M32:
Repaso instrucciones de máquina

- Registros: R0 a R31, R0 siempre es 0
- Operaciones aritméticas: ADD, ADDX, SUB, SUBX

  Ejemplo en formato 1: **ADD R3, R4, R11**
- Operaciones lógicas: AND, OR, XOR, SLL, SRL, SRA

  Ejemplo en formato 2: XOR R23, -1, R24
- Lectura de memoria: LDW, LDUH, LDSH, LDUB, LDSB

  Ejemplo de lectura: LDW [R1+R2], R3
- Escritura en memoria: STW, STH, STB

  Ejemplo de escritura: **STB R3, [R5-101]**
- Saltos condicionales: BA, BE, BNE, BG, BGE, BGU, ...

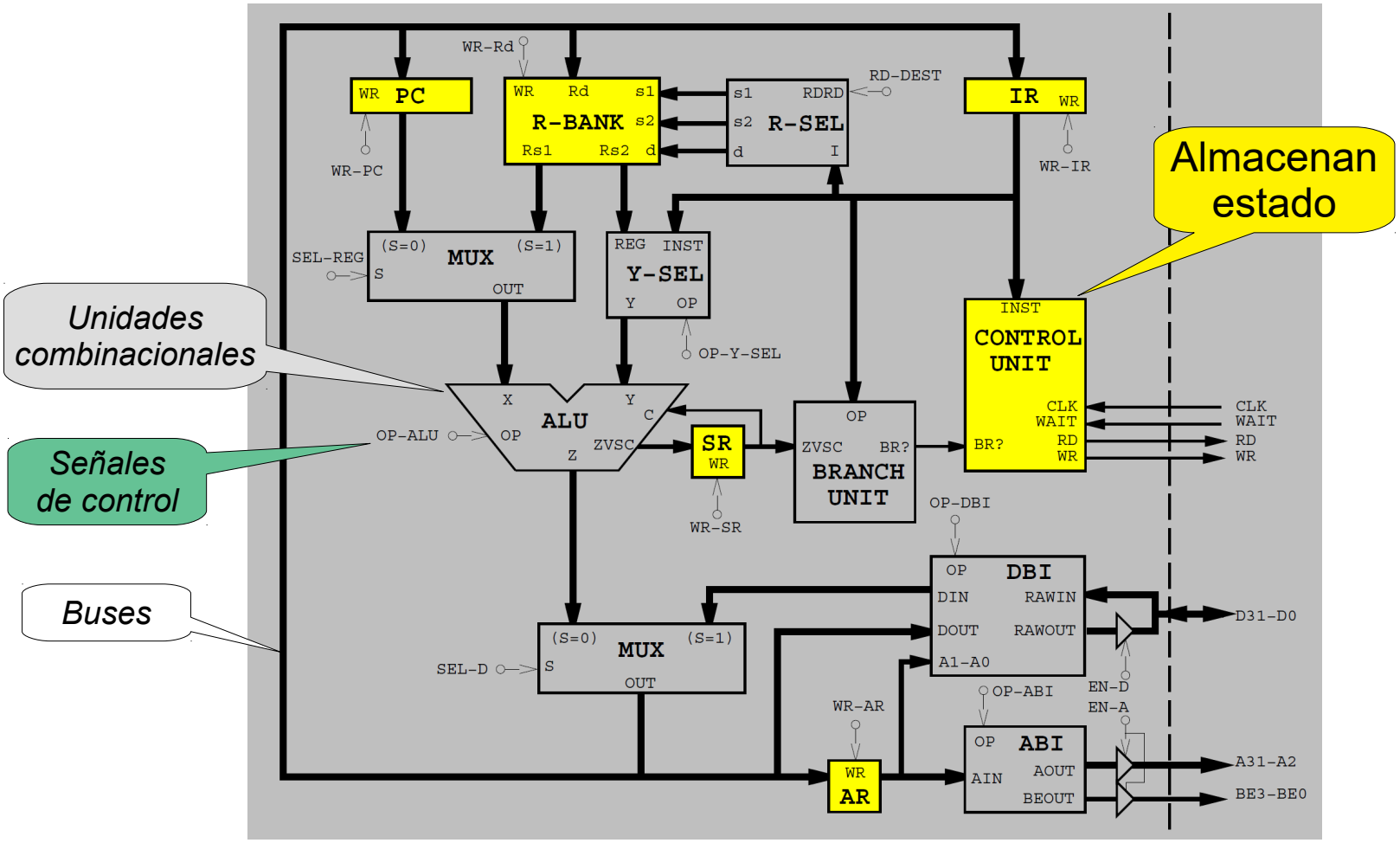
  Ejemplo de salto: SUB R5, 10, R0

  BG etiqueta
- Call y return: JMPL

  Ejemplo: JMPL [R1+R0], R31

M32: Implementación de una CPU

- La figura muestra la descomposición en bloques de M32:

