Arquitectura de Computadores

SEGMENTACIÓN DE CAUCE
Distintas latencias de operadores

Manejo de CF

- En Ops con enteros
  - Todas las operaciones tomaban un ciclo
  - Orden emisión = órden terminación de instrucciones
  - Al haber paradas, el cauce no avanza, UFs quedan ociosas.
- Con CF, las operaciones toman más de 1 ciclo
  - Ejemplo: ADDD y SUBD: 5c., MULTD 10c. DIVD 20c. ...
  - Operadores en EX multiciclo o segmentados (varias fases monociclo)
  - (No es opción aumentar todas las etapas monociclo ralentizando reloj)

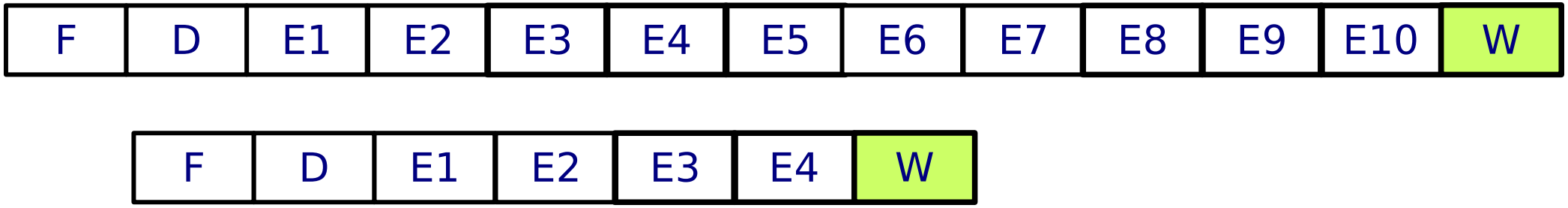
Manejo de CF

- El solapamiento de instrucciones con patrones de ejecución introduce nuevps:
  - Riesgos de datos (WAW)
  - Riesgos Estructurales
  - Interrupciones Imprecisas
- Porque el instante en que se escribe en los registros (WB) deja de ser fijo
- Porque, aunque la emisión sea ordenada, la terminación es fuera de orden (llegada a fase WB)

Tipos de riesgos de datos

- Read after write (RAW)
  - causados por una dependencia verdadera
  - No se pueden evitar
- Write after write (WAW)
  - uso del mismo recurso de escritura por dos instrucciones por demás independientes
  - se le llama dependencia de salida
- Write after read (WAR)
  - Una instrucción escribe en un recurso que una instrucción previa a ésta lee
  - Se llama antidependencia
- WAW y WAR son dependencias de nombrado y pueden evitarse usando ciertas técnicas específicas

