

Riesgos en la segmentación

Riesgo: Situación que previene que la siguiente instrucción pueda ser ejecutada en el ciclo de reloj correspondiente.

Riesgos estructurales: conflictos de hardware entre instrucciones.

Riesgos de datos: Causado por dependencias **reales** entre datos de instrucciones

Riesgos de control: Saltos y branches.

Los riesgos reducen el desempeño ideal ganado por la técnica de pipeline.

Stalls

Los riesgos provocan que el pipeline se *detenga* (**stall**)

- Las instrucciones calendarizadas antes de la instrucción detenida deben terminar su ejecución.
- Las instrucciones calendarizadas después de la instrucción detenida deben ser detenidas igualmente.

Se debe tomar en cuenta el tiempo detención por instrucción:

$$CPI_{\text{pipelined}} = \text{IdealCPI} + \frac{\text{Pipeline stall clock cycles per instruction}}{\text{instruction}}$$

Si se toma IdealCPI = 1:

$$\text{Speedup} = \frac{CPI_{\text{unpipelined}}}{1 + \text{Pipeline stall clock cycles per instruction}}$$

Riesgos

Definición

Unidades Funcionales
- FU

Riesgos estructurales

Riesgos de Datos

Adelantamiento

Referencias

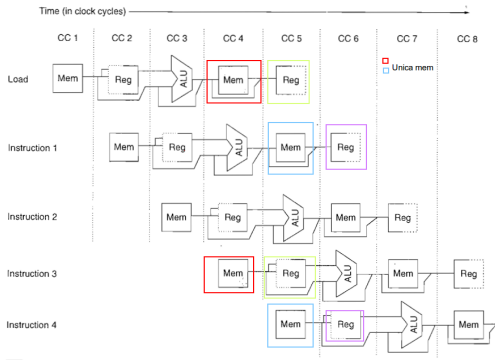
Unidades Funcionales - FU

Una unidad funcional es un elemento, dentro del hardware de un procesador, que realiza una función específica: Ejemplos:

Riesgos Estructurales

ALU Multiplicadores Contadores
fpALU Comparadores Entre otros.

En un procesador con pipeline se requieren unidades funcionales duplicadas para alojar recursos en todas las posibles combinaciones de instrucciones.



Cuando no hay recursos necesarios para evitar conflictos en uso de hardware se tiene un **riesgo estructural**

Posibles Soluciones

Uso de stalls.

Solución simple

Disminuye el rendimiento -> 1 ciclo más

Duplicar hardware

Solución más compleja. Puede requerir lógica de control adicional

Puede ser cara (más hardware, más potencia)

No disminuye el desempeño

Riesgos de datos

Los **riesgos de datos** ocurren cuando en el pipeline se cambia el orden de acceso a lectura/escritura de operandos, de forma que el orden difiere de la ejecución secuencial en un procesador sin pipeline.

DADD	R1, R2, R3
DSUB	R4, R1, R5
AND	R6, R1, R7
OR	R8, R1, R9
XOR	R10, R1, R11

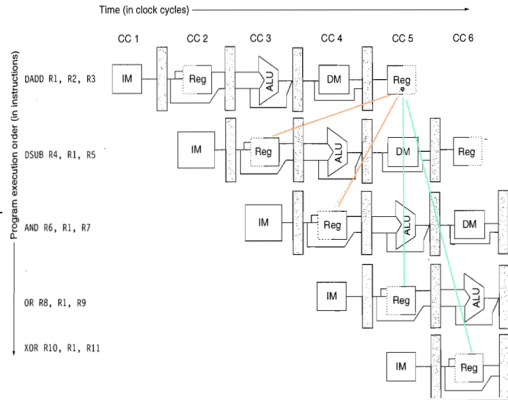
Posibles Soluciones

Uso de stalls.

Solución simple

Disminuye el rendimiento -> requiere más de 1 ciclo adicional

Adelantamiento: Consiste en mover el resultado de un registro directamente hacia la siguiente etapa donde se necesita, si esperar al WB.



Riesgos de control

Los **riesgos de control** ocurren cuando al tener la ejecución de una instrucción *branch* se puede modificar o no el valor del PC, alterando el flujo de ejecución del programa.


- Dependiendo de si el salto se toma o no (etapa ID), la siguiente instrucción será o no la correcta.

Posibles Soluciones

Stall de un ciclo.

- Después de cada salto se realiza dos IF.

Predicción de salto: Estrategia basada en métodos estadísticos o probabilistas para tratar de predecir el si salto es tomado o no. En caso de fallar la predicción, se debe vaciar (flush) el pipeline.

Stall							
Branch instruction	IF	ID 	EX	MEM	WB		
Branch successor		IF	IF	ID	EX	MEM	WB
Branch successor + 1				IF	ID	EX	MEM
Branch successor + 2					IF	ID	EX