eman ta zabal zazu
universidad del país vasco
euskal herriko unibertsitatea

departamento de electricidad y electrónica
elektrika eta elektronika saila

Examen de S.E.T.I.

1er curso de Ingeniería Electrónica
26 de junio de 2001

Cuestiones teóricas.

1 Relación entre ASCII, ISO-8859-1, ISO/IEC-10646-1 y Unicode. (No se pide una explicación de cada uno, sino la relación existente entre ellos).

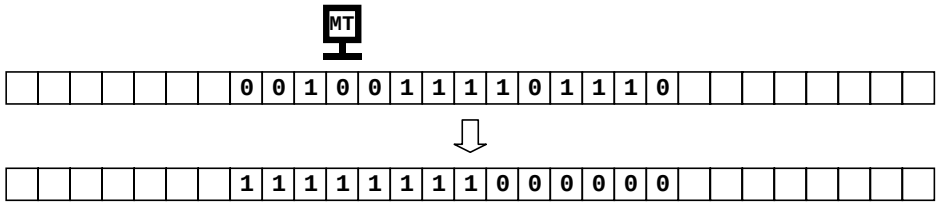
2 Un microprocesador necesita una señal de "reloj" para ejecutar en serie sus acciones internas. Evidentemente no podemos aumentar la frecuencia de dicha señal a nuestro gusto para aumentar la velocidad de ejecución. ¿Por qué?.

Cuestiones prácticas.

3 Represente las siguientes cantidades en binario según el formato de números reales IEEE-754 restringida a 10 bits con 4 de exponente y 5 mantisa:

a) 0.0
b) 0.125
c) 3,14159265358979323846264338327 95

4 Escriba la tabla de una máquina de Turing que, dada una cinta en blanco con una sección conexa de unos y ceros distribuidos aleatoriamente, reordena estos dejando los unos a un lado y los ceros a otro. La máquina podrá encontrarse situada inicialmente en un punto cualquiera dentro de la sección de unos y ceros.



5 Escriba una rutina en ensamblador del μP68000 que recoja en una tabla el número de veces que aparece cada carácter en un bloque de memoria. La tabla será de 256 posiciones de modo que pueda almacenar las cuentas de todos los posibles caracteres codificables en un byte. El prototipo de la rutina será el siguiente:

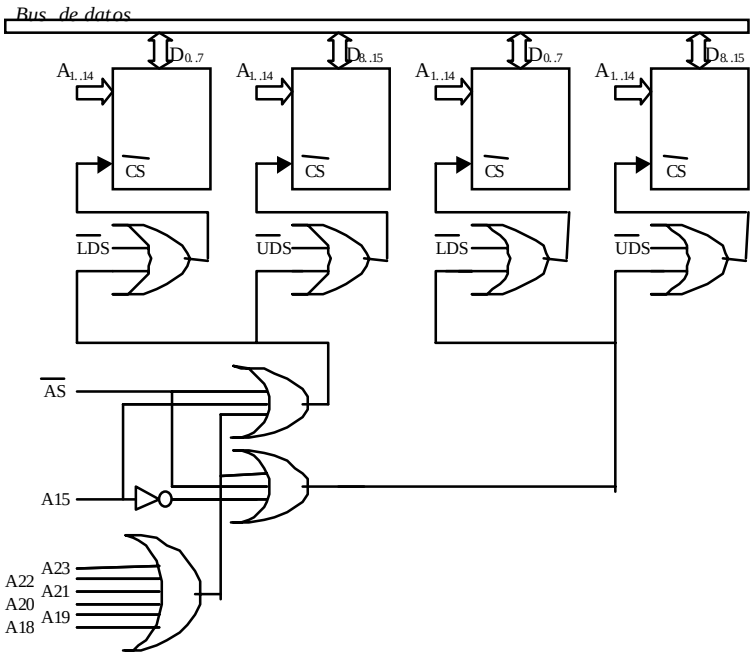
```
;RUTINA CUENTA_CHAR
;FUNCION: recoge en una tabla de 256 posiciones el número de veces que aparece
;         cada carácter en un bloque de memoria
;ENTRADA: A0(L) dirección del bloque a analizar
;         D0(W) número de bytes a analizar
;         A1(L) dirección de la tabla de resultados
;SALIDA: el contenido de la tabla
;MODIFICA: no altera ningún registro.
```

