



Ruta de Datos y Control

LIMITACIONES Y PELIGROS EN EJECUCIÓN EN PIPELINING

Dr. Yván Jesús Túpac Valdivia
Junio 2020



Universidad Católica
San Pablo

Departamento de Ciencia
de la Computación

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

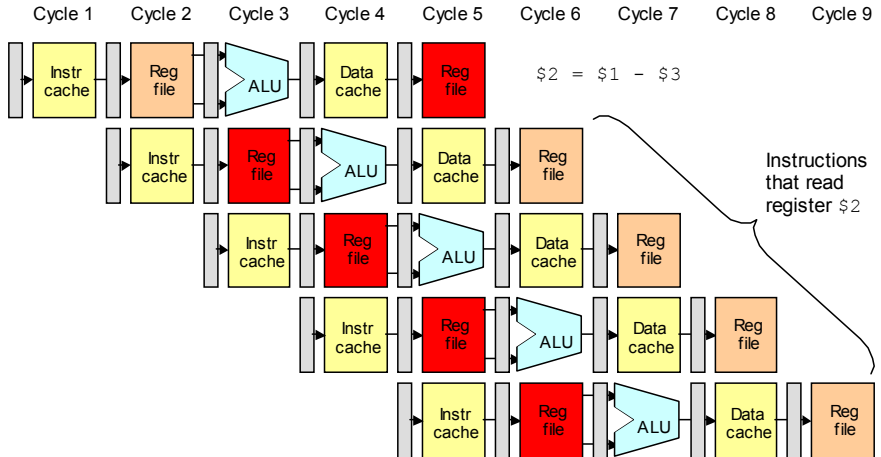
- 1 Limitaciones de rendimiento
 - Dependencias de datos
 - Tratamiento de bifurcaciones en pipeline
 - Predicción de bifurcación

- 2 Pipelining avanzado
 - Mejoras en pipelines profundos
 - Manejo de excepciones en pipelining
 - Ejecución de Instrucciones

Limitaciones de rendimiento y peligros en ejecución

Dependencias de datos y Peligros

Dependencias de datos en una *pipeline*



Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

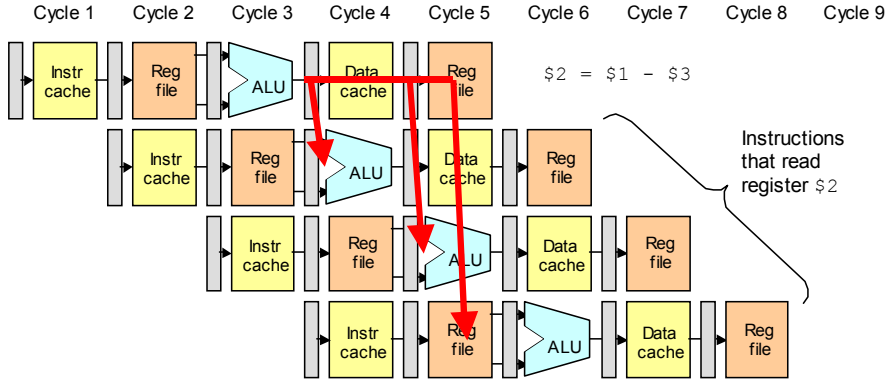
Manejo de excepciones en pipelining

Ejecución de Instrucciones

Limitaciones de rendimiento y peligros en ejecución

Solución de dependencias por *forwarding*

Cuando una instrucción previa escribe **resultados de cálculos ALU en registro**, la dependencia **se resuelve** usando *forwarding*.



Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

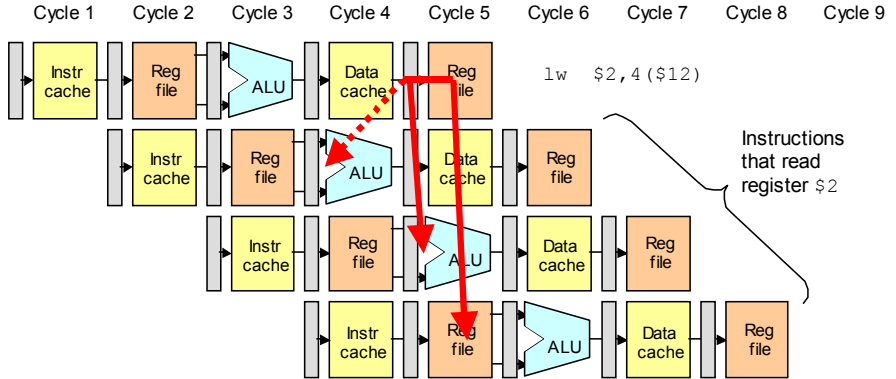
Manejo de excepciones en pipelining

Ejecución de Instrucciones

Limitaciones de rendimiento y peligros en ejecución

Dependencias que obligan a colocar burbuja

Cuando una instrucción previa escribe un **valor leído de memoria** en registro, la dependencia **no puede resolverse** por *forwarding*



Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

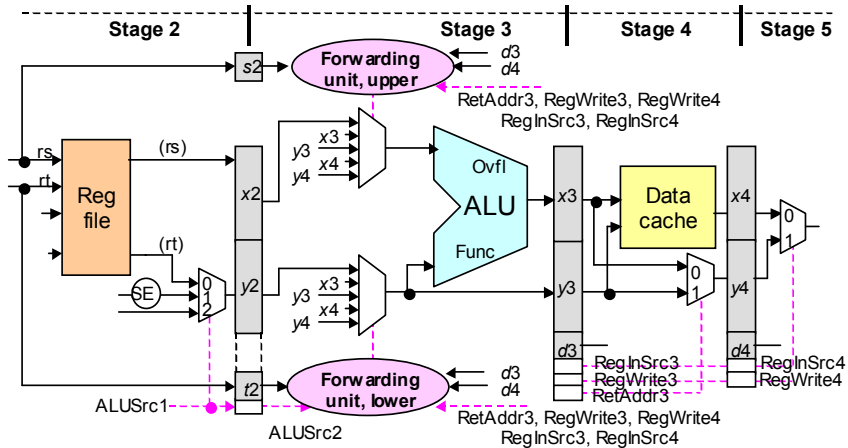
Manejo de excepciones en pipelining

Ejecución de Instrucciones

Data forwarding

Unidad de forwarding

Hardware para forwarding en una ruta de datos pipeline MiniMIPS



Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

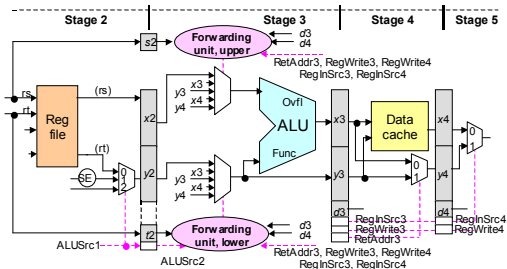
Manejo de excepciones en pipelining

Ejecución de Instrucciones

Data forwarding

Diseño de la unidad de forwarding

Veamos el diseño de “*Forwarding unit upper*”



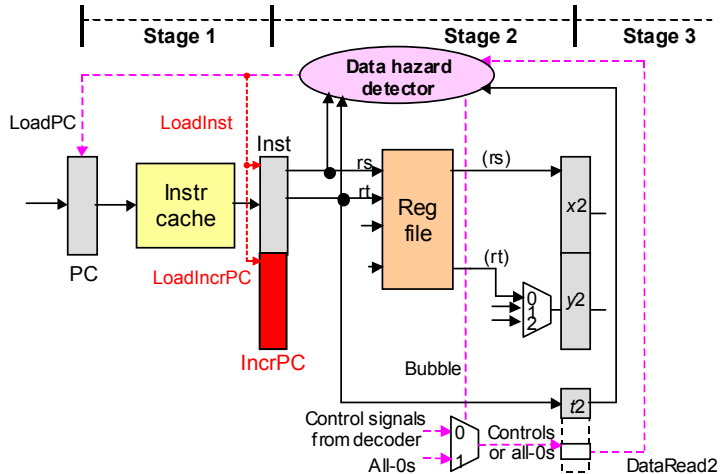
RegWrite3	RegWrite4	s2matchd3	s2matchd4	RetAddr3	RegInSrc3	RegInSrc4	Choose
0	0	x	x	x	x	x	x2
0	1	x	0	x	x	x	x2
0	1	x	1	x	x	0	x4
0	1	x	1	x	x	1	y4
1	0	1	x	0	1	x	x3
1	0	1	x	1	1	x	y3
1	1	1	1	0	1	x	x3



