

Final 8/3/2018

1) Rango:  $0 \sim 10V$        $3,922V = V_{in} \Rightarrow x$

a)

$$\frac{10}{2^4} = 3V$$

$$\frac{10}{2^2} = 2,5V$$

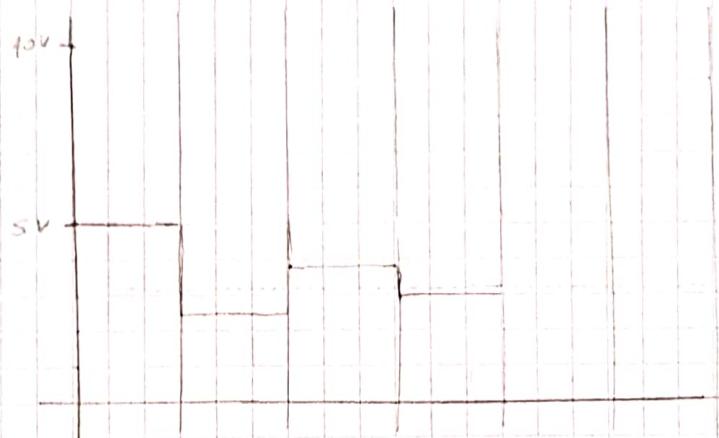
$$\frac{10}{2^3} = 1,25V$$

$$\frac{10}{2^4} = 0,625$$

$$\frac{10}{2^5} = 0,3125$$

$$\frac{10}{2^6} = 0,15625$$

$$\frac{10}{2^7} =$$



1º CLK  $x > 2 \rightarrow N \oplus$

2º CLK  $x > 2,5 \rightarrow S \oplus$

3º CLK  $x > 2,5 + 1,25 \rightarrow S \oplus$

4º CLK  $x > 2,5 + 1,25 + 0,625 \rightarrow N \oplus$

5º CLK  $x > 2,5 + 1,25 + 0,3125 \rightarrow N \oplus$

6º CLK  $x > 2,5 + 1,25 + 0,15625 \rightarrow S \oplus \rightarrow \boxed{3,90625V}$

b)  $15\text{KHz} = f_{clk} \therefore T_{clk} = \frac{1}{f_{clk}} = 66,66 \mu\text{seg}$

$$t_c = (n+1) \cdot T_{clk} = (8+1) \cdot 66,66 \mu\text{seg} = 600 \mu\text{seg}$$

como es sólo el módulo del HOLD:  $\frac{t_c}{2} = \frac{600 \mu\text{seg}}{2} = 300 \mu\text{seg}$

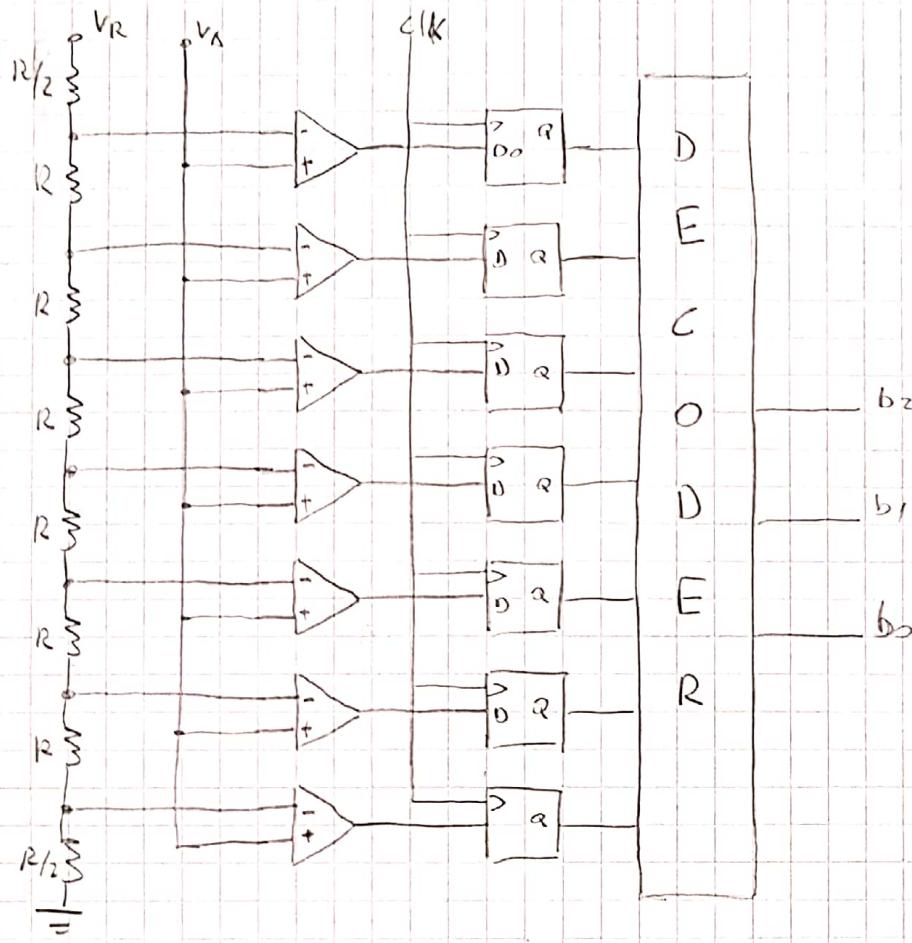
3)  $LSB = \frac{1V}{2^n} = \frac{1V}{2^8} = 3,9 \text{mV}$

$$eq = \frac{1}{2} LSB = \boxed{1,95 \text{mV}}$$

a) Ninguna de las anteriores.

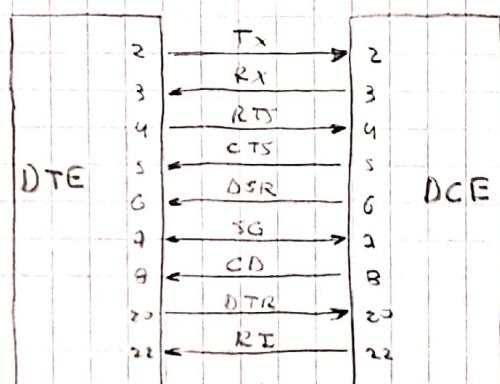
2)

3 bits  $\therefore 2^n - 1 = 2^3 - 1 = 7$  comparaciones.



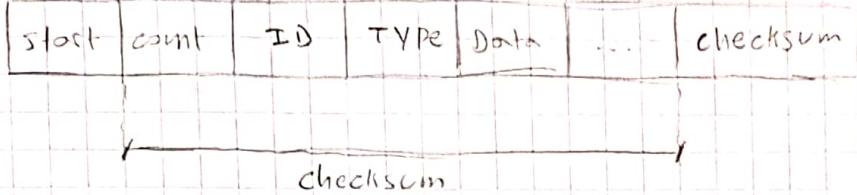
4) El conversor A/D doble rampa:

- Se utiliza para instrumentación.
- tiene un tiempo de conversión lento.

5)  
a)

NOTA

b)



El checksum es el total de 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 LATCH DIVISOR, tanto en el nivel bajo, como en el alto (DLL y DDU). Este divisor se necesita, tanto LSB como MSB, para obtener la velocidad deseada, por ejemplo 9600 bps.

8) EOR R0, R0, R1

EOR R1, R0, R1

EOR R0, R0, R1

EOR R2, R2, R1

FOR R1, R0, R1

    1

ETJR R0, R0, R1

    1

R0 = 0 - 1 - 1

R1 = 1 0 0

R2 R1

0 0 1 0

0 1 1

0 0 1

1 1 0

7) ldc r2, [r0]

add r3, r2, LSL#3 → r2 = 8r2

add r3, #7

str r3, [r1]

NOTA

Final 24/05/2018

- 1) Los principales características de la arq. ARM son:
- a) verdaderas.
  - b) verdaderas
  - c) verdaderos (2 fuentes y un destino)
  - d) falso  $\rightarrow$  la ejecución de cada instrucción puede ser o no condicional.
  - e) verdaderas

f) Verdadero, por ejemplo add R0, R2, R1 LSL #5

3) Falso  $\rightarrow$  solo se tiene el nombre de los registros (0-R15)

2) Los otros RISC no comparten instrucciones de 3 direcciones.  
(inciso a del punto 1).

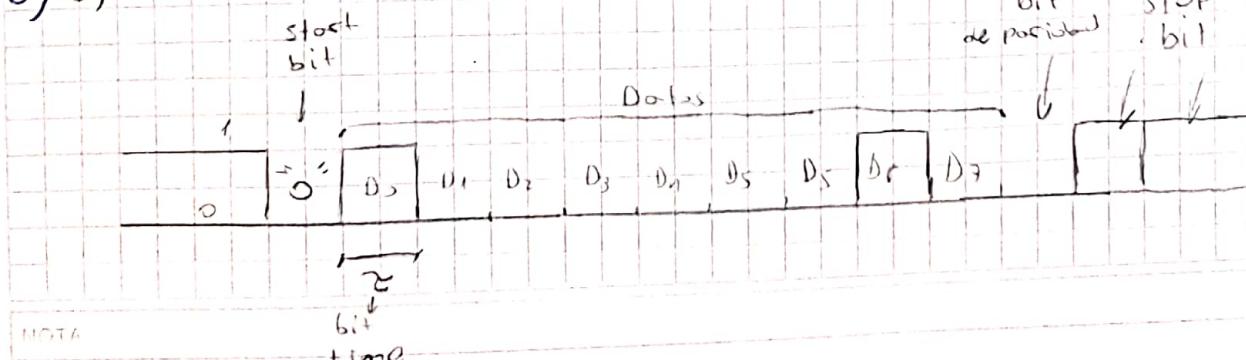
3) Todo: arquitectura ADC usa, al menos, un bloque de comparadores de algún tipo. (opción b).

