

Pipeline (Segmentación)

EL - 4311

Estructura de
Microprocesadores

Ing. José Alberto
Díaz García

Segmentación (Pipeline)

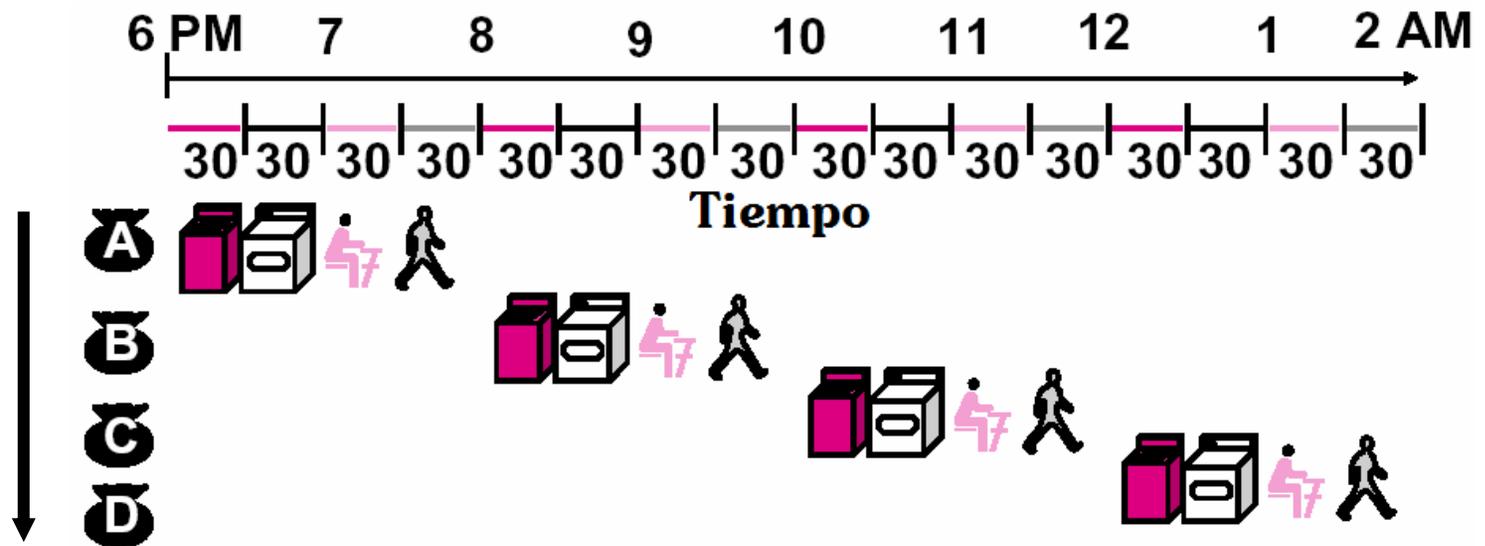
- **Es una técnica de implementación por medio de la cual se puede traslapar la ejecución de instrucciones.**
- **En la actualidad la segmentación es una de las tecnologías utilizadas para hacer procesadores más rápidos.**

Segmentación una tecnología natural

- **Ejemplo una lavandería**
- **Luis, Juan, Gabriela, Lucrecia cada uno tiene una carga de ropa para lavar, secar y doblar.**
- **Las lavadoras toman 30 minutos**
- **Las secadoras toman 30 minutos**
- **Doblarla toma 30 minutos**
- **Acomodarla toma 30 minutos**

