

# Tema 4. Ejercicios

Arquitectura de los Computadores

# Ejercicio 1

Suponed el siguiente fragmento de código:

```
Loop:   LD      R1, 0(R2)  ; load R1 de la dirección 0+R2
        DADDI   R1, R1, #1 ; R1=R1+1
        SD      0(R2), R1  ; store R1 en dirección 0+R2
        DADDI   R2, R2, #4
        DSUB    R4, R3, R2 ; R4=R3-R2
        BNEZ    R4, Loop   ; salta a loop si R4<>0
```

Asumid que el valor inicial de R3 es R2+396

- a) Lista todas las dependencias de datos del código anterior. Apunta el registro, instrucción fuente e instrucción destino, Por ejemplo si hay una dependencia para el registro R1 desde la instrucción LD a la instrucción DADDI apuntamos: R1 LD DADDI

# Ejercicio 1

- a) Lista todas las dependencias de datos del código anterior. Apunta el registro, instrucción fuente e instrucción destino, Por ejemplo si hay una dependencia para el registro R1 desde la instrucción LD a la instrucción DADDI apuntamos: R1 LD DADDI

R1	LD	DADDI
R1	DADDI	SD
R2	LD	DADDI
R2	SD	DADDI
R2	DSUB	DADDI
R4	BNEZ	DSUB

# Ejercicio 1

Suponed el siguiente fragmento de código:

```
Loop:   LD      R1, 0(R2)  ; load R1 de la dirección 0+R2
        DADDI   R1, R1, #1 ; R1=R1+1
        SD      0(R2), R1  ; store R1 en dirección 0+R2
        DADDI   R2, R2, #4
        DSUB    R4, R3, R2  ; R4=R3-R2
        BNEZ    R4, Loop    ; salta a loop si R4<>0
```

Asumid que el valor inicial de R3 es R2+396

- b) Muestra la temporización de la secuencia de instrucciones en un diagrama para un pipeline de 5 etapas como el estudiado que no tenga implementado el adelantamiento. Suponed que el salto se realiza deteniendo la segmentación. Si todas las referencias a memoria tardan 1 ciclo de reloj, ¿Cuántos ciclos tardará el blucle en ejecutarse?

# Ejercicio 1

- b) Muestra la temporización de la secuencia de instrucciones para un pipeline de 5 etapas como el estudiado que no tenga implementado el adelantamiento. Suponed que el salto se realiza deteniendo la segmentación. Si todas las referencias a memoria tardan 1 ciclo de reloj, ¿Cuántos ciclos tardará el blucle en ejecutarse?

Forwarding is performed only via the register file. Branch outcomes and targets are not known until the end of the execute stage. All instructions introduced to the pipeline prior to this point are flushed.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LD	R1, 0(R2)	F	D	X	M	W													
DADDI	R1, R1, #1		F	s	s	D	X	M	W										
SD	0(R2), R1					F	s	s	D	X	M	W							
DADDI	R2, R2, #4								F	D	X	M	W						
DSUB	R4, R3, R2									F	s	s	D	X	M	W			
BNEZ	R4, Loop												F	s	s	D	X	M	W
LD	R1, 0(R2)																	F	D

Since the initial value of R3 is R2 + 396 and equal instance of the loop adds 4 to R2, the total number of iterations is 99. Notice that there are 8 cycles lost to RAW hazards including the branch instruction. Two cycles are lost after the branch because of the instruction flushing. It takes 16 cycles between loop instances; the total number of cycles is  $98 \times 16 + 18 = 1584$ . The last loop takes two addition cycles since this latency cannot be overlapped with additional loop instances.

# Ejercicio 1

Suponed el siguiente fragmento de código:

```
Loop:   LD      R1, 0(R2)  ; load R1 de la dirección 0+R2
        DADDI   R1, R1, #1 ; R1=R1+1
        SD      0(R2), R1  ; store R1 en dirección 0+R2
        DADDI   R2, R2, #4
        DSUB    R4, R3, R2  ; R4=R3-R2
        BNEZ    R4, Loop    ; salta a loop si R4<>0
```

Asumid que el valor inicial de R3 es R2+396

- c) Dibuja de nuevo el diagrama de temporización suponiendo que hay forwarding. Suponed que para los saltos se utiliza predecir el salto como no efectivo. Si todas las referencias a memoria tardan 1 ciclo de reloj, ¿Cuántos ciclos tardará el blucle en ejecutarse?

