

- Mínimo de aprobación: un problema y un ejercicio
- Cada hoja debe tener Nombre y CI.
- Utilice solo un lado de las hojas

- Deben estar numeradas y la primer hoja debe decir el total de hojas
- Incluya un solo problema por hoja
- Sea prolijo

Ejercicio 1

a) Dados **A=01101100** y **B=10100111** indicar su valor decimal si están codificados en:

1. Binario natural
2. Enteros en magnitud y signo
3. Enteros en complemento a dos

b) Realizar la resta de **A – B** en **complemento a dos** indicando y justificando si hubo o no overflow.

c) Tomando **A** como parte entera y **B** como parte fraccionaria de un número en punto fijo convertirlo a punto flotante. Si la conversión no es exacta se deberá truncar indicando el error de conversión.

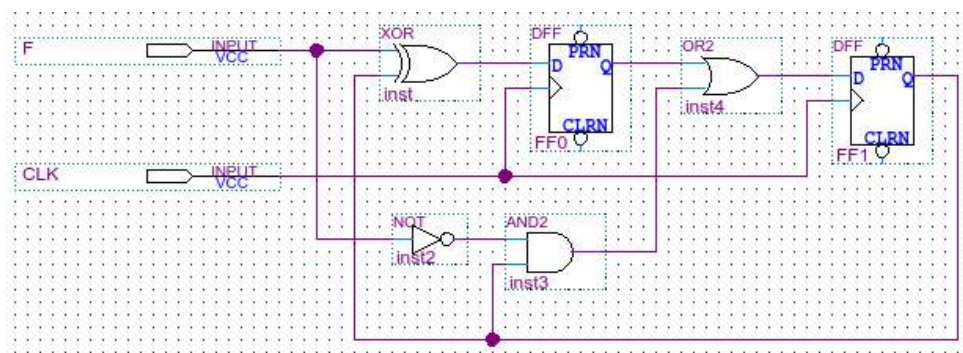
Utilizar la representación en punto flotante: 1 bit de signo, 5 bits para el exponente y 10 bits para la mantisa

$$(N=(-1)^s * 2^{(e-15)} * 1,f)$$

Ejercicio 2

En el circuito de la figura la entrada F está sincronizada con el flanco de subida del reloj CLK, con el retardo que se indica. Se pide:

a) Dar en forma literal todas las ecuaciones que determinan la frecuencia máxima de funcionamiento del circuito, indicando en cada caso a qué compuerta y a qué FF corresponde cada tiempo involucrado.



b) Calcular la frecuencia máxima en función de los siguientes valores numéricos:

$0 < t_G < 5\text{ns}$ retardo de las compuertas.

$0 < t_F < 10\text{ns}$ retardo de la entrada F con respecto al flanco de CLK.

$0 < t_P < 8\text{ns}$ tiempo de propagación de los FF.

$t_{SU} = 9\text{ns}$ tiempo de set-up de los FF.

$t_H = 0\text{ns}$ tiempo de hold de los FF.

Problema 1

Se requiere construir un circuito modo nivel para medir el tiempo de reacción de un paciente a una señal luminosa. Para ello se dispone de una señal **L** que indica en 1 cuando una luz está encendida y un botón **B**.

La luz se enciende y apaga en forma rápida a intervalos más o menos regulares, y el paciente debe responder oprimiendo el botón cuando detecta que la luz se encendió.

Si el paciente oprime el botón (**B=1**) cuando la luz está encendida, deberá activarse la salida **OK=1** hasta que la luz se apague.

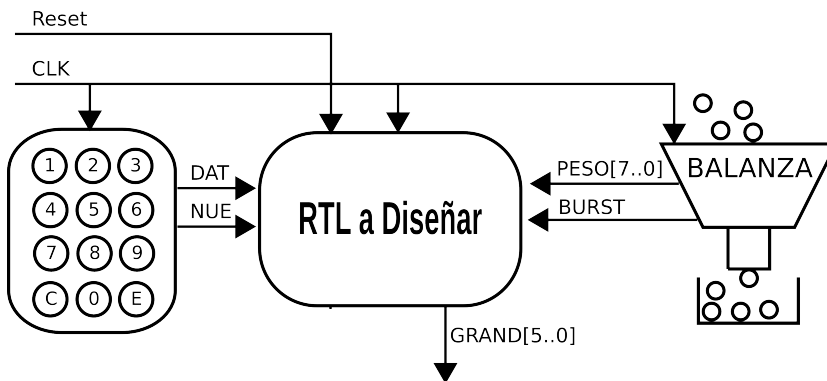
Si el paciente oprime el botón antes que la luz se encienda, entonces la salida **OK** no debe activarse, pero si suelta el botón y lo oprime nuevamente antes que la luz se apague, entonces sí debe activarse **OK**.

