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

13 de septiembre de 2007

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

1 Escríbase una función que realice la conversión de un número entero sin signo (el cual será un parámetro, de tipo unsigned long int), a su representación como cadena de caracteres, expresada en cualquier base entre 2 y 36 (la cual será el otro parámetro). Para representar los dígitos con valor mayor que 10 utilícense las letras mayúsculas (A=10, B=11, etc.) de modo similar al utilizado para hexadecimal, pero en este caso pudiendo utilizarse hasta Z=35.

Utilícese el sencillo algoritmo de divisiones sucesivas entre la base (tras cada división, los restos nos van proporcionando los dígitos de menor a mayor peso; el cociente se emplea como dividendo en la siguiente iteración; el proceso termina cuando el cociente resultante es cero). Recuérdese que C cuenta con el operador módulo (%).

Cada uno de los dígitos obtenidos con las divisiones sucesivas ha de convertirse a un carácter ('0','1','2',...'A','B', 'C',...) y almacenarse en un array.

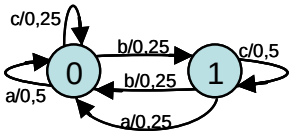
Una vez terminado el proceso, es necesario formar una cadena con esos dígitos, copiándolos en orden inverso y añadiendo el cero de terminación de cadena al final. La dirección de comienzo de esta cadena será el valor de retorno de la función. Además, el espacio de memoria para almacenar esa cadena ha de ser reservado dinámicamente dentro de la propia función.

Nótese también que, dado el tipo del número a convertir de base, el máximo número de dígitos (y por tanto caracteres) que requerirá su representación será de 32.

Cuestiones.	PARTE 2	1 + 1 puntos

2 En una representación numérica de reales a 8 bits como se ha visto en clase (seeemmmm) ¿cómo se representan: a) el cero, b) el 3.14159. c) el 0,3125 ?

3 Determinar la probabilidad de los estados de la fuente de la figura, así como la entropía de la misma. La probabilidad de que esta fuente emita en un momento dado la secuencia "abc" es de 5/96; justifíquelo.



Estructura de un microprocesador.	1,5 puntos

4 A la luz de su conocimiento del procesador virtual y del 68000, explique de la forma más sintética posible el funcionamiento de los programas como lo haría a alguien que, sabiendo programar en un lenguaje de alto nivel, desconoce todo sobre hardware.

Mapeo de memoria.	1 punto

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

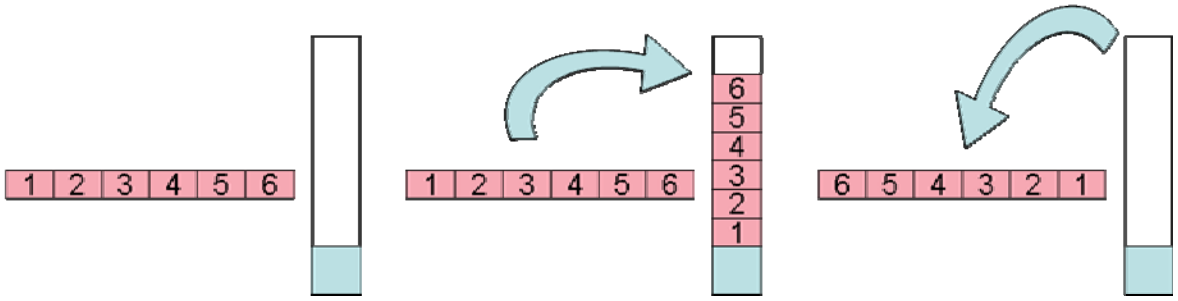
a) 1 Mbyte de ROM situado a partir de 0 mediante pastillas estructuradas en bits.

b) 4 pastillas de RAM de 2 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 Una PILA (nos referimos a la estructura de datos) puede utilizarse para invertir el orden de los elementos que vamos introduciendo en ella, ya que la extracción se realiza en orden inverso:



Subrutina INV_CADENA

Invierte el orden de una cadena de caracteres

Subrutina INV_VECTOR

Invierte el orden de un vector de enteros de 16 bits. El vector tiene una cabecera de 16 bits (entero sin signo) que contiene la longitud.

Subrutina INV_AGENDA

Invierte el orden de una agenda telefónica. La agenda está compuesta por registros que contienen un número telefónico (entero de 32 bits) y la dirección (long word) de una cadena de caracteres (nombre/apellido). El último registro de la agenda contiene el número telefónico 0.