4) Un conversor D/A R-2R necesita:

$2^N$  resistencias, (opción c).

(\* Un DAC Kelvin requiere  $2^N$  resistencias.)

5) a)

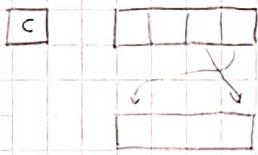


b)  $v = \frac{l}{T}$ ;  $v$  = velocidad de transmisión.  
 $T$  = tiempo de bit.

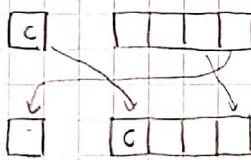
c) El protocolo es Full-Duplex porque el RS232-C permite enviar y transmitir datos simultáneamente.

- 8) a - pre-invertido.  
 b - post-invertido.  
 c - pre-invertido.  
 d - pre-invertido.

9) La diferencia entre ROR y RRX, es que el RRX tiene en cuenta el carry en el desplazamiento.



ROR



RRX

7)

```

    mov r0, #0
    mov r3, #7
for: cmp r0, #8
      bhs fin

```

ldr r4, [r1, r0]

sub r4, r3, r0

str r4, [r1, r0]

add r0, r0, #1

b for

f.n:   

r0 = i

r1 = a

r2 = b

r3 = f

NOTA

Final 11/10/2018

1)

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

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

$$V_{REF} = 3,3V$$

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

$$FT_s = 20 \mu A/\text{oC}$$

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

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

$$\therefore G = \frac{V_{LSB}}{V_{RES}} = \frac{3,22 \text{ mV}}{2 \text{ mV}} = 1,6111$$

2)

$$\text{a) } \frac{10}{2^1} = 5V$$

$$X = 3,922V$$

$$\frac{10}{2^2} = 2,5V$$

1° CLK → X > 5V → No → Desacto 5V

2° CLK → X > 2,5V → Si → Refresgo 2,5V

3° CLK → X > (2,5V + 1,25V) → Si → Refresh 1,25V

4° CLK → X > (2,5V + 1,25V + 0,625V) → No → Desacto 0,625V

5° CLK → X > (2,5V + 1,25V + 0,3125V) → No → Desacto 0,3125V

6° CLK → X > (2,5V + 1,25V + 0,15625V) → Si

$$6^{\text{th}} \text{ CLK} = 2,5V + 1,25V + 0,15625V = 3,90625V$$

b) P/approx. successivas:

$$t_c = T_{CLK} \cdot (n+1) = \frac{1}{15 \text{ kHz}} \cdot (8+1) = 600 \mu \text{A}$$

$$\frac{600 \mu \text{A}}{2} = 300 \mu \text{A} = t_{H-LD}$$

3)

a)  $N = 3$

$$f_{\text{Smin}} = 2 \cdot N \cdot f_s = 2 \cdot 3 \cdot 10 \text{ MHz} = 60 \text{ MHz}$$

$$f_c = 10 \text{ MHz}$$

mejorada por segundo = frec. de muestra =  $f_s$

b)

$$t_{\text{cmax}} = \frac{1}{f_{\text{min}}} = \frac{1}{60 \text{ MHz}} = 6,67 \text{ nsec}$$

4)

a) Flash  $\rightarrow t_c = \text{Tclk}$

b) Aprox. Sucessivas  $\rightarrow t_c = \text{Tclk}(n+1)$

c) Doble Rampa  $\rightarrow t_c = 2^{n+1} \cdot \text{Tclk}$

\* Simple Rampa  $\rightarrow t_c = (2^n + 1) \cdot \text{Tclk}$

5) LDR Rd, [Rn, Rm]!  $\equiv$  LDR Rd, [Rn] Verändern.  
ADD Rn, Rn, Rm

6)  $r_3 = 0x8000$

a) STR r6, [r3, #12]  $\rightarrow r3 = r3 + 4096$

b) STC r3, [r7], #4  $\rightarrow r3 = r3 - 4$

c) LDRH r5, [r3], #8  $\rightarrow r3 = r3 + 10240$   $r3 = r3 + 0x10000$

d) LDR r2, [r3, #16]  $\rightarrow r3 = r3 + 0x10000$

7)

a) BIC r2, #0xFFFF

b) AND r2, #0x0000

8)

mov r2, #7 0111

ldr r3, [r0] 1010

bic r3, #0x07 1010.1000:

add r3, r3, r2

str r3, [r1]

9)

Final 22/02/2018

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

$n = 10$  bits

$$FT = 20 \mu A/\text{oc}$$

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

$$\%R = 1k\Omega$$

$$R_{ES} = 0,1^{\circ}\text{C}$$

$$V_{RCS} = 20 \frac{\mu A}{\text{oc}} \cdot 1k\Omega \cdot 0,1\% = 2 \text{ mV}$$

$$G = \frac{V_{LSB}}{V_{RCS}} = \boxed{1,61}$$

2)

a)  $\Delta B = \frac{F_C}{N} = \frac{100 \text{ kHz}}{4} = \boxed{25 \text{ kHz}}$

b)  $f_{S,\min} = 2 N \cdot f_C = 2 \cdot 4 \cdot 100 \text{ kHz} = \boxed{800 \text{ kHz}}$

c)  $t_{C,\max} = \frac{1}{f_{\min}} = \boxed{1,25 \text{ useg}}$

3)

a) Flash  $\rightarrow t_C = n \cdot T_{CK} \rightarrow$   $n$  cycles de clock por bit

b) Approx. successivas  $\rightarrow t_C = T_{CK} (n+1)$

c) Double rampa  $\rightarrow t_C = 2^{n+1} \cdot T_{CK}$

\* Simple rampa  $\rightarrow t_C = (2^n + 1) \cdot T_{CK}$

9)

a)  $V_{LSB} = \frac{FS}{2^n}$ , FS es independiente de n. [Verdadero]

b)

$E_g = \pm \frac{1}{2} LSB$  : Verdadero.

c)  $n=8$ 

$FS = 10V$

$$V_{LSB} = \frac{FS}{2^n} = \frac{10V}{2^8} = 39,06 \text{ mV} \cong 39,1 \text{ mV}$$

$$39,06 \text{ mV} = 39062 \text{ ppm FS} \neq 3906 \text{ ppm FS} \times \therefore \text{Falso}$$

$$10 \cdot 0,39\% = 39 \text{ mV} \vee$$

5) AND r0, r1, r2

0110

1100

0000

0000

MVN r0, r0

0110

0110

1001000

0110111

add32 r0, r0, r1

Subt r0, r0, r1

7) str r0, [r1, #12] b) pre-indexados

8) mov r0, #0 → continue

for {dr r2, [r1], #4}

cmp r2, #1

add32 r0, #1

b for

NOTA

Final 26/7/2013

1)

$$F_T = 20 \frac{\mu V}{K\Omega}$$

$$V_{LSB} = \frac{V_{REF}}{2^n} = \frac{5V}{2^{13}} = 610,35 \mu V$$

$$R_{ES} = 1 \Omega = 0,001 K\Omega$$

$$V_{REF} = 5V$$

$$V_{RES} = 20 \frac{\mu V}{K\Omega} \cdot 0,001 K\Omega = 20 \mu V$$

$$n = 13 \text{ bits}$$

$$G = \frac{V_{LSB}}{V_{RES}} = 30517$$

2)  $V_{REF} = 3V$

$$n = 8 \text{ bits}$$

$$E_{Ripple} = \frac{1}{4} V_{LSB} = \frac{V_{REF}}{4 \cdot 2^8} = \frac{3V}{4 \cdot 2^8} = \boxed{2,929 mV}$$

$$E_G = \frac{1}{2^n \cdot 4} = \boxed{976,56 \text{ PPM}}$$

3)

4)

$$2) \quad N = 9600$$

$$N = \frac{1}{T} \quad \therefore T = \frac{1}{N} = 104,16 \text{ ms}$$

b)

$$\text{baudios} = \text{bits} + \text{binitis}$$

$$\text{binitis} = \text{bit de orange} + \text{bit de parada} = 2$$

.

$$\text{bits} = \text{bps} = \text{baudios} - \text{binitis} = 9600 - 2 = 9598 \text{ bps}$$