Escriba una rutina en ensamblador del µP68000 que, dada la tabla obtenida con la función anterior, proporcione diversos valores que se especifican en el siguiente prototipo:

```
;RUTINA CUENTA_GRUPOS
;FUNCION: dada una tabla de cuentas de aparición de caracteres devuelve
;ENTRADA: A1(L) dirección de la tabla de resultados
;SALIDA: D0(0..15)  - número de caracteres ASCII (<128)
;        D0(16..31) - número de caracteres no ASCII (>127)
;        D1(0..15)  - número de caracteres alfanuméricos (letras+dígitos)
;        D1(16..31) - número de caracteres no alfanuméricos
;        D2(0..15)  - número de dígitos
;        D2(16..31) - número de letras
;        D3(0..15)  - número de letras minusculas
;        D3(16..31) - número de letras mayusculas
;
;MODIFICA: sólo los registros de salida D0-D3
```

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0 ...	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
10 ...	LF	VT	FF/NP	CR	SO	SI	DLE	DC1	DC2	DC3
20 ...	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
30 ...	RS	US	blanco	!	"	#	$	%	&	'
40 ...	(	)	*	+	,	-	.	/	0	1
50 ...	2	3	4	5	6	7	8	9	:	;
60 ...	<	=	>	?	@	A	B	C	D	E
70 ...	F	G	H	I	J	K	L	M	N	O
80 ...	P	Q	R	S	T	U	V	W	X	Y
90 ...	Z	[	\	]	^	_	`	a	b	c
100 ...	d	e	f	g	h	i	j	k	l	m
110 ...	n	o	p	q	r	s	t	u	v	w
120 ...	x	y	z	{	|	}	~	del		

6 Dibuje el mapa de memoria resultante del esquema de la figura.



Examen de S.E.T.I.

1er curso de Ingeniería Electrónica
26 de junio de 2001

SOLUCIONES a las cuestiones prácticas

3

a) 0.0

0	0	0	0	0	0	0	0	0	0

$$(-1)^{signo}\ 2^{-bias}\ mantisa = (-1)^0\ 2^{-7}\ 0 = 0.0$$

b) 0.125

0	0	1	0	0	0	0	0	0	0

$$(-1)^{signo}\ 2^{exponente-bias}\ (1+mantisa) = (-1)^0\ 2^{4-7}\ (1+0) = 2^{-3} = 0.128$$

c) 3,14159265358979323846264338327 95

no se puede representar de modo exacto con la precisión de que se dispone. La cantidad más próxima es lo siguiente:

0	1	0	0	0	1	0	0	1	0

$$(-1)^{signo}\ 2^{exponente-bias}\ (1+mantisa) = (-1)^0\ 2^{8-7}\ (1+1/2+1/16) = 2\ (1+0.5+0.0625) = 3.125$$

esto es inferior al valor que se pretendia representar.

Comprobemos que el siguiente valor representable no es más próximo al indicado:

$$(-1)^0\ 2^{8-7}\ (1+1/2+1/16+1/32) = 2\ (1+0.5+0.0625+0.03125) = 3.1875$$

4

Estado	Carácter	Carácter	movimiento	estado	Comentario
0	*	id.	I	0	Se situa en el primer carácter de la izquierda.
0	#	#	D	1	
1	1	1	D	1	Localiza el "0" más a la izquierda. Si no hay, todo son "0" y para
1	0	0	D	2	
1	#	#	I	H	
2	0	0	D	2	Localiza el primer "1" fuera de lugar (hay "0"s a la izquierda) y pone "0". Si no hay, ha terminado
2	1	0	I	3	
2	#	#	I	H	
3	0	0	I	3	retrocede por la secuencia de "0"s hasta el primero
3	*	id.	D	4	
4	0	1	D	1	Pone "1" en lugar del primer "0" y vuelve a buscar un "1" fuera de lugar.

notas:

- El estado inicial es el 0.
- La máquina puede estar inicialmente sobre uno cualquiera de los caracteres a ordenar. No se contempla el caso de que se "engañe" a la máquina poniendola a funcionar sobre un caracter en blanco o sobre una cadena de simbolos con alguno distinto de "0" o "1".

