



Universidad de Murcia
Facultad de Informática



Universidad Politécnica de Cartagena
Escuela Técnica Superior de Ingeniería de
Telecomunicación

TÍTULO DE GRADO EN
CIENCIA E INGENIERÍA DE DATOS
Fundamentos de Computadores

Tema 3: Estructura del computador

Boletín de autoevaluación de teoría / problemas

CURSO 2023 / 24

Índice general

| | |
|---|----------|
| I. Ejercicios resueltos | 2 |
| II. Ejercicios propuestos | 5 |
| III. Soluciones a los ejercicios resueltos | 6 |

Ejercicios resueltos

1. Un amigo tuyo tiene un iMac de Apple de hace unos años, con procesador Intel i5, y te gusta mucho una aplicación que tiene para manejar el correo electrónico. Tu amigo te dice que tiene los discos originales, y que te la copia para instalarla en tu PC de HP (procesador Intel i7 y SO Windows 11). Tú no tienes claro que le vaya a funcionar debido a:
 - a) La placa madre (motherboard) está realizada con circuitos electrónicos diferentes.
 - b) No hay ningún problema, pues ambos tienen un procesador de la misma familia de Intel y, por tanto, el ISA de los dos ordenadores es el mismo y las aplicaciones son intercambiables entre ellos.
 - c) El iMac de Apple trabaja en Hexadecimal, mientras que el PC de HP trabaja con números en octal.
 - d) Los Sistemas operativos son distintos (Windows y macOS).
 - e) Los dos ordenadores son de marcas distintas (uno es de Apple y el otro de HP).
2. Si el conjunto de instrucciones (ISA) de mi ordenador Intel i9 es x86_64, compatible con x86:
 - a) Puede ejecutar instrucciones del ISA MIPS, puesto que MIPS e Intel i9 pertenecen a la misma familia de procesadores.
 - b) Los programas que ejecuta deben ser de 16 bits, ya que x86 se diseñó para procesadores de 16 bits.
 - c) Mi ordenador puede ejecutar programas de 16, 32, 64 o 128 bits, ya que x86_64 es un ISA que apareció como evolución del x86.
 - d) Podría ejecutar un programa que funciona en un procesador de AMD contemporáneo, pues soportan x86_64.
3. Considera el ISA “de juguete” visto en clase en la asignatura y responde a las siguientes preguntas:
 - a) ¿Podría la evolución de la arquitectura de la CPU para incluir una instrucción adicional y ampliar el conjunto de instrucciones?
 - b) En caso de respuesta afirmativa para el caso anterior, ¿podría un programa compilado para la CPU anterior ejecutarse en la CPU evolucionada?
 - c) Realiza los cambios que estimes oportunos en el repertorio de instrucciones para añadir una instrucción que permita restar el valor del registro B al valor del registro A y describe cómo sería su funcionamiento.
 - d) Escribe un programa de ejemplo en código máquina con el ISA resultante que haga uso de las instrucciones disponibles para restar los números que se encuentran en las direcciones de memoria 0 y 1 de la RAM, y guardar su resultado en la dirección 15.
 - e) Utilizando los identificadores RA y RB para referenciar los dos registros disponibles, escribe el mismo código del ejercicio anterior, pero ahora en ensamblador, justificando la respuesta.
 - f) ¿Cómo se podría garantizar la compatibilidad hacia delante en el caso expuesto?
4. Las etapas de ejecución de una instrucción en un computador moderno son, en orden, las siguientes:
 - a) Carga de instrucción, decodificación, lectura de operandos, ejecución, escritura de resultado.
 - b) Carga de instrucción, ejecución, lectura del resultado, búsqueda de la siguiente instrucción.
 - c) Decodificación de la instrucción, ejecución, escritura del resultado, actualización del contador de programa.
 - d) Cálculo de la dirección de la instrucción, acceso a memoria, ejecución y decodificación del resultado.

