EXERCISE 1 (A)

Given the following loop taken from a high level program:

```
do      {
         BASEC[i] = BASEA[i] + BASEB[i] + INC1 + INC2;
         i++;
        }
while (i != N)
```

The program has been compiled in MIPS assembly code assuming that registers $4 and $7 have been initialized with values 0 and 4N respectively. The symbols BASEA, BASEB and BASEC are 16-bit constant. The processor clock cycle is 2 ns.

```
L1:     lw   $2, BASEA ($4)
        addi $2, $2, INC1
        lw   $3, BASEB ($4)
        addi $3, $3, INC2
        add  $5, $2, $3
        sw   $5, BASEC ($4)
        addi $4, $4, 4
        bne  $4, $7, L1
```

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

- **Draw the pipeline scheme by marking in RED the RAW (Read After Write) data hazards and in BLUE the control hazards:**

Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	Type of Hazards
`L1: lw$2,BASEA($4)`	IF	ID	EX	M	WB								
`addi $2,$2,INC1`		IF	ID	EX	M	WB							RAW $2
`lw $3,BASEB($4)`			IF	ID	EX	M	WB						
`addi $3,$3,INC2`				IF	ID	EX	M	WB					RAW $3
`add $5,$2,$3`					IF	ID	EX	M	WB				RAW $3 (RAW $2)
`sw $5,BASEC($4)`						IF	ID	EX	M	WB			RAW $5
`addi $4, $4, 4`							IF	ID	EX	M	WB		
`bne $4,$7,L1`								IF	ID	EX	M	WB	RAW $4
									IF				CNTR

- Insert in the following pipeline scheme the stalls needed to solve the previous data and control hazards:

Instruction	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	C 11	C 12	C 13	C 14	C 15	C 16	C 17	C 18	C 19	C 20	C 21	C 22	C 23	C 24	C 25	C 26	C 27	C 28	C 29	C 30
`L1:lw$2,BASEA($4)`	**IFS**	**IFS**	**IFS**	**IF**	ID	EX	M	WB																						
`addi $2,$2,INC1`					IF	**IDS**	**IDS**	**IDS**	**ID**	EX	M	WB																		
`lw $3,BASEB($4)`						**IFS**	**IFS**	**IFS**	IF	ID	EX	M	WB																	
`addi $3,$3,INC2`										IF	**IDS**	**IDS**	**IDS**	**ID**	EX	M	WB													
`add $5,$2,$3`											**IFS**	**IFS**	**IFS**	IF	**IDS**	**IDS**	**IDS**	**ID**	EX	M	WB									
`sw $5,BASEC($4)`															**IFS**	**IFS**	**IFS**	IF	**IDS**	**IDS**	**IDS**	**ID**	EX	M	WB					
`addi $4, $4, 4`																			**IFS**	**IFS**	**IFS**	IF	ID	EX	M	WB				
`bne $4,$7,L1`																							IF	**IDS**	**IDS**	**IDS**	**ID**	EX	M	WB

- Indicate synthetically in the first column the NUMBER OF STALLS to be inserted before (or during) each instruction to solve data and control hazards:

Num. of Stalls	Instruction	Type of Hazards
3	`L1:lw$2,BASEA($4)`	CNTR
3	`addi $2,$2,INC1`	RAW $2
	`lw $3,BASEB($4)`	
3	`addi $3,$3,INC2`	RAW $3
3	`add $5,$2,$3`	RAW $3
3	`sw $5,BASEC($4)`	RAW $5
	`addi $4, $4, 4`	
3	`bne $4,$7,L1`	RAW $4

- **Express the formulas, then calculate the following metrics:**
  - Instruction Count per iteration **(IC)** = **8**
  - Number of stalls per iteration = **18**
  - **CPI** per iteration: **CPI = Number of clock cycles / IC = (IC+ # stalls + 4) /IC = 30 / 8 = 3,75**
  - **Throughput** (expressed in **MIPS**) per iteration: **MIPS =** f_{CLOCK} **/ (CPI * 10^6) = (500 * 10^6) / (3,75 * 10^6) = 133, 3**
  - Asymptotic **CPI** (**N** cycles) : **CPI** $_{AS}$**= (IC + # stalls) / IC = (8 + 18) / 8 = 3,25**
  - Asymptotic **Throughput** (expressed in **MIPS**) (**N** cycles): **MIPS**$_{AS}$ **=** f_{CLOCK} **/ (CPI**$_{AS}$ *** 10^6) = (500 * 10^6) / (3,25 * 10^6) = 153,8**