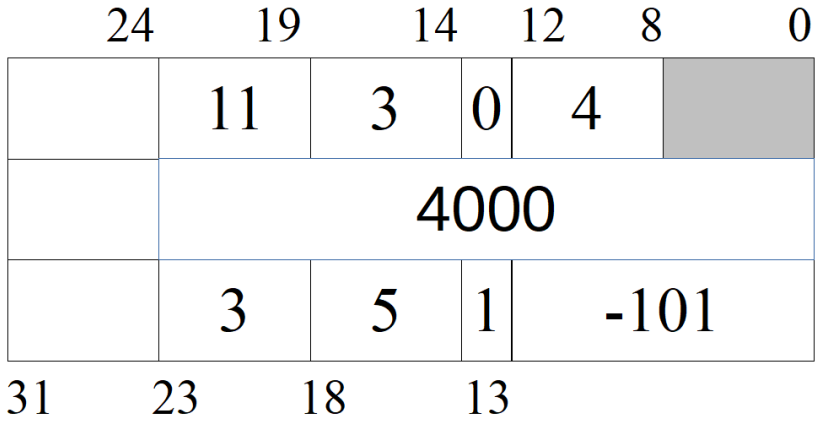


- Unidad de Control: genera las señales de control
- ALU: realiza operaciones aritmético/lógicas
- R-BANK: Almacena los 31 registros

Codificación de instrucciones

- Código de operación: bits 31 a 24
- Número del registro de destino: bits 23 a 19
- Número del primer registro fuente: bits 18 a 14
- Número del segundo registro fuente: bits 12 a 8 (si bit 13 es 0)
- Valor inmediato: bits 12 a 0 (si bit 13 es 1)
- Desplazamiento en saltos: bits 23 a 0

instrucción en assembler	*cod. op.*	*reg. dest*	*reg. src 1*		*reg. src 2 o val. imm.*
`ADD R3, R4, R11`		11	3	0	4
`BG` *etiqueta*		4000			
`STB R3, [R5-101]`		3	5	1	-101



