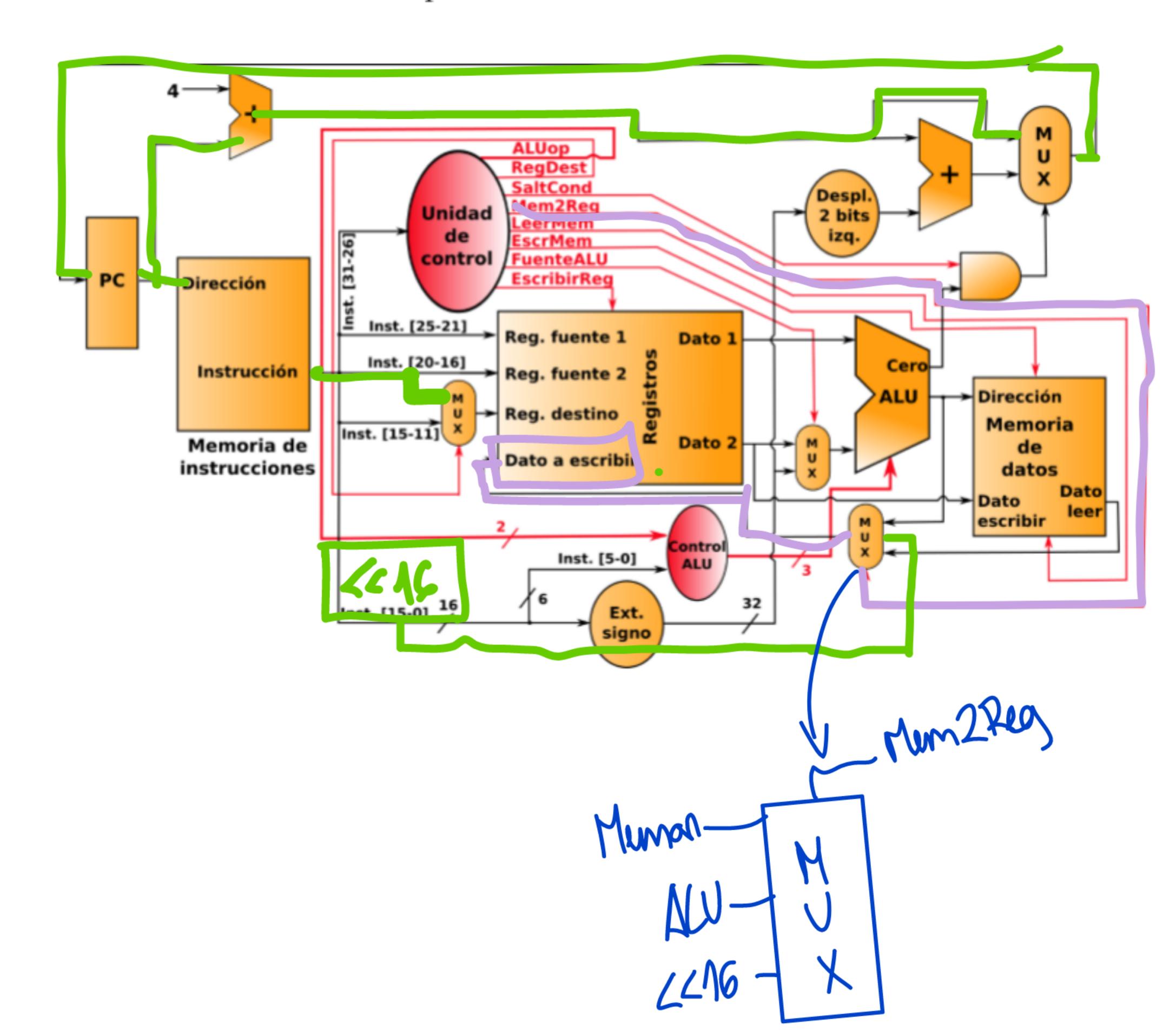
### FUNDAMENTOS DE COMPUTADORES.

## Tema 5. Sección de procesamiento y de control.

1. Suponiendo el camino de datos para el MIPS visto en clase se asignan diferentes tiempos de ejecución a las unidades funcionales que la componen. Calcular el tiempo por instrucción y el rendimiento de las distintas configuraciones con respecto a la más rápida. ¿Cuál es el CPI?

	Config. A	Config. B	Config. C	Config. D
ALU	2 ns	2 ns	2 ns	2  ns
Sumador PC	2 ns	2 ns	1 ns	$1 \mathrm{\ ns}$
Sumador saltos	2 ns	2 ns	1 ns	2 ns
Banco de registros	1 ns	$1 \mathrm{\ ns}$	1 ns	$1 \mathrm{\ ns}$
Memoria instrucc.	2 ns	1 ns	2 ns	$1 \mathrm{\ ns}$
Memoria datos	2 ns	2 ns	2 ns	2 ns

- 2. A partir de la implementación del camino de datos visto en clase (ver figura):
  - Añadir el hardware necesario para que se pueda ejecutar la instrucción lui.
  - Especificar la ruta de datos sobre el dibujo.
  - Modificar el control para incluir la nueva instrucción.