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

```
; Subrutina:     INV_CADENA
; Descripción:   Invierte el contenido de una
                 cadena de caracteres.
; Entrada:       A0.L  Dirección de la
                 cadena
; Modifica:      …
```

```
; Subrutina:     INV_VECTOR
; Descripción:   Invierte el contenido de un
                 vector de enteros de 16 bits
; Entrada:       A0.L Dirección del vector
; Modifica:      …
```

```
; Subrutina:     INV_AGENDA
; Descripción:   Invierte el contenido de una
                 agenda telefónica
; Entrada:       A0.L Dirección de la agenda
; Modifica:      …
```

Programa principal

El programa principal deberá tomar una cadena de caracteres, un vector de enteros y una agenda telefónica, e invertir el orden de todos ellos.

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

13 de septiembre de 2007

SOLUCIONES Y COMENTARIOS

1

```
/* Todos los accesos secuenciales se han implementado con apuntadores de forma
 * ortodoxa, aunque también se podría haber usado indexación.
 *     El array "digito" se utiliza para realizar la conversión a caracteres de
 * forma directa y sencilla. Obviamente hay otras maneras de hacerlo.
 */
      #include <stdlib.h>

      char *convbase(unsigned long num, unsigned int base)
      {
            static char digito[]="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
            static char almacen[32];
            char *numstr, *pa, *pn;
            int ndig;

            pa = almacen;

            do {
                  *pa++ = digito[num % base];
                  num /= base;
            } while (num > 0);

            ndig = pa - almacen;
            numstr = malloc(ndig+1);

            for (pn = numstr; ndig > 0; ndig--)
                  *pn++ = *--pa;

            *pn = '\0';

            return numstr;
      }
```

2

a) el cero.

El entorno del cero, incluido este, se representa con el exponente mínimo **(000)**. La mantisa se multiplica en este caso directamente con la potencia de dos correspondiente **(2^(1-3))**, por lo que basta con que esta sea cero **(0000)**. El signo puede ser tanto positivo como negativo.

0	0	0	0	0	0	0	0

1	0	0	0	0	0	0	0

b) el 3.14159

La potencia de dos inmediatamente inferior es 2^1, por lo que el exponente será **100(4-3=1)**. Buscamos la mantisa correspondiente para **2(1+m)=3.14159**, y obtenemos **m=0,570795**. Como representamos la mantisa a 4 bits tenemos valores entre **0** y **15/16** en incrementos de **1/16**. Hemos de determinar el valor entero **i** que más aproxime a **m** la expresión **i/16≅0,570795 (i≅0,570795*16=9,13272)** que resulta ser **i=9 (1001)**.

0	1	0	0	1	0	0	1

c) el 0,3125

La potencia de dos inmediatamente inferior es 2^{-2}, por lo que el exponente será **001(1-3=-2)**. Buscamos la mantisa correspondiente para **(2^{-2})(1+m)=0.3125**, y obtenemos **m=0,25**. Este valor representado en diecisisavos es **4/16**, por lo que la mantisa es **4 (0100)**.

0	0	0	1	0	1	0	0

3 Los dos estados tienen la misma estructura frecuencial: una transición de probabilidad 0,5 y dos de probabilidad 0,25, por lo que la entropía de la fuente será idéntica a la de los estados (es la media ponderada por la probabilidad de cada estado, pero como ambas entropías son iguales, la ponderación no tiene efecto)

$$H=H(S0)=H(S1)=(1/2)\ \log_2 2 + 2\ (1/4)\ \log_2 4 = (1/2) + 1 = (3/2)\ bits$$

Los estados no son equiprobables. La probabilidad de S0 en un momento dado será la suma probabilidades de estar en S0 y transitar sobre sí mismo, más la de estar en S1 y transitar a S0.

$$P(S0)=P(S0)\ 0{,}75 + P(S1)\ 0{,}5$$

por lo que $P(S0) = 2\ P(S1)$, y como $P(S0)+P(S1)=1$, entonces $P(S0)=2/3$ y $P(S1)=1/3$

