Examen Primer Parcial **ESTRUCTURA DE COMPUTADORES** **Enero 2021**

Apellidos y Nombre	DNI	Grupo

NOTA IMPORTANTE: puesto que el uso de la calculadora está permitido durante este examen, todos los cálculos deberán llegar hasta su valor final y deberán incluir la unidades en que se expresan

1 **(1.75 puntos)** La ruta de los datos monociclo del procesador MIPS R2000, que se muestra en la Figura 1, tiene respecto a la que se ha estudiado en clase un elemento funcional nuevo, llamado Mult-Div, que le permite ejecutar las instrucciones de multiplicación y división de la arquitectura. Si la línea Op_Multdiv vale 0, multiplica y si es 1 divide. Los registros HI y LO, ambos de 32 bits, permiten almacenar el resultado.

mult Rs,Rt # HI LO = Rs * Rt (HI almacena la parte alta y LO la parte baja del resultado)

COP	RS	RT	RD	desp5	Función
000000	XXXXX	XXXXX	00000	00000	011000

div Rs,Rt # LO = Rs / Rt (cociente división entera) y HI = Rs % Rt (resto división entera)

COP	RS	RT	RD	desp5	Función
000000	XXXXX	XXXXX	00000	00000	011010

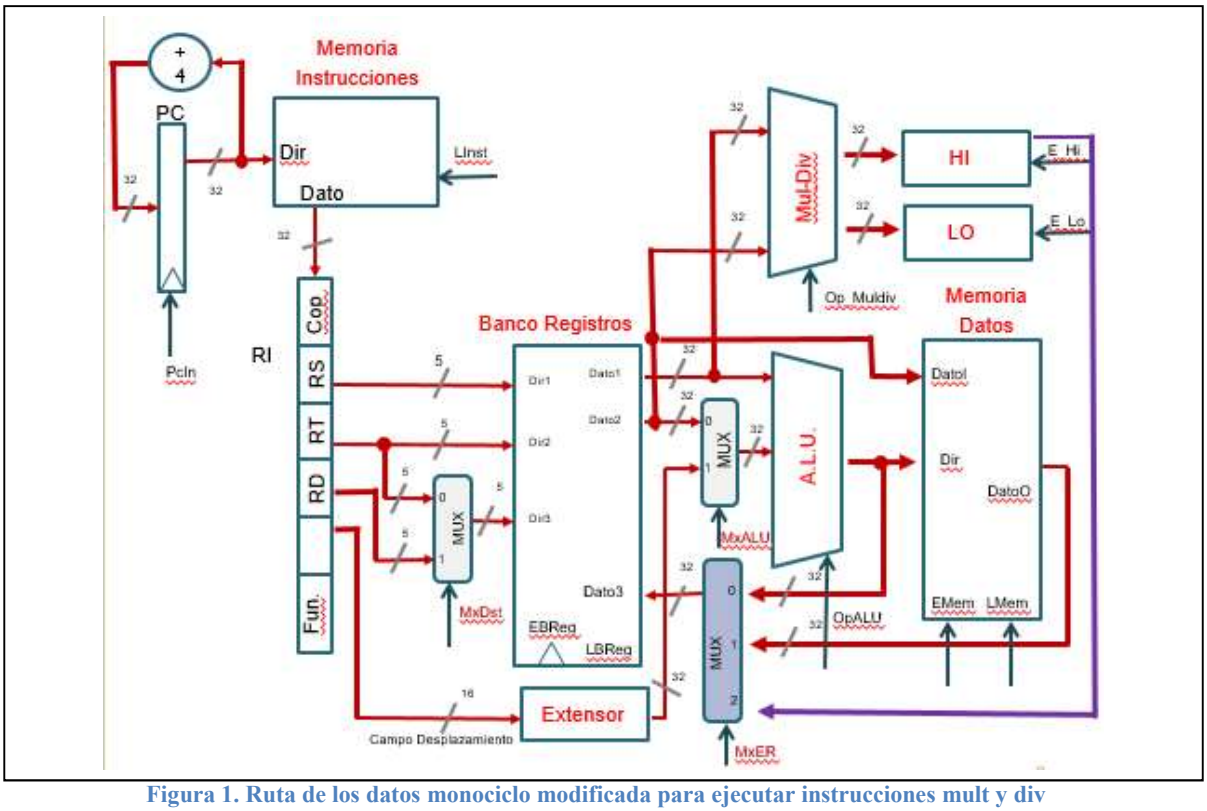


Figura 1. Ruta de los datos monociclo modificada para ejecutar instrucciones mult y div

a) ***(1 punto)*** Complete la tabla correspondiente a las señales de control para ejecutar sobre la misma las instrucciones que se detallan. Si alguna de ellas no puede ejecutarse en esta ruta escriba: "No se puede".

