

EXAMEN PARCIAL D'IC

18 d'Abril de 2001

Contesteu tots els problemes en fulls separats.

Temps: 2h 30min.

PROBLEMA 1 (2 punts)

Efectuar las siguientes operaciones de aritméticas indicando claramente los cálculos realizados:

a) Escribir en *Complemento a dos* los siguientes números, usando el menor número de bits:

1481 -551 -388

b) Escribir en decimal los siguientes números escritos en *Complemento a dos*:

10100010 00101001 11011011

c) Realizar las siguientes operaciones, con 6 bits:

010010 110001 010101
+ 110011 - 111000 + 010000

c.1) Considerando los números como naturales, ¿Cuál de ellas produce overflow?

c.2) ¿Y si fueran números enteros representados en complemento a dos?

Solución

a) Primero pasamos el número a binario mediante divisiones sucesivas:

```
1481 / 2
  1  740 / 2
    0  370 / 2
      0  185 / 2
        1  92 / 2
          0  46 / 2
            0  23 / 2
              1  11 / 2
                1  5 / 2
                  1  2 / 2
                    0  1
```

$1481_{10} = 10111001001$ pero como se trata de un número positivo en complemento a dos es preciso añadir un bit de signo adicional. Este bit debe ser un 0 para indicar que se trata de un número positivo. Por tanto precisamos 11 bits:

$1481_{10} = 010111001001$

$551_{10} = 1000100111$ pero como se trata de un número positivo en complemento a dos es preciso añadir un bit de signo adicional. Este bit debe ser un 0 para indicar que se trata de un número positivo. Por tanto precisamos 12 bits:

$$551_{10} = 01000100111$$

En un segundo paso es preciso transformar el número cambiándolo de signo, para ello realizaremos la operación multiplicar por -1 (cambiar 0s por 1s y 1s por 0s, y sumar 1).

$$\begin{array}{r} 01000100111 \\ 10111011000 \\ + \quad \quad \quad 1 \\ \hline 10111011001 \end{array}$$

$$-551_{10} = 10111011001$$

$388_{10} = 110000100$ pero como se trata de un número positivo en complemento a dos es preciso añadir un bit de signo adicional. Este bit debe ser un 0 para indicar que se trata de un número positivo. Por tanto precisamos 10 bits:

$$388_{10} = 0110000100$$

En un segundo paso es preciso transformar el número cambiándolo de signo, para ello realizaremos la operación multiplicar por -1 (cambiar 0s por 1s y 1s por 0s, y sumar 1).

$$\begin{array}{r} 0110000100 \\ 1001111011 \\ + \quad \quad \quad 1 \\ \hline 1001111100 \end{array}$$

$$-388_{10} = 1001111100$$

b)

10100010 es un número negativo, por tanto primero lo transformaremos cambiándolo de signo:

$$\begin{array}{r} 10100010 \\ 01011101 \\ + \quad \quad \quad 1 \\ \hline 01011110 \end{array}$$

$$01011110 = 2^6 + 2^4 + 2^3 + 2^2 + 2^1 = 94$$

Finalmente el número que buscamos es el -94.

00101001 es un número positivo, por tanto podemos transformarlo directamente a decimal:

$$00101001 = 2^5 + 2^3 + 2^0 = 41$$

11011011 es un número negativo, por tanto primero lo transformaremos cambiándolo de signo:

$$\begin{array}{r}
 11011011 \\
 00100100 \\
 + \quad \quad 1 \\
 \hline
 00100101
 \end{array}$$

$$00100101 = 2^5 + 2^2 + 2^0 = 37$$

Finalmente el numero que buscamos es el **-37**.

- c) Las operaciones de suma y resta son independientes de la interpretación que realicemos de los números. Primero realizaremos las operaciones y luego interpretaremos los resultados según el tipo de datos.

$$\begin{array}{r}
 010010 \\
 + \quad 110011 \\
 \hline
 1000101
 \end{array}$$

Números naturales: Esta operación produce overflow ya que hay carry.

