



IIC2343 – Arquitectura de Computadores (II/2012)

**Solución Examen**

**Pregunta 1**

- a) ¿Cuáles fueron los aporte de Shannon al desarrollo de los computadores? **(1 pto.)**

**Solución:** Shannon logró implementar físicamente circuitos lógicos de lógica booleana mediante relés. Para esto definió, por cada conector lógico binario, una compuerta lógica que luego era implementada mediante relés.

- b) Comente sobre los problemas relacionados con los números de punto flotante y los números reales. **(1 pto.)**

**Solución:** Los números de punto flotante no pueden representar de manera completa y correcta todos los números reales, debido a que los números de punto flotante tienen un espacio acotado para representar una cantidad posiblemente infinita de cifras. Además, el problema de números infinitos inherente a los sistemas posicionales agrava esta situación.

- c) Describa al menos 2 ventajas de la arquitectura Harvard por sobre la Von Neuman. **(1 pto.)**

**Solución:** La arquitectura Harvard, al tener memorias de datos e instrucciones separadas, puede realizar la lectura de instrucciones y de datos de manera simultánea. Además, esto permite evitar hazards estructurales por acceso simultáneo a instrucciones y datos.

- d) ¿Qué implicancia tiene en el tamaño de los programas el eliminar la conexión entre memoria de datos y PC en el computador básico? **(1 pto.)**

**Solución:** El eliminar esta conexión implica eliminar el soporte para subrutinas, por lo que en la gran mayoría de los casos los programas tendrán un mayor tamaño debido a la repetición de código.

- e) ¿Es siempre el esquema de interrupciones el más eficiente para transferir datos con los dispositivos? Justifique su respuesta en cualquiera de los dos casos. **(1 pto.)**

**Solución:** No necesariamente. Existen dispositivos como el teclado donde las transferencias son tan pocas y pequeñas que el *overhead* del uso de interrupciones hace el proceso más lento.

f) ¿En que se basa el concepto de memoria caché para ser efectivo en la práctica?. (1 pto.)

**Solución:** Se basa en los principios de localidad espacial y localidad temporal.

## Pregunta 2

- a) Un computador tiene un espacio direccionable de 36 bits y una memoria física de 2GB, dividida en marcos de 2KB. ¿Cuántas páginas virtuales y marcos físicos tiene este computador? Considerando los flags vistos en clases, ¿qué tamaño tiene una tabla de páginas? **(2 ptos.)**

**Solución:**

- N° páginas:  $\frac{2^{36}}{2^{11}} = 2^{25}$
  - N° marcos:  $\frac{2^{31}}{2^{11}} = 2^{20}$
  - Tamaño páginas:  $2^{25} \times (20 + 2)$  bits
- b) Un procesador usa direcciones de memoria de 16 bits, páginas de 256 palabras y una TLB fully-associative de 8 entradas con política de reemplazo LRU. En un instante, la TLB de un proceso se encuentra en el estado indicado a continuación:

N° página	N° marco	Validez	LRU
0x01	0x11	1	0
0x00	0x00	0	7
0x10	0x13	1	1
0x20	0x12	1	5
0x00	0x00	0	6
0x11	0x14	1	4
0xAE	0x15	1	2
0xFF	0x16	1	3

Asumiendo que todas las páginas utilizadas actualmente por el proceso están en la TLB y que los marcos 0x17, 0x18, 0x19 se encuentran libres, actualice la tabla anterior después de ejecutar la siguiente secuencia de accesos a memoria **(4 ptos.)**:

- 0x11F0 (Lectura)
- 0x1301 (Escritura)
- 0x20AE (Escritura)
- 0x2332 (Escritura)
- 0x20FF (Lectura)
- 0x3415 (Escritura)

**Solución:** Para la solución sólo es necesario actualizar los valores de la tabla al final de todos los accesos, no después de cada uno.

- 0x11F0 (Lectura): Hit.
- 0x1301 (Escritura): Miss sin sustitución (se utiliza entrada no válida dada por LRU), se mapea página 0x13 a marco 0x17.
- 0x20AE (Escritura): Hit.

- 0x2332 (Escritura): Miss sin sustitución (se utiliza entrada no válida dada por LRU), se mapea página 0x23 a marco 0x18.
- 0x20FF (Lectura): Hit.
- 0x3415 (Escritura): Miss con sustitución dada por LRU, se mapea página 0x34 a marco 0x19.

