

Plancha 4

Arquitectura ARM

Arquitectura del Computador
Licenciatura en Ciencias de la Computación

9 de noviembre de 2023

1) Suponga que un arreglo `array` contiene 10 palabras de 4 bytes. Un compilador asocia las variables `x` e `y` con los registros `r0` y `r1`, respectivamente. Suponga que la dirección inicial de la matriz está contenida en el registro `r2`. Traduzca las siguientes declaraciones C en instrucciones de Assembler ARM:

a) `x = array[7] + y;`

b) `array[10] = array[8] + y;`

En cada uno de los casos realizar el código completo para poder compilar y depurar.

2) Suponga que el registro `r3` contiene el valor `0x8000`. ¿Qué contendría el registro después de ejecutar las siguientes instrucciones?

a) `STR r6, [r3, #12]`

b) `STRB r7, [r3], #4`

c) `LDRH r5, [r3], #8`

d) `LDR r12, [r3, #12]!`

3) Suponiendo que se tiene una arquitectura ARM *little-endian*,

a) ¿qué contendría el registro `r4` después de ejecutar las siguientes instrucciones?

`STR r6, [r3]`

`LDRB r4, [r3]`

El registro `r6` contiene el valor `0xBEEFFACE` y el registro `r3` contiene `0x8000`.

b) ¿Qué pasaría si fuera *big-endian*?

4) ¿Qué constante se cargaría en el registro `r7` con las siguientes instrucciones?

a) `MOV r7, #0x8C, 4`

b) `MOV r7, #0x42, 30`

c) `MVN r7, #2`

d) `MVN r7, #0x8C, 4`

5) Usando el esquema de rotación de bytes, calcule la instrucción y la rotación necesarias para cargar las siguientes constantes en el registro `r2`:

a) `0xA400`

b) `0x7D8`

c) `0x17400`

d) `0x1980`

6) Indique si las siguientes constantes se pueden cargar o no en un registro sin crear un *literal pool* y usando solo una instrucción:

a) `0xC0000034`

b) `0x12340000`

c) `0x77777777`

d) `0xFFFFFFFF`

e) `0xFFFFFFFFE`

7) Indicar el inconveniente en cada una de las siguientes instrucciones ARM e indicar otra alternativa válida:

a) `ADD r3, r7, #1023`

b) `SUB r11, r12, r3, LSL #32`

8) Desde la versión ARMv6T2 de la arquitectura ARM, puede usarse la siguiente instrucción para cargar un valor inmediato de 32 bits o la dirección de 32 bits de una etiqueta en un registro:

```
ldr r1, =0x12abcdef
ldr r1, =etiqueta
```

Ahora bien, si todas las instrucciones tienen 32 bits de longitud (o 16 en el perfil THUMB), ¿cómo explica que en una sola instrucción pueda usarse un valor inmediato de ese tamaño?

9) El algoritmo del campesino ruso puede implementarse en el lenguaje c como:

```
unsigned int campesino_ruso(unsigned int i, unsigned int j)
{
    unsigned int res = 0;

    while (j > 1) {
        if (j & 1) {
            res += i;
            j -= 1;
        } else {
```

```

        i *= 2;
        j /= 2;
    }
}
return res + i;
}

```

Escriba esta función para arquitectura ARM, utilizando solo una instrucción de salto en su implementación.

Pista: para este ejercicio deberá utilizar las capacidades de ejecución condicional de las instrucciones de la arquitectura ARM

10) Escriba una función que dado un argumento i entre 0 y 31 en `r0`, calcule y retorne 2^i . Para calcular este valor sólo puede usar instrucciones de copia (`MOV`).

11) Escriba una función que dados 4 valores de punto flotante representando una matriz de 2x2, devuelva su determinante.