5. Los valores inmediatos en una instrucción son:
- Valores que se recogen de memoria cuando la instrucción se ejecuta.
 - Valores que están codificados dentro de la propia instrucción.
 - Valores que se traen de la memoria caché del procesador, para que estén directamente accesibles de forma rápida.
 - Valores que se encuentran en los registros del procesador.
6. ¿Es posible que una instrucción tarde menos de 1 ciclo de reloj en ejecutarse?:
- No, porque cada instrucción se ejecuta en una serie de pasos, cada uno de los cuales tarda un ciclo de reloj.
 - Sí, porque hay etapas de la ejecución de las instrucciones que duran menos de un ciclo de reloj.
 - No, porque los procesadores modernos son superescalares.
 - Sí, porque un procesador puede duplicar unidades funcionales y paralelizar la ejecución de instrucciones.
7. Cuando decimos que los procesadores Intel i9 son superescalares y superpipeline es porque...
- Tienen una segmentación en muchas etapas y son capaces de terminar de ejecutar más de una instrucción por ciclo.
 - Es un procesador segmentado con etapas de ejecución muy largas.
 - Tiene multitud de unidades funcionales duplicadas.
 - Su diseño incluye muchos procesadores (*cores*) integrados para trabajar en paralelo, lo cual proporciona un rendimiento superior a un sistema monoprocesador.
8. Identificar el tipo de localidad en el acceso a memoria en cada uno de los casos siguientes:
- Ejecución de las mismas instrucciones dentro de un bucle.
 - Acceso a todos los bytes que codifican una imagen en memoria.
 - Recorrido de todos los datos de una lista en Python.
 - Lectura de la información de todos los datos guardados en un objeto Python relativo a una persona.
9. En una caché asociativa por conjunto de 2 vías (2 líneas por conjunto):
- Cada acceso al mismo conjunto siempre referencia al mismo bloque de memoria.
 - El acceso al conjunto solamente no determina el bloque de memoria al que se quiere acceder.
 - Se incluye un único tag por cada conjunto.
 - Se incluye un único bit de validez por cada conjunto.
10. Considera una caché asociativa por conjunto de 4 vías (4 líneas por conjunto), con un direccionamiento de caché formado por 2 bits de tag, 2 bits de identificador de conjunto y 2 bits de offset de byte en cada bloque. ¿Cuál es el tamaño de la caché?
11. Los puertos son...
- Controladoras del dispositivo.
 - Líneas de comunicación entre distintos componentes.
 - Sondeo, interrupciones y DMA.
 - Registros para E/S que no son de la CPU.

12. Si quiero desarrollar una aplicación donde el tiempo de espera por tareas de E/S puede ser muy alto, pero no quiero bloquear el desempeño del programa, ¿cuál sería el mecanismo de comunicación CPU-dispositivo más apropiado? Justifica la respuesta.
13. Estamos interesados en saber el tiempo de ejecución de un programa que contiene 356 instrucciones. El ordenador donde se ejecuta tiene una CPU que funciona a 2.5 GHz, si bien es capaz de ejecutar instrucciones a una media de 1.75 CPI.
14. En referencia a la velocidad de acceso a los distintos tipos de memoria que tiene un computador...
 - a) El tiempo de acceso a la memoria principal es más bajo que el tiempo de acceso a los registros de la CPU.
 - b) Mientras que la CPU es cada vez más rápida y el tiempo de acceso a los registros de la CPU es cada vez más bajo, el tiempo de acceso a los discos de almacenamiento secundario (disco “duro”) prácticamente se mantiene.
 - c) El tiempo de acceso a la memoria SRAM es mayor que el tiempo de acceso a la memoria DRAM.
 - d) Los discos en estado sólido (SSD) son igual de rápidos que una memoria DRAM o SRAM.
15. Para ayudar a una memoria ficticia de 64 MB tenemos una memoria caché de 32 KB, distribuida en 8 vías y usando un tamaño de bloque es de 32 bytes. ¿Cuál será el número de conjuntos de la caché? Considerando una caché inicialmente vacía, muestra la evolución de los primeros 4 conjuntos y si se producen aciertos o fallos, cuando se resuelven las siguientes direcciones de memoria: 0x0000, 0x001F, 0xF000, 0x0040, 0x0050, 0xA070.
16. ¿Qué es una rutina de servicio de interrupción (RSI)?
 - a) Es un conjunto de instrucciones que trata la aparición de una interrupción de E/S.
 - b) Es una función a la que se llama cuando se detecta un cambio en la E/S usando sondeo.
 - c) Es un sistema que no se usa cuando se hace uso del mecanismo de acceso directo a memoria (DMA).
 - d) Es un procedimiento con el código necesario para chequear continuamente el estado de un dispositivo de E/S.

