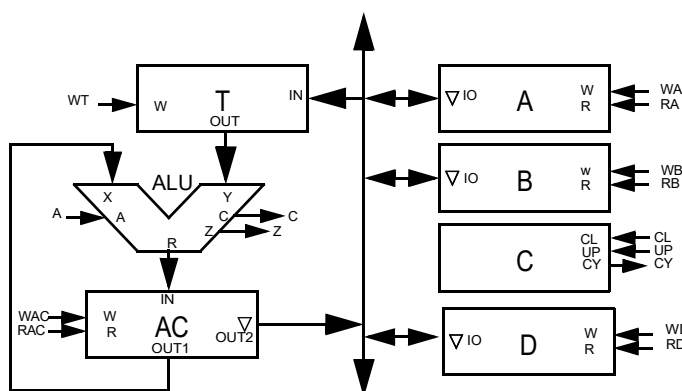
CRITERIO DE CORRECCIÓN

El contenido de cada registro vale 3 puntos. Las salidas valen 4 puntos.

Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

3.- [3 Puntos] Utilizando la unidad de datos de la figura (ALU, todos los registros y buses de 8 bits):



A, B y D: Registros con entrada/salida bidireccional y señales de lectura y escritura.

C: Contador ascendente (UP=1) con clear síncrono (CL=1), salida de carry (CY).

T: Registro con bus de entrada y salida diferenciados. Sin alta impedancia.

AC: Registro con entrada y dos salidas. Una de ellas de alta impedancia.

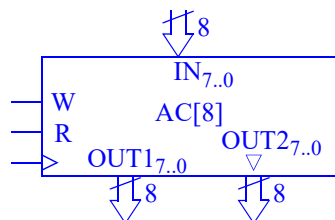
ALU: Realiza las operaciones de suma (A=1) y resta (A=0). C es la salida de carry y Z la de cero.

- Describa a nivel RT los registros AC y C.
- Suponiendo números sin signo, diseñe la carta ASM (unidad de datos y unidad de control) de un circuito que realice la división A/B y deje el cociente en C y el resto en D. Tenga en cuenta la posible división por 0.
- Implemente la carta ASM usando la técnica de un biestable por estado.

SOLUCIÓN

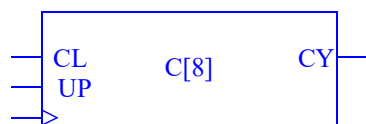
a)

Registro AC:



WR	AC←	OUT1=	OUT2=
00	AC		HiZ
01	AC		[AC]
10	IN	[AC]	HiZ
11	IN		[AC]

Registro C:



CL UP	C←	CY=
1 -	0	1 sii
01	C+1	[C]=255
00	C	

c)

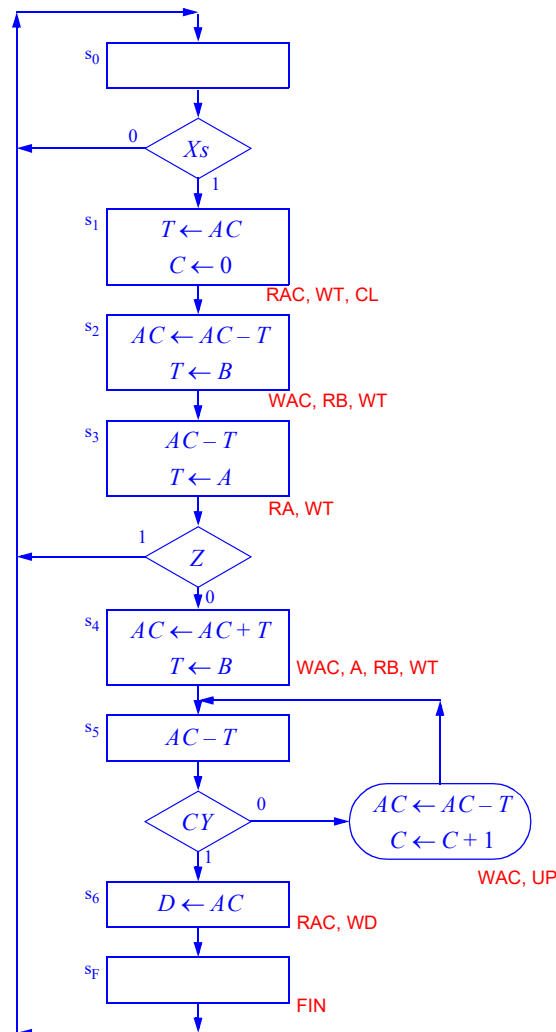
Carta ASM de la unidad de datos (y de control en rojo):