- "*" significa *"cualquier carácter no contemplado por una regla en el estado dado"*. "id." significa *"escribir el mismo carácter leido"*. Esta notación se utiliza por comodidad pero no es admitida por el formalismo de la máquina de Turing. Para ajustarse al formalismo es adecuado reemplazar cada una de estas reglas por tantas como sean precisas sustituyendo "*" por cada uno de los caracteres no especificados en el estado correspondiente, e "id." por el mismo carácter en cada caso (pueden surgir así reglas innecesarias, pero ello no afecta al funcionamiento de la máquina).

5

El enunciado no especifica el tamaño de los contadores en la tabla. En esta solución consideraremos que los contadores son words, y por tanto el desplazamiento en la tabla para un determinado caracter es el doble de su valor ASCII.
No se dice si la tabla esta inicialmente a cero o no. Es razonable pensar que esta rutina no debe preocuparse de ello. Si al llamarla la tabla contiene ciertos valores distintos de cero las cuentas se incrementarán a partir de ellos, lo que puede ser util para contar caracteres en más de una zona de memoria.

```
;RUTINA CUENTA_CHAR
;FUNCION: recoge en una tabla de 256 posiciones el número de veces que aparece
;         cada carácter en un bloque de memoria
;ENTRADA: A0(L) dirección del bloque a analizar
;         D0(W) número de bytes a analizar
;         A1(L) dirección de la tabla de resultados
;SALIDA: el contenido de la tabla
;MODIFICA: no altera ningún registro.
;
CUENTA_CHAR: MOVEM.W D0/D1,-(SP) ;CONSERVA VALORES DE REGISTROS A ALTERAR
             BRA.S   CICLO_0     ;                    Y COMIENZA EL CICLO
CONTINUA_0:  CLR.W   D1          ;EN TAMAÑO WORD
             MOVE.B  (A0)+,D1    ;COGE CADA CARACTER
             LSL.W   #1,D1       ;Y EL DESPLAZAMIENTO ES EL DOBLE DE SU VALOR
             ADDQ.W  #1,0(A1,D1) ;INCREMENTA EL CONTADOR ADECUADO
CICLO_0:     DBRA.S  D0,CONTINUA_0
             MOVEM.W (SP)+,D0/D1 ;RESTABLECE EL VALOR DE LOS REGS. ALTERADOS
             RTS
```

Para resolver la cuenta de grupos de caracteres nos valdremos de una subrutina capaz de llevar a cabo la suma de un rango de contadores dentro de la tabla.

```
;FUNCIÓN SUMA_RANGO
;            SUMA TODOS LOS VALORES QUE SE ENCUENTRAN DENTRO DE UN RANGO DE
;            INDICES EN UNA TABLA DE WORDS
;ENTRADA:    A1(L)   DIRECCION DE LA TABLA
;            D5(W)   INDICE INICIAL
;            D6(W)   INDICE FINAL
;SALIDA:     D7(W)   LA SUMA TOTAL
;MODIFICA:   NADA (SOLO D7.W)
;OBSERVACIONES:      TIENE LA LIMITACIÓN EN LA SUMA FINAL DADA POR EL TAMAÑO
;                    WORD USADO EN D7
SUMA_RANGO:  MOVEM.L A2/D5/D6,-(SP)
             CLR.W   D7
             SUB.W   D5,D6      ;NUMERO DE ELEMENTOS A SUMAR MENOS 1
             LSL.W   #1,D5      ;APUNTA AL PRIMER ELEMENTO A SUMAR
             LEA     (A1,D5),A2
CONTINUA_1:  ADD.W   (A2)+,D7
             DBRA.S  D6,COTINUA_1
             MOVEM.L (SP)+,A2/D5/D6
             RTS
```