Si el paciente oprime el botón después que la luz se apagó, entonces **OK** no debe activarse.

Diseñar un circuito modo nivel libre de carreras, espurios y azares que tenga como entradas **L** y **B** y genere la señal **OK**.

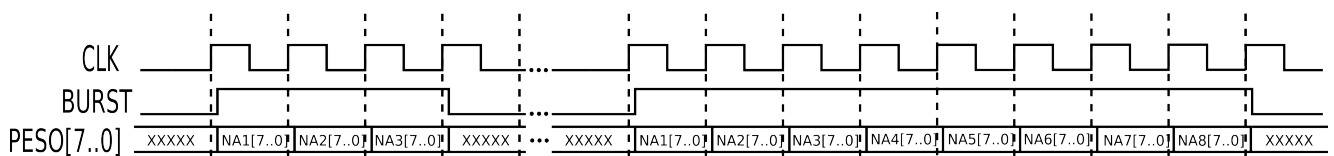
Problema 2

Se desea diseñar un sistema de control que permita determinar una característica acerca de un cajón de naranjas: se desea conocer la cantidad de naranjas que superan un peso mínimo establecido. La siguiente figura muestra el diagrama de bloques del sistema.



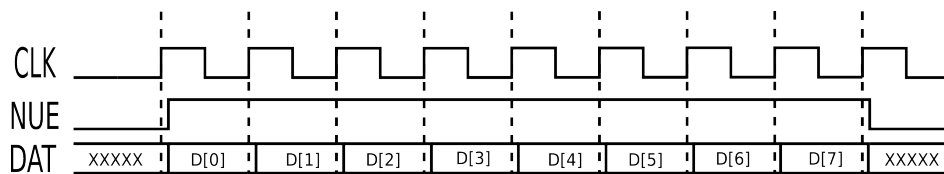
Para la implementación del sistema se cuenta con una balanza a la cual se le coloca el cajón de naranjas y esta pesa cada una de ellas. El peso de cada naranja es un número de 8 bits. Una vez que tiene el peso de cada naranja los envía en ráfaga, una medida de peso por período de reloj a través de su salida **PESO[7..0]**. Cada cajón tiene a lo sumo 63 naranjas.

Para indicar que se encuentra enviando una ráfaga, cuenta con una salida **BURST** la cual mantiene en alto ("1") mientras se encuentra enviando una ráfaga. A continuación se muestran dos ejemplos de envío de pesos por parte de la balanza. Se garantiza que entre ráfagas distintas **BURST** tiene un nivel bajo de varios períodos de reloj.



Durante cada ráfaga el circuito RTL a diseñar debe contar la cantidad de naranjas que superan un peso mínimo (**UMBRAL[7..0]**). Luego de un reset, el **UMBRAL[7..0]** se debe establecer en 128, sin embargo este puede modificarse durante el funcionamiento del sistema. Para esto, se cuenta con un teclado numérico. Cuando un usuario presiona un número (entre 0 y 255) y luego la tecla E, éste envía el número a través de sus salidas **DAT** y **NUE** utilizando el siguiente protocolo serie. La señal **NUE** se mantiene en bajo mientras no se está enviando ningún dato. Esta señal sube al mismo tiempo que en la señal **DAT** se encuentra el LSB del dato a enviar. A continuación se envían los 8 bits de datos, uno por período de reloj. La señal **NUE** se mantiene en alto durante toda la transmisión de datos. A continuación se muestra un ejemplo del envío de un dato por parte del teclado.