Data forwarding

Hardware para inserción de burbujas

Detector de peligros para una ruta de datos *pipeline* MiniMIPS



Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

Data forwarding

Completación de la ruta de datos y control

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

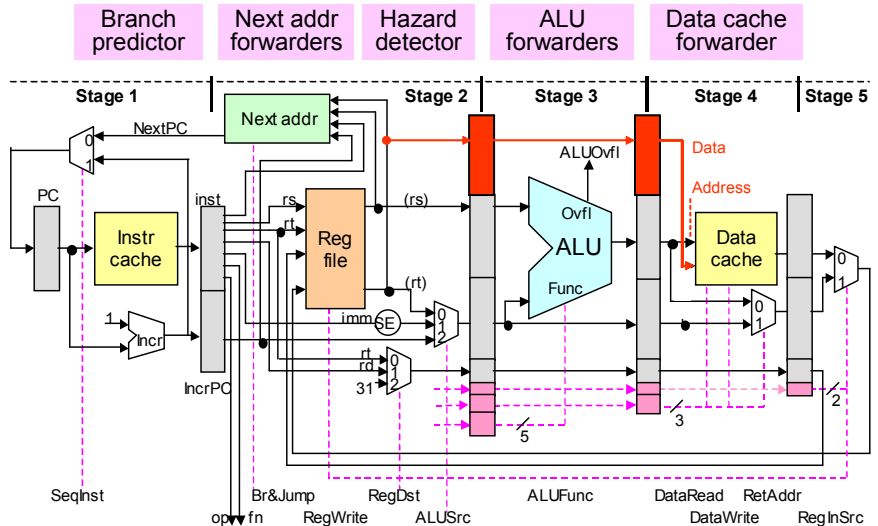
Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones



Limitaciones de rendimiento

Peligros por bifurcación

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

Soluciones por Software:

- El compilador inserta instrucciones **no-op** (burbuja) después de cada bifurcación (simple pero costoso)
- **Redefinir la bifurcación** para que su efecto se vea en **$PC + 8$**
- **Usar reordenamiento** para completar los espacios en el retardo de la bifurcación **con instrucciones útiles**

Soluciones por Hardware:

- Uso de un mecanismo similar al de **detección de peligros** usado para mandar vaciar la pipeline
- **Apostar a que nunca se cumple** una bifurcación y en caso de error, mandar **vaciar la pipeline** (rudimentario pero bastante efectivo)
- Implementar **estrategias** de predicción de bifurcación **más elaboradas**



Universidad Católica
San Pablo

Predicción de bifurcación

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

Cómo predecir si se va a dar una bifurcación:

- Una primera opción es asumir que la condición de bifurcación **nunca se cumplirá**
- Observar el contexto del programa para decidir:
 - Una bifurcación “hacia atrás”, tiene **más probabilidad de cumplirse**
 - Una bifurcación “hacia adelante”, **es menos probable de cumplirse**
 - Una bifurcación que proviene de un bloque `if-then-else` es más difícil de predecir
- Permitir que el programador o compilador **dejen señuelos**
- Decidir de acuerdo a un **histórico** almacenado en una tabla (métodos de predicción dinámica)
- **Combinar estos factores:** procesadores más modernos usan técnicas más elaboradas por sus **pipelines son profundos**, cuya penalización por error es alta

Predicción de bifurcación

Bifurcaciones “hacia adelante” y “hacia atrás” – Ejemplo

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

Buscar el máximo en la lista A en memoria iniciando en \$s1, con longitud en \$s2, colocando resultado en \$t0

Solución

Barrer la lista guardando el máximo hallado hasta el momento en \$t0.

