



1. ¿Qué diferencia a los sistemas combinacionales de los secuenciales y de los programables?

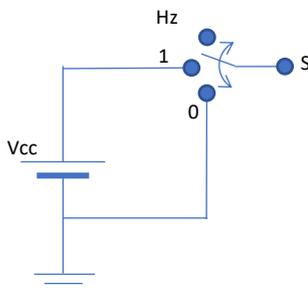
En un sistema combinacional las salidas en un instante  $t$  dependen sólo de las entradas en el mismo instante  $t$ . En un sistema secuencial las salidas dependen de las entradas en el instante  $t$  y de las entradas en instantes anteriores

2. Describir el comportamiento de un biestable JK síncrono.

La salida en el instante  $t$  ( $Q_t$ ) depende del estado de las entradas  $J$ ,  $K$  y de la salida en el instante anterior  $Q_{t-1}$  (según la tabla) en el flanco activo del reloj y permanece estable hasta el siguiente.

$Q_{t-1}$	J	K	$Q_t$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

3. Dibujar y describir el modelo electromecánico de una salida tree-state.



Una salida tree-state se puede modelar con un conmutador de tres posiciones conectado a la fuente de alimentación (1), a tierra (0) o desconectado (Hz).

Cuando el conmutador está en la posición 0 la salida es 0V.

Cuando el conmutador está en la posición 1 la salida es Vcc

Cuando el conmutador está en posición Hz la salida esta desconectada ofreciendo un estado de alta impedancia.

4. ¿Cuáles son los componentes de una CPU?

- Unidad de control (UC)
- Unidad aritmética (ALU)
- Registros

5. ¿Cuál es la función del registro PC?

Guardar la dirección de la dirección de memoria donde se encuentra la próxima instrucción a ejecutar o el próximo argumento de la instrucción en ejecución.

6. ¿Qué tipo de señales contiene el Bus-C?

Señales de nivel y señales de flanco



7. ¿Qué tipos de UC conoces y en qué se diferencian?

Cableadas y microprogramadas.

- Las cableadas implementan una máquina de estados
- Las microprogramadas emiten las salidas en secuencias almacenadas en una memoria.

8. Enumera todos los pasos a dar por la UC para leer y ejecutar la instrucción: NOP

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el IRCi para permitir que el código de operación se copie del bus-C al IRC
- Incrementar el PC

9. Enumera todos los pasos a dar por la UC para leer y ejecutar la instrucción: MOV 2000 D0

Leer el código de operación

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el IRCi para permitir que el código de operación se copie del bus-D al IRC
- Desactivar IRCi, MR y PCo
- Incrementar el PC

Leer el MSB de la dirección contenida en la instrucción

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el IRHi para permitir que el MSB de operación se copie del bus-D al IRH
- Desactivar IRCi, MR y PCo
- Incrementar el PC

Leer el LSB de la dirección contenida en la instrucción

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el IRHi para permitir que el MSB de operación se copie del bus-D al IRH
- Desactivar IRHi, MR y PCo
- Incrementar el PC

Leer la posición de memoria indicada

- Activar el IRAo para volcar el contenido del IRA en el bus-A



Universidad Francisco de Vitoria  
Fundamentos de ingeniería informática  
PCP 2 (5 de noviembre de 2019)

m1

- Esperar la estabilización del bus-A
  - Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
  - Esperar a la finalización del ciclo de lectura
  - Activar el D0i para permitir que el MSB de operación se copie del bus-D al D0
  - Desactivar D0i, MR y PCo
  - Incrementar el PC
10. Diseñar un circuito lógico que permita que en un display de 7 segmentos el led iluminado de vueltas por el círculos superior e inferior del ocho a una tasa de 1s; es decir, que, cada segundo, cambien las luces que están encendidas siguiendo la siguiente secuencia: (ad), (bc), (g), (ef), (ad). (bc), (g), (ef) ...

Dado que son cuatro parejas de leds que hay que encender sucesivamente utilizaremos un contador 0-3, con dos biestables con salidas Q0 y Q1, para que en cada estado se encienda tan solo un led. En el estado 0 se habrá de encender el led a, en el estado 1 el g, en el estado 2 el e y en el estado 3 el f.

