

Examen final IC (Parte 1)

- Duración del examen: 2:30 horas.
- Los problemas tienen que resolverse en las **HOJAS DE RESPUESTAS**.
- No podéis utilizar calculadora, móvil, apuntes, etc.
- La solución del examen se publicará en Atenea mañana por la tarde.
- La revisión será el **21 de junio** a las **11:00** en el aula **D6-114**. Las notas definitivas se publicarán en Atenea al día siguiente.

Para algunos de los ejercicios son útiles las figuras y tablas que se muestran en la hoja aparte

La puntuación de las preguntas de esta parte del examen corresponden a 6 puntos de la nota final del examen.

Pregunta 1) (0.2 puntos)

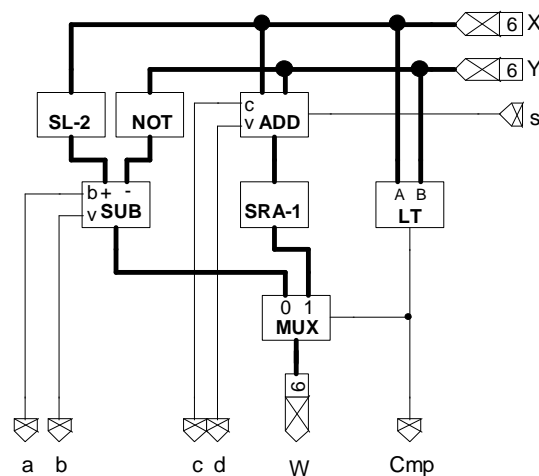
Cada fila de la tabla tiene 3 columnas con: el vector de 8 bits X escrito en hexadecimal, el valor que representa X interpretado como un número natural codificado en binario (X_u) y el valor que representa X interpretado como un número entero codificado en complemento a dos (X_s). Completa las casillas que faltan.

X (hexa)	X_u	X_s
	143	
8E		

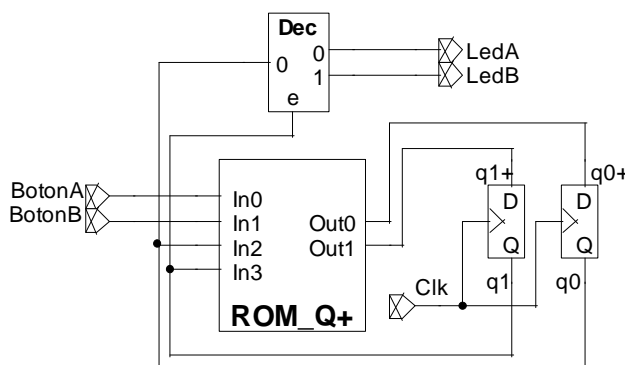
Pregunta 2) (0.4 puntos)

Dado el siguiente circuito combinacional, los valores de los vectores de 4 bits de entrada X e Y y el valor de la señal s, rellena la tabla indicando el valor de las señales de salida a, b, c, d y Cmp, el valor del vector de 4 bits de la salida W y su interpretación como número natural (W_u) y número entero (W_s) para cada combinación de valores de entrada (cada fila de la tabla). El bloque LT realiza la comparación de números enteros $A_s < B_s$ activando el bit de salida a 1 cuando es cierta.

s	X	Y	a	b	c	d	Cmp	W (4 bits)	W_u	W_s
1	1011	0110								
1	0011	1001								

**Pregunta 3)** (0.3 puntos)

Dibuja el grafo de estados del siguiente circuito (se da la leyenda) sabiendo que el valor inicial de los biestables es $q_1=1$ i $q_0=0$.



Contenido ROM_Q+

	q_1^+	q_0^+
M[0]	0	0
M[1]	0	0
M[2]	0	1
M[3]	0	1
M[4]	0	1
M[5]	1	0
M[6]	0	1
M[7]	1	0
M[8]	1	0
M[9]	1	0
M[10]	1	1
M[11]	1	1
M[12]	1	1
M[13]	1	1
M[14]	1	0
M[15]	0	1

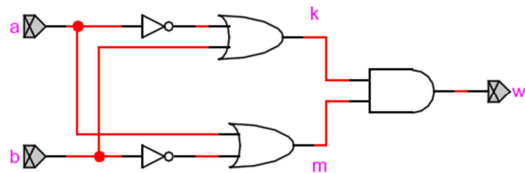
Pregunta 4) (0.4 puntos)

A partir del siguiente texto que describe el funcionamiento de un circuito lógico secuencial (CLS) de Moore, completa el grafo de estados que describe su funcionamiento y que se encuentra en la hoja de respuestas: le faltan sólo algunos arcos y las etiquetas.