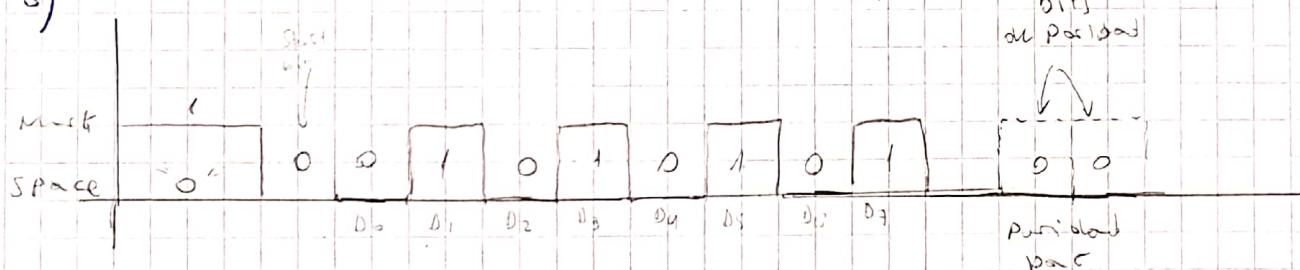
$$9598 \approx 9600 \text{ bps}$$

$$\frac{9600}{\text{no bytes}} = \frac{9600}{10} = 960 \text{ bps}$$

$\downarrow$   
8 + 2  
start ✓  
stop

bits  
de parada  
 $\nwarrow$   
---  
parada

5)



$$0xAA = 10101010$$

6) mov r0, #0

mov r1, #0

while: cmp r0, #200

addlo r1, r1, r0, LSL #1  $\rightarrow r1 = r1 + 2r0$

add r0, #1

b while.

NOTA

9) La diferencia entre la instrucción de salto B y la instrucción de salto BL es que este último, además de realizar el salto como hacia la instrucción B, guarda la posición de retorno en el link register, es decir en el registro R14.

10) Tipos de direccionamiento:

a) Direccionamiento por corrimiento

$ldr Rd, [Rn, #offset]$ ; Ejemplo =  $ldr r0, [r1, #4]$

b) Direccionamiento pre-indexado:

$ldr r0, [Rn #offset]!$ ; Ejemplo =  $ldr r0, [r1, #4]!$

Aquí lo que sucede es:

$r0$  carga el contenido de  $(r1 + \#4)$  y luego  $r1$  se actualiza

c) Direccionamiento post-indexado:

$ldr Rd, [Rn] #offset$ ; Ejemplo =  $ldr r0, [r1] \#4$

Aquí lo que sucede es:

$r0$  carga el contenido de  $r1$  y luego le suma  $\#4$

8)

Final 22/02/2018

HOJA N°

FECHA

1)

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

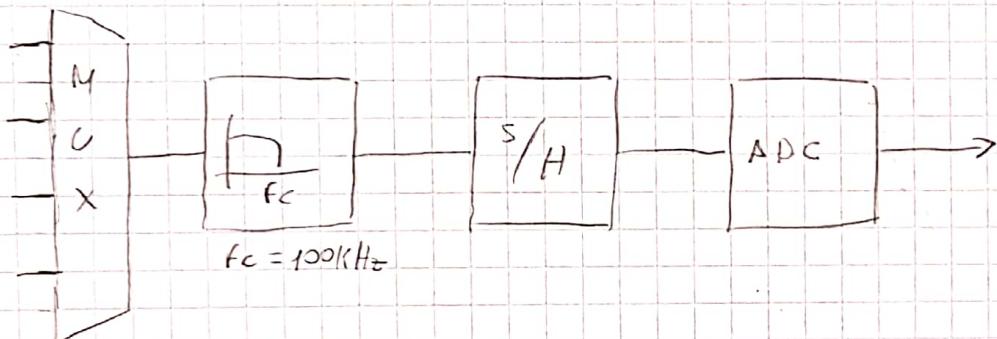
$$V_{LSB} = \frac{V_{ref}}{2^n} = 3,22 \text{ mV}$$

$n = 10$  bit

$$V_{res} = 20 \frac{\mu\text{A}}{\text{C}} \cdot 1 \text{ k}\Omega \cdot 0,1\% = 2 \text{ mV}$$

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

2)



$$\alpha) AB = \frac{Fc}{N} = \frac{100 \text{ kHz}}{4} = \boxed{25 \text{ kHz}}$$

$$\beta) f_{min} = 2 \cdot N \cdot Fc = \boxed{300 \text{ kHz}}$$

$$\gamma) t_{count} = \frac{1}{f_{min}} \approx \boxed{1,25 \text{ us}}$$

3)

$$\text{a) flash} \rightarrow t_c = n \cdot T_{clk}$$

$$\text{b) Aprox. SCC.} \rightarrow t_c = (n+1) \cdot T_{clk}$$

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

$$\star \text{Simple Round} \rightarrow t_c = (2^n + 1) \cdot T_{clk}$$

NOTA

4)

a)  $FS = \Delta V$

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

Verdadero  $\rightarrow \Delta V (FS)$  es independiente de  $n$

b) Verdadero

$$e_A = \pm \frac{1}{2} LSB$$

c)

$n = 8$  bits

$$V_{LSB} = \frac{FS}{2^n} = \frac{10V}{2^8} = 39,0625 mV$$

$FS = \Delta V = 10V$

a)  $39,1 mV \approx 39,06 mV$



b)  $V_{LSB} = \frac{FS}{2^n} = \frac{1}{2^n} \cdot FS$

$$\frac{1}{2^8} \cdot FS = \boxed{3906 ppm FS}$$

c)  $10V \cdot 0,39\% = 39 mV$



Verdadero

5) ORR R<sub>0</sub>, R<sub>1</sub>, R<sub>2</sub>

MVN R<sub>0</sub>, R<sub>3</sub>

6) mov r0, #9

mov r1, #h

- con signs -

cmp r0, r1

addge r0, r0, r1

sublt r0, r0, r1

7) STR r0, [r1, #12]

Pre-inicializar

Final 03/03/2018

1)

$$\Delta V = 10V$$

$$V_{in} = 3,922V = X$$

a)

$n = 8$  bits

0  $1^{\circ}$  clk  $\rightarrow X_7, 5V \rightarrow$  No.  $\therefore$  Descarto 5

$$\frac{10}{2^1} = 5V$$

1  $2^{\circ}$  clk  $\rightarrow X_7, 2,5 \rightarrow$  Si  $\therefore$  retengo 2,5

$$\frac{10}{2^2} = 2,5V$$

1  $3^{\circ}$  clk  $\rightarrow X_7, (2,5 + 1,25) \rightarrow 3,75 \rightarrow$  Si'.  $\therefore$  retengo 1,25

$$\frac{10}{2^3} = 1,25V$$

0  $4^{\circ}$  clk  $\rightarrow X_7, (2,5 + 1,25 + 0,625) \rightarrow 4,375 \rightarrow$  No.  $\therefore$  Descarto 0,625

0  $5^{\circ}$  clk  $\rightarrow X_7, (2,5 + 1,25 + 0,3125) \rightarrow 4,0625 \rightarrow$  No.  $\therefore$  Descarto 0,3125

$$\frac{10}{2^4} = 0,625V$$

1  $6^{\circ}$  clk  $\rightarrow X_7, (2,5 + 1,25 + 0,15625) \rightarrow [3,90625V]$

Gen 6° clk

$$\frac{10}{2^5} = 0,3125V$$

$$\frac{10}{2^6} = 0,15625V$$

b)  $f = 15$  KHz

$$t_c = (n+1) \cdot T_{clk} = (8+1) \cdot \frac{1}{15\text{KHz}} = 600 \mu\text{seg}$$

\* Como es sólo tiempo de HOLD  $\rightarrow \frac{600\mu\text{seg}}{2} = 300\mu\text{seg}$

3)  $n = 8$  bits

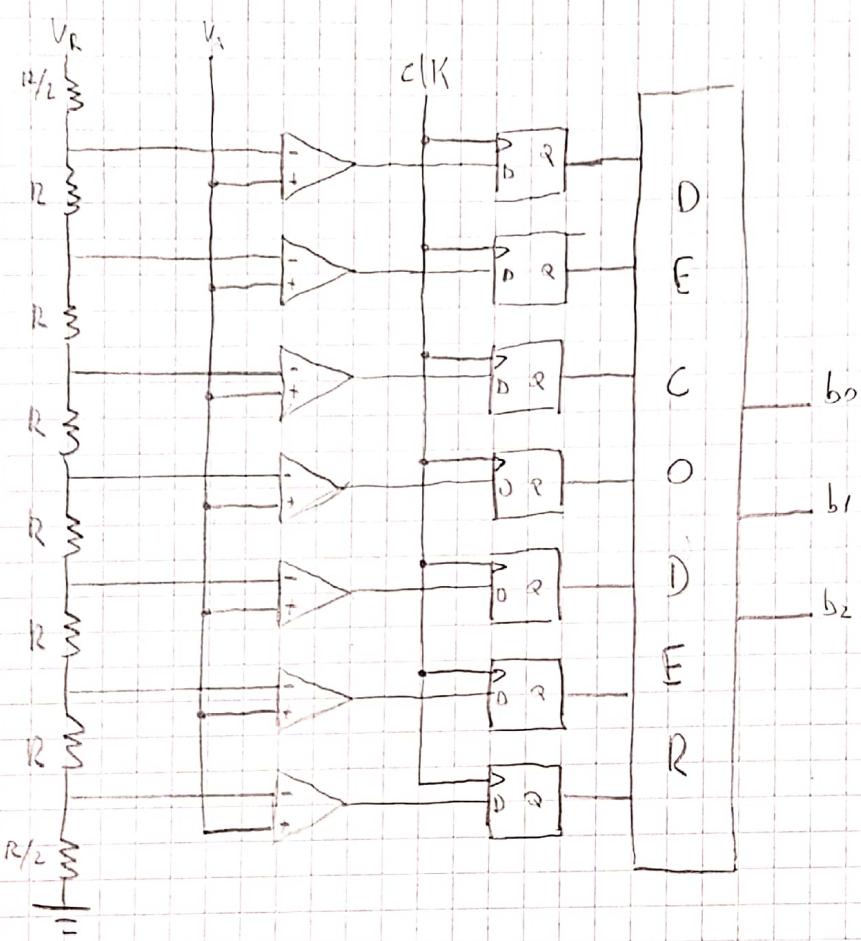
$$\Delta V = 1V$$

$$V_{USB} = \frac{1V}{2^n} = 3,90\text{ mV}$$

$$\epsilon_f = \pm \frac{1}{2} \text{ LSB} = \boxed{1,95\text{ mV}}$$

a) Ninguna de las anteriores.

