Estructura de computadores.

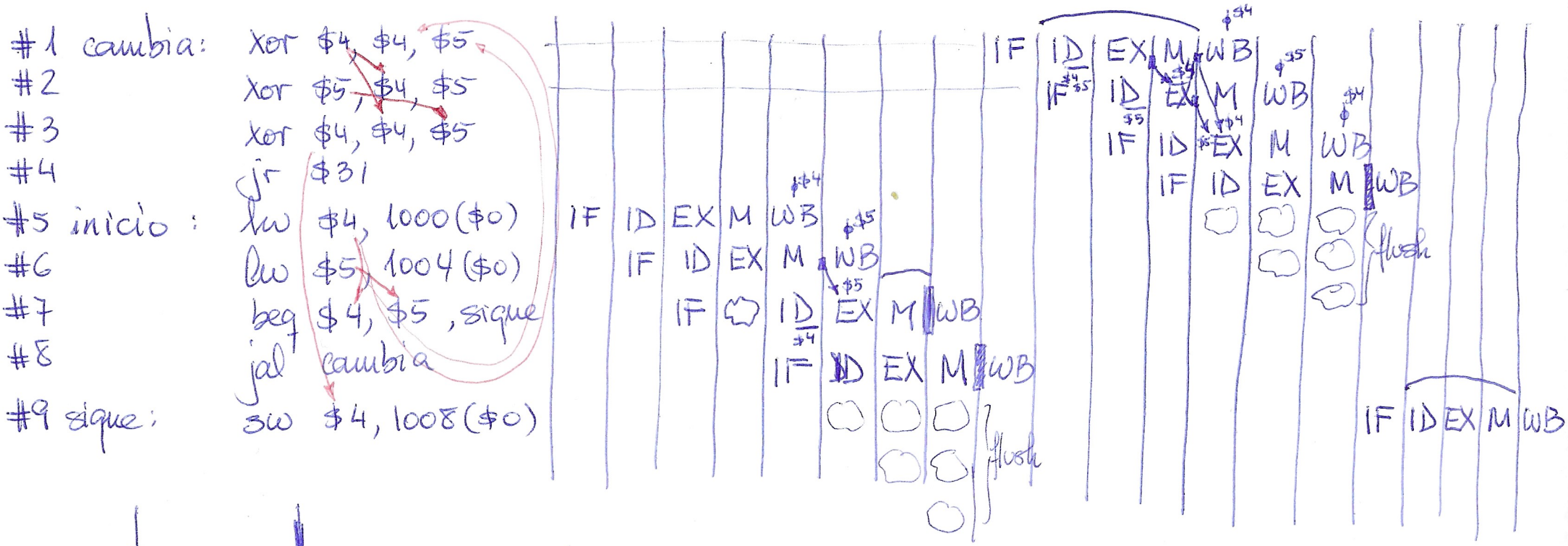
Tema 2. Mejora del rendimiento del procesador con la segmentación.

Grupo 2: #7 y #9

#7 Considerar la estructura segmentada en 5 etapas del MIPS, con hardware para la detección de riesgos por dependencia de datos y todos los caminos de anticipación (cortocircuitos) habilitados. Respecto a las dependencias de control, el controlador implementa la suposición de salto no realizado. Todos los saltos se resuelven en la etapa de memoria. El banco de registros permite lectura y escritura simultánea de un mismo registro sin conflicto. Sea el siguiente programa:

```
#1 cambia: xor  $4, $4, $5
#2         xor  $5, $4, $5
#3         xor  $4, $4, $5
#4         jr   $31
#5 inicio: lw   $4, 1000($0)
#6         lw   $5, 1004($0)
#7         beq  $4, $5, sigue
#8         jal  cambia
#9 sigue:  sw   $4, 1008($0)
```

Se sabe que las posiciones de memoria 1000, 1004 y 1008 almacenan palabras cuyo valor es 0000BEB0, 0000CAFE y 0000DED0 respectivamente (hex.). Realizar el diagrama temporal multiciclo teniendo en cuenta que el programa comienza su ejecución en la línea etiquetada con 'inicio' y termina cuando se completa la última instrucción #9.



cortocircuito

salto computado

instrucciones : $4 + 4 + 1 = 9$

ciclos : $4 + 4 + 1 + 3 + 4 + 3 + 1 = 20$

- 4: llenado cauce
- 4 (#5-8): penaliz. load-uso
- 1: penaliz. salto bdo tomado y sí salta
- 3: penaliz. salto no tomado y sí salta
- 4 (#1-4)
- 1 (#9)

$$CPI = \frac{20}{9} = 2.22$$

Soluciones a los riesgos por dependencia de datos.

- #1 - #2

  #1 xor produce $4 en EX
  #2 xor consume $4 en EX
  } cortocircuito EX-EX

- #1 - #3

  #3 xor consume $4 en EX, en ese momento, #1 guarda $4 en el registro de segmentación M|WB : cortocircuito M-EX

- #2 - #3

  #3 xor consume $5 en EX. y #2 lo produce en EX y en ese momento el valor $5 está en el registro de segmentación EX|M luego cortocircuito EX-EX

- #5 - #7

  hay burbuja por load-uso y hay anticipación en el banco de registros. El valor $4 se carga en ID de #7 de forma correcta. y no hace falta ningún cortocircuito.

- #6 - #7

  #7 consume $5 en EX y #6 lo produce en EX. Al haber una burbuja por load-uso, en el momento de EX de #7, el valor de $5 está en el registro de segmentación M|WB luego se necesita cortocircuito M-EX.

No es necesario para realizar el ejercicio:

Xor: or lógico bit a bit < iguales → 0 / distintos → 1

Xor	BEBØh	1011	1110	1011	0000
	CAFEh	1100	1010	1111	1110
Xor	744Eh	0111	0100	0100	1110
	BEBØ	1011	1110	1011	0000

traza del código:

PC		$4	$5	MEM 1008	$31	PC+4
#5	lw $4, 1000($0)	0000 BEBO	–	0000 DEDO	–	#6
#6	lw $5, 1004($0)	0000 BEBO	0000 CAFE	=	–	#7
#7	beq $4, $5, sigue	0000 BEBO	0000 CAFE	=	–	#8
#8	jal cambia	0000 BEBO	0000 CAFE	=	#9	#1
#1	xor $4, $4, $5	0000 744E	0000 CAFE	=	#9	#2
#2	xor $5, $4, $5	0000 744E	0000 BEBO	=	#9	#3
#3	xor $4, $4, $5	0000 CAFE	0000 BEBO	=	#9	#4
#4	jr $31	0000 CAFE	0000 BEBO	0000 DEDO	#9	#9
#9	sw $4, 1008($0)	0000 CAFF	0000 BEBO	0000 CAFE	#9	#10