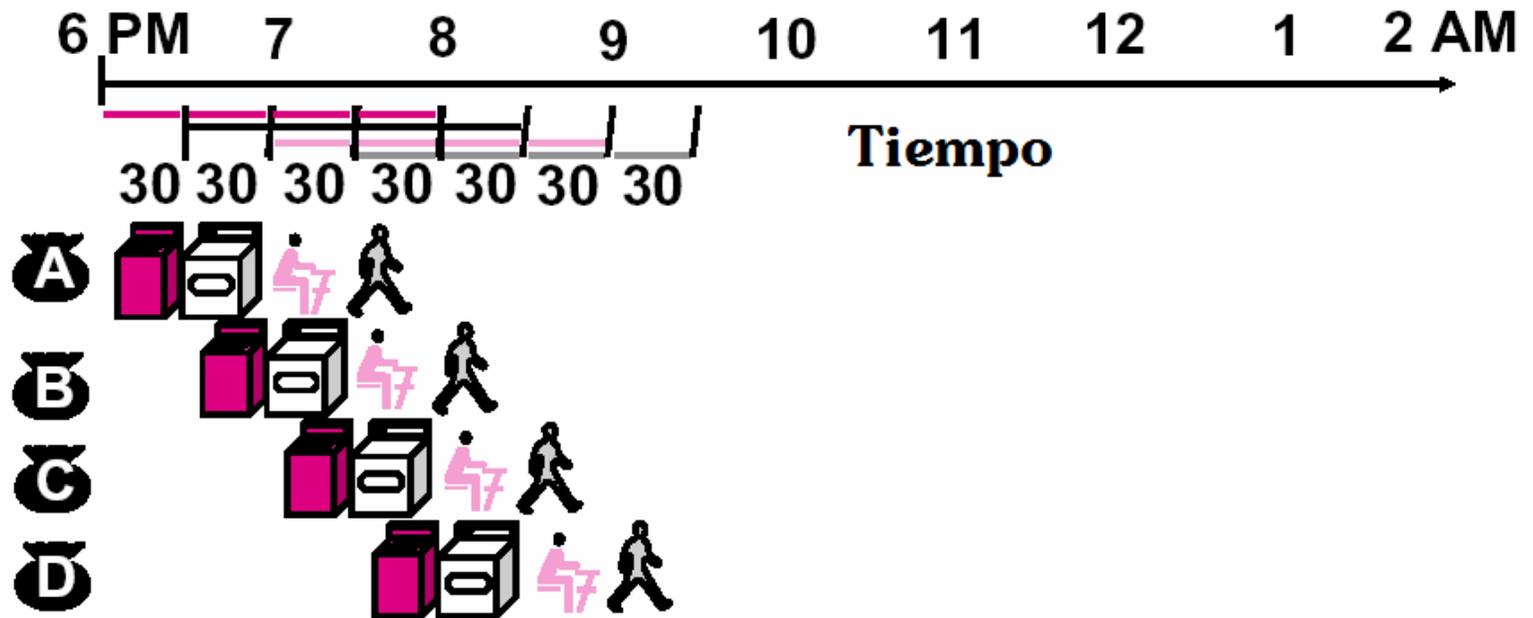


Lavandería secuencial



- La lavandería secuencial toma 8 horas en procesar cuatro cargas.
- Si lo realizamos con la tecnología segmentada ¿Cuánto tardaría?

Lavandería segmentada



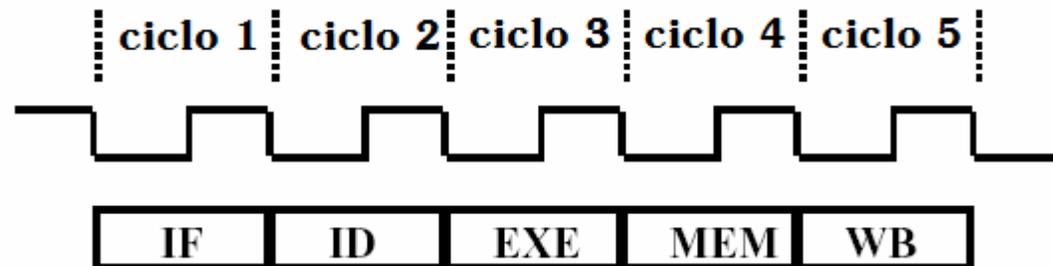
- Tardaría 3.5 horas en procesar las cuatro cargas.

Lecciones sobre segmentación

- **La segmentación no ayuda en la realización de una única tarea, ayuda en la realización de una carga de trabajo.**
- **Se pueden realizar múltiples tareas simultáneamente utilizando diferentes recursos.**
- **La velocidad se incrementa si se aumentan el número de segmentos.**
- **La razón de segmentación está dada por el segmento más lento.**
- **El desbalance en el largo de los segmentos reduce la velocidad (speedup)**
- **El tiempo en llenar y vaciar los segmentos reduce la velocidad.**
- **Almacén para las dependencias.**