2)

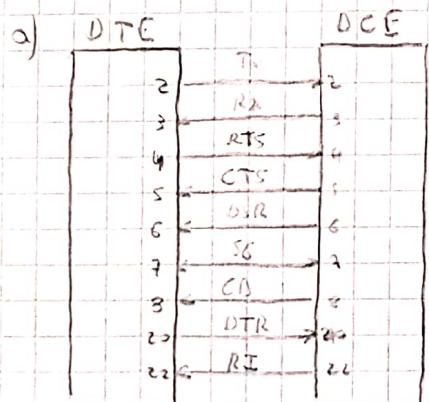


a) opción a

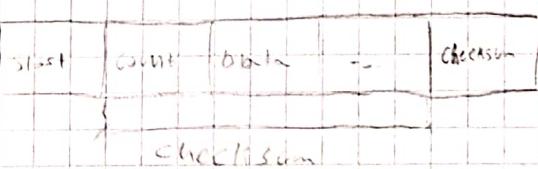
opción b

opción c  $\rightarrow$  como PLC llega hasta 1B con toda la func.

5)



b)



El checksum es el total de todos los bytes, sin contar el byte de start. Si supera los dos dígitos, se elimina el MSB.

c) El S17-3 P5 mite acceder a habilitar los resistores de LATCH DIVISOR,

tanto en nivel alto como en bajo (DLE y DLM). Este permite obtener la velocidad de transmisión, por ejemplo, 9600 baudios.

6)

$r_3 \rightarrow$  registro base  
 $\text{mov } r_1, \#0 \rightarrow i$

for :  $\text{cmp } r_1, \#100$

bge fin

$\text{lde } r_0, [r_3, r_1, LSL \#2]$

$\text{add } r_0, r_0, LSL \#7 \rightarrow r_0[i] = r_0[1].128$

$\text{str } r_0, [r_3, r_1, LSL \#2]$

$\text{add } r_1, \#1$

b for

fin:

7)  $\text{mov } r_2, \#7$

$(dr r_3, [r_0])$

$\text{bic } r_3, r_3, \#7$

$\text{add } r_3, r_3, r_2$

$\text{str } r_3, [r_1]$

8) EOR R<sub>0</sub>, R<sub>0</sub>, R<sub>1</sub>

EOR R<sub>1</sub>, R<sub>0</sub>, R<sub>1</sub>

EOR R<sub>0</sub>, R<sub>0</sub>, R<sub>1</sub>

9)  $LDR R_d, [R_n, R_m]! \equiv LDR R_d, [R_n, R_m]$   
 $AOD R_n, R_n, R_m$

[verdadero]

# Fundamentos

①

$\alpha = 0.015$

$$V_{LSB} = \frac{V_{REF}}{2^N} = \frac{3.3V}{4096} = 3.22\mu V$$

$V_{REF} = 3.3V$

$$I_T = 20 \frac{\mu A}{^{\circ C}} \cdot 100 = 200 \frac{\mu A}{^{\circ C}}$$

$$V_{PUSH} = FT \cdot I_{T, 0} = \frac{0.1}{4} \cdot 200 \frac{\mu A}{^{\circ C}}$$

$$I_T = 20 \frac{\mu A}{^{\circ C}}$$

$$V_{REF} = 20 \frac{\mu V}{^{\circ C}}, 0,1^{\circ C} = 2 \mu V$$

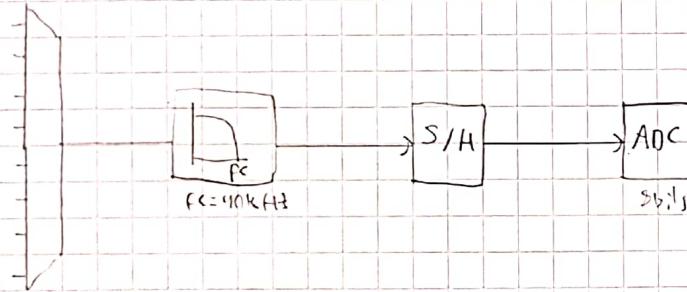
Gráfico de  $V_{REF}$  vs  $T$

$\Delta T = 0,1^{\circ C}$  por cada

$$G = \frac{V_{LSB}}{V_{REF}} = \frac{3.22 \mu V}{2 \mu V} = 1.61$$

A

②



$$\textcircled{a} \quad AB = f_C = \frac{40 \text{ kHz}}{8} = 5 \text{ kHz}$$

$$\textcircled{b} \quad f_{min} = 2 \cdot N \cdot f_C = 2 \cdot 4 \cdot 40 \text{ kHz} = 320 \text{ kHz}$$

$$\textcircled{c} \quad t_{crossover} = \frac{1}{f_{min}} = \frac{1}{320 \text{ kHz}} = 3,125 \mu s$$

③

$$\textcircled{d} \quad t_{crossover} = 0,7 \text{ clk} \quad (\text{Supuestamente es el tiempo skew entre } 37\% \text{ (crossover)})$$

$$\textcircled{e} \quad t_{crossover, real} = (n+1) \cdot 7 \text{ clk}$$

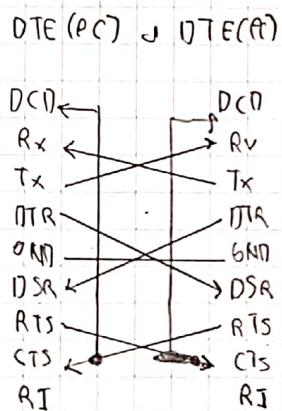
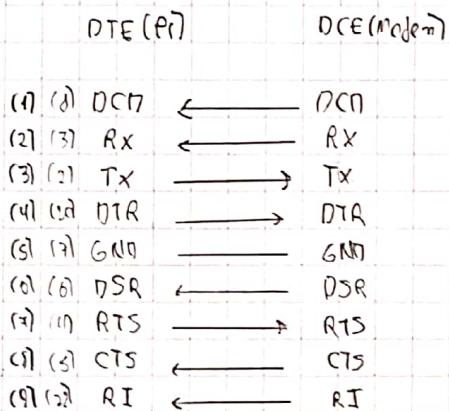
$$\textcircled{2} \quad T_{clock} = 2^{n+1} T_{clk}$$

$$\times \text{Si plena} = (2^n + 1) T_{clk}$$

II

③

(creando DTE-DCE)



### ④ Técnica checksum

Es un método de detección de errores, en el mismo se comprueba una suma (checksum), propuesta al receptor en transmisión sincrónica asincrónica, se transmite al final del mensaje un byte (sum) cuyo valor es la suma de los bytes de todos los caracteres que han sido transmitidos en el mensaje. El receptor implementa una rutina que suma todos los bytes de los recibidos y al resultado se le sumará el último byte (que posee la información de los de la suma de los caracteres transmitidos) y si la recepción fue correcta da zero.