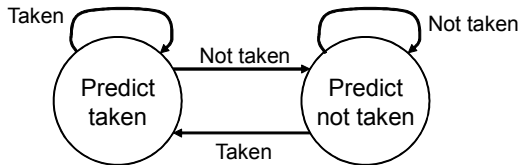
```

                                lw      $t0,0($s1)      # initialize maximum to A[0]
                                addi    $t1,$zero,0     # initialize index i to 0
loop:                          add     $t1,$t1,1       # increment index i by 1
                                beq     $t1,$s2,done    # if all elements examined, quit
                                add     $t2,$t1,$t1     # compute 2i in $t2
                                add     $t2,$t2,$t2     # compute 4i in $t2
                                add     $t2,$t2,$s1     # form address of A[i] in $t2
                                lw      $t3,0($t2)     # load value of A[i] into $t3
                                slt     $t4,$t0,$t3     # maximum < A[i]?
                                beq     $t4,$zero,loop  # if not, repeat with no change
                                addi    $t0,$t3,0       # if so, A[i] is the new maximum
                                j        loop           # change completed; now repeat
done:                          ...                    # continuation of the program
```

Predicción de bifurcación

Predicción simple con histórico de 1-bit

Esquema de predicción de bifurcación con 2 estados



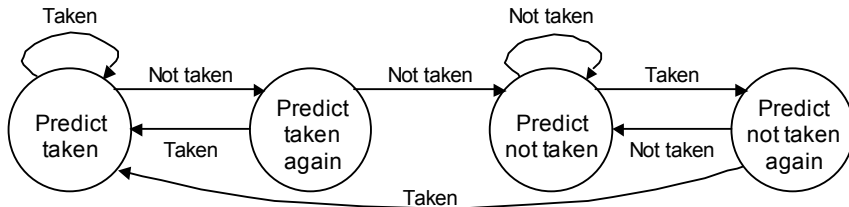
Dificultades con esta propuesta

- Cada bifurcación en un bucle tiene dos errores de predicción
 - 1 en la primera iteración (el bucle se repite, pero la historia dice que va a salir)
 - 2 en la última iteración (se acaba el bucle pero por historia parece que aún se repetirá)

Predicción de bifurcación

Predicción simple con histórico de 2-bits

Esquema de predicción de bifurcación con 4 estados



Ejemplo de aplicación

L1: 10 iter
L2: 20 iter
....
beq <c2> L2
....
bltz <c1> L1

El impacto de los esquemas de previsión de bifurcación es:

- **Siempre:** 11 previsiones erradas (94.8% de acierto)
- **1-bit:** 20 previsiones erradas (90.5% de acierto)
- **2-bit:** 11 previsiones erradas (94.8% de acierto)



Predicción de bifurcación

Otros algoritmos de predicción de bifurcación

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

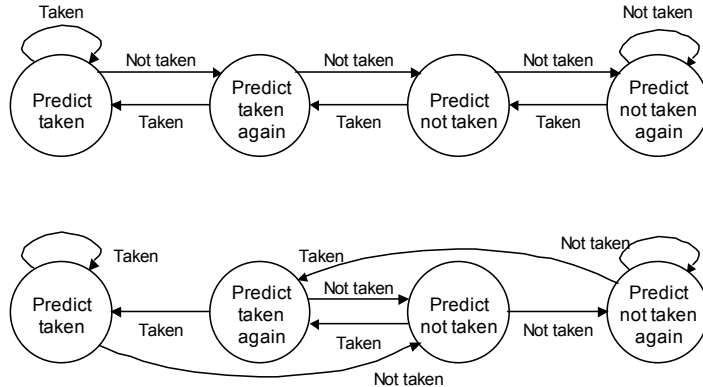
Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

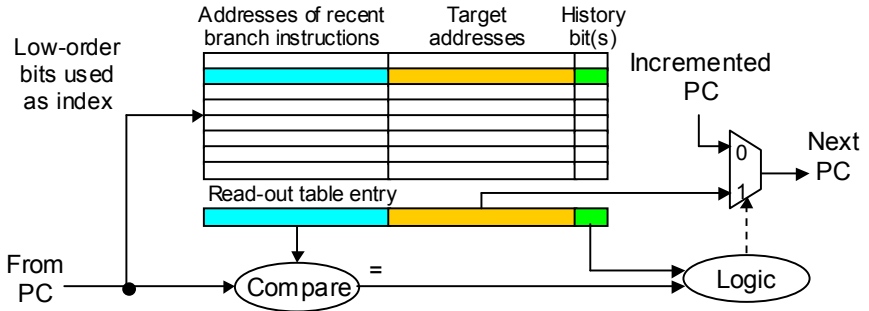
Ejecución de Instrucciones



Predicción de bifurcación

Hardware de predicción de bifurcación

Componentes de un esquema de hardware de predicción de bifurcación



Este esquema usado para actuar en el valor del contador de programa (PC) es muy similar al usado en el [mapeamiento directo](#) de direcciones de memoria cache.

