Apellidos _____ Nombre _____

Arquitectura de Computadores. Examen final 12/02/2015. Problemas.

1. Supongamos un procesador segmentado con planificación dinámica mediante el algoritmo de Tomasulo que dispone de una unidad de multiplicación/división en PF (2 estaciones de reserva) y 3 unidades de suma/resta en PF (una estación de reserva cada una). Las unidades funcionales de PF no están segmentadas y emplean 4 ciclos para la suma y la resta, 6 para la multiplicación y 9 para la división. Los buffer de Load y el de Store tienen 4 posiciones cada uno. Existe una unidad de Load y otra de Store, ambas no segmentadas, cuya latencia es 2 ciclos de reloj.

Además tiene las siguientes características.

- Los datos que se escriben en la etapa de escritura NO se pueden usar para la ejecución de una instrucción hasta el ciclo siguiente
- Hay un solo bus de datos común (CDB)
- Las estaciones de reserva se liberan al final de la fase de escritura.

a) Indica en qué ciclo de reloj se realiza cada una de las etapas de ejecución para las instrucciones siguientes. Utiliza llamadas en la tabla para explicar separadamente las dependencias que motivan cualquier retraso que se produzca en la temporización. **[2 puntos]**

	issue	exec	write	
LD F2, 0(R1)	1	2-3	4	
LD F4, 0(R2)	2	4[1]-5	6	[1] UF no segmentada
LD F6, 0(R3)	3	6[1]-7	8	[1] UF no segmentada
MUL F8,F2,F4	4	7[3]-12	13	[3]Dependencia LDE
SUB F12,F8,F6	5	14[4]-17	18	[4]Dependencia LDE
ADD F10,F12,F4	6	19[5]-22	23	[5]Dependencia LDE
DIV F8,F8,F10	7	24[6]-32	33	[6]Dependencia LDE
SUB F12,F2,F6	8	9-12	14[2]	[2] CDB ocupado
ADD F12,F12,F8	15[7]	34[8]-37	38	[7] ER ocupada; [8]Dependencia LDE
SD 0(R1), F12	16	39[9]-40	----	[9]Dependencia LDE
SD 0(R2), F10	17	24[10]-25	----	[10]Dependencia LDE

b) Indica cual es el contenido de las estaciones de reserva y registros al final del ciclo 20. **[1 punto]**

En el ciclo 20 las operaciones aritmético lógicas que están activas son:

	issue	exec	write	
LD F2, 0(R1)	1	2-3	4	
LD F4, 0(R2)	2	4[1]-5	6	[1] UF no segmentada
LD F6, 0(R3)	3	6[1]-7	8	[1] UF no segmentada
MUL F8,F2,F4	4	7[3]-12	13	[3]Dependencia LDE
SUB F12,F8,F6	5	14[4]-17	18	[4]Dependencia LDE
ADD F10,F12,F4	**6**	**19[5]-22**	**23**	**[5]Dependencia LDE**
DIV F8,F8,F10	**7**	**24[6]-32**	**33**	**[6]Dependencia LDE**
SUB F12,F2,F6	8	9-12	14[2]	[2] CDB ocupado
ADD F12,F12,F8	**15[7]**	**34[8]-37**	**38**	**[7] ER ocupada; [8]Dependencia LDE**
SD 0(R1), F12	**16**	**39[9]-40**	**----**	**[9]Dependencia LDE**
SD 0(R2), F10	**17**	**24[10]-25**	**----**	**[10]Dependencia LDE**

ER	Oper	Qi	Vi	Qj	Vj	Busy
Add1	+	0	[Valor F12]	0	[Valor F4]	yes
Add2	+	0	[Valor F12]	Mul1	######	yes
Add3	######	######	######	######	######	No
Mul1	/	0	[Valor F8]	**Add1**	######	yes
Mul2	######	######	######	######	######	No

→ Contenido no válido

Reg	F0	F2	F4	F6	F8	F10	F12	F14	F16
VALOR	[Valor]	[Valor]	[Valor]	[Valor]				[Valor]	[Valor]
Qi					Mul1	**Add1**	Add2		

2. Sea el bucle

```
for (i=0;i<1000;i++) {
            b[i,5] = (a[i]+c[i]+2)*3;
            d[i] = a[i]*b[i,5];
}
```

que se desea ejecutar en una computador similar al VMIPS, pero con las siguientes diferencias:

- tiene dos pipes de carga/almacenamiento
- tiene 2 pipes de suma
- el tamaño de los registros vectoriales es de 128 componentes