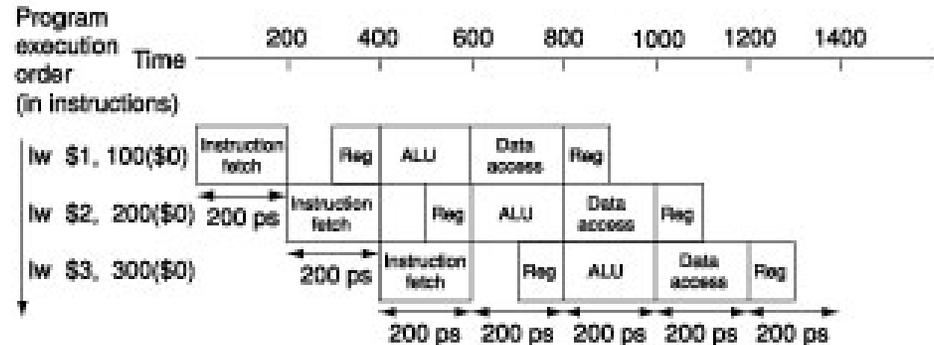
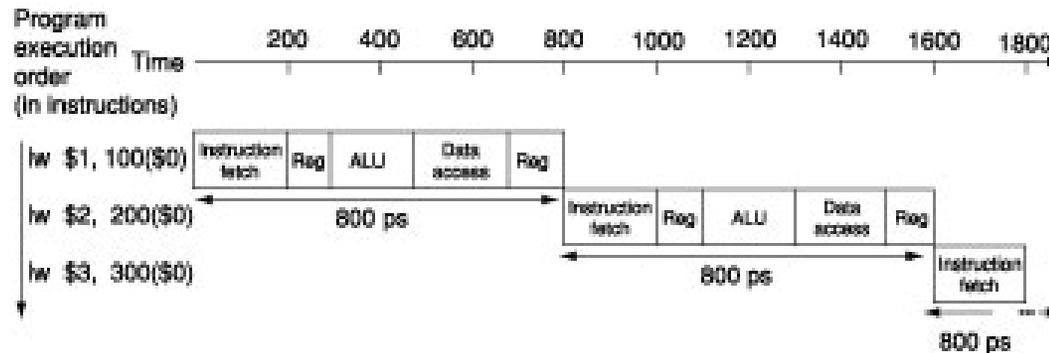
Los 5 estados de las instrucciones MIPS

- Las instrucciones MIPS presentan cinco estados (cada estado toma un ciclo de reloj):



- **IF: Búsqueda de instrucción**
 - Búsqueda de una instrucción de la memoria de instrucciones
- **ID: Búsqueda de registros y decodificación de instrucciones**
- **EXE: Ejecución o cálculo de dirección**
- **MEM: Acceso a la memoria de datos.**
- **WB: Escribir datos en el archivo de registros.**

SEGMENTACION



- La segmentación traslapa los pasos.
- Mejora el rendimiento ya que aumenta el procesamiento de instrucciones.

SEGMENTACIÓN

- **¿Qué lo hace tan fácil?**
 - Todas las instrucciones son del mismo largo
 - Pocos formatos de instrucciones
 - Los operandos con memoria solo aparecen en las instrucciones de carga y almacene
- **¿Qué lo hace difícil?**
 - **Riesgos:** Hay situaciones en la segmentación cuando la siguiente instrucción no se puede ejecutar en el próximo ciclo de reloj.
 - **Hay tres riesgos:**
 - **Riesgos estructurales**
 - **Riesgos de control**
 - **Riesgos de datos**

Riesgos estructurales

- El hardware no puede manejar la combinación de instrucciones que deben ejecutarse en el mismo ciclo. (suponiendo que se cuenta con una única memoria)
- Ejemplo: IF y MEM no pueden ejecutarse en el mismo ciclo ya que ambas necesitan acceder datos de la memoria.

ciclo	1	2	3	4	5	6	7	8
inst 1	IF	ID	EX	MEM	WB			
inst 2		IF	ID	EX	MEM	WB		
inst 3			IF	ID	EX	MEM	WB	
inst 4				IF	ID	EX	MEM	WB

↙ riesgo estructural

RIESGOS ESTRUCTURALES

- **SOLUCIÓN:** Retardar la instrucción 4 hasta el ciclo 7.

ciclo	1	2	3	4	5	6	7	8	9
inst 1	IF	ID	EX	MEM	WB				
inst 2		IF	ID	EX	MEM	WB			
inst 3			IF	ID	EX	MEM	WB		
inst 4				atasco	atasco	atasco	IF	ID	EX ..