EXERCISE 1 (B)

Let us assume the following ***optimizations*** have been introduced in the pipeline:

- **In the Register File, it is possible to read and write at the same address at the same clock cycle;**
- **Forwarding paths have been introduced;**
- **Program Counter calculation for branches has been anticipated in the ID stage;**

- **Draw the pipeline scheme by marking in RED the RAW (Read After Write) data hazards and in BLUE the control hazards still remaining in the pipeline:**

Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	Type of Hazards
L1: lw$2,BASEA($4)	IF	ID	EX	M	WB										CNTR
addi $2,$2,INC1		IF	ID	EX	M	WB									RAW $2
lw $3,BASEB($4)			IF	ID	EX	M	WB								
addi $3,$3,INC2				IF	ID	EX	M	WB							RAW $3
add $5,$2,$3					IF	ID	EX	M	WB						
sw $5,BASEC($4)						IF	ID	EX	M	WB					
addi $4, $4, 4							IF	ID	EX	M	WB				
bne $4,$7,L1								IF	ID	X	M	WB			RAW $4

- **Insert in the following pipeline scheme the stalls needed to solve the hazards by marking in GREEN the forwarding paths used:**

Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	Forwarding path
L1: lw$2,BASEA($4)	IF-S	IF	ID	EX	M	WB											
addi $2,$2,INC1			IF	ID	EX-S	EX	M	WB									MEM-EX $2
lw $3,BASEB($4)				IF	ID-S	ID	EX	M	WB								
addi $3,$3,INC2						IF	ID	EX-S	EX	M	WB						MEM-EX $3
add $5,$2,$3							IF	ID-S	ID	EX	M	WB					EX-EX $3
sw $5,BASEC($4)									IF	ID	EX	M	WB				EX-EX $5
addi $4, $4, 4										IF	ID	EX	M	WB			
bne $4,$7, L1											IF	ID-S	ID	EX	M	WB	EX-ID $4

- **Indicate** synthetically in the first column the number of stalls to be inserted before (or during) each instruction to solve the hazards:

Num. of Stalls	Instructions	Type of Hazard
1	`L1: lw $2,BASEA($4)`	CNTR
1	`addi $2,$2,INC1`	RAW $2
	`lw $3,BASEB($4)`	
1	`addi $3,$3,INC2`	RAW $3
	`add $5,$2,$3`	
	`sw $5,BASEC($4)`	
	`addi $4, $4, 4`	
1	`bne $4,$7,L1`	RAW $4

- **Express the formulas, then calculate the following metrics:**

  - Instruction Count per iteration **(IC)** = **8**
  - Number of stalls per iteration = **4**
  - **CPI** per iteration: **CPI** = **# cycles / IC = (IC+ # stalls + 4) /IC = 16 / 8 = 2**
  - **Throughput** (expressed in **MIPS**) per iteration: **MIPS** = f_{CLOCK} / (CPI * 10^6) = (500 * 10^6) / (2 * 10^6) = 250
  - Asymptotic **CPI** (**N** cycles) : **CPI** $_{AS}$= (IC+ # stalls) / IC = (8 + 4) / 8 = 1,5
  - Asymptotic **Throughput** (expressed in **MIPS**) (**N** cycles): **MIPS**$_{AS}$ = f_{CLOCK} / (CPI_{AS} * 10^6) = (500 * 10^6) / (1,5 * 10^6) = 333,33

EXERCISE 1 (C)

Besides the previous optimizations, we assume to introduce **Static branch prediction for backward branches: branch always taken**

- **Draw the pipeline scheme by marking in RED the RAW (Read After Write) data hazards and in BLUE the control hazards still remaining in the program:**

Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	Type of Hazards
`L1: lw$2,BASEA($4)`	IF	ID	EX	**M**	WB										
`addi $2,$2,INC1`		IF	ID	**EX**	M	WB									**RAW $2**
`lw $3,BASEB($4)`			IF	ID	EX	**M**	WB								
`addi $3,$3,INC2`				IF	ID	**EX**	M	WB							**RAW $3**
`add $5,$2,$3`					IF	ID	EX	M	WB						
`sw $5,BASEC($4)`						IF	ID	EX	M	WB					
`addi