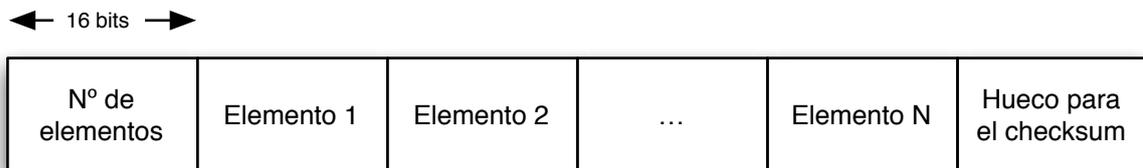




## ARQUITECTURA DE COMPUTADORES

### Recopilación de preguntas de examen de años anteriores válidas para 2018/2019

1. Los ciclos de lectura y escritura del  $\mu P$  68000 finalizan cuando éste detecta la activación de la señal  $\overline{DTACK}$  (*Data Transfer Acknowledge*). ¿Qué debe ocurrir si el 68000 direcciona un dispositivo inexistente y por tanto nunca se va a activar  $\overline{DTACK}$ ?
2. Las instrucciones capaces de alterar la parte alta del SR (el byte de sistema) del 68000 son privilegiadas, es decir, sólo se pueden ejecutar en modo supervisor. ¿Por qué? ¿Cómo se pasa de modo usuario a modo supervisor?
3. ¿Qué diferencia hay entre las instrucciones BHI (*Branch if Higher*) y BGT (*Branch if Greater Than*)?
4. ¿Qué tienen en común las excepciones de *Error de bus* y *Error de dirección*?
5. Escribese una subrutina CHKSUM en lenguaje ensamblador del 68000 que calcule el *checksum* de un vector de words (datos de 16 bits), el cual será añadido al final del vector en el hueco destinado al efecto. Dicho vector estará almacenado en memoria de la siguiente forma:



El único parámetro de entrada de dicha subrutina será la dirección de comienzo del vector, y se pasará a través de la pila.

El checksum se calculará sumando todos los elementos del vector (salvo su cabecera) y cambiando el signo (complemento a dos) de la suma resultante.

6. (2p) Suponga que desea mapear 4 Mbytes de memoria RAM en el espacio de direcciones del 68000 a partir de la dirección \$400000, y que para ello dispone de chips RAM de 1Mx4. Determine cómo deberían estar organizados en el espacio de direcciones y qué rangos de direcciones abarcarían.
7. Explique (ayudándose de un ejemplo gráfico si lo desea) la diferencia entre una CPU *big-endian* y otra *little-endian*. ¿Qué problemas puede suponer esa diferencia en la práctica?
8. Interprete las siguientes sumas suponiendo tanto que se trata de números con signo como sin signo, especificando en cada caso cuál será el estado de las condiciones C, V y N:  
  
0011 0100 0111 1001  
+1001 +0110 +1010 +1100
9. Suponga una CPU que tiene un bus de direcciones de 16 líneas (A0 a A15) y un bus de datos de 8 líneas (D0-D7).
  - (a) Dibuje su espacio de direcciones, con su estructura, rango de direcciones que abarca y capacidad total.
  - (b) Se desea mapear memoria ROM en los primeros 16 Kbytes del espacio de direcciones, y memoria RAM en los últimos 32 Kbytes. Se dispone para ello de chips ROM de 8Kx8 y de chips RAM de 16Kx4. Disponga gráficamente los chips necesarios en el espacio de direcciones, especificando claramente las direcciones en las que comienza y termina cada fila de chips.
10. Los ciclos de lectura y escritura del  $\mu P$  68000 finalizan cuando éste detecta la activación de la señal  $\overline{DTACK}$  (*Data Transfer Acknowledge*). ¿Qué debe ocurrir si el 68000 direcciona un dispositivo inexistente y por tanto nunca se va a activar  $\overline{DTACK}$ ?
11. Explique *brevemente* la utilidad de que una CPU disponga de (al menos) dos modos de privilegio, como los modos usuario y supervisor del 68000. En el caso de dicho  $\mu P$ , ¿cómo se puede pasar de un modo a otro?