inicio de la instrucción 4 

RIESGOS DE CONTROL

- **Los riesgos en el control ocurren porque el procesador no “sabe” lo suficiente.**
 - Cuando se producen saltos condicionales
 - Saltos incondicionales
- **SOLUCIÓN:**
 - **Atascamiento (Stall): se puede retrasar la ejecución de la unidad de salto después de que se conozca la información.**
 - **Predicción (predict): se puede predecir esta información en forma temprana.**
 - **Retardar el salto.**

Riesgos de control (utilizando Atascamiento)

- Asuma que se puede calcular la dirección de salto y actualizar el PC en el estado ID.
- Ejemplo:

ciclo	1	2	3	4	5	6	7	8
add \$4, \$5, \$6	IF	ID	EX	MEM	WB			
beq \$1, \$2, 40		IF	ID	EX	MEM	WB		
lw \$3, 300(\$0)			Stall	IF	ID	EX	MEM	WB



como no se sabe si el salto se realizará, se retarda lw hasta el estado IF (4 ciclo).
Si esto se realiza, se presenta un riesgo estructural en el ciclo 4 y en el 5

Riesgos de control

- **Solución correcta**

ciclo	1	2	3	4	5	6	7	8	9	10
add \$4, \$5, \$6	IF	ID	EX	MEM	WB					
beq \$1, \$2, 40		IF	ID	EX	MEM	WB				
lw \$3, 300(\$0)			stall	stall	stall	IF	ID	EX	MEM	WB

Riesgos de control (utilizando predicción)

- Siempre se predice que el salto no se realizará
- Si la predicción es incorrecta, entonces se realiza el atascamiento

ciclo	1	2	3	4	5	6	7	8
add \$4, \$5, \$6	IF	ID	EX	MEM	WB			
beq \$1, \$2, 40		IF	ID	EX	MEM	WB		
lw \$3, 300(\$0)			IF	ID	EX	MEM	WB	

si no salta

ciclo	1	2	3	4	5	6	7	8
add \$4, \$5, \$6	IF	ID	EX	MEM	WB			
beq \$1, \$2, 40		IF	ID	EX	MEM	WB		
			atasco	atasco	atasco	atasco	atasco	
40						IF	ID	EX ...

si salta

Riesgos de control (utilizando retardo en el salto)

- El ensamblador automáticamente reemplaza la instrucción por una instrucción que no afecte el salto.
- **NOTA: MIPS utiliza la sección de salto retardada**

add \$4, \$5, \$6
beq \$1, \$2, 40
 lw \$3, 300(\$0)

posición de salto retardada

como la instrucción "add" no afecta el salto, se puede reemplazar la posición de salto por la instrucción "add"



Reordenando los códigos

ciclo	1	2	3	4	5	6	7	8
beq \$1, \$2, 40	IF	ID	EX	MEM	WB			
add \$4, \$5, \$6		IF	ID	EX	MEM	WB		
lw \$3, 300(\$0)			IF	ID	EX	MEM	WB	

Riesgos de datos

- Los riesgos de datos ocurren porque una instrucción depende del resultado de una instrucción previa en el segmento
- **NOTA:** las instrucciones “add” o “sub” no escriben el resultado hasta el estado WB.
- **Ejemplo:**

ciclo	1	2	3	4	5	6	7	8	9
add \$s0, \$t0, \$t1	IF	ID	EX	MEM	WB				
sub \$t2, \$s0, \$t3		IF	stall	stall	stall	ID	EX	MEM	WB



necesita esperar hasta WB ya que \$s0 tendrá su resultado hasta el estado WB

- Algunas veces se puede resolver (o reducir) atascamientos para los riesgos de datos utilizando la técnica adelantamiento (forwarding)

Riesgos de datos (forwarding)

- **Forwarding:** Como el resultado de la operación “add” (\$s0) se obtiene despues del estado EX, se puede adelantar el resultado (\$s0) al próximo EX para realizar la operación “sub”.

ciclo	1	2	3	4	5	6
add \$s0, \$t0, \$t1	IF	ID	EX	MEM	WB	
sub \$t2, \$s0, \$t3		IF	ID	EX	MEM	WB



adelantar \$s0 a la próxima instrucción

Riesgos de datos (forwarding)

- **Forwarding: trabaja muy bien. Sin embargo no prevé atascamientos en los segmentos.**
- **Ejemplo: como la instrucción “lw” no carga el dato deseado en la memoria hasta el estado MEM, no se puede adelantar el resultado (\$s0) hasta el estado MEM.**

ciclo		1	2	3	4	5	6	7
lw	\$s0 20(\$t1)	IF	ID	EX	MEM	WB		
sub	\$t2, \$s0, \$t3		IF	ID	stall	EX	MEM	WB



adelantando \$s0 a la próxima instrucción

Ruta de datos segmentada

- **En esta sección se mostrará la ruta de datos de un solo ciclo para modificarla en una ruta de datos segmentada**

