



IIC2343 – Arquitectura de Computadores (II/2018)

Examen

El examen se compone de 5 preguntas. Debe responder cada una en una hoja separada, aunque puede utilizar más de una hoja para responder a una pregunta. Si no responde a una pregunta, debe entregar una hoja en blanco para ella. Recuerde poner su nombre en todas las hojas de respuesta.

Al momento de entregar, **evite doblar bordes o corchetear las hojas**, si no que sencillamente indique cuántas hojas son para esa pregunta y qué hoja es cada una en caso de utilizar más de una, procurando que esté lo más lisa posible. Todas las hojas serán escaneadas y las correcciones del examen se harán sobre lo escaneado.

Pregunta 1

Considere el siguiente programa:

```
a = 10
b = 20
res = 0
while a > 0:
    res = res+b
    a = a-1
```

Asumiendo que las variables **a**, **b** y **res** almacenarán números de a lo más 8 bits, responda las siguientes preguntas:

- a) Construya un circuito secuencial que implemente el programa. **(3 ptos.)**

Respuesta:

Una solución aceptada con todo el puntaje es que indiquen que usarán el computador básico con el programa escrito en assembler. No es necesario que escriban el código.

Otra es que implementen un circuito ad-hoc que debiese implementar dos registros para los inputs y un acumulador para el output y una ALU.

- b) Construya un circuito combinacional que implemente el programa **(3 ptos.)**

Respuesta:

Acá se admite la opción de que expresen parcialmente que usarán algún decoder para señalar 2^8 pines donde cada uno estará asociado a una celda de mux que selecciona con el pin o un operando o un 0 y eso conectado a un sumador. Al conectarlos en serie se obtiene lo pedido.

Otra opción es que implementen los sumadores de forma escalonada usando los bits de un lado para saber si suman en esa zona el otro operando o rellenan con 0, y los bits que hagan falta se rellenan con 0.

Pregunta 2

Considere un computador Von Neumann, al que se le desea conectar un acelerador matemático que permita soportar de manera nativa operaciones de punto flotante.

- a) Diseñe la microarquitectura del coprocesador, de manera que contenga al menos 8 registros de 80-bits, y soporte para operaciones aritméticas. Puede utilizar cualquier componente visto en clases. **(2 ptos.)**

Respuesta:

En general acá no se puede dar una pauta concreta, si no que se debe analizar que la respuesta entregada tenga sentido. Se permite el uso de un bloque FPU y en general se espera que hayan instanciado 8 registros de 80-bits conectados todos a 2 muxes, y cada mux es una entrada de la FPU. Luego la salida de resultado debiese retroalimentar los 8 registros.

Además algunos alumnos en función de la parte c) podrían haber incorporado algunos buffers y registros de comunicación.

- b) Indique como conectaría el dispositivo al computador Von Neumann. Incluya un diagrama de alto nivel. **(1 pto.)**

Respuesta:

Se espera el diagrama de IO general que se pone en clases. Además debiesen explicar como planean conectar el coprocesador (ports o memory-mapping, ambos son admitidos).

- c) Indique cómo serían entregados los datos y comandos al coprocesador. Detalle el tipo de comunicación, modo de interacción e incluya un diagrama. Es posible que, al contestar este ítem, deba modificar su respuesta de los anteriores. **(3 ptos.)**

Respuesta:

Se debe vigilar que esta respuesta sea concordante consigo misma. Además pueden proponer cambios o extensiones a lo anterior.

En general se esperaría que definan un registro de comandos y dos de parámetros para realizar las operaciones y establezcan una tabla de comandos y estado (75% del puntaje). Lo restante se debe asignar en función de cómo planteen el problema de transferir los 80 bits.

Pregunta 3

Sobre un computador Von Neumann con *memory-mapped* I/O se desea implementar memoria virtual con la capacidad de designar páginas en las que se puede ejecutar código y otras en las que no se puede ejecutar. Al respecto:

- a) ¿Cómo lograría que una página en particular no pueda ser ejecutada? **(3 ptos.)**

Respuesta:

La respuesta buscada es que agreguen un bit de metadata a la tabla de páginas que señale si dicha página es ejecutable o no. En la realidad dicho bit es el NX. Luego para el avance del PC deberían verificar (o al enviar el address) dicho bit en la tabla y si es no-ejecutar levantar una excepción.

- b) ¿Qué piezas deben ser agregadas para soportar este esquema? **(2 ptos.)**

Respuesta:

Para soportar memoria virtual deberían agregar al menos la MMU, que se compone de registro PTBR y un Adder, además de TLB. Ausencia de TLB descontar 0.1. Además para la revisión del bit se requiere un comparador (puerta AND).

- c) ¿Tiene sentido hacer esto para un computador Harvard? Comente. **(1 pto.)**

Respuesta:

La respuesta es no, dado que las memorias de instrucciones y datos están completamente separadas.