El uso en "CUENTA_GRUPOS" de la rutina "SUMA_RANGO" va a ser muy repetitivo, consistiendo siempre en dos lineas para cargar los registros D5 y D6 con los valores inmediatos que determinan el rango de los contadores a sumar, la llamada a la rutina, y la copia del resultado (D7) en un registro diferente. Por ello será adecuado escribir una macro que nos clarifique el código de la rutina.

```
RANGO       MACRO
            MOVE.W  #\1,D5     ;INDICE BAJO DEL RANGO
            MOVE.W  #\2,D6     ;INDICE ALTO DEL RANGO
            BSR     SUMA_RANGO ;HACE LA SUMA
            MOVE.W  D7,\3      ;Y LA LLEVA AL DESTINO
            ENDM

;RUTINA CUENTA_GRUPOS
;FUNCION: dada una tabla de cuentas de aparición de caracteres devuelve
;ENTRADA: A1(L) dirección de la tabla de resultados
;SALIDA: D0(0..15)  - número de caracteres ASCII (<128)
;        D0(16..31) - número de caracteres no ASCII (>127)
;        D1(0..15)  - número de caracteres alfanuméricos (letras+dígitos)
;        D1(16..31) - número de caracteres no alfanuméricos
;        D2(0..15)  - número de dígitos
;        D2(16..31) - número de letras
;        D3(0..15)  - número de letras minusculas
;        D3(16..31) - número de letras mayusculas
;
;MODIFICA: sólo los registros de salida D0-D3
;
CUENTA_GRUPOS:
            RANGO   128,255,D0 ;LA CUENTA DE NO ASCII'S (FUERA DE LUGAR)
            MOVE.W  D0,D1      ;A SU VEZ ES PARCIAL DEL NUMERO TOTAL
            SWAP    D0         ;LOS NO ASCII'S AHORA EN SU LUGAR

            RANGO   0,127,D0   ;LA CUENTA DE ASCII'S EN SU LUGAR
            ADD.W   D7,D1      ;NUMERO TOTAL DE CARACTERES
            SWAP    D1         ;QUE QUEDA PARA EL  FINAL EN D1(16..31)

            RANGO   65,90,D3   ;LA CUENTA DE MAYUSCULAS (FUERA DE LUGAR)
            SWAP    D3         ;Y AHORA EN SU SITIO
            MOVE.W  D7,D2      ;TAMBIEN ES PARCIAL PARA LETRAS

            RANGO   97,122,D3  ;LA CUENTA DE MINUSCULAS EN SU SITIO
            ADD.W   D7,D2      ;CUENTA DE LETRAS (FUERA DE LUGAR)
            MOVE.W  D2,D1      ;Y CUENTA PARCIAL PARA CARACTERES ALFANUMERICOS
            SWAP    D2         ;CUENTA DE LETRAS EN SU SITIO

            RANGO   48,57,D2   ;LA CUENTA DE DIGITOS EN SU SITIO
            ADD.W   D7,D1      ;CUENTA DE ALFANUMERICOS EN SU SITIO

            MOVE.W  D1,D7      ;Y COPIA EN D7
            SWAP    D1         ;TOMA LA CUENTA TOTAL (ALFANUM. FUERA DE LUGAR)
            SUB.W   D7,D1      ;Y OBTIENE LOS NO ALFANUM. (FUERA DE LUGAR)
            SWAP    D1         ;PONE D1 CORRECTAMENTE
            RTS
```