Instrucción	Form	EBreg	MxER	MxDst	OpALU	MxALU	Emem	Lmem	Op_Multdiv	E_Hi	E_Lo
add rd,rs,rt	R	1	0	1	010	0	0	0	X	0	0
mult rs, rt	R	0	X	x	xxx	x	0	0	0	1	1
J destino	J	No	Se	puede							
lw rt,desp(rs)	I	1	1	0	010	1	0	1	x	0	0

Tabla 1 Tabla de verdad apartado 1.a

b) ***(0,75 puntos)*** Se desea incluir la instrucción *mfhi rd*, con formato R, como una instrucción más del juego de instrucciones básico del MIPS visto en clase. La instrucción lleva el contenido del registro HI al registro especificado del banco de registros. Se codificaría como se muestra en la figura 2:

COP	RS	RT	RD	desp5	Función
000000	00000	00000	00010	00000	010000

Figura 2 Codificación *mfhi $2*, de tipo R

- **(0,5 puntos)** Indique sobre la ruta de datos de la Figura 1 las modificaciones necesarias para llevar a cabo la instrucción mfhi descrita. Cable desde HI al Mux MxER, entrada 2. Modificado en figura 1.

- **(0,25 puntos)** Rellene la tabla con las señales de control necesarias. En binario.

Instrucción	Form	EBreg	MxER	MxDst	OpALU	MxALU	Emem	Lmem	Op_Multdiv	E_Hi	E_Lo
mfhi rd	R	1	10	1	xxx	xxx	0	0	x	0	0

Tabla 2 Tabla de verdad apartado 1.b

2 **(0,75 puntos)** Considérese la ruta de datos segmentada de cinco etapas (LI, DI, EX, M, ER) estudiada en clase y supóngase que las etapas tienen los siguientes retardos: 60 ns las memorias, 55 ns el banco de registros para lectura y escritura, y 49 ns la unidad aritmético-lógica. Los registros de segmentación tienen un retardo de 5 ns. Resto de unidades funcionales tienen un tiempo despreciable.

a) ***(0,25 puntos)*** ¿Cuál es tiempo de ciclo de esta ruta de datos segmentada? Justifique su respuesta

$T_S = \max\{60,55,49\} + 5 = 65$ ns

b) ***(0,25 puntos)*** ¿Cuál es la aceleración máxima conseguida respecto a la versión no segmentada? Justifique su respuesta

$T_{NS} = 60 + 55 + 49 + 60 + 55 = 279$ ns

$S = \frac{T_{NS}}{T_S} = \frac{279}{65} = 4.3$

c) ***(0,25 puntos)*** Suponiendo que se rediseña como un procesador superescalar de grado 4 (4 vías), cuyos cauces segmentados son similares al especificado en el enunciado ¿qué productividad máxima se podría alcanzar?

$\chi_S = \frac{1000}{65}$MIPS = 15.38 MIPS por tanto el procesador superescalar será de $\chi_S = \frac{4000}{65}$MIPS = 61.5 MIPS

3 **(1.5 puntos)** En el procesador segmentado en cinco etapas del ejercicio anterior se va a ejecutar el siguiente fragmento de código en ensamblador del MIPS R2000. Asuma que los conflictos por dependencias de datos y control se solucionan mediante la inserción de ciclos de parada y una latencia de salto 2.

```
(1)  Dim:     lui $t3,0x2000
(2)           ori $t3,$t3,0x1000
(3)           ori $t4,$0,100
(4)  for1:    beq $t4,$0,fin
(5)           lw $t5,0($t3)
(6)           sll $t5,$t5,2
(7)           sw $t5,0x1000($t3)
(8)           addi $t4,$t4,-1
(9)           addi $t3,$t3,4
(10)          j for1
(11)  fin:  jr $ra
```

	Registro	instrucción en que se escribe	instrucción en que se lee
Riesgo	$t3	(1)	(2)
Riesgo	$t4	(3)	(4)
Riesgo	$t5	(5)	(6)
Riesgo	$t5	(6)	(7)
Riesgo			

Tabla 3. Riesgos de datos

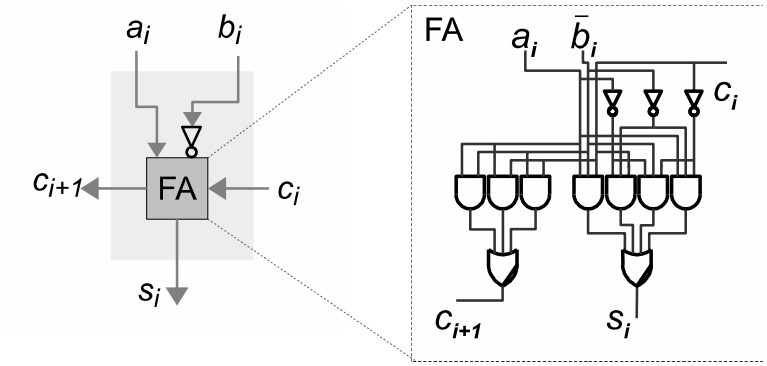
a) **(0.4 puntos)** Indique los riesgos por dependencias de datos que existe utilizando la Tabla 3 (el número de riesgos no tiene por qué ser igual al número de filas)

b) **(0.8 puntos)** Indique para dicho código (justifique las respuestas)