Riesgos de datos WAW

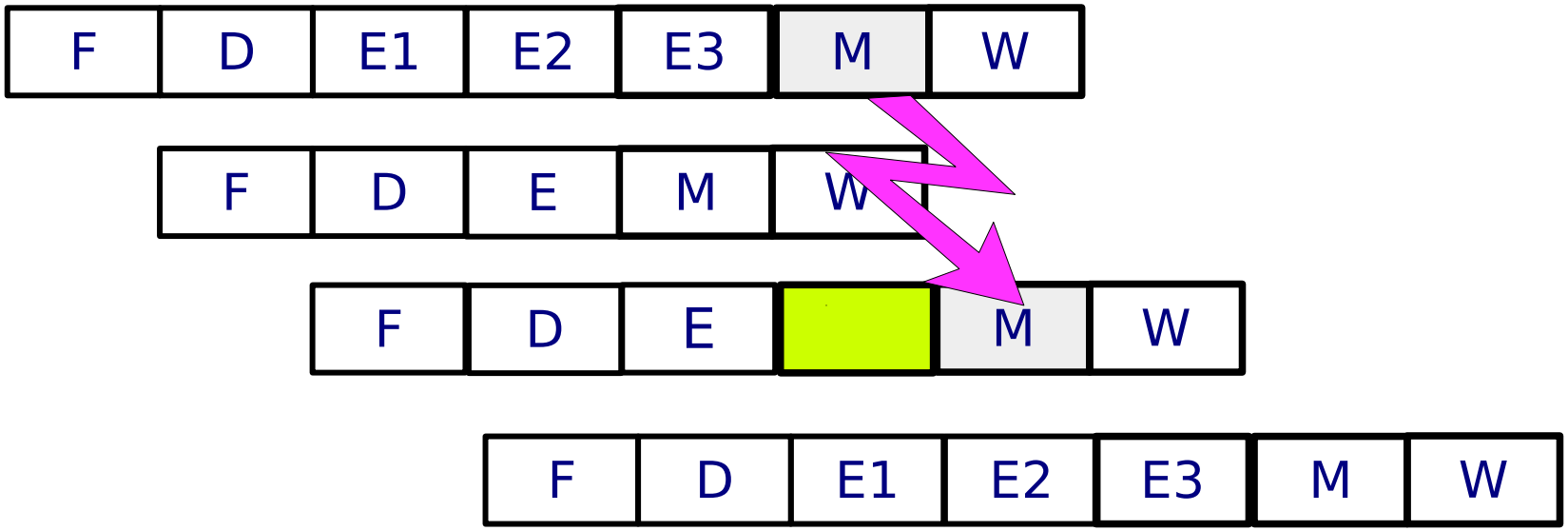


DIVD F0,F2,F4
ADDD F0,F10,F8

10 ciclos div
4 ciclos sumas/restas

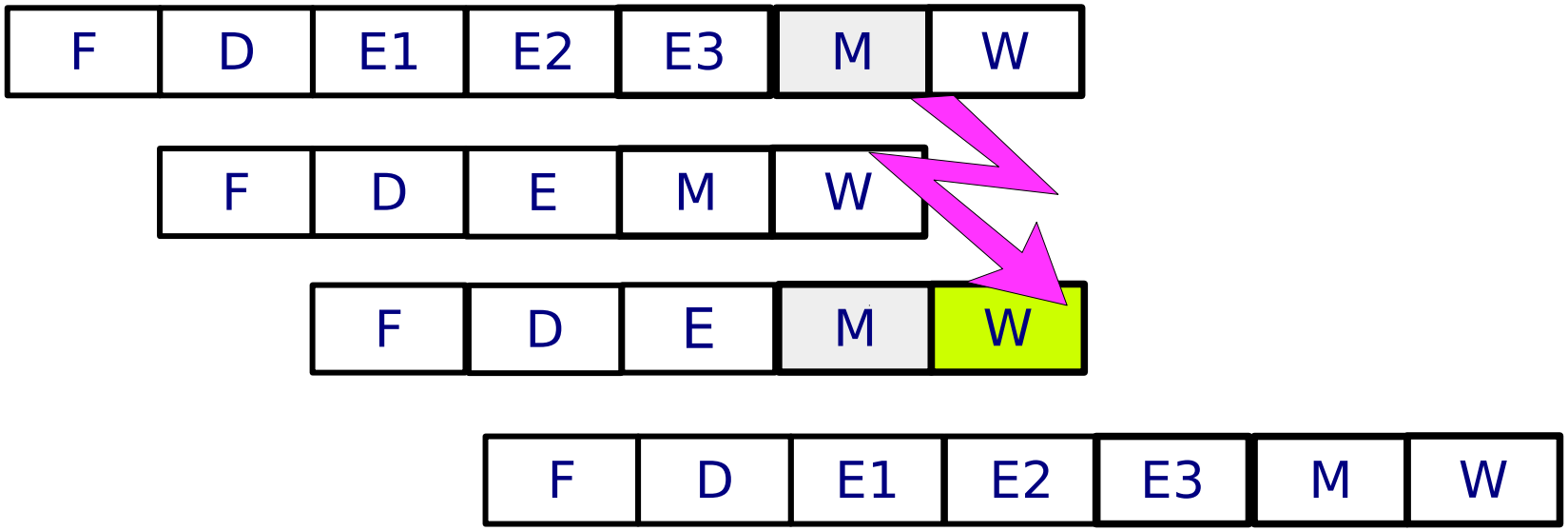
Obviamos la etapa ME para este ejemplo

Posibles riesgos estructurales



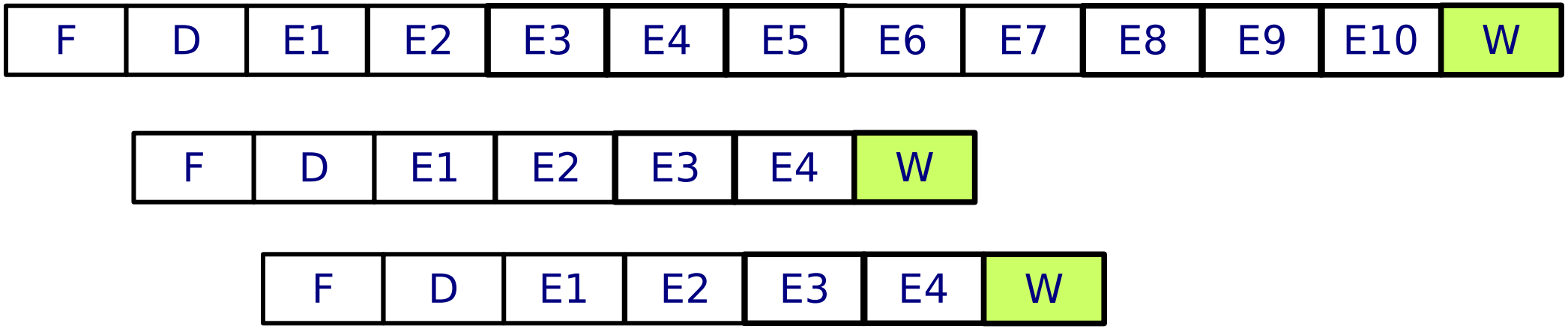
Con frecuencia se omite M para ops proceso en CF (cauces no homogéneos)

Riesgos estructurales



W: el banco de registros de enteros != banco de registros de CF
M:si se trata de instrucciones que no acceden a Memoria (proceso ambas) o proceso CF y LW/SW no hay riesgo

Interrupciones imprecisas




```
DIVD   F0,F2,F4
ADDD  F10,F10,F8
SUBD  F12,F12,F14
```

ADDD y SUBD terminan antes que DIVD
Si DIVD produce una excepción aritmética después de completarse ADDD y/o SUBD, se habrá destruido el valor de F10 y F12 y no se puede restaurar el estado de la máquina anterior a DIVD

Obviamos la etapa ME para este ejemplo

Opciones de solución

- WAW: detener emisión detectando el riesgo (Lab: DLX)
- Interrupciones imprecisas
  - Forzar la terminación (WB) en orden manteniendo en cola los resultados
    - Colas pueden ser largas y costosas
  - Permitir interrupción imprecisa
    - Las UF de CF deben determinar en los 3 primeros ciclos de la segmentación si hay riesgo de que se produzca una excepción al entrar en EX
    - Impedir que se completen las siguientes (introduciendo paradas)

Aumento de rendimiento

- Solventar los riesgos de segmentación
- Técnicas ESTÁTICAS
  - Optimización de compiladores
  - Es software
  - Antes de la ejecución del programa
- Técnicas DINÁMICAS
  - Implementadas en el Hardware
  - Durante la ejecución del programa en el cauce
  - Con ayuda de extra hardware
    - Cierto aumento de coste, consumo, complejidad
- No son excluyentes

