

ESTRUCTURA DE COMPUTADORES (Grado en MI)
EXAMEN FINAL DE JULIO -parcial 1- (4 de julio de 2018)

PROBLEMAS

1 [2 puntos] Considere un computador con una frecuencia de reloj de 2GHz y modelo de ejecución registro-memoria y conteste razonadamente a las siguientes cuestiones.

a) ¿Qué componentes del computador (registro, memoria, etc.) están involucrados en cada una de las fases de ejecución de la instrucción aritmética *ADD .R1, .R2, [.R3]*?

b) Sabiendo que en este computador se pueden direccionar 256GB de memoria, ¿Cuántos bits se emplean para las direcciones de memoria? ¿A qué registros del procesador afecta esta característica?

c) Suponiendo que en la ejecución de cada instrucción de cierto programa se emplease una media de 160 ciclos, calcule el número medio de instrucciones ejecutadas por segundo.

SOLUCIÓN

a) Los componentes involucrados en cada fase son los siguientes:

- *fetch*: La memoria, sus registros de direcciones y datos (AR y DR), el registro contador de programa (PC) y el registro de instrucción (RI)
- *decodificación*: El registro RI y la Unidad de control
- *ejecución*:
 - busqueda de operandos*: El banco de registros, la memoria y los registros AR y DR
 - operación*: La unidad aritmético lógica (ALU) y el registro de estado (RE)
 - almacenamiento del resultado*: El banco de registros

b) Para direccionar 256GB ($2^8 \times 2^{30} = 2^{38}$ bytes) se necesitan 38 bits. Esta característica afecta al tamaño de los registros que contienen direcciones, como son el contador de programa, el puntero de pila y el registro de direcciones de memoria (PC, SP y AR). Además, si el juego de instrucciones del computador dispone de modos de direccionamiento relativo o indirecto a registro, también afectaría al tamaño de los registros generales.

c) Dado que la frecuencia de reloj es de 2 GHz, el tiempo de ciclo es $1/(2 \times 10^9)s = 0,5ns$, por lo que el tiempo medio de ejecución por instrucción es $160 \text{ ciclos} \times 0,5ns/\text{ciclo} = 80ns$. En consecuencia, el computador puede ejecutar una media de $1/(80 \times 10^{-9}) = 12.500.000 \text{ instrucciones/s} \rightarrow 12,5MIPS$

2 [5 puntos]. Se tiene un formato de coma flotante de 16 bits cuyo bit superior representa el bit de signo, los cinco siguientes el exponente expresado en exceso a 16 y los diez bits siguientes la magnitud de la mantisa, normalizada con bit implícito y con la coma a la izquierda de dicho bit.

- a) Determine el rango de representación del formato especificando cómo representaría el cero.
- b) $A = H'3CB5$ es una representación en este formato. Determine su valor decimal.
- c) Represente en el formato el número decimal $B = -50,0125$.
- d) Realice la operación $A+B$ utilizando 2 bits de guarda, bit retenedor y redondeo al más próximo, dejando el resultado en el formato y determinando su valor decimal.
- e) Rediseñe el formato de representación sin variar el número total de bits para que se pueda representar el número 10^{10} .

SOLUCIÓN

1. Rango y resolución del formato.

Exponente: Como está representado en exceso a 16 su rango es $[-16, 15]$. Como el formato tiene bit implícito, hay que reservar un valor del exponente para la representación del cero. Reservamos el valor

-16 (todo ceros). Por lo tanto, el rango del exponente es $[-15, 15]$. El cero se representará con los 16 bits a cero.

Mantisa: Como está representada en signo magnitud su rango es simétrico, siendo:

$$\text{Mantisas positivas: } \begin{cases} ,1000000000 & \rightarrow 2^{-1} \\ ,1111111111 & \rightarrow 1 - 2^{-11} \end{cases}$$

El rango es: $\pm [2^{-1} \cdot 2^{-15}, (1 - 2^{-11}) \cdot 2^{15}] \cup 0$

La resolución es: $2^{-11} \cdot 2^E$

2. Valor decimal de un número.

$$A = \text{H}'3\text{CB5} = 0011 \ 1100 \ 1011 \ 0101 = 0 \ 01111 \ 0010110101 = +,10010110101 \cdot 2^{-1}$$

$$A = +,010010110101 = +0,294189453$$

3. Representación de un número decimal.

$$B = -50,0125 = -110010,00000011 = -,11001000000 \cdot 2^6$$

Quedando en el formato $B = 1 \ 10110 \ 1001000000 = \text{H}'\text{DA40}$.

