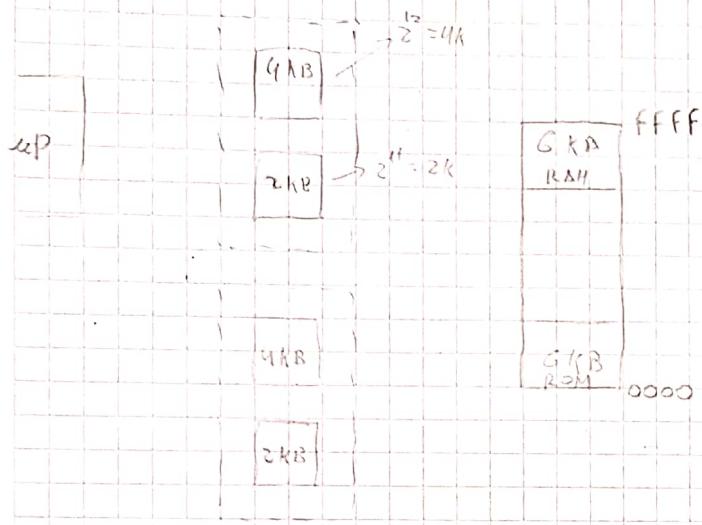


13/04/2017

Mapeo:



A_{31}

	0	0	0	0	0	0	0	0	A_0
A_{31}	0	0	0	0	0	0	0	0	
	3	0	0	0	0	0	0	0	{ 16 MB
	8	0	F	F	F	F	F	F	
	8	1	0	0	0	0	0	0	{ 16 HB
	8	1	F	F	F	F	F	F	
	8	2	0	0	0	0	0	0	{ 16 HB
	8	2	F	F	F	F	I	F	
	3	3	0	0	0	0	0	0	{ 16 HB
	3	3	I	F	F	F	F	F	

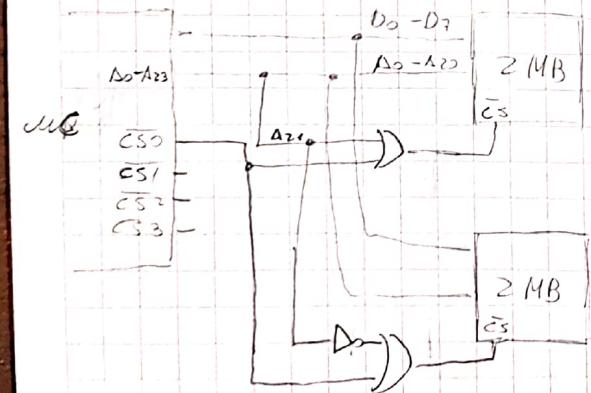
2/03/2017

Banco

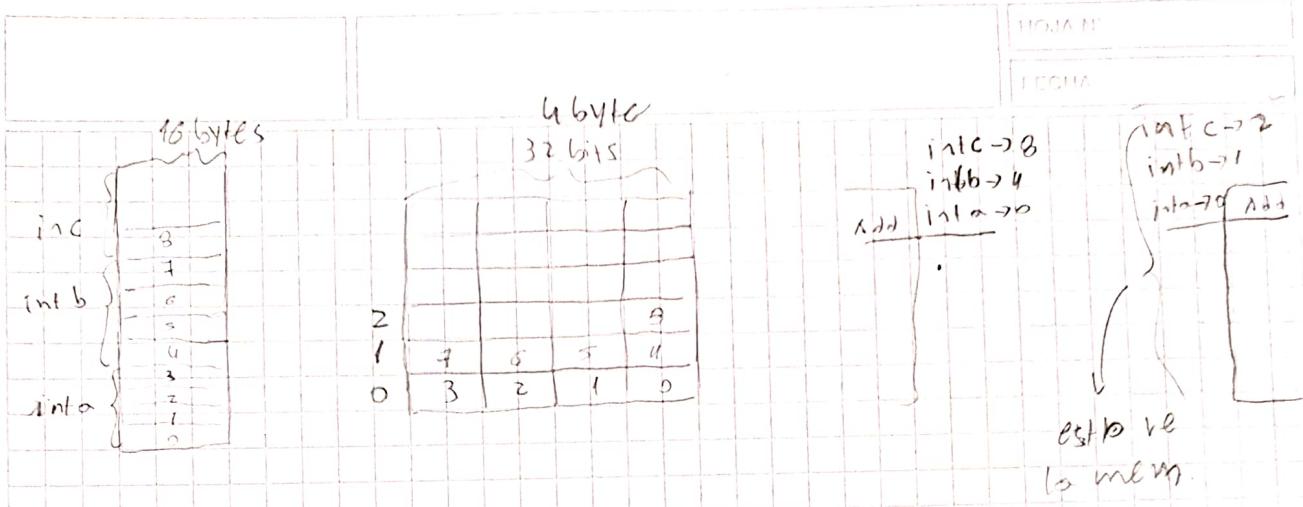
	0	A ₂₃	iH						
estó	8	0	0000	0	0	0	0	0	
ej.	8	0	0001	F	F	F	F	F	
internos	8	0	0010	I	I	I	I	I	
del	8	0	0011	F	F	F	F	F	
MC	8	0	0100	0	0	0	0	0	
(para la comparación)	8	0	0101	F	F	F	F	F	
etc.	8	0	0110	I	I	I	I	I	

ext

Mostrar 2 mem. de 2MB cada uno
 con LPC2214, en banco 0
 hacer circuito.



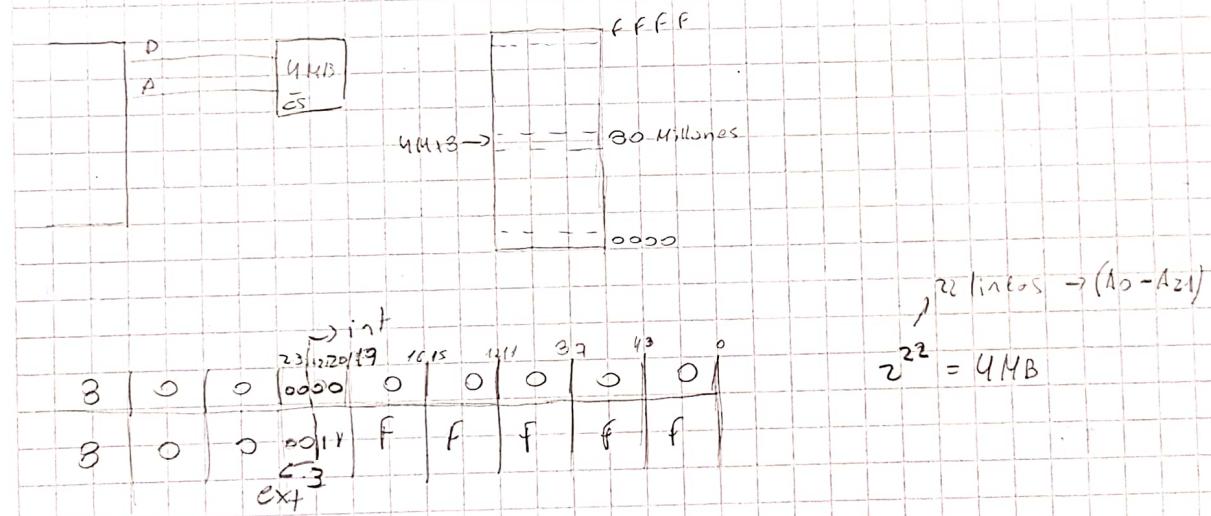
NOTA



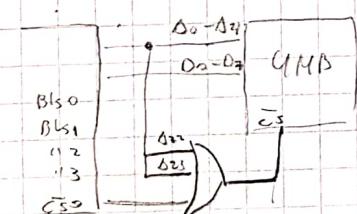
16/05/2017

(1)

4 MB → 4M × 8



(2)



Los BLS sirven para

avisarle al micro que sum datos de 16 bits

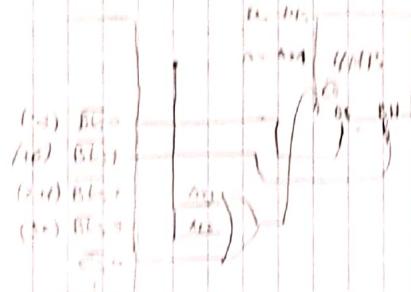
Q. 1. 1992 का एक अधिकारी

प्रति वर्ष दो लाख रुपये की बैंक बॉलरी का बिल आया।

2

वह बैंक की बौद्धि विशेष
बैंक वालों की बौद्धि

सभी वर्ष दो लाख रुपये की बैंक बॉलरी का बिल आया।

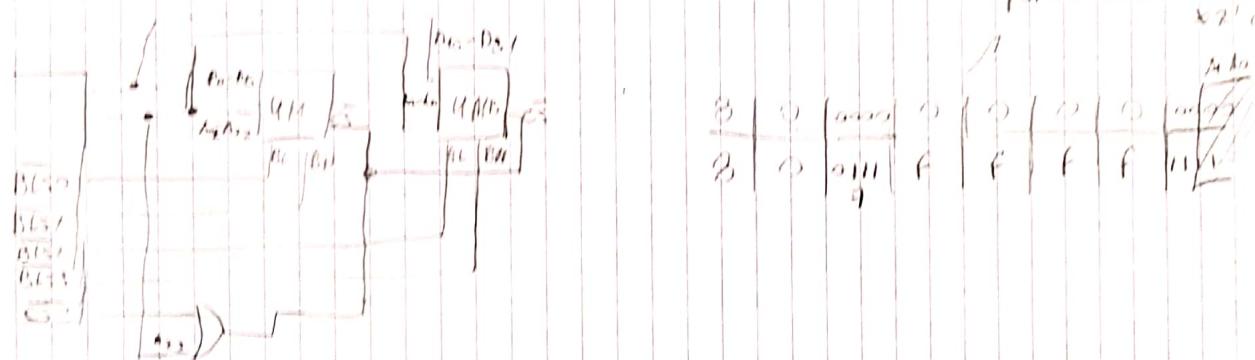


4 MB

प्रति वर्ष दो लाख रुपये की बैंक बॉलरी का बिल 11

प्रति वर्ष दो लाख 8 MB

2 लाख का बिल 11

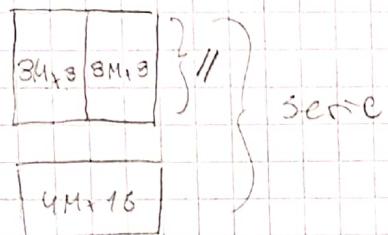


3 Q / 5 / 2019

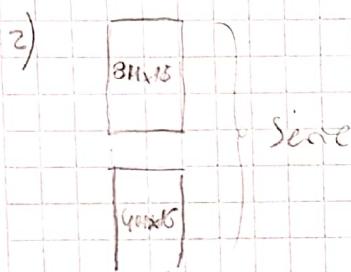
प्रति वर्ष दो लाख रुपये की बैंक बॉलरी का बिल 11.