El circuito a diseñar debe ser capaz de recibir un dato a través del teclado en cualquier momento y



actualizar la variable **UMBRAL[7..0]**. Sin embargo, si el dato se recibe durante la mitad de una ráfaga (**BURST=1**), se debe esperar a que esta se termine para actualizar el valor de **UMBRAL[7..0]** a utilizar.

Adicionalmente el circuito a diseñar debe contar con una salida **GRAND[5..0]** la cual debe indicar la cantidad de naranjas con peso mayor al mínimo. La salida debe indicar la cantidad de la última ráfaga de datos obtenida desde que **BURST** baja a "0" hasta que sube para comenzar una nueva ráfaga. Mientras se está recibiendo una ráfaga de pesos se deben mantener la salida en 0.

Se pide: descripción RTL, bloque de datos y control.

Solución Problema 1

Diagrama de estados LB/Ok

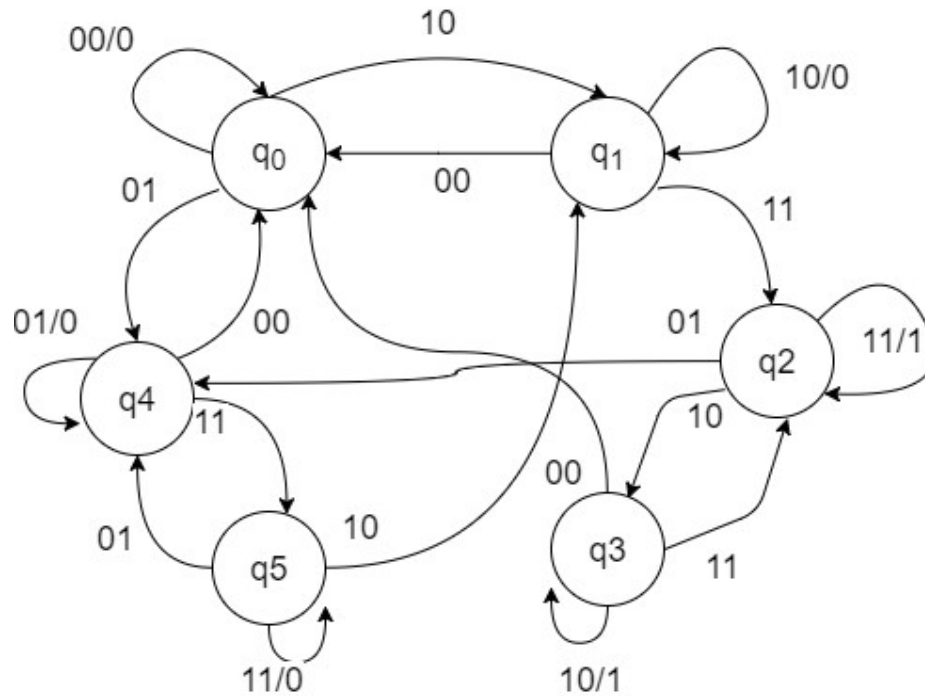
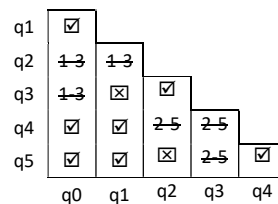


Tabla de estados

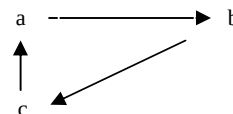
q _n \LB	q _{n+1}				Ok			
	00	01	11	10	00	01	11	10
q0	q0	q4		q1	0			
q1	q0		q2	q1				0
q2		q4	q2	q3			1	
q3	q0		q2	q3				1
q4	q0	q4	q5			0		
q5		q4	q5	q1			0	

Minimización de estados



Parejas compatibles: (0,1) (0,4) (0,5) (1,4) (1,5) (2,3) (4,5), no hay implicancias
hay varias formas de minimizar: (0,4,5) (1) (2,3) otra (0) (1,4,5) (2,3); elegimos (0,1) (2,3) (4,5)

Eliminación de carreras, como son 3 estados se puede usar el renglón vacío, o se puede hacer un ciclo en la transición de b hacia c, pasando por a; ya que es carrera no crítica.