# Ejercicio 1

- c) Dibuja de nuevo el diagrama de temporización suponiendo que hay forwarding. Suponed que para los saltos se utiliza predecir el salto como no efectivo. Si todas las referencias a memoria tardan 1 ciclo de reloj, ¿Cuántos ciclos tardará el blucle en ejecutarse?

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LD	R1, 0(R2)	F	D	X	M	W													
DADDI	R1, R1, #1		F	D	s	X	M	W											
SD	R1, 0(R2)			F	s	D	X	M	W										
DADDI	R2, R2, #4					F	D	X	M	W									
DSUB	R4, R3, R2						F	D	X	M	W								
BNEZ	R4, Loop							F	s	D	X	M	W						
(incorrect instruction)										F	s	s	s	s					
LD	R1, 0(R2)										F	D	X	M	W				

Again we have 99 iterations. There are two RAW stalls and a flush after the branch since the branch is taken. The total number of cycles is  $9 \times 98 + 12 = 894$ . The last loop takes three addition cycles since this latency cannot be overlapped with additional loop instances.

# Ejercicio 1

Suponed el siguiente fragmento de código:

```
Loop:   LD      R1, 0(R2)  ; load R1 de la dirección 0+R2
        DADDI   R1, R1, #1 ; R1=R1+1
        SD      0(R2), R1  ; store R1 en dirección 0+R2
        DADDI   R2, R2, #4
        DSUB    R4, R3, R2 ; R4=R3-R2
        BNEZ    R4, Loop   ; salta a loop si R4<>0
```

Asumid que el valor inicial de R3 es R2+396

- d) Dibuja de nuevo el diagrama de temporización suponiendo que hay forwarding. Suponed que para los saltos se utiliza predecir el salto como efectivo. Si todas las referencias a memoria tardan 1 ciclo de reloj, ¿Cuántos ciclos tardará el bucle en ejecutarse?



# Ejercicio 1

- d) Dibuja de nuevo el diagrama de temporización suponiendo que hay forwarding. Suponed que para los saltos se utiliza predecir el salto como efectivo. Si todas las referencias a memoria tardan 1 ciclo de reloj, ¿Cuántos ciclos tardará el bucle en ejecutarse?

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LD	R1, 0(R2)	F	D	X	M	W													
DADDI	R1, R1, #1		F	D	s	X	M	W											
SD	R1, 0(R2)			F	s	D	X	M	W										
DADDI	R2, R2, #4					F	D	X	M	W									
DSUB	R4, R3, R2						F	D	X	M	W								
BNEZ	R4, Loop							F	s	D	X	M	W						
LD	R1, 0(R2)									F	D	X	M	W					

Again we have 99 iterations. We still experience two RAW stalls, but since we correctly predict the branch, we do not need to flush after the branch. Thus, we have only  $8 \times 98 + 12 = 796$ .

# Ejercicio 1

Suponed el siguiente fragmento de código:

```
Loop:  LD      R1, 0(R2)  ; load R1 de la dirección 0+R2
       DADDI   R1, R1, #1 ; R1=R1+1
       SD      0(R2), R1  ; store R1 en dirección 0+R2
       DADDI   R2, R2, #4
       DSUB    R4, R3, R2  ; R4=R3-R2
       BNEZ    R4, Loop    ; salta a loop si R4<>0
```

Asumid que el valor inicial de R3 es R2+396

- e) Suponer ahora que se tiene un pipeline de 10 etapas en el que cada etapa del cauce de 5 etapas se divide en 2. Hay que tener en cuenta que los datos que se adelantan, son adelantados desde el final del par de etapas al comienzo del par de etapas donde se necesitan. Por ejemplo, los datos se adelantarían desde la salida de la segunda etapa EX a la entrada de la primera etapa EX, lo que causaría un retardo de un ciclo. Muestra de nuevo el diagrama de temporización para la secuencia de instrucciones suponiendo que implementa el forwarding y que se utiliza predecir el salto como efectivo. Si todas las referencias a memoria tardan 1 ciclo de reloj, ¿Cuántos ciclos tardará el bucle en ejecutarse?

# Ejercicio 1

- e) Suponer ahora que se tiene un pipeline de 10 etapas en el que cada etapa del cauce de 5 etapas se divide en 2. Hay que tener en cuenta que los datos que se adelantan, son adelantados desde el final del par de etapas al comienzo del par de etapas donde se necesitan. Por ejemplo, los datos se adelantarían desde la salida de la segunda etapa EX a la entrada de la primera etapa EX, lo que causaría un retardo de un ciclo. Muestra de nuevo el diagrama de temporización para la secuencia de instrucciones suponiendo que implementa el forwarding y que se utiliza predecir el salto como efectivo. Si todas las referencias a memoria tardan 1 ciclo de reloj, ¿Cuántos ciclos tardará el bucle en ejecutarse?