En caso de no utilizar una macro debe escribirse directamente el código como resulta de la expansión de la misma: (se han cambiado los comentarios que generaría la expansión de la macro)

```
;RUTINA CUENTA_GRUPOS
;FUNCION: dada una tabla de cuentas de aparición de caracteres devuelve
;ENTRADA: A1(L) dirección de la tabla de resultados
;SALIDA: D0(0..15)  - número de caracteres ASCII (<128)
;        D0(16..31) - número de caracteres no ASCII (>127)
;        D1(0..15)  - número de caracteres alfanuméricos (letras+dígitos)
```

```
;        D1(16..31) - número de caracteres no alfanuméricos
;        D2(0..15)  - número de dígitos
;        D2(16..31) - número de letras
;        D3(0..15)  - número de letras minusculas
;        D3(16..31) - número de letras mayusculas
;
;MODIFICA: sólo los registros de salida D0-D3

CUENTA_GRUPOS:
             MOVE.W  #128,D5    ;CUENTA NO ASCII'S
             MOVE.W  #255,D6
             BSR     SUMA_RANGO
             MOVE.W  D7,D0      ;LA CUENTA DE NO ASCII'S (FUERA DE LUGAR)
             MOVE.W  D0,D1      ;A SU VEZ ES PARCIAL DEL NUMERO TOTAL
             SWAP    D0         ;LOS NO ASCII'S AHORA EN SU LUGAR

             MOVE.W  #0,D5      ;CUENTA ASCII'S
             MOVE.W  #127,D6
             BSR     SUMA_RANGO
             MOVE.W  D7,D0      ;LA CUENTA DE ASCII'S EN SU LUGAR
             ADD.W   D7,D1      ;NUMERO TOTAL DE CARACTERES
             SWAP    D1         ;QUE QUEDA PARA EL  FINAL EN D1(16..31)

             MOVE.W  #65,D5     ;CUENTA MAYUSCULAS
             MOVE.W  #90,D6
             BSR     SUMA_RANGO
             MOVE.W  D7,D3      ;LA CUENTA DE MAYUSCULAS FUERA DE LUGAR
             SWAP    D3         ;Y AHORA EN SU SITIO
             MOVE.W  D7,D2      ;TAMBIEN ES PARCIAL PARA LETRAS

             MOVE.W  #97,D5     ;CUENTA MINUSCULAS
             MOVE.W  #122,D6
             BSR     SUMA_RANGO
             MOVE.W  D7,D3      ;LA CUENTA DE MINUSCULAS EN SU SITIO
             ADD.W   D7,D2      ;CUENTA DE LETRAS (FUERA DE LUGAR)
             MOVE.W  D2,D1      ;Y CUENTA PARCIAL PARA CARACTERES ALFANUMERICOS
             SWAP    D2         ;CUENTA DE LETRAS EN SU SITIO

             MOVE.W  #48,D5     ;CUENTA DIGITOS
             MOVE.W  #57,D6
             BSR     SUMA_RANGO
             MOVE.W  D7,D2      ;LA CUENTA DE DIGITOS EN SU SITIO
             ADD.W   D7,D1      ;CUENTA DE ALFANUMERICOS EN SU SITIO

             MOVE.W  D1,D7      ;Y COPIA EN D7
             SWAP    D1         ;TOMA LA CUENTA TOTAL (ALFANUM. FUERA DE LUGAR)
             SUB.W   D7,D1      ;Y OBTIENE LOS NO ALFANUM. (FUERA DE LUGAR)
             SWAP    D1         ;PONE D1 CORRECTAMENTE
             RTS
```

6 Las líneas A_{18} a A_{23} deberán estar a cero para seleccionar alguna memoria (La zona gris de la primera parte de la figura no puede corresponderse con las memorias).

Las líneas A_{16} y A_{17} no están utilizadas, por lo que cualquiera de las cuatro combinaciones posibles para los valores de estas, dan acceso a los mismos dispositivos físicos (lo que denominamos copias). Esto se muestra en la zona central de la figura.

La línea A_{15} distingue entre el par de la derecha y el de la izquierda. Si el resto de señales es adecuado, cuando A_{15} es cero, se selecciona el par de memorias de la izquierda (numeradas como 1 y 2. en la figura). Cuando A_{15} es uno sucede lo mismo para el par de memorias de la derecha (numeradas como 3 y 4. en la figura).

Las líneas $A_1..A_{14}$ entran en las pastillas, luego cada una tiene 2^{14} posiciones direccionables. Como su tamaño de datos es de 8 bits tenemos que cada una cubre un espacio de $2^4\,2^{10}$ bytes =16 Kbytes. Las dos de la izquierda comparten direcciones, "mapeandose" cada una en la parte par o impar en función de las señales `LDS` y `UDS` (la de la izquierda(1) para direcciones impares —lower— y la de la derecha(2) para direcciones pares —upper—). Lo mismo cabe decir de las dos de la derecha.