12. Analice la siguiente subrutina:

```
RUTINAX MOVEA. L 4 (A7) , A0
MOVEA. L 8 (A7) , A1
CLR. L D0
MOVE. W 12 (A7) , D0

ADDA. L D0 , A1

SUBQ. W #1 , D0
CICLO MOVE. B (A0) + , - (A1)
DBF D0 , CICLO

RTS
```

- Determine qué hace y cuáles son sus parámetros, y documéntela apropiadamente (cabecera completa y comentarios).
  - ¿Por qué se borra el registro D0 (en tamaño *long*) al principio?
  - Escriba un pequeño programa principal que se limite a llamar a la RUTINAX, incluyendo el paso de parámetros.
13. Explique (ayudándose de un ejemplo gráfico si lo desea) la diferencia entre una CPU *big-endian* y otra *little-endian*. ¿Qué problemas puede suponer esa diferencia en la práctica? ¿Afecta a texto codificado en UTF-8?
14. ¿Qué significa que se ha producido un rebose (*overflow*) en una operación aritmética? ¿Cómo se puede detectar tras ejecutar una instrucción de la CPU de suma o resta de números enteros, sin signo o con signo? Ponga un ejemplo de suma de dos números de 4 bits que produzca un rebose si se interpretan como números con signo (complemento a dos), pero no lo haga si se interpretan como números sin signo.
15. Suponga que desea mapear memoria RAM en los primeros 4 MB del espacio de direcciones del 68000, para lo cual dispone de chips de 1Mx4. Determine gráficamente la disposición de los chips necesarios en un esquema del espacio de direcciones, así como los rangos de direcciones que ocupan. Una vez hecho esto, determine qué posiciones en concreto de qué chips se verán afectadas por una transferencia de tamaño byte con la dirección 23184D<sub>16</sub>.
16. Dado el siguiente código de 68000:

```
LEA $2000 , A0
LEA $3000 , A1
MOVE. W #99 , D0
LOOP MOVE. B (A0) + , (A1) +
DBF D0 , LOOP
```

- ¿Qué hace ese código?
  - Determine razonadamente de cuántas palabras de 16 bits se compondrá cada instrucción (palabra de instrucción más posibles palabras adicionales).
  - Calcule cuántos ciclos de lectura y escritura se generan durante el procesamiento (lectura y ejecución) de cada instrucción.
17. ¿Qué diferencia hay entre un salto absoluto y uno relativo? (Como conceptos genéricos de cualquier tipo de CPU.) Ponga un ejemplo de cada uno utilizando instrucciones del 68000.
18. En un momento determinado dos dispositivos generan sendas peticiones de interrupción al 68000: uno de ellos de nivel 2 y el otro de nivel 5. (Suponemos que, en caso de haber dos o más peticiones de interrupción simultáneas, el controlador de interrupciones codifica la de mayor nivel en las entradas IPL0-IPL2.) Cada petición de interrupción se desactivará en la rutina de servicio correspondiente. Describa qué sucederá a partir del momento en que el 68000 detecte la interrupción (suponemos que tiene codificada prioridad 0 en el SR y se encuentra en modo usuario).
19. Determine qué pondrá el 68000 en sus líneas del bus de direcciones, así como en las señales  $\overline{UDS}$  y  $\overline{LDS}$ , al realizar los siguientes ciclos de lectura o escritura: dirección 0016A2F4 tamaño word, la misma dirección en tamaño byte, y la dirección 0016A2F5 en tamaño byte.



