eman ta zabal zazu
universidad euskal herriko
del país vasco unibertsitatea

departamento de electricidad y electrónica
elektrika eta elektronika saila

Examen de S.E.T.I.

1er curso de Ingeniería Electrónica
26 de enero de 2007

Ejercicio de C. **PARTE 1** 1,5 puntos

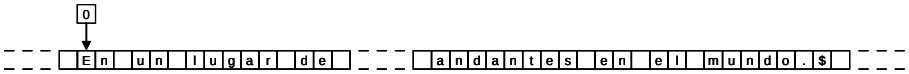
1 Escribir una función que elimine de una cadena *s1* cualquier carácter que aparezca en otra cadena *s2*. Nótese que no se trata de sustituir tales caracteres por blancos, sino de eliminarlos. El algoritmo que se utilizará será el siguiente:

Reservar dinámicamente espacio suficiente para almacenar una cadena *stmp* de la misma longitud que *s1*.
Para cada carácter de *s1*:
    Comprobar si el carácter está en *s2*
    Si no está, añadir el carácter a *stmp*
Copiar *stmp* a *s1*
Liberar el espacio reservado para *stmp*

Los parámetros de la función deben ser los apuntadores a sendas cadenas *s1* y *s2*. Su valor de retorno será el número de caracteres eliminados de *s1*.

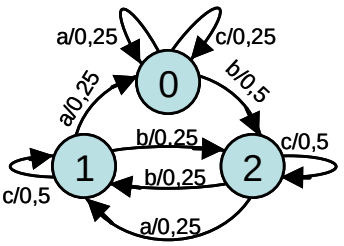
Cuestiones. **PARTE 2** 1 + 1 puntos

2 Si sabe usted en qué consiste el funcionamiento de la Máquina de Turing sabrá explicar qué es capaz de hacer la que se muestra en la figura. No se trata de explicar lo que hace paso a paso, sino de decir en una frase cuál es su función.



S	R	W	M	S
0	n	n	R	1
0	$	$	L	h
0	other	same	R	0
1	p	p	L	2
1	b	b	L	2
1	other	same	R	0
1	$	$	L	h
2	other	m	R	0

3 Obtener la entropía de la fuente de la figura.
(Se trata de una fuente con vocabulario {a,b,c} en la que los estados no coinciden con el último símbolo emitido, cosa que no tiene trascendencia alguna en el cálculo de la entropía)
(Advertencia: no se deje intimidar por la complejidad a primera vista de la estructura y obsérvela cuidadosamente antes de calcular)



Estructura de un microprocesador. 1,5 puntos

4 a) Nuestro computador virtual dispone de un sumador y un restador en su ALU. Como sabemos, la notación en complemento a dos permite que estos circuitos puedan ser usados tanto para el caso en que usemos enteros con signo como sin él. No obstante, el bit FLAG sólo refleja un posible rebose (carry) en caso de trabajar sin signo. ¿Qué podemos hacer para controlar un posible rebose (overflow) en notación con signo? (Piense tanto en hardware como en software).
[overflow(A+B=R): $A_{15}=B_{15}=1$ $R_{15}=0$ || $A_{15}=B_{15}=0$ $R_{15}=1$] [overflow(A-B=R): $A_{15}=1$ $B_{15}=R_{15}=0$ || $A_{15}=0$ $B_{15}=R_{15}=1$]

b) Si disponemos de esa posibilidad en hardware deberíamos disponer de una instrucción de salto condicional (sea JMV – JuMp on oVerflow). Para no eliminar ninguna instrucción podemos reordenar los códigos del siguiente modo:

Operación	Dir. directo	Dir. inmediato	Dir. indirecto	Operación	No ext. Dir.
Sumar al AC	000000 0 ADD	010000 16 ADD-C	100000 32 ADD-I		
Restar del DC	000001 1 SUB	010001 17 SUB-C	100001 33 SUB-I		
Y lógico con el AC	000010 2 AND	010010 18 AND-C	100010 34 AND-I		
O lógico con el AC	000011 3 OR	010011 19 OR-C	100011 35 OR-I		
				NO lógico del AC	110100 52 NOT
				Desplazar a izquierda el AC	110101 53 SHL
				Desplazar a derecha el AC	110110 54 SHR
				Incrementar el AC	110111 55 INC
				Decrementar el AC	111000 56 DEC
Cargar AC de memoria	001001 9 LOD	011001 25 LOD-C	101001 41 LOD-I		
Descargar AC en memoria	001010 10 STO		101010 42 STO-I		
Saltar si overflow	**001011 11 JMV**			Parar	111011 59 HLT
Saltar	001100 12 JMP		101100 44 JMP-I		
Saltar si AC=0	001101 13 JMZ		101101 45 JMZ-I		
Saltar si AC<0	001110 14 JMN		101110 46 JMN-I		
Saltar si FLAG activado	001111 15 JMF		101111 47 JMF-I		

¿Qué implica esto para el hardware de nuestra máquina?.