Nº página	Nº marco	Validez	LRU
0x01	0x11	1	5
0x13	0x17	1	3
0x10	0x13	1	6
0x20	0x12	1	1
0x23	0x18	1	2
0x11	0x14	1	4
0xAE	0x15	1	7
0x34	0x19	1	0

### Pregunta 3

- a) Determine el número de ciclos que se demora el siguiente código, detallando en un diagrama los estados del pipeline por instrucción. El pipeline tiene forwarding entre todas sus etapas, el manejo de stalling es por software (instrucción NOP) y predicción de salto asumiendo que no ocurre. Indique en el diagrama cuando ocurre forwarding, stalling y flushing. **(4 ptos.)**

DATA			
var1		2	
var2		2	
res		0	
i		0	
CODE			
start:	MOV	A,(res)	
	MOV	B,(var2)	
	ADD	A,B	
	MOV	(res),A	
	MOV	A,(i)	
	ADD	A,1	
	MOV	(i),A	
	MOV	B,(var1)	
	JNE	start	
	MOV	A,(res)	
	ADD	A,3	
	MOV	(res),A	

**Solución:** Por cada instrucción ejecutada se indica la existencia de forwarding (recibido), flushing o stalling.

1. MOV A,(res):
2. MOV B,(var2):
3. NOP: Stalling para permitir forwarding entre instrucciones 2. y 4.
4. ADD A,B: Forward recibido desde registro MEM/WB de instrucción 2. a entrada etapa EX.
5. MOV (res),A: Forward recibido desde registro MEM/WB de instrucción 4. a entrada etapa MEM.
6. MOV A,(i):
7. NOP: Stalling para permitir forwarding entre instrucciones 6. y 8.
8. ADD A,1: Forward recibido desde registro MEM/WB de instrucción 6. a entrada etapa EX.
9. MOV (i),A: Forward recibido desde registro MEM/WB de instrucción 8. a entrada etapa MEM.
10. MOV B,(var1)
11. NOP: Stalling para permitir forwarding entre instrucciones 10. y 12.
12. JNE start: Forward recibido desde registro MEM/WB de instrucción 10. a entrada etapa EX.
13. FLUSH  $\Rightarrow$  ~~MOV A,(res):~~
14. FLUSH  $\Rightarrow$  ~~NOP: Stalling para permitir forwarding entre instrucciones 13. y 15.~~

15. FLUSH  $\Rightarrow$  ~~ADD A,3: Forward recibido desde registro MEM/WB de instrucción 10. a entrada etapa EX.~~
16. MOV A,(res):
17. MOV B,(var2):
18. NOP: Stalling para permitir forwarding entre instrucciones 2. y 4.
19. ADD A,B: Forward recibido desde registro MEM/WB de instrucción 2. a entrada etapa EX.
20. MOV (res),A: Forward recibido desde registro MEM/WB de instrucción 4. a entrada etapa MEM.
21. MOV A,(i):
22. NOP: Stalling para permitir forwarding entre instrucciones 6. y 8.
23. ADD A,1: Forward recibido desde registro MEM/WB de instrucción 6. a entrada etapa EX.
24. MOV (i),A: Forward recibido desde registro MEM/WB de instrucción 8. a entrada etapa MEM.
25. MOV B,(var1)
26. NOP: Stalling para permitir forwarding entre instrucciones 10. y 12.
27. JNE start: Forward recibido desde registro MEM/WB de instrucción 10. a entrada etapa EX.
28. MOV A,(res)
29. NOP: Stalling para permitir forwarding entre instrucciones 28. y 30.
30. ADD A,3: Forward recibido desde registro MEM/WB de instrucción 28. a entrada etapa EX.
31. MOV (res),A: Forward recibido desde registro MEM/WB de instrucción 30. a entrada etapa MEM.

También es valido entregar como respuesta para las instrucciones 13, 14 y 15 la siguiente secuencia MOV A,(res); ADD A,3; MOV (res),A sin indicar forwarding. Dado que la cantidad de instrucciones ejecutadas fue 31, la cantidad de ciclos que tomó el programa fue 35.

- b) Reordene el código anterior para disminuir el número de ciclos que demora, pero manteniendo el mismo resultado. Indique explícitamente cuanto es el ahorro. **(2 ptos.)**

**Solución:** Para disminuir la cantidad de ciclos utilizados, basta mover la instrucción MOV B,(var1) un línea hacia arriba, lo que produce que no se deba introducir un NOP para asegurar la consistencia. De esta manera, el programa puede ejecutarse en 33 ciclos.