FUNDAMENTOS DE MICROPROCESADORES
PROBLEMAS DE LA UNIDAD 4.- EL PROCESADOR II: LA RUTA DE DATOS y SU CONTROL

4.1. En la arquitectura de MIPS de ciclo único estudiada en clase, hay identificados varios elementos de hardware. Dadas las instrucciones:

1. `add $s1, $s2, $s2` y **2.** `lw $t1, Offset($t2)`

Se quiere saber para cada una:

a. ¿Cuáles son los valores de las señales de control generadas?
b. ¿Qué recursos o elementos hardware (excluyendo los multiplexores y el registro PC) hacen algo útil para esta instrucción?

4.2. Las distintas entidades digitales tienen latencias distintas (latencia: tiempo necesario para hacer su trabajo). La latencia mínima de una instrucción está determinada por la latencia de los bloques a lo largo de su camino crítico (el de mayor latencia). En la arquitectura de MIPS de ciclo único estudiada en clase, suponer las siguientes latencias:

I-MEM	Add	Mux	ALU	BancoReg	D-MEM	Control	Ext_signo	Despl_iz-2	And
500 ps	150 ps	30 ps	180 ps	220 ps	1000 ps	65 ps	90 ps	20 ps	20 ps

a. ¿Cuál es el camino crítico para una instrucción **`and`**? ¿Cuál es la duración del ciclo de reloj si el único tipo de instrucción admitida son las instrucciones en la ALU (**`add`**, **`and`**, etc.)?
b. ¿Cuál es el camino crítico para una instrucción de carga **`lw`**? ¿Cuál es la duración del ciclo de reloj si sólo se admiten instrucciones de carga (**`lw`**)?
c. ¿Cuál es el camino crítico para una instrucción de salto condicional **`beq`**?
d. ¿Cuál es la duración del ciclo de reloj si se admiten las instrucciones **`add, beq, lw y sw`**?

4.3. Si partimos como base de la arquitectura de MIPS de ciclo único estudiada en clase, se pide incluir de forma independiente en la ruta de datos tres nuevas instrucciones no implementadas:

1. `sll $rd, $rt, despl` **#** rd <= rt << despl
2. `jal address` **#** PC <= (PC+4)[31:28] & address & "00"
3. `add3 $rd, $rs, $rt, $rx` **#** rd <= rt + rs+ rx

a. En el caso de que los haya, ¿cuáles de los bloques ya existentes pueden utilizarse para la nueva instrucción?
b. En el caso de que se pudiera, ¿qué bloques funcionales nuevos se necesitarían para la nueva instrucción?
c. En el caso de que las haya, ¿qué nuevas señales de control es necesario añadir a la unidad de control para poder ejecutar la nueva instrucción?

4.4. Dada la arquitectura uniciclo para MIPS estudiada en clase, suponer que una de las siguientes señales de control funciona mal: i*) RegWrite, y ii) MemWrite*
a) Suponer que siempre vale '0' con independencia del valor que se quiera aplicar
b) Suponer que siempre vale '1' con independencia del valor que se quiera aplicar
Explicando en cada caso y para señal el motivo, ¿qué instrucciones del set estudiado funcionarán mal?

4.5. Dada la arquitectura de ciclo único de MIPS, se señalan dos instrucciones en lenguaje máquina 0x8C430010 y 0x1023000C. Se conoce que todos los datos en memoria valen 0xFF y que el contenido de los registros señalados en la tabla adjunta es:

$0	$at	$v0	$v1	$a0	$a1	$a2	$t0	$t4	$ra
0	-16	-2	-3	4	-10	-6	-1	8	-4

Para cada una de las instrucciones anteriores, se pide:

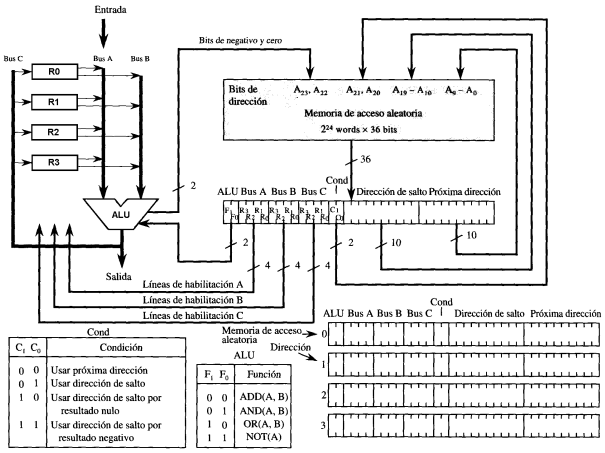
a. ¿Cuál es la salida de la extensión de signo y la salida de la unidad "Despl.Izq.2" ubicado en la ruta de datos del salto incondicional?
b. ¿Cuáles son los valores en binario de las entradas de la unidad de control de la ALUControl[2:0]?
c. ¿Cuál es la nueva dirección del registro PC después de ejecutar la instrucción?
d. Mostrar los valores en hexadecimal de las salidas de cada multiplexor durante la ejecución. Si se desconoce el valor indicarlo con una X.
e. ¿Cuáles son los valores en hexadecimal de las entradas de la ALU y de las 2 unidades de suma?
f. ¿Cuáles son los valores de las entradas del banco de registros?. Señale en binario las entradas de 5 bits y en hexadecimal la de 32 bits.

FUNDAMENTOS DE MICROPROCESADORES
PROBLEMAS DE LA UNIDAD 4.- EL PROCESADOR II: LA RUTA DE DATOS y SU CONTROL

4.6. En la figura adjunta se muestra una arquitectura particular. La ruta de datos tiene cuatro registros y una ALU. El control de la ruta de datos es de ciclo único y se realiza por medio de una memoria y de un registro que almacena la palabra de control (microinstrucción) en cada caso leída. Se pide:

a) Escribir en las 4 primeras posiciones de la memoria de control, la palabra necesaria para ejecutar las siguientes sentencias

M0: R1 ← ADD (R2 , R3)
M1: Saltar SI NEGATIVO (N=1) a 15_{10} ; (bneg 15)
M2: R3 ← AND (R1 , R2)
M3: Saltar siempre a 20_{10} ; (jump 20)



b) Escribir un seudo-código ensamblador y las palabras de control equivalentes que implementen la operación de multiplicar R2 <= R0xR1. Considerar que inicialmente R2 = 0 y R3=1. El programa debe terminar en un bucle infinito y el microcódigo debe estar almacenado a partir de la posición 0 de la memoria.

Nota: Si bien no hay líneas que así lo indiquen, debe interpretarse que los bits N y Z son ambos '0' cuando C_1C_0 son '00'. Esto significa que si no hay posibilidad de salto, las líneas A_{22} y A_{23} son ambas '0'.
Nota: Cada bit de los campos A, B y C corresponde directamente a un registro. Así, la palabra 1000 selecciona el registro R3, no el registro R8, inexistente

4.7. En la primera columna de la tabla adjunta se muestra un programa escrito para MIPS que se ejecuta en la arquitectura uniciclo estudiada. Se pide indicar en una tabla el de las señales de control indicadas, al ejecutar el código dado, así como el valor final de los registros indicados.

Registros:					
$pc	$s1	$s2	$s3	$s4	$s5

Datos iniciales: $pc = 0x0800, $s1 = $s2 = $s3 = $s4 = $s5 = 0x00

Señales de Control						
Jump	MentoReg	MemWrite	Branch	RegDst	RegWrite	ALUScr

Código

```
    .text 0x0800
    lw $s1, B($0)
    lw $s2, C($0)
    and $s3, $s1, $s2
    sw $s3, D($0)
    add $s4, $s1, $s2
    lw $s5, D($0)
fin: j fin
    .data 0x2000
A: 0x00000005
B: 0x0000000C
C: 0x00000007
D: 0x0000002F
```

4.8. En la arquitecura uniciclo del procesador MIPS, se ejecuta el código adjunto. Se pide definir en una tabla el valor de las señales de control indicadas, necesarias para ejecutar el código dado, así como el valor final de los registros señalados, sabiendo que en el ciclo 1 de reloj se está ejecutando la primera instrucción.