Mapeo de memoria. 1 punto

5 Diseñe el circuito necesario para que un computador con microprocesador 68000 disponga de:

a) 1 Mbyte de ROM situado a partir de 0 mediante 2 pastillas estructuradas en bytes.
b) 2 pastillas de RAM de 8 Mbits estructuradas en bytes en el final del espacio direccionable.

No se impone ninguna restricción sobre la posibilidad de que existan copias de las pastillas en otras posiciones del espacio direccionable siempre y cuando queden al menos 4MBytes libres y, claro está, que los distintos circuitos no se solapen. (muestre el mapa resultante)

Programación del 68000. **PARTE 3** 4 puntos (1 + 1 + 1 + 1)

6 Programación del 68000. Ordenación por Selección

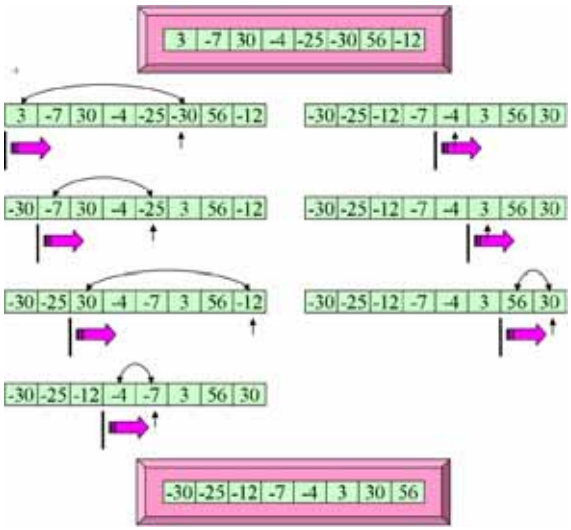
El algoritmo de Ordenación por Selección es un sencillo pero no muy eficaz algoritmo de ordenación. Su funcionamiento es el siguiente:

- Buscar el mínimo elemento de la lista
- Intercambiarlo con el primero
- Buscar el mínimo en el resto de la lista
- Intercambiarlo con el segundo

Y en general:

- Buscar el mínimo elemento entre una posición i y el final de la lista
- Intercambiar el mínimo con el elemento de la posición i

En la imagen puede verse un ejemplo práctico:



Se pide que se escriban las siguientes subrutinas y programa principal:

Subrutina ESCRIBE

Esta subrutina tomará la dirección de un vector de enteros de 16 bits con signo y su longitud, y escribirá por pantalla el contenido del mismo, enseñando un número por línea. Para escribir números por pantalla, contaremos con la llamada de sistema (Trap #15; DC.W 5), que escribe por pantalla el número de 32 bits con signo contenido en D0. Para escribir caracteres por pantalla, contaremos con la llamada de sistema (Trap #15; DC.W 1), que escribe por pantalla el carácter contenido en D0. Por último, comentar que para generar un salto al inicio de la siguiente línea es necesario escribir los caracteres 10 y 13.

Subrutina MINIMO

Encuentra la posición del elemento mínimo de un vector de enteros (16 bits, con signo).

Subrutina ORDENA

Realiza la ordenación **ascendente** de un vector de enteros (16 bits, con signo) mediante el algoritmo de selección.

Las descripciones ***podrían*** ser las de la derecha:

```
; Subrutina:    ESCRIBE
; Descripción:  Escribe por pantalla el
;               contenido de un vector de
;               enteros de 16 bits con signo.
; Entrada:      A0.L  dirección del vector
;               D0.W  tamaño
; Modifica:     CCR…
```

```
; Subrutina:    MINIMO
; Descripción:  Localiza el elemento mínimo
;               de un vector.
; Entrada:      A0.L  dirección del vector
;               D0.W tamaño
; Salida:       D1.W índice del elemento
;               mínimo
; Modifica:     CCR…
```

```
; Subrutina:    ORDENA
; Descripción:  Ordena un vector.
; Entrada:      A0.L  dirección del vector
;               D0.W tamaño
; Modifica:     CCR…
```

Programa principal

El programa principal deberá tomar un vector de enteros **(16 bits, con signo)** desordenado, ordenarlo y enseñar el resultado por pantalla.

eman ta zabal zazu
universidad euskal herriko
del país vasco unibertsitatea

departamento de electricidad y electrónica
elektrika eta elektronika saila

Examen de S.E.T.I.