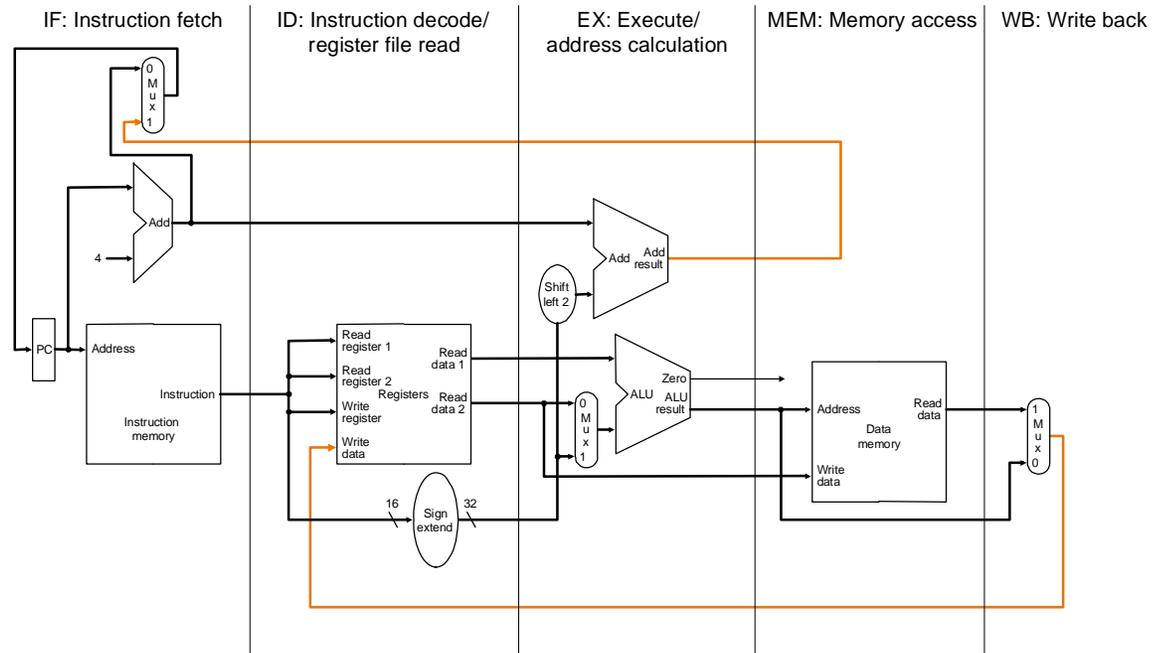
EL - 4311

Estructura de
Microprocesadores

Ing. José Alberto
Díaz García

Idea básica de la ruta de datos

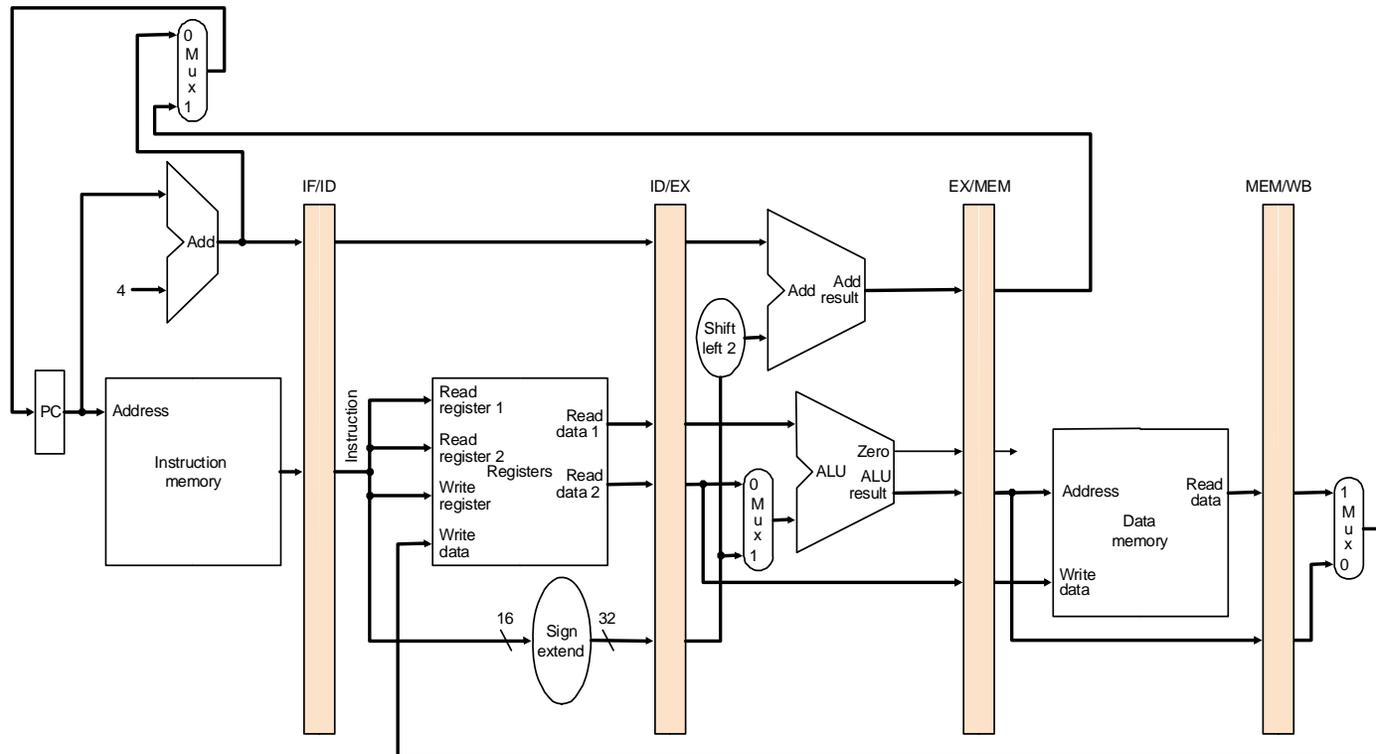
- Se puede dividir la ruta de datos de un solo ciclo en los siguientes cinco segmentos



