Compendio de respuestas segundo parcial

Arquitectura de Computadoras

2022

Recordar que:

- **El parcial consta de 5 preguntas.**
- **Para aprobar el curso debe contestar correctamente una pregunta en cada parcial.**
- **Para obtener la exoneración parcial debe contestar correctamente tres preguntas en cada parcial.**

Pregunta 1

Fragmento 1: dado que el xor actualiza las banderas, la bandera Z valdrá 1 luego de su ejecución, por lo tanto el salto no se tomará (jnz salta si $Z=0$), provocando que la instrucción inc cx se ejecute una única vez.

Fragmento 2: la primera vez que se ejecuta el salto jnz, se hace con el valor $CX=1$, por lo tanto, dado que inc también actualiza la bandera Z, $Z=0$ y por lo tanto el salto se toma. La segunda vez se ejecutará con $CX=2$ y por tanto también Z valdrá 0, con lo cual se tomará nuevamente el salto. Así sucesivamente hasta que $CX=0$ (y por lo tanto $Z=1$), lo cual se dará cuando CX pase de 0xFF a 0, al dar overflow la instrucción inc, lo cual sucederá en su ejecución número 2^{16}.

Pregunta 2

a) Líneas INT/IRQ independientes en la CPU, Mecanismo INT/INTA, Controlador de Interrupciones
b) x86 utiliza un controlador de interrupciones. Cuando un controlador de E/S solicita una interrupción, el controlador de interrupciones genera el pedido a la CPU a través de la señal INT. La CPU cuando acepta la interrupción activa la señal INTA y el controlador de interrupciones coloca en el bus de datos la identificación asociada a la entrada IRQ por la que llegó el pedido.

Pregunta 3

Cada casillero en las tablas representa un byte.

`short entero`:

dirección	dato
0x300	entero (parte baja)
0x301	entero (parte alta)

`char letra`:

dirección	dato
0x300	letra

`short arregloEntero[2]`:

dirección	dato
0x300	entero[0] (parte baja)
0x301	entero[0] (parte alta)
0x302	entero[1] (parte baja)
0x303	entero[1] (parte alta)

`struct letrasEnteros`:

dirección	dato
0x300	letras[0]
0x301	letras[1]
0x302	entero(parte baja)
0x303	entero(parte alta)

`letrasEnteros arregloRegistros`:

dirección	dato
0x300	arregloRegistros[0].letras[0]
0x301	arregloRegistros[0].letras[1]
0x302	arregloRegistros[0].entero(parte baja)
0x303	arregloRegistros[0].entero(parte alta)
0x304	arregloRegistros[1].letras[0]
0x305	arregloRegistros[1].letras[1]
0x306	arregloRegistros[1].entero(parte baja)
0x307	arregloRegistros[1].entero(parte alta)

Pregunta 4

```
cmp byte ptr ES:[BX], 0
```

Esta instrucción efectúa la resta entre el byte contenido en la dirección ES * 16 + BX, con 0, y actualiza el registro de estado (flags) con los valores correspondientes según el resultado de la resta. Los modos de direccionamientos son:

ES:[BX] -> Indirecto por registro.
0 -> Inmediato.

```
pop AX
```

Esta instrucción carga la palabra de 16 bits en la dirección SS * 16 + SP en AX y aumenta en 2 el valor de SP.
El modo de direccionamiento es directo a registro.

```
mov DL, DS:[0x30]
```

Esta instrucción carga el byte contenido en la dirección DS * 16 + 0x30 en DL. Los modos de direccionamiento son:

DL -> Directo a registro

[0x30] -> Directo a memoria

Pregunta 5

El problema de coherencia de caché sucede cuando se pierde el sincronismo entre el contenido de la caché y el de la memoria principal, afectando la consistencia en los datos. Lo importante no es la falta de sincronismo, si no el efecto no deseado sobre la consistencia de los datos vistos por distintas partes del sistema que los utilizan.

El problema se puede dar en distintas situaciones:

1. Cuando se utilizan sistemas de acceso directo a memoria (DMA) para optimizar los intercambios de datos entre los dispositivos de entrada/salida y la memoria principal, el problema se da, por ejemplo, cuando el DMA escribe en la memoria principal en un bloque que está almacenado en la caché.

2. Cuando se utiliza cache con write-back, la cache pierde el sincronismo con la memoria principal cada vez que se realizan escrituras, hasta que se reemplaza el bloque y se actualiza en memoria. El problema de consistencia se puede dar cuando el DMA lee en un bloque almacenado en la caché que fue modificado por una escritura en el cache.

