## Pipelining e Instrucciones de Brinco



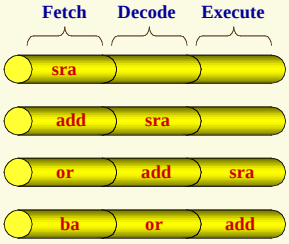
## Pipelining

- ✓ "Pipelining", en el contexto de implementación de procesadores, es la idea de procesar instrucciones por etapas como en una línea de producción.
- ✓ Idealmente las instrucciones entran a la primera etapa a una razón de una por ciclo y salen de la última etapa a una razón de una por ciclo.
- ✓ Idealmente las instrucciones pasan de una etapa a otra al cabo de un ciclo.



## Pipeline Ideal

| Loc | Inst. |
|---|---|
| 400 | sra |
| 404 | add |
| 408 | or |
| 412 | ba |





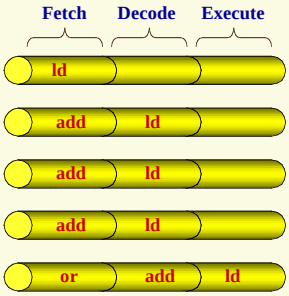

## Pipeline Interlocks

- ✓ La razón principal para que el "pipeline" no tenga una operación ideal se debe a "pipeline interlocks".
- ✓ "Pipeline interlocks" son interdependencias entre instrucciones que no permiten que las instrucciones avancen en el "pipeline" a razón de una por ciclo.
- ✓ Estos "interlocks" afectan negativamente el rendimiento del procesador.



## Interlocks por Acceso a Operandos

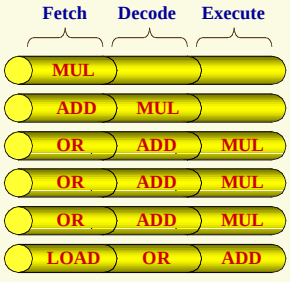
ld r5, r7, r4
add r4, r8, r2
or r2, r3, r6



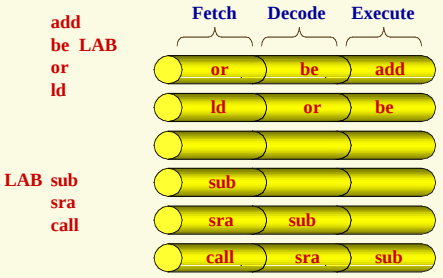



## Interlocks por Instrucciones que Requieren Varios Ciclos de Ejecución

MUL
ADD
OR
LOAD



## Interlocks por Instrucciones de Brinco que Brincan




44

## Mecanismos para Reducir el Efecto de los Interlocks

- ✓ **Superscalar**
  - *Múltiples unidades funcionales (suma, multiplicación división) que pueden operar en paralelo*
  - *Puede producir más de una instrucción por ciclo*
  - *Instrucciones que requieren varios ciclos de ejecución no detienen otras instrucciones*
- ✓ **Superpipeline**
  - *Muchas etapas de pipeline que permite reducir el periodo del reloj del sistema*

45

## Mecanismos para Reducir el Efecto de los Interlocks

- ✓ **Very Long Instruction Word (VLIW)**
  - *Varias instrucciones pequeñas independientes se acomodan en un word largo*
  - *Muchas unidades funcionales*
  - *Puede producir más de una instrucción por ciclo*
  - *Instrucciones que requieren varios ciclos de ejecución no detienen otras instrucciones*

46

## Instrucciones de Brincos

- ✓ **Se ejecutan frecuentemente**
- ✓ **Consumen un tiempo considerable del tiempo de ejecución de programas en lenguajes de alto nivel (de 12% y un 30%)**
- ✓ **El tiempo consumido por estas instrucciones se debe mayormente a interlocks**

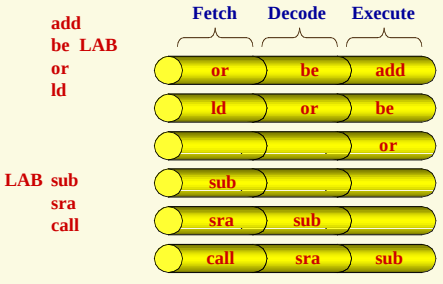
47

## Soluciones para Reducir el Tiempo Consumido por Brincos

- ✓ **Delayed branches**
- ✓ **Branch prediction**
- ✓ **Multiple Prefetching**

48

## Efecto de Delayed Branches




49

## Efectividad del Delayed Branch

- ✓ **Depende del compilador**
  - *Acomodar instrucción útil después de la instrucción de branch*
  - *Acomodar nop después de la instrucción de branch*
- ✓ **Mejora si el retraso se puede hacer de más de una instrucción**

50

## Branch Prediction

- ✓ **Al entrar la instrucción de brinco a la unidad de ejecución se se presume una de dos posibilidades:**
  - *Se va a producir el brinco*
    Luego de la instrucción de brinco se envía a la unidad de ejecución la secuencia de instrucciones a donde se brinca
  - *No se va a producir el brinco*
    Luego de la instrucción de brinco se envía a la unidad de ejecución la secuencia de instrucciones que sigue a la instrucción de brinco

51

## Branch Prediction

- ✓ **Si la predicción falla**
  - *Se cancelan las instrucciones que se enviaron después de la instrucción de brinco*
  - *Se envía la otra secuencia de instrucciones*

52

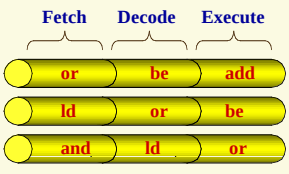
## Tipos de Predicciones

- ✓ **Predicción estática**
  - *En el formato de la instrucción se especifica la predicción (si brinco se va a dar o no)*
  - *El compilador decide la predicción*
- ✓ **Predicción dinámica**
  - *El harware hace la predicción a base de resultados previos*

53

## Predicción de No Brinco Correcta

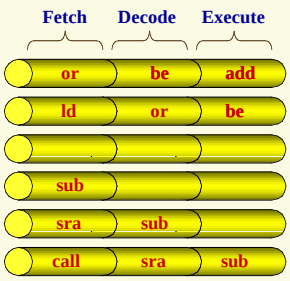
```
add
be  LAB
or
ld
and

LAB sub
    sra
    call
```




54

## Predicción de No Brinco Incorrecta

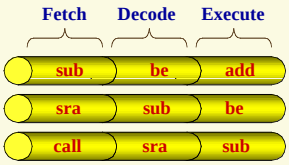
```
add
be  LAB
or
ld
and

LAB sub
    sra
    call
```




55

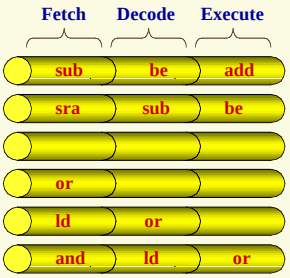
## Predicción de Brinco Correcta

```
    add
    be  LAB
    or
    ld
    and

LAB sub
    sra
    call
```




## Predicción de Brinco Incorrecta

```
    add
    be  LAB
    or
    ld
    and

LAB sub
    sra
    call
```




## Multiple Prefetching

- ✓ Se procesan ambas secuencias de instrucciones
- ✓ Una vez se determina si el brinco se da o no se cancela una de las secuencias de instrucciones
- ✓ Se puede remover la instrucción de brinco
- ✓ Requiere de una unidad de manejo de brincos

## Efecto de Multiple Prefetching

```
    sethi
    add
    be  LAB
    or
    ld
    and

LAB sub
    sra
    call
```