

## EXAMEN OAC

### PARTE TEORICA

1. Una computadora RISC tiene la misma cantidad de instrucciones que un CISC si:

- a) El desensamblaje del algoritmo es aritmético y solamente se trabaja entre registros
- b) El lenguaje de programación es el mismo
- c) Ambas son correctas
- d) Ninguna de las anteriores

La diferencia principal entre las arquitecturas RISC (Reduced Instruction Set Computer) y CISC (Complex Instruction Set Computer) es la cantidad de instrucciones que utilizan. Las computadoras RISC tienen un conjunto reducido de instrucciones muy simples, mientras que las CISC tienen un conjunto de instrucciones más complejas y variadas.

Sin embargo, la cantidad de instrucciones no es lo único que determina la arquitectura de una computadora. En el caso de la pregunta, si el algoritmo utiliza solamente instrucciones aritméticas y se trabaja exclusivamente entre registros, entonces el desensamblaje de ese algoritmo sería idéntico en ambas arquitecturas. Por lo tanto, la respuesta correcta es la opción "a".

2. El ISA está basado en

- a) El hardware
- b) El sistema operativo
- c) El compilador
- d) Ninguna de las anteriores

El ISA (Instruction Set Architecture) o Arquitectura de Conjunto de Instrucciones es el conjunto de instrucciones que una computadora puede entender y ejecutar. El ISA está directamente relacionado con el hardware de la computadora, ya que define la forma en que se realizan las operaciones y la manera en que los programas se comunican con el procesador.

3. De las siguientes opciones:

Opción 1: Procesador de 1GHz con un programa que ejecuta a un CPI promedio de 1

Opción 2: Procesador de 2GHz con un programa que ejecuta a un CPI promedio de 2

- a) Opción 1
- b) Opción 2
- c) Son iguales ¿?
- d) Falta información

4. En un programa se sabe que solo existen operaciones aritméticas y flotantes, si se quiere hallar cuantas instrucciones aritméticas se realizan por segundo se debe saber:
- a) MIPS Y FLOPS
  - b) Numero de instrucciones aritméticas y tiempo de ejecución
  - c) Tiempo de ejecución, CPI y frecuencia del procesador
  - d) a y b
  - e) a y c
  - f) a,b y c
5. El riesgo de control esta asociado a
- a) Estructuras condicionales (if,else)
  - b) Ciclos iterativas (for,while)
  - c) Ambos
  - d) Ninguna de las anteriores

El riesgo de control se asocia a las estructuras de control de flujo que desvían el flujo normal de ejecución de un programa. Por lo tanto, la respuesta correcta es la opción "c) Ambos".

6. En un set de instrucciones MIPS (asumiendo los tiempos de ejecución vistos en clase por etapa), se realiza la siguiente porción de código:

```
...  
mul r1,r1,r2  
mul r2,r1,r1  
...
```

El método de pipeline sin forwarding:

- a) Disminuye el tiempo de ejecución
- b) Aumenta el tiempo de ejecución
- c) Mantiene el tiempo de ejecución
- d) Falta información

En este código, primero se realiza una multiplicación entre los registros r1 y r2, y el resultado se almacena en el registro r1. Luego, se realiza otra multiplicación entre los registros r1 y r1, y el resultado se almacena en el registro r2.

Si se utiliza el método de pipeline sin forwarding, las dos instrucciones de multiplicación se ejecutan en diferentes etapas del pipeline, y la segunda instrucción depende del resultado de la primera. Por lo tanto, cuando se ejecuta la segunda instrucción, el resultado de la primera aún no se ha almacenado en el registro r1, y debe esperar a que se complete la escritura del resultado en el registro.

7. El conocimiento de los riesgos asociados al pipeline es una ayuda para

- a) El desarrollo del compilador de lenguaje de alto nivel
- b) Encontrar fallas en el diseño del hardware**
- c) Justificar el uso de RISC sobre CISC
- d) Ninguna de las anteriores

El conocimiento de los riesgos asociados al pipeline es esencial para detectar fallas en el diseño del hardware y mejorar el rendimiento de los procesadores.

La alternativa "a) El desarrollo del compilador de lenguaje de alto nivel" no es la respuesta correcta porque los riesgos asociados al pipeline son problemas del hardware, no del software. El pipeline es una técnica utilizada en el diseño de hardware para mejorar el rendimiento del procesador al ejecutar múltiples instrucciones en paralelo. Por lo tanto, el conocimiento de los riesgos asociados al pipeline es importante para los diseñadores de hardware, no para los desarrolladores de software o compiladores.

