Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

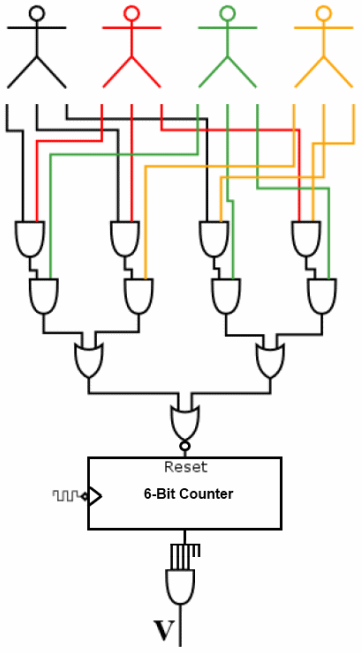
IIC2343 – Arquitectura de Computadores (II/2011)

Solución Examen

Pregunta 1 (I1)

Diseñe un sistema para la toma de decisiones de un directorio de una empresa constituido por 4 personas que cambian constantemente de opinión. Uno de los estatutos de la empresa dice que el directorio vota a favor una moción si al menos 3 de los miembros del directorio mantienen su voto a favor por más de 1 minuto seguido (esto no quiere decir necesariamente que los mismos tres miembros deben mantener su voto por más de un minuto). El circuito digital que diseñe debe estar constituido solamente por compuertas lógicas y elementos de almacenamiento (flip-flops, registros, contadores, etc.) y además dispone de una señal de clock de 1 Hz. Como variables de entrada considere $D1$, $D2$, $D3$, $D4$, que son los votos de cada miembro del directorio (1 voto a favor y 0 voto en contra), cada una de estas variables se mantiene en 1 sólo si el director correspondiente mantiene presionado un botón. La salida debe ser una variable binaria V que señale el voto final del directorio (1 a favor y 0 en contra). **(6 ptos.)**

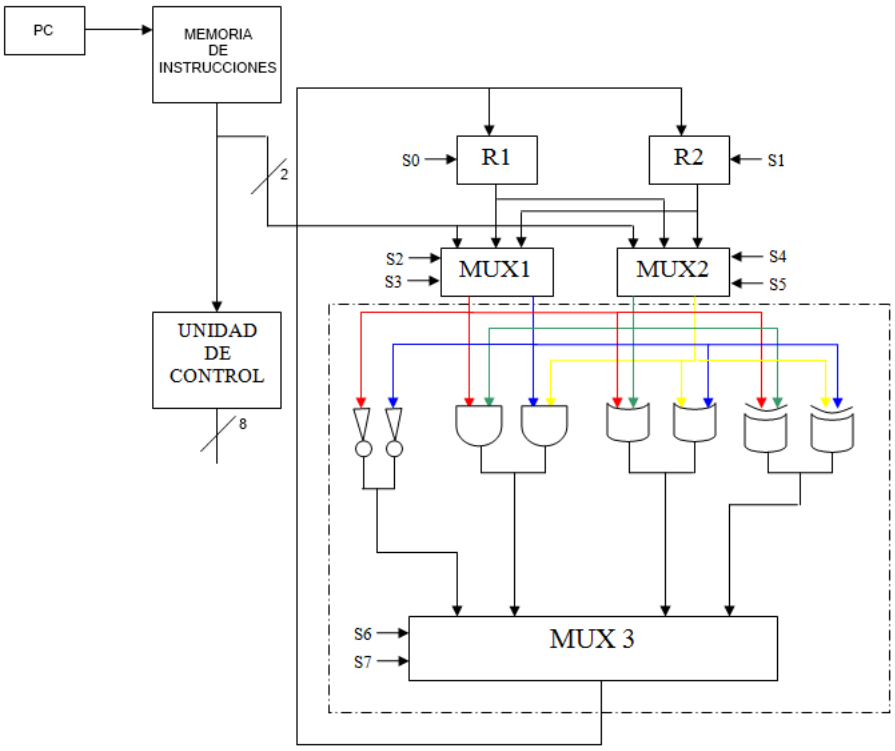
Solución:



Pregunta 2 (I2)

a) Diseñe un computador con arquitectura Harvard, de dos bits y sin memoria de datos, que realice las operaciones AND, OR, XOR y NOT. El computador debe tener dos registros acumuladores, R1 y R2, y permitir el uso de literales. Diseñe la ALU y muestre claramente las señales de control. **(3 ptos.)**

Solución:



b) Escriba un programa en assembly x86 que ordene de mayor a menor una lista de N bytes ($0 < N < 256$) enteros sin signo almacenados en la variable *arreglo*. Por problemas de capacidad de memoria, la lista ordenada debe quedar en el mismo lugar de la memoria y no se puede utilizar memoria RAM adicional para realizar el ordenamiento. El valor de N está almacenado en la variable N. **(3 ptos.)**