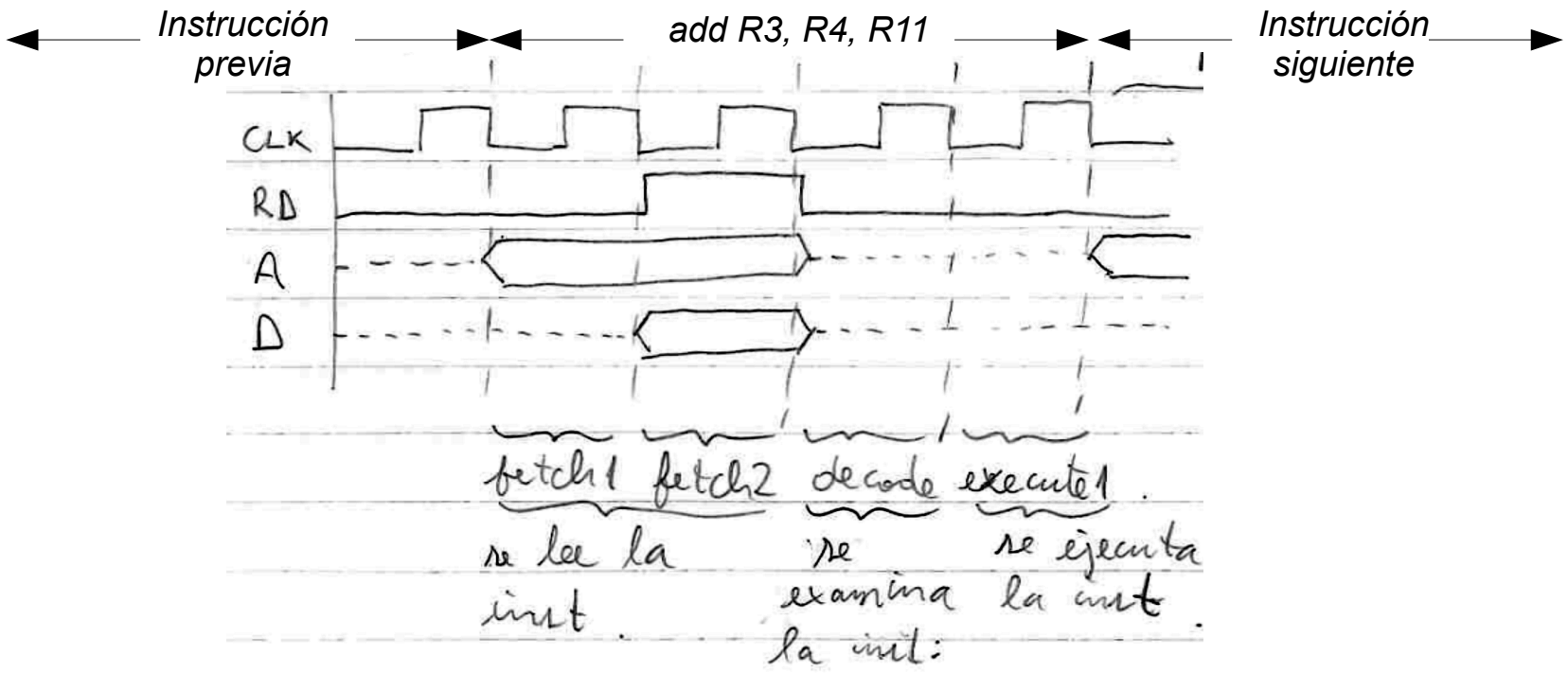
Funcionamiento

En cada ciclo del reloj la unidad de control genera las señales de control para llevar los datos de los registros a los buses y unidades de cálculo

Restricciones

- Las señales de control permanecen constantes durante todo un ciclo del reloj. Solo cambian en el pulso de bajada del reloj.
- En un bus se puede colocar un y solo un dato en un ciclo del reloj
- La actualización del PC, SR y Rd ocurre solo en el pulso de bajada del reloj.
- Cada unidad de cálculo puede realizar un y solo un cálculo en cada ciclo del reloj.

Etapas en la ejecución de la instrucción: add R3, R4, R11



Generación de las señales de control

Para especificar la manera en que se ejecutan las instrucciones ciclo a ciclo se usan 2 niveles de abstracción: transferencias entre registros y señales de control

Ciclo	***Transferencias entre registros***	***Señales de control***
$fetch_1$:	AR ← PC goto $fetch_2$	OP-Y-SEL ← @0 OP-ALU ← @OR WR-AR EN-A OP-ABI ← @W
$fetch_2$:	IR ← Mem^w[AR] if WAIT goto $fetch_2$ else decode	OP-DBI ← @LDW OP-ABI ← @W SEL-D WR-IR EN-A RD
decode:	PC ← PC+4 goto $execute_1$	OP-Y-SEL ← @4 OP-ALU ← @ADD WR-PC
$execute_1$: [op≡add]	R11 ← R3 + R4 goto $fetch_1$	OP-Y-SEL ← @INST SEL-REG OP-ALU ← @ADD WR-Rd WR-SR

Importante

- En un mismo ciclo el orden en que se indican las transferencias o señales de control es irrelevante porque ocurren al mismo tiempo.
- Para reducir el tamaño de la especificación, las señales de control no especificadas permanecen en 0.
- Las transferencias entre registros están restringidas por lo que permiten hacer las componentes, buses y señales de control.

Generación de las señales de control

Si la instrucción es STB R3, [R5-101]

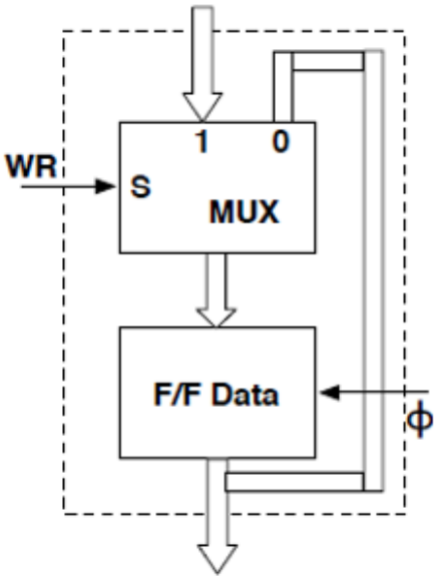
Ciclo	***Transferencias entre registros***	***Señales de control***
$execute_1$: [op≡stb]	AR ← R5-101 goto $execute_2$	SEL-REG WR-AR EN-A OP-Y-SEL ← @INST OP-ALU ← @ADD OP-ABI ← @W
$execute_2$: [op≡stb]	$Mem^{b[}$AR] ← $Trunc^{b}$(R3) if WAIT goto $execute_2$ else goto $fetch_1$	RD-DEST OP-ALU ← @OR EN-A OP-Y-SEL ← @DISP SEL-REG EN-D WR OP-DBI ← @STB OP-ABI ← @B