- 1) 2 लाख 8 MB × 8 = 16 MB = 64 MB
- 2) 1 लाख 8 MB × 16 = 16 MB = 64 MB
- 3) 1 लाख 8 MB × 16 = 16 MB = 32 MB
- 4) 1 लाख 8 MB × 16 लाख = 16 MB = 32 MB
- 5) 2 लाख 4 MB × 8 = 2 लाख 32 MB

1) $2 \times 3MB$ y $1BM$



no tiene el mismo tamaño
∴ no se puede en //

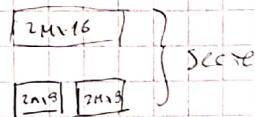
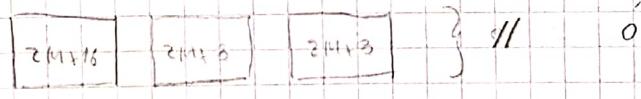


3) No se puede

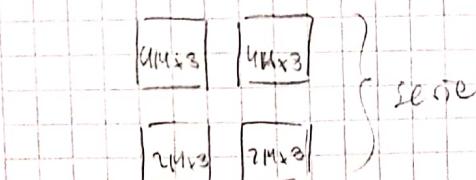
x3 no son compatibles

ni en tamaño ni en bds de datos

4) $2M \times 16 + 2 \times 2M \times 3$

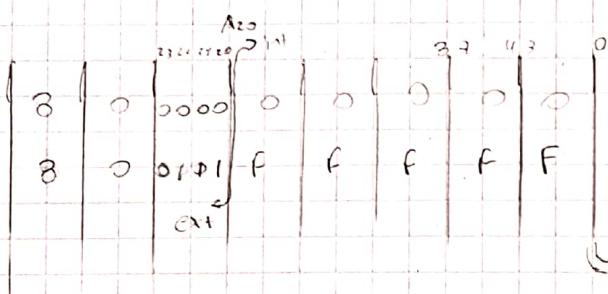


5) $2 \times 4M \times 3$ y $2 \times 2M \times 3$



NOTA

Manpear UCI se $2M \times 3$ en un bus de 32 bits → posib
se 80000000 (hexa)



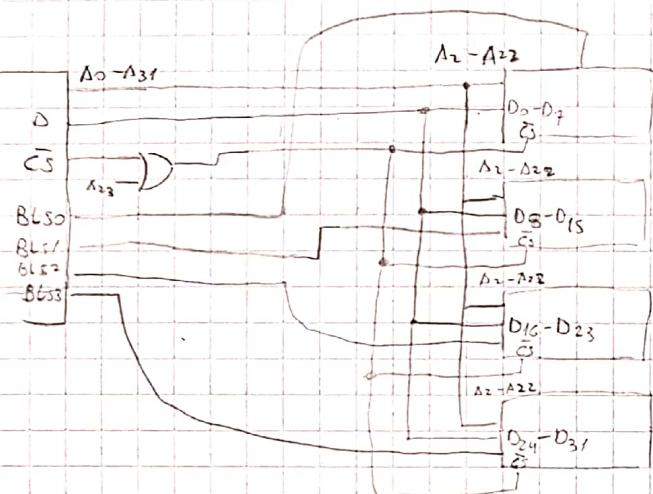
UCI se $2M \times 3$

≡

1 C2 se $2M \times 32$

8 MB (esta se tiene?)
mappear

↳ A₀-A₁ b eliminado



06/06/2017

Assembler

add r1, r2 → r1 ← suma r2

mov r1, #100

mov r2, #200

add r1, r2

add r2, r1, r2

↳ Poco que cambie es startes

add c → hace la suma con memoria

NOTA

Mov R1, #100

Mov R2, #200

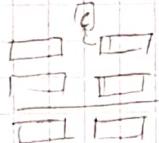
Mov R3, #20

Mov R4, #30

Adds R3, R4

adc R1, R2

$$+ \begin{array}{r} R1 \\ R2 \end{array} \begin{array}{r} R3 \\ R4 \end{array}$$



Adds R1, R2

$$+ \begin{array}{r} R1 \\ R2 \end{array} \begin{array}{r} R3 \\ R4 \end{array}$$

adc R0, #0

$$+ \begin{array}{r} R0 \\ R1 \end{array} \begin{array}{r} R3 \\ R4 \end{array}$$

Adds R1, R3

$$+ \begin{array}{r} R0 \\ R1 \end{array} \begin{array}{r} R3 \\ R4 \end{array}$$

adc R0, #0

$$+ \begin{array}{r} R0 \\ R1 \end{array} \begin{array}{r} R3 \\ R4 \end{array}$$

Calcular el valor absoluto de un número.

Mov R2, #0

Cmp R1, #0

ble: negativo

Mov R2, R1

b sigue

Negativo: Mov R3, #0

Sub R3, R1

Mov R2, R1

b sigue.

Sigue: -

-

13/06/2017

if ($r_0 > 20$)

$r_0 = 20;$

mov r0, #...

mov r1, r0

cmp r1, #20

bhi mayor

bls size = (No pudeo usar la misma etiqueta)

mayor:

mov r0, #20

b size

size: =

cmp r0, #20

movhl r0, #20

cmp r0, #20

bls menor

mov r0, #20

menor: =

if ($r_0 < 0$)

$r_1 = 1;$

else

$r_1 = 0;$

mov r0, #...

mov r2, #0

comp r2, #0

bls mayor

blo menor

mayor:

mov r1, #0

b size

menor:

mov r1, #1

b size

size: =

cmp r0, #0

movhl r1, #1

movpl r1, #0

cmp r0, #0

bmi negativo

mov r1, #0

b continuar

negativo:

mov r1, #1

continuar:

if ($r_1 := 0$)

$r_1 = 1;$ distintos

else

$r_1 = 0;$

cmp r1, #0

beq igual

distintos: =

b continuar

igual: =

continuar: =

if ($r_2 > -20$) if ($r_0 < 10$) if ($r_0 > 10 \& r_0 < 20$)
 si : $r_1 = 1$; si :
 no : else no :
 cmp $r_2, \# -20$ $r_1 = 0$; cmp $r_0, \# 10$
 bgt si cmp $r_0, \# 10$ bls no
 no : mov lt $r_1, \# 1$ cmp $r_0, \# 20$
 b continuar mov ge $r_1, \# 0$ bhs no
 si : si : =
 = b continuar
 continuar : no : =
 =
 if ($r_0 < 10 \& r_0 > 20$)
 si : if ($r_1 = '=' \& (r_1 > 'a' \& r_1 \leq 'z')$)
 no :
 cmp $r_0, \# 10$ si :
 blt si no :
 cmp $r_0, \# 20$ cmp $r_1, \# 'n'$
 bgt si beq si
 no : cmp $r_1, \# 'a'$
 b continuar bto no
 si : cmp $r_1, \# 'z'$
 = bhi no
 continuar : si : =
 =
 b continuar b cont
 no : no : =
 continuar : cont : =

TECNICAS DIGITALES II

HOJA N°

FECHA

Final 23/02/2017

1)

$$\Delta V = -10 - 60^{\circ}\text{C} \rightarrow \Delta V = 70^{\circ}\text{C}$$

$$\text{res} = 0,01^{\circ}\text{C}$$

$$\text{cuantos} = \frac{70^{\circ}\text{C}}{0,01^{\circ}\text{C}} = 7000$$

+26

$$n = \frac{\ln(\text{cuantos})}{\ln(2)} = \frac{\ln(7000)}{\ln(2)} = 12,77 \rightarrow \boxed{13 \text{ bits}}$$

2)

FALTAN DATOS

3) Con $n = 13$ bits

$$E_{\text{Total}} = 0,05\%$$

$$E_a = \frac{1}{2^n} = \frac{1}{2^{13}} = \boxed{30,51 \text{ PPM}}$$

$$4) \text{Tolerancia} = \frac{E_a}{10} = \frac{30,51 \text{ PPM}}{10} = \boxed{3,05 \text{ PPM}}$$

5)
?

6)
?

7)
?

8) La configuración de interrupciones requiere de los siguientes pasos.

* Configuración del periférico: Además de la configuración de los registros de la aplicación, se deben definir cuáles y cuántos eventos del micro generan interrupciones. Esto se debe a que los periféricos generan más de un evento.

* Configuración de la NVIC: Aquí debe configurarse el controlador de interrupciones para que atienda la interrupción. También se puede definir la prioridad.

NOTA

* Vector de interrupción: Despues de los items anteriores, se debe programar la función que atenderá la interrupción. Esta se escribirá como una función estandar, pero debe cargarse la dirección de la misma en el vector de interrupciones.

9) La diferencia entre saltos condicionales lógicos y aritméticos radica justamente en el condicional que se use.

10) Existen 3 tipos de punto flotante:

- Float (32 bits)
- Double (64 bits)
- Long Double (80 bits)

Campos	Tamaño
Sígno	1
Exponente	8
Mantisa	23
Total	32

Campos	Tamaño
Sígno	1
Exponente	11
Mantisa	52
Total	64

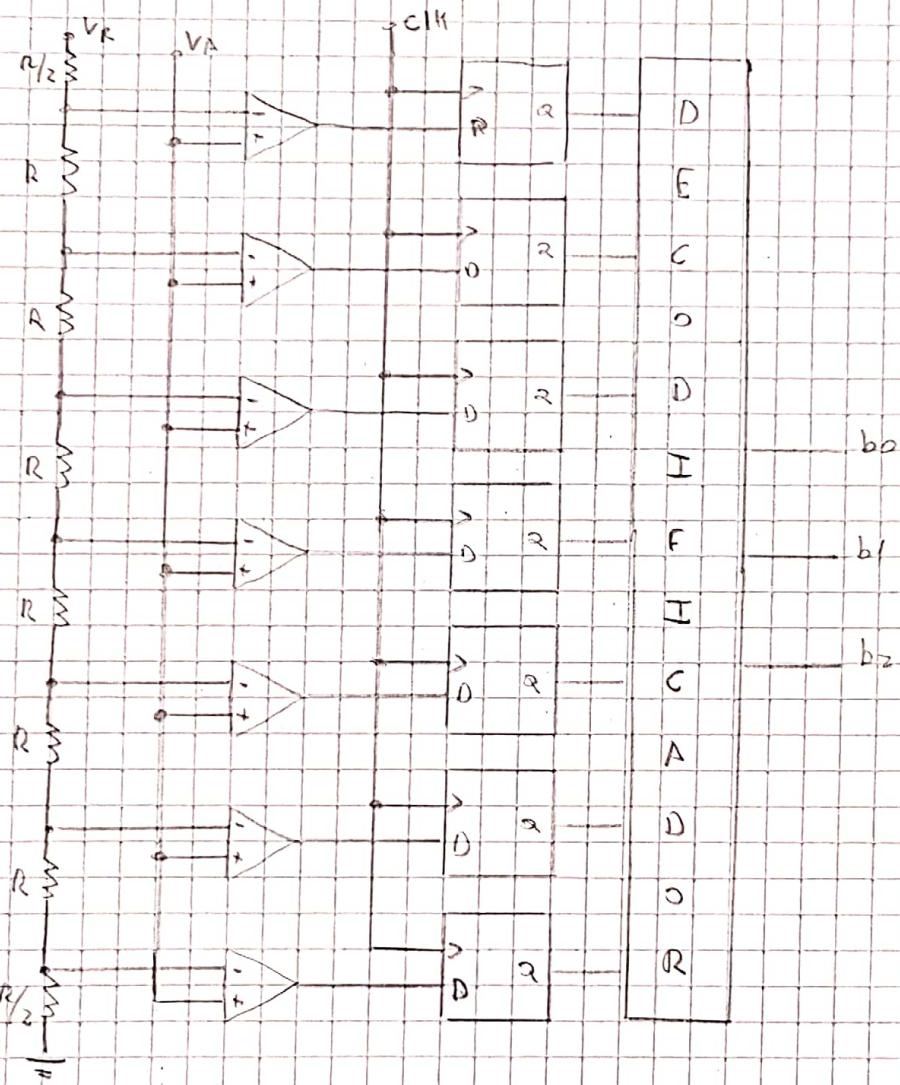
Campos	Tamaño
Sígno	1
Exponente	15
P. Ent. Mant.	1
Mantisa	63
Total	80

Por lo entero de Mantisa.

Final 4/07/2017

1) FALTAN DATOS

2) Esquema de conversor Flash: para 3 bits



3) El error de cuantización se define como la mitad del bit menos significativo (LSB):

$$\boxed{e_q = \pm \frac{1}{2} \text{ LSB}}$$

No necesariamente se debe duplicar el número de bits para reducir a la mitad el error, pero si se debe aumentar el número de bits.