El algoritmo de la división consistirá en contar en C cuántas veces se puede restar B a A. Para ello, se comienza haciendo AC=0 (se carga T con AC y se hace AC=AC-T). Después se comprueba si B=0 (se carga B en T y se ordena AC+T), en cuyo caso se

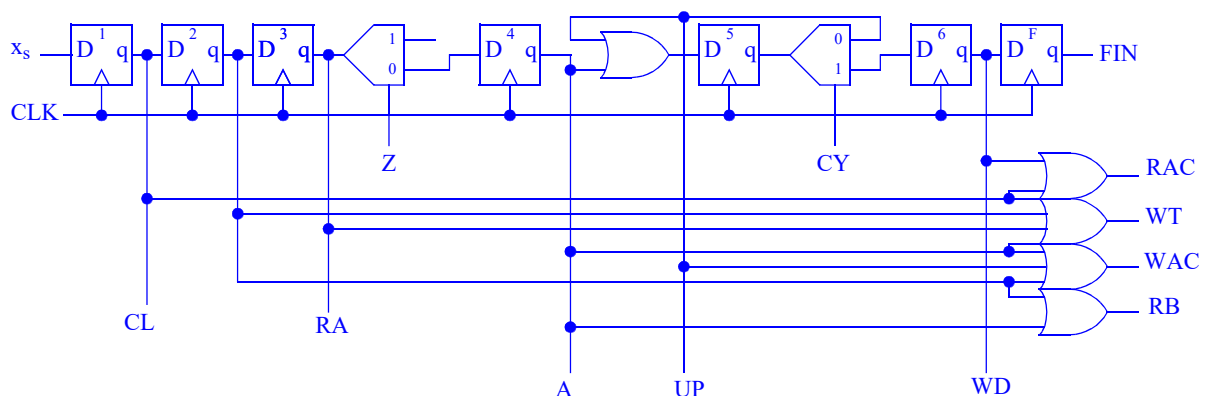
Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

sale por ser un error. Si B no es cero, se pone $C=0$ para usarlo como cociente y se hace $AC=A$ (se carga T con A y se hace $AC=AC+T$). Se vuelve a cargar B en T y se inicia un bucle en el que se ordena la resta $AC-T$. Si no se activa el carry es señal de que el resultado es positivo. En ese caso, se guarda el resultado en AC y se incrementa el cociente. En caso contrario, se ha terminado. Sólo queda pasar el contenido de AC, el resto, al registro D y salir.



Implementación de la UC con la técnica de un biestable por estado:



Apellidos:.....**SOLUCIÓN**.....

Nombre:.....

CRITERIO CORRECCIÓN

a)

- [60%] AC
- [40%] C

b) Carta ASM

c) [30%] Implementación

4.- (2021ED.C2) La subrutina `strlen` calcula la longitud de una cadena de caracteres tipo C. Recibe en R0 la dirección del primer carácter de la cadena y devuelve en R1 la longitud de la misma.

a) Explique con palabras, pseudocódigo o diagrama de flujo el algoritmo que va a usar.

b) Escriba la subrutina en ensamblador del CS3.

NOTA: Una cadena de caracteres es un vector de bytes codificados en ASCII. Las cadenas de caracteres tienen una longitud variable. El último elemento de la cadena va seguido a modo de terminador del carácter NUL cuyo código ASCII es 0. El terminador no forma parte de la cuenta. En el siguiente ejemplo, la longitud es 4. Es posible una cadena con longitud 0.

H	o	l	a	NUL
---	---	---	---	-----

SOLUCIÓN

```

;-----
; strlen                                     v1.0
;
; Calcula la longitud de una cadena de caracteres tipo C.
; Entradas:
;   R0: dirección de la cadena.
; Salidas:
;   R1: Longitud.
;-----
strlen:    ldi        r1, 0        ;Contador de caracteres
strlenBuc: ld         r2, (r0)     ;Leer carácter
          cpi         r2, 0       ;Es fin de cadena?
          breq        strlenFin   ;...sí. Salir
          addi         r1, 1       ;...no, contabilizar carácter
          addi         r0, 1       ;Avanzar puntero
          jmp         ststrlenBuc;Siguiente carácter
strlenFin: ret

```