- ¿Qué necesitamos para realizar esta separación?
- **Registros**

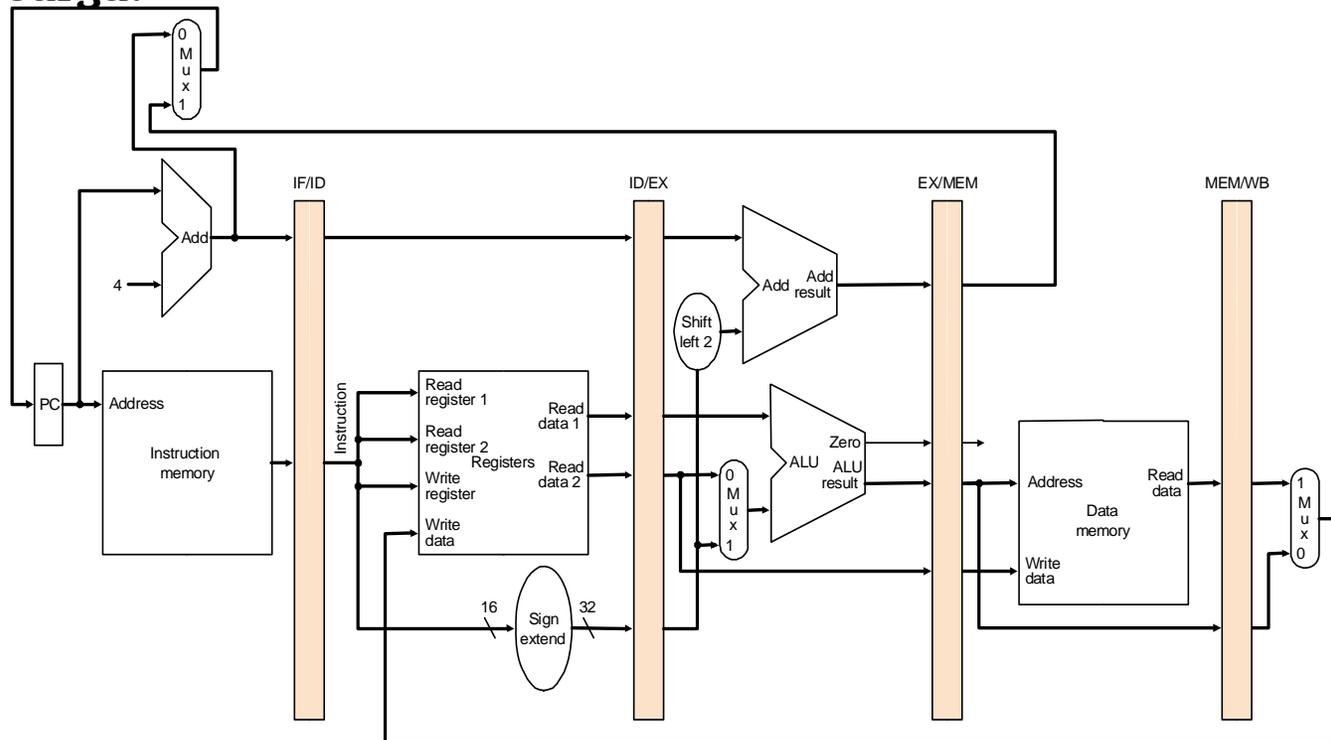
Ruta de datos segmentada

- Con el fin de almacenar información entre los estados, se necesitan de los registros (Nota: no son necesarios los registros entre WB e IF, ya que todas las instrucciones deben actualizar algunos datos en la memoria, registros o en el PC)