1er curso de Ingeniería Electrónica
26 de enero de 2007

SOLUCIONES Y COMENTARIOS

1

```
#include <string.h>
#include <stdlib.h>

int strdel(char *s1, char *s2)
{
  char *stmp, *p1, *p2, *pt;
  int ndel;

  pt = stmp = malloc(strlen(s1)+1);
  for (p1 = s1, ndel= 0; *p1; p1++) {
    for (p2 = s2; *p2 && *p1 != *p2; p2++)
      ;
    if (*p2)
      ndel++;
    else
      *pt++ = *p1;
  }
  *pt = '\0';
  strcpy(s1, stmp);
  free(stmp);
  return ndel;
}
```

2

La máquina recorre un texto de izquierda a derecha, hasta encontrar un finalizador '$', cambiando 'n' por 'm' en caso de encontrar errores 'np' o 'nb' (p.ej. si pone 'canbiar' lo corrige dejando 'cambiar').

Nota: por no complicar el ejercicio la máquina no corrige errores en que intervengan mayúsculas ni contempla la posibilidad de los cambios 'nnp'→'nmp' y 'nnb'→'nmb' (en un texto correcto no deben presentarse estas situaciones, pero el criterio de 'robustez' aconsejaría tener en cuenta estas posibilidades)

3

El modelo tiene tres estados idénticos en su estructura frecuencial: dos transiciones con probabilidad 0,25 y una con 0,5. En consecuencia la entropía de todos los estados es la misma:

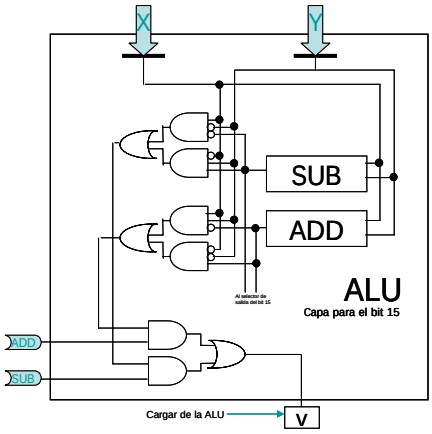
$$0{,}25\times\log_2(1/0{,}25)+0{,}25\times\log_2(1/0{,}25)+0{,}5\times\log_2(1/0{,}5) = 2\times0{,}25\times\log_2(4)+0{,}5\times\log_2(2)= 1{,}5 \text{ bits}$$

Independientemente de la probabilidad de cada estado, al hacer la media ponderada de tres entropías iguales el resultado será el mismo: 1,5 bits

Nota: obsérvese que las probabilidades los tres estados no son idénticas

4

a) Bastará con implantar los circuitos combinacionales correspondientes en la ALU para generar el nuevo "flag" de rebose (V) (véase la figura).