8. La única forma de generar un ciclo iterativo en lenguaje ensamblador es a través de comparadores y saltos condicionales

- a) Verdadero
- b) Falso**

Aunque el uso de comparadores y saltos condicionales es común para implementar ciclos iterativos en lenguaje ensamblador, existen otras formas de lograrlo, como la implementación de ciclos utilizando instrucciones de salto incondicional (como el salto a la instrucción de inicio del ciclo) y el uso de registros como contadores.

9. Los registros XMM no tiene la capacidad de almacenar número enteros

- a) Verdadero
- b) Falso**

La afirmación es falsa. Los registros XMM son registros de propósito general que pueden almacenar tanto números enteros como números en punto flotante.

10. Si una función tipo void codificada en asm, es llamada e igualada a una variable tipo int desde C, dicha variable tomara el valor de retorno de que este en rax

- a) Verdadero
- b) Falso

Una función tipo void no tiene un valor de retorno, por lo que no asignará ningún valor a la variable a la que se le iguala. Además, el registro rax es utilizado para almacenar el valor de retorno de funciones que retornan un valor.

11. La función extern void función(char \*a,float \*\*b,int c, double d) recibe la información de los argumento de entrada de la siguiente manera:

- a) Primero: RDI  
Segundo: RSI  
Tercero: RCX  
Cuarto: XMM0
- b) Primero: RDI  
Segundo: XMM0  
Tercero: RSI  
Cuarto: XMM1
- c) Primero: RDI  
Segundo: XMM0  
Tercero: RCX  
Cuarto: XMM1
- d) Primero: RDI  
Segundo: XMM2  
Tercero: RCX  
Cuarto: XMM4
- e) Ninguna de las anteriores

12. Un sistema tiene una memoria principal de 16 MiB, direccionable por byte, y una memoria cache de 64 bloques con 32 bytes por cada bloque. Además, se utiliza mapeo asociativo por 8 vías. Por tanto, el formato para acceder a la cache es el siguiente:

- a) #bits tag: 10  
#bits set: 10  
#bits offset: 4
- b) #bits tag: 12  
#bits set: 18  
#bits offset: 4
- c) #bits tag: 9  
#bits set: 11  
#bits offset: 5
- d) #bits tag: 13  
#bits set: 6  
#bits offset: 5

e) Ninguna de las anteriores

#bits tag: 16  
#bits set: 3  
#bits offset: 5

13. Un sistema tiene una memoria principal, direccionable por byte, con una capacidad de 4 GiB. Así mismo, se sabe que tiene una memoria cache de 32 KiB. El tamaño de bloque es de 16 bytes cada uno y se utiliza mapeo directo. Por tanto el formato de conversión es:

- a) #bits tag: 10  
#bits set: 18  
#bits offset: 4
- b) #bits tag: 17  
#bits set: 11  
#bits offset: 4
- c) #bits tag: 17  
#bits set: 10  
#bits offset: 5
- d) #bits tag: 16

#bits set: 12  
#bits offset: 4

e) Ninguna de las anteriores

14. Considerando que “a” es un arreglo de 100 elementos enteros y que la memoria cache tiene un tamaño de bloque de 512 bytes, se le brinda la siguiente porción de código:

```
...  
for ( int i = 0; i < 20 ; i++){  
    printf( “ %d\n ”, a [ i ] )  
}  
...
```

Se puede afirmar que:

- a) Se aprovecha la localidad temporal del arreglo a
  - b) Se aprovecha la localidad espacial del arreglo a
  - c) Se aprovecha la localidad temporal y espacial de a
  - d) No se aprovecha ningún tipo de localidad en a
15. Un sistema es capaz de generar direcciones desde 0x00000000 hasta 0xFFFFFFFF. Se ha generado las direcciones desde 0XAAA1100 hasta 0XAAABFFFF de manera consecutiva. Si se sabe que la memoria cache contiene bloques de 128 bytes, se pide hallar el valor de la primera etiqueta si se utiliza mapeo asociativo.
- a) 2,795,588
  - b) 2,774,357
  - c) 1,397,794
  - d) Falta información
  - e) Ninguna de las anteriores
16. Cuando se utiliza la librería ctypes y se llama a funciones en C y ASM desde Python es cierto que:
- a) Se debe utilizar malloc en C si es que existe un error asociado al manejo de memoria.
  - b) Se podrá utilizar GDB sin problemas al ejecutar el intérprete de Python.
  - c) El uso de la memoria RAM es manejado por Python
  - d) Todas son falsas
  - e) a, b y c
17. Las graficas temporales que acumulan el tiempo de las funciones ejecutadas a través de ctypes:
- a) Requieren el uso de un filtro mediano para eliminar outliers.

- b) No requiere de la aplicación de ningún filtro si se tomo la mediana
- c) No requiere de la aplicación de ningún filtro si se tomo la media armónica.
- d) a, b y c
- e) solo b y c