Arquitectura de Computadores

SEGMENTACIÓN DE CAUCE
PLANIFICACIÓN ESTÁTICA

Planificación estática

- El Software (compilador) es responsable de la planificación de instrucciones para minimizar detenciones debidas a riesgos (datos, control o estructurales)
- La optimización de compiladores se desarrolló inicialmente en los años 60 y se popularizó mid 90's- mid 2000
- Se trata, entre otros, de:
  - Reordenación de instrucciones
  - Desenrollado de bucles
  - Predicción estática de Saltos
- Experimentaremos con todo ello en el LABO

Reordenación

- Distanciar instrucciones con dependencias de datos
- Hasta ahora con enteros, en cauces como DLX: RAW
- Lo extendemos al uso de CF
  - Latencias distintas en la etapa de ejecución
  - Mayor número de paradas
  - Riesgos WAW
  - Los riesgos WAR no ocurren si se leen los registros en orden en la etapa ID (emisión en orden)

Desenrollado

- Los bucles son estructuras frecuentes en los programas y conllevan un añadido por iteración:
  - actualización de contador,
  - salto condicional y
  - huecos de retardo de salto (si se pierden)--→
- Desenrollado de bucles pretende ahorrar iteraciones, evitando parte de esos ciclos añadidos
- Si un bucle realiza un cálculo n veces
  - desenrollado factor 2
    - se replica el cálculo 2 veces dentro del bucle
    - se itera n/2 veces
  - desenrollado factor 4
    - se replica 4 veces el cálculo
    - el bucle iterará solo n/4 veces

Desenrollado

- Desenrollado de factor k en bucle de n iteraciones:
  - se repite 4 veces el cálculo dentro del bucle
  - el bucle itera n/k veces
- Esto ahorra k veces los ciclos del control del bucle:
  - actualizar el contador del bucle k vs 1
  - ejecutar la instrucción de salto k vs 1
  - y los huecos de retardo de saltos k vs 1
- MÁS en el LABORATORIO