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LD	R1, 0(R2)	F1	F2	D1	D2	X1	X2	M1	M2	W1	W2										
DADDI	R1, R1, #1		F1	F2	D1	D2	s	s	s	X1	X2	M1	M2	W1	W2						
SD	R1, 0(R2)			F1	F2	D1	s	s	s	D2	X1	X2	M1	M2	W1	W2					
DADDI	R2, R2, #4				F1	F2	s	s	s	D1	D2	X1	X2	M1	M2	W1	W2				
DSUB	R4, R3, R2					F1	s	s	s	F2	D1	D2	s	X1	X2	M1	M2	W1	W2		
BNEZ	R4, Loop									F1	F2	D1	s	D2	X1	X2	M1	M2	W1	W2	
LD	R1, 0(R2)										F1	F2	s	D1	D2	X1	X2	M1	M2	W1	W2

We again have 99 iterations. There are three RAW stalls between the LD and ADDI, and one RAW stall between the DADDI and DSUB. Because of the branch prediction, 98 of those iterations overlap significantly. The total number of cycles is  $10 \times 98 + 19 = 999$ .

# Ejercicio 1

Suponed el siguiente fragmento de código:

```
Loop:  LD      R1, 0(R2)  ; load R1 de la dirección 0+R2
       DADDI   R1, R1, #1 ; R1=R1+1
       SD      0(R2), R1  ; store R1 en dirección 0+R2
       DADDI   R2, R2, #4
       DSUB    R4, R3, R2 ; R4=R3-R2
       BNEZ    R4, Loop   ; salta a loop si R4<>0
```

Asumid que el valor inicial de R3 es R2+396

- f) Suponed que en el cauce de 5 etapas, la etapa más larga necesita 0.8ns y que el retardo de los registros de segmentación es de 0.1ns. ¿Cuál es la duración del ciclo de reloj para el pipeline de 5 etapas?. Si el cauce de 10 etapas divide cada etapa en la mitad, ¿Cuál es la duración del ciclo de reloj para el pipeline de 10 etapas?.

# Ejercicio 1

- f) Suponed que en el cauce de 5 etapas, la etapa más larga necesita 0.8ns y que el retardo de los registros de segmentación es de 0.1ns. ¿Cuál es la duración del ciclo de reloj para el pipeline de 5 etapas?. Si el cauce de 10 etapas divide cada etapa en la mitad, ¿Cuál es la duración del ciclo de reloj para el pipeline de 10 etapas?.

0.9ns for the 5-stage pipeline and 0.5ns for the 10-stage pipeline.

# Ejercicio 1

Suponed el siguiente fragmento de código:

```
Loop:   LD      R1, 0(R2)  ; load R1 de la dirección 0+R2
        DADDI   R1, R1, #1 ; R1=R1+1
        SD      0(R2), R1  ; store R1 en dirección 0+R2
        DADDI   R2, R2, #4
        DSUB    R4, R3, R2 ; R4=R3-R2
        BNEZ    R4, Loop   ; salta a loop si R4<>0
```

Asumid que el valor inicial de R3 es R2+396

- g) Determinad el CPI para el bucle en el cauce de 5 etapas y en el de 10 etapas . Aseguraros de contar sólo desde cuando la primera instrucción alcanza la etapa WB al final. No contar el comienzo de la primera instrucción. Utilizad el valor del ciclo de reloj calculado en el apartado (f) para obtener el tiempo medio de ejecución por instrucción para cada máquina.

# Ejercicio 1

- g) Determinad el CPI para el bucle en el cauce de 5 etapas y en el de 10 etapas . Aseguraros de contar sólo desde cuando la primera instrucción alcanza la etapa WB al final. No contar el comienzo de la primera instrucción. Utilizad el valor del ciclo de reloj calculado en el apartado (f) para obtener el tiempo medio de ejecución por instrucción para cada máquina.

CPI of 5-stage pipeline:  $796 / (99 \times 6) = 1.34$ .

CPI of 10-stage pipeline:  $999 / (99 \times 6) = 1.68$ .

Avg Inst Exe Time 5-stage:  $1.34 \times 0.9 = 1.21$ .

Avg Inst Exe Time 10-stage:  $1.68 \times 0.5 = 0.84$ .