

	FUNDAMENTOS DE COMPUTADORES 6 de septiembre de 2011
	Nombre _____ DNI _____ Apellidos _____ Grupo _____

Ejercicio 1.- (1'5 ptos.) Dado el siguiente código escrito en lenguaje C

```

int v[16];                // Dirección de comienzo 0x0000.
                          // Cada entero ocupa 4 bytes.
int mask[4];             // Dirección de comienzo 0x0040

1  for(i=0; i<12; i++) {
2      tmp = 0;
3      for(j=0; j<4; j++){
4          tmp = tmp+mask[j]*v[i];
5      }
6      v[i]=tmp;
7  }
    
```

Programa una subrutina en lenguaje ensamblador del ARM que implemente el bucle interno (líneas 2-5). Esta rutina recibirá como parámetros de entrada la dirección del vector mask y la posición i-ésima del vector v y devolverá el valor de tmp:

```
int for2(int mask[], int vi[]);
```

Nota: Respete el convenio estándar de paso de parámetros de entrada y salida, así como de salvaguarda de registros utilizados.

Ejercicio 2.- Suponga los siguientes tiempos de conmutación para los elementos funcionales de la ruta de datos mono-ciclo:

I-Mem	Add	Mux	ALU	BReg	D-Mem	Control
450	150	35	140	210	380	90

a) (1 pto.) Determine, justificando la respuesta, el tiempo de ciclo mínimo para ejecutar la instrucción beq (salta si igual).

b) (1 pto.) Considere la instrucción add \$13, \$2, \$3. Indique cuál es el valor de las señales de control RegDst, MemWrite, ALUSrc y MemtoReg; así como el valor de las entradas RA, RB y RW del Banco de Registros y la salida del extensor de signo.

Ejercicio 3.- Dado el siguiente código del MIPS:

```
lw    $2, 4($4)
```

```

lw    $6, 8($0)
subi  $5, $0, 12
slt   $7, $10, $11
add   $4, $2, $3
lw    $9, 0($5)
add   $1, $7, $4
sw    $5, 12($1)
    
```

a) (0'5) Considerando la implementación del procesador segmentada, determine los conflictos de datos.

b) (0'5) Reescriba el programa introduciendo operaciones NOP para eliminar los posibles conflictos.

Ejercicio 4.- Se va a ejecutar el código del ejercicio 1 en una máquina con direcciones de 16 bits que dispone de una memoria cache de acceso directo con 4 bloques de 16 bytes cada uno.

a) (0'75) Indique en qué (marco de) bloque de la memoria cache se encontrará el elemento $v[1]$. ¿Qué etiqueta tendrá el (marco de) bloque 2 de la cache tras la ejecución del código? Justifique las respuestas.

b) (0'75) Ignorando los accesos a la variable *tmp*, ¿cuántos fallos de cache se producirán? Justifique la respuesta, indicando además cuántos reemplazamientos se producen (y en qué iteraciones).