Ej.: 40H 35H 0EH

SUMA:

$$40H + 35H + 0EH = 83H$$

$$\begin{array}{r}
 83H = 1000\ 0011 \\
 0111\ 1100 \\
 + \quad \quad \quad \\
 \hline
 0111\ 1101 = 7DH \quad (\text{junto con el resto})
 \end{array}$$

Final 07-12-18

⑤

int fun(int x, int y) {

if (x+y) < 0  
return 0;

else  
return 1;

3

main: mov r1, #2

mov r2, #3

bl fun

;

funcion: add r3, r1, r2

cmp r3, #0

bge else

mov r0, #0

b fin

else: mov r0, #1

fin: mov pc, lr

funcion: ⑥ uso  $x = rs$ ,  $y = rs$  y guardo el retorno en la dirección apuntada por r3 (anterior)

push { r5, r6, lr }

add r8, r5, r6

cmp r8, #0

bge else

mov r9, #0

str r9, [r3]

b end

else: mov r9, #1

str r9, [r3]

end: pop { r5, r6 }

mov pc, lr

⑥

hex = 0x0000 FF EE



0xEE	0xb009
0xFF	0xb001
0x90	0xb002
0xAF	0xb003

R1

Rn+1

Rn+2

Rn+3

③

④

str r0, [r1, #12];

esta en filtro : r0

⑤

mov r5, #12

mov r7, #0      ⑥ condición

tbl detector

⑦ variable

loop: b loop

detector: push {r0, lr}      ⑧ conjunto de pines

ciclo: chf r7, #249

b eq fincicle

ldr b r4, [r0], #1

chp r4, r5      ⑨ r5 detecta byte

b eg encodet

b ciclo

encodetdo!

mov r4, #0

mov r3, r0  
b end.

fincicle:

mov r7, #0

mov r3, #1

end: pop {r0, lr}

bxr pc, lr

Final 24-05-18

④

⑤ Verdadero

⑥ Verdadero

⑦ Verdadero (2 partes y verdad)

⑧ Falso porque no sea evitado ej: b eliges

⑨ Verdadero (en estim)

⑩ Verdadero (y, y, y, y, y, y)

⑪ Falso solo se vienen los registradores

②

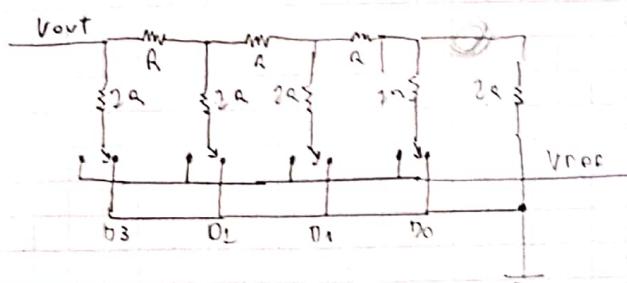
(avanza más no tiene instrucción de selección de fuente y destino)

③

④ Alguno comprender tiene que tener un bucle que regrese al final

⑤

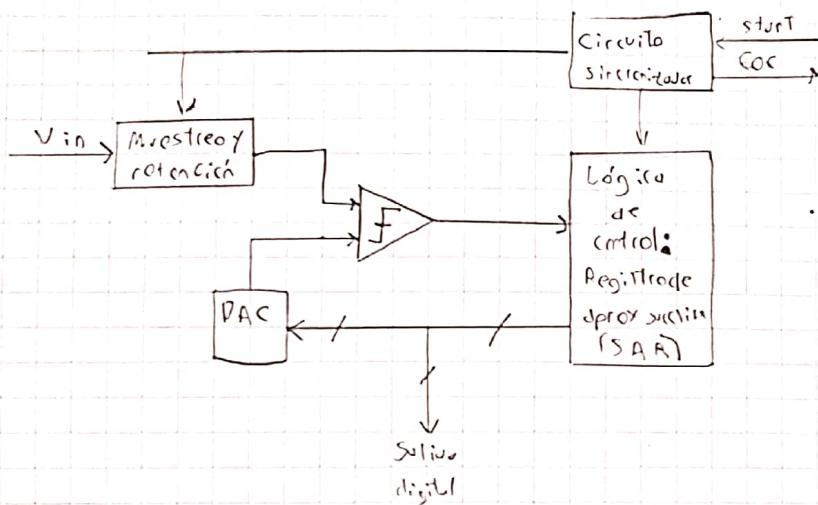
2N resistencias



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

⑤

A/Dc aproximación sucesiva



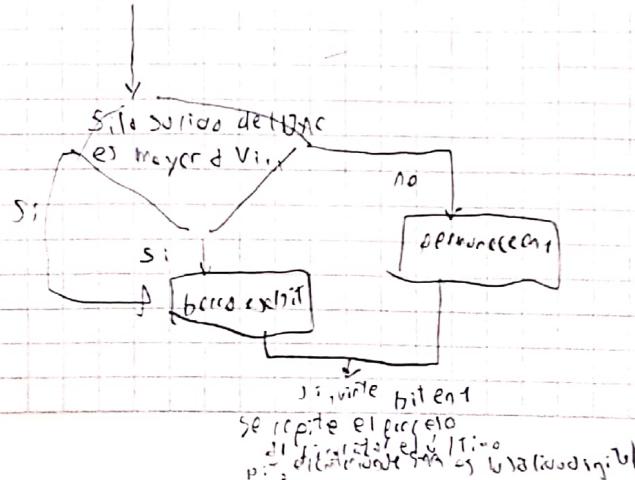
⑥  $t_{c} = T_0(N+1)$

El tiempo de conversión depende del número de bits del registro  $N$ , se realizan  $N$  inversiones de signo para si  $V_{in}$  es 2 el resultado del DAC va a ser menor 1 si pasa en 0 es señal, repitiéndose el proceso. El 1 es porque la pulsación digital final no pasa al registro de salida hasta el siguiente blanqueo de reloj; en el que también se informa el fin de conversión (EOC).

⑦

Los señales START, COMPD, STIN, CTR, RDY, RST, y BORDE TELES son bits de los SAR dejando el A/D bien

↓  
Los salidas de los SAR activan el DAC interno



Fri 24-05-18

⑦

```
for (i=0; i<s; i++)  
    s[i] = b[r-i];
```

3

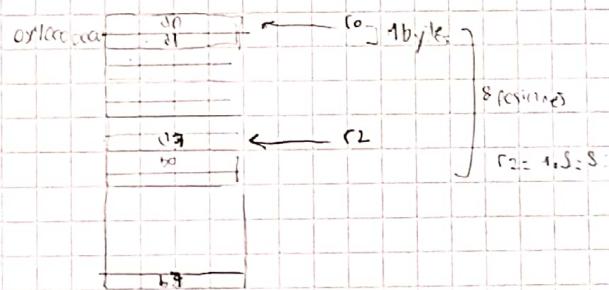
for r0, #0x10000000

One size twice the array

mov r1, #0      @ r1=i  
add r2, r0, #7

for: cmp r1, #8      @

seg end .



b for

end: b end

⑧

① str r6, [r4, #4] previous value (sin actualizar)

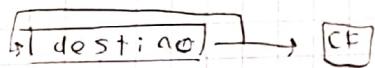
② ldrb r4, [r3, r2]! previous value de actualizado

③ lsr r3, [r2], #5 post: > r4

④ ldrsh r12, [r6] from normal to actualizado

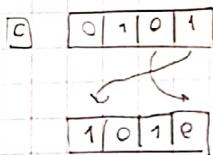
①

ROR

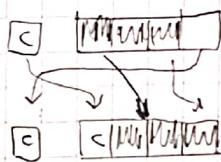


los bits se corren desde LSB a MSB

RAX



RCR tiene en cuenta carry



RAX tiene en cuenta carry

RAX tiene en cuenta el carry en el desplazamiento

②

MOV R3, #0      ⑥ int := 0  
 MOV R2, #0x4CC0000      ⑥ r2 dirección origen

for: cmp r1, #8

beq end

a.

rsd r3, r1, #7      ⑥  $r3 = 7 - r1 = 7 - i$

ldr r4, [r2, r3]      ⑥  $r3 = 7 - 0 = 7$ ;  $r3 = 7 - 1 = 6$  no se actualiza r2

str r4, [r2, r3]

add r1, r1, #1

b for

⑩

b) strcmp

loop: bloop

⑪

strcmp: push {r1, r2, lr}

ciclo: ldr r3, [r1], #1

cmp r3, #0x00

beg fin

ldr r4, [r2], #1

cmp r4, #0x00

beg fin

cmp r3, r4

beg verfin

verfin: cmp r3, #0x00

beg igual

b ciclo

fin: pop {r1, r2, lr}

mov r0, 0x00000000

mov pc, lr

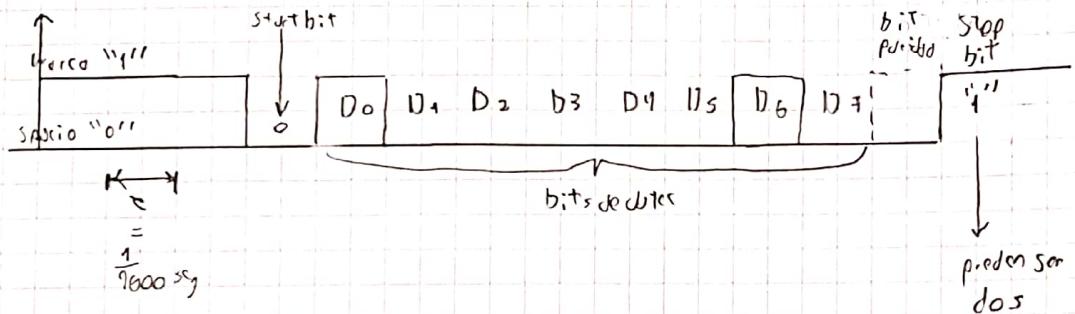
igual: pop {r1, r2, lr}

mov r0, 0xFFFFFFFF

mov pc, lr

④

Trama RS-232



$$T = \frac{1}{V}$$

$$V = \frac{1}{T} \quad (\text{Velocidad de transmisión})$$

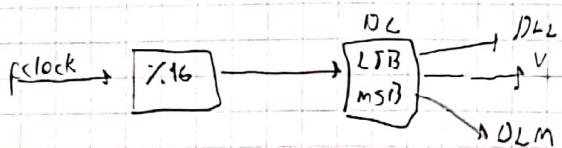
El bit de paridad: Para: vale "0" ó "1" dependiendo de la cantidad de "1" paríodo que sea par

$$V = \frac{F_{\text{clock}}}{16 \times D_L}$$

El chip UART 16550 contiene 16 registros, el bit 7 del registro LCR me permite habilitar los registros del latch divisor, en su parte baja y alta (DLH y DLW).