Registros:				
$pc	$s1	$s2	$s3	$ra

Datos iniciales: $pc = 0x00, $s1 = 0x208C; $s2 = 0x2140; $s3 = 0x2000; $ra = 0x00

Señales de Control							
Jump	MentoReg	MemWrite	Branch	RegDst	RegWrite	ALUScr	PCScr

Código

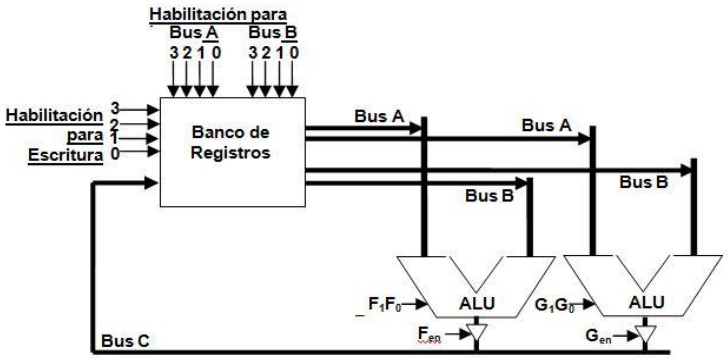
```
  .text 0x0000
    add $s3, $s1, $s2
    beq $s1, $s2, eti
    jal eti
    lui $s1, 0x1000
  .text 0x0100
eti: sw $s3, 4($s1)
    lw $s2, 4($s1)
    xor $s1, $s2, $s2
```

Nota: Si se accede a una posición de memoria cuyo valor no se puede conocer por los datos del enunciado, se supondrá que el contenido de dicha posición de memoria es 0x0A.

FUNDAMENTOS DE MICROPROCESADORES
PROBLEMAS DE LA UNIDAD 4.- EL PROCESADOR II: LA RUTA DE DATOS y SU CONTROL

4.9. En la figura adjunta se muestra un sistema con dos ALUs independientes, cuyos operandos se leen/escriben de/en un banco de 4 registros. Se adjunta una tabla para el control del sistema en donde se señalan las 4 operaciones que cada ALU puede ejecutar.

Código F_1 F_0	Función	Sintaxis Instrucción
00	C = A + B	ADD (A,B,C)
01	C = A + 1	INC (A,1,C)
10	C = A * 2	MUL2 (A,2,C)
11	C = A * 8	MUL8 (A,8,C)
Código G_1 G_0	Función	Sintaxis Instrucción
00	C = A OR B	OR (A,B,C)
01	C = A AND B	AND (A,B,C)
10	C = A XOR B	XOR (A,B,C)
11	C = A NAND B	NAND (A,B,C)



a) Escribir el seudocódigo para ejecutar una secuencia de control que calcule:
- **a.1.** La función XOR de los contenidos de los registros R_0 y R_1 y deje el resultado en el registro R_0. ($R_0 \leftarrow R_0 \oplus R_1$)
- **a.2.** La diferencia entre los contenidos de los registros R_1 y R_2 y deje el resultado en el registro R_3 ($R_3 \leftarrow R_1 - R_2$)
- **a.3.** El producto de los contenidos de los registros R_0 y R_1 y deje el resultado en el registro R_2. Suponer inicialmente que $R_2 = 0$ y $R_3 = 1$. Suponer que existen las mismas instrucciones de salto (brach y jump) vistas en MIPS. ($R_2 \leftarrow R_0 \times R_1$)

b) Identificar las señales de control para realizar las instrucciones pertinentes en cada uno de los programas anteriores. En el caso 3, no indicar el control para las señales de salto utilizadas.