Si la instrucción es BG etiqueta

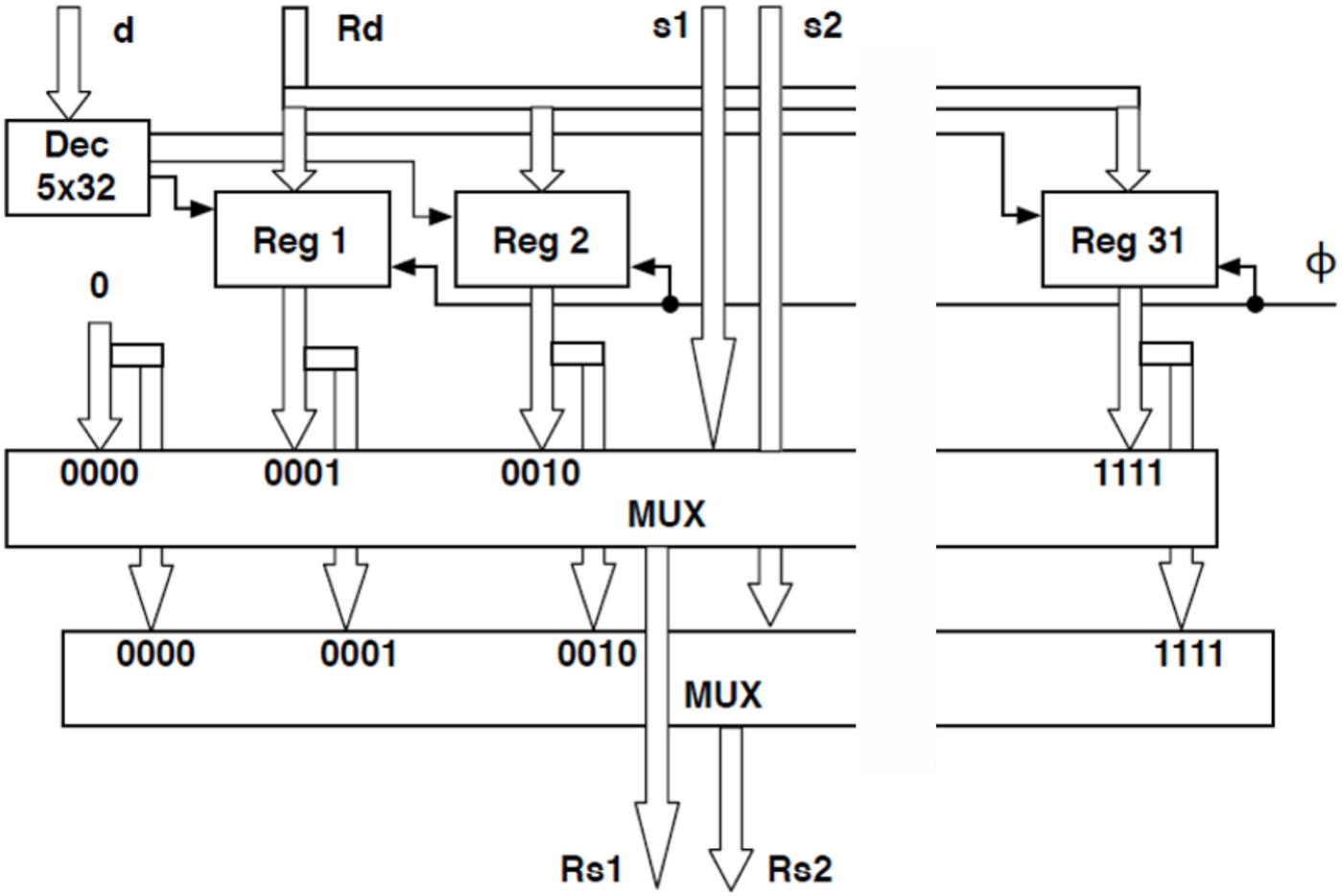
Ciclo	***Transferencias entre registros***	***Señales de control***
$execute_1$: [op≡bg]	if br? goto $execute_2$ else goto $fetch_1$	nada
$execute_2$: [op≡bg]	PC ← PC+Ext^{s}(IR[23-0]) goto $fetch_1$	WR-PC OP-Y-SEL ← @DISP OP-ALU ← @ADD