Número de Instrucciones ejecutadas (I)	$3 + (7 * 100) + 1_{beq} + 1_{jr} = 705$ instrucciones
Número de ciclos de parada (P)	$2_{ori} + 2_{beq} + (2_{beq} * 101$ veces$) + (2_{sll} + 2_{sw} + 2_j)*100$ veces $+ 2_{jr} = 4 + 202 + 600 + 2 = 808$ paradas
Número de ciclos totales de ejecución (T)	$I + P + 4 = 705 + 808 + 4 = 1517$ ciclos
CPI	$1 + P / I = 1 + 808/705 = 2.15$

c) **(0.3 puntos)** Rellene el diagrama de ejecución solo para las tres instrucciones que se indican, asumiendo que es la primera vez que se ejecutan:

4 (1.2 puntos) Considere el circuito de la figura 3, que combina un sumador FA y una puerta NOT. Vea también los retardos de los componentes.



Puerta	Retardo
NOT	0.5 ns
AND	1 ns
OR	1.5 ns

Figura 3. Combinación de un FA y una puerta NOT

a) (0.4 puntos) Si las entradas a_i, b_i y c_i llegan simultáneamente al circuito de la figura 3, ¿cuáles son los retardos de las salidas c_{i+1} y s_i?

Salida c_{i+1}: $t_{NOT} + t_{AND} + t_{OR} = 3$ ns

Salida s_i: $2\times t_{NOT} + t_{AND} + t_{OR} = 3.5$ ns

b) (0.4 puntos) Si las entradas a_i y b_i llegan en tiempo 0 y la entrada c_i a los 3 ns ¿en qué momento son válidas las salidas c_{i+1} y s_i?

Salida c_{i+1}: $3\text{ns} + t_{AND} + t_{OR} = 5.5$ ns

Salida s_i: $3\text{ ns} + t_{NOT} + t_{AND} + t_{OR} = 6$ ns

Con ese circuito como bloque básico, se puede construir el operador de la figura 4 que hace la resta $A–B$ de dos enteros de 32 bits.

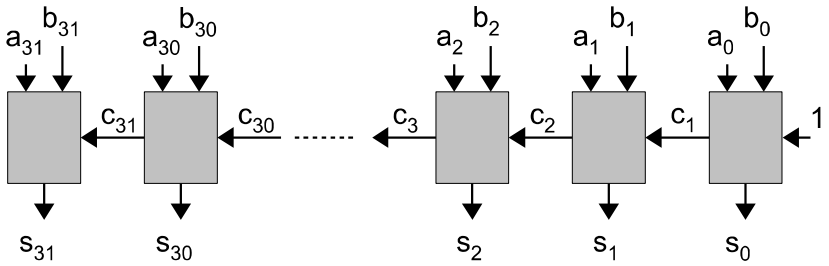


Figura 4: operador de resta

c) (0.2 puntos) ¿Cuál será el retardo del bit s_{31}? Indique cómo lo ha obtenido. Por ejemplo: "$15 \times t_{AND} + 4 \times (t_{NOT} + t_{OR}) = 77$ ns"

Retardo s_{31}: $t_{NOT} + 31 \times (t_{AND} + t_{OR}) + (t_{NOT} + t_{AND} + t_{OR}) = 0.5 + 77.5 + 3 = 81$ ns

d) (0.2 puntos) ¿Cuál será la productividad máxima en MOPS del operador?

Productividad = 1 operación / 81 ns ≈ 12,3 MOPS

5 (1 punto) Para el diseño de un operador secuencial de multiplicación para números con signo de 32 bits, dispone de los siguientes componentes:

- Un sumador/restador capaz de calcular HI±M con retardo de 1 ns
- Un sumador/restador capaz de calcular HI±M o HI±2·M con retardo total de 1.2 ns
- Registros de desplazamiento de un bit y de dos bits con carga paralela. La carga paralela tarda 0.1 ns y el desplazamiento (de uno o dos bits) otros 0.1 ns.