qn\LB	qn+1				Ok			
	00	01	11	10	00	01	11	10
a	a	c	b	a	0			0
b	a	a	b	b			1	1
c	a	c	c	a		0	0	

Eliminación de espurios

qn\LB	qn+1				Ok			
	00	01	11	10	00	01	11	10
a	a	c	b	a	0	0	X	0
b	a	a	b	b	0	X	1	1
c	a	c	c	a	0	0	0	0

Codificación

a0 – 00, b – 01, c – 10

qn\LB	qn+1				Ok			
	00	01	11	10	00	01	11	10
00	00	10	01	00	0	0	X	0
01	00	00	01	01	X	X	1	1
11	XX	XX	XX	XX	X	X	X	X
10	00	10	10	00	0	0	0	0

Mapas k + eliminación de azares

Y1

y1 y0\L.B	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	X	X	X	X
10	0	1	1	0

$$Y1 = !L.B.!y0 + y1.B$$

Y0

y1 y0\L.B	00	01	11	10
00	0	0	1	0
01	0	0	1	1
11	X	X	X	X
10	0	0	0	0

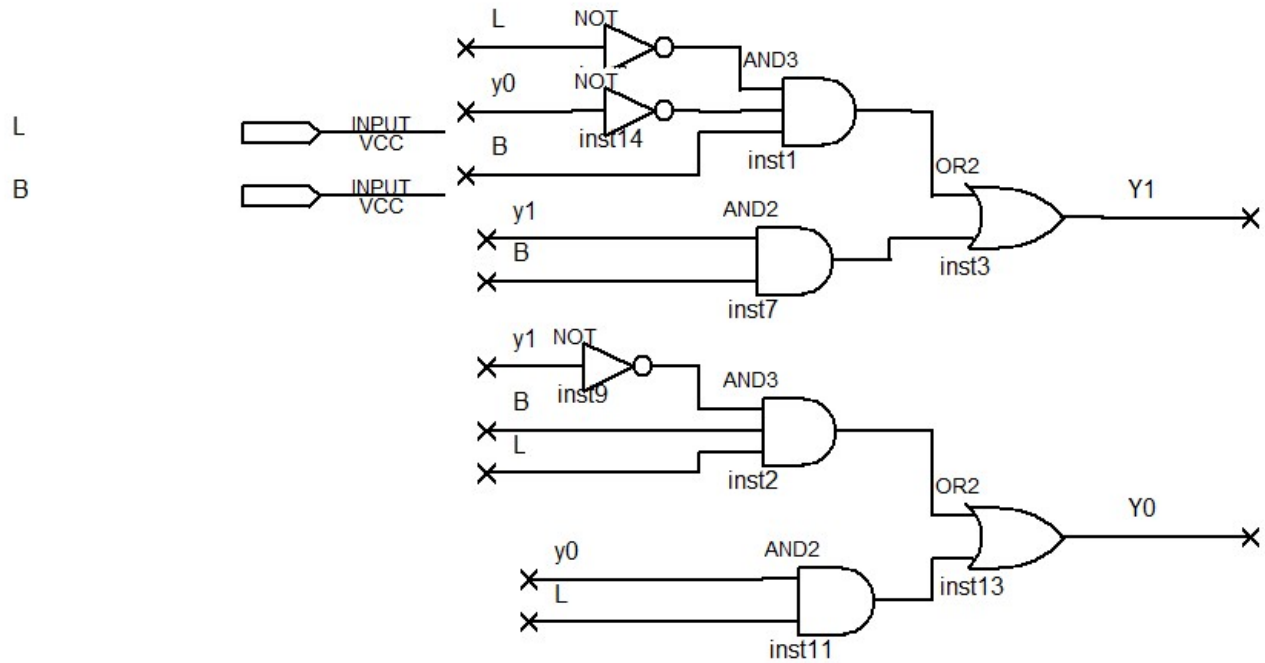
$$Y0 = L.B.!y1 + y0.L$$

Ok

y1 y0\L.B	00	01	11	10
00	0	0	X	0
01	X	X	1	1
11	X	X	X	X
10	0	0	0	0

ON = y0

Circuito



Solución

Secuencia

MODULE: RTL_EX_DIC2017

INPUTS: DAT, NUE, PESO[8], BURST

MEMORY: UMB_A[8], UMB_F[8], CONT_GRA[6]