El CLS a diseñar es un reconocedor de una secuencia con dos entradas (x, y) y una salida w. La salida w vale 1 al ciclo siguiente de que las entradas hayan recibido la última pareja de bits (xy) de la secuencia: (00), seguido de (01) que puede repetirse un número indeterminado de veces, seguido de (10) y finalmente (11). En cualquier otro caso la salida w vale 0.

Pregunta 5) (0.4 puntos)

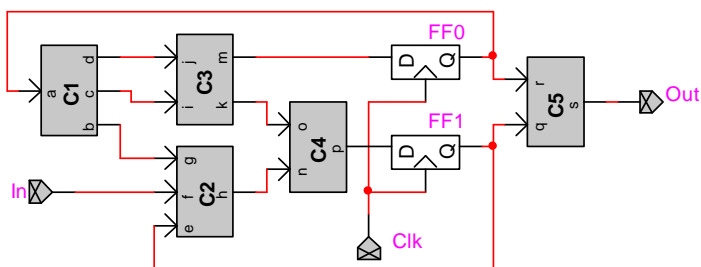
Completad el siguiente cronograma de las señales del esquema lógico sabiendo que los tiempos de propagación de las puertas son: $T_{p(Not)} = 10$, $T_{p(Or-2)} = 20$, $T_{p(And-2)} = 30$ u.t. Debéis operar adecuadamente con las zonas sombreadas (no se sabe el valor que tienen) y dibujar la señal sombreada cuando no se pueda saber si vale 0 o 1. Podéis usar las dos filas sin nombre (entre la b y la k y entre la k y la m) para lo que queráis. Solo se evaluarán las filas de las señales k, m y W



	0			50			100			150			
a													
b													
k													
m													
W													

Pregunta 6) (0.5 puntos)

El siguiente circuito secuencial está inmerso en un sistema más complejo. Se sabe que la entrada In se estabiliza pasadas 80 u.t. desde la llegada del flanco ascendente de reloj y la salida Out tiene que estar estable 40 u.t. antes de la llegada del flanco ascendente de reloj. Deseamos calcular el tiempo de ciclo mínimo del sistema suponiendo que el camino crítico pasa por este circuito. Los bloques C1, C2, C3 y C4 son circuitos combinacionales cuya implementación interna desconocemos.



Los tiempos de propagación de los bloques combinacionales que hay en este circuito son los siguientes (TC_{i-n-m} indica el tiempo de propagación del bloque C_i desde la entrada n a la salida m medido en u.t.):

TC_1	TC_2	TC_3	TC_4	TC_5
a-b 60	e-h 70	i-k 10	n-p 50	q-s 90
a-c 10	f-h 50	i-m 30	o-p 50	r-s 20
a-d 70	g-h 20	j-k 20		
		j-m 50		

El tiempo de propagación de cualquier biestable del sistema es de 100 u.t.

6.a) Escribid **todos los caminos críticos** que haya listando la secuencia de bloques que los forman e indicando la entrada y salida de cada bloque por la que pasa el camino. (0.3 puntos)

6.b) ¿Cuál es el tiempo de ciclo mínimo del sistema? (0.2 puntos)

Pregunta 7) (0.4 puntos)

a	b	c	w
0	0	0	1
0	0	1	1
0	1	0	X
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Dada la tabla de verdad correspondiente a una función que calcula w, responde a las siguientes preguntas.

