

C1 (1,25 ptos)	C2 (1,25 ptos)	C3 (1,25 ptos)	C4 (1,25 ptos)	C5 (1,25 ptos)	P1 (2,5 ptos)	P2 (2,5 ptos)	TOTAL

## E.P.S.- UAM.- Arquitectura e Ingeniería de Computadores, Febrero 2004

Apellidos y Nombre:			
DNI :	Grupo 1 (mañana)	Grupo 2 (mañana)	Grupo 3 (tarde)

**ATENCION:** Sólo se debe contestar a 4 de las 5 cuestiones. En el caso de contestar a las 5 sólo se considerarán las 4 primeras.

**C1 (1,25 puntos).**- La siguiente descripción en VHDL modela un contador genérico:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity COUNTER is
    generic (L: INTEGER:=8);
    port (
        CLK: in STD_LOGIC;
        RESET: in STD_LOGIC;
        Q: out STD_LOGIC_VECTOR (L-1 downto 0)
    );
end COUNTER;

architecture PRACTICA of COUNTER is
begin

    -- Valor inicial del contador
    Q <= (others => '0');

    process(CLK, RESET)
        variable COUNT_INT: STD_LOGIC_VECTOR(L-1 downto 0);
    begin
        if reset='1' then
            COUNT_INT := (others => '0');
        elsif (CLK'event) and (CLK='1') then
            COUNT_INT := COUNT_INT + 1;
        end if;

        Q <= COUNT_INT;
    end process;