OUTPUTS: GRAND[6]

0. $\text{CONT_GRA}[5..0] \leftarrow 000000 \text{ b}$
 $\text{UMB_A}[7..0] \leftarrow 1000 \ 0000 \text{ b}$
 $\text{UMB_F}[7..0] \leftarrow 1000 \ 0000 \text{ b}$
1. $\text{CONT_GRA}[5..0] * \text{BURST} \leftarrow \text{grande. } 00001$
 $\text{UMB_F}[7..0] * \text{NUE} \leftarrow \text{DAT}, \text{UMB_F}[7..1]$
 $\text{UMB_A}[7..0] * \text{NUE} * \text{BURST} \leftarrow \text{UMB_F}[7..0]$
 $\rightarrow (!\text{BURST}, \text{BURST}) / (1, 2)$
2. $\text{CONT_GRA}[5..0] * \text{BURST} * \text{grande} \leftarrow \text{INC}(\text{CONT_GRA}[5..0])$
 $\text{UMB_F}[7..0] * \text{NUE} \leftarrow \text{DAT}, \text{UMB_F}[7..1]$
 $\text{UMB_A}[7..0] * \text{NUE} * \text{BURST} \leftarrow \text{UMB_F}[7..0]$
 $\rightarrow (!\text{BURST}, \text{BURST}) / (1, 2)$

END SEQUENCE

CONTROL RESET (0)

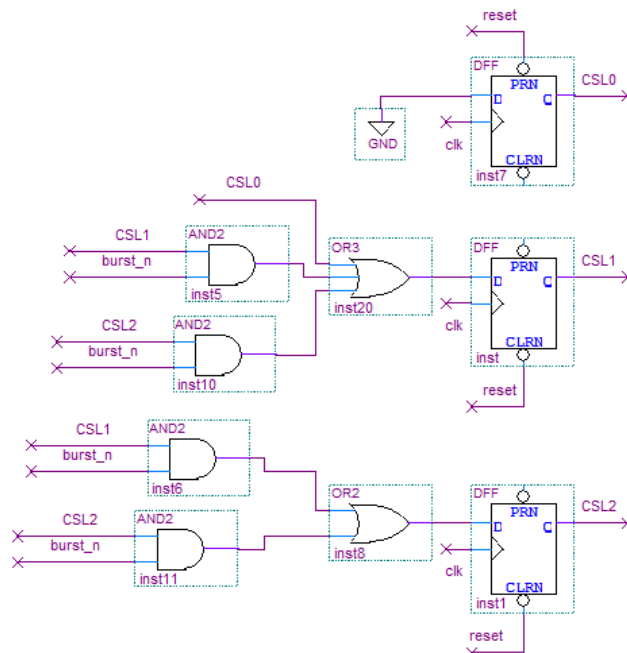
$\text{grande} = \text{MAYOR}(\text{PESO}[7..0], \text{UMB_A}[7..0])$