Final 27/07/2017

1) Suponemos:

$$\lambda(1) = 3 \operatorname{sen}(3000t) ; V_{ref} = 3V$$

$$n = ?$$

$$V_{res} = 0,5mV$$

$$t_5 = ?$$

$$V_p = 3V \therefore V_{pp} = 6V$$

$$\text{Cuentos} = \frac{V_{pp}}{V_{res}} = \frac{6V}{0,5mV} = 12000 \text{ cuentos}$$

$$n = \frac{\ln(\text{cuentos})}{\ln(2)} = 13,55 \Rightarrow \boxed{14 \text{ bits}}$$

$$t_5 = \frac{1}{2^n \cdot 2\pi \cdot F} = \frac{1}{2^n \cdot \omega} = \frac{1}{2^{14} \cdot 3000} = \boxed{20,34 \text{ nSes}}$$

2)

$$G = \frac{V_{LSB}}{V_{res}} \quad \frac{V_{LSB}}{V_{res}} = \boxed{0,366}$$

$$3) E_G = \frac{1}{4 \cdot 2^n} = \frac{1}{4 \cdot 2^{14}} = \boxed{15,26 \text{ ppm}}$$

$$4) \text{Tolerancia} = \frac{E_G}{10} = \boxed{1,526 \text{ ppm}}$$

$$5) e_3 \rightarrow \pm \frac{1}{2} \text{ LSB}$$

No necesariamente se debe duplicar el n de bits para reducir el error a la mitad.

6) mov r2, #3

mov r1, #h

cmp r0, r1

add r3, r0, r0, lsl 1

add r4, r0, r0, lsl 1

7) add r2, r0, r0, lsl #5 $\rightarrow r2 = r0 + 32r0 = 33r0$

add r2, r1, r2, r0, lsl #1 $\rightarrow r2 = 33r0 + 2r0$

8) Una máscara de interrupción condiciona la interrupción para pasar de la etapa de solicitud a la etapa de reconocimiento de la misma. Hay interrupciones enmascarables y no enmascarables.

9) ?

10) *float: 32 bits

* Double: 64 bits

*long Double: 80 bits

• Signo: 1 bit

• Signo: 1 bit

• Signo: 1 bit

• Exponente: 8 bits

• Exponente: 11 bits

• Exponente: 15 bits

• Mantisa: 23 bits

• Mantisa: 52 bits

• Mantisa: 63 bits

• Total: 32 bits

• Total: 64 bits

• P. entero mantisa: 1 bit

• Total: 80 bits

NOTA

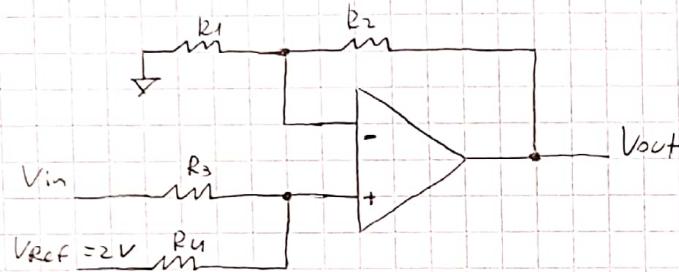
Reposo ADC (CAS) 2017:

- 1) Se desea leer un ADC, una señal que varía entre -2 V y $+2 \text{ V}$. El ADC posee una entrada de 0 a 5 V . Realizar el circuito de adaptación de señal.

-2 V } Rango de 4 V
 $+2 \text{ V}$

0 V } Rango de 5 V
 5 V

$$G = \frac{\Delta V_{out}}{\Delta V_{in}} = \frac{5}{4} = 1.25$$



$$V_o = \frac{R_2}{R_3 + R_4} \cdot V_{in} + \frac{R_2}{R_4} \cdot V_{ref}$$

$$S = \frac{R_2}{R_3} \cdot 2 \cdot \frac{R_2}{R_4} \cdot 2 ; R_3 = R_4 \therefore S = \frac{R_2}{R_3} \cdot 4 \Rightarrow \frac{R_2}{R_3} = \frac{S}{4} = \frac{1.25}{4}$$

$$\frac{R_1}{R_3} (V_{in} + V_{ref})$$

Buscamos una Relación:

$$\text{Si } R_1 = R_2 = 15 \text{ k}\Omega \rightarrow R_3 = R_4 = \frac{R_1}{S/4} = 12 \text{ k}\Omega$$

- 2) Se desea medir una señal que varíe entre 0 y 2 V , con una resolución de 0.5 mV . Calcular ADC y amplificador.

0 V } Rango = 2 V
 2 V

$$\text{Cuentos} = \text{Pasos} = \frac{V_{pp}}{V_{res}} = \frac{2 \text{ V}}{0.5 \text{ mV}} = 4000 \text{ Pasos}$$

$$2^n = \text{Pasos} \rightarrow 2^n = 4000 \rightarrow n = \frac{\ln 4000}{\ln 2} = 11.96 \approx 12 \text{ bits}$$

cantidad
de bits

NOTA

$$2^{12} = 4096 \text{ pasos} ; V_{REF} = 5V$$

$$V_{ISB} = \frac{5V}{4096} \approx 1,22 \text{ mV} = \frac{V_{REF}}{2^n}$$

$$G = \frac{V_{ISB}}{V_{RES}} = \frac{\frac{V_{REF}}{2^n}}{V_{RES}} = \boxed{1,491406}$$

3) Se desea medir una señal que varía entre 0 y 3V, con una resolución de 0,5mV. Calcular ADC y amplificador.

$$0V \quad } \quad V_{PP} = 3V \quad \text{pasos} = \frac{V_{PP}}{V_{RES}} = \frac{3V}{0,5mV} = 6000 \text{ pasos}$$

$$n = \frac{\ln(6000)}{\ln(2)} = 12,55 \rightarrow 13 \text{ bits}$$

$$2^{13} = 8192 ; \quad G = \frac{V_{ISB}}{V_{RES}} = \frac{\frac{V_{REF}}{2^n}}{V_{RES}} = \boxed{1,22070}$$

$$V_{REF} = 5V$$

4) Se desea medir una señal que varía entre -1V y 1V, con una resolución de 0,1mV. calcular ADC y amplificador.

$$-1V \quad } \quad V_{PP} = 2V \quad ; \quad \text{pasos} = \frac{V_{PP}}{V_{RES}} = \frac{2V}{0,1mV} = 20000$$

$$n = \frac{\ln(20000)}{\ln(2)} = 14,23 \approx 15 \text{ bits} ; \quad V_{REF} = 5V$$

$$G = \frac{V_{ISB}}{V_{RES}} = \frac{\frac{V_{REF}}{2^n}}{V_{RES}} = \boxed{1,52537}$$

NOTA

5) Una balanza se pesa 0 a 5 Kg, es medida con un ADC de 12 bits de resolución. Calcule el menor paso posible en números enteros de gramos.

0 Grams

5000 grams

$$12 \text{ bits} = n \quad \text{Posos} = 2^n = 2^{12} = 4096 \text{ posos}$$

$$\frac{5000}{4096} = 1,22 \text{ Gramos} \rightarrow 12 \text{ Gramos}$$

$$5000g \xrightarrow{\leftarrow 1} 4096$$

6) Para el ejercicio 3, determinar los errores máximos admisibles en la tensión de referencia y la ganancia del amplificador.

$$\text{Ecuatización} = \frac{1}{2} \cdot V_{lsb} = \frac{1}{2} \cdot \frac{V_{REF}}{2^n} = 0,30517 \times 10^{-3}$$

$$E_T = E_{quant} + E_G + E_{ripple}$$

$$E_{quant} > E_G + E_{ripple}$$

$$E_{ripple} = \frac{V_{REF}}{2^n \cdot 4} = \frac{1}{4} \cdot V_{lsb} = 0,1525 \times 10^{-3}$$

$$0,1525 \times 10^{-3} + 0,30517 \times 10^{-3} = E_{total}$$

$$E_{ganancia} = \frac{1}{2^n \cdot 4} = 30,5175 \times 10^{-6}$$

= 30 ppm
6 partes por
millón

$$E_{total} = 3,488 \times 10^{-3}$$

$$E_{ganancia} = \frac{1}{4 \cdot \text{posos}} = \frac{1}{4 \cdot 6000} = 91 \text{ ppm}$$

? forma más optimista
se considera E_G

7) Del ejercicio 6, determinar los errores máximos admisibles de la tensión de referencia y la ganancia del amplif.

$$\text{Error de } \frac{1}{2} \cdot V_{LSB} = 0,076 \text{ mV}$$

$$E_{\text{Ripple}} = \frac{1}{4} \cdot V_{LSB} = 0,038 \text{ mV}$$

$$E_{\text{ganancia}} = \frac{1}{2^n \cdot 0.1} = 7,629 \times 10^6 = 7,6 \text{ PPM}$$

$$E_{\text{ganancia}} = \frac{1}{4 \text{ pasos}} = \frac{1}{4,20000} = 12,5 \times 10^6 \approx 12,5 \text{ PPM}$$

8) Se dispone de una señal de media cuya componente armónica más elevada es de 1KHz. Determinar tiempo máximo de conversión y tiempo máximo de muestras.

$$(t_{\text{muestreo}}) = t_{\text{sampling}}$$

$$t_{\text{sampling}} = t_{\text{muestreo}} = \frac{1}{2^n \cdot 2\pi \cdot F} \quad (\text{ts})$$

$$f_{\text{muestreo}} = S \cdot f_{\text{señal}} = 5 \times 1 \text{ KHz} = 5 \text{ KHz}$$

$$t_{\text{conversion}} = \frac{1}{f_{\text{muestreo}}} = \frac{1}{5 \text{ KHz}} = 200 \mu\text{seg} \rightarrow \text{zoom en o menor}$$

$$\text{Suponiendo } n=12 \text{ bits}$$

$$t_{\text{sampling}} = t_{\text{muestreo}} = t_s = \frac{1}{2^n \cdot 2\pi \cdot f_{\text{señal}}} \approx 39 \text{ nseg}$$

Ejercicio de Parcial:

* Una señal varía entre 3V y -3V, dada un ADC de 10 bits y 3V de Vref, calcular ganancia para máxima resolución y resolución obtenida en la señal.

$n = 10$ bits

$V_{REF} = 3V$

$2^n = 2^{10} = 1024$ Pasos

$V_{PP} = 6V$

$\text{Pasos} = \frac{V_{PP}}{V_{RES}} \rightarrow V_{RES} = \frac{V_{PP}}{\text{Pasos}} = \boxed{5,359 \text{ mV}}$

$G = \frac{V_{REF}}{V_{PP}} = \frac{3V}{6V} = 0,5$

$G = \frac{V_{RES}}{V_{RCs}} = \frac{5,359}{2^n} = \boxed{0,5}$

* Se desea medir una señal de tipo $x(t) = 3 \sin(1000t)$, encontrar cantidad de bit del ADC y ganancia para obtener una resolución exacta en la señal de 0,5 mV con una $V_{REF} = 3V$.

$V_{REF} = 3V$

$V_{RES} = 0,5 \text{ mV}$

$\text{Pasos} = \frac{V_{PP}}{V_{RES}} = \frac{6V}{0,5 \text{ mV}} = 12000 \text{ pasos}$

$V_{PP} = 6V$

$n = \frac{\ln(12000)}{\ln(2)} = 13,55 \approx \boxed{14 \text{ bits}} ; 2^n = 16384$

$G = \frac{V_{REF}}{2^n} = \boxed{0,3662}$

$G = \frac{V_{REF}}{V_{RES}} = \frac{3V}{0,5 \text{ mV}} = \frac{1}{2} \Rightarrow \boxed{0,5}$

* Se desea medir la temperatura de un horno que varía entre los 20°C y 270°C , con una resolución de $0,2^\circ\text{C}$. Calcular mínima cantidad de bits del ADC y error de ganancia.