20. En el espacio de direcciones del 68000 se desea mapear 8 Mbytes de memoria RAM a partir de la dirección  $800000_{16}$  empleando dispositivos de estructura  $1M \times 4$ . Así mismo, se desea mapear 256 Kbytes de memoria ROM a partir de la dirección  $000000_{16}$  usando dispositivos  $64K \times 8$ . Dibuje un esquema de la organización de los dispositivos necesarios en el mapa del espacio de direcciones, así como los rangos de direcciones que ocuparían. ¿Cuántas líneas de direccionamiento requieren ambos tipos de dispositivos?
21. Explique (a grandes rasgos) en qué consiste la codificación de texto UTF-8. Un archivo de texto codificado en UTF-8 ¿es inmune al diferente orden de los bytes big-endian y little-endian?
22. ¿En qué circunstancias se genera la excepción de *bus error*? Dicha excepción ¿aborta el ciclo de lectura/escritura en curso, o se procesa al finalizar la instrucción durante la cual se ha provocado? ¿Tiene sentido retornar de su rutina de servicio de excepción?
23. La subrutina que se muestra a continuación realiza cierta operación con todos los elementos de un vector (cuya dirección de comienzo es uno de sus parámetros). Se trata de un vector de words cuyo primera word contiene el número de elementos que lo forman (sin contar esa primera word).

```

RUTINAX MOVE.L 4(A7),A0
MOVE.W (A0)+,D7
BEQ.S FIN
SUBQ.W #1,D7
LOOP MOVE.W (A0),D0
MULU 8(A7),D0
SWAP D0
TST.W D0
BEQ.S NOOFLOW
MOVEQ #-1,D0
NOOFLOW SWAP D0
MOVE.W D0,(A0)+
DBF D7,LOOP
FIN RTS

```

- (a) Explique qué hace esa subrutina. Pistas: la instrucción **MULU** multiplica  $16 \times 16 \rightarrow 32$  bits; la instrucción **MOVEQ** opera siempre en tamaño 32 bits, y su operando origen está en el rango -128 a +127.
  - (b) ¿Es realmente necesaria la instrucción **SUBQ**? Por otra parte, ¿no bastaría con comprobar si tras ejecutar la instrucción **MULU** se pone a 1 el bit *V* del *CCR*?
  - (c) Determine razonadamente qué ciclos de lectura y escritura se generarán durante el procesamiento (lectura y ejecución) de cada instrucción.
  - (d) Escriba unas instrucciones que realicen una llamada a esa subrutina, incluyendo el paso de parámetros (y su eliminación, en su caso).
24. El microprocesador Intel 8086, origen de la familia x86 y de la misma época que el Motorola 68000, tiene un bus de datos de 16 bits (líneas  $D0$  a  $D15$ ) y un bus de direcciones de 20 bits (líneas  $A0$  a  $A19$ ). Al igual que ocurre con el 68000, las direcciones son de byte (es decir, a cada byte de su espacio de direcciones le corresponde una dirección). Sin embargo, a diferencia del 68000, nótese que sí existe una señal  $A0$  en el bus de direcciones del 8086. Cuenta además con una señal  $\overline{BHE}$  (*Bus High Enable*) que en combinación con  $A0$  se interpreta de la siguiente forma:

$\overline{BHE}$	$A0$	Tamaño del dato y líneas utilizadas
0	0	Word a través de $D0-D15$
0	1	Byte a través de $D8-D15$
1	0	Byte a través de $D0-D7$
1	1	Sin significado

- (a) Compare este método con el que utiliza el 68000 para distinguir entre las transferencias de datos tamaño word, y las de tamaño byte por sendas mitades de su bus de datos. ¿Son funcionalmente equivalentes?
- (b) Dibuje la estructura del espacio de direcciones del Intel 8086.
- (c) Deseamos ubicar 512 KB de RAM a partir de la dirección  $000000_{16}$  empleando dispositivos de estructura  $128K \times 8$ . Dibuje un esquema de la organización de los dispositivos necesarios en el mapa del espacio de direcciones, así como los rangos de direcciones que ocuparían. ¿Cuántas líneas de direccionamiento y de datos requieren esos dispositivos?



(d) Puesto que el Intel 8086 es little-endian, ¿Cómo se almacenaría el número entero  $666.666_{10}$  (en 32 bits) a partir de la dirección  $0A36E_{16}$ ?

25. ¿Qué problemas viene a resolver Unicode frente a codificaciones como las ASCII-extendido? ¿Para qué se define adicionalmente el formato de transformación Unicode UTF-8?