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

1 Limitaciones de rendimiento

- Dependencias de datos
- Tratamiento de bifurcaciones en pipeline
- Predicción de bifurcación

2 Pipelining avanzado

- Mejoras en pipelines profundos
- Manejo de excepciones en pipelining
- Ejecución de Instrucciones

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

- La mayoría de los procesadores actuales con alto rendimiento son “*super pipelined*”, con *pipelines* muy profundas (exigen más cabeceras).
- Límite de rendimiento en *pipelines* profundas (cabeceras con tiempo de operación constante). Así, *pipelines* de 20-30 etapas están en el límite práctico.
- La penalización por mala predicción de penalización es más grande y se requieren más burbujas por las dependencias de datos y control.
- Fallos en el caché de instrucciones (a verse más adelante) son más críticos que las malas previsiones de bifurcación.

Otras alternativas

Mantener la “estructura lineal” del *pipeline* mantiene limitaciones en pipelines profundos, es necesario otro tipo de arquitecturas “no lineales”



Pipelining Avanzado

Pipeline dinámico

Canales de ejecución de instrucciones especializados con emisión en orden, posible finalización fuera de orden y retiro en orden, para disminuir la **limitación de rendimiento** al usar pipelines profundos

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

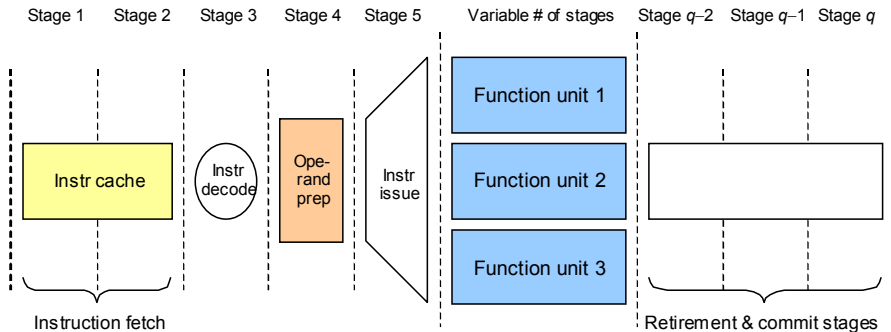
Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones



Universidad Católica
San Pablo

Pipelining Avanzado

Mejoras alcanzadas para pipelining profundo

Métodos por hardware

- Observar si una instrucción **podría quedarse atascada** (uso de ejecución fuera de orden, retiro en orden)
- Tratamiento de muchas instrucciones (requiere **más puertos de acceso** a registros)
- Eliminar **dependencias falsas** mediante renombrado de registros
- Usar **buenas predicciones de bifurcaciones**, uso de **ejecución especulativa**

Ejecución especulativa:

Consiste en ejecutar completamente instrucciones posteriores a un salto condicional, sin saber si realmente se ejecutarán. Se suele evitar el cambio en el estado de la máquina hasta validar la predicción del salto condicional.

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

Pipelining Avanzado

Mejoras alcanzadas para pipelining profundo

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

Métodos por software

- **Compilación optimizada** para ejecución en *pipeline* (uso de reordenamiento por ejemplo)
- “Desenrollar bucles” para **reducir la cantidad** de ejecución de bifurcaciones

```
loop: calcular con indice[i]
      addi i, i, 1
      b loop if not done
```

```
loop: calcular con indice[i]
      calcular con [i+1]
      addi i, i, 2
                                     b loop if not done
```



Universidad Católica
San Pablo

Pipelining Avanzado

Arquitecturas y su variación en CPI

Efecto de la **arquitectura del procesador, métodos de predicción de bifurcación y ejecución especulativa** en el CPI