Con ello tendremos la siguiente tabla de verdad

Q1	Q0	ad	bc	g	ef
0	0	1			
0	1		1		
1	0			1	
1	1				1

De ella se obtienen las siguientes ecuaciones:

$$ad = \overline{Q_1} \cdot \overline{Q_0}$$

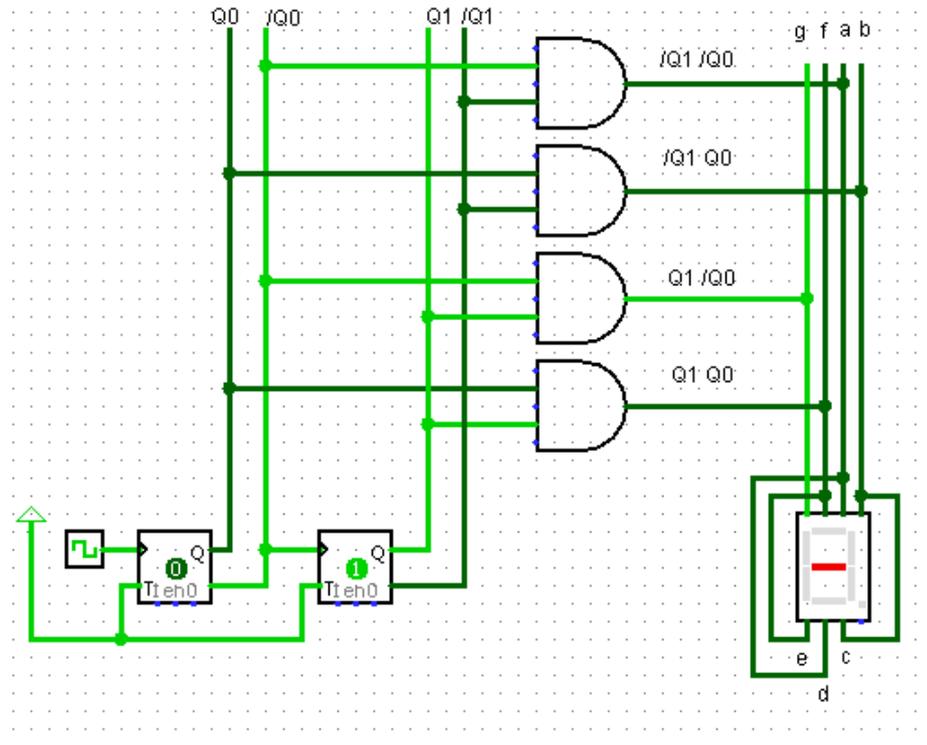
$$bc = \overline{Q_1} \cdot Q_0$$

$$g = Q_1 \cdot \overline{Q_0}$$

$$ef = Q_1 \cdot Q_0$$



Utilizaremos un contador de dos biestables T, conectando la entrada del primero a power, y la entrada del segundo a la salida del primero



11. El contenido del fichero .hex generado por el ensamblador del CH-2020 es el siguiente.

```
0000 20 01 00
0100 55 10 54 00 71 4D 50 40 68 10 40 E0 E5 C1 C4 4C
0110 4C 4D 4D 60 10 40 51 02 02 09 68 10 40 2B EC 01
```

Descodificar el contenido, anotar la dirección, el código máquina, las instrucciones y los comentarios pertinentes (NO TRIVIALES). Finalmente, e interpretar la operación que realiza el programa y anotarlo.

Dir.	Contenido / Código máquina	Instrucción / Nemónico	Comentario
0000	20 01 00	jmp 100	Saltar al inicio del programa
0100	55 10	mov #10 h0	A0 = puntero posiciones pares
0102	54 00	mov #00 l0	
0104	71	mov a0 a1	A1 = puntero posiciones impares
0105	4D	inc a1	
0106	50 40	mov #40 d0	contador = 40
0108	68 10 40	mov d0 1040	
010B	E0	mov (a0) d0	Intercambiar posición par e impar
010C	E5	mov (a1) d1	
010D	C1	mov d0 (a1)	
010E	C4	mov d1 (a0)	
010F	4C	inc a0	Avanzar puntero a pares
0110	4C	inc a0	



0111	4D	inc a1	Avanzar puntero impar
0112	4D	inc a1	
0113	60 10 40	mov 1040 d0	Decrementar el contador en 2
0116	51 02	mov #2 d1	
0118	02	clc	
0119	09	sub	
011A	68 10 40	mov d0 1040	
011D	2B EC	jr NZ EC (010B)	¿Fin de la cuenta?
011F	01	stop	

¿Qué operación realiza el programa decodificado?

En el vector que se encuentra en la dirección \$1000 de longitud \$40 se intercambian los contenidos de las posiciones pares con los de la posición siguiente.