end PRACTICA;

```

Al ejecutar un *testbench* correcto sobre este diseño se comprueba que la salida es correcta cuando reset='1'. Posteriormente se pone reset='0'.

**a)** ¿Cuál es el valor de la salida después de 2 ciclos de reloj?

- a1)** "00000000"
- a2)** "00000001"
- a3)** "00000010"
- a4)** "000000X0"
- a5)** "UUUUUUUU"
- a6)** "XXXXXXXX"

**b)** En el caso que consideres que el funcionamiento del contador no es correcto ¿Dónde está el fallo?

**C2 (1,25 puntos).**- Sea un sistema donde los buses para direcciones y datos son ambos de 16 bits. El sistema dispone de una unidad cache asociativa de 2 vías con un total de 16 bloques y 8 palabras por bloque. Se sabe que en las estrategias de reemplazamiento la cache utiliza un algoritmo LRU. Se supone que la cache esta inicialmente vacía y que la primera vía se llena antes que la segunda. Se pide, utilizando la estructura de la figura:

- a) Rellenar la columna ETIQUETA en la vía que corresponda con su contenido en binario tras la ejecución de la siguiente secuencia de referencias a memoria: 35FE, FD34, FDD2, 43D8, 32FE, 4536, 87D2, 35F0, 5430, 4530, 9826, 94C2, 2368, FD34, AC74

Vía-1		Vía-2	
ETIQUETA	DATOS	ETIQUETA	DATOS

- b) Indicar la secuencia del fallos (F) y aciertos (A) provocada

	35FE	FD34	FDD2	43D8	32FE	4536	87D2	35F0	5430	4530	9826	94C2	2368	FD34	AC74
A/F	F	F													

**C3 (1,25 puntos).**- En la tabla adjunta se muestra la estadística de uso de un conjunto de instrucciones para un determinado procesador con estructura Von Neumann.

	Load/Store	Saltos Condicionales	Otras
PORCENTAJE	15%	25%	60%

El procesador está segmentado en seis etapas. **1)** Captura instrucción, **2)** Decodificación y detección de riesgos, **3)** Captura de operandos y cálculo de la dirección efectiva en saltos. **4)** Ejecución, cálculo de la dirección efectiva en accesos a memoria y en su caso cálculo de la condición. **5)** Acceso a memoria y **6)** Escritura de resultados en el banco de registros. Se supone que los únicos riesgos que reducen el rendimiento ideal del sistema (CPI = 1) son lo riesgos de control. Todas las instrucciones pasan por todos los segmentos. Se pide:

- a) Sabiendo que el 80% de los saltos son efectivos, indicar el CPI real del sistema sabiendo que al detectar un riesgo, el procesador se detiene hasta que se soluciona. Utilizar el diagrama adjunto.

a)	01	02	03	04	05	06	07	08	09	10	11	12	b)	01	02	03	04	05	06	07	08	09	10	11	12
S1	Bcc												S1	Bcc											
S2		Bcc											S2		Bcc										
S3			Bcc										S3			Bcc									
S4				Bcc									S4				Bcc								
S5					Bcc								S5					Bcc							
S6						Bcc							S6						Bcc						

- b) Utilizando **necesariamente** las variables definidas por la ley de Amdahl calcular la mejora que se produce si, se diseña un sistema de predicción estática en donde siempre predice efectivo.

La aceleración mejorada es  $F_m =$

La fracción mejorada es  $F_m =$

La aceleración global es  $AG =$

**C4 (1,25 puntos).**- Sea un sistema de memoria virtual paginado en tres niveles, con un tamaño de página de 4 kbytes y con las siguientes características, direccionamiento virtual hasta 64 Gbytes y memoria física o real de tan solo 32 Mbytes. El sistema dispone de una MMU en donde se encuentra un TLB completamente asociativo de 64 entradas y la tabla de páginas del primer nivel. El tamaño de cada tabla de los niveles 2 y 3 es de una página, y se encuentran repartidas entre memoria principal y secundaria.

El sistema consta de una unidad cache real asociativa de 2 vías con 8 bytes en cada bloque, y por el diseño del sistema se puede acceder al TLB y al mismo tiempo realizar la indexación para acceder a la cache.

Sabiendo que el descriptor de página de cualquier nivel ocupa 32 bits, se pide, justificando brevemente la respuesta:

**a)** La división en campos de la dirección virtual para el acceso al TLB y el tamaño y la estructura del TLB.

**b)** La división en campos de la dirección virtual para el acceso a las tablas de páginas y el tamaño de la tabla de páginas del primer nivel.

**c)** La división en campos de la dirección real para el acceso a la cache y el tamaño y la estructura de esta. No es necesario considerar bits de control alguno

**C5 (1,25 puntos).**– El mismo programa se ejecuta en dos procesadores distintos. El procesador A permite planificación dinámica utilizando el algoritmo del marcador y el B se basa en el algoritmo de Tomasulo con un solo bus de datos común (CDB). Ambos disponen de: 1 unidad funcional para acceso a memoria con una latencia en ejecución de 6 ciclos, 2 unidades para sumar/restar que tardan 2 ciclos y otras 2 unidades para multiplicar/dividir que emplean 10 ciclos. Notación de las instrucciones: Operación destino, fuente, fuente

**a)** Complete las tablas adjuntas indicando los ciclos de inicio-finalización de cada etapa, como se muestra para la instrucción I1

Instrucción	Procesador A: Marcador				Procesador B: Tomasulo		
	Emisión	Lectura Operandos	Ejecución	Escritura	Emisión	Ejecución	Escritura
I1: LD R2, 100(R1)	1	2	3-8	9			
I2: ADD R1, R1, 4	2						
I3: MUL R4,R2,R3							
I4: ADD R3,R5, R6							
I5: LD R6, 100(R4)							
I6: SUB R4, R2,R3							

**b)** Para cada uno de los procesadores responda en qué ciclo (o ciclos) se accede al banco de registros para escribir un valor en el registro R4.

**P1 (2,5 puntos).**- Un programa de un millón de instrucciones, que contiene un 10% de loads y un 5% de stores, se ejecuta en un procesador que funciona a 200 MHz (PowerPC 405). Si no hay detenciones en memoria, el código se ejecuta con un CPI de 1,1 ciclos/instrucción

Las características principales del sistema son: **1)** Arquitectura load/store, ancho de palabra para instrucciones y datos de 32 bits. Espacio de direccionamiento tanto real como virtual de 4 GB. Bus de datos de 64 bits con tiempo de ciclo de 10 ns. **2)** Caches separadas para instrucciones y datos, de 16 KB y asociativas de dos vías, con 32 bytes por bloque. La caché de datos se puede configurar como de escritura directa EDSAE (sin captura en fallo de escritura) o de post-escritura. **3)** Las caches son reales, y se conectan a la memoria principal a través de un bus de 64 bits. Cada bloque contiene tres bits adicionales: validez, LRU y modificación (este último sólo si es necesario). **4)** La tasa de aciertos de la caché de instrucciones es del 95%, y la de datos, el 92%. Si se emplea la alternativa de post-escritura, se estima que un 40% de los bloques están modificados. **5)** Una MMU basada en un TLB unificado completamente asociativo de 64 entradas. La tabla de páginas es mantenida por el sistema operativo, de tal manera que si se produce un fallo en el TLB se lanza una excepción. En el 95% de las ocasiones esa excepción necesita sólo de 50 instrucciones, porque la página a la que hace referencia el fallo ya se encuentra en la memoria principal. Pero en el 5% restante la página tiene que ser recuperada de la memoria secundaria, para lo que son necesarias 300 instrucciones, y además, la tarea queda suspendida durante 10 ms.

Se pide:

- a) Campos en los que se descompone la dirección en el acceso a la cache. Recursos necesarios para implementar las caches
- b) Suponiendo 100% de aciertos en el TLB, tiempo necesario para ejecutar el programa con la configuración de escritura directa en la cache de datos
- c) Bajo las condiciones anteriores, indicar la aceleración (mejora) al pasar a una configuración post-escritura
- d) Incremento en el tiempo de ejecución para el caso de post-escritura suponiendo una tasa de aciertos en el TLB de 99,5%

La unidad de control predice estáticamente todo los saltos que decrementan el PC como efectivos y el resto como no efectivos. Todas las instrucciones pasan por todas las etapas.

Inicialmente el contador de programa contiene la dirección de la instrucción I1. Suponga que inicialmente el valor de F2 es la constante a. El registro R1 = 0 y el registro R2= 8\*i (con i=4) y el registro R0 esta definido con valor 0.

b) Utilizando el cronograma adjunto muestre la evolución temporal de la primera iteración de la secuencia de instrucciones. En este apartado no considere ningún tipo de adelantamiento salvo que las etapas **ID** y **W** pueden acceder en el mismo ciclo de reloj al banco de registros (la etapa **W** accede en la primera mitad y la etapa **ID** en la segunda mitad). ¿Cuántos ciclos tarda la primera iteración? ¿En qué ciclo comienza la segunda iteración?

c) Repita el apartado (b) pero considerando el adelantamiento de datos entre etapas mas eficiente posible. Marque con una flecha en el cronograma entre que etapas se produce adelantamiento de datos. ¿Cuántos ciclos tarda la primera iteración? ¿En qué ciclo comienza la segunda iteración?

**d)** Se mejora el sistema del apartado anterior, añadiendo en la etapa de ejecución otra unidad funcional (UF) para la ejecución de las operaciones de CF, independiente de la UF de enteros. Pese a esta mejora se mantiene un cauce escalar simple en donde la emisión y finalización de las instrucciones es en orden. Indique en el cronograma adjunto la ejecución de las instrucciones. ¿Cuántos ciclos tarda la primera iteración? ¿En qué ciclo comienza la segunda iteración?

[illegible]

**c)**

[illegible]

**d)**

[illegible][illegible]