26. La subrutina que se muestra a continuación toma tres parámetros en A0 (L), D0 (W) y D1 (W). Básicamente accede a un elemento concreto de cierta estructura de datos, cuyo valor retorna en D7 (W). Además utiliza el bit V del CCR para indicar si algún parámetro está fuera del rango admisible.

```
RUTINAX CMP.W    (A0),D0
BCC.S    ERROR
CMP.W    2(A0),D1
BCC.S    ERROR
MULU    2(A0),D0
MULU    #2,D0
LEA     4(A0,D0.L),A0
MULU    #2,D1
MOVE.W  (A0,D1.L),D7    ;tambien pone V=0
BRA.S    FIN
ERROR    ORI     #%00000010,CCR    ;pone V=1
FIN      RTS
```

(a) ¿Qué forma tiene la estructura de datos a la que accede esa subrutina?

(b) Determine razonadamente qué ciclos de lectura y escritura se generarán durante el procesamiento (lectura y ejecución) de cada instrucción.

(c) Tal como está, esa subrutina modifica los registros A0, D0 y D1 (aparte, claro está, del D7 y el CCR, donde retorna los resultados). ¿Qué podemos hacer para que globalmente no modifique ninguno de esos tres registros?

(d) Escriba un programa principal en el que se realice una llamada a la subrutina pasándole unos parámetros adecuados. Para ello inicialice una estructura del tipo que maneja esa subrutina a partir de la dirección  $2000_{16}$  empleando las pseudoinstrucciones DC apropiadas.

27. Deseamos colocar 1,5 Mbytes de memoria RAM en el inicio del mapa del espacio de direcciones de un 68000. Para ello disponemos de dispositivos 512Kx4 y 256Kx8. Diseñe las posibles disposiciones con las que lo conseguiríamos y calcule los rangos de direcciones resultantes para cada una. Así mismo, determine el nº de líneas de datos y de direcciones de ambos tipos de dispositivos.

28. Desde un programa que se ejecuta en modo usuario se desea hacer uso de una rutina del *kernel* del sistema operativo que proporciona cierta funcionalidad. Dicha rutina ha de ejecutarse en modo supervisor. ¿De qué manera se puede conseguir? ¿Qué secuencia de eventos tiene lugar hasta que el flujo de ejecución vuelve al programa de usuario?

29. La subrutina que se muestra a continuación toma dos parámetros: el primero es una dirección de comienzo de cierta estructura de datos, y el segundo es un dato de tamaño byte. Ambos se le pasan a través de la pila. Proporciona un valor de retorno en el registro D7.

```
RUTX    MOVE.L    6(A7),A0
        CLR.W     D7
LOOP    MOVE.B    (A0)+,D0
        BEQ.S     FIN
        CMP.B     4(A7),D0
        BNE.S     LOOP
        ADDQ.W   #1,D7
        BRA.S     LOOP
FIN     RTS
```

(a) Analice su funcionamiento. ¿Qué es lo que hace concretamente?

(b) Describa gráficamente qué estructura tiene su *stack frame* (marco de pila). Aunque en este caso no es estrictamente necesario, modifique la subrutina usando las instrucciones **LINK** A6, #0 y **UNLK** A6 en sus lugares apropiados. ¿Cómo quedaría ahora su *stack frame*?



- (c) Escriba un programa principal en el que se realice una llamada a la subrutina con unos parámetros adecuados. Para ello inicialice a partir de la dirección  $2000_{16}$  una estructura del tipo que maneja la susodicha subrutina empleando la pseudoinstrucción apropiada.
- (d) Enumere razonadamente qué ciclos de lectura y escritura se realizan durante la obtención (*fetch*) y ejecución de la primera y de la última instrucción. ¿Cuántos ciclos de escritura se realizan en total durante una ejecución completa de la subrutina (sin incluir la llamada)?

### **Preguntas “tipo” de los temas añadidos este curso**

- 30. Tres matemáticos, Hilbert, Godel y Turing, destacan entre otros en el desarrollo de un objetivo determinado. ¿Qué objetivo y por qué aportaciones concretas destacan?
- 31. De qué elementos consta una Máquina de Turing (en detalle, como se expresa matemáticamente, pero puede hacerse con lenguaje “normal” -sin notación matemática-)
- 32. ¿Qué entendemos por “Fuente Markoviana”?
- 33. ¿Qué quiere decir que una Fuente Markoviana sea de orden 0, 1, 2...?