12. El vector A es un array de bytes que contienen de forma predeterminada una secuencia de bytes, que concluye con la primera aparición del primer contenido de valor 0. El vector\_B es un array de bytes de longitud conocida TAMANO (por ejemplo 256). Se pide rellenar el vector B con la secuencia de bytes A repetida las veces que sea necesaria hasta completar el vector B: Ej: si A = 01 02 03 04 00 y TAMANO = 10  
→ B = 01 02 03 04 01 02 03 04 01 02

```
;Rellenar con patrón
TAMANO = 23          ;Tamaño del vector destino
TERMINADOR = 0      ;Terminador en el vector origen

Constantes = 1000
vector_A    byte 10 20 30 40 50 60 00    ;Vector origen

Variables = 1010
contador    space 1 ; Contador para controlar el bucle

Resultados = 1100
vector_B    space TAMANO                ;Vector destino

Reset = 0
    jmp Inicio

Inicio = 100
    mov    #MSB(vector_B) h1    ; A1 apunta vector destino
    mov    #LSB(vector_B) l1
    mov    #TAMANO d0           ; Iniciar contador
    mov    d0 contador

    mov    #MSB(vector_A) h0    ; A0 apunta vector origen
    mov    #LSB(vector_A) l0
```



Universidad Francisco de Vitoria  
Fundamentos de ingeniería informática  
PCP 2 (5 de noviembre de 2019)

m1

Repetir

```
; Copiar origen en destino
mov    (a0) d1          ; Leer origen
mov    #TERMINADOR d0  ; Es un terminador
cmp
jr     nz Copiar

mov    #MSB(vector_a) h0 ; A0 apunta de nuevo al inicio
mov    #LSB(vector_a) l0
mov    (a0) d1
```

Copiar

```
mov    d1 (a1)          ; Copiar el dato en el destino
inc    a1               ; Avanzar punteros
inc    a0

mov    contador d0     ;¿Final de la cuenta?
mov    #1 d1
clc
sub
mov    d0 contador
jr     nz Repetir
```

Terminar

```
stop
```



1. ¿Qué es y qué utilidad tiene un cronograma?

Es un diagrama que representa en el tiempo el estado y las transiciones de las diversas señales que se estudian. Permite establecer/conocer la sincronía entre los diversos dispositivos controlados por las señales estudiadas

2. Describir el comportamiento de un biestable SR asíncrono

- Mientras las entradas S y R sean a 0 la salida permanece inalterada.
- Mientras la entrada S sea 1 y R sea 0 la salida permanece en 1
- Mientras la entrada S sea 0 y R sea 1 la salida permanece en 0
- No está definido el valor de la salida cuando las dos entradas son 1

3. ¿Para qué y cómo son útiles las salidas tree-state?

Es posible conectar múltiples salidas de este tipo sin correr el riesgo de producir un cortocircuito de la fuente a tierra.

4. ¿Cuál es la función de la ALU?

Realizar operaciones aritméticas o lógicas binarias o unarias.

5. ¿Cuál es la función del registro IRC?

Almacenar el código de operación mientras se decodifica y ejecuta la instrucción.

6. ¿Cuál es la función del Bus-C?

Transportar las ordenes de la UC a las demás unidades del ordenador.

7. ¿Cómo funciona una UC cableada?

Es un circuito secuencial que implementa una máquina de estados. Emite secuencias de símbolos de salida que representan las diversas combinaciones de estados de las señales del bus C

8. Enumera todos los pasos a dar por la UC para leer y ejecutar la instrucción: ADD

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el IRCi para permitir que el código de operación se copie del bus-C al IRC
- Incrementar el PC

9. Enumera todos los pasos a dar por la UC para leer y ejecutar la instrucción: MOV #1A D0

Leer el código de operación

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura

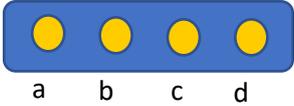


- Activar el IRCi para permitir que el código de operación se copie del bus-D al IRC
- Desactivar IRCi, MR y PCo
- Incrementar el PC

Leer el dato contenido en la instrucción

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el DOi para permitir que el dato se copie del bus-D al DO
- Desactivar DOi, MR y PCo
- Incrementar el PC

10. La reciente aparición de la iluminación led en los automóviles modernos a impulsado a los diseñadores a producir juegos de luces para mejorar la interpretación de la señal de los intermitentes. Queremos diseñar un circuito lógico que maneje los cuatro leds que componen el intermitente derecho según la siguiente secuencia que se repite indefinidamente con una temporización entre cada cambio es de 1s.

<ul style="list-style-type: none"> <li>• Todos los leds apagados</li> <li>• Sólo encendido a</li> <li>• Sólo encendidos a y b</li> <li>• Sólo encendidos a, b y c</li> <li>• Todos los leds encendidos</li> </ul>	<p>La organización de los leds es:</p> 
---	---

Utilizaremos un contador 0-4 en el que cada estado corresponda a una configuración de los leds  
En el estado 0: todos los leds estarán apagados, en el 1: sólo estará encendido el a, en el 2: estarán encendidos a y b, en el 3: los a, b y c, en el 4 todos los leds. Para controlar 5 estados son precisos 3 biestables T en configuración contador con reset en el estado 5.