Lo primero que se hace es configurar la velocidad, cantidad de bits de datos, paridad, tipo paríodo, p.í. de paríodo.

El protocolo RS-232 es full duplex porque permite enviar y transmitir datos simultáneos.



$$D_L = \frac{f_{\text{clock}}}{16 \times V} = \frac{1,8432 \text{ MHz}}{16 \times 9600 \text{ sps}} = 0000 \text{ DLL}$$

Für 22-02-18

①

$$N = 10 \text{ bits}$$

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

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

② G =

$$10 \text{ bit} = 0,1^{\circ C}$$

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

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

$$V_{LSB} = 20 \text{ mV}, 0,1^{\circ C} = 2 \text{ mV}$$

$$G = \frac{V_{LSB}}{V_{ref}} = \frac{3,22 \text{ mV}}{2 \text{ mV}} = 7,61 \quad (\text{Mangage weiter abgerundet falls ggf. misst wird in } 7,6^{\circ C})$$

③

$$f_C = 100 \text{ kHz}$$

$$AB = \frac{f_C}{f_{min}} = \frac{100 \text{ kHz}}{4} = 25 \text{ kA}$$

$$f_{min} = 2 \text{ kHz} \cdot f_C = 2 \cdot 4 \cdot 100 \text{ kHz} = 800 \text{ kHz}$$

$$t_{on} = \frac{1}{f_{min}} = \frac{1}{800 \text{ kHz}} = 1,25 \mu s$$

④

$$\text{⑤ } t_{c,flush} = n_c T_{clk}$$

$$\text{⑥ } t_{c,opere} = (n+1) T_{clk}$$

$$\textcircled{1} \quad \text{Ciclo de } 12 \text{ bits} = 2^{12} T_{clk}$$

\textcircled{4}

\textcircled{5}

$$FS = \Delta V$$

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

$\Delta V$  es el rango de medida de  $n$ ,  $V_{LSB}$  es independiente de  $n$

\textcircled{6}

$$eg = \pm \frac{1}{2} LSB \text{ (verdadera)}$$

\textcircled{7}

$$LSB = \frac{\Delta V}{2^n} = \frac{19V}{2^8} = 0,03906 \text{ V}$$

$$1LSB = \cancel{\frac{FS}{2^n}} - \frac{FS}{2^8} = 3,906 \times 10^{-3} FS = 3,906$$

$$1LSB = \frac{FS}{2^n} = \frac{FS}{2^8} = 3,906 \text{ ppm} FS = \frac{3,906}{1000000} \cdot 10 = 0,03906$$

$$1LSB = \frac{FS}{2^n} = \frac{FS}{2^8} = 0,39\% FS$$

Verdadero

Final 22-08-18

④

ORL R0, R1 ORL R2

R0	R1	R2
0 0	1	0010
0 1	0	1100
1 0	0	1110
1 1	0	0001

ORL R0, R1 ORL R2

⑥ R0 = R1 ORL R2

MVN R0, R0

⑦ R0 = R0

⑤

if (g >= h)

g = g + h;

else

g = g - h

CMP R0, R1

ADDGE R0, R0, R1

SUBLT R0, R0, R1

⑥

STR R0, [ R1 #12 ]

primero dato sin actualizar el registro R1

⑦

MV R0, R0

MV R0, R0

AND R0, R0, R1, ISR #1

⑧ Desplaza bits en 1, el bit más significativo es el bit de carry

ANDC R0, R0, R1

⑨ Resetea R0 si la lleva carry seteo (es decir seteo)

Cup 1, 110

bag 1000

6 devueltas las unidades de 1000 es cero

6 puro miles bne cuando los ceros son 1000 (cero)

F17/20-12-18

①

Ejercicio nº 1 y condición: La diferencia entre los registros de salida es que se da una cierta condición o cumplir, indicado por el estado de los banderas, además se puede reducir al número de ciclos necesarios para un cierto algoritmo

cmp r4, #0

add r0,r1,r2

⑥ se ejecuta sin ninguna condición

②

cmpl r4, #0

addsg r0,r1,r2

⑥ se ejecuta si no es cero

③

Los banderas son

N (0's): indica el signo entero del bit más significativo del resultado de la ALU (1 si es negativo)

Z: Zero si el resultado da cero (1 si no es cero)

C: Cada ALU operación copia este bit para su uso en la siguiente operación o si se necesita preservar

V: overflow

Los banderas nos sirven para tener las diferentes condiciones, por ejemplo si un registro es igual a otro  $Z=1$

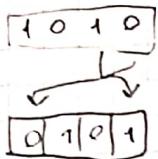
④

xor r12, r12

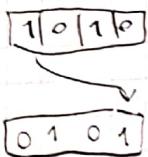
sub r12, r12

②

Nearest ASL, Ni ROL



ROL



ASL (your)

③

7, 3, 5, 7, 5, 4, 2, 0

MOV r0, #0x500      6 LEDs

MOV r8, #0  
ADD r1, r9, #1  
MOV r7, #0

b1 sequence

NOT r2, r0, #6

b1 sequence 2

<<<<<< ; push {r7,lr}

STRB r8, [r7], #2

CMP r7, #3

bne sequence 1;

MOV r7, #0

POP {r7,lr}

MOV PC, r

scenario 2: push {r2, lr}

strb r8,[r2],#-2

cmp r7,#3

beq scenario2

mov r7,#0

pop {r2, lr}

④

⑤

$$e_g = \frac{1}{2} \text{ LSB} = \frac{1}{2} \frac{\Delta V}{2^0} = \frac{1V}{2^0} \cdot \frac{1}{2} = 1,95 \text{ mV}$$

$$e_g = \frac{1}{512} \cdot \frac{1}{2} = \frac{1}{1024} = 0,97 \text{ mV}$$

$$e_g = \frac{1V}{2^1} \cdot \frac{1}{2}$$

$$2 e_g = \frac{1V}{2^0}$$

$$2^n = \frac{1V}{2^0}$$

$$\ln(2^n) = n = \ln\left(\frac{1}{2}e_g\right)$$

$$n =$$

$$1 \text{ LSB} = \frac{\Delta V}{2^n} = \frac{1V}{2^8} = \frac{1}{256}$$

$$e_g = \frac{1}{2} \text{ LSB} = \frac{1}{512}$$

$$\frac{e_g}{2} = \frac{1}{2^{12}} \cdot \frac{1}{2} = \frac{1}{1024}$$

$$e_g = \frac{1}{1024} = \frac{1}{2} \cdot \frac{\Delta V}{2^n}$$

$$2^n = \frac{\Delta V}{2} \cdot 1024$$

$$n = \ln \left( \frac{\Delta V \cdot 1024}{2} \right)$$

$$n \approx 9.675$$

$$e_g = \frac{1}{2} \cdot \frac{\Delta V}{2^{2.8}} = 7.5 \times 10^{-6}$$

$$e_g' = e_g = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2^8} \cdot \frac{1V}{2} = \frac{1V}{2^{10}} = \frac{1}{1024}$$

$$LSB = \frac{\Delta V}{2^{n'}} = \frac{e_g'}{2^9}$$

$$n' = 9$$

i.e. 9 bits

F: 11 11-10-18

②

③

$$\frac{10V}{2^1} = 5V$$

$$Y = 3,972V$$

$$\frac{10}{2^2} = 2,5V$$

$$1^{\text{er}} \text{ clock } \times 2,5V \rightarrow \text{no} \rightarrow \text{descarto SI}$$

$$2^{\text{do}} \text{ clock } \times 2,5V \rightarrow \text{si} \rightarrow \text{rele-}2,5V$$

$$3^{\text{er}} \text{ clock } \times 2,5V + 1,5V \rightarrow \text{si} \rightarrow \text{rele-}0,725$$

$$4^{\text{to}} \text{ clock } \times 2,5V + 1,25 + 0,625 \rightarrow \text{no} \rightarrow \text{descarto 0,625}$$

$$5^{\text{th}} \text{ clock } \times 2,5V + \frac{10}{2^2} + 10 + \frac{10}{2^3} \rightarrow \text{no} \rightarrow \text{descarto}$$

$$5^{\text{th}} \text{ clock } \times 2,5V + 1,25 + 0,3125 \rightarrow \text{no} \rightarrow \text{descarto 0,3125}$$

$$6^{\text{th}} \text{ clock } \times 2,5V + 1,25 + 0,15625 \rightarrow \text{si} \rightarrow \text{rele-}0,15625$$

$$\text{Valor de SAA em } 5^{\text{th}} \text{ clock} = 2,5V + 1,25V + 0,3125V = 3,90625V$$