#### Control de la ALU

La unidad de control genera 2 bits (**ALUop**) a partir del código de operación

A partir de esos dos bits y del campo funct generamos el código de control de la ALU

Instrucción	code	ALUop	funct	Acción de	Control de	
IIIStruction			Tunct	la ALU	la ALU	
lw	100011	00	XXXXXX	suma	010	
sw	101011	00	XXXXXX	suma	010	
beq	000100	01	XXXXXX	resta	110	
add	000000	10	100000	suma	010	
sub	000000	10	100010	resta	110	
and	000000	10	100100	and	000	
or	000000	10	100101	or	001	
			Codiop de frago	<b>*</b>		

## Señales de control

Ci<mark>III</mark>US

Señal	Efecto si no activa (0)	Efecto si activa (1)	Activa
			en
RegDest	Reg. destino, campo rt	Reg. destino, campo	add,sub
	(bits 20-16)	rd (bits 15-11)	and,or
SaltCond	PC←PC+4	Si cero, salta	beq
Mem2Reg	Reg. destino ←ALU	Reg. destino ← Mem	lw
LeerMem	Nada	Lee memoria de datos	lw
EscrMem	Nada	Escribe memoria de	sw
		datos	
FuenteALU	ALU2←Reg. 2	ALU2←Inmediato	lw, sw
EscribirReg	Nada	Escribe en registro	add,sub,
			and,or,lv

Tipo de instrucció n	Cargar instrucción(mem oria de instrucciones)/Su mador PC	nco de		Memoria de datos (leer o escribir)		A (ns) (Sumato rio)	B (ns)	C (ns)	D (ns)
Aritmétic	Cargar	Leer	ALU (2)		Escribir	6	5	6	5
a	instrucción(2,1,2,	registros (1)	,		registro (1)				
lw	Cargar instrucción(2,1,2, 1)	Leer registros (1)	ALU (2)	Leer memoria (2)	Escribir registro (1)	8	7	8	7
SW	Cargar instrucción(2,1,2, 1)	Leer registros (1)	ALU (2)	Escribir memoria (2)		7	6	7	6
Salto	Cargar	Leer	ALU (2)			5	4	5	4
condicion		registros (1)	, ,						
Jump	Cargar instrucción(2,1,2, 1)					2	Acceso a la memoria de instrucciones es + rapido que el del PC	2	1
Tiempo de ciclo						8	7	8	7
						- La lw	es la instrucción m	ás lenta	

- El tiempo de ciclo lo determina el tiempo de la instrucción + lenta -> lw

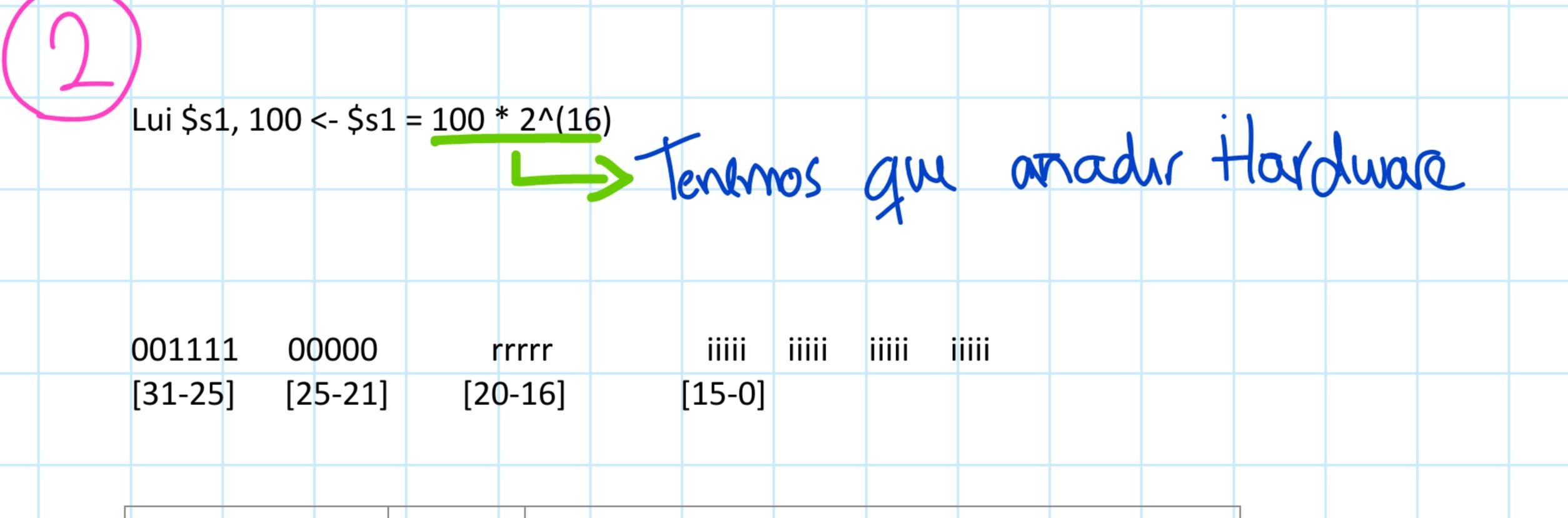
10 -> lui

**16**/<sub>30</sub>

**18**/30

- Las configuraciones B y D son las + rápidas -> Rendimiento 8/7 sobre A y C ( se ve en la instrucción + lenta)

- Procesador visto en clase -> monociclo -> 1 instrucción por ciclo == 1 ciclo por instrucción -> CPI=1.



RegDest	0	Reg destino en bits [20-16]	
SaltoCond	0	No es un salto	
LeerMem	0	No leemos la memoria	
EscrMem	0	No escribimos en memoria	
Mem2Reg (bit1 Mem2Reg (bit0	•	Seleccionamos de Inmediato desplazado 16 bits	
Fuente ALU	-	Nos da igual	
EscrReg	1	Escribimos el nuevo valor en el registro destino	
ALUop (bit1)	-	Nos da igual	
ALUop (bit0)	-	Nos da igual	
		nes Mem2Reg (bit1) = 0 siempre. $\frac{1}{100} = 0 - 100$	00
00 -> salida ALU 01 -> 'Dato leer'		oria de datos (ej lw) $2(SB = 00) \rightarrow 4/6$	. 01

# Instrucción lui