Con esta información podemos construir la siguiente tabla de verdad

Q2	Q1	Q0	L0	L1	L2	L3	R
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	1	0	0	0
0	1	1	1	1	1	0	0
1	0	0	1	1	1	1	0
1	0	1	x	x	x	x	1
1	1	0	x	x	x	x	1
1	1	1	x	x	x	x	1



L0		Q1 Q0			
		00	01	11	10
Q2	0	0	1	1	1
	1	1	x	x	x

$L_0 = Q_2 + Q_1 + Q_0$

L1		Q1 Q0			
		00	01	11	10
Q2	0	0	0	1	1
	1	1	x	x	x

$L_1 = Q_2 + Q_1$

L2		Q1 Q0			
		00	01	11	10
Q2	0	0	0	1	0
	1	1	x	x	x

$L_2 = Q_2 + Q_1 Q_0$

L3		Q1 Q0			
		00	01	11	10
Q2	0	0	0	0	0
	1	1	x	x	x

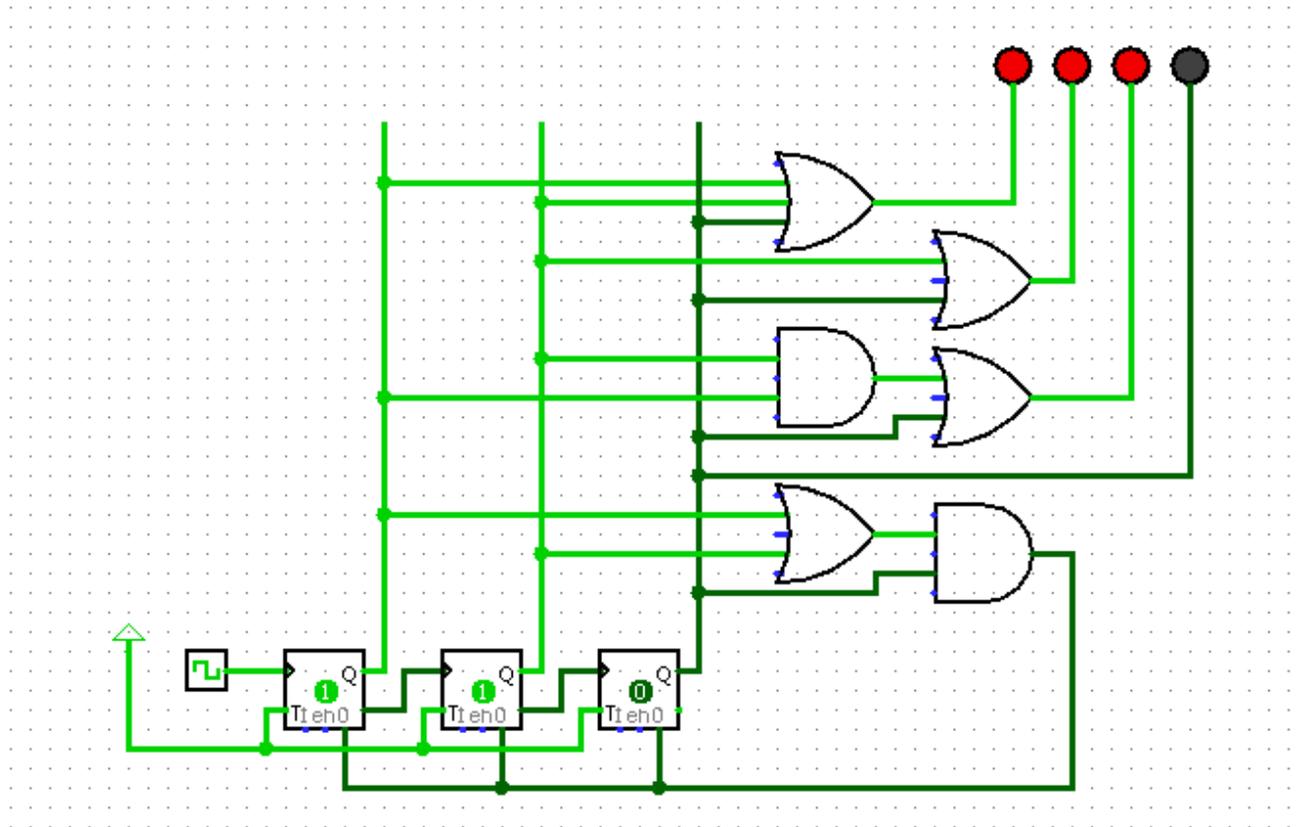
$L_3 = Q_2$

R		Q1 Q0			
		00	01	11	10
Q2	0	0	0	0	0
	1	0	1	1	1

$R = Q_2 Q_1 + Q_2 Q_0 = Q_2(Q_1 + Q_0)$



El circuito correspondiente



11. El contenido del fichero .hex generado por el ensamblador del CH-2020 es el siguiente. Decodificar el contenido, Anotar los comentarios pertinentes e interpretar la operación que realiza.

```
0000 20 01 00
0100 55 10 54 00 E0 55 10 54 40 48 E4 C0 69 10 40 A8
0110 51 10 02 0A 2B 07 A0 51 00 02 0A 2A 07 48 60 10
0120 40 20 01 0A 01
```

Decodificar el contenido, anotar la dirección, el código máquina, las instrucciones y los comentarios pertinentes (NO TRIVIALES). Finalmente, e interpretar la operación que realiza el programa y anotarlo.

Dir.	Contenido / Código máquina	Instrucción / Nemónico	Comentario
0000	20 01 00	jmp 100	Ir al inicio del programa
0100	55 10	mov #10 h0	D0= contenido del primer dato
0102	54 00	mov #00 10	
0104	E0	mov (a0) d0	
0105	55 10	mov #10 h0	A0 apunta al final del vector
0107	54 40	mov #40 10	



0109	48	dec a0	
010A	E4	mov (a0) d1	Guardar dato actual
010B	C0	mov d0 (a0)	Sustituir por dato anterior
010C	69 10 40	mov d1 1040	
010F	A8	mov h0 d0	Comparar A0 con inicio vector
0110	51 10	mov #10 d1	
0112	02	clc	
0113	0A	cmp	
0114	2B 07	jr NZ 07 (011D)	
0116	A0	mov l0 d0	
0117	51 00	mov #00 d1	
011A	0A	clc	
011B	2A 07	jr Z 07 (0124)	Si son iguales: Terminar
011D	48	dec a0	Retroceder el puntero
011E	60 10 40	mov 1040 d0	Recuperar el dato actual