Nota: Para leer/escribir de/en uno de los 4 registros, se debe aplicar una palabra de control en el bus que corresponda (para R_0 será "0001" y para R_3 será "1000"). Para activar la salida de una de las dos ALUs, se debe activar '1' la señal de habilitación correspondiente F_{en} o G_{en}.

4.10. Se ejecuta el código adjunto escrito para MIPS con arquitectura uniciclo. En el código dado, también figura parte del contenido de memoria. Se pide definir en una tabla el valor de las señales de control indicadas, necesarias para ejecutar el código dado, así como el valor final de los registros señalados, sabiendo que en el ciclo 1 de reloj se está ejecutando la primera instrucción. Señale también el contenido final de las posiciones de memoria indicadas (A, B, C y D).

Código

```
    .text 0x0000
    lw $s1,4($s2)
    or $s3,$s1,$s2
    jal etiq
    add $s1,$s1,$s2
etiq: sw $s3,D($0)
    addi $s3,$s2,4
    sw $s1,-4($s3)
.data 0x2000
A: 0x00000001
B: 0x00000010
C: 0x00000100
D: 0x00001000
```

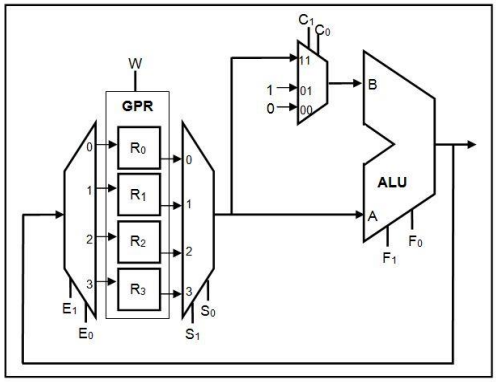
Registros:				
$pc	$s1	$s2	$s3	$ra

Datos iniciales: $pc = 0x0000, $s1 = 0x2000; $s2 = 0x2004; $s3 = $ra = 0x0000

Señales de Control						
Jump	MentoReg	MemWrite	Branch	RegDst	RegWrite	ALUScr

4.11. En la tabla adjunta se muestran los bits de control de la ALU que figura en la ruta de datos de un cierto sistema digital. Los demás controles deben ser identificados por el estudiante, puesto que controlan a tres multiplexores y la línea W se utiliza para permitir la escritura en el banco de registros (GPR).

F_1	F_0	Función
0	0	A **+** B
0	1	A **OR** B
1	0	A **AND** B
1	1	A **NAND** B



Se quiere ejecutar una operación que intercambie los contenidos de los registros **R_3 y R_0**. Señale la palabra de control utilizando el menor número de ciclos que sean necesarios. En cada ciclo justificar brevemente la respuesta.

Nota: Todos los registros del GPR son de lectura/escritura y se puede utilizar cualquiera de ellos en la operación.

FUNDAMENTOS DE MICROPROCESADORES
PROBLEMAS DE LA UNIDAD 4.- EL PROCESADOR II: LA RUTA DE DATOS y SU CONTROL

4.12. Sea un sistema procesador basado en MIPS con arquitectura uniciclo igual que el estudiado en clase. Se quiere ejecutar el código que se adjunta. Se pide definir en una tabla el valor de las señales de control indicadas, necesarias para ejecutar el código dado, así como el valor final de los registros señalados, sabiendo que en el ciclo T de reloj se está ejecutando la primera instrucción. Señalar el contenido final de las posiciones de memoria indicadas (A, B y C).

