EXERCISE 1 (A) – PIPELINE BASIC (ASSIGNMENT)

Given the following loop expressed in a high level language:

```
for (i =0; i < N; i ++)
      vectA[i] = vectA[i] + vectB[i];
```

The program has been compiled in MIPS assembly code assuming that registers `$t6` and `$t7`have been initialized with values 0 and 4 N respectively. The symbols VECTA, VECTB and VECTC are 16-bit constant. The processor clock frequency is **1 GHz.**

INSTRUCTION	Comment
`FOR1:beq $t6,$t7, END`	`# if ($t6 == $t7) goto END`
`lw $t2,VECTA($t6)`	`# $t2 <- VECTA [$t6];`
`lw $t3,VECTB($t6)`	`# $t3 <- VECTB [$t6];`
`add $t2,$t2,$t3`	`# $t2 <- $t2 + $t3;`
`sw $t2,VECTA($t6)`	`# VECTA[$t6] <- $t2;`
`addi $t6,$t6,4`	`# $t6 <- $t6 + 4;`
`j FOR1`	`# goto FOR1;`
`END:`	

Let us consider a single iteration of the loop executed by 5-stage pipelined MIPS processor **without** any optimization in the pipeline.

- Identify the **RAW (Read After Write)** data hazards by marking in RED and control hazards in BLUE
- Identify the number of stalls to be inserted **before each instruction (or between the stage IF and ID of each instruction)** necessary to solve the hazards.

Num. Stalls	INSTRUCTION	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	Hazard Type
	`FOR1:beq $t6,$t7, END`	IF	ID	EX	ME	WB							
	`lw $t2,VECTA($t6)`		IF	ID	EX	ME	WB						
	`lw $t3,VECTB($t6)`			IF	ID	EX	ME	WB					
	`add $t2,$t2,$t3`				IF	ID	EX	ME	WB				
	`sw $t2,VECTA($t6)`					IF	ID	EX	ME	WB			
	`addi $t6,$t6,4`						IF	ID	EX	ME	WB		
	`j FOR1`							IF	ID	EX	ME	WB	

EXERCISE 1 (A) – PIPELINE BASIC (SOLUTION)

Let us consider a **single iteration** of the loop executed by 5-stage pipelined MIPS processor **without** any optimization in the pipeline.

- Identify the **RAW (Read After Write)** data hazards by marking in RED and control hazards in BLUE
- Identify the number of stalls to be inserted **before each instruction (or between the stage IF and ID of each instruction)** necessary to solve the hazards.

SOLUTION (not including inter-iteration dependencies)

Num. Stalls	INSTRUCTION	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	Hazard Type
3	`FOR1:beq $t6,$t7, END`	IF	ID	EX	ME	WB							CNTR
3	`lw $t2,VECTA($t6)`		IF	ID	EX	ME	WB						CNTR
	`lw $t3,VECTB($t6)`			IF	ID	EX	ME	WB					
3	`add $t2,$t2,$t3`				IF	ID	EX	ME	WB				RAW $t3 (RAW $t2)
3	`sw $t2,VECTA($t6)`					IF	ID	EX	ME	WB			RAW $t2
	`addi $t6,$t6,4`						IF	ID	EX	ME	WB		
	`j FOR1`							IF	ID	EX	ME	WB	

More detailed solution (more detailed scheme with also inter-iteration dependencies)

Num. Stalls	INSTRUCTION	C1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Hazard Type
3	`FOR1:beq $t6,$t7, END`	IF-S	IF-S	IF-S	IF	ID	EX	ME	WB																CNTR (RAW $t6 inter-iteration)
3	`lw $t2,VECTA($t6)`					IF-S	IF-S	IF-S	IF	ID	EX	ME	WB												CNTR
	`lw $t3,VECTB($t6)`									IF	ID	EX	ME	WB											
3	`add $t2,$t2,$t3`										IF	ID-S	ID-S	ID-S	ID	EX	ME	WB							RAW $t3 (RAW $t2)
3	`sw $t2,VECTA($t6)`											IF-S	IF-S	IF-S	IF	ID-S	ID-S	ID-S	ID	EX	ME	WB			RAW $t2
	`addi $t6,$t6,4`															IF-S	IF-S	IF-S	IF	ID	EX	ME	WB		
	`j FOR1`																			IF	ID	EX	ME	WB	