7.a) Dibujad el mapa de Karnaugh marcando claramente las agrupaciones de unos adecuadas para obtener la expresión mínima en suma de productos de la función w. Escribid la expresión mínima en suma de productos de w. (0.2 puntos)

7.b) Si implementamos directamente la expresión en suma de minterms de la función w considerando las x como 0, ¿Cuántas puertas And y OR y de cuántas entradas hacen falta? (0.1 puntos)

7.c) Si implementamos la función w con una ROM. ¿Cuántas palabras y cuántos bits por palabra tiene la ROM? (0.1 puntos)

Pregunta 8) (0.4 puntos)

8.a) Escribid la fórmula que da el valor de un número natural en función de los 3 dígitos que lo representan en el sistema convencional en base 6. (0.1 puntos)

8.b) Escribid la fórmula que da el valor de un número entero en función de los k bits en complemento a dos que lo representan. (0.2 puntos)

8.c) Expresad el rango de los números naturales que se pueden representar en el sistema convencional en base 3 para el caso de un vector X de 3 dígitos. (0.1 puntos)

Pregunta 9) (1.3 puntos)

A partir de la descripción de la funcionalidad del procesador de propósito específico (PPE) que se da a continuación, diseña una implementación formada por una unidad de proceso y una unidad de control ambas de propósito específico. Para la unidad de proceso (UP) haz un diseño ad-hoc usando los bloques combinacionales y secuenciales vistos en clase. Especifica la unidad de control (UC) mediante un grafo de estados de Moore.

A la unidad de proceso (UP) del PPE le llega del exterior el bus X, de 8 bits, por donde entran los datos y de la UP sale el bus W, de 8 bits, por donde sale el resultado al exterior.

A la unidad de control (UC) del PPE le llega del exterior una señal (de un bit) denominada Begin (que se usa para sincronizar la entrada de datos por el bus X y determinar cuándo hay que comenzar un nuevo cálculo) y de la UC sale hacia el exterior una señal denominada End (que se usa para sincronizar la salida del resultado por W e indicar que terminó el cálculo en curso).

De la UP a la UC no va ninguna señal de condición y de la UC a la UP tampoco va ninguna señal de palabra de control.

Usamos la siguiente notación para especificar la funcionalidad del PPE:

- Begin(c) y End(c) indican el valor binario presente en las señales Begin y End en el ciclo genérico c.
- $X(c)_u$ y $W(c)_u$ indican el número natural representado en los n bits del bus X y W en el ciclo c.

La funcionalidad (el cálculo y la sincronización) que se realiza indefinidamente el PPE es la siguiente:

```
If (Begin(c) == 1) {
     $W(c+3)_u = 9 * X(c+1)_u + X(c+2)_u$ ; no se detecta resultado no representable en n bits
    End(c+3) = 1;
}
```

Se considera, además, que:

- La señal End valdrá 0 el resto de ciclos en los que no se especifica su valor en el pseudocódigo anterior.
- No importa el valor que tenga el bus W el resto de ciclos no especificados en el pseudocódigo anterior (por lo que podéis poner XX en el cronograma).
- No se ignorará el valor de la señal Begin durante ningún ciclo. Si Begin vale 1 durante los ciclos c+1 y c+2 se abortará el cálculo en curso y se comenzará el nuevo cálculo con el primer dato que llegará al ciclo siguiente. Si Begin vale 1 durante el ciclo c+3 (mientras se está mostrando el resultado del cálculo que ya terminó) indica que llamamos c a este ciclo (porque Begin vale 1) y que al ciclo siguiente llegará por la entrada X el primer valor del nuevo cálculo, el $X(c+1)$.

Comentario de ayuda: $9 = 8 + 1$.

9.a) Rellena el cronograma simplificado de la hoja de respuestas (todos los buses son de 8 bits). Escribe la secuencia de valores para End y W para los ciclos del 4 al 13. Cuando el valor de la señal W no importe indícalo con XX.

9.b) Dibuja el circuito de la unidad de proceso (UP) considerando que: a) solo tiene dos registros Rx y Rw, b) La entrada X de la UP está conectada a la entrada D de Rx y c) la salida Q de Rw está conectada a la salida W de la UP. Dibuja, también, el grafo de estados de la unidad de control (UC).

