

COMPUTER STRUCTURE

UNIT 4.- PROCESSOR II: DESIGN AND CONTROL OF THE DATAPATH. SINGLE-CYCLE

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. The different elements of the processor have their own delay (delay: time from a change in the input to a change in the output). The critical path for an instruction is given by adding the delays in the longest path. For the single-cycle MIPS architecture, consider the following delays:

I-MEM	Add	Mux	ALU	RegFile	D-MEM	Control	Sign-ext	ShiftLeft2	And
500 ps	150 ps	30 ps	180 ps	220 ps	1000 ps	65 ps	90 ps	20 ps	20 ps

- a. Which is the critical path for an “and” instruction? Which would be the clock cycle if all the instructions were ALU instructions (“and”, “add”, etc)?
- b. Which is the critical path for a “lw” instruction? Which would be the clock cycle if all the instructions were “lw” instructions?
- c. Which is the critical path for a “beq” instruction?
- d. Which would be the clock cycle if the instructions possible instructions are “and”, “beq” and “lw”?

4.3. Starting from the single-cycle MIPS architecture studied in class, we want to include three new instructions (independently from each other):

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. Which are the blocks (if any) that can be used for the new instructions?
- b. Which are the new blocks (if any) that are needed for each instruction?
- c. Which are the new control signals (if any) that are needed for each instruction?

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. The machine code instructions 0x8C430010 and 0x1023000C are going to be executed in a single-cycle MIPS. All memory addresses have the value 0x0FF and the initial content of the registers is shown in the following table:

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

For each of the previous instructions answer the following questions:

- a) Which is the output of the sign-extend block and the “Shift.Left.2” block in the unconditional jump datapath?
- b) Which is the binary value of the ALU control signal ALUControl[2:0]?
- c) Which is the new value of PC after the instruction?
- d) Write the hexadecimal value of the output of each multiplexer during the execution. If the value can not be known, write X.
- e) Write the hexadecimal values of the ALU inputs and the inputs of both adding blocks.
- f) Write the values of all the inputs to the register file. Write it in binary for the 5-bit inputs and in hexadecimal for the 32-bit inputs.

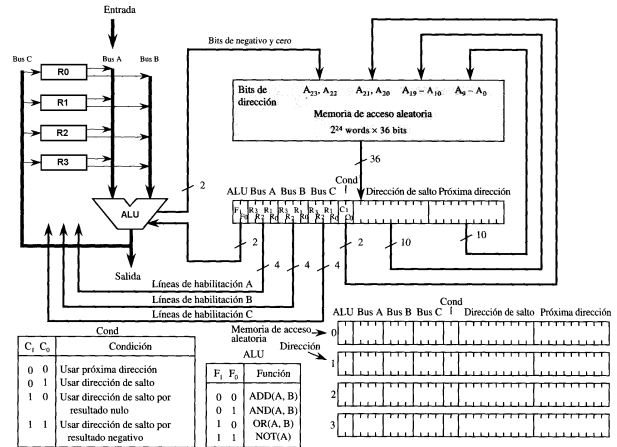
COMPUTER STRUCTURE

UNIT 4.- PROCESSOR II: DESIGN AND CONTROL OF THE DATAPATH. SINGLE-CYCLE

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

- b) Escribir un pseudo-código ensamblador y las palabras de control equivalentes que implementen la operación de multiplicar $R2 \leftarrow R0 \times R1$. 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. The following program is executed in a single-cycle MIPS processor. Please, fill in the signals and registers values of the table when the code is executed.

Note: The table includes the initial content of four data memory elements.

Registers:					
\$pc	\$s1	\$s2	\$s3	\$s4	\$s5

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

Control signals						
Jump	MentoReg	MemWrite	Branch	RegDst	RegWrite	ALUScr

Code
.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 arquitectura 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.

COMPUTER STRUCTURE

UNIT 4.- PROCESSOR II: DESIGN AND CONTROL OF THE DATAPATH. SINGLE-CYCLE

4.9. The following figure shows a single-cycle architecture with two independent ALUs whose operands are read/written from a 4 registers register file. The table shows the control signals of the ALUs and the operations that can be executed.

a) Write the code for a program that does the following:

a.1. XOR of the contents of the registers R_0 y R_1 storing the result in register R_0 .

$$(R_0 \leftarrow R_0 \oplus R_1)$$

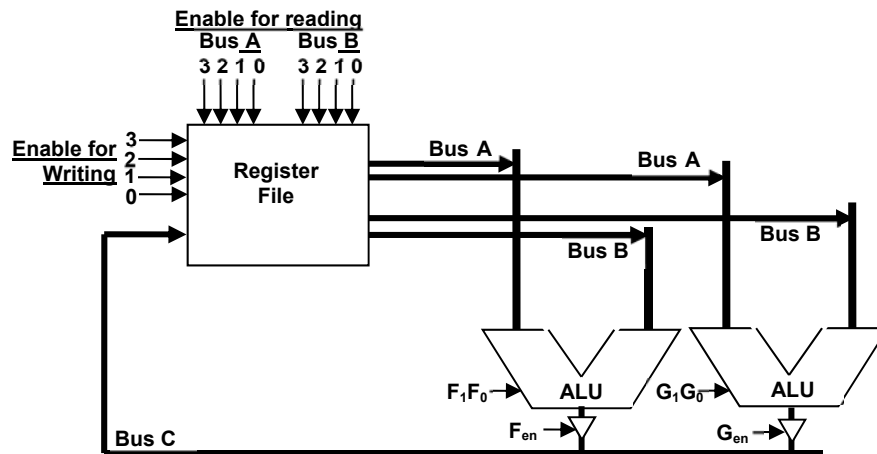
a.2. The difference of registers R_1 and R_2 storing the result in register R_3