0121	20 01 0A	jmp 010A	Repetir
0124	01	stop	

¿Qué operación realiza el programa decodificado?

Desplaza todos los datos del vector una posición hacia arriba, copiando en el final el valor que había en el principio. → Rota los datos hacia arriba.

12. El vector\_A es un array de 15 bytes que contienen de forma predeterminada una secuencia de bytes. El vector\_B es un array de 256 bytes. Cada elemento del vector A contiene un índice, para cada uno de ellos, escribir un \$FF en el elemento indicado del vector B, dejando todos los demás a 0.

```
;Rellenar con indices
TAMANO = 100
NINDICES = 0F

Indices = 1000
vector_A byte 01 02 03 04 12 33 57 9A 0A0 0A3 0A7 0B1 0C7 0E0 15

Variables = 1100
vector_B space TAMANO
contador space 1

Reset = 0
    jmp Inicio

Inicio = 100
    ;Borrar el contenido del vector destino
    mov #MSB(vector_B) h1          ; A1 apunta vector destino
    mov #LSB(vector_B) l1
```



Universidad Francisco de Vitoria  
Fundamentos de ingeniería informática  
PCP 2 (5 de noviembre de 2019)

m2

```
mov #0 l0                ; l0 el dato a escribir
mov #TAMANO d0           ; d0 contador de repeticiones
mov #1 d1                ; d1 decremento del contador

Borrar
mov  l0 (a1)             ; borrar cada elemento
inc a1
clc
sub
jr   nz Borrar          ; Repetir si no se ha borrado todo

;Leer los indices y marcar los elementos
mov  #MSB(vector_A) h0   ; A0 apunta al vector indices
mov  #LSB(vector_A) l0
mov  #NINDICES d0
mov  d0 contador

Marcar
mov  (a0) d0             ; Leer el indice actual
inc  a0

mov  #MSB(vector_B) h1   ; A1 apunta vector destino
mov  #LSB(vector_B) l1
mov  l1 d1               ; suma el índice
clc
add
mov  d0 l1
mov  h1 d0
mov  #0 d1
add
mov  d0 h1

mov  #0FF d0             ; Escribe $FF en la posición indicada
mov  d0 (a1)

mov  contador d0
mov  #1 d1
clc
sub
mov  d0 contador
jr  nz Marcar

Terminar
stop                    ; Detener la CPU
```



1. ¿ Que es una máquina de estados?

