

Práctica 1. Revisión de las instrucciones sintéticas

1. Sean los siguientes datos:

A: .word 0x20, 0x40, 0x10, 0x01, 0x03, 0x22, 0x08

B: .word 10, 24, 55, 67, 89, 90, 110

C: .space 100

- a) Escriba el código que implemente un bucle para efectuar la operación $C[i] = A[i] + B[i]$ para cada componente de los vectores y luego muestre por pantalla el vector C con las componentes separadas por comas.
- b) Una vez escrito el programa detecte qué instrucciones son sintéticas y por cuáles se sustituyen, por ejemplo:

- `move rd, rs` \iff `addu rd, $zero, rs`
- `sub`, `add` con constantes (`addi`)
 `addi` con 2 operandos
 `sub`, `add` con constantes de 32 bits
 `subi`
- `li`
- `la`
- `lw rd, Etiqueta`
 `sw rd, Etiqueta`
 `lw` o `sw` con desplazamientos de 32 bits
- `blt`, `bgt`, etc.

Fuerce la inclusión de algunas de estas instrucciones en el programa y escriba en el diario por qué instrucciones se traducen. Observe que hay nemónicos que pueden corresponder a instrucciones naturales o sintéticas, según la forma en que se empleen. Mencione en el diario algunos ejemplos en que eso ocurra.

- c) Pruebe otras instrucciones sintéticas y determine por qué instrucciones naturales se traducen. Puede encontrar una lista de instrucciones sintéticas en la ayuda del simulador.
- d) En los ajustes desactive la opción "*Permit extended (pseudo) instructions and formats*" y compile nuevamente el programa sin modificarlo. Analice los errores que aparecen. Escriba ahora el mismo programa sin emplear instrucciones sintéticas ¿Ha disminuido la legibilidad del programa?

2. Pruebe ahora el simulador *QtSPIM* y navegue por su menú.

- a) Cargue y ejecute los programas anteriores (con instrucciones sintéticas y sin ellas). Actúe sobre el ajuste "*Allow pseudoinstructions*" para que se admitan o no las instrucciones sintéticas. ¿Se traducen todas las instrucciones sintéticas de la misma forma en *mars* que en *QtSPIM*? Si eso no ocurre, escriba en el diario los casos que haya detectado. ¿Por qué podrían traducirse las instrucciones sintéticas de forma distinta si el procesador que se está simulando es el mismo?
- b) Compare las traducciones a lenguaje máquina de las instrucciones naturales de bifurcación (`beq` y `bne`) en los dos simuladores ¿se traducen igual en ambos? Si las traducciones fueran diferentes, ¿cuál de las dos es la correcta?

3. En el simulador *mars* active en algunas instrucciones la casilla *Bkpt* (*Breakpoint*, punto de ruptura) y ejecute el programa de forma continua. Observará que el programa se detiene en el punto de ruptura marcado y puede analizarse el contenido de los registros, memoria, etc. Esto es muy útil cuando se pretende depurar un programa que no da el resultado esperado.