Solución:

```
org 100h

LEA BX, arreglo
MOV SI, [N]

outer_loop:
  DEC SI
  JZ end_outer
  MOV DI, [N]

inner_loop:
  DEC DI
  JZ outer_loop

compare:
  MOV AL, [BX + DI - 1]
  MOV DL, [BX + DI]
  CMP AL, DL
  JNL inner_loop

swap:
  MOV [BX + DI], AL
  MOV [BX + DI - 1], DL
  JMP inner_loop

end_outer:
```

Pregunta 3 (I3)

a) Describa el flujo completo de manejo de interrupciones en un computador x86. **(3 ptos.)**
Solución:

i. Dispositivo envía señal IRQ al controlador.
ii. PIC revisa su registro IMR, si la interrupción no está enmascarada la atiende, marcando un 1 en el bit correspondiente del Interrupt Request Register.
iii. PIC decide cual de las interrupciones actuales es prioritaria (menor IRQ) y marca un 1 en el bit correspondiente del In-Service Register.
iv. PIC envía interrupción (INT) a la CPU.
v. CPU termina de ejecutar la instrucción actual.
vi. CPU revisa si el flag de interrupciones está activo (IF = 1), en cuyo caso va a atender a la interrupción.
vii. CPU deshabilita la atención de más interrupciones (IF=0).
viii. CPU guarda en el stack los condition codes.
ix. CPU envía INTA para saber quien interrumpió.
x. PIC revisa In-Service Register para saber el id del IRQ que está siendo atendido, y envía mediante bus de datos.
xi. CPU usa el id para buscar dirección de ISR en el vector de interrupciones.
xii. CPU llama a la ISR asociada al dispositivo (CALL Mem[id]).
xiii. ISR respalda el estado actual de la CPU.
xiv. ISR ejecuta su código.
xv. ISR envía un comando EOI al PIC, con lo cual éste setea en 0 el In-Service Register, indicando que terminó la atención de la interrupción.
xvi. ISR devuelve el estado previo a la CPU.
xvii. ISR retorna.
xviii. CPU recupera condition codes desde el stack.
xix. CPU rehabilita la atención de interrupciones (IF=1).

b) La memoria principal de un computador tiene 16 bloques y la caché 4. La CPU ejecuta un programa que requiere los bloques 5, 7, 5, 4, 10, 5, 3, 12, 12, 15, 0, 5, 10, 5, 4 en el orden señalado. Suponiendo que inicialmente la memoria caché está vacía y que la función de correspondencia de la memoria caché es directa, indique cuál es la tasa de aciertos de la memoria caché. **(3 ptos.)**
Solución:

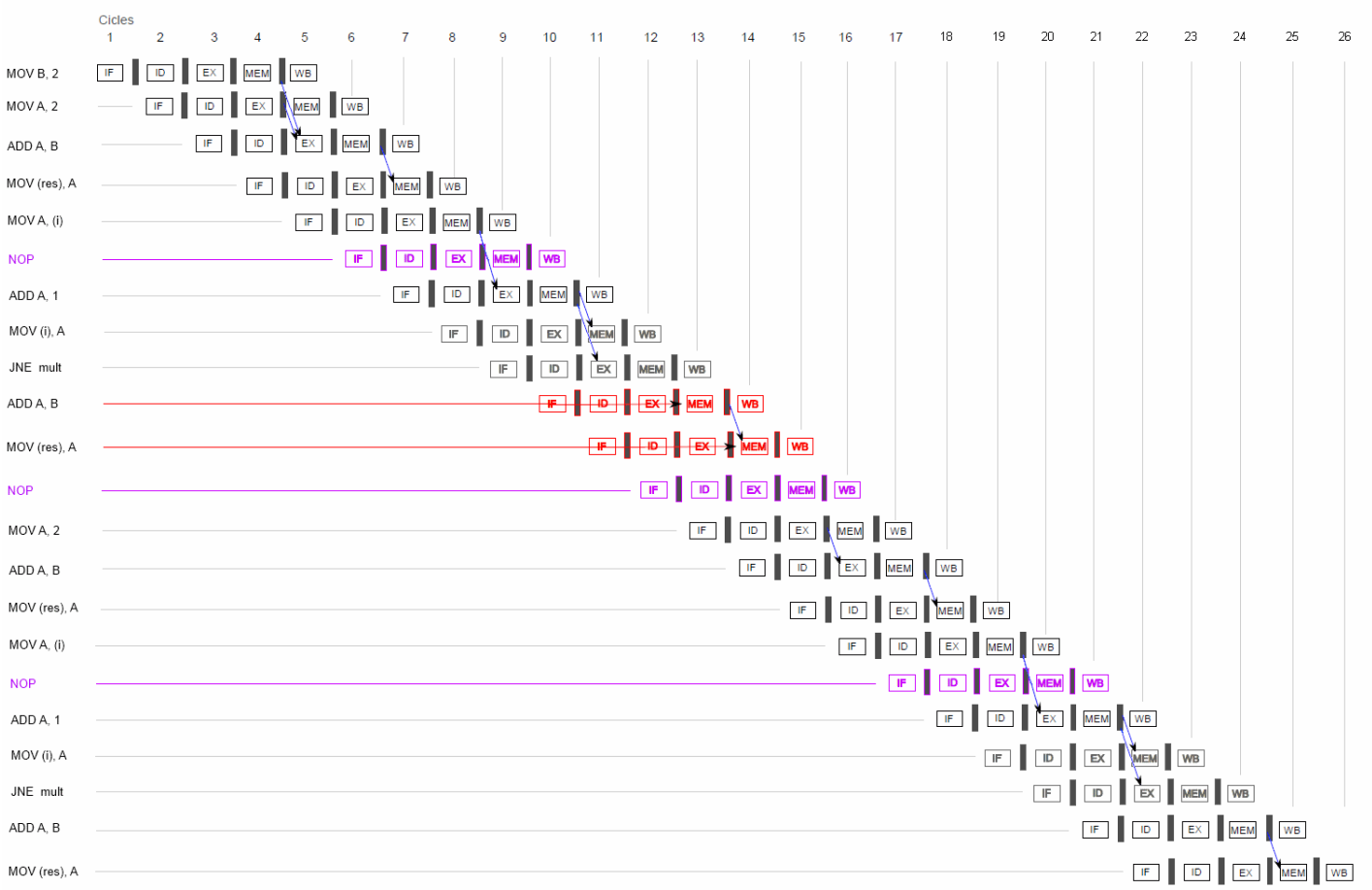
Bloque	5	7	5	4	10	5	3	12	12	15	0	5	10	5	4
0				4				12	12		0				4
1	5		5			5						5		5	
2					10								10		
3		7					3			15					
hit/miss	m	m	h	m	m	h	m	m	h	m	m	h	h	h	m
hit rate=	0.4														