Una máquina de estados es un modelo para la especificación de sistemas secuenciales basado en que los valores de las salidas en el instante  $t$  dependen de las entradas en el instante  $t$  y del estado de la máquina en el instante ( $t$ ). El comportamiento se describe especificando las condiciones en las que se producen las transiciones de un estado a otro y, en el modelo de Mealy, los valores de salida asociadas a cada transición.

2. Describir el comportamiento de un biestable T síncrono

- Cuando en el flanco activo del reloj la entrada T toma el valor 0 el valor de la salida se mantiene.
- Cuando en el flanco activo del reloj la entrada T toma el valor 1 el valor de la salida cambia.

3. ¿ Qué relación hay entre un biestable JK y un biestable T?

Un biestable T se puede construir con un biestable JK en el que sus dos entradas estén cortocircuitadas.

4. ¿Cuál es la función de la UC?

Establecer qué función realiza la circuitería anexa y cuándo, ello permite sincronizar las acciones del hardware.

5. ¿Cuál es la función del registro IRA?

Almacenar temporalmente la dirección asociada a instrucciones con direccionamiento directo.

6. ¿Cuál es la función del Bus-D?

Transportar los datos entre de la memoria principal y/o los registros.

7. ¿Cómo funciona una UC microprogramada?

Descargando los valores de las salidas almacenadas en una memoria en posiciones consecutivas.

8. Enumera todos los pasos a dar por la UC para leer y ejecutar la instrucción:

XOR

Leer el código de operación

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el IRCi para permitir que el código de operación se copie del bus-D al IRC
- Desactivar IRCi, MR y PCo
- Incrementar el PC

Ejecutar la instrucción

- Activar la señal de flanco XOR

9. Enumera todos los pasos a dar por la UC para leer y ejecutar la instrucción:

MOV L1 (A0)



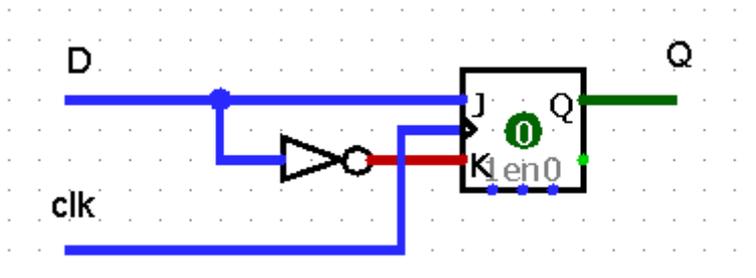
Leer el código de operación

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el IRCi para permitir que el código de operación se copie del bus-D al IRC
- Desactivar IRCi, MR y PCo
- Incrementar el PC

Escribir el contenido de d0 en memoria

- Activar la señal A0o para volcar el contenido de registros A0 en el bus-A
- Activar la señal L1o para volcar el contenido del registro L1 en el bus D
- Esperar la estabilización del bus-A y del bus-D
- Activar el MW para que la memoria de inicie el ciclo de escritura del dato en el Bus-D en la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de escritura
- Desactivar L1o, MW.

10. Disponemos de un biestable JK, con él y puertas lógicas obtener un circuito que se comporte como un biestable D:





Universidad Francisco de Vitoria  
Fundamentos de ingeniería informática  
PCP 2 (8 de noviembre de 2019)

v1

11. El contenido del fichero .hex generado por el ensamblador del CH-2020 es el siguiente. Descodificar el contenido, Anotar los comentarios pertinentes e interpretar la operación que realiza.

```
0000 20 01 00
0100 55 10 54 00 50 40 68 10 41 50 00 68 10 40 E4 4C
0110 60 10 40 02 0A 29 03 69 10 40 60 10 41 51 01 02
0120 09 68 10 41 2B E8 01
```

Descodificar el contenido, anotar la dirección, el código máquina, las instrucciones y los comentarios pertinentes (NO TRIVIALES). Finalmente, e interpretar la operación que realiza el programa y anotarlo.

Dir.	Contenido / Código máquina	Instrucción / Nemónico	Comentario
0000	20 01 00	jmp 100	En reset ejecutar el programa
0100	55 10	mov #10 h0	A0 = dirección vector + tamaño
0102	54 00	mov #00 10	
0104	50 40	mov #TAMANO d0	contador (1041) = \$40
0106	68 10 41	mov d0 1041	
0109	50 00	mov #0 d0	máximo (1040) = 0
010B	68 10 40	mov d0 1040	
010E	E4	mov (a0) d1	Leer el siguiente dato
010F	4C	inc a0	Avanzar el puntero
0110	60 10 40	mov 1040 d0	Comparar con máximo(1040)
0113	02	clc	
0114	0A	cmp	
0115	29 03	jr nc 03	Si dato > máximo (salto = 0117+0003 = 011A)
0117	69 10 40	mov d1 1040	máximo = dato
	Seguir		
011A	60 10 41	mov 1041 d0	Decrementar contador
011D	51 01	mov #1 d1	
011F	02	clc	
0120	09	sub	
0121	68 10 41	mov d0 1041	
0124	2B E8	jr nz E8	Repetir si no es el final de cuenta: 0126+FFE8 = ≠010E
0126	01	stop	Detener la CPU



¿Qué operación realiza el programa decodificado?