Express the formulas, then calculate the following metrics:

- Instruction Count **(IC)** = **7**
- Number of stalls per iteration = **12**
- **CPI** per iteration: **CPI = # cycles / IC = (IC+ # stalls + 4) /IC = 23 / 7 = 3,29**
- **Throughput** (expressed in **MIPS**) per iteration: **MIPS =** f_{CLOCK} **/ (CPI * 10^6) = (10^9) / (3,29 * 10^6) = 303,95**
- Asymptotic **CPI** (**N** cycles) : **CPI** $_{AS}$**= (IC + # stalls) / IC = (7 + 12) / 7 = 2,71**
- Asymptotic **Throughput** (expressed in **MIPS**) (**N** cycles): **MIPS**$_{AS}$ **=** f_{CLOCK} **/ (CPI$_{AS}$ * 10^6) = (10^9) / (2,71 * 10^6) = 369**

EXERCISE 1(B) – PIPELINE OPTIMIZATIONS

Assuming there are the following optimisations in the pipeline

- **In the Register File it is possible the read and write at the same address in the same clock cycle;**
- **Forwarding**
- **Computation of PC and TARGET ADDRESS for branch & jump instructions anticipated in the ID stage**

1. Identify the **RAW (Read After Write)** data hazards and the control hazards in the pipeline scheme
2. Identify the number of stalls to be inserted **before each instruction (or between the stage IF and ID of each instruction)** necessary to solve the hazards.
3. Identify in the last columns the hazard type and the forwarding path used:

SOLUTION:

Num. Stalls	INSTRUCTION	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	Hazard Type	Forwarding Path
1	`FOR1:beq $t6,$t7, END`	IF	ID	EX	ME	WB							CNTR	
1	`lw $t2,VECTA($t6)`		IF	ID	EX	ME	WB						CNTR	
	`lw $t3,VECTB($t6)`			IF	ID	EX	ME	WB						
1	`add $t2,$t2,$t3`				IF	ID	EX	ME	WB				LD-USE $t3	MEM-EX $t2
	`sw $t2,VECTA($t6)`					IF	ID	EX	ME	WB				EX-EX $t2
	`addi $t6,$t6,4`						IF	ID	EX	ME	WB			
	`j FOR1`							IF	ID	EX	ME	WB		

More detailed solution (more detailed scheme and inter-iteration dependencies)

Num. Stalls	INSTRUCTION	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	Hazard Type	Forwarding Path
1	`FOR1:beq $t6,$t7, END`	IF-S	IF	ID	EX	ME	WB									CNTR	
1	`lw $t2,VECTA($t6)`			IF-S	IF	ID	EX	ME	WB							CNTR	
	`lw $t3,VECTB($t6)`					IF	ID	EX	ME	WB							
1	`add $t2,$t2,$t3`						IF	ID-S	ID	EX	ME	WB				LD-USE $t3	MEM-EX $t3
	`sw $t2,VECTA($t6)`							IF-S	IF	ID	EX	ME	WB				EX-EX $t2
	`addi $t6,$t6,4`									IF	ID	EX	ME	WB			
	`j FOR1`										IF	ID	EX	ME	WB		
1*	`FOR1:beq $t6,$t7, END`											IF-S	IF	ID	EX		(read $t6 inter-iteration)

`(*) next iteration of the loop`

Calculate the following metrics:

- Instruction Count **(IC)** = **7**
- Number of stalls per iteration = **3**
- **CPI** per iteration: **CPI = # cycles / IC = (IC+ # stalls + 4) /IC = 14 / 7 = 2**
- **Throughput** (expressed in **MIPS**) per iteration: **MIPS =** f_{CLOCK} **/ (CPI * 10^6) = (10^9) / (2 * 10^6) = 500**
- Asymptotic **CPI** (**N** cycles) : **CPI** $_{AS}$**= (IC + # stalls) / IC = (7 + 3) / 7 = 1,43**
- Asymptotic **Throughput** (expressed in **MIPS**) (**N** cycles): **MIPS**$_{AS}$ **=** f_{CLOCK} **/ (CPI$_{AS}$ * 10^6) = (10^9) / (1,43 * 10^6) = 699,3**