$T_{PP} = 250^\circ\text{C}$

$\text{Pasos} = \frac{T_{PP}}{T_{RES}} = 1250 ; n = \frac{\ln(1250)}{\ln(2)} = 10,23 \approx \boxed{11 \text{ bits}}$

$T_{RES} = 0,2^\circ\text{C}$

$E_G = \frac{1}{2^n \cdot 4} = \boxed{122 \text{ ppm}}$

Assembler (2017) 2º Parcial

* if ($R_1 > R_2$)

$R_3 = R_4 + R_5;$

else

$R_3 = R_4 - R_5;$

sigue:

cmp r1, r2

bne else

add r3, r4, r5

b size

else:

sub r3, r4, r5

sigue:

* for ($i = 0, i < 3, i++$)

{ $R_3 = R_1 + R_2;$

}

sigue

mov r0, #0

for: cmp r0, #3

bge size

add r3, r1, r2

add r0, r0, #1

b for

sigue:

* do {

$R_3 = R_1 + R_2;$

$i = i + 1;$

} while ($i != 8$)

sigue

mov r0, #0

do: add r3, r1, r2

add r0, r0, #1

cmp r0, #8

bne do

sigue:

* while ($R_1 < R_2$)

{ $R_2 = R_2 - R_3;$

}

sigue

cmp r1, r2

bne sigue

Ejercicios de Assembler:

1) mov r1, #valor11
 mov r2, #valor12
 mov r3, #valor21
 mov r4, #valor22
 adds r5, r1, r3
 adc r5, r2, r4

2) mov r4, #disc
 mov r0, #0x01
 lsl r0, r4
 and r3, r0
 icmp r3, #0
 bne fin
 orr r6, r2 (1<<16)

Fin: =

3) cmp r1, #100 bhs falso cmp r1, #200 bls falso mov r0, #1 b fin falso: mov r0, #0 b fin	cmp r1, #100 bhi verdad cmp r2, #200 blo verdad mov r0, #1 b fin verdad: mov r0, #0 b fin	cmp r1, #10 beq verdad cmp r1, #15 beq verdad cmp r1, #20 beq verdad cmp r3, #20 bne falso mov r0, #1 b fin	cmp r1, #100 bhi verdad falso: mov r0, #0 b fin cmp r2, #15 bne falso verdad: cmp r2, #15 bne falso mov r0, #0 b fin
--	--	--	---

Fin: =

NOTA:

9) `mov r1, #num ; add r1, r1, r1, r1, lsl #3 → r1 = r1 + r1 + 3(r1)`
`mov r2, #10`
`mul r3, r2, r1`
`mov r1, r3`

5) `ldr r1, #valor`
`ldr r2, #valor`
`mov r3, #1`
`cmp r1, #0`
`bne salir`

otro: `mul r3, r2, r3`
`sub r1, r1, #1`
`bne otro`
`b salir`

salir: `=`
`=`

6) `mov r0, #0 accu`
`mov r1, #0`

size: `add r0, r1`
`-1`
`add r1, #1`
`cmp r1, #100`
`blo size`
`:`
`:`

7) `mov r1, #rect`
`mov r2, #0`

size: `ldr r2, [r1], #1`
`cmp r2, #'a'`
`subhs r2, r2, #'(a'-'A')`
`str r2, [r1], #1`
`cmp r1, #0`
`bne size`
`b salir`

salir: `=`
`=`

8)

```
    mov r0, #ROM
    mov r1, #vect
    mov r2, #0
    mov r3, #15
    s3ce ldrsh r1, [r0], #2
    add r2, r2, r1
    subs r3, r3, #1
    bnp r3, #0
    bne s3ce
    mov r0, r0, asr #4
    strsh r0, PCOM
    b salr
    salr:
```

9)

```
    mov r2, #Vectw
    mov r3, #Vectb
    mov r4, #20
    mov r1, #0
    ldrb r0, [r3], #1
    ldrb r0, [r2],
```

10)

Final : 24/07/2017

- 1) Señal senoidal. Datos: la señal y el error. Calcular n^* y tiempo de adquisición máximo para ese error.
- 2) Esquema de un conversor A/D Flash. Explicar cómo funciona.
- 3) Calcular el error de cuantización de un conversor. Si quiero reducir el error a la mitad, ¿duplico la cantidad de bits? Justificar.
- 4) Multiple Choice de conversor Doble Rampa. (No tengo el choice)
- 5) Assembler:

```
if (3 > h)
    g = g + h
else
    g = g - h
```
- 6) Código assembler para: negar los tres bits más significativos de la variable en R1 y guardarla en R2.
- 7) No tengo la consigna.
- 8) Sin usar MUL, multiplicar R1 por 35 y guardarlo en R2, sólo con dos operaciones/pasos.
- 9) Encontrar el num 42 en un vector de 250 elementos. Guardar la posición en donde se encuentra el 42. Si no está el numero, poner en R1 un "-1".

Respostas:

- 1) Supongamos:

- señal: $s \sin(\omega t)$ ($V_{pp} = 10V$)

- error: 0,5mV

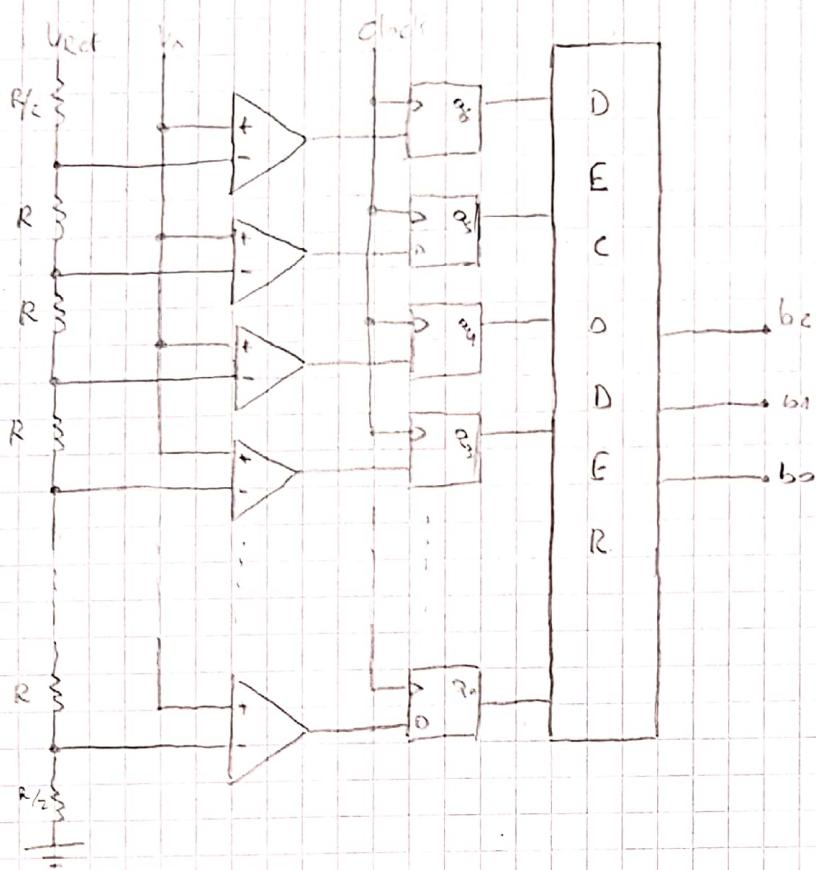
$$E_s = \frac{1}{2^n \cdot 2\pi f} ; \text{ suponiendo } f = 1 \text{ KHz} \Rightarrow L_s = \frac{1}{2^{10} \cdot 2\pi \cdot 1 \text{ KHz}} = 9,75 \mu \text{V}$$

$$E_q = 0,5 \text{ mV} = \frac{1}{2} \text{ LSB} \therefore \text{LSB} = 2 \cdot E_q = 1 \text{ mV}$$

$$\text{LSB} = \frac{\Delta V}{2^n} \therefore n = \ln\left(\frac{\Delta V}{\text{LSB}}\right) \xrightarrow{\Delta V = 10V} \ln(2) = 3,32 \approx \boxed{4 \text{ bits}}$$

NOTA

2) Conversor A/D flash (3 bits)



El conversor A/D flash es el más rápido de los conversores, ya que su tiempo de conversión es más corto. Esto se debe a que las comparaciones las realiza en "paralelo", como se puede observar en el diagrama. Los comparadores utilizan una tensión de referencia y una tensión analógica. Cuando la analógica sobrepasa a la de referencia entonces a la salida del comparador hay un "1". Esta etapa se podría decir que es la de "sample". La etapa de "hold" la realizan los flip-flops, refiriéndose la salida del comparador y llevándola a la decodificación, que realiza la decodificación necesaria para convertir 7 entradas en 3 salidas.

El número de entradas (comparadores) depende de la cantidad de salidas (bits) que se desee, mediante la fórmula: $2^n - 1$. En este caso de 3 bits: $2^3 - 1 = 7$ comparadores.

NOTA

3) Si usamos una analogía de una balanza romana, con distintos pesos de un valor determinado, se puede observar y deducir que el error de la balanza o del peso es de la mitad de la peso más pequeña.

Análogamente, la pesa más pequeña, en el ámbito digital, es el bit menos significativo (LSB). Por lo que el "error de cuantización" es analíticamente:

$$E_q = \pm \frac{1}{2} \text{ LSB}$$

Si se quiere un menor error de cuantización, se aumenta el número de bits. Esto se debe a que hay mayores niveles de comparación.

Ejemplo.

$$\text{Si } E_q = 1 \text{ mV} \quad E_q = \frac{1}{2} \text{ LSB} = \frac{\Delta V}{2^n} ; \Delta V = 5 \text{ V}$$

$$\therefore n = \frac{\ln(\frac{\Delta V}{\text{LSB}})}{\ln(2)} = 1,33 \equiv 2 \text{ bits}$$

Si queremos: $E_q = 0,5 \text{ mV}$

$$\therefore n = \frac{\ln(\frac{\Delta V}{\text{LSB}})}{\ln(2)} = 2,32 \equiv 3 \text{ bits}$$

No necesariamente se debe duplicar el número de bits para tener la mitad del error, pero sí se debe aumentar los bits.

S) mor r1, 3
 mor r2, 6
 cmp r1, r2
 bez suma
 sub r1, r1, r2
 b fin
 suma: add r1, r1, r2
 b fin
 fin =

NOTA

6)

mov cl, 3

rol r1, cl → (rota los 3 bits más significativos de r1 a la izq.)

mov cl, cl (nada > 65 bits)

rot r1, cl (vuelva los bits a la derecha)

mov r2, r1 (guarda r1 en r2)

3) add r2, r1, r1, LSL #5 ($r_2 = r_1 + 32r_1$)

add r3, r2, r1, #1 ($r_3 = r_2 + 2r_1 \rightarrow (r_3 = 33r_1 + 2r_1) \Rightarrow (r_3 = 35r_1)$)

(mal)

4) add r3, r1, r1, LSL #5 ($r_3 = r_1 + 32r_1$)

add r2, r3, r1, LSL #1 ($r_2 = 33r_1 + 2r_1 \rightarrow r_2 = 35r_1$)

Final: 23/11/2017

- 1) Determine el número de bits necesario para un conversor A/D simple rampa si se desea realizar la conversión de una señal sinusoidal de 1V de amplitud pico a pico y frecuencia de 1MHz siendo el máximo error tolerable de 2mV. Indique claramente todas las fórmulas necesarias o emplear. (5%)
- 2) Dado un sistema de adquisición de datos de 8 canales analógicos, que mediante un multiplexor se conectan a un filtro "antialiasing" cuya frecuencia de corte es de 90kHz, seguido por un "Sample/Hold" y un conversor A/D de 8 bits de aproximaciones sucesivas:
- Indicar el ancho de banda permitido para la señal de 90% de los canales (5%).
 - Indicar la frecuencia de muestras ^{mínima} para el "Sample/Hold." (5%)
 - Indicar el tiempo de conversión máximo para el conversor A/D. (5%)
- 3) Trace el diagrama de bloques del conversor A/D tipo "Flash". (5%)
- Indicar el tiempo de conversión. Justificar. (5%)
 - Indicar la cantidad de comparadores necesarios para un conversor de 8 bits (5%).
- 4) Dibuje detalladamente la trama de la comunicación serie del protocolo RS-232-C.
- Indique en el diagrama a qué se denomina el tiempo del bit "z". (5%)
 - ¿Cómo se define la velocidad de transmisión "v"? (5%)
 - El RS-232-C es: ① Simplex - ② Half-Duplex - ③ Full-Duplex. Justificar (5%)

5) Supongamos que tenemos el siguiente código C:

```
for (i=0; i<8; i++) {  
    a[i] = b[7-i];  
}
```

El índice "i" está declarado como un entero. Asumo que los arreglos "a" y "b" contienen solamente datos de tamaño byte y están ubicados en memoria RAM a partir de la 0x40000000. Traducirlo en instrucciones de lenguaje Assembly de ARM (15%).