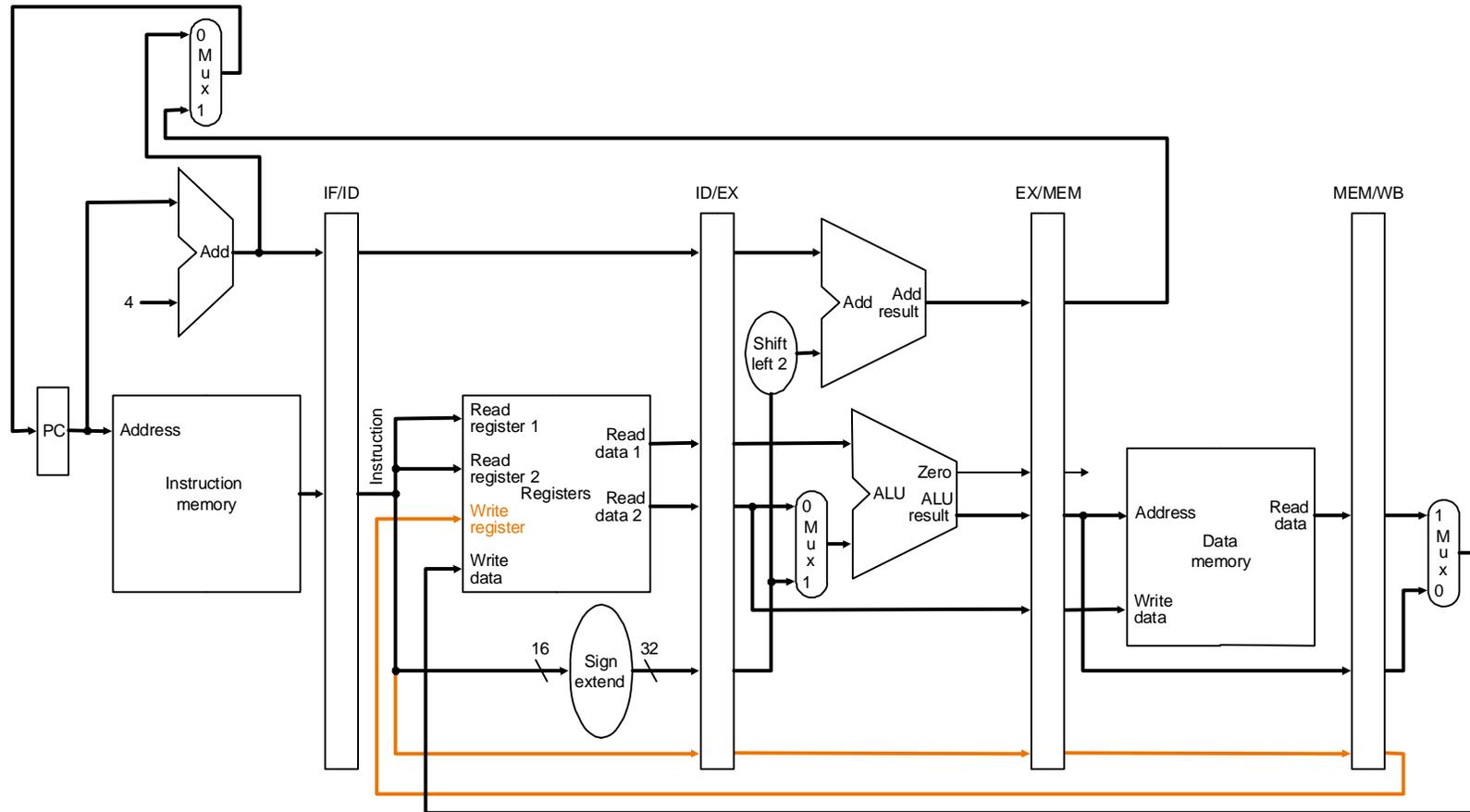


Hay un problema ¿Dónde?

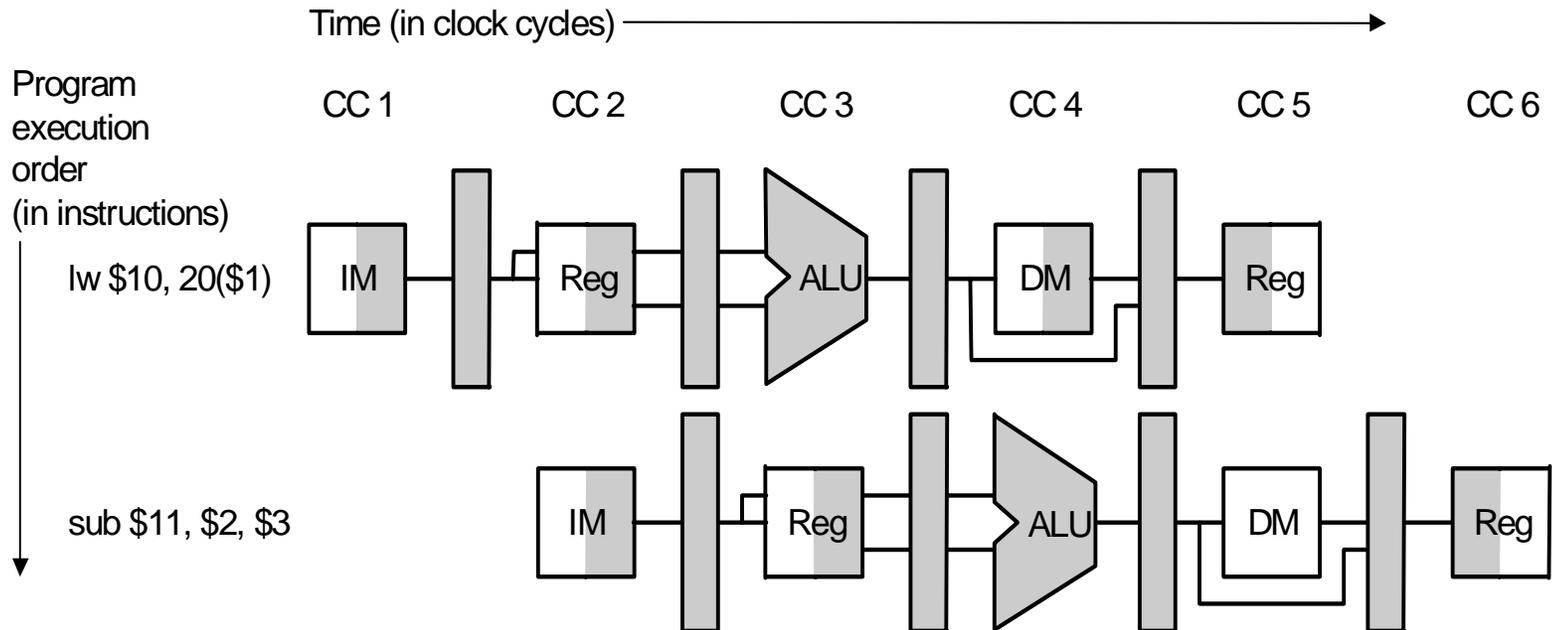
- El registro de escritura cambia después del estado final en la instrucción “lw”.
- Por lo tanto se necesita cambiar esta ruta de datos para conservar el número del registro destino en la instrucción de carga.



Ruta de datos corregida



Representación gráfica de los segmentos



- **Puede ayudar para responder preguntas como:**
 - **¿Cuántos ciclos se necesitan para ejecutar este programa?**
 - **¿Qué hace la ALU durante el ciclo 4?**
 - **Utilice esta representación para ayudar a entender la ruta de los datos.**