



Solución Examen

Pregunta 1

- a) Modifique la arquitectura del computador básico, para que funcione con lógica ternaria en vez de binaria. Más específicamente, modifique los tamaños de los elementos (buses, registros, señales de control, etc) de modo que el nuevo computador tenga una capacidad similar a la versión binaria. Asuma que existen todos los componentes vistos en clases en versión ternaria. **Nota:** No es válido utilizar los valores ternarios como si fueran binarios. **(3 ptos.)**

Solución: Si bien la principal modificación era el redimensionamiento de los buses, se esperaba mencionar que cambia el tamaño de los registros y algunas señales de control. La fórmula usada es $\lceil \log_3(2^n) \rceil$ para un bus binario de n bits. En general no debe perderse rango de números posibles, ganar rango es indiferente. Se aceptó para los buses de 8 bits tamaños de 5 o 6 trits, sin embargo, lo más correcto es 6 trits. Deben reducir la cantidad de señales de control de la ALU a 2 trits, los muxer de 3 elementos o menos a 1 trit, u muxer de 4 elementos a 2 trits. NO deben redimensionar el reg. status.

- b) Agregue a la microarquitectura del computador básico los elementos necesarios para soportar multiprogramación, tanto desde el punto de vista de memoria como de CPU. **(3 ptos.)**

Solución: e debe agregar una MMU, TLB, PTBR, Supervisor mode reg., Soporte para I/O, RTC.

Debe añadirse algún mecanismo para respaldar el estado de los registros para cambios de contexto, donde lo más sencillo es respaldar a memoria.

Se espera que mencionen las conexiones de los elementos, donde el Supervisor mode Reg. se conecta a la CU y con un enabler a las señales de write para la TLB y el PTBR, adicionalmente esta señal debe ser leída por la MMU para activar o desactivar la traducción de direcciones de memoria.

La MMU se debe conectar antes de la memoria principal, interviniendo principalmente el bus de Address. La MMU se conecta con la TLB y el PTBR. Idealmente deberían haber diseñado la MMU, como un computador similar al PC Básico y que corre un programa simple para realizar traducciones.

Para el soporte I/O, se puede añadir un address decoder al bus de direcciones de memoria, y una conexión directa a algún mux de entrada al PC. Para interrupciones, por ejemplo del RTC, es posible conectar un bit de interrupción a la CU, decir que esta ahora será de tipo microcode y que detonará una serie de operaciones.

Note que el PCB es un problema de software y por tanto, dando un mecanismo válido para respaldar el estado de la CPU, no se considera para esta pregunta. La conexión de un HDD para realizar swapping no es obligatoria y por tanto no se considera.

Pregunta 2

- a) En un computador con múltiples niveles de memoria caché, ¿es necesario que todos compartan las mismas características (tamaños totales, tamaño de línea, correspondencia, sustitución, división datos/instrucciones, etc.)? En cualquiera de los dos casos, indique y justifique cuáles podrían ser valores para estas características. ¿Cambiarían estos criterios si el computador tiene más de un procesador? ¿Qué partes de la caché compartiría? **(3 ptos.)**

Solución: No, no es necesario que en un computador con múltiples niveles de memoria caché se compartan características:

- Tamaños de la memoria: Conviene que la capacidad de cada nivel aumente mientras aumenta el nivel para aprovechar de mejor manera la localidad espacial.
- Tamaño de línea: Conviene que se comparta el tamaño de línea entre niveles de caché para que sea más sencillo el traspaso de información entre niveles.
- Correspondencia y sustitución: Da igual que correspondencia y sustitución utilicen los distintos niveles de memoria caché.
- División datos/instrucciones: Conviene que la memoria más cercana al procesador (L1) sea split y el resto de los niveles unified.

Si hay más de un procesador no cambiarían las características, pero si convendría que cada procesador tuviera al menos su propio nivel L1 y que se comparta al menos uno de los niveles superiores de memoria entre procesadores.

- b) Un sistema operativo implementa un protocolo de seguridad de 2 etapas. En la primera, los **programas** son analizados por un antivirus/antimalware, mientras que en la segunda, cada una de las instrucciones de los **procesos** es analizada antes de ser ejecutada por el procesador. Indique como implementaría la segunda etapa en un computador con soporte para todos los esquemas de I/O vistos en clases. **(3 ptos.)**

Solución: Para la segunda etapa basta con el siguiente procedimiento:

1. Se guarda el valor de la instrucción en la posición $PC + 1$.
2. Se genera una interrupción de hardware que le entrega el control al sistema operativo.
3. El sistema operativo realiza todas las acciones necesarias para analizar la instrucción guardada.
4. Si el análisis arroja que la instrucción no es maliciosa, se devuelve el control al programa actual y se ejecuta la instrucción analizada. En caso contrario, se toma alguna medida que evite la ejecución del programa.

Pregunta 3

- a) Las instrucciones en una ISA RISC, sin tomar en cuenta su etapa *fetch*, toman un ciclo de clock. Asumiendo que puede realizar cambios a la microarquitectura de un computador x86, indique como podría transformar en tiempo de ejecución cada instrucción de la ISA x86 a una serie de instrucciones RISC. Relacione esto con la implementación de un pipeline. **(3 ptos.)**

Solución: Una posible solución es incluir la capacidad por hardware, posiblemente dentro de la Control Unit, que registre una instrucción compleja como varias instrucciones sencillas del tipo mover de registros a memoria o viceversa y operar registros y regule el paso de estas, por ejemplo, deteniendo el flujo de la lectura de instrucciones mientras prosigue con . Esta separación en sub-instrucciones se asemeja a la idea del *pipeline*.

No es una opción hacer esto desde el *assembler*, pues se pide expresamente que sea una implementación en *hardware*. Tampoco lo es alterar los tiempos de ejecución de las instrucciones más lentas/rápidas para equipararlas con las demás, pues sigue teniéndose el mismo esquema a fin de cuentas.

En general, los puntajes se asignan de la siguiente forma:

- Identificar por qué las instrucciones toman más tiempo (i.e acceder varias veces por memoria), [0.5 pts] ó
- identificar a qué tipo de instrucciones se subdividen las complejas (ver solución propuesta). [0.5 pts]
- Menciones a la solución propuesta de forma incompleta (i.e dejar muy en el aire cómo se logra), [0.75 pts] ó
- responder correctamente, sea lo propuesto u otra solución válida. [1.5-2 pts, según se asignó el parcial de 0.5]
- Relacionar con el *pipeline* correctamente. [1 pts, y 0.5 si no está muy bien]

- b) Un fabricante de dispositivos móviles de reproducción de audio, está intentando realizar una importante reducción de costo en sus equipos, mediante la eliminación del soporte de hardware de números de punto flotante, sólo dejando soporte nativo para números enteros. Tomando esto en consideración, indique y diagrame cómo se puede implementar la multiplicación de números de punto flotante utilizando sólo elementos que trabajen con números enteros (sumadores, multiplicadores, etc.). Ponga especial cuidado en los tamaños de los números, para no perder precisión ni rango. **(3 ptos.)**

Solución: La respuesta es evidente: diseñar un componente que realice la operación de la multiplicación en números de punto flotante, es decir, implemente en *hardware* el algoritmo con el que se realiza esta operación:

- Manejo del signo [0.5 pts]
- Suma de los exponentes (sin olvidar que $exp_{registrado} = exp_{real} + 127$ en el caso de 32 bits) [1 pts]
- Multiplicación de las mantisas (considerando que no se registra el primer 1) [1.5 pts]

Por supuesto, la respuesta varía si es que se define otro formato de punto flotante, en cuyo caso también debe estar explicitado.