Registros:				
$pc	$s1	$s2	$s3	$ra

Datos iniciales: $pc = $ra = 0x0000, $s1 = 0x3F41; $s2 = 0x2004; $s3 = 0x003F

Señales de Control						
Jump	MentoReg	MemWrite	Branch	RegDst	RegWrite	ALUScr

Código

```
    .text 0x0000
        lw $s1, 4($s2)
        and $s2, $s1, $s3
        beq $s1,$s2,etiq
        sw $s2, -4($s3)
        j fin
    .text 0x001C
etiq:   jal fin
        or $s3,$s1,$s2
fin:    addi $s3,$s1,5
        sw $s3,C($0)
    .data 0x2000
A:      0x00000002
B:      0x00000007
C:      0x00000013
```

4.13. Sea un sistema procesador basado en MIPS con arquitectura uniciclo igual que el estudiado en clase. Se quiere ejecutar el código que se adjunta:

Se pide definir en una tabla el valor de las señales de control indicadas, necesarias para ejecutar el código dado, así como el valor final de los registros señalados y el de las posiciones de memoria indicadas. Complete la tabla hasta el ciclo T+6, sabiendo que en el ciclo T de reloj se está ejecutando la primera instrucción. Complete toda la información de la tabla que se pueda, pero no añada más columnas aunque queden instrucciones sin ejecutar.

Registros:			
$pc	$s1	$s2	$s3

Datos iniciales: $pc = 0x0000, $s1 = 0xABCD; $s2 = 0x1234; $s3 = 0x0002

Señales de Control						
MemWrite	RegWrite	MentoReg	RegDst	ALUScr	Branch	Jump

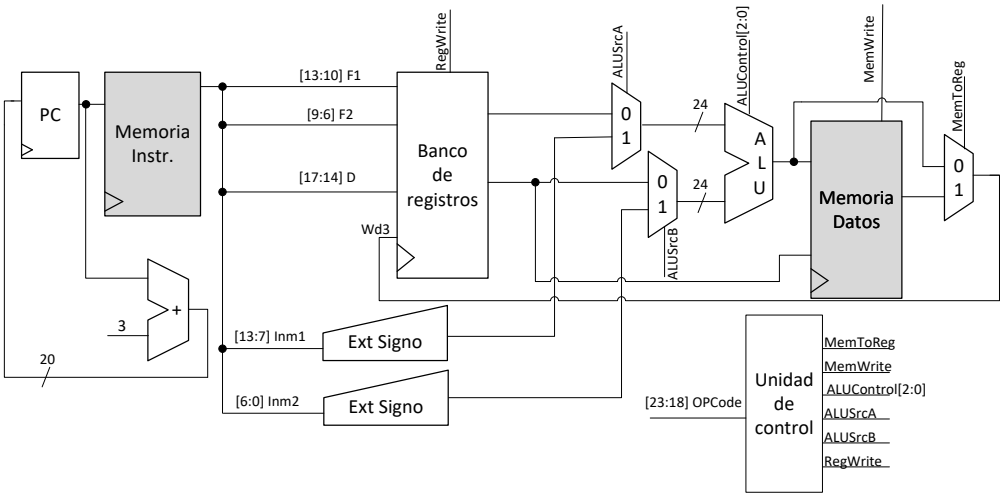
Código

```
.text 0x0
addi $s1, $0, 4
lw $s2, A($s1)
j eti1
.text 0x100
addi $s2, $0, 4
eti1: beq $s2, $s3, eti2
addi $s2, $0, 8
eti2: sw $s2, B($s1)
sllv $s1, $s2, $s3
```

```
    .data 0x2000
A:      0x00000001
B:      0x00000002
C:      0x00000003
```

FUNDAMENTOS DE MICROPROCESADORES
PROBLEMAS DE LA UNIDAD 4.- EL PROCESADOR II: LA RUTA DE DATOS y SU CONTROL

4.14. En la figura que se adjunta se puede observar el esquemático de un procesador no MIPS. Se conocen tres tipos de instrucciones en las que siempre hay un registro de destino (D), pero donde los operandos fuentes pueden ser: dos registros (F1, F2), dos datos inmediatos (Inm1, Inm2), o un registro (F1) y un dato inmediato (Inm2).