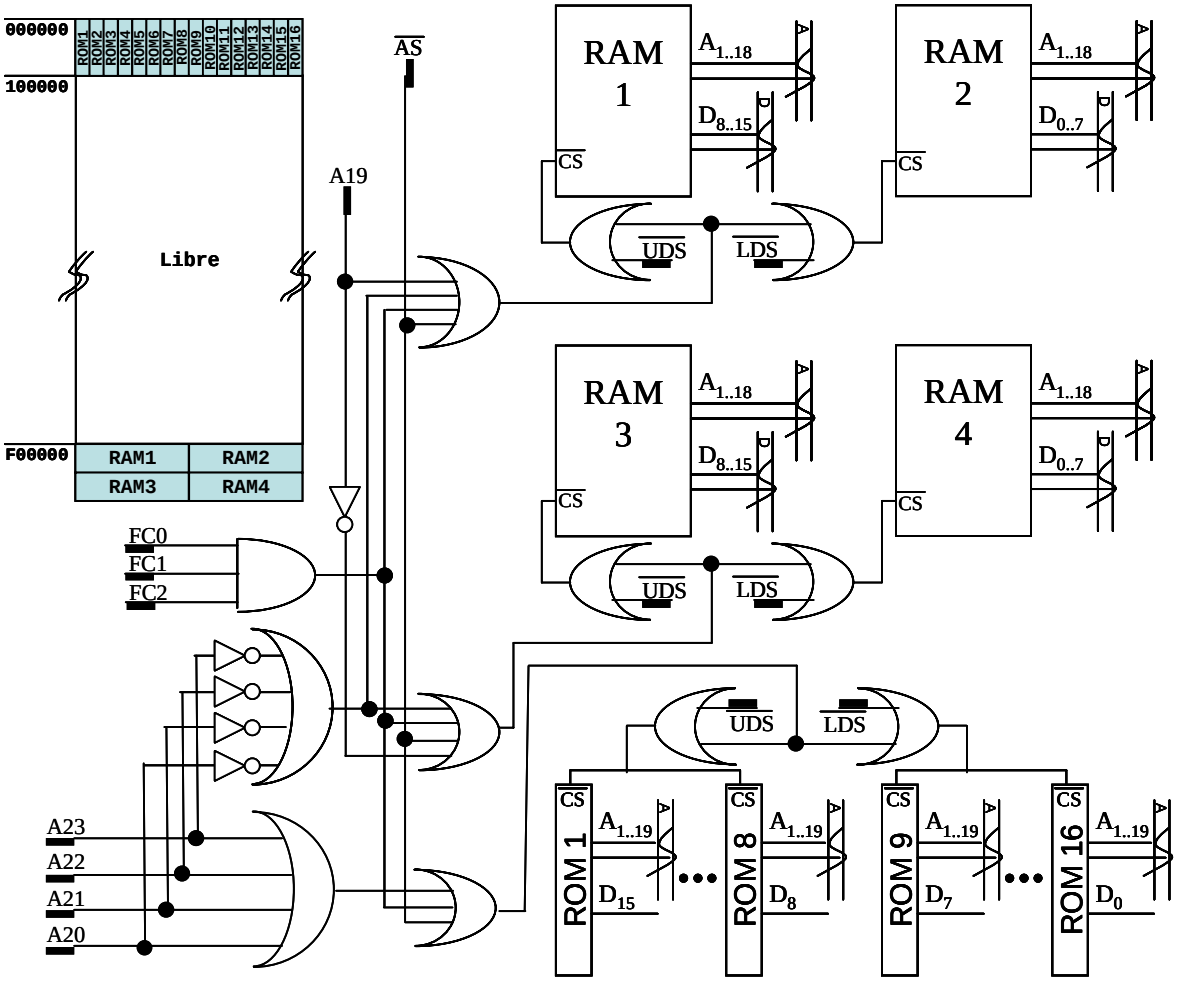
La probabilidad de que la fuente emita "abc" en un momento dado será la suma de la probabilidad de que lo haga estando en S_0 más la de la probabilidad de que lo haga estando en S_1, ponderada por la probabilidad de estar en dichos estados:

$$P(S_0)\ P(abc|S_0) + P(S_1)\ P(abc|S_1) = (2/3)\ 0.5\ 0.25\ 0.5 + (1/3)\ 0.25\ 0.25\ 0.5 = 5/96$$