$$(R_3 \leftarrow R_1 - R_2)$$

a.3. The product of registers R_0 and R_1 storing the result in register R_2 . Suppose the initial values $R_2 = 0$ and $R_3 = 1$. Also suppose that the same control flow instructions of MIPS (branch and jump) are available.

$$(R_2 \leftarrow R_0 \times R_1)$$

b) Write the control signals for the instructions of each of the previous codes. For case 3, ignore the control signals of the control flow instructions.



Code $F_1 F_0$	Function	Instruction Syntax	Code $G_1 G_0$	Function	Instruction Syntax
00	$C = A + B$	ADD (A,B,C)	00	$C = A \text{ OR } B$	OR (A,B,C)
01	$C = A + 1$	INC (A,1,C)	01	$C = A \text{ AND } B$	AND (A,B,C)
10	$C = A * 2$	MUL2 (A,2,C)	10	$C = A \text{ XOR } B$	XOR (A,B,C)
11	$C = A * 8$	MUL8 (A,8,C)	11	$C = A \text{ NAND } B$	NAND (A,B,C)

Note: In order to read/write in any of the 4 registers, the appropriate control word must be applied in the corresponding bus A, B or C (for R_0 it would be "0001" and for R_3 it would be "1000"). For enabling the output of one of the two ALUs, F_{en} or G_{en} must be set to '1'.

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).