Sabiendo esa información conteste a las siguientes preguntas.

a) ¿Qué formato tiene una instrucción con dos registros fuente como operandos fuente? Utilice los nombres del esquemático (D, F1…) e indique cuántos bits ocupa cada campo.

b) ¿Qué formato tiene una instrucción con dos datos inmediatos como operandos fuente? Utilice los nombres del esquemático (D, F1…) e indique cuántos bits ocupa cada campo.

c) ¿Qué formato tiene una instrucción con un registro y un dato inmediato como operandos fuente? Utilice los nombres del esquemático (D, F1…) e indique cuántos bits ocupa cada campo.

d) ¿Qué tamaño de palabra tiene este procesador? Justifique brevemente la respuesta.

e) ¿Cuántos bytes ocupa una instrucción? Justifique brevemente la respuesta.

f) ¿Cuál es la máxima capacidad de la memoria de instrucciones, expresada en bytes? Justifique brevemente la respuesta.

g) ¿Cuántos registros pueden ser direccionados con este procesador? Justifique brevemente la respuesta.

h) ¿Cuántas operaciones diferentes como máximo puede realizar la ALU? Justifique brevemente la respuesta.

4.15. Durante la ejecución de un cierto código ensamblador para MIPS, en el ciclo de reloj de valor T, el valor de las señales de control, el de alguno de los registros y el de algunas posiciones de la memoria de datos, son los señalados en las tablas facilitadas. Se pide:

CONTROL ciclo actual T						
ALUSrc	Jump	MemtoReg	MemWrite	PCSrc	RegDst	RegWrite
1	**0**	**X**	**1**	**0**	**X**	**0**

REGISTROS ciclo actual T			
$s1	$s2	$s3	$s4
0x00FF	**0x2000**	**0x200C**	**0x0000**

a) Analizando las señales de control para el ciclo de reloj actual de valor T, identificar la instrucción que se está ejecutando y completar el código restante con el valor en binario de las señales de control indicadas.

b) Completar con su valor en hexadecimal el de los registros y el de las posiciones de memoria indicadas (A, B, C) al ejecutar el código facilitado.

Código

```
.text 0x0000
        addi $s1, $0, 0x2000
        lw $s2, 8($s1)
        add $s1, $s2, $s3
.text 0x0020
        and $s2, $s1, $s3
        sw $s2, -4($s3)
        beq $s1, $s2, etiq
        lw $s4, 4($s2)
        addi $s3, $s1, -1

        or $s2, $s1, $s2
fin:    j fin
etiq:   lw $s3,C($0)
        slt $s3, $s2, $s1
        add $s2, 4($s1)
.data 0x2000
A: 0x0001
B: 0xFF00
C:0x0400
```

Código

FUNDAMENTOS DE MICROPROCESADORES
PROBLEMAS DE LA UNIDAD 4.- EL PROCESADOR II: LA RUTA DE DATOS y SU CONTROL

4.16. Se adjunta un código escrito para un sistema basado en MIPS, con arquitectura uniciclo igual que la estudiada en la asignatura. El programa comenzó su ejecución con el valor $pc = 0x0000, y se sabe que en el ciclo actual T, $pc = 0x00000104 y comienza la ejecución de una nueva instrucción.

```
.text 0x0000
        lw $s1, A($0)
        lw $s2, B($0)
        and $s3, $s1, $s2
        j eti_1
.text 0x100
        add $s3, $s1, $s2
eti_1:  beq $s2, $s3, eti_2
        addi $s1, $s1, - 8
eti_2:  sub $s2, $s2, $s3
        sw $s2, A($s1)
        xor $s2, $s1, $s1
```

Contenido **actual** (**T**) memoria datos
.data 0x2000
A: 0x00000010
B: 0x000000FF
C: 0x000000AA

