

# Práctica 2 : Pipeline segmentado en MIPS

## Ejercicio 1 :

Request	08389
User	g510
Queue	lb_stall
Program	p2-funcs.asm
Status	finished - Program passed
Date	23-10-2023 15:17
Time	1.32867s
Lboard Cont. Instr.	1042
Lboard Tiempo ejecución (ciclos)	2787
Lboard Pen. riesgos datos	1067
Lboard Pen. riesgos control	666
Lboard CPI	2.67
Lboard Excepc.	0
Lboard Superados	3
Output	This is a leaderboard request, output not available.

Request	12877
User	g510
Queue	lb_stall
Program	p2-funcsmejorado2.asm
Status	finished - Program passed
Date	29-10-2023 17:20
Time	1.37270s
Lboard Cont. Instr.	1268
Lboard Tiempo ejecución (ciclos)	1880
Lboard Pen. riesgos datos	528
Lboard Pen. riesgos control	72
Lboard CPI	1.48
Lboard Excepc.	0
Lboard Superados	3
Output	This is a leaderboard request, output not available.

El rendimiento inicial es malo por los altos riesgos de datos provocados por la no anticipación y que las estructuras condicionales no están previstas para ser usadas con un procesador que prediga no tomado.

Para rebajar los riesgos de control, como el procesador predice no tomado y en su mayoría, las instrucciones de salto de este programa son bucles, se cambian los beq y bne por su contrario y justo después se pone una instrucción j, así conseguiremos que todas las iteraciones del bucle excepto la última sean no tomado y evitaremos esos riesgos (esta explicación es aplicable también al resto de ejercicios). Para rebajar los riesgos de datos hemos reescrito instrucciones MIPS formadas por más de una instrucción máquina y después las hemos reordenado, evitando así esas dependencias de datos.

Speed up:  $2787/1880=1,4824 \Rightarrow$  mejora del 48,24%

## Ejercicio 2 :

Request	08390
User	g510
Queue	lb_forw
Program	p2-funcs.asm
Status	finished - Program passed
Date	23-10-2023 15:18
Time	1.35614s
Lboard Cont. Instr.	1042
Lboard Tiempo ejecución (ciclos)	1699
Lboard Pen. riesgos datos	423
Lboard Pen. riesgos control	222
Lboard CPI	1.63
Lboard Excepc.	0
Lboard Superados	3
Output	This is a leaderboard request, output not available.

Request	12875
User	g510
Queue	lb_forw
Program	p2-funcsmejorado2.asm
Status	finished - Program passed
Date	29-10-2023 17:18
Time	1.34121s
Lboard Cont. Instr.	1268
Lboard Tiempo ejecución (ciclos)	1368
Lboard Pen. riesgos datos	64
Lboard Pen. riesgos control	24
Lboard CPI	1.08
Lboard Excepc.	0
Lboard Superados	3
Output	This is a leaderboard request, output not available.

El rendimiento inicial, aunque mejor que el del apartado anterior, sigue siendo malo ya que el código no está bien distribuido. Para reducir los riesgos de control hemos hecho básicamente lo mismo que

en el apartado anterior, pero como ahora se resuelven en la etapa ID, hay menos riesgos. Para los de datos hemos reordenado instrucciones también, teniendo en cuenta que ahora si hay anticipación, por lo cual lo más importante era reducir los riesgos provocados por los lw

Speed up:  $1899/1368 = 1,388 \Rightarrow$  Mejora del 38,8%

### Ejercicio 3 :

#### ATOI VIEJO

Request	06861
User	g510
Queue	lb_pipe_atoi
Program	atoi.asm
Status	finished - Program passed
Date	18-10-2023 17:11
Time	11.24448s
Lboard Cont. Instr.	6198
Lboard Tiempo ejecución (ciclos)	7686
Lboard Pen. riesgos datos	1300
Lboard Pen. riesgos control	88
Lboard CPI	1.24
Lboard Runtime Excepc.	0
Lboard Error Contexto	0
Lboard Casos	23
Lboard % superado	92.00
Output	This is a leaderboard request, output not available.

#### ATOI NUEVO

Request	12520
User	g510
Queue	lb_pipe_atoi
Program	atoiMejorado.asm
Status	finished - Program passed
Date	29-10-2023 11:09
Time	11.25836s
Lboard Cont. Instr.	5632
Lboard Tiempo ejecución (ciclos)	6020
Lboard Pen. riesgos datos	200
Lboard Pen. riesgos control	88
Lboard CPI	1.07
Lboard Runtime Excepc.	0
Lboard Error Contexto	0
Lboard Casos	23
Lboard % superado	92.00
Output	This is a leaderboard request, output not available.

Lo que hemos realizado para reducir el número de ciclos del programa ha sido cambiar de sitio las instrucciones del bucle principal para que haya los mínimos riesgos de datos. Además, con los riesgos de control hemos seguido la misma lógica que en el ejercicio 1.

Speed up:  $7686/6020 = 1,277 \rightarrow$  ha mejorado un 27,7%

#### ITOA VIEJO

Request	07937
User	g510
Queue	lb_pipe_itoa
Program	itoa.asm
Status	finished - Program passed
Date	22-10-2023 12:24
Time	12.49159s
Lboard Cont. Instr.	3989
Lboard Tiempo ejecución (ciclos)	4697
Lboard Pen. riesgos datos	382
Lboard Pen. riesgos control	218
Lboard CPI	1.18
Lboard Runtime Excepc.	1
Lboard Error Contexto	0
Lboard Casos	27
Lboard % superado	96.43
Output	This is a leaderboard request, output not available.

#### ITOA NUEVO

Request	12522
User	g510
Queue	lb_pipe_itoa
Program	itoaMejorado.asm
Status	finished - Program passed
Date	29-10-2023 11:11
Time	12.73074s
Lboard Cont. Instr.	3853
Lboard Tiempo ejecución (ciclos)	4222
Lboard Pen. riesgos datos	43
Lboard Pen. riesgos control	218
Lboard CPI	1.10
Lboard Runtime Excepc.	1
Lboard Error Contexto	0
Lboard Casos	27
Lboard % superado	96.43
Output	This is a leaderboard request, output not available.

Lo que hemos hecho en itoa para reducir el número de ciclos del programa ha sido:

- Cambiar la comprobación del salto según el valor de \$a1, ya que si vamos incrementando un registro a la vez que vamos haciendo la comprobación, evitamos un gran número de riesgos de control y de datos.
- Cambiar el sitio de las instrucciones del bucle principal y el de cambio de posición para eliminar esa dependencia de datos y de control
- Para los riesgos de control hemos seguido también la misma lógica que en el ejercicio 1

Speed up:  $4697/4222 = 1,113 \rightarrow$  ha mejorado un 11,3%