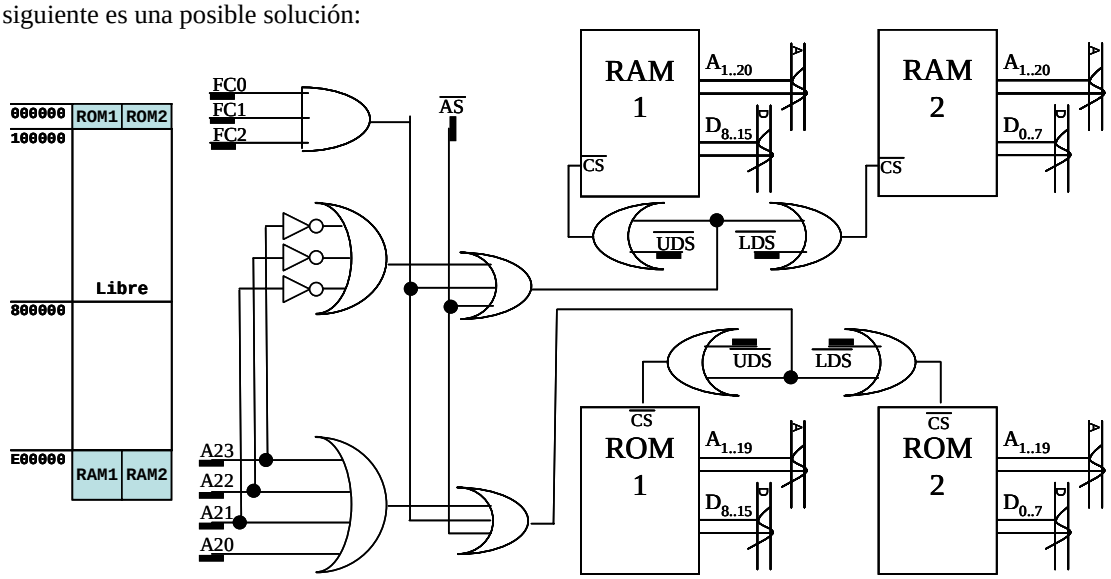
b) Para la función combinacional no era preciso tener en cuenta los dos bits del modo de direccionamiento en el caso de las instrucciones que se han cambiado. Ahora esto sigue siendo cierto para todas menos para la HLT, ya que se solapa con nuestra nueva instrucción JMV.
El demultiplexador de direccionamiento deberá dar las cuatro salidas y para el código de instrucción 1011 se ejecutará JMV si el direccionamiento es directo y HLT si es "sin direccionamiento".

JVM: 4. Si V=1 Cargar PC desde el IR
5. Poner COUNT a cero

Nota: por no complicar el ejercicio no se ha incluido JVM-I, que sería lo razonable.

5

La siguiente es una posible solución:



6

```
; ****************
; PROGRAMA PRINCIPAL
; ****************
        org     $1000

        lea     VECTOR,a0
        move.w  N,d0
        bsr.s   ORDENA
        bsr.s   ESCRIBE
        trap    #15
        dc.w    0

; ****************
; SUBRUTINAS
; ****************

; Subrutina:     MINIMO
; Descripción: Localiza el elemento mínimo
;   de un vector.
; Entrada: A0.L  dirección del vector
;          D0.W tamaño
; Salida:  D1.W índice del elemento mínimo
; Modifica: CCR

MINIMO  movem.l  a0/d0/d2/d3,-(a7)
        clr.w    d1 ; d1: indice del minimo
        move.w   (a0)+,d2 ; d2: valor del minimo
        move.w   #1,d3  ; d3:indice
        subq.w   #1,d0  ; d0:num. de ciclos
        bra.s    for1
do1     cmp.w    (a0)+,d2
        blt.s    seguir1
        move.w   d3,d1
        move.w   -2(a0),d2
seguir1 addq.w   #1,d3
for1    dbra     d0,do1
        movem.l  (a7)+,a0/d0/d2/d3
        rts

; Subrutina:     ORDENA
; Descripción: Ordena un vector.
; Entrada: A0.L  dirección del vector
;          D0.W tamaño
; Modifica: CCR

ORDENA  movem.l  a0/d0-d3,-(a7)
        move.w   d0,d2
        subq.w   #1,d2 ;d2: num de iteraciones
        bra.s    for2
do2     bsr.s    MINIMO
        add.w    d1,d1
        move.w   (a0,d1.w),d3
        move.w   (a0),(a0,d1.w)
        move.w   d3,(a0)+
        subq.w   #1,d0
for2    dbra     d2,do2
        movem.l  (a7)+,a0/d0-d3
        rts

; Subrutina:     ESCRIBE
; Descripción: Escribe por pantalla
;   el contenido de un vector de
;   enteros de 16 bits sin signo.
; Entrada: A0.L  dirección del vector
;          D0.W  tamaño
; Modifica: CCR

ESCRIBE movem.l  a0/d0/d1,-(a7)
        move.w   d0,d1
        bra.s    for3
do3     move.w   (a0)+,d0
        ext.l    d0
        trap     #15
        dc.w     5
        move.w   #10,d0
        trap     #15
        dc.w     1
        move.w   #13,d0
        trap     #15
        dc.w     1
for3    dbra     d1,do3
        movem.l  (a7)+,a0/d0/d1
        rts
; ****************
; DATOS
; ****************
        org     $2000

N       dc.w    11

VECTOR  dc.w    67
        dc.w    -3
        dc.w    7
        dc.w    -7
        dc.w    -93
        dc.w    9
        dc.w    0
        dc.w    15
        dc.w    123
        dc.w    -1
        dc.w    20
```