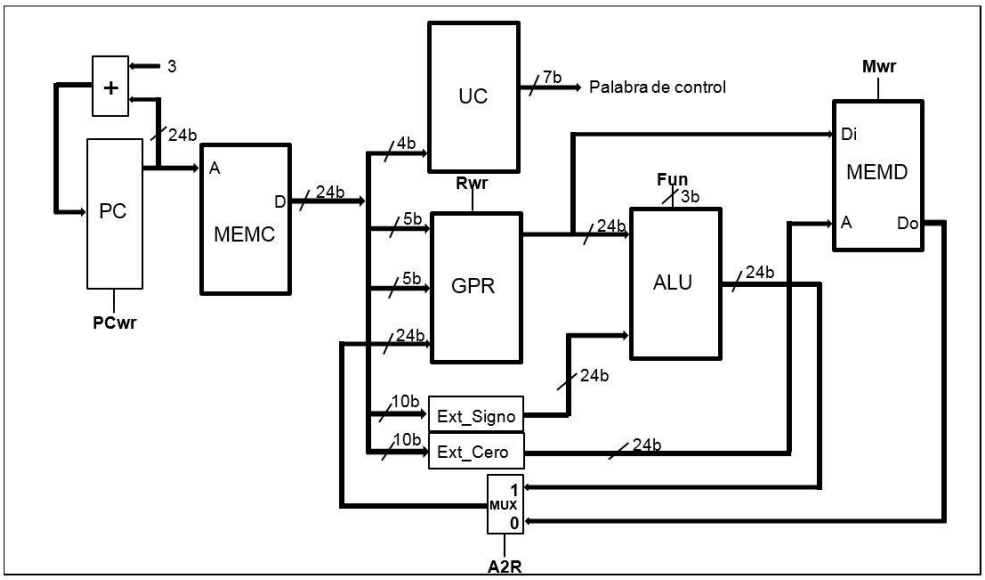
a. Con la información facilitada, se pide completar en una tabla los valores de las señales de control indicadas desde la instrucción actual (ciclo "T") hasta completar 4 ciclos señalados (ciclo "T+3").

MemWrite	**RegWrite**	**MemtoReg**	**RegDst**	**ALUSrc**	**Branch**	**Jump**

b. Señalar el valor de los registros y de las posiciones de memoria indicados, antes de terminar el ciclo "T+4". Si se desconoce algún valor de los solicitados, escriba en la respuesta el valor 0xFFFF.

%pc	**$s1**	**$s2**	**$s3**	**%ra**	**A**	**B**	**C**

4.17. En la figura adjunta se muestra el esquema de un determinado procesador uniciclo. En la figura se distinguen cinco bloques fundamentales para el procesador, las dos unidades de memoria (MEMC, MEMD), el banco de registros (GPR), la unidad aritmético-lógica (ALU) y la unidad de control (UC). Asimismo el esquema presenta otros elementos digitales conocidos que junto a los bits de control, se emplean para configurar la ruta de datos de este sistema.
En base a la información facilitada en el esquema, justificando de forma breve cada respuesta, se pide:



a) Señale el tamaño de palabra en bits del procesador.
b) Señale el máximo número de instrucciones diferentes para este procesador.
c) Señale el máximo número de registros en el GPR.
d) Escriba la palabra de control para la instrucción **addi r1, r2, 8** # (r1 = r2 + 8) (**$Fun_{(2:0)}$ =** XXX**)**
e) Escriba la palabra de control para la instrucción **sw r1, 8** # (r1 => MEMD(8)) (**$Fun_{(2:0)}$ =** XXX**)**
f) Suponiendo que la ALU tenga incluida las operaciones necesarias, señale las razones que impiden ejecutar las siguientes instrucciones.

add r1, r2, r3	#(r1 = r2 + r3)
lw r1, 5(r2)	# r1 = MEMD[(5+r2)]