3. En sistemas multi-procesador con memoria compartida se dan los mismos problemas que suceden al usar DMA, cuando un procesador lee o escribe en bloques de memoria almacenados en la cache de otro procesador y además, en el caso de write-back, se puede dar la situación que el mismo bloque de memoria esté simultáneamente en más de una cache y se desincronizará su valor cuando se escribe en las caches.

Para el caso del cache con write back, si el problema ocurre en una lectura, el problema se resuelve deteniendo el DMA para actualizar la memoria con el contenido de la caché. Si se trata de una escritura del DMA, el problema se resuelve invalidando la entrada del caché y el próximo acceso al bloque producirá un fallo (“miss”) provocando la actualización de la cache.

Pregunta 6

Un hazard de control es un problema que ocurre en una CPU con pipeline, cuando al encontrarse con una instrucción de salto, se deben descartar instrucciones previamente cargadas por no ser parte del flujo lógico del programa.

Salto demorado y predicción de saltos son dos técnicas para mitigar este problema. La técnica de salto demorado consiste en que el pipeline siempre ejecute la instrucción posterior a un salto, y de este modo, se evita descartar instrucciones. Para no alterar el orden lógico del programa, el compilador (o el programador de bajo nivel) se encarga de colocar una instrucción útil en dichas posiciones de memoria (en caso de no encontrarse una instrucción útil se puede colocar una instrucción NOP). Es decir, esta técnica sí requiere asistencia del compilador (o del programador de bajo nivel).

Predicción de saltos es una técnica en la cual se agrega hardware especializado encargado de intentar predecir si un salto condicional se tomará o no, según diferentes criterios pero usualmente en función de los resultados anteriores de ese salto (si ya se tomó), y/o de otros saltos en el programa. El compilador no interviene en esta técnica ya que se implementa en hardware

Pregunta 7

Como la memoria caché es de 64 KB (216 bytes) y el tamaño de bloque es de 16 bytes (24 bytes), entonces el caché tiene 4096 líneas (212). Si la memoria es de 4GB entonces podemos asumir que la dirección es de 32 bits.

Para el caso de función de correspondencia totalmente asociativa, la dirección se divide en TAG y BYTE. El campo byte, dado que la memoria tiene bloques de 16 bytes y la memoria es direccionable por bytes, requiere 4 bits para su direccionamiento, por lo tanto los restantes 28 bits forman el campo TAG.

31 - - TAG - - 4	3 - - BYTE - - 0

Para el caso de función de correspondencia directa, la dirección se divide en TAG, LÍNEA y BYTE. El campo línea, dado que el caché tiene 4096 líneas, ocupa 12 bits, y por tanto el campo tag ocupa los restantes 16.

31 - - TAG - - 16	15 - - LÍNEA - - 4	3 - - BYTE - - 0

Por último, para el caso de correspondencia asociativa por conjuntos de n vías, la dirección se divide en TAG, CONJUNTO y BYTE. Para este caso particular de 8 vías, dado que el caché tiene 4096 líneas, este tiene 4096 / 8 = 512 conjuntos. Por lo tanto el campo conjunto ocupa 9 bits. Los 19 bits restantes forman el TAG.

31 - - TAG - - 13	12 - - CONJUNTO - - 4	3 - - BYTE - - 0

Pregunta 8

La comunicación de la CPU con los controladores de E/S se realiza a través de posiciones de memoria especiales de estos dispositivos, que son accesibles también para la CPU. Existen dos tipos de arquitecturas de E/S respecto a la forma de acceso desde la CPU:
1. La CPU dispone de un espacio de direcciones reservado para la E/S, que es accedido por instrucciones especiales (in y out). Un ejemplo de este enfoque es la familia de procesadores Intel.
2. La CPU accede a los controladores de E/S como si se trataran de posiciones normales de memoria, utilizando para tales fines cualquier instrucción que acceda a memoria. Un ejemplo de este diseño es la familia de procesadores SPARC y otros diseños RISC.

Pregunta 9

Stack:

dir: SS:0x3EFA valor: 0x3F00

dir: SS:0x3EFC valor: 0x0ACA

dir: SS:0x3EFE valor: 0x00BA

BP = 0x3EFA

SP = 0x3EFA

Pregunta 10

En el bus de direcciones se ve el número de puerto al cual se está accediendo y en el bus de datos aparece el dato a escribir en el caso de una escritura, o el valor del puerto de entrada salida leído en el caso de una lectura.

Si se trata de E/S aislada, en el bus de control se enciende las señales IO_RD o IO_WR en caso de lectura o escritura respectivamente, mientras que si la E/S está mapeada a memoria, se encienden los bits de MEM_RD y MEM_WR en su lugar.