4 Esta es una pregunta muy abierta en la que el alumno puede mostrar su conocimiento tanto global como de detalle del funcionamiento de un programa. En general es razonable comenzar hablando de "compilación" para llevar la explicación hacia la ejecución de programas por microprocesadores (esto deja de lado muchos aspectos relacionados con sistemas operativos, entradas/salidas, etc. que no se corresponden con lo visto en la asignatura). Una vez situados en la ejecución de programas a nivel de máquina, cabe comentar algo sobre codificación de instrucciones y como dichos códigos determinan la entrada de señales a una función combinacional que, conjuntamente con la salida de un contador, va generando, a ritmo de una señal de reloj, las señales de control necesarias para que los datos fluyan por los registros internos de CPU combinándose adecuadamente. Cabe hablar de cómo las operaciones más complejas se llevan a cabo en circuitos adecuados que conforman lo que se denomina ALU; de cómo el avance de la ejecución es controlado por el PC que facilita el acceso a instrucciones y operandos; de las fases fecth y ejecución; y de cuanto más se considere preciso (direccionamiento de datos, situaciones de excepción, etc)

Nota.- La evaluación de las respuestas a esta pregunta no se ha realizado en función de su "completitud", sino de su "consistencia".

5



6

```
        org             $1000

        LEA             CADENA,A0
        BSR.S           INV_CADENA
        LEA             VECTOR,A0
        BSR.S           INV_VECTOR
        LEA             AGENDA,A0
        BSR.S           INV_AGENDA

        TRAP            #15
        DC.W            0
```

```
; Subrutina:           INV_CADENA
; Descripción:         Invierte el contenido de una cadena de caracteres.
; Entrada:             A0.L  Dirección de la cadena
; Modifica:            CCR
INV_CADENA
        MOVEM.L         A0/A1,-(A7)
        MOVE.L          A0,A1           ; copiamos el inicio de cadena

        BRA.S           WHILE1          ; while (!no final)
DO1     MOVE.B          (A1)+,-(A7)     ;   guardar en la pila
WHILE1  TST.B           (A1)
        BNE.S           DO1

        BRA.S           WHILE2          ; while (!no final)
DO2     MOVE.B          (A7)+,(A0)+     ;   guardar en la cadena
WHILE2  TST.B           (A0)
        BNE.S           DO2

        MOVEM.L         (A7)+,A0/A1
        RTS
```

```
; Subrutina:           INV_VECTOR
; Descripción:         Invierte el contenido de un vector de enteros de 16 bits
; Entrada:             A0.L Dirección del vector
; Modifica:            …
INV_VECTOR
        MOVEM.L         A0/A1/D0/D1,-(A7)
        MOVE.W          (A0)+,D0        ; obtenemos la longitud
        MOVE.W          D0,D1           ; copiamos la longitud
        MOVE.W          A0,A1           ; copiamos la direccion del primer elemento

        BRA.S           FOR3            , volcamos todos los elementos
DO3     MOVE.W          (A1)+,-(A7)     ;   hacia la pila
FOR3    DBRA            D1,DO3

        BRA.S           FOR4            , volcamos todos los elementos
DO4     MOVE.W          (A7)+,(A0)+     ;   desde la pila
FOR4    DBRA            D0,DO4

        MOVEM.L         (A7)+,A0/A1/D0/D1
        RTS
```

```
; Subrutina:          INV_AGENDA
; Descripción:        Invierte el contenido de una agenda telefónica
; Entrada:            A0.L Dirección de la agenda
; Modifica:           …
INV_AGENDA
       MOVEM.L       A0/A1,-(A7)
       MOVE.L        A0,A1           ; copiamos el inicio de la agenda

       BRA.S         WHILE5          ; while (!no final)
DO5    MOVE.L        (A1)+,-(A7)     ;   direccion --> pila (OJO al orden)
       MOVE.L        -8(A1),-(A7)    ;   telefono  --> pila (OJO al orden)
WHILE5 TST.L         (A1)+
       BNE.S         DO5

       BRA.S         WHILE6          ; while (!no final)
DO6    MOVE.L        (A7)+,(A0)+     ;   telefono  --> agenda
       MOVE.L        (A7)+,(A0)+     ;   direccion --> agenda
WHILE6 TST.L         (A0)
       BNE.S         DO6

       MOVEM.L       (A7)+,A0/A1
       RTS




       ORG           $2000

CADENA DC.B          "Esta es la cadena",0
VECTOR DC.W          7
       DC.W          12,78,345,821,53,1,723
AGENDA DC.L          943723451
       DC.L          NOMBRE1
       DC.L          667342538
       DC.L          NOMBRE2
       DC.L          946012938
       DC.L          NOMBRE3
       DC.L          634564237
       DC.L          NOMBRE4
       DC.L          0
       DC.L          0
NOMBRE1              DC.B          "Jose manuel",0
NOMBRE2              DC.B          "casa",0
NOMBRE3              DC.B          "Taller",0
NOMBRE4              DC.B          "David Peña",0
```