Pregunta 10) (0.4 puntos)

Indica el valor que debe tener cada uno de los bits de la palabra de control de la UPG (sin subsistema de I/O ni memoria, ver anexo) para que realice, durante un ciclo, la acción concreta especificada mediante el mnemotécnico. Indica con x las casillas cuyo valor **no importe** para la ejecución de la instrucción. En caso de que no se pueda realizar la acción tachar toda la línea de señales.

Mnemotécnico	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
CMPLTU R2, R7, R4									
OUT R1 // MOVEI R2, 0x99									
SHLI -, R5, -7									

Pregunta 11) (0.6 puntos)

Completad el fragmento de grafo de estados de la UC de **propósito específico** para que junto con la UPG formen un procesador que realice la funcionalidad descrita mediante el siguiente código en C (el código no tiene que hacer algo útil). Indicad los arcos que faltan, las etiquetas de los arcos (z, lz, o nada) y completad las casillas de cada palabra de control que se especifica con mnemotécnicos a la derecha de cada nodo del grafo. En las comparaciones, hay que interpretar los datos como valores **naturales**. (El operador || es el operador booleano Or, que retorna VERDADERO o FALSO). El grafo se encuentra en la hoja de respuestas. Nota: Se evalúa primero $R0 \geq 127$ y solo se evalúa $R1 \geq R2$ si la primera evaluación da FALSO.

```
if ((R0>=127) || (R1>=R2)) {  
    R3 = R4/4  
}else {R3 = R3*4  
}  
R4 = R3-R5;
```

Pregunta 12) (0.7 puntos)

Completad (en la hoja de respuestas) el fragmento de programa en lenguaje ensamblador SISA para que el procesador formado por la unidad de control de propósito general (UCG) junto con la UPG realice la misma funcionalidad que la descrita en la **pregunta 11** en el fragmento de código en C. El código SISA ya escrito solamente escribe en los registros R3, R4 (para implementar la funcionalidad descrita) y R7 (para valores temporales). En las comparaciones, hay que interpretar los datos como valores **naturales**. Rellenad la parte que falta. No se pueden utilizar etiquetas para los saltos.

Anexo:

Formato Instrucciones SISA

15 14 13 12	11 10 9	8 7 6	5 4 3	2 1 0	Name	Mnemonic
0 0 0 0	a a a	b b b	d d d	f f f	Logic and Arithmetic Operations	AND, OR, XOR, NOT, ADD, SUB, SHA, SHL
0 0 0 1	a a a	b b b	d d d	f f f	Compare Signed and Unsigned	CMPLT, CMPL, -, CMPEQ, CMPLTU, CMPEU, -, -
0 0 1 0	a a a	d d d	n n n	n n n	Add Immediate	ADDI
0 0 1 1	a a a	d d d	n n n	n n n	Load	LD
0 1 0 0	a a a	b b b	n n n	n n n	Store	ST
0 1 0 1	a a a	d d d	n n n	n n n	Load Byte	LDB
0 1 1 0	a a a	b b b	n n n	n n n	Store Byte	STB
0 1 1 1	a a a	d d d	x x x	x x x	Jump and link register	JALR
1 0 0 0	a a a	0 1	n n n	n n n	Branch on Zero	BZ
	d d d	0	n n n	n n n	Branch on Not Zero	BNZ
1 0 0 1	a a a	1	n n n	n n n	Move Immediate	MOVI
	d d d		n n n	n n n	Move Immediate High	MOVHI
1 0 1 0	a a a	1	n n n	n n n	Input Output	IN OUT
1 0 1 1						Future extensions
1 1 x x						

Funcionalidades ALU

F	b ₂ b ₁ b ₀	11	10	01	00
0	0 0 0	—	X	CMPLT (X,Y)	AND(X,Y)
0	0 0 1	—	Y	CMPL (X,Y)	OR(X,Y)
0	0 1 0	—	MOVHI (X,Y)	—	XOR(X,Y)
0	0 1 1	—	X&(~1)	CMPEQ (X,Y)	NOT(X)
1	0 0 0	—	—	CMPLTU (X,Y)	ADD(X,Y)
1	0 0 1	—	—	CMPEU (X,Y)	SUB(X,Y)
1	0 1 0	—	—	—	SHA(X,Y)
1	1 1 1	—	—	—	SHL(X,Y)