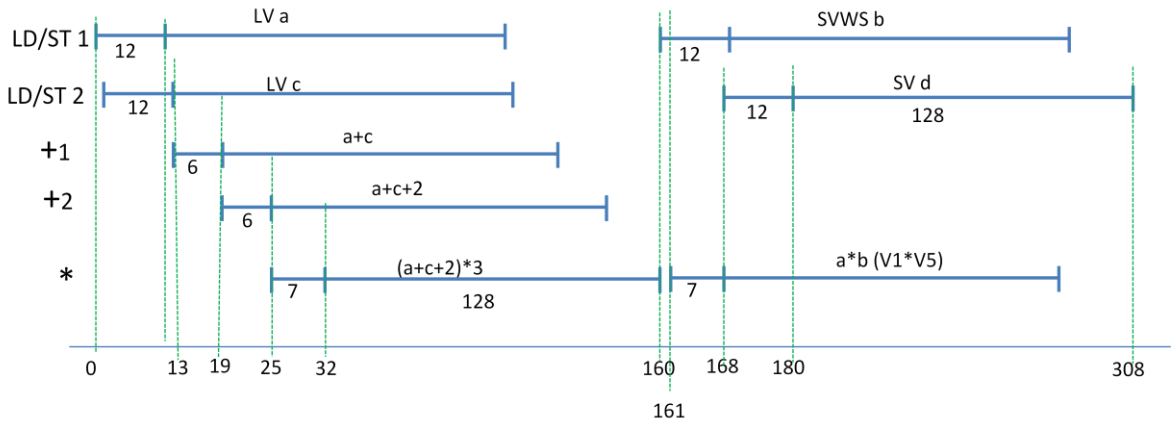
Se supone que la matriz B tiene 1000 filas por 17 columnas y está almacenada en memoria por filas. La memoria del VMIPS está entrelazada en 16 bancos. El tiempo de ciclo es 1,25 ns.

Se pide:

a) Tradúcelo el programa al correspondiente bloque de código vectorial del VMIPS, usando los registros vectoriales para tratar de evitar accesos a memoria innecesarios. Nota: basta con escribir la parte vectorial del código, indicando el contenido que deben tener los registros escalares que se usen. **[1 punto]**

```
LV V1,Ra                ; Ra: Inicialmente, dirección del a[0]
LV V2,Rc                ; Rc: Inicialmente, dirección del c[0]
ADDVV V3,V1,V2          ; a+c
ADDVS V4,V3,F_dos       ; a+c+2.  F_dos: contiene la cte 2
MULVS V5,V4,F_tres      ; (a+c+2)*3.  F_tres: contiene la cte 3
SVWS  (R_b5, R_17), V5  ; R_b5: Inicialmente, dirección de b[0,5]
                        ; R_17: contiene la cte 17 que es el stride
MULVV V6,V1,V5          ; a*b[-,5]
SV    Rd,V6             ; Rd: Inicialmente, dirección de d[0]
Actualizar punteros
Iterar el bucle si no se ha finalizado el cálculo
```

b) Dibuja el diagrama de tiempo y determina el tiempo de ejecución, teniendo en cuenta la penalización de la ejecución por bloques. **[1.5 puntos]**



$T_{chime} = 2$; $T_{start} = 308 - (128*2) = 52$ (Nota.- Tb es igual a 13+6+6+7+1+7+12); $T_{loop} = 15$

$T_n = \lceil n/MVL \rceil x (T_{loop} + T_{start}) + n x T_{chime} = \lceil 1000/128 \rceil x (15 + 52) + 1000 x 2 = 8*67 + 2000 = 2536$ ciclos

c) Calcula el rendimiento asintótico del programa en MFLOPS. **[0.5 puntos]**

Cuando n → ∞: Tn = (n/128) (15+52) + 2n = 2,52 n ciclos
R∞ = (Nº op PF) / (2,52 n) FLOP/ciclo = 4n / 2,52n = 1,59 FLOP/ciclo
Pasamos a MFLOPS: R∞ = (1,59 FLOP/ciclo) / ($1,25*10^{-9}$ s/ciclo) = 1272 MFLOPS

3. Una red omega conecta 256 procesadores con 256 módulos de memoria usando conmutadores de grado 16. Si suponemos que cada elemento de conmutación tiene un retardo de 4ns, que cada referencia a memoria implica acceder a un módulo de memoria cuyo tiempo de acceso es 5ns, y que la red está libre de problemas de contención:

a) Determinar el número de ciclos que se requieren para satisfacer una solicitud de lectura de un procesador si la frecuencia del procesador es de 2 GHz. **[1 pto]**

$N=K^n$ → $256=16^n$, de donde n=nº etapas=2

El tiempo implicado en satisfacer la solicitud de lectura comprende el tiempo necesario para atravesar la red desde el procesador hasta el módulo correspondiente, acceder al dato y luego volver a atravesar la red hasta el procesador solicitante ($T= 2xT_{red}+T_{mem}$)

Para atravesar la red debemos considerar el número de etapas y el retardo asociado a cada una de ellas, es decir, el retardo del conmutador: T_{red}=4ns x 2etapas = 8ns