Con estas componentes ha de diseñar dos operadores secuenciales: uno que aplique el algoritmo de Booth simple y el otro que aplique recodificación por parejas. En ambos casos, el autómata de control requiere 0.2 ns por ciclo para controlar el número de iteración, evaluar los bits del multiplicador y generar las señales de control adecuadas. Además, necesita un ciclo de reloj para iniciar una operación.

Complete la siguiente tabla comparativa de ambos diseños.

	Booth simple	Booth con recodificación por parejas
Número de ciclos	32+1 = 33	16 + 1 = 17
Periodo de reloj (ns)	1.4	1.6
Frecuencia de reloj (MHz)	714	625
Tiempo total de operación (ns)	46.2	27.2
Productividad (MOPS)	21.6	36.8

6 (0.8 puntos) La siguiente función *float facts(int $a0)* codificada en ensamblador del MIPS R2000 calcula el factorial de un número entero (en *$a0*) y devuelve el resultado (en *$f0*) en formato de coma flotante de simple precisión:

```
# float facts(int $a0)
facts:
                li $t0,1
                ble $a0,$t0,facts_01
                mtc1 $a0,$f0
                cvt.s.w $f0,$f0
facts_loop:
                addiu $a0,$a0,-1
                mtc1 $a0,$f2
                cvt.s.w $f2,$f2
                mul.s $f0,$f0,$f2
                bgt $a0,$t0,facts_loop
                jr $ra
facts 01: # caso especial. 0! = 1! = 1
                li $t0,1
                mtc1 $t0,$f0
                cvt.s.w $f0,$f0
                jr $ra
```

a) (0.4) ¿Cuántas operaciones en coma flotante (suma, resta, multiplicación, división o conversión de tipo) requiere el cálculo de *facts(22)*?

Si $n>1$, la función *facts* calcula el factorial de n mediante $n-1$ iteraciones del bucle `facts_loop` . En total hace n conversiones de tipo y $n-1$ multiplicaciones en coma flotante. Para n=22, tenemos 43 operaciones.

b) (0.4 puntos) Calcule en MFLOPS la productividad en coma flotante del código anterior cuando calcula *facts(22)* en 1 µs de tiempo de ejecución.

La productividad será P = 43 operaciones /10^–6 segundos = 43 MFLOPS

7 (2.4 puntos) Un chip de memoria SDRAM DDR3 tiene una capacidad de 1 GB y un ancho de palabra de 16 bits.

a) (1.2 puntos) Complete la siguiente tabla sabiendo que el chip está organizado en ocho bancos de 32 K (32768) filas cada uno. Utilice, donde sea apropiado, los prefijos habituales, como por ejemplo "18 K palabras" o "2 MB"

Número de palabras que contiene el chip	2^30 bytes / 2 bytes/palabra = 2^29 = 512 M palabras
Número de bits de la dirección para seleccionar banco	3 bits
Capacidad en bytes de cada banco	1 GB de capacidad / 8 bancos = 128 MB /banco
Capacidad en bytes de cada fila de un banco	128 MB/banco / 32 K filas = 4KB
Número de bits de la dirección que seleccionen una fila	15 bits
Número de palabras que contiene una fila	4 KB/fila / 2B/palabra = 2K palabras

b) (0.6 puntos) Conectado a un bus que funciona a 800 MHz, con temporización CL = tRCD = tRP = 10 ciclos, calcule

Duración en ns de un ciclo de reloj	1 / 800·10^6 ciclos/segundo = 1.25 ns/ciclo
Latencia de CAS resultante en ns	10 ciclos × 1.25 ns/ciclo = 12.5 ns
Ancho de banda del chip en MBps	800·10^6 ciclos/segundo × 2 transferencias/ciclo × 2 bytes / transferencia = 3200 MBps

c) (0.6 puntos) ¿Cuántos chips como este serán necesarios para organizar un módulo DIMM con palabras de 64 bits en una única fila? Conectado al bus de 800 MHz con temporización CL = tRCD = tRP = 10 ciclos, ¿cuáles serán la capacidad y el ancho de banda del módulo resultante?

Numero de chips	Harán falta 64/16 = 4 chips. Por lo tanto, la capacidad y el ancho de banda se multiplicarán por 4
Capacidad total del módulo en GB	4 GB
Ancho de banda total del módulo en MBps	12800 MBps

8 (0.6 puntos) Un computador basado en el MIPS R2000 tiene conectados un módulo *M* de 256 MB de memoria a partir de la dirección 0x20000000.

a) (0.3 puntos) ¿Cuál será el rango de direcciones contenidas dentro del módulo *M*? Conteste en hexadecimal

0x20000000-0x2FFFFFFF

b) (0.3 puntos) ¿Cuál será la función de selección activa por nivel bajo apropiada para el módulo *M*?

Sel = $A_{31}+A_{30}+A_{29}*+A_{28}$