6) Indique si las siguientes instrucciones usan los modos de direccionamiento pre- ó post-indirecto: (10%)

a - STR r0, [r4, #9].

c - LDRB r9, [r3, r2]!

b - LDR r3, [r10], #6

d - LDRSH r12, [r6]

7) ¿Cuál es la diferencia entre las instrucciones ROR (Rotate Right) y RRX (Rotate Right Extended)? (10%)

8) Escriba una subrutina que se invogue con "BL stamp" la cual debe ser capaz de comparar dos cadenas de caracteres y determinar si son idénticas. Los cadenas se hallan en memoria, están apuntadas por los registros r1 y r2 y finalizan con el carácter nulo. Este procedimiento retornará r3 con 0xFFFFFFFF en caso de igualdad ó 0x00000000 en el supuesto de que las cadenas sean distintas. (15%).

Respostas:

1) $V_{pp} = 1V$

$$F = 1\text{MHz}$$

$$\epsilon = 2mV$$

$$n = ?$$

$$\text{error} = \frac{\Delta V}{2V} = \frac{V \cdot 2\pi f L_s t_s}{2V} = \pi f L_s t_s$$

$$\text{error} = \pi f L_s t_s \rightarrow t_s = \frac{\text{error}}{\pi f} = 0,6366 \text{ nSes}$$

$$t_s = \frac{1}{2\pi f \cdot 2^n}$$

\downarrow

$$\hookrightarrow n = \frac{\ln(\frac{1}{\text{error}})}{\ln(2)} = 7,96 \cong [8 \text{ bits}]$$

Outra forma:

$$\log 2mV = \frac{1}{2} \text{ LSB} \therefore \text{LSB} = 4mV$$

$$n = \frac{\ln(\frac{1V}{4mV})}{\ln(2)} = \frac{\ln(\frac{1V}{4mV})}{\ln(2)} = 7,96 \cong [8 \text{ bits}]$$

2)

a.

$$A \cdot B = 40 \text{ KHz} = f_c$$

$$b - \text{Freq. de Muestreo} = f_s$$

$$f_s \geq 2f_c; \text{ donde } f_c = \text{freq. del Corte}$$

$$f_s \geq 80 \text{ KHz}$$

c - Tiempo de Conversión: t_c

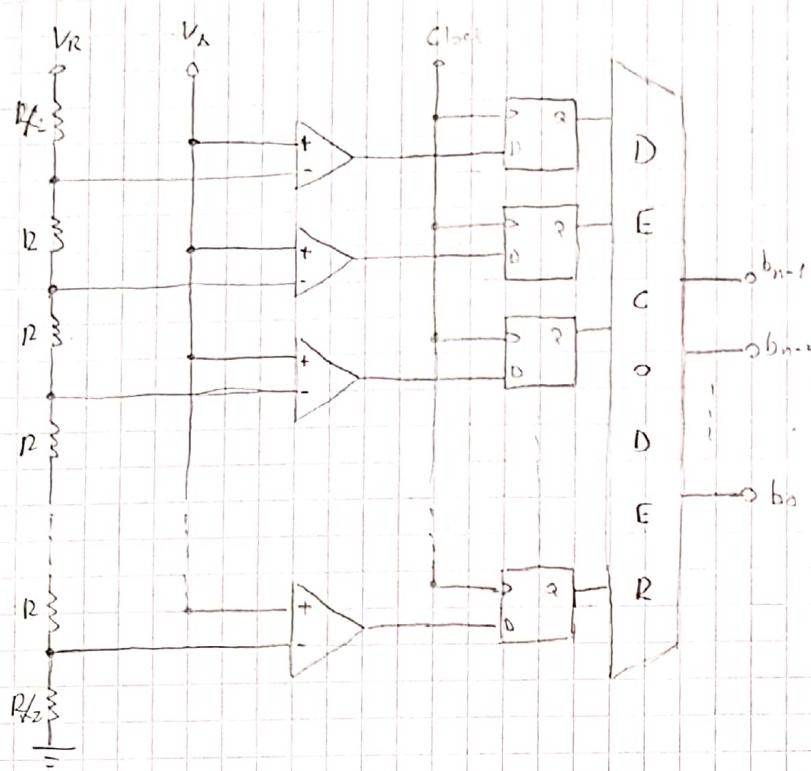
$$t_c = \frac{T_c}{N}; \text{ donde } T_c = \frac{1}{5f_c} = \frac{1}{200 \text{ KHz}}$$

$$N = \text{nº de canales.}$$

$$\therefore t_c = \frac{T_c}{3} = \frac{1}{40f_c} = \frac{1}{1.6 \text{ MHz}} = [625 \text{ nSes}]$$

NOTA

3)



a- El tiempo de conversión depende prácticamente del "Slew-Rate", es decir el tiempo que tarda el comparador en reaccionar a la salida ante cambios en la entrada. Por ende, se debe buscar A.O.'s con un slew-rate bajo, para una mayor rapidez de comparación.

b- La cantidad de comparadores necesarios está dada por la fórmula:

$$N = 2^n - 1 \quad ; \text{ donde } N = \text{Cantidad de comparadores}$$

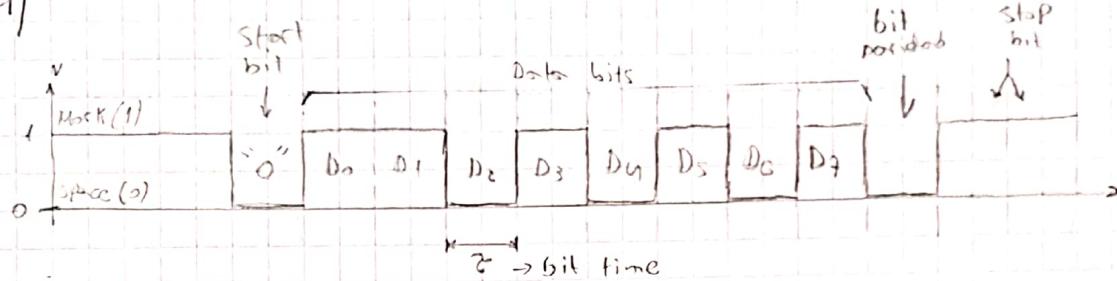
$n = \text{número de bits.}$

Para un convertidor de 8 bits:

$$N = 2^8 - 1 = \boxed{255 \text{ comparadores}}$$

NOTA

4)



Palabra: "AB" = 10101011

* tiempo de duración del bit "z" (bit time)

* bit de comienzo de transmisión o comunicación: establecido al de No transmisión. (Space)

* Estado de No transmisión ("1")

* bits de datos (Data Bits)

* bit de paridad (parity bit); vale "0" ó "1", dependiendo de que la paridad sea par o impar y de la cantidad de "1" en el Data Bits.

* bit de parada de transmisión o comunicación (Stop Bit). Pueden ser 1 ó 2 bits.

b.- Velocidad de Transmisión "v":

$$v = \frac{1}{z}; \text{ donde } z = \text{bit time}$$

$$v = \left[\frac{\text{bits}}{\text{s}} \right] = \text{bps}$$

5) mov r1, #0

for: cmp r1, #8
bge fin
ldrb r2, [#i] → (r2 = a[i])
subs r3, [#7, #r]
stcb r3, [r2]
adds r1, #1
b por

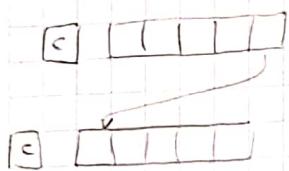
fin:

NOTA: \equiv

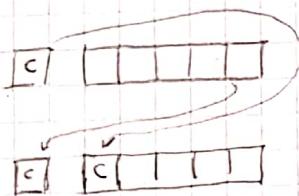
6)

- a - pre-incremento
- b - post-incremento
- c - pre-incremento
- d - post-incremento

7) ROR



ROR



Final 23/11/2017

$$1) V_{pp} = 1V \quad error = \frac{\Delta V}{2V} = \frac{V \cdot \pi \cdot f \cdot t_s}{2V}$$

$f = 1\text{MHz}$

$$\epsilon = 2mV$$

$$error = \pi \cdot f \cdot t_s \rightarrow t_s = \frac{error}{\pi \cdot f} = 0,6366 \text{ nSes}$$

$$n = ?$$

$$t_s = \frac{1}{2\pi f \cdot 2^n}$$

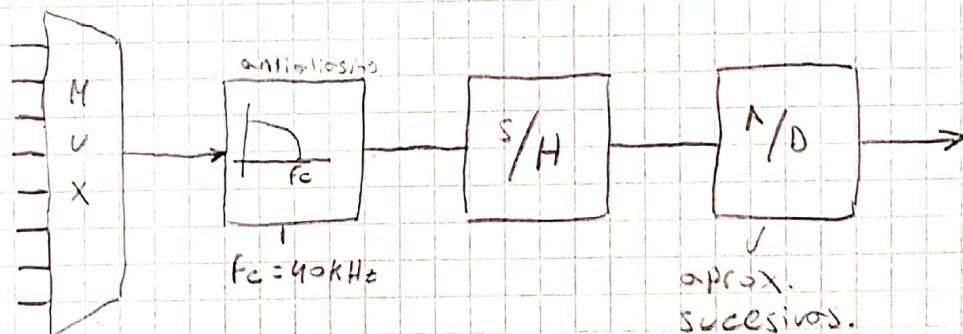
$$\rightarrow n = \frac{\ln\left(\frac{1}{2\pi f \cdot t_s}\right)}{\ln(2)} = 7,96 \cong [8 \text{ bits}]$$

otra forma

$$e_f = \frac{1}{2} \cdot LSB \rightarrow 2mV = \frac{1}{2} LSB \therefore \rightarrow LSB = 2 \cdot 2mV = [4mV = LSB]$$

$$n = \frac{\ln\left(\frac{1V}{LSB}\right)}{\ln(2)} = \frac{\ln\left(\frac{1V}{4mV}\right)}{\ln(2)} = 7,96 \cong [8 \text{ bits}]$$

2)



$$a) \Delta B = \frac{F_c}{N} = \frac{40\text{kHz}}{8} = [5\text{kHz}]$$

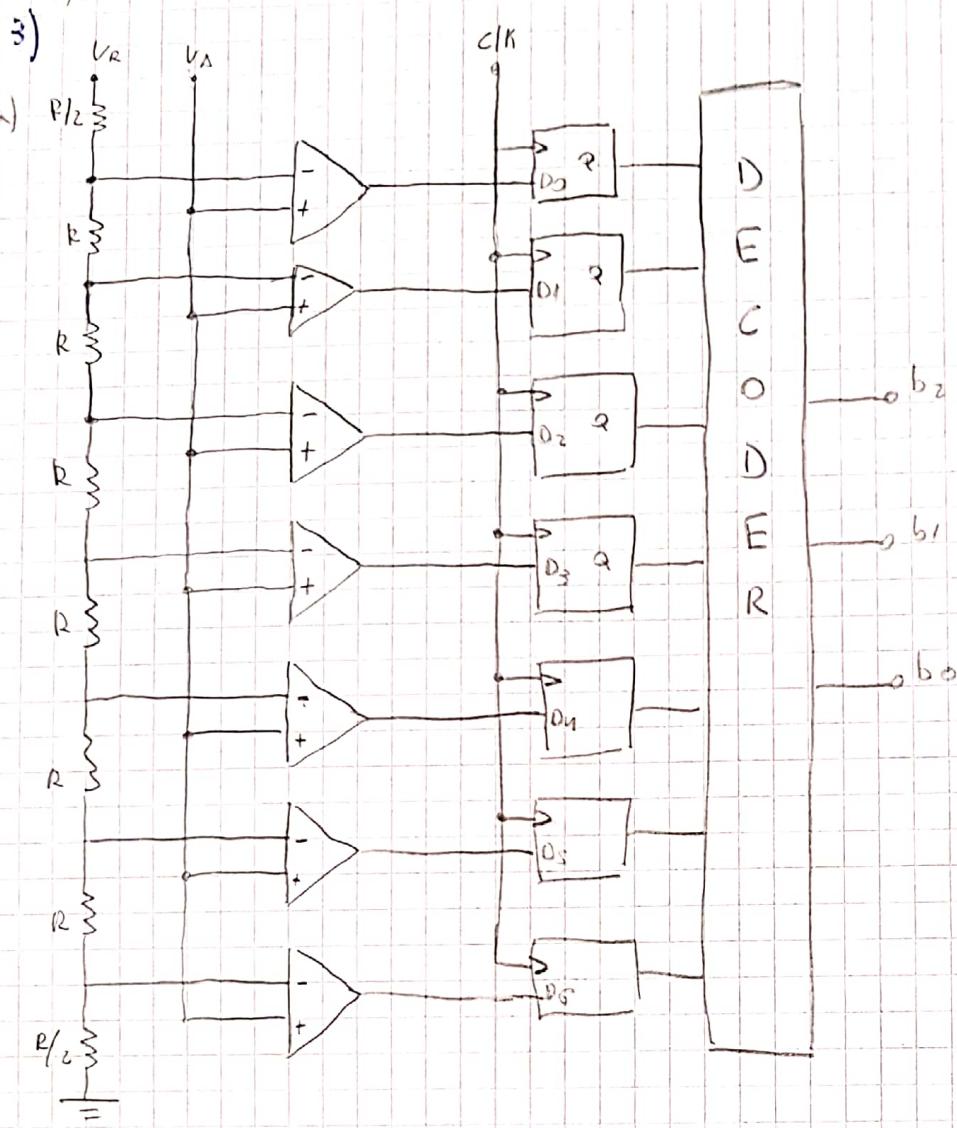
↑
cuantos

$$b) F_{min} = 2 \cdot N \cdot F_c = [640 \text{ kHz}]$$

$$c) t_{max} = \frac{1}{F_{min}} = [1,56 \text{ us}]$$

NOTA

A/D "flash"