Pregunta 4

Considere un computador **x86** monoprocesador, que posee un *pipeline* de N etapas. En base a esto, conteste las siguientes preguntas:

- a) ¿Qué tipo de *hazard* podría generarse entre las etapas **Fetch** y **Mem**? (1 pto.)

Respuesta:

Hazard estructural ya que ambas etapas compiten por la **RAM**

- b) Considere que se agrega soporte para multiprogramación al computador (*preemptive scheduling*), con la restricción de que sólo se puede hacer cambio de contexto una vez que todas las etapas de la **instrucción en ejecución** hayan terminado. ¿Qué modificaciones se deben hacer al computador para soportar esto? Considere que el computador ahora debe soportar memoria virtual y *CPU scheduling*. (2 ptos.)

Respuesta:

Además de que deben señalar que en **MEM** deberán agregar la MMU y TLB, deben señalar que tendrán que agregar una unidad de control para context-switch que haga stall del pipeline hasta que la última instrucción pasa por todo el pipeline antes de hacer el cambio (cambiar PTBR, cambiar TLB, cambiar PC).

- c) Considere que ahora se relaja la restricción del cambio de contexto, permitiendo que se pueda hacer en cualquier momento (no es necesario que todas las etapas de la instrucción actual hayan terminado). En base a esto conteste lo siguiente:

- i. Desde el punto de vista del contenido del *pipeline*, ¿cuál es la diferencia con el caso anterior? (1 pto.)

Respuesta:

Ahora habrá mas de un proceso en el pipeline potencialmente, pero no se perderan más ciclos que los estrictamente necesarios por el cambio de contexto

- ii. Desde el punto de vista de tiempo de ejecución, ¿cuál esquema es preferible? (1 pto.)

Respuesta:

Este segundo esquema es preferible ya que pierde menos ciclos.

- iii. ¿Qué modificaciones se deben hacer al computador para soportar este esquema? (1 pto.)

Respuesta:

O bien en los registros intermedios ahora se deberán transportar la TLB y el PTBR o bien deberán hacer que en el cambio de contexto se guarden los registros intermedios en el PCB.

Pregunta 5

El siguiente programa se ejecutó en un computador con arquitectura **x86** que tiene una *caché* de 4 líneas de 2 palabras cada una:

Dirección	Label	
0	loop1:	MOV var2,1
1		MOV AL,var2
2		MUL var2
3		CMP var1,AL
4		JL end
5		INC var2
6	end:	JMP loop1
7		DEC var2
8	var1	RET
9		db ?
10	var2	db 0

Al ejecutar el programa completo se obtuvo la siguiente secuencia de accesos a memoria:

0-10-1-10-2-10-3-9-4-5-10-6-1-10-2-10-3-9-4-5-10-6-1-10-2-10-3-9-4-7-10-8

Esta secuencia de accesos a memoria generó los siguiente estados en la *caché*:

Dir	L0	L1	L2	L3
0	0-1			
10	0-1	10-11		
1	0-1	10-11		
10	0-1	10-11		
2	0-1	10-11	2-3	
10	0-1	10-11	2-3	
3	0-1	10-11	2-3	
9	0-1	10-11	2-3	8-9
4	0-1	10-11	2-3	4-5
5	0-1	10-11	2-3	4-5
10	0-1	10-11	2-3	4-5
6	6-7	10-11	2-3	4-5
1	6-7	10-11	0-1	4-5
10	6-7	10-11	0-1	4-5
2	6-7	10-11	0-1	2-3
10	6-7	10-11	0-1	2-3

Dir	L0	L1	L2	L3
3	6-7	10-11	0-1	2-3
9	8-9	10-11	0-1	2-3
4	8-9	10-11	4-5	2-3
5	8-9	10-11	4-5	2-3
10	8-9	10-11	4-5	2-3
6	6-7	10-11	4-5	2-3
1	0-1	10-11	4-5	2-3
10	0-1	10-11	4-5	2-3
2	0-1	10-11	4-5	2-3
10	0-1	10-11	4-5	2-3
3	0-1	10-11	4-5	2-3
9	8-9	10-11	4-5	2-3
4	8-9	10-11	4-5	2-3
7	6-7	10-11	4-5	2-3
10	6-7	10-11	4-5	2-3
8	8-9	10-11	4-5	2-3

En base a esta información, responda lo siguiente:

- ¿Qué valores puede tener la variable **var1** para que efectivamente se genere la secuencia de accesos detallada previamente? **(2 ptos.)**
- ¿Cuál es el *hit-rate*? **(1 pto.)**
- ¿Qué tipo de *caché* es: *unified* o *split*? **(1 pto.)**
- ¿Qué función de correspondencia y algoritmo de reemplazo utiliza esta *caché* ? **(2 ptos.)**