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

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
```

COMPUTER STRUCTURE

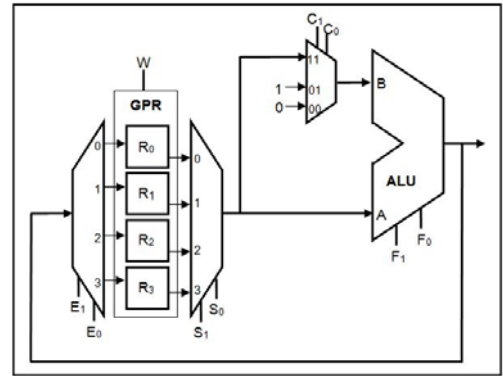
UNIT 4.- PROCESSOR II: DESIGN AND CONTROL OF THE DATAPATH. SINGLE-CYCLE

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 **R₃** y **R₀**. 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.



4.12. Sea un sistema procesador basado en MIPS con arquitectura uniclo 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,eti1
sw $s2, -4($s3)
j fin
.text 0x001C
eti1: 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 uniclo 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

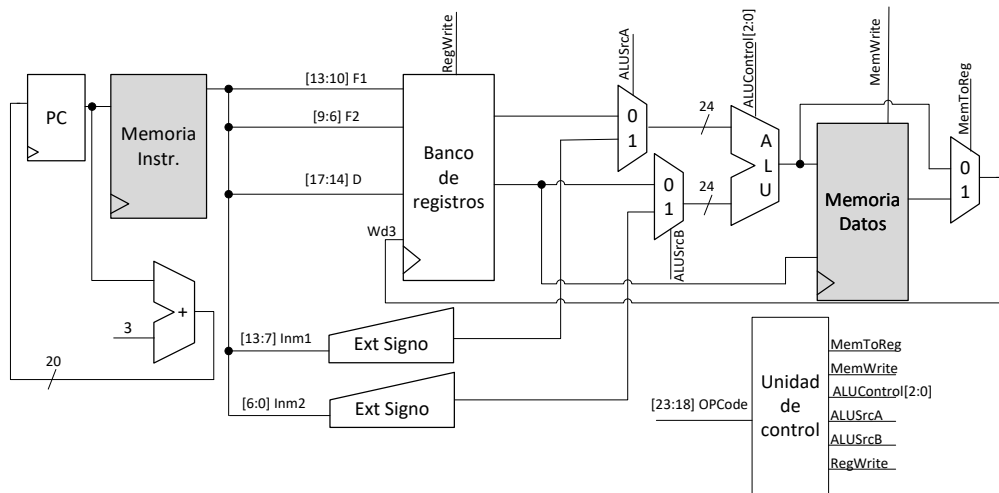
```
.text 0x0
addi $s1, $0, 4
lw $s2, A($s1)
j eti1
.text 0x100
addi $s2, $0, 4
eti1: beq $s2, $s3, eti2
addi $s2, $0, 8
eti2: sw $s2, B($s1)
sllv $s1, $s2, $s3
```

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

COMPUTER STRUCTURE

UNIT 4.- PROCESSOR II: DESIGN AND CONTROL OF THE DATAPATH. SINGLE-CYCLE

4.14. The figure shows the schematic of a microprocessor different from MIPS. There are three instruction types. In all the three types there is destination register (D), but the source operands can be: two registers (F1, F2), two immediates (Inm1, Inm2), or one register (F1) and an immediate (Inm2).



Please, answer the following questions.

- Which is the format of an instructions with two registers as source operands? Use the names in the schematic (D, F1...) and indicate the number of bits for each field.
- Which is the format of an instructions with two immediates as source operands? Use the names in the schematic (D, F1...) and indicate the number of bits for each field.
- Which is the format of an instructions with one register and one immediate as source operands? Use the names in the schematic (D, F1...) and indicate the number of bits for each field.
- Which is the word size of this processor? Justify briefly your answer.
- How many bytes are necessary for the machine code of each instruction? Justify briefly your answer.
- Which is the maximum capacity of the instruction memory (in bytes)? Justify briefly your answer.
- How many registers are available in this processor? Justify briefly your answer.
- How many different operations can be performed in the ALU? Justify briefly your answer.

4.15. During the execution of the following MIPS code, in the cycle T the values of some control signals, registers and memory addresses are as indicated in the tables.

CONTROL current cycle T						
ALUSrc	Jump	MemtoReg	MemWrite	PCSrc	RegDst	RegWrite
1	0	X	1	0	X	0
REGISTERS current cycle T						
\$s1	\$s2	\$s3	\$s4			
0x00FF	0x2000	0x200C	0x0000			

- Analysing the control signals at cycle T, identify the instruction being executed and fill in the rest of the control signals table after executing the rest of the code.
- Fill in (in hexadecimal) the values of the registers and memory addresses of the table after executing the code (cycle T+5).

Code
.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

COMPUTER STRUCTURE

UNIT 4.- PROCESSOR II: DESIGN AND CONTROL OF THE DATAPATH. SINGLE-CYCLE

4.16. Se adjunta un código escrito para un sistema basado en MIPS, con arquitectura unicyclo 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 et1_1
.text 0x100
        add $s3, $s1, $s2
et1_1:   beq $s2, $s3, et1_2
        addi $s1, $s1, - 8
et1_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”).

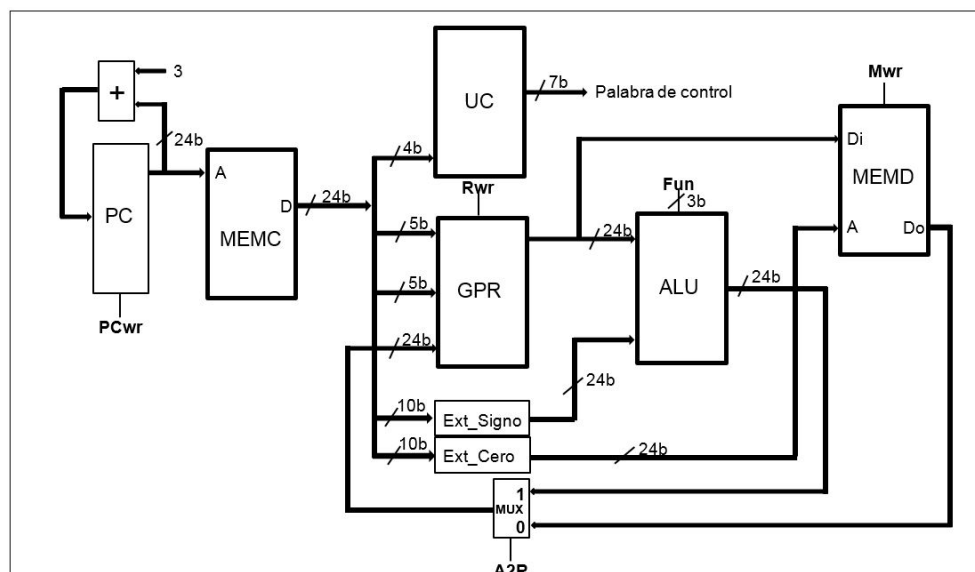
MemWrite	RegWrite	MemtoReg	RegDst	ALUSrc	Branch	Jump
-----------------	-----------------	-----------------	---------------	---------------	---------------	-------------

- 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. The figure shows the architecture of a single-cycle processor. There are five main blocks: both memories (MEMC, MEMD), register file (GPR), arithmetic-logic unit (ALU) and control unit (UC). There are other known digital blocks that, along with the control signals, are used for configuring the datapath. According to the shown architecture, answer the following questions briefly justifying each answer:

According to the shown architecture, answer the following questions briefly justifying each answer:



- Indicate the word size of this processor.
- Maximum number of different instructions in this processor.
- Maximum number of registers in the GPR.
- Write the control signals for the instruction **addi r1, r2, 8** # (r1 = r2 + 8) (**Fun**_(2:0) = XXX)
- Write the control signals for the instruction **sw r1, 8** # (r1 => MEMD(8)) (**Fun**_(2:0) = XXX)
- Supposing the ALU can perform the necessary operations, indicate other reasons why the following instructions can not be performed.

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