Números enteros: No hay overflow, la suma de dos números de distinto signo nunca produce overflow.

$$\begin{array}{r}
 110001 \\
 - \quad 111000 \\
 \hline
 1111001
 \end{array}$$

Números naturales: Esta operación produce underflow ya que hay carry en la resta.

Números enteros: No hay overflow, la resta de dos números del mismo signo nunca produce overflow.

$$\begin{array}{r}
 010101 \\
 + \quad 010000 \\
 \hline
 100101
 \end{array}$$

Números naturales: Esta operación no produce overflow ya que no hay carry.

Números enteros: Hay overflow, la suma de dos números positivos produce un numero negativo como resultado.

PROBLEMA 2 (2 punts)

Diseñar un Sistema Lógico Combinacional que calcule la operación

$$C = \text{TRUNCAR}((2*A+B-1) / 2)$$

considerando que A, B y C son números enteros representados en *Complemento a dos* con dos bits cada uno. El sistema también ha de generar como salida un bit de **OVERFLOW** que se activa en caso de que el resultado no se pueda representar en los 2 bits de C. En ese caso el valor de C quedará indeterminado.

Nota. La función TRUNCAR devuelve el entero más pequeño, p.e. TRUNCAR(1.4) = 1 y TRUNCAR(-1.4) = -2.

a) Construir la tabla de verdad de este sistema indicando claramente los bits que corresponden a los números A, B y C, y a la salida OVERFLOW.

b) Sintetizar a dos niveles de forma mínima las funciones lógicas de C y OVERFLOW.

a) Tabla de verdad para la salida C y la salida OVERFLOW (O):

A	A ₁	A ₀	B	B ₁	B ₀	C	C ₁	C ₀	O
0	0	0	0	0	0	-1	1	1	0
0	0	0	1	0	1	0	0	0	0
0	0	0	-2	1	0	-2	1	0	0
0	0	0	-1	1	1	-1	1	1	0
1	0	1	0	0	0	-1	1	1	0
1	0	1	1	0	1	1	0	1	0
1	0	1	-2	1	0	-1	1	1	0
1	0	1	-1	1	1	-1	1	1	0
-2	1	0	0	0	0	-3	X	X	1
-2	1	0	1	0	1	-2	1	0	0
-2	1	0	-2	1	0	-4	X	X	1
-2	1	0	-1	1	1	-3	X	X	1
-1	1	1	0	0	0	-2	1	0	0
-1	1	1	1	0	1	-1	1	1	0
-1	1	1	-2	1	0	-3	X	X	1
-1	1	1	-1	1	1	-2	1	0	0

b) Síntesis mínima a dos niveles, para ello realizaremos el mapa de Karnaugh para cada bit de salida:

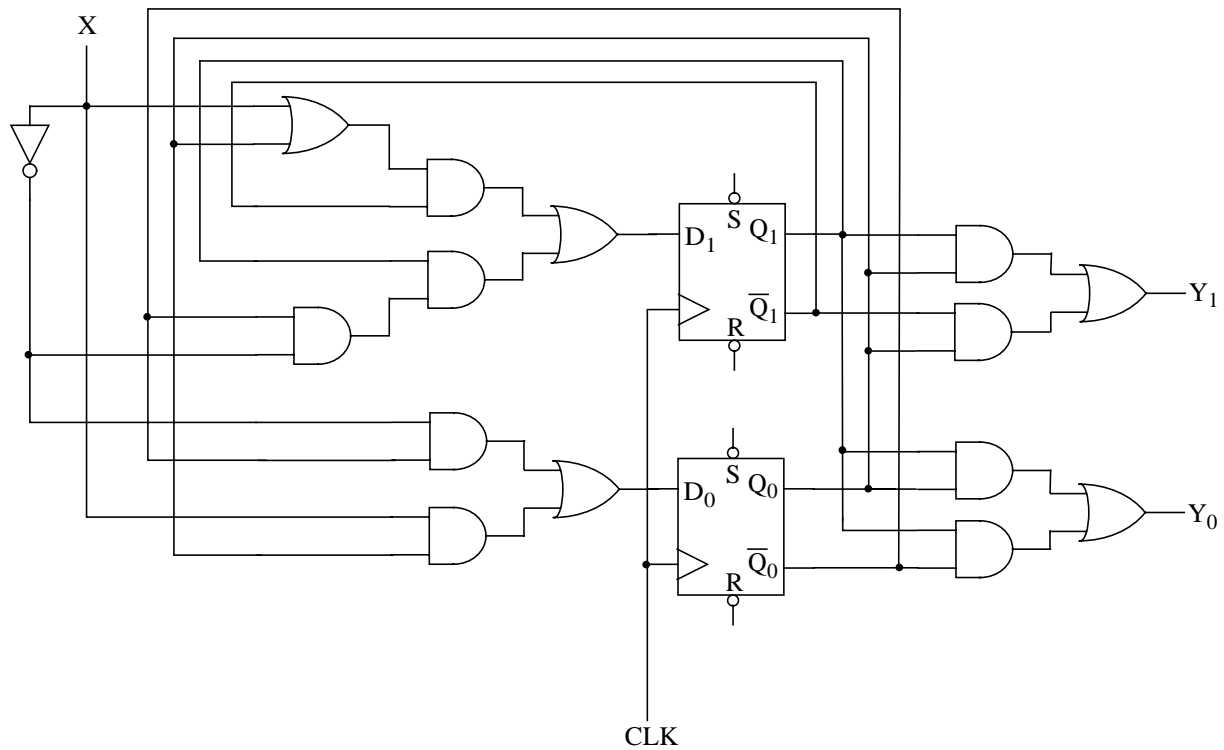
A ₁ A ₀ /B ₁ B ₀	00	01	11	10
00	1	0	1	1
01	1	0	1	1
11	1	1	1	X

A_1A_0 $/B_1B_0$	00	01	11	10
10	X	1	X	X

Solució

PROBLEMA 3 (3 punts)

Donat el següent sistema seqüencial:



Es demana:

- Obtenir la taula de transició d'estats, la taula de sortides i el graf d'estats de Moore.

- b) Suposar que volem afegir una senyal de Reset al circuit. Volem que la sortida d'aquest sistema, mentre la senyal de Reset està activa, sigui $Y_1=1$ i $Y_0=0$. Modifica el circuit anterior per aconseguir aquesta funcionalitat. NOTA: El senyals de Set i Reset son actius a baix nivell (0 lògic).
- c) Per l'anterior sistema, considerar els següents retards per les portes lògiques: AND,OR=10 unitats de temps. NOT=0 unitats de temps. Es desitja que el circuit funcioni amb un temps de cicle de 25 unitats de temps. Explica si el circuit anterior pot funcionar correctament. En cas que no, proposa una possible modificació del circuit.

PROBLEMA 4 (3 punts)

Es vol construir un sistema seqüencial per controlar el nombre de cops consecutius que un bit **Data** val 1. El valor màxim d'aparicions consecutives que volem controlar es **6**. El sistema ha de generar una sortida **Num** indicant el nombre d'uns consecutius detectats en cada moment. Cada cop que finalitzi una seqüència consecutiva d'uns cal tornar a iniciar el procés, contant la longitud de la següent seqüència. Si el nombre d'uns en una seqüència supera el 6 activarem un senyal específic anomenat **Overflow**. Aquest senyal torna a zero cada cop que la seqüència d'uns que ha produït el desbordament finalitza.

Es demana:

- a) Indiqueu quines son les entrades i sortides del sistema, i quin nombre de bits tenen.
- b) Quants estats calen per descriure aquest sistema?
- c) Dibuixeu un graf d'estats utilitzant el model de Moore per aquest sistema. Indiqueu clarament la finalitat de cada estat, el valor de les sortides, així com els valors de les entrades a cada arc.
- d) Determineu una codificació pels estats del model de Moore. Fent servir aquesta codificació, implementeu aquest sistema fent servir bistables de tipus D. Indiqueu clarament la taula de transicions i de sortida del sistema. Feu servir dues memòries ROM per implementar les funcions de transició i de sortida. Dibuixar un esquema de la implementacio indicant les mides de les ROMs.
- e) Com s'ha de inicialitzar el sistema? Feu un esquema amb els senyals imprescindibles.