Calcular el valor máximo entre los datos de un vector de longitud \$40

12. vector es un array de 100 bytes que contienen una información determinada. Se pide cambiar el orden de los elementos del vector, de modo que al final del programa en el vector aparezcan los mismos datos que había al principio, pero de modo que en la posición 0 se encuentre el dato que antes se encontraba en la posición 99, en la 1 lo que antes estaba en la 98 ... en la posición 99 lo que antes estaba en la 0.

```
;Reordenar

TAMANO = 100

Variables = 1000
vector   space TAMANO
fin_vector

Reset = 0
    jmp Inicio

Inicio = 100
    mov #MSB(vector) h0      ; A0 apunta arriba en vector
    mov #LSB(vector) l0

    mov #MSB(fin_vector) h1  ; A1 apunta abajo en el vector
    mov #LSB(fin_vector) l1
    dec a1

Repetir
    mov (a0) d0              ; Intercambiar el dato de arriba
    mov (a1) d1              ; con el de abajo.
    mov d0 (a1)
    mov d1 (a0)

    inc a0                   ; Avanzar el puntero de arriba
    dec a1                   ; Retroceder el puntero de abajo

    mov h0 d0                ; Comprobar si se han cruzado los punteros
    mov h1 d1
    clc
    cmp
    jr  C Terminar
    jr  NZ Repetir

    mov l0 d0
    mov l1 d1
    clc
```



Universidad Francisco de Vitoria  
Fundamentos de ingeniería informática  
PCP 2 (8 de noviembre de 2019)

v1

```
cmp
jr  C Repetir

Terminar
stop
```



Universidad Francisco de Vitoria  
Fundamentos de ingeniería informática  
PCP 2 (9 de noviembre de 2019)

v2

1. ¿Qué información es necesaria para definir el comportamiento de una máquina de estados?

Entradas, salidas estados y transiciones.

2. Describir el comportamiento de un biestable D síncrono.

- Cuando en el flanco activo del reloj la entrada T toma el valor 0 el valor de la salida es 0.
- Cuando en el flanco activo del reloj la entrada T toma el valor 1 el valor de la salida es 1.

3. ¿Qué relación hay entre un biestable JK y un biestable D?

Un biestable D se puede construir uniendo las entradas J y K con una puerta NOT

4. ¿Utilidad y función de los registros de la CPU?

Almacenar temporalmente los datos, resultados y direcciones que se utilizarán en un tiempo próximo.

5. ¿Cuál es la función del registro SP?

Permite gestionar la pila almacenando la dirección del TOS

6. ¿Cuál es la función del Bus-A?

Transportar direcciones desde un registro a otro o a la memoria.

7. ¿Cómo funciona una ALU?

Combinando los operandos de entrada en las operaciones binarias o uno en las operaciones unarias para conseguir un resultado.

8. Enumera todos los pasos a dar por la UC para leer y ejecutar la instrucción: INC A1

Leer el código de operación

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar el IRCi para permitir que el código de operación se copie del bus-D al IRC
- Desactivar IRCi, MR y PCo
- Incrementar el PC

Ejecutar la instrucción

- Activar la señal de flanco A1incr



9. Enumera todos los pasos a dar por la UC para leer y ejecutar la instrucción: MOV (A0) D1

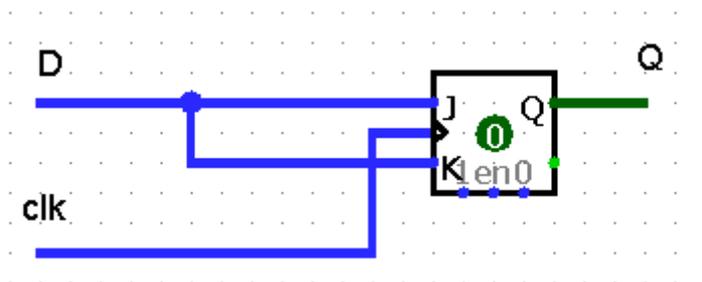
Leer el código de operación

- Activar el PCo para volcar el contenido del pc en el bus-A
- Esperar la estabilización del bus-A
- Activar el MR para que la memoria de inicio al ciclo de lectura de la dirección indicada en el Bus-A
- Esperar a la finalización del ciclo de lectura
- Activar la señal de flanco IRCi para permitir que el código de operación se copie del bus-D al IRC
- Desactivar MR y PCo
- Incrementar el PC