Implementación de registros

PC y SR

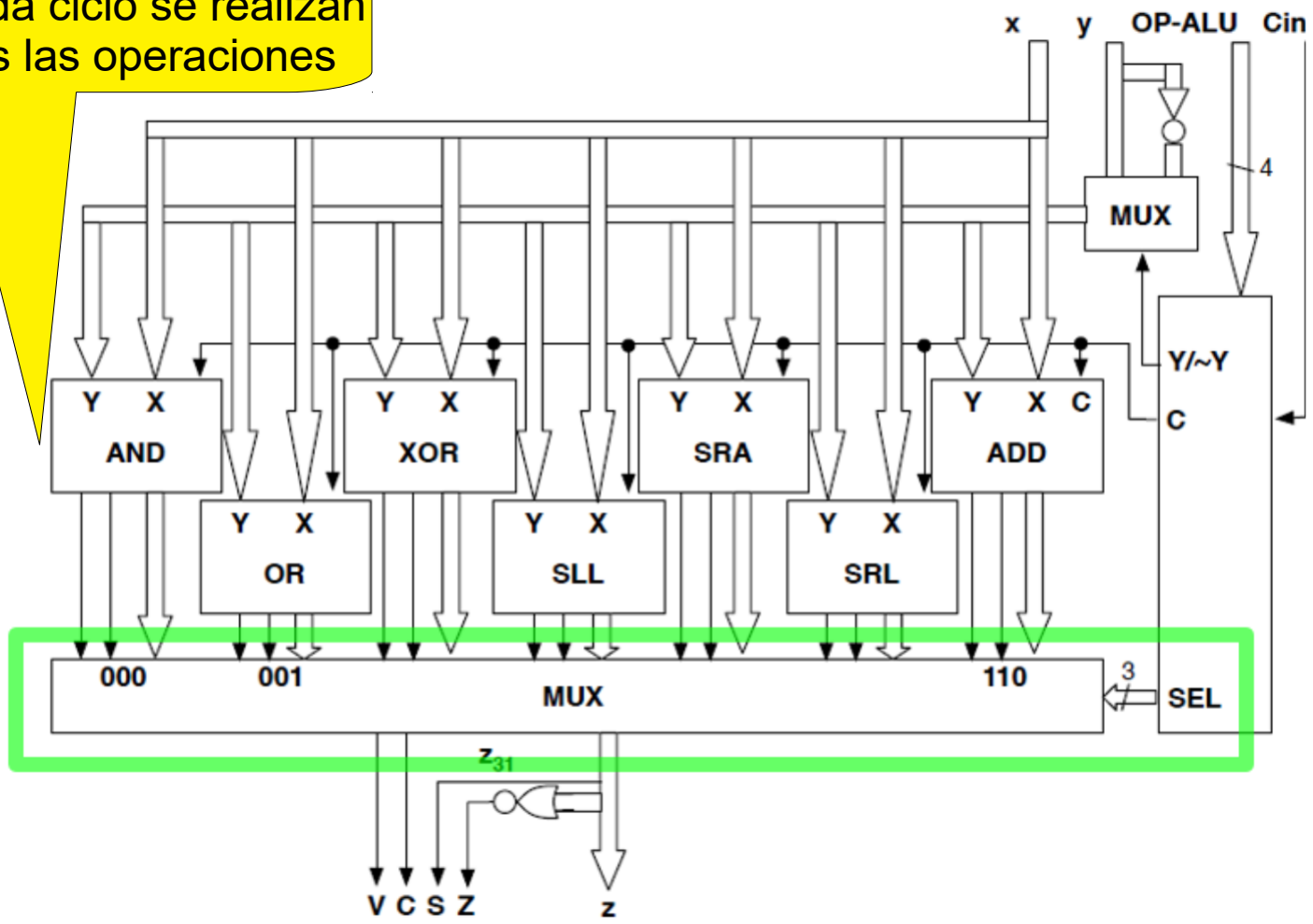


Banco de registros: R-BANK



Implementación de la ALU unidad aritmético lógica

En cada ciclo se realizan todas las operaciones



OP-ALU		C_{in}	C	SEL	Y/~Y	Calcula
ADD	0000	X	0	110	1	$x \oplus y$
ADDX	0001	0	0	110	1	$x \oplus y \oplus c$
		1	1	110	1	$x \oplus y \oplus c$
SUB	0010	X	1	110	0	$x \oplus \neg y \oplus 1$