④

$$t_c = T_{CK}(n+1) = \frac{1}{f_{CK}}(n+1) = \frac{1}{75 \text{ KHz}}(8+1) = 6,09 \mu s$$

$$t_{Hold} = \frac{t_c}{2} = \frac{6,09 \mu s}{2} = 3,00 \mu s$$

⑤

⑥

$$j = 2 \text{ Nsec} = 2,8 \cdot 10^{-9} \text{ sec} = 2,8 \text{ nsec}$$

⑦

$$t_{Setup} = \frac{1}{55} = \frac{1}{160 \text{ MHz}} = 6,25 \text{ ns}$$

④

$$\textcircled{a} \quad t_{cpl, \text{bus}} = T_{\text{clk}} n$$

$$\textcircled{b} \quad t_{cpl, \text{opcodes}} = T_{\text{clk}} (n+1)$$

$$\textcircled{c} \quad t_{cpl, \text{immediate}} = 2^{n+1} T_{\text{clk}}$$

$$* \quad t_{cpl, \text{register}} = 2^n + 1 \quad T_{\text{clk}}$$

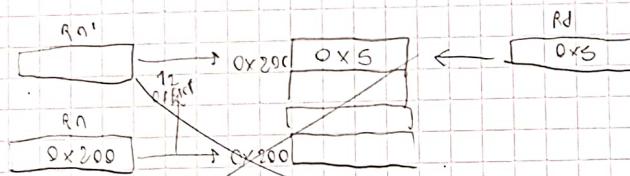
Ejercicio 14.10.1J

⑤

ldr Rd, [Rn, Rm]! es equivalente a

ldr Rd, [Rn, Rm] C cargo Rd = Rn + Rm y Rn = Rn + Rm (se cancela)

sd Rr, Rn, Rm



Resumen de la ejecución: primero guarda el dato en 0x200 y en la próxima memoria lo guarda en 0x20C

⑥

r3 apunta a 0x8000

① str r6, [r3, #12] carga/rí el valor de r6 en 0x800C si r3 = 0x8000 no cambia

X

② strb r3, [r3], #4 almacena 0x8000 en la dirección indicada por r3, luego se incrementa 4  
nose si guarda solo 0x80000000 (r3 sigue teniendo 0x8000)

③ ldrh r5, [r3], #8 copiaría lo que hay en 0x8000 y lo que incrementa en 8 (r3 = 0x8000 + 8 = 0x8008)

④ dr r2, [r3, #16] copia en r12 lo que hay en r3 + 16 = 0x8000 + 0x10 = 0x8010, y r3  
queda en 0x8010 (se cancela)

EOR r12, r12, r12

⑥ corregirse para 3 veces

SVB r12, r12, r12

END r12, #0x0000000000

BIC r12, r12, r12

$$\textcircled{6} \quad r12 = r12 + r12$$

MOV r2, #7

⑥ lo usó para crear el 1 de la v11111113

LDR r3, [r0]

⑥ copia logico de r0 en r3 (para meter MOV r3, r0 si no lo hace se pone 0)

BIC r3, #0x0f

⑥ pone en cero la v1111113 oses r3 = r3.  $\frac{0111}{28 veces} = r3. \frac{1000}{23 veces}$

ADD r3, r3, r2

⑥ sumo 7 porque sum 31 ( $0b\ 00010111101110001110111101101000 + 111$ )

STR r3, [r1]

⑥ guarda el valor en r1 (hacer {r3, r3})

⑥ en una instrucción ORR R1, R0, #0x7

7)

MOV r9, #0x200

MOV r7, #0

MOV r2, #0

arreglo: ldrsb r3, [r9], #1 → cmp r2, #249 beg end  
add r2, r2, #1

CMP r3, #0

bge arreglo

⑥ si r3 es >= 0 sigue recorriendo

TST r3, #1

⑥ es <= 0 ANI pero afecta solo los banderas si es igual es igual

bne arreglo

⑥ Si termina r3 en cero es PUR, con TST afecta solo los banderas, luego

ADD r3, r1, #1

hace es una and,  $r3 = r3. 00000001$  (si es cero z=1 y se ejecuta otro)

b arreglo

cero beg etiqueta)

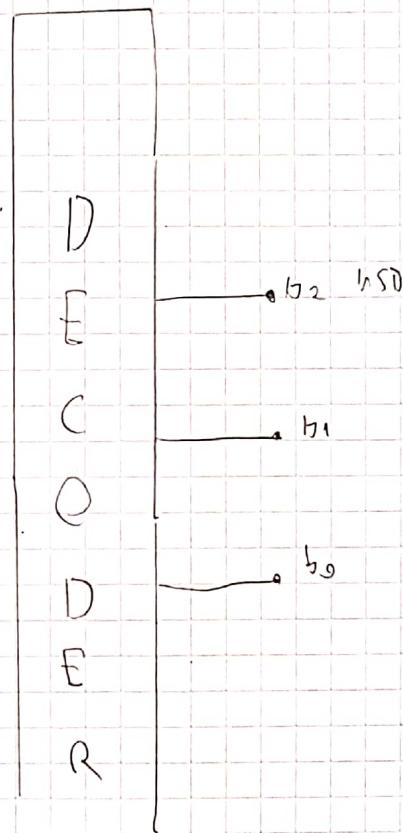
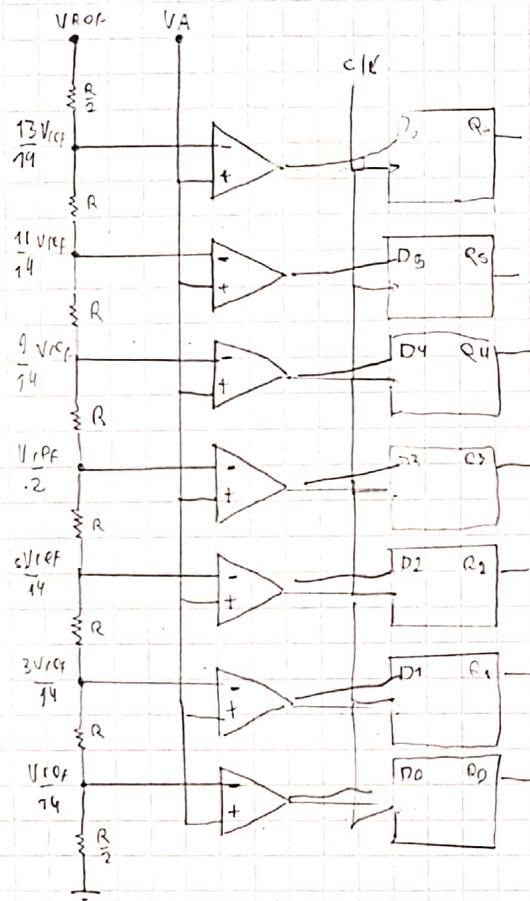
end:

MOV r7, r1

1.../08-03-12

②

Conversor A/D tipo flash de 3 bits



00000000	01000000	11000000
00000000	00000000	00000000
00000001	00000001	00000001
00000010	00000010	00000010
00000011	00000011	00000011
00000100	00000100	00000100
00000101	00000101	00000101
00000110	00000110	00000110
00000111	00000111	00000111
00001000	00001000	00001000
00001001	00001001	00001001
00001010	00001010	00001010
00001011	00001011	00001011
00001100	00001100	00001100
00001101	00001101	00001101
00001110	00001110	00001110
00001111	00001111	00001111
00010000	00010000	00010000
00010001	00010001	00010001
00010010	00010010	00010010
00010011	00010011	00010011
00010100	00010100	00010100
00010101	00010101	00010101
00010110	00010110	00010110
00010111	00010111	00010111
00011000	00011000	00011000
00011001	00011001	00011001
00011010	00011010	00011010
00011011	00011011	00011011
00011100	00011100	00011100
00011101	00011101	00011101
00011110	00011110	00011110
00011111	00011111	00011111
00100000	00100000	00100000
00100001	00100001	00100001
00100010	00100010	00100010
00100011	00100011	00100011
00100100	00100100	00100100
00100101	00100101	00100101
00100110	00100110	00100110
00100111	00100111	00100111
00101000	00101000	00101000
00101001	00101001	00101001
00101010	00101010	00101010
00101011	00101011	00101011
00101100	00101100	00101100
00101101	00101101	00101101
00101110	00101110	00101110
00101111	00101111	00101111
00110000	00110000	00110000
00110001	00110001	00110001
00110010	00110010	00110010
00110011	00110011	00110011
00110100	00110100	00110100
00110101	00110101	00110101
00110110	00110110	00110110
00110111	00110111	00110111
00111000	00111000	00111000
00111001	00111001	00111001
00111010	00111010	00111010
00111011	00111011	00111011
00111100	00111100	00111100
00111101	00111101	00111101
00111110	00111110	00111110
00111111	00111111	00111111

27 cm: Mector

z-1 cm: mector

④

Un convertidor A/D de alta rapidez se utiliza para implementación (multicapa digital); el tiempo de conversión es corto porque tiene que operar g-a dentro del convertidor, se inicia durante y para hasta que se escoge C.