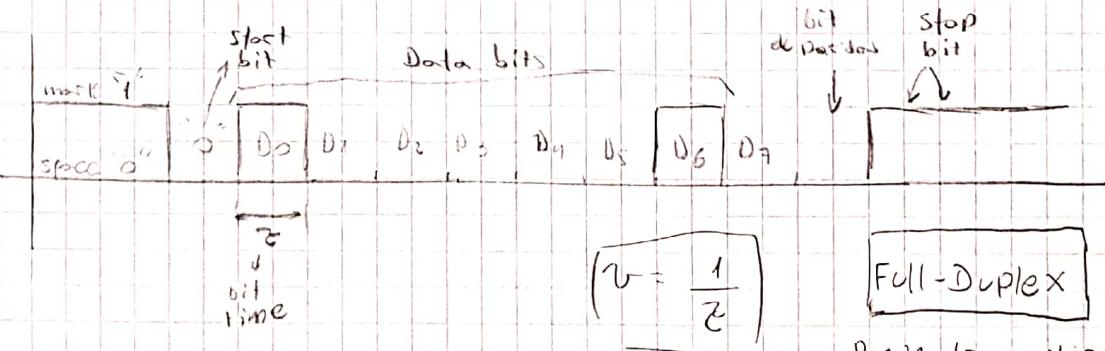


b) El tiempo de conversión es un ciclo de clock.

c) $2^n - 1$ comparaciones , $n = \text{bits}$

$$2^8 - 1 = 255 \text{ comparaciones.}$$

4)



Puede transmitir y recibir simultáneamente

NOTA

Final 23-11-17

①

$$V_{LSB} = \frac{V_{ref}}{2^n}$$

$$\text{error} = \frac{\Delta V}{2V} = \frac{V_{LSB} \cdot 2^n}{2V} = n_f \cdot \tau_s$$

$$e_f = \frac{1}{2} \cdot \tau_s = 2 \mu V$$

$$\text{error} = n_f \cdot \tau_s$$

$$LSB = 4 mV$$

$$\tau_s = \text{error} = \frac{2 \mu V}{n_f \cdot 1 MHz} = 0,6300 ns$$

$$V_{ref} = V_{LSB} \cdot 2^n$$

$$\ln\left(\frac{V_{ref}}{V_{LSB}}\right) = n_f \cdot \ln 2^n$$

$$\tau_s = \frac{1}{2\pi f_s} \cdot 2^n$$

$$\ln\left(\frac{V_{ref}}{V_{LSB}}\right) = n_f \cdot \ln 2$$

$$n_f = \ln\left(\frac{V_{ref}}{V_{LSB}}\right) / \ln(2)$$

$$n_f = \ln\left(\frac{V_{ref}}{V_{LSB}}\right) = \ln\left(\frac{12 mV}{4 mV}\right)$$

$$n_f = 3,96 = 8 \text{ bits}$$

$$n = 2^8 - 1 = 255 \text{ comodas}$$

②

El tiempo de conversión depende principalmente del seleno-rotor de los convertidores utilitarios (vea el apartado de decodificación (3cp))

$$2^n - 1 = 2^8 - 1 = 255 \text{ comodas}$$

③

$$AB = EC = \frac{40 kHz}{8} = 5000 Hz$$

Número

$$f_{in} = 2 \text{ Nyquist} f_s = 2 \cdot 8 \cdot 5000 Hz = 80 kHz$$

$$t_{max} = \frac{1}{f_{in}} = 1,5625 \mu s$$

Fürst 02-07-17

①

$$F7 = 20 \frac{\mu V}{9} \quad (\rightarrow \text{pionen})$$

$$\rho_{\text{Wires}} = \Delta_{\text{Pionen}} = \frac{50000}{0,1g} = 500000$$

$$V_{\text{ref}} = 20 \frac{\mu V}{9} \cdot 919 = 2 \mu V$$

$$\rho_{\text{Wires}} = 2^7$$

$$V_{LSB} = 2 \mu V = \frac{V_{\text{ref}}}{2}$$

$$\ln(\rho_{\text{Wires}}) = \ln 2^7$$

$$\eta = \ln \frac{V_{\text{ref}}}{2 \mu V} = \ln \frac{1}{2}$$

$$\eta = \ln(\rho_{\text{Wires}}) = \ln 6 = 16 \text{ bits}$$

⑤

$$t_s = \frac{1}{2\pi f_2} = \frac{1}{2,72 \cdot 10^6 \cdot 2^{10}} = 106 \text{ ps}$$

②

$$F7 = 0,012 \frac{\mu V}{9}$$

$$V_{\text{ref}} = 0,012 \frac{\mu V}{9} \cdot 919 = 1,2 \mu V$$

$$V_{\text{ref}} = 3 \mu V$$

$$V_{LSB} = \frac{V_{\text{ref}}}{2^7} = \frac{3 \mu V}{2^7} = 4,7 \mu V$$

$$\epsilon_{AD} = 0,19$$

$$G = \frac{V_{LSB}}{V_{\text{ref}}} = 61548$$

③

$$\epsilon = \epsilon_{AD} + \epsilon_{\text{Vref}} + \epsilon_G$$

$$\epsilon = \epsilon_{AD} - \epsilon_{\text{Vref}} = \epsilon_G$$

$$= \frac{V_{LSB}}{2} + \frac{1}{4} V_{LSB} + \frac{1}{2^{10}} \cdot 9$$

$$\epsilon_G = 0,05 \cdot V_{\text{ref}} - \frac{1}{2} V_{LSB} - \frac{1}{4} V_{LSB}$$

$$= \frac{V_{\text{ref}}}{2 \cdot 2^7} + \frac{V_{\text{ref}}}{4 \cdot 2^7} + \frac{1}{4 \cdot 2^7}$$

$$\epsilon_G = 1,9 \mu V$$

$$\epsilon = \epsilon_{AD} + 2 \epsilon_G$$

$$\epsilon_A = \epsilon_{AD} \frac{0,05}{100} \cdot V_{\text{ref}} = 0,05 \cdot 3 \mu V - \frac{V_{LSB}}{2} = 7385 \mu V$$

④

$$\text{tolerancia} = \frac{\sum G}{V_{ref}} = \frac{+39,5 \mu V}{3V}$$

$$\text{tolerancia} = 246 \text{ ppm}$$

⑤

$$t_{min} = \frac{1}{f_m} = \frac{1}{555}$$

$$f_s = \frac{\omega}{2\pi} = \frac{0,72 \times 10^6 \text{ rad/s}}{2\pi} = 114,592 \text{ kHz}$$

$$t_{min} = \frac{1}{5 \cdot 114,592 \text{ kHz}} = 1,745 \text{ ps}$$

⑥

sensibilidad en conversor de proximaciones sucesivas de 72 bits y 5V de V_{ref}

$$t_c = T_{clk} (N+1)$$

$$T_{clk} = \frac{t_c}{N+1} = \frac{1,745 \text{ ps}}{12+1} = 134,231 \text{ ns}$$

$$f_{clk} = \frac{1}{T_{clk}} = \frac{1}{134,23 \text{ ns}} = 7,45 \text{ MHz}$$

Fest 02-09-17

(6)

Método de interrupciones: Condiciona las interrupciones, para pasar de la etapa de solicitud a la de reconocimiento de su misma. Por que hay interrupciones encasillables y no encasillables, en denotadas también búsquedas (flags)

(7)

Una bifurcación o salto cambia el PC en la dirección donde se ubica la nueva instrucción a ejecutar, tiene dos tipos de salto: simple y branch en link BL, además como en cualquier instrucción el salto puede ser condicional o incondicional.

Tanto B como BL poseen un campo offset de 24 bits, al número en este campo es dividido entre 4 para que la instrucción sea ejecutada, con ayuda de un shift de 2 bits. B y BL tienen $\pm 32\text{ MB}$. Si queremos superar el barrera pero directamente en el PC: mov PC, 0x04000000

Incondicional: B directo

Condicional: BEQ directo

Task 23.02.11

①

$$\text{Add: } \Gamma_{q_1} \cap \Gamma_{q_2} \cap \Gamma_{q_3} = \emptyset$$

$$6 \cdot 19 = 114 + 18 \cdot 2^2 = 18(1+2^2) = 114$$

$$\Gamma_{3b} = \Gamma_{q_1} \cup \Gamma_{q_2} \cup \Gamma_{q_3}$$

$$6 \cdot 110 = 19 \cdot 2^3 + 19 = 19(7^2 + 1) = 113 = 19 \cdot 3^2 + 25 + 1$$

②

$$\rho_{\text{DIN}} = \frac{\Delta t}{\text{const}} = \frac{60^\circ\text{C} - (-10^\circ\text{C})}{0,01^\circ\text{C}} = 4000$$

$$2^n = \rho_{\text{DIN}}$$

$$n = \ln(\rho_{\text{DIN}}) / \ln(2) = 12,37 = 13 \text{ bits}$$

③

$$\text{Simplifying: } V_{\text{DIN}} = \Gamma_{\text{DT}} \cdot 10^3 = 20 \frac{\text{mV}}{^\circ\text{C}}, \quad 0,01^\circ\text{C} = 200 \text{ mV}$$

$$\Gamma_{\text{DT}} = 20 \frac{\text{mV}}{^\circ\text{C}}$$

$$G = \frac{V_{\text{DIN}}}{V_{\text{DIN}}} = \frac{V_{\text{ref}}}{V_{\text{DIN}}}$$

$$V_{\text{ref}} = 10 \text{ V}$$

$$G = \frac{10}{2} = 0,103,52$$

④

$$\epsilon = \epsilon_{\text{AID}} + \epsilon_B + \epsilon_{\text{Vref}} = 0,05 \text{ V} \cdot V_{\text{ref}} = 5 \times 10^{-4} \cdot V_{\text{ref}} = 5 \text{ mV}$$

$$\epsilon_{\text{AID}} = \frac{1}{2} V_{\text{ref}} = \frac{1}{2} \cdot \frac{V_{\text{ref}}}{2^n} = 6,03 \text{ mV}$$

$$\epsilon_B = \frac{\epsilon_{\text{AID}}}{2} = \frac{6,03}{2} \text{ mV}$$

$$\epsilon_B = \frac{\epsilon_{\text{AID}}}{2} ; \quad \epsilon_{\text{Vref}} = \frac{1}{4} V_{\text{ref}} = \frac{V_{\text{ref}}}{4 \cdot 2^n}$$

$$\epsilon_B = 2,96 \text{ mV}$$

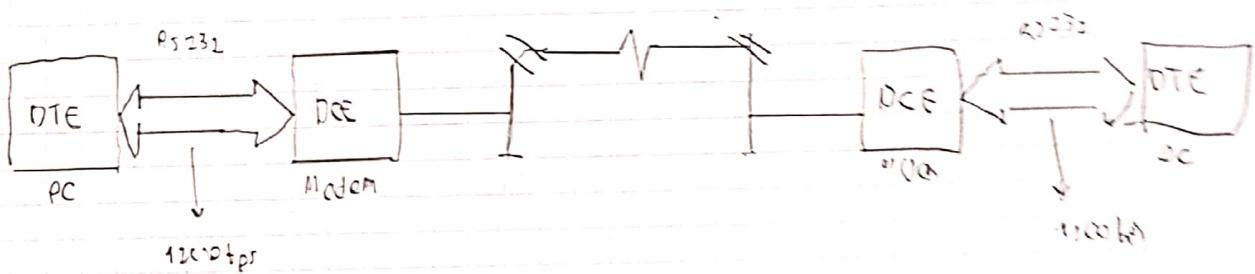
①

$$\text{Tolerancia} + \frac{\epsilon_0}{V_{REF}} = \frac{2,11}{100} \approx 2,11\% \text{ ppm}$$