Calculate the speedup with respect to the previous case (EX. 1A):

- **Speedup = CPI$_{AS1A}$/CPI$_{AS1B}$ = 2.71 /1.43 = 1.9**

EXERCISE 1 (C) – BRANCH PREDICTION

Assuming there are the previous optimisations in the pipeline with static branch prediction BTFNT (BACKWARD TAKEN FORWARD NOT TAKEN) with BRANCH TARGET BUFFER.

1. Identify the **RAW (Read After Write)** data hazards and control hazards.
2. Identify the **number of stalls** to be inserted before each instruction (or between the stage IF and ID of each instruction) necessary to solve the hazards.
3. Identify the Static Branch Prediction **(Taken/Not Taken)**
4. Identify in the last columns the hazard type and the forwarding path used:

SOLUTION:

Num. Stalls	INSTRUCTION	PRED T/NT	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	Hazard Type	Forwarding Path
	`FOR1:beq $t6,$t7, END`	NT	IF	ID	EX	ME	WB								
	`lw $t2,VECTA($t6)`	-		IF	ID	EX	ME	WB							
	`lw $t3,VECTB($t6)`	-			IF	ID	EX	ME	WB						
1	`add $t2,$t2,$t3`	-				IF	ID	EX	ME	WB				LD-USE $t3	MEM-EX $t2
	`sw $t2,VECTA($t6)`	-					IF	ID	EX	ME	WB				EX-EX $t2
	`addi $t6,$t6,4`	-						IF	ID	EX	ME	WB			
	`j FOR1`	T							IF	ID	EX	ME	WB		

More detailed solution (more detailed scheme and inter-iteration dependencies)

Num. Stalls	INSTRUCTION	T/NT	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	Hazard Type	Forwarding Path
	`FOR1:beq $t6,$t7, END`	NT	IF	ID	EX	ME	WB									
	`lw $t2,VECTA($t6)`			IF	ID	EX	ME	WB								
	`lw $t3,VECTB($t6)`				IF	ID	EX	ME	WB							
1	`add $t2,$t2,$t3`					IF	ID-S	ID	EX	ME	WB				LD-USE $t3	MEM-EX $t3
	`sw $t2,VECTA($t6)`						IF-S	IF	ID	EX	ME	WB				EX-EX $t2
	`addi $t6,$t6,4`								IF	ID	EX	ME	WB			
	`j FOR1`	T								IF	ID	EX	ME	WB		
(*)	`FOR1:beq $t6,$t7, END`	NT									IF	ID	EX	ME		EX-ID $t6 (inter-iteration)

`(*) next iteration of the loop`

Calculate the following metrics:

- Instruction Count **(IC)** = **7**
- Number of stalls per iteration = **1**
- **CPI** per iteration: **CPI = # cycles / IC = (IC+ # stalls + 4) /IC = 12 / 7 = 1.71**
- **Throughput** (expressed in **MIPS**) per iteration: **MIPS =** f_{CLOCK} **/ (CPI * 10^6) = (10^9) / (1.71 * 10^6) = 584,8**
- Asymptotic **CPI** (**N** cycles) : **CPI** $_{AS}$**= (IC + # stalls) / IC = (7 + 1) / 7 = 1,14**
- Asymptotic **Throughput** (expressed in **MIPS**) (**N** cycles): **MIPS**$_{AS}$ **=** f_{CLOCK} **/ (CPI**$_{AS}$ *** 10^6) = (10^9) / (1,14 * 10^6) = 877,2**

Calculate the speedup with respect to the previous case (EX. 1B):

- **Speedup =** **CPI**$_{AS1B}$**/CPI**$_{AS1C}$ **= 1.43 /1.14 = 1.25**