Leer (A0) en memoria

- Activar la señal A0o para volcar el contenido del registro A0 en el bus-A
- Esperar la estabilización del bus-A
- Activar la señal MR para iniciar el ciclo de lectura de la memoria de la dirección en el bus-A
- Esperar a la finalización del ciclo de lectura y la estabilización del bus-D
- Activar la señal de flanco D1i para copiar lo presente en el bus-D en el registro D1
- Desactivar A0o y MR

10. Disponemos de un biestable JK, con él y puertas lógicas obtener un circuito que se comporte como un biestable T:





Universidad Francisco de Vitoria  
Fundamentos de ingeniería informática  
PCP 2 (9 de noviembre de 2019)

v2

11. El contenido del fichero .hex generado por el ensamblador del CH-2020 es el siguiente. Descodificar el contenido, Anotar los comentarios pertinentes e interpretar la operación que realiza.

```
0000 20 01 00
0100 55 10 54 00 57 10 56 40 50 40 51 01 02 09 68 10
0110 80 E4 4C E0 02 09 C1 4D 60 10 80 51 01 02 09 68
0120 10 80 2B ED 01
```

Descodificar el contenido, anotar la dirección, el código máquina, las instrucciones y los comentarios pertinentes (NO TRIVIALES). Finalmente, e interpretar la operación que realiza el programa y anotarlo.

Dir.	Contenido / Código máquina	Instrucción / Nemónico	Comentario
0000	20 01 00	jmp 100	En reset ejecutar el programa
0100	55 10	mov #10 h0	A0 puntero al vector
0102	54 00	mov #00 10	de datos (\$1000)
0104	57 10	mov #10 h1	A1 apunta al vector
0106	56 40	mov #40 l1	de resultados (\$1040)
0108	50 40	mov #40 d0	Inicia contador a
010A	51 01	mov #1 d1	uno menos que la longitud
010C	02	clc	
010D	09	sub	
010E	68 10 80	mov d0 1080	
0111	E4	mov (a0) d1	Tomar el primer dato
0112	4C	inc a0	Avanzar al dato siguiente
0113	E0	mov (a0) d0	Leer el siguiente dato
0114	02	clc	Restar actual - anterior
0115	09	sub	
0116	C1	mov d0 (a1)	Guardar la diferencia en el vector de resultados
0117	4D	inc a1	Avanzar puntero resultados
0118	60 10 80	mov 1080 d0	Decrementar contador
011B	51 01	mov #1 d1	
011D	02	clc	
011E	09	sub	
011F	68 10 80	mov d0 1080	
0122	2B ED	jr nz ED (0111)	Repetir si no se han realizado todas las repeticiones saltando a 0124+FFED=0111
0124	01	stop	



Universidad Francisco de Vitoria  
Fundamentos de ingeniería informática  
PCP 2 (9 de noviembre de 2019)

v2

¿Qué operación realiza el programa decodificado?

Dado un vector de datos de longitud \$40 guarda en otro de longitud \$3F las diferencias entre un dato y el que le sucede en el array.

$\text{resultado}(i) = \text{datos}(i+1) - \text{dato}(i)$  para todo  $0 \leq i < 40$



12. vector es un array de 100 bytes que contiene inicialmente una información determinada. Se necesita programa que intercambie en todo el array los contenidos de las posiciones pares con las impares.

```
;Pares e impares

TAMANO = 40

Variables = 1000
vector      space TAMANO
descontador space 1

Reset = 0
    jmp Inicio

Inicio = 100
    mov  #MSB(vector) h0      ; A0 apunta a pares
    mov  #LSB(vector) l0

    mov  a0 a1                ; A1 apunta a impares
    inc  a1

    mov  #TAMANO d0          ; Iniciar el descontador
    mov  d0 descontador

Repetir
    mov  (a0) d0              ; Intercambiar el dato de arriba
    mov  (a1) d1              ; con el de abajo.
    mov  d0 (a1)
    mov  d1 (a0)

    inc  a0                    ; Avanzar el puntero a pares
    inc  a0
    inc  a1                    ; Avanzar el puntero a impares
    inc  a1

    mov  descontador d0      ; Decrementar descontador
    mov  #1 d1
    clc
    sub
    sub
    mov  d0 descontador
    jr   NZ Repetir

Terminar
    stop
```