2º Grado Informática
Estructura de Computadores
2 de febrero de 2015

Nombre:	
DNI:	**Grupo:**

Examen de Problemas (3,0 p)

1. **Ensamblador**. (0,75 puntos). Dado el siguiente fragmento de código escrito en C, escriba el código de la función equivalente utilizando el ensamblador de IA32:

```
int max (int a, int b)
{
if (a > b)
     return a;
else
     return b;
}
```

Utilizando la función en ensamblador anterior, implemente el código de la siguiente función:

```
void maxV (int *v1, int *v2, int *v3, int N)
{
int i;
for (i = 0; i < N; i++)
     v3[i] = max (v1[i], v2[i]);
}
```

Para el desarrollo de este problema ha de seguirse la convención de paso de parámetros _cdecl.

2. **Ensamblador** (0,7 puntos). Dada una definición de un vector de enteros similar a esta:

```
array: .int 10, 20, 30, 40, 6, -5, 9, 8, 10, 35, 45, 56, 7
```

Escriba un fragmento de programa que sume el valor de todos los números del array que sean menores de 30. Suponga que los valores a sumar nunca serán grandes en valor absoluto y por tanto no hay que preocuparse por el desbordamiento.

3. **Unidad de control** (0,25 puntos). Una unidad de control microprogramada utiliza una memoria de control de 1024 palabras de 32 bits cada una. Las microinstrucciones tienen 3 campos: tipo/condición de salto, dirección de salto y bits de control. El último campo (bits de control) tiene 16 bits.

a) ¿Cuántos bits tiene el registro de microinstrucción?

b) ¿Cuántos bits hay en el campo de dirección siguiente?

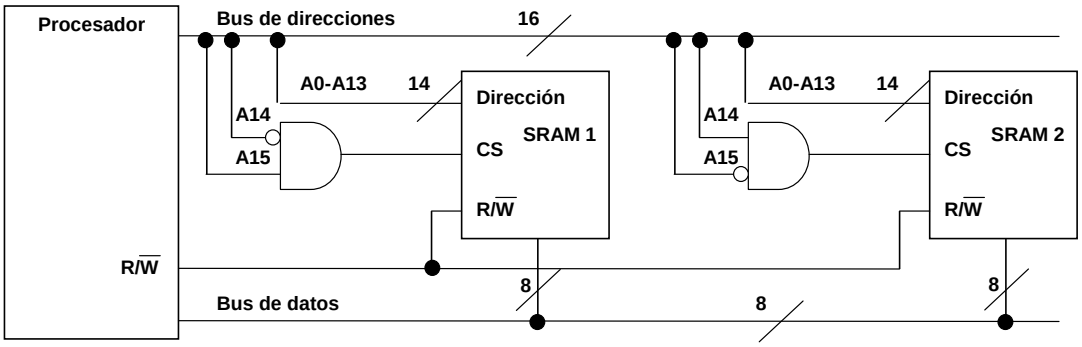
c) ¿Cuántos bits tiene el contador de microprograma?

d) ¿Cuántos bits hay en el campo de tipo/condición de salto?

e) ¿Cuántos bits tiene cada uno de los buses que entran al multiplexor que selecciona la dirección a cargar en el contador de microprograma?

4. **Entrada/Salida** (0,5 puntos). Un sistema basado en un microprocesador de 8 bits (8 bits de datos y 16 bits de direcciones) con E/S independiente debe conectarse a un puerto de salida de 8 bits en la dirección `0xFF`. Antes de escribir un dato en ese puerto hay que leer de un puerto de entrada de 8 bits, también en la dirección de E/S `0xFF`, hasta que el bit 7 sea 1. Dibuje un esquema del sistema y escriba una función **`void write_block (char *buffer)`** en C/C++ que envíe un bloque de 100 caracteres hacia el puerto de salida. Para escribir en el puerto use la función **`out (unsigned short port, char c)`** y para leer la función **`char in (unsigned short port)`**.

5. **Diseño del sistema de memoria** (0,4 puntos). Se dispone del sistema de memoria de la figura:



Indique el tamaño (en formato x × y) y el rango de direcciones (en hexadecimal) de cada módulo SRAM.

6. **Memoria cache** (0,4 puntos). Una determinada implementación de la arquitectura x86-64 puede usar direcciones físicas de 52 bits, con una caché L1 de 32 KB de instrucciones y otra caché L1 de 32 KB de datos, asociativa por conjuntos con 8 vías y con líneas de 64 bytes. Muestre el esquema de una dirección de memoria física y explique cómo se sabe si la dirección física de una instrucción está en la caché L1.