$\text{GRAND}[5..0] = !\text{BURST. CONT_GRA}[5..0]$

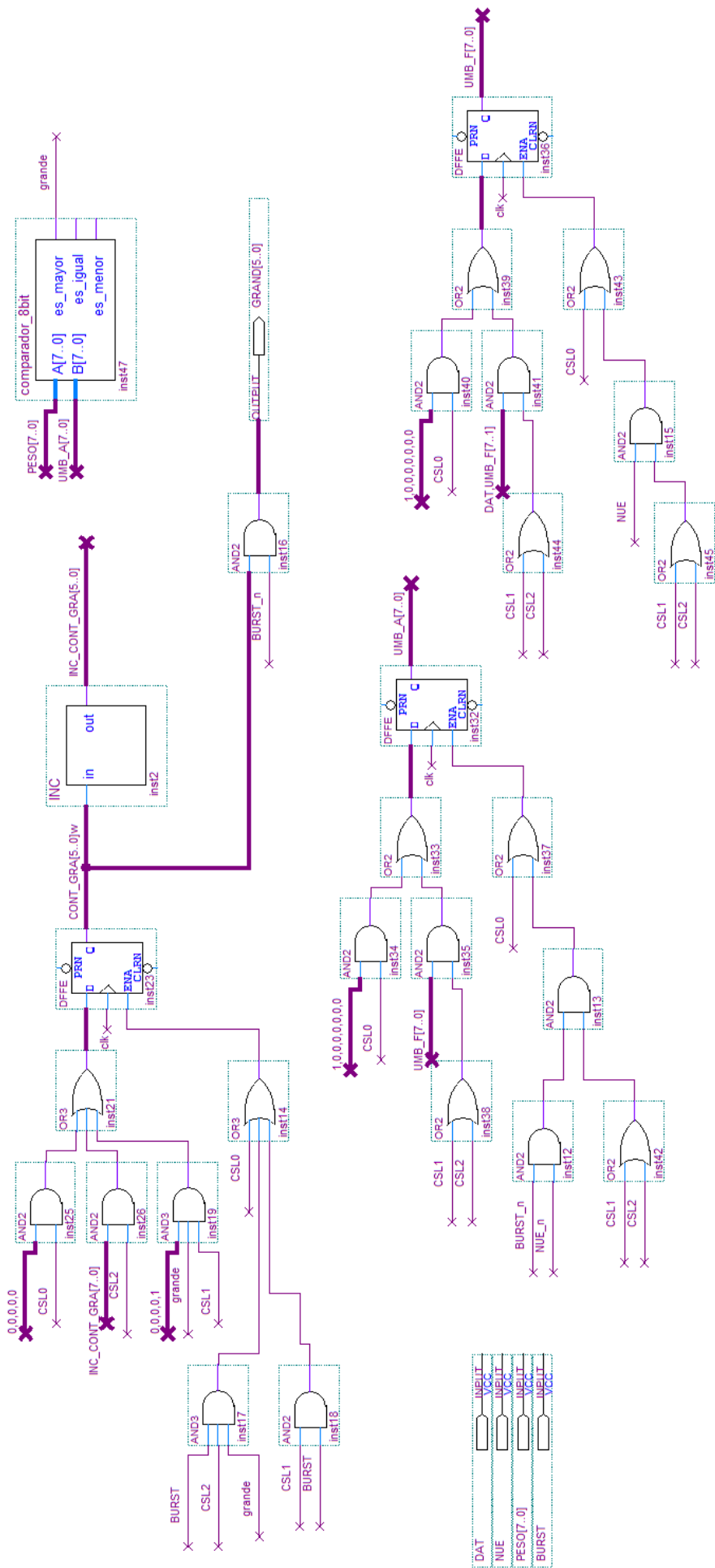
END

Bloque

Control



Bloque
Datos



Ejercicio 1

a) 1. A=108, B=167

2. A=108, B=-39

3. A=108, B es negativo, calculo opuesto $01011000+1=01011001 \Rightarrow B=-89$

b) $01101100 + 01011001 = 11000101$, hay overflow porque 2 sumandos positivos dan resultado negativo (también puede verse que hay acarreo entrante sobre el bit7 y no hay saliente)

c) Pto. fijo = 01101100,10100111

Para pasarlo a punto flotante se corre la coma 6 lugares 01,10110010100111

s=0

f=1011001010

e-15=6, e=21, e=10101

Error $111 \times 2^{-8} = 7 \times 2^{-8} = 0,02734$

Se puede calcular también como $1/64+1/128+1/256=0,02734$

Ejercicio 2

a) $T1 \geq t_{Fmax} + t_{Gxormax} + tsu0$

$T2 \geq t_{Fmax} + t_{Gnotmax} + t_{Gandmax} + t_{Gormax} + tsu1$

$T3 \geq t_{P0max} + t_{Gormax} + tsu1$

$T4 \geq t_{P1max} + t_{Gandmax} + t_{Gormax} + tsu1$

$T5 \geq t_{P1max} + t_{Gxormax} + tsu0$

$T1 \geq 10+5+9=24$

$T2 \geq 10+5+5+5+9=34$

$T3 \geq 8+5+9=22$

$T4 \geq 8+5+5+9=27$

$T5 \geq 8+5+9=22$

$f_{max} = 1/34ns = 29.4MHz$