

Problemas sobre el juego de instrucciones MIPS (Ejemplos resueltos)

March 24, 2021

1 Introducción

Los siguientes problemas nos servirán para profundizar en algunos aspectos del procesador MIPS32. En unas hojas separadas se completan con problemas sobre el stack frame. Algunos de estos problemas están extraídos de [1].

2 Problemas

Problema 1. Para este problema usaremos un pseudocódigo que debe entenderse del siguiente modo: letras sin corchetes se corresponden a variables (por ejemplo, g , h , \dots), letra con corchetes referencia una posición en un array de valores de 32 bits (por ejemplo, $A[0]$, $B[4]$). Al traducir el pseudocódigo a ensamblador las variables deben hacerse corresponder en lo posible con registros y asumimos que el array tiene su dirección base en un registro. Supongamos un compilador capaz de traducir el pseudocódigo a ensamblador MIPS 32 (sin pseudoinstrucciones) ¿cómo crees que traduciría las siguientes sentencias?

1. $g = h + A[8]$
2. $A[2] = h + A[1]$
3. $A[10] = A[9] + A[8]$

Problema 2. Sea el siguiente programa MIPS 32

```
.data

valor: .word 0xF0E1D243
valor1: .word 0
.text

la $s0, valor
lbu $t0, 3($s0)
```

¿Qué valor debería tener $\$t0$ tras ejecutar la última línea si el procesador es little endian?. ¿Y si es big endian?. Contestar las dos últimas preguntas asumiendo que en la última línea de código `lbu` es sustituido por `lb`.

Notas:

- **lbu** tiene formato I y su acción es: $R[rt] = \{\text{ZeroExt}(M[R[rs] + \text{SignExtImm}](7:0))\}$
- **lb** tiene formato I y su acción es: $R[rt] = \{\text{SignExt}(M[R[rs] + \text{SignExtImm}](7:0))\}$
- Advertir que el reference card de Mips tiene un error en este punto. El mismo error está en la definición de `lh` y `lhu`.

Problema 3. Sean los siguientes ejemplos de instrucción beq.

1. etiqueta: beq t0,t0, etiqueta
2. beq t0,t0, etiqueta (etiqueta está 32000 instrucciones después)
3. beq t0,t0, etiqueta (etiqueta está 32768 instrucciones antes)

Calcular el valor del campo “inmediato” en la codificación I de la instrucción, teniendo en cuenta que si PC contiene la dirección de la siguiente instrucción después de beq, el efecto de la instrucción beq es:

```
if ( R[rs]==R[rt]) PC=PC+BranchAddr donde  
BranchAddr = {14{inmediato[15]},inmediato, 0,0}
```

Problema 4. La instrucción jump (j etiqueta) utiliza el formato J donde un valor inmediato ocupa 26 bits. Si asumimos que PCorigen es la dirección de la instrucción que j, el efecto de la instrucción jump es:

PC=JumpAddr donde JumpAddr={PCorigen[31:28],inmediato,0,0}

¿Por qué se dice que esta instrucción sólo permite el salto dentro del “mismo segmento de la memoria” que la instrucción de salto?. Calcular el tamaño del segmento en MB.

Problema 5. Marca qué instrucciones son correctas e incorrectas asumiendo que el registro \$s3 contiene el valor 0x0000CDE04

1. lw \$t0,3(\$s3)
2. lb \$t0,3(\$s3)
3. lh \$t0,2(\$s3)
4. sw \$t0,4(\$s3)

Problema 6. Intenta traducir las siguientes pseudoinstrucciones en instrucciones propias del procesador MIPS (Problema extraído de [1]).

1. li \$t0,4
2. li \$t0,-1
3. li \$t1,0xFF001234

Problema 7. Supongamos que la dirección de una variable es 0x00400000. Usando instrucciones de MIPS, ¿cómo establecería el registro \$s0 para que fuera un registro base a esta dirección?. ¿Y si la dirección fuera 0x00400004?

Problema 8. ¿Qué diferencia hay entre las siguientes instrucciones?

1. ori s0,s0,0x0004
2. ori s0,zero, 0x0004

Problema 9. La pseudoinstrucción move permite copiar el contenido de un registro en otro. ¿Cómo implementarías mediante instrucciones una pseudoinstrucción como move \$t1,\$t2 (\$t1 es el registro destino)

Problema 10. ¿Qué pseudoinstrucción crees que puede ser equivalente a addiu \$t0,\$zero,-320?

Problema 11. Consideremos la pseudoinstrucción lw \$t0, etiqueta, donde etiqueta representa la dirección 0x001000010. ¿Cómo podría traducirse a instrucciones propias del procesador MIPS?.

References

- [1] Bradley Kjell. Programmed introduction to mips assembly language. Curso de la Central Connecticut State University, URL: <http://micron66.tripod.com/d3.pdf>, 2002. Accessed: 2016-04-04.