4. Suma $A + B$. Los números a sumar son:

$$A = +,10010110101 \cdot 2^{-1} \quad y \quad B = -,11001000000 \cdot 2^6$$

Determinamos la diferencia de exponentes: $E_A - E_B = -1 - 6 = -7$. Hay que desplazar la mantisa de A siete lugares a la derecha.

Suma de las mantisas (resta):

$$\begin{array}{rcl} B & -,11001000000 \ 000 \cdot 2^6 & \\ A & +,00000001001 \ 011 \cdot 2^6 & \\ \hline A + B & -,11000110110 \ 101 \cdot 2^6 & \text{Normalizado} \\ & +,00000000000 \ 100 & \text{Redondeo} \\ & \hline & -,11000110111 \ 001 & \cdot 2^6 \end{array}$$

$$A + B = -,11000110111 \cdot 2^6 = 1 \ 10110 \ 1000110111 = \text{H}'\text{DA37}$$

$$A + B = -110001,10111 = -49,71875$$

5. Rediseño del formato.

En este formato el número más grande representable es $(1 - 2^{-11}) \cdot 2^{15}$. En general, llamando p al número de bits de la mantisa y E_{max} al máximo exponente, tendremos que dicho número sería $(1 - 2^{-p}) \cdot 2^{E_{max}}$.

Así, se debe cumplir: $(1 - 2^{-p}) \cdot 2^{E_{max}} > 10^{10}$

Como 2^{-p} es despreciable frente a la unidad podemos aproximar por: $2^{E_{max}} > 10^{10}$

Tomando logaritmos: $E_{max} \cdot \log 2 > 10 \rightarrow E_{max} > 33,21$

Necesitamos representar los exponentes en exceso a 64, es decir, con 7 bits. Así pues, el formato queda con un bit para el signo, siete para el exponente y ocho para la mantisa.

3 [3 puntos]. Sea un computador de dos direcciones con modelo de ejecución registro-registro, con palabra de 32 bits y direccionamiento a nivel de byte que dispone únicamente de direccionamiento inmediato, directo a registro e indirecto a registro. Indique las secuencias de instrucciones en ensamblador IEEE para el computador indicado equivalentes a cada una de las instrucciones que se muestran a continuación. Si es necesario utilice los registros $RT1$, $RT2$ y $RT3$ como registros auxiliares.

`MOVE [/1000], #64[.R1--]`

`ADD [.R1], [/1000], #8[--.R2]`

`DBZ [.R1], [#4[.R2++]]` * Decrementa y salta si cero

SOLUCIÓN

```
MOV  [/1000], #64[.R1--]
      ; M(R1+64) <--- M(M(1000))
      ; R1 <--- R1 - 4

LD   .RT1, #1000
LD   .RT1, [.RT1]
LD   .RT1, [.RT1]
LD   .RT2, #64
ADD  .RT2, .R1
SUB  .R1 , #4
ST   .RT1, [.RT2]

ADD  [.R1], [/1000], #8[--.R2]
      ; R2 <--- R2 - 4;
      ; M(R1) <--- M(M(1000)) + M(R2+8)

LD   .RT1, #1000
LD   .RT1, [.RT1]
LD   .RT1, [.RT1]
SUB  .R2 , #4
LD   .RT2, #8
ADD  .RT2, .R2
LD   .RT2, [.RT2]
ADD  .RT1, .RT2
ST   .RT1, [.R1]

DBZ  [.R1], [#4[.R2++]]
      ; M(R1) <--- M(R1) - 1
      ; Salta si cero a M(M(R2+4)) es decir, PC <--- M(R2+4)
      ; R2 <--- R2 +4

LD   .RT1, #4
ADD  .RT1, .R2
LD   .RT1, [.RT1]
ADD  .R2 , #4
LD   .RT2, [.R1]
SUB  .RT2, #1
ST   .RT2, [.R1]
BZ   [.RT1]
```

NOTAS: 20 de julio de 2018
REVISIÓN: 23 de julio de 2018

DURACIÓN: 1 h y 30 minutos
PUNTUACIÓN: Especificada en cada ejercicio