Pregunta 4: ILP

a) Para el siguiente código, determine el número de ciclos que se demora, detallando los estados del pipeline para cada instrucción. El pipeline tiene forwarding implementado entre todas sus etapas, el manejo de stalling se realiza por software y se tiene predicción de salto asumiendo que no ocurre. Detalle en el diagrama cuando ocurre forwarding, stalling y flushing. **(5 ptos.)**

```
    DATA
            res     0
            i       0
    CODE
            MOV B,2
    mult:   MOV A,2
            ADD  A, B
            MOV (res), A
            MOV A, (i)
            ADD  A,1
            MOV (i), A
            JNE  mult
            ADD  A, B
            MOV (res), A
```

Solución:



b) Describa una modificación a los mecanismos de stalling y flushing usados anteriormente, para disminuir el número de ciclos que toma ejecutar el código. **(1 pto.)**
Solución: Existen múltiples alternativas para esta pregunta. Una posible respuesta es que la predicción de saltos alterne entre predecir salto la primera vez, la segunda predecir que no hay salto y así sucesivamente. Con esto se logra evitar el flushing, por lo que se ahorran 3 ciclos.

Pregunta 5: Resumen del Curso

Responda verdadero o falso y justifique. Cada correcta es $0,6$ ptos. y cada incorrecta $-0{,}3$ ptos.

a) El número decimal –45 en complemento a dos de 8 bits se representa como: 1101 0001
Solución: Falso, 45 en binario es 0010 1101, su inverso 11010010 y su complemento a 2 es 1101 0011.

b) Los bits del exponente en el estándar IEEE754 representan se interpretan como un número en complemento a 2.
Solución: Falso, el exponente se interpreta como un número binario sin signo.

c) Los condition codes permiten la existencia de salto condicionales en el computador básico.
Solución: Verdadero, la verificación de estos flags, obtenidos en base a comparaciones, permite decidir el salto.

d) OISC (One Instruction Set Computer) es un caso particular de la arquitectura Harvard.
Solución: Falso, es un caso particular de una ISA RISC.

e) Las instrucciones de la ISA x86 toman 2 ciclos en ejecutarse.
Solución: Falso, al ser una ISA CISC, las instrucciones son complejas por lo que pueden tomar más de 2 ciclos, como por ejemplo la instrucción MUL.

f) En un computador con acceso directo a memoria, la CPU le delega al DMA la transferencia de datos entre la memoria y un periférico.
Solución: Verdadero, el controlador de DMA se encarga de la transferencia y le informa cuando esta ha terminado.

g) Dos formas de establecer la correspondencia entre la caché y la memoria principal son: directa e indirecta.
Solución: Falso, las funciones de correspondencia son directa, fully associative y n-way associative.

h) La MMU es parte integral del manejo de caché en un computador.
Solución: Falso, la MMU es parte integral del manejo de la memoria virtual.

i) La función del ciclo fetch de una CPU es ejecutar una instrucción.
Solución: Falso, su función es obtener la instrucción desde la memoria.

j) La arquitectura Harvard permite evitar completamente los hazards estructurales.
Solución: Falso, la arquitectura Harvard evita el hazard estructural de los accesos simultáneos a memoria, pero no evita los que tienen relación con otras unidades funcionales.