Por tanto T = 2x8ns + 5ns = 21 ns. Como f= 2GHz → t_{ciclo} = 0.5 ns, por lo que T = 21/0.5 = **42 ciclos**

b) Repetir el apartado a) para el caso en que los conmutadores son de grado 2 y tienen un retardo de 2ns. **[0.25 ptos]**

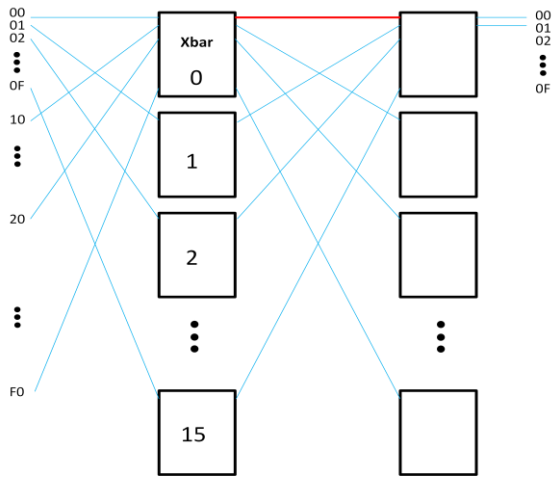
Conmutadores grado 2; $256=2^n$, de donde n=nº etapas=8 T_{red}=2ns x 8etapas = 16ns

T = 2x16ns + 5ns = 37 ns. Como f= 2GHz → t_{ciclo} = 0.5 ns, por lo que T = 37/0.5 = **74 ciclos**

c) Suponiendo que los conmutadores de grado 16 han sido implementados por medio de una red tipo crossbar, al realizar la comunicación entre el procesador 0 y el módulo de memoria 0 ¿qué comunicaciones se bloquean debido a la contención en los canales de comunicación entre los conmutadores? **[0.75 ptos]**

La solución es muy similar a la del ejercicio 14 de la hoja 4. Como tenemos 16 conmutadores de grado 16 en cada etapa, representamos por comodidad cada procesador y módulo de memoria con notación hexadecimal, de modo que la permutación $\sigma(a,b) = (b,a)$ siendo $a,b \in \{0..F\}$

Al conectar el procesador 0 con el módulo 0 queda bloqueada cualquier comunicación que atraviese el enlace rojo. Es decir, cualquier comunicación que conecte los procesadores 16, 32, 48,.....240 con los módulos de memoria 0, 1,2 ,.....15



4. Para la cadena de referencias a memoria que se indica en la primera fila de la Tabla, calcula el coste de ejecutarla en un multiprocesador (3 procesadores) de bus compartido que soporta el protocolo MSI y usa *BusUpgrd.* Indica el resultado en forma de una tabla que muestre también la evolución en el tiempo del contenido de los marcos de bloque de cache y su estado; donde cada columna es un acceso y cada fila un marco de bloque.

La memoria principal contiene 16 bloques (bloque 0, 1,2, …15) y cada una de las caches sólo dos marcos de bloque. Las referencias a memoria indican el bloque de memoria principal al que acceden; por ejemplo P1 R 0 quiere decir que el procesador 1 lee el bloque de memoria 0 (análogamente se usa W para indicar una escritura).

Asumir que todas las caches son directas, están inicialmente vacías, y que se usa el siguiente modelo de coste: acierto de lectura/escritura: 1 ciclo, fallos que requieren una transacción del bus simple: 60 ciclos, fallos que requieren una transferencia de un bloque de cache: 90 ciclos.

Suponer que se utiliza en todos los casos caches con asignación en escritura (write allocate) y post escritura (write-back). A modo de ejemplo se ha completado la primera columna. **[2 ptos]**

CADENA	P1 R 0	P2 W 1	P3 W 0	P1 R 1	P3 R 0	P2 R 1	P3 W 0	P1 R 0	P1 W 0
Bloque 0 P1	0 S	0 S	I	I	I	I	I	0S	0M
Bloque 1 P1	I	I	I	1 S	1 S	1S	1S	1S	1S
Bloque 0 P2	I	I	I	I	I	I	I	I	I
Bloque 1 P2	I	1 M	1M	1S	1S	1S	1S	1S	1S
Bloque 0 P3	I	I	0 M	0M	0M	0M	0M	0S	I
Bloque 1 P3	I	I	I	I	I	I	I	I	I
Ciclos	90	90	90	90	1	1	1	90	60

CADENA	P3 R 0	P1 W 2	P2 R 2	P1 W 1
Bloque 0 P1	0S	2M	2S	2S
Bloque 1 P1	1S	1S	1S	1M
Bloque 0 P2	I	I	2S	2S
Bloque 1 P2	1S	1S	1S	I
Bloque 0 P3	0S	0S	0S	0S
Bloque 1 P3	I	I	I	I
Ciclos	90	90	90	60