Arquitectura	Métodos usados (práctica)	CPI
Sin <i>pipeline</i> , multiciclo	Emisión y ejecución en estricto orden	5-10
Sin <i>pipeline</i> , traslapada	Emisión en orden con múltiples unidades de función	3-5
<i>Pipelined</i> estática	Ejecución en orden predicción de bifurcación simple	2-3
<i>Superpipelined</i> dinámica	Ejecución fuera de orden, predicción avanzada de bifurcación	1-2
<i>Superscalar</i>	2 a 4 vías, <i>interlock</i> y especulación	0.5-1
<i>Superscalar</i> avanzada	4-8 vías, especulación agresiva	0.2-0.5

100 necesarios para 1 TIPS
100000 para un PIPS

$3.3 \text{ inst/ciclo} \times 3\text{GHz} \cong 10\text{GIPS}$



Pipelining Avanzado

Desarrollo de la línea de μ procesadores Intel©

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

A long time ago in a galaxy far,
far away....

Pipelining Avanzado

Desarrollo de la línea de μ procesadores Intel©

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

- Al inicio, hubo el 8080 que dio lugar al 80x86 (instrucciones IA32 ISA)
- De 6 a 12 o más etapas pipeline:
 - 80286
 - 80386
 - 80486
 - Pentium (8058)
- De 12 a más etapas *pipeline* con **ejecución reordenada**, uso de microinstrucciones ejecutadas fuera de orden y retiradas en orden
 - Pentium PRO
 - Pentium II
 - Pentium III
 - Celeron
- De 24 a más etapas pipeline
 - Pentium 4

Pipelining Avanzado

Desarrollo de la línea de μ procesadores ARM©

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

ARM (Advanced RISC Machines)

- ARMv1, v2, v3 con pipeline de 3 etapas: (fetch- decode-exec)
- ARMv5: ARM9 implementa:
 - Pipeline de 5 etapas (fetch-decode-execute-memoria-writeback)
 - Arquitectura Harward (Cachés de instrucciones y datos separados).
- ARMv6: ARM11 implementa:
 - Pipeline de 8 etapas para procesamiento entero
 - Pipeline separado para load/store
 - Completación fuera de orden para operaciones como store
 - Predicción de bifurcación dinámica y folding
- ARMv7: CortexA con pipeline de hasta 13 etapas y predicción avanzada de bifurcación

Manejo de excepciones

Excepciones generan los mismos riesgos que las bifurcaciones

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones

Para un *pipelining*, las excepciones se muestran como las bifurcaciones:

- ¿Cómo manejar las excepciones que están por entrar al *pipeline*? (dejarlas correr hasta completarse y retirar sus resultados)
- Qué hacer con las instrucciones en el *pipeline* posteriores a la excepción? (desecharlas tal que no afecten el estado)

Excepciones precisas y excepciones imprecisas

- **Excepciones precisas:** efecto de eliminar el efecto de *pipepining* y paralelismo al forzar comportamiento como en una ejecución secuencial pura con resultado consistente. Es preferible esta situación si hay facilidad en el manejo de la excepción
- **Excepciones imprecisas:** resultan más caóticas, pero pueden ser tratadas por un hardware más rápido (considerando que el manejador pueda limpiarlas y ofrecer excepciones más precisas.



Universidad Católica
San Pablo

Arquitecturas para MicroMIPS

Comparación

Limitaciones de rendimiento

Dependencias de datos

Tratamiento de bifurcaciones en pipeline

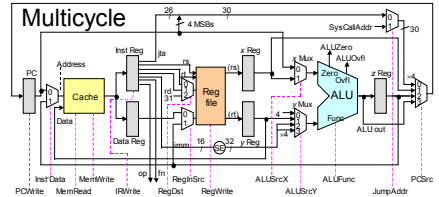
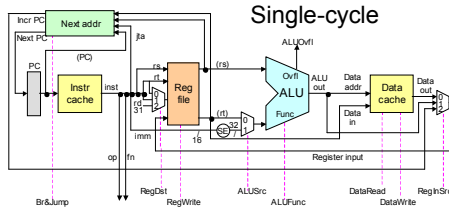
Predicción de bifurcación

Pipelining avanzado

Mejoras en pipelines profundos

Manejo de excepciones en pipelining

Ejecución de Instrucciones



125 MHz
CPI = 1

500 MHz
CPI \approx 4

500 MHz
CPI \approx 1.1