①

$$\frac{10V}{2^1} = 5$$

$$1^{\text{er}} \text{ ciclo: } 3,922V < 5V \quad (\text{descarto}) \quad SAR = 0000\ 0000$$

$$2^{\text{do}} \text{ ciclo: } 3,922V > 2,5V \quad (\text{retengo}) \quad SAR = 0100\ 0000$$

$$3^{\text{er}} \text{ ciclo: } 3,922V > 3,75V \quad (\text{retengo}) \quad SAR = 0110\ 0000$$

$$4^{\text{to}} \text{ ciclo: } 3,922V < 3,75V + \frac{10}{2^4} \quad (\text{descarto}) \quad SAR = 0110\ 0000$$

$$5^{\text{to}} \text{ ciclo: } 3,922V < 3,75V + \frac{11}{2^5} \quad (\text{descarto}) \quad SAR = 0110\ 0000$$

$$6^{\text{to}} \text{ ciclo: } 3,922V < 3,75V + \frac{10}{2^6} = 3,90625V \quad SAR = \underline{01100100}$$

$$Tc = (n+1) \cdot T_{clk} = \frac{1}{f_{clk}} \cdot (8+1) = \frac{1}{15kHz} \cdot 9 = 600 \mu s$$

$$T_{hold} = \frac{Tc}{2} = 300 \mu s \quad (\text{nose})$$

②

③

$$e_g = \frac{1}{2} V_{LSB} = \frac{1}{2} \frac{\Delta V}{2^n} = \frac{1V}{2^{10}} = 1,95 mV$$

$$e_g' = \frac{1}{2} e_g = \frac{1}{2} \frac{1}{2} \frac{\Delta V}{2^n} = \frac{1}{2} \frac{\Delta V}{2^{11}} \quad (\text{se ha dividido } e_g \text{ entre el divisor})$$

Final 08-03-18

②

```
int i;  
int tmp[100];  
for (i=0; i<100; i++)  
    tmp[i] = tmp[i] * 125;
```

MOV R3, 0X400  
MOV R6, R3  
MOV R7, #0

MOV R2, #10      ⑥ recorrer el arreglo R2=i

MOV R3, 0X400  
MOV R6, 0X404  
FOR R3, H0

⑦

b1\_funcion

loop: b loop

funcion: push {R3, lr}

for: ldr r4, [R3], #4

mov r5, R4, #125

str r5, [R6], #4

add R2, R2, #1

cmp R2, #100

bne for

pop {R3, lr}

mvn pc, lr

for: ldr r4, [R3], #4

add r5, R4, R4, #125

str r5, [R6], #4

add R2, R2, #1

cmp R2, #100

bne for

⑥ Tengo que copiar R3 en R5 porque si no

se va a empelar a guardar en 0X404

②

ldr R3, R7

bic r0, #0x0007

oar M, r0, H0x7

ldr r4, #0x0007  
ldr r5, [r0]

add r0, r0, #0x0007

str r0, [r1]

bic r5, r4

add r5, r0, r3

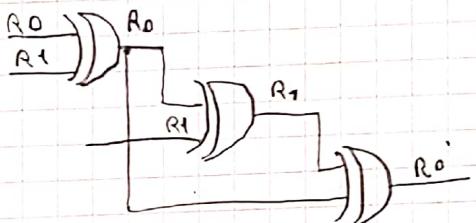
str r5, [r1]

③

eor R0, R0, R1

1111 Fn

0100 A1



rdr R1, R0, R1

1011 R0

0100 S1

1111 A1

1011 R0

0100 ED

④

ldr Rd, [Rn, Rm]!

copd Rn+Rm, Rn=Rm

ldr Rd, [Rn, Rm]

add Rn, Rn, Rm  
⇒ forming parallel adder

Final 26-07-18

④

$R0 = 0;$

$R1 = 0;$

while ( $R0 < 200$ ) {

$R1 = R1 + R0 / 2$

$R0++;$

}

⑤

MOV R0, R0

MOV R1, R0

while : CMP R0, #199

bge end

ADD R1, R1, R0, LSL #1

ADD AC, AC, #1

b white

END; b end

⑥



ldrb R1, [R0]

gve acc0

Salida:  $R12, \{23, 55\}$  por PUSH  $\{39, 85\}$

Entrada:  $R12, \{23, 55\}$  por POP  $\{34, 85\}$

- ⑥ Diferencia es que el punto de retorno en el LR, en vez de en R14 y el programa entra a la siguiente linea de código siendo que si bien a esto debió pasar por R14.

Tipo de retorno: R10

⑦ Direcciones de retorno (prediccion correcta)

1er Ed, [R12, #4000]      1er R9, [R12, #4]

segundo el primero incluido en R14 en R9 y R12=0

Dirección de predicción:

1er Ed, [R9, #4000]      1er R9, [R9, #4]:

se cumple Ed = R9 & R9 > 0

⑧ Dirección de ejecución:

1er Ed, [R9], #offst

Segundo R9 = R9 y luego se actualiza la base

Für 26.07.18

①

$$FT = 20 \frac{mV}{kg}$$

$$V_{LSB} = V_{REF} = \frac{5V}{2^n} = 610,35 mV$$

curg 4 kg

$$V_{LSB} = 610,35 mV$$

$$r_0 = 10 \text{ (per ref)}$$

$$V_{PSSO} = FT \cdot R_{ES} = 20 \frac{mV}{kg} \cdot 1 \times 10^3 \frac{kg}{m} = 20 mV$$

$$V_{REF} = 5V$$

$$I_0 = 131,95$$

$$G = \frac{V_{LSB}}{V_{ES}} = \frac{610,35 mV}{20 mV} = 30,519,5$$

②

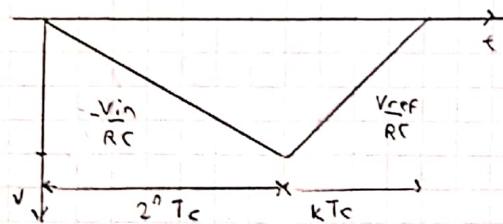
③

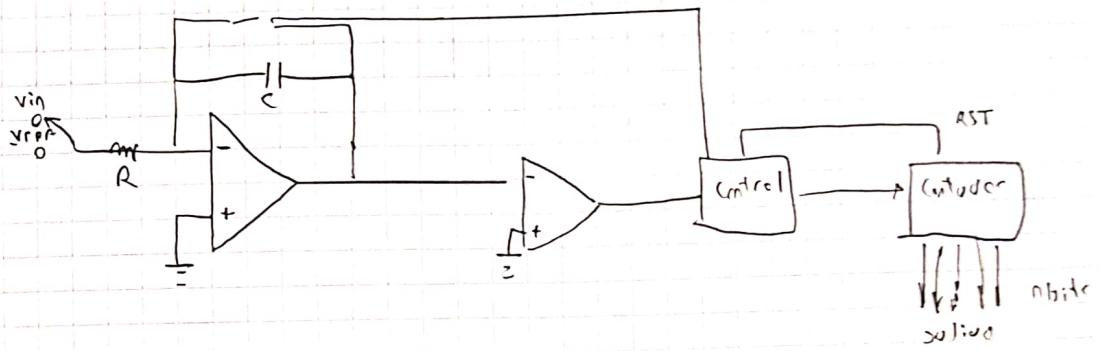
$$\epsilon_{triple} = \frac{1}{4} V_{LSB} \cdot \frac{1}{n} \frac{V_{REF}}{2^n} = \frac{1}{4} \frac{3V}{2^n} = 2,92 mV$$

$$\epsilon_{GJmax} \frac{1}{4} \frac{1}{2^n} = 9,76 \times 10^{-4}$$

④

⑤





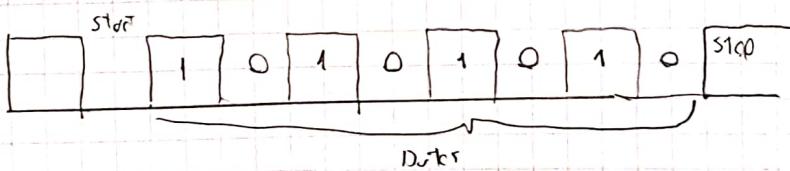
Se inicia con  $C$  descargado, se mantiene la señal nula  $V_{in}$  (tension de entrada) por medio de el amplificador configurado como integrador, se integra la señal en modo pendiente positiva. El contador comienza hasta un tiempo corto ( $0 \text{ a } 2^n - 1$  voltios) en ese momento la llave se cierra a  $-V_{ref}$  y comienza a contar inversamente, hasta que el capacitor se descharge completamente.

$$2^n T_c \left( -\frac{V_{in}}{A_C} \right) = k T_c \frac{V_{ref}}{R_C}$$

$$-V_{in} = K \frac{V_{ref}}{2^n}$$

⑤

Transmisor RS232 en 8N1 que transmite 0xA9



⑥

$$(b/s) = \frac{9600}{n \cdot t_{bit}} = \frac{9600}{40} = 960 \text{ bps} \quad (\text{número de bits debidos por s})$$

$$t_{bit} (\text{tiempo de bit}) = \frac{1}{\sqrt{9600}} = 104.16 \mu s$$