Práctica 2 sesión 1 Arquitecturas avanzadas de procesadores

Resumen

En esta sesión de prácticas quiero que elijáis uno de los siguientes códigos ensamblador y:

- Identifiquéis los riesgos de dependencia de datos.
- Corrijáis estos riesgos de la manera más eficiente posible.
- Justifiquéis en MIPSIM la segmentación que se produce, las señales que se activan, datos, etc.

1. Presentación de los dos programas en ensamblador

1.1. Opción 1 de bucle sumador

En este algoritmo, cargamos los tres registros primeros del bloque de datos, los multiplicamos por dos, y los almacenamos en el bloque de datos en los tres registros siguientes.

¡Ojo! Si insertáis burbujas Nop para evitar el solapamiento de datos, corregid acorde el offset del salto en la instrucción bneq

```
LI $4,0002
LI $3,000c
LI $2,0000
ADDI $3,$3,0004
LW $1,0000($2)
ADD $1,$1,$1
SW $1,0000($3)
ADDI $2,$2,0004
SUBI $4,$4,0001
BNEQ $4,$0,-1c
```

1.2. Opción 2 de bucle trasladador de información

En este algortimo vamos a coger un bloque de información y lo vamos a mover en el segmento de datos.

¡Ojo! Si insertáis burbujas Nop para evitar el solapamiento de datos, corregid acorde el offset del salto en la instrucción bneq

```
LI $2, 10
LI $3, 3
LI $1, 0
LW $4, 0($1)
SW $0, 0($1)
SW $4, 0($2)
ADDI $1, $1, 4
ADDI $2, $2, 4
SUBI $3, $3, 1
BNEQ $3, $0, ffe5
```

2. ¿Qué os pedimos en la práctica?

Tenéis que copiar el código proporcionado en el programa **MIPSIM**, en su segmento de código correspondiente.

Para ambos ejemplos vamos a cargar en el segmento de datos los valores 1, 2 y 3 en los tres primeros registros, como se muestra en la figura 1

dispondréis en las sesiones 1 y 2 de las prácticas, de 10 minutos para realizar una defensa verbal del ejercicio, donde se valorará:

- Control del código ensamblador en MIPS, con una explicación a **muy bajo nivel, valores, direcciones de memoria, etc.** No nos interesa la abstracción a alto nivel.
- Los códigos suministrados tienen solapamiento de datos, el código final debe de funcionar, y se valorará la solución más eficiente e ingeniosa.
- Explicación detallada de las líneas de control que se activan y desactivan en todo el proceso de segmentación, y su porqué.

Tomároslo con calma, el objetivo es que habléis con propiedad y sepáis transmitir que controláis el proceso de segmentación de un procesador MIPS

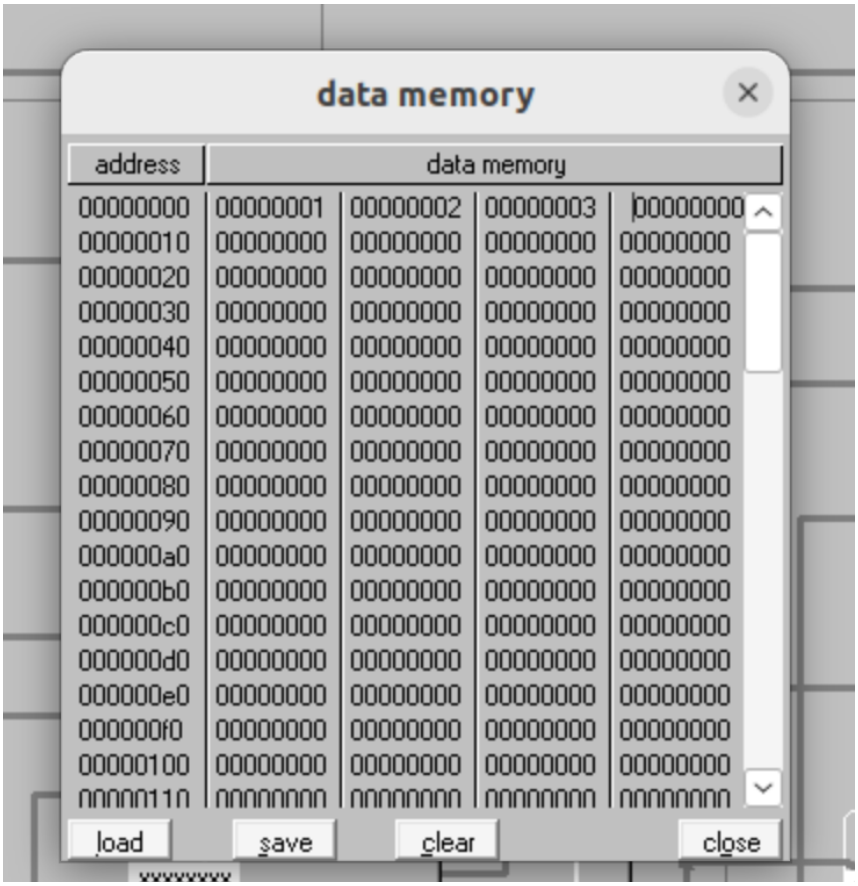

Figura 1: Datos iniciales del programa.