Ejercicios propuestos

1. Describe brevemente las diferencias entre el ISA (Instruction Set Architecture) y la micro-arquitectura del procesador.
2. ¿Qué es mejor, una CPU CISC o una RISC? Justifica la respuesta.
3. ¿Qué hacen las instrucciones SIMD de los procesadores modernos?
 - a) Ejecutan operaciones sobre múltiples datos a la vez (paralelismo de datos).
 - b) Ejecutan varios tipos distintos de instrucciones de forma paralela.
 - c) Permiten hacer operaciones sobre escenarios en 3D en videojuegos, pero solo dentro de la tarjeta gráfica.
 - d) Ofrecen un mecanismo de paralelismo a nivel de tarea o hilo.
4. Los tipos de circuitos o componentes electrónicos que conforman las CPUs actuales son de los tipos:
 - a) Combinacionales, pues realizan computación sin estado.
 - b) Secuenciales, ya que es necesario almacenar bits.
 - c) Combinacionales y secuenciales.
 - d) Combinacionales, secuenciales y de control.
5. Explica brevemente los conceptos de concurrencia y paralelismo, e indica la diferencia que hay entre ambos.
6. Explica que es la jerarquía de memoria y qué elementos suele tener en los ordenadores actuales.
7. Resume para qué sirven los buses de dirección, control y datos.
8. Justifica por qué la entrada/salida de un ordenador puede ser, no solo con el usuario final, sino también con otras máquinas (ordenadores).
9. Explica qué es un driver y la función que tiene en el correcto funcionamiento de la E/S.
10. Sobre las tecnologías de almacenamiento utilizadas hoy en día:
 - a) Los discos en estado sólido se desarrollan con tecnología *flash*.
 - b) Los discos duros magnéticos convencionales suelen tener un sector por cada pista.
 - c) Las memorias volátiles mantienen el valor que almacenan cuando el ordenador se apaga.
 - d) Las memorias de tipo ROM están desarrolladas con tecnología que permite su acceso para escritura de manera eficiente.

Soluciones a los ejercicios resueltos

1. La respuesta correcta es la d), debido a que la aplicación en cuestión está compilada (traducida) para ser usada con un sistema operativo totalmente distinto (macOS). Las aplicaciones hacen uso de librerías propias del sistema operativo, y los archivos binarios generados por el compilador, ensamblador y enlazador están preparados para ejecutarse en macOS.
2. La respuesta correcta es la d). Si se usa un repertorio estándar de instrucciones x86_64, el programa podría ejecutarse tanto en un procesador de Intel como de AMD. Esto no sería así en caso de que el programa utilizara instrucciones propias de cada procesador, que podrían haberse añadido para acelerar ciertos cálculos. Notar que, si bien el procesador Intel i9 incluye las instrucciones x86_64, estas no son de 128 bits. Además, los procesadores modernos que han evolucionado a partir del isa x86, podrían ejecutar instrucciones de 16 o 32 bits, pero también soportan instrucciones más eficientes usando la arquitectura más reciente de 64 bits.
3. La respuesta a los distintos apartados son las siguientes:
 - a) Según la descripción dada del ISA en las clases de teoría, se hace uso de 4 bits para codificar el código de operación de cada una de las 4 instrucciones soportadas. Esto se hace para facilitar el desarrollo de la electrónica digital necesaria, pues solamente hace falta detectar un bit a 1 dentro de los cuatro disponibles para identificar una instrucción. No obstante, y aunque la electrónica necesaria sería un poco más compleja, con 4 bits se podrían codificar un total de 16 instrucciones. Por tanto, la respuesta es que sí se puede incluir una instrucción adicional.
 - b) Si la ampliación del ISA se realiza correctamente y las instrucciones anteriores siguen teniendo el mismo código de operación y realizando la misma función, la inclusión de una nueva operación no influiría en el funcionamiento de la CPU al ejecutar un programa que incluyese exclusivamente instrucciones del ISA original. Es por esto que la respuesta es que sí podría ejecutarse, ya que se garantizaría la compatibilidad hacia atrás.
 - c) Una propuesta posible de cambio podría ser añadir una instrucción con nemónico “SUB”, con código de operación 0011, ya que no está actualmente en uso. Como operandos, se usarán los 4 bits restantes de la instrucción para identificar los registros con los que se realizará la resta. Los primeros dos bits (los más significativos) identificarán el primer registro, mientras que los dos restantes (los menos significativos) nombrarán el segundo registro. La instrucción, de forma equivalente a la que tiene nemónico “ADD”, tomará los registros indicados como operandos, realizará la resta del primero menos el segundo, y guardará el resultado en el segundo registro. Se supone que la ALU que tiene la CPU ha sido modificada para soportar la resta de dos valores. El nuevo repertorio de instrucciones sería el siguiente:

| NEMÓNICO | OPCODE | OPERANDOS | DESCRIPCIÓN |
|----------|--------|--|--|
| LOAD_A | 0010 | Dir. memoria (4 bits) | Lee de RAM en registro A |
| LOAD_B | 0001 | Dir. memoria (4 bits) | Lee de RAM en registro B |
| STORE_A | 0100 | Dir. memoria (4 bits) | Escribe registro A en RAM |
| ADD | 1000 | ID registro A (2 bits) + ID registro B (2 bits) | Suma el valor de los registros indicados y guarda el resultado en el registro B |
| SUB | 0011 | ID registro 1 (2 bits) – ID registro 2 (2 bits) | Resta el valor de los registros indicados y guarda el resultado en el registro 2 |

- d) Se incluye a continuación un programa en código máquina que hace uso del ISA resultante para realizar la función requerida, utilizando hexadecimal para indicar el valor de los bytes en memoria:

0x20
0x11
0x34
0x4F

- e) Para las instrucciones en ensamblador usaremos los nemónicos listados en el conjunto de instrucciones mostrado anteriormente, los identificadores especificados para los registros, y las direcciones de memoria como valores inmediatos. Consideraremos que los operandos de cada instrucción se incluyen separados por comas, para los casos donde hay más de uno.

LOAD_A 0x0
LOAD_B 0x1
SUB RB,RA
STORE_A 0xF

- f) La compatibilidad hacia delante, por un lado, se permite en el ISA visto en clase al disponer de un conjunto de bits para el código de operación que soporta instrucciones adicionales. Sin embargo, para garantizar que los programas desarrollados con el nuevo ISA puedan funcionar sobre una CPU fabricada con el ISA original, sería necesario que todas instrucciones “SUB” fueran traducidas (emuladas). Una posibilidad para ello sería usar la instrucción “ADD”, cambiando previamente el valor del segundo registro a complemento a 2. No obstante, para que esto fuera factible, se debería disponer en el ISA original de alguna instrucción que permitiese realizar este cambio.

4. Según hemos visto en clase, la secuencia de etapas corresponde con la respuesta a), seguido de una actualización de la siguiente instrucción que se iría a buscar.
5. Según hemos visto en clase, la respuesta correcta es la b), puesto que los valores inmediatos se incrustan en la instrucción que se ha cargado para ejecutarse.
6. La respuesta correcta es la d), puesto que hemos visto que hay procesadores capaces de segmentar las instrucciones y terminar de ejecutar más de una instrucción por ciclo. Son los llamados procesadores superescalares.
7. La respuesta correcta es la a), que define los dos conceptos relativos a un nivel superior de desarrollo del concepto de procesador segmentado para conseguir ejecutar múltiples instrucciones por ciclo.
8. Los tipos de localidad que se identifican en los casos expuestos son:
 - a) Ejecución de las mismas instrucciones de un bucle. Localidad temporal, en este caso en el acceso a instrucciones.
 - b) Acceso a todos los bytes que codifican una imagen en memoria. Localidad espacial, en este caso en el acceso a datos.
 - c) Recorrido de todos los datos de una lista en Python. Localidad espacial, relativa al acceso a datos.
 - d) Lectura de la información de todos los datos guardados en un objeto Python relativo a una persona. Localidad espacial, relativa al acceso a datos.
9. La respuesta correcta es la b), puesto que para cada conjunto disponibles de dos líneas (recordar que la caché es de dos vías), y habría que comparar también los tags para buscar el bloque de memoria al que se quiere acceder y comprobar si está en alguna de la dos líneas.
10. El cálculo del tamaño de una caché se calcula según $C = S * E * B$, siendo S el número de conjuntos, E el número de vías (líneas por conjunto) y B el número de bytes por bloque. En nuestro caso, S tiene valor 4, E tiene valor 4, y B tiene valor 4. Por lo tanto $C = 64bytes$.

11. Según hemos visto en clase, la respuesta correcta es la d), pues los puertos son registros externos a la CPU donde se puede acceder para controlar la E/S, y que están integrados en la controladora del dispositivo.
12. El sondeo estaría descartado, por requerir demasiado tiempo para las comprobaciones continuas en el puerto de E/S. Sería más apropiado poder usar interrupciones. Sin embargo, no se especifica la cantidad de datos a transferir, y es un factor determinante. En caso de necesitar transferir unos pocos bytes, un mecanismo básico de interrupciones sería suficiente. Sin embargo, en caso de necesitar un ancho de banda significativo, es más recomendable usar DMA, pues permite delegar el trabajo de transferencia de datos a la controladora de DMA, quien coordina un acceso directo entre el dispositivo y la memoria sin la intervención del procesador.
13. Para calcular el tiempo de ejecución de un programa, tenemos que saber el número de instrucciones del programa, y luego llevar cuenta de la capacidad de la CPU que usamos, considerando su frecuencia de funcionamiento y los ciclos que tarda de media en ejecutar cada instrucción. En verdad, cada CPU tarda un número de ciclos variable según qué tipo de instrucción ejecutemos, pero vamos a obviar este hecho. Por tanto $t_{exe}(seg) = ins/programa * ciclos/ins * seg/ciclo$. En nuestro caso tenemos $t_{exe}(seg) = 356 * 1,75 * (1/2,5 * 10^9) = 0,0000002492$. Pasando el resultado a una unidad apropiada, serían 249,2 ns.
14. La respuesta correcta es la b), ya que la tendencial actual es que los discos duros convencionales se han mantenido en su tiempo de acceso, mientras que la CPU y las memorias próximas SRAM y DRAM han mejorado en velocidad y tiempo de acceso, respectivamente.
15. Las distintas peticiones a memoria generarían el siguiente resultado:
 - a) 0x0000. Fallo. Se trae a caché el bloque M[0-31] y se mete en el conjunto 0.
 - b) 0x001F. Acierto. Coincidencia de tag en conjunto 0.
 - c) 0xF000. Fallo. Se trae a caché el bloque M[61440-61471] y se mete en el conjunto 0.
 - d) 0x0040. Fallo. Se trae a caché el bloque M[64-95] y se mete en el conjunto 2.
 - e) 0x0050. Acierto. Coincidencia de tag en el conjunto 2.
 - f) 0xA070. Fallo. Se trae a caché el bloque M[41056-41087] y se mete en el conjunto 3.

No se producen reemplazamientos porque ninguno de los conjuntos se llega a llenar. El estado de la caché tras las lecturas de estas posiciones de memoria es el siguiente:

| | | | | | | | | | |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---------------------------|---------------------------|
| Set 0 | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | 1 - 1111 - M[61440-61471] | 1 - 0000 - M[0-31] |
| | | | | | | | | | |
| Set 1 | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | | |
| | | | | | | | | | |
| Set 2 | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | | 1 - 0000 - M[64-95] |
| | | | | | | | | | |
| Set 3 | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | v - tag - block | | 1 - 1010 - M[41056-41087] |
| | | | | | | | | | |

16. La respuesta correcta es la a), según hemos visto en clase.