Tolerancia: $\frac{2,11\%}{V_{REF}}$

②

Circuito de sincronismo de RS-232



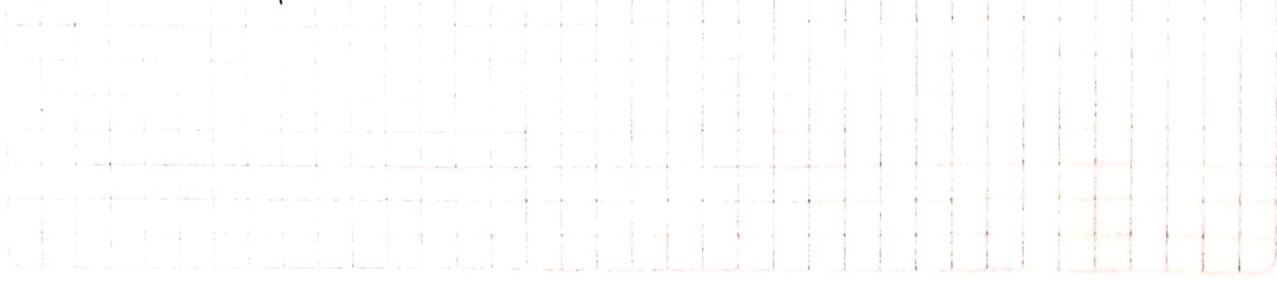
Es posible comunicar dos DTEs (dos PCs) con los DCE y la linea telefonica, basta con el uso de un cable null modem si este cruzado, se hace creer a cada PC que el otro es el receptor.

Con RS-232 o del tipo full-duplex (bidireccional y simultaneo) se necesitan 3 cables: Tx, Rx, Gnd

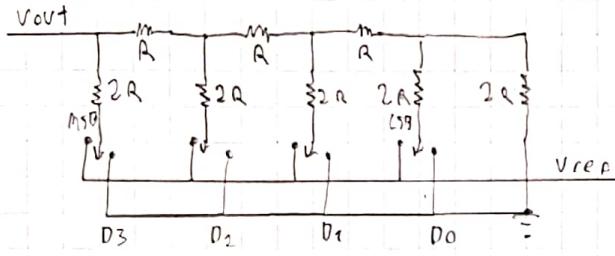
$$V = \frac{f_{clock}}{16 \times DL}$$

La frecuencia del clock te determina la velocidad de comunicacion, para lo cual primero debes utilizar DLL (registros de buchdivisores) para manejar el bit 7 del registro

ICP del chip V485T 16550



Conversor DAC R2R



$$\cancel{V_{out} = V_{ref} \left(\frac{D_0}{4} \right)}$$

$$V_{out} = V_{ref} \left(\frac{D_0}{2^4} + \frac{D_1}{2^3} + \frac{D_2}{2^2} + \frac{D_3}{2^1} \right)$$

Final 07/12/2017

1) $V_{ref} = 3,3V$

$$V_{USB} = \frac{V_{ref}}{2^n} = \frac{3,3V}{2^{10}} = 3,22mV$$

$n = 10$ bits

$$FT = 20 \frac{\mu V}{^{\circ}C}$$

$$\beta / R = 1k\Omega$$

$$V_{res} = 0,1^{\circ}C$$

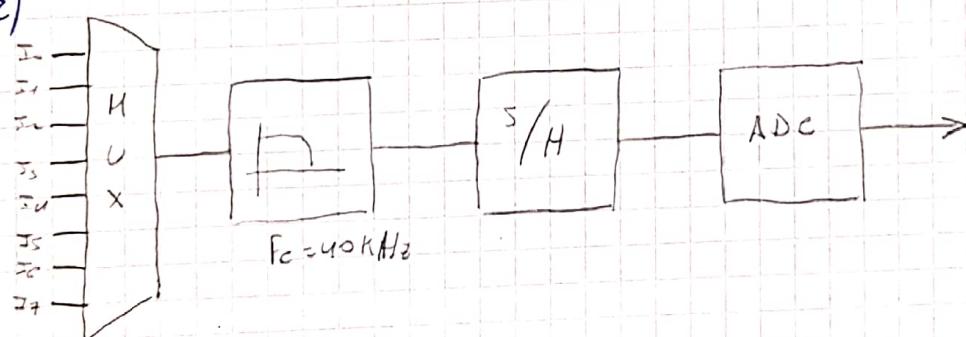
$$FT = 20 \frac{\mu V}{^{\circ}C} \cdot 1k\Omega = 20 \frac{mV}{^{\circ}C}$$

$G = ?$

$$V_{res} = 20 \frac{mV}{^{\circ}C} \cdot 0,1^{\circ}C = 2 mV$$

$$G = \frac{V_{GSB}}{V_{res}} = 1,61$$

2)



a) $AB = \frac{F_c}{N} = \frac{40kHz}{8} = 5kHz$

b) $f_{min} = 2 \cdot N \cdot F_c = 640 kHz$

c) $t_{max} = \frac{1}{f_{min}} = \frac{1}{640kHz} = 1,56 \mu s$

NOTA

3)

a) Flash \rightarrow $t_c = n \cdot T_{clk} \rightarrow$ Un ciclo de clock por bit.

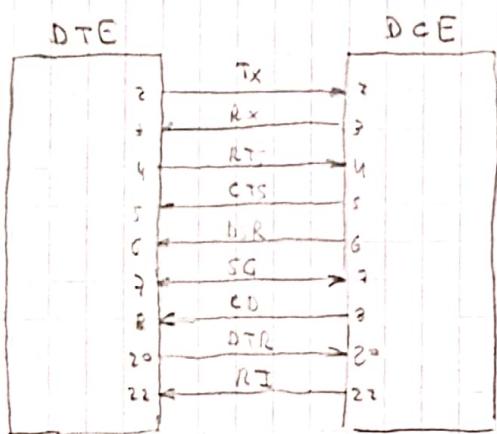
b) Async. Successors $\rightarrow t_c = T_{clk} (n+1)$

c) Double Rompón $\rightarrow t_c = 2^{n+1} \cdot T_{clk}$

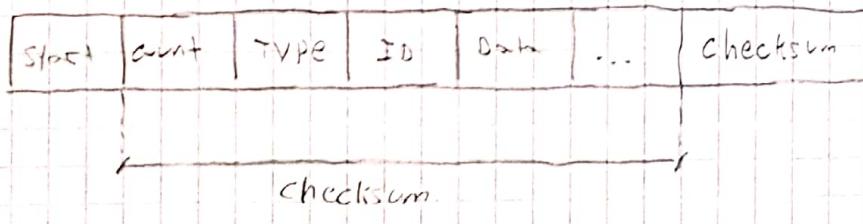
* Simple Rompón $\rightarrow t_c = (2^n + 1) \cdot T_{clk}$.

4)

a)



b)



El checksum es la suma total de todos los datos, excepto el byte de "start". Si la suma superan los dos dígitos (hexadecimales), se elimina el bit más significativo (MSB).

c). El bit 7 tiene como finalidad acceder a habilitar los registros del LATCH DIVISOR, tanto en nivel bajo como en alto (DCU y DCU)

Sirve generalmente para saber la velocidad de transmisión, por ejemplo 9600 bauds.

5) $\begin{array}{l} \text{mov } (t, \#x) \\ \text{mov } (r_2, \#y) \end{array}$ } inicialización $r_0 = \text{el Cn}$
 $\text{add } (r_3, r_1, r_2)$
 $\text{cmp } (r_3, \#0)$
 $\text{movlo } r_0, \#0$
 $\text{movhs } r_0, \#1$

6) LDRH $r_{11}, [r_0]$

r_{11} antes $\rightarrow 0x12345678$

r_{11} despues $\rightarrow \boxed{0x0000FFEE}$

7). STR $r_0, [r_1, \#12]$

o Pre-indexado

8)

Reposto final

Final 22/02/2017

$$1) n = 10 \text{ bits}$$

$$V_{ref} = 3,3 \text{ V}$$

$$\text{Transductor} = 20 \mu\text{A}/\text{C}$$

$$\text{a) } V_{LSB} = \frac{V_{REF}}{2^n} = \frac{3,3 \text{ V}}{2^{10}} = 3,22 \text{ mV}$$

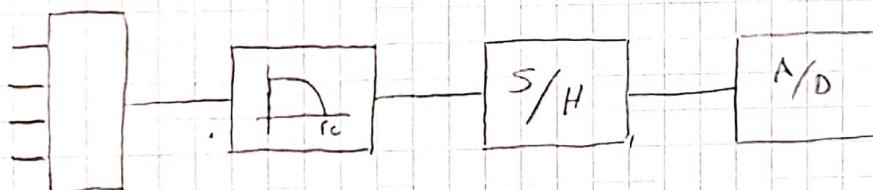
$$f_T = 20 \mu\text{A}/\text{C} \quad \text{con } R = 1 \text{ k}\Omega$$

$$20 \mu\text{A}/\text{C} \cdot 1 \text{ k}\Omega = 20 \frac{\text{mV}}{\text{C}}$$

$$V_{RES} = 20 \frac{\text{mV}}{\text{C}} \cdot 0,1\% = 2 \text{ mV}$$

$$\therefore G = \frac{V_{LSB}}{V_{RES}} = \frac{3,22 \text{ mV}}{2 \text{ mV}} = 1,6114$$

4)



$$\text{a) } f_B = 100 \text{ KHz} \rightarrow \frac{100 \text{ KHz}}{4} = 25 \text{ KHz}$$

$$\text{b) } f_{\text{min meas}} = 2 \cdot f_{\text{sample}} = 2 \cdot 100 \text{ KHz} = 200 \text{ KHz} \quad 2 \cdot 4 \cdot 200 \text{ KHz} = 16 \text{ MHz}$$

$$\text{c) } t_{\text{conv. máx}} = T$$

$$\therefore t_{\text{conv. máx}} = \frac{1}{16 \text{ MHz}} = 625 \text{ nseg}$$

3)

$n \cdot T_c \rightarrow 1$ ciclo de clock por cada bit

a) Tipos flash: El tiempo de conversión de conversor A/D tipo flash está determinado por el slew-rate del A/D, que es el tiempo que demora en reaccionar su salida ante cambios en la entrada.

b) Aproximaciones Sucesivas: el tiempo de conversión es:

$$T_c = n + 2 ; \text{ donde } n = \text{nº de bits.}$$

c) Doble Rampa: el tiempo de conversión de este tipo de conversor es:

$$2^{n+1} \cdot T_{clk} ; \text{ donde } T_{clk} = \text{Ciclos de clock} ; n = \text{nº de bits.}$$

4)

a) Falso. El rango de entrada total tiene relación con el nº de bits puestos que con dicho rango se puede calcular las cuentas para luego calcular el nº de bits. Esto también se usa para el cálculo de V_{LSB}

$$V_{LSB} = \frac{\Delta V}{2^n}$$

Versadero.

b- Verdadero. Este rango se define como error de cuantización $\epsilon_q = \pm \frac{1}{2} LSB$

c. Falso.

$$FS = 10V ; n = 8$$

$$2^8 = 256$$

$$\frac{FS}{V_{RES}} = 256 \therefore V_{RES} = \frac{FS}{256} = 39,06 mV \approx 39,1 mV$$

$$39,06 mV = 39062,5 \mu V = 39062,5 ppm \neq 3906 ppm$$

para FS :

$$10V - 100\%$$

$$39,06 mV - x = 10,39\% \cdot FS$$

9)

NOR:		R3
R1	R2	
0	0	1
0	1	0
1	0	0
1	1	0

en Assembler:

$$\text{orr } r3, r1, r2 \rightarrow r3 = r1 \text{ or } r2$$

$$\text{mnw } r0, r3 \rightarrow r0 = \overline{r3}$$

6) $g = r1; h \rightarrow r2$ if ($g \geq h$)

$$g = g + h;$$

else

$$g = g - h;$$

$\text{cmp } r1, r2$
 $\text{addge } r1, r1, r2$
 $\text{subs } r1, r1, r2$

7) a- $\text{STR } r0, [r1, \#12]$

b- pre-indexado.

8) $r0 \rightarrow \text{contador}$ $\text{mov } r0, \#0$ $\text{otra: ldr } r2, [r1], \#4$ $\text{cmp } r2, \#1$ $\text{addeq } r0, \#1$ $b \text{ otra}$

6. mvi 7/12/2017: Fotó en PC.

1) $n = 10$

$$V_{ISB} = \frac{V_{REF}}{2^n} = \frac{3.3V}{2^{10}} = 3.22 \text{ mV}$$

$$V_{REF} = 3.3V$$

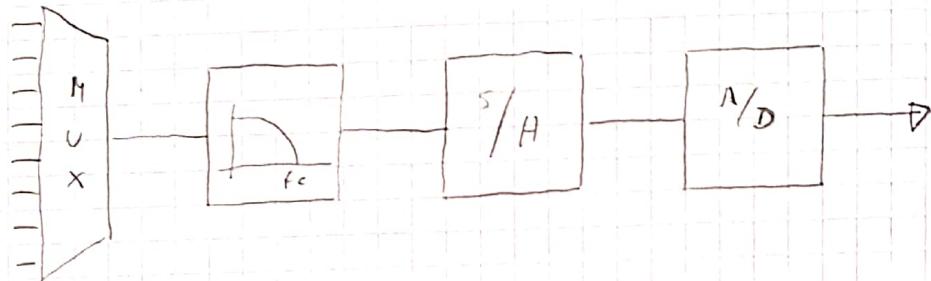
$$\Delta = 30^\circ C$$

$$R_{CS} = 21^\circ C$$

$$20 \frac{\mu A}{^\circ C} \cdot 1 \text{ K}\Omega = 20 \frac{\text{mV}}{^\circ C} \therefore V_{RES} = 20 \frac{\text{mV}}{^\circ C} \cdot 21^\circ C = 12 \text{ mV}$$

$$G = \frac{V_{ISB}}{V_{RES}} = \frac{3.22 \text{ mV}}{2 \text{ mV}} = 1.61$$

2)



a) $A_B = 140 \text{ kHz}$ \rightarrow 4 canales $\therefore A_B = \frac{f}{n \text{ canales}} = \frac{40 \text{ kHz}}{8} = 5 \text{ kHz}$

b) $f_{m\min} = 2 \cdot f_{m\max \text{ serial}} = 2 \cdot 40 \text{ kHz} = 80 \text{ kHz}$ $\rightarrow 2 f_{m\max} = 2 \cdot n \text{ canales} \cdot f = 2 \cdot 8 \cdot 40 \text{ kHz} = 640 \text{ kHz}$

c) $t_{s\max} = \frac{1}{f_{m\min}} = \frac{1}{80 \text{ kHz}} = 1.25 \mu \text{s}$

3) a) tipo flash = $n \cdot T_{clk} \rightarrow$ un ciclo de clock por cada bit

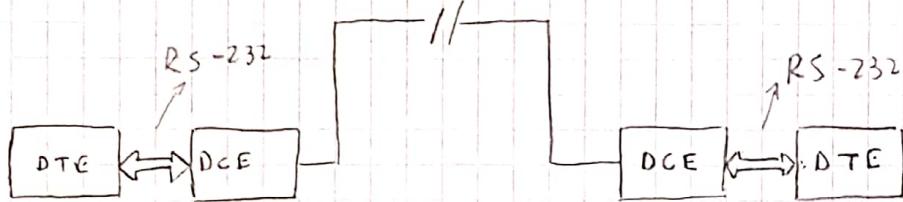
b) Apox. sucesivas = $T_c = n + z$; donde n es el nº de bits.

c) Doble Rampa = $2^{n+1} \cdot T_c$; donde n es el nº de bits.

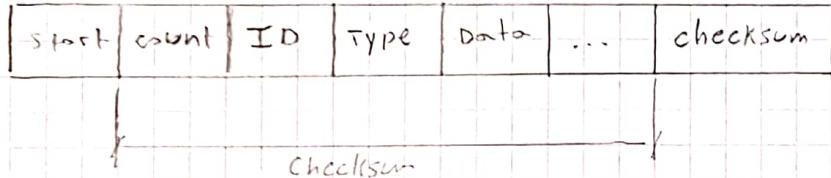
DTE = Data Terminal Equipment
DCE = " communication Equipment

i)

a)



b)



El checksum es la suma de todos los bytes, excepto el byte de 'start'. Si el resultado supera los dos dígitos, se elimina el bit más significativo.

c) El bit 7 tiene como finalidad acceder o habilitar los registros del WATCH DIVISOR, tanto en el nivel bajo, como en el alto (DLL y DLM). Este divisor se necesita tanto USB como MSB, para obtener la velocidad deseada, por ejemplo 9600 bps.

5) if ($x+y < 0$) {
 return 0;
} else
 return 1;
}

$x = r1 : y = r2$
 mov r0, #0
 add r0, r1, r2
 movlo r0, H>
 movhs r0, #1

6) LDRH r11, [r0]
 r0 \rightarrow $0x\text{EE}$
 $0x\text{FF}$

FFF después de la carga

$0x\text{0000}\text{FFEE}$

7) a) STR r0,[r1,#12]; b) pre-indexado.

8)

Final 23/11/17

1) $n = ?$

$$f = 1 \text{ MHz}$$

$$V_{PP} = 1 \text{ V}$$

$$\epsilon = 2 \text{ mV}$$

$$e = \frac{1V}{2V} = \pi \cdot F \cdot t_s$$

$$t_s = \frac{1}{2\pi F 2^n}$$

$$t_s = \frac{\epsilon}{\pi f} = \frac{2mV}{\pi \cdot 1 \text{ MHz}} = 0,6366 \text{ nSes}$$

$$\therefore n = \frac{\ln(\frac{1}{2\pi f t_s})}{\ln(2)} = 7,96 \approx \boxed{8 \text{ bits}}$$

2)

a) $AB = \frac{f_c}{N}$, $f_c = \text{frec. de corte}$; $N = \text{nº de canales}$

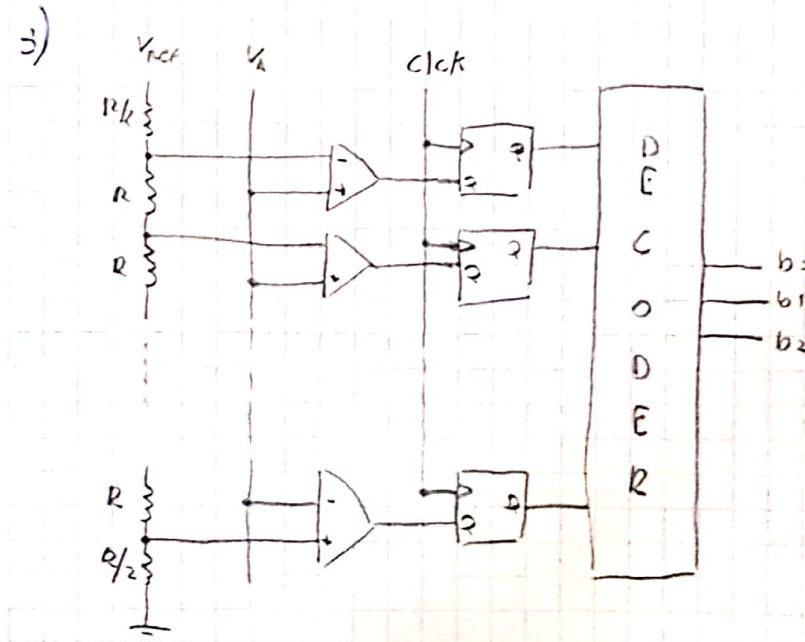
$$AB = \frac{f_c}{N} = \frac{40 \text{ kHz}}{8} = \boxed{5 \text{ kHz}}$$

b) $f_{m_{min}} = 2 \cdot N \cdot f_c = 2 \cdot 8 \cdot 40 \text{ kHz} = \boxed{640 \text{ kHz}}$

c)

$$t_{cmax} = \frac{1}{f_{m_{max}}} = \frac{1}{640 \text{ kHz}} = \boxed{1,5625 \text{ uses}}$$

d)



b) El tiempo de conversión es un pulso de clock, puesto que este convierte todos en paralelo. Hay que tener en cuenta que el tiempo de conversión va a depender del slew-rate de los A.O.s, que se define como el tiempo que tarda en reaccionar la salida, ante cambios en la entrada.

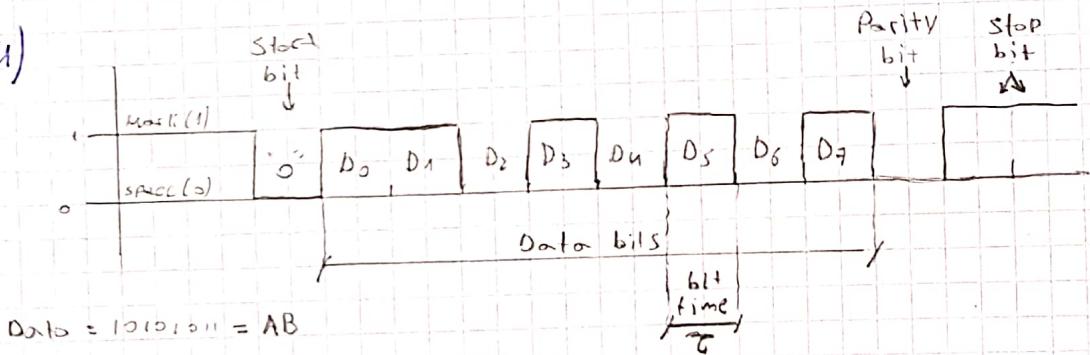
c) Si se tiene un inversor tipo flash de 3 bits, se necesitan:

$2^n - 1$ comparadores

Para este caso:

$$2^3 - 1 = 256 - 1 = \boxed{255 \text{ comparadores}}$$

ii)



b) Tiempo de velocidad $V = \frac{1}{T}$ [bps], donde $T = \text{bit time}$

c) El protocolo RS-232-C es full-Duplex, porque envía y recibe datos simultáneamente.

$Rx \rightleftharpoons Tx$

5)

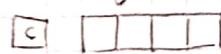
a. post

b. post

c. pre

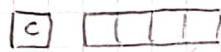
d. pre.

7) ROR:



Rotación hacia la
derecha

RRX:



Rotación hacia la derecha,
incluyendo el acarreo

8)

b) strcmp

Push {r1 - r3}

atras: ldrb r4, [r1], #1
cmp r4, #0
beq salir
ldr r5, [r2], #1
cmp r5, #0
beq salir
cmp r4, r5
beq igual
mvn r6, #0

igual: mov r6, #0
b atras

salir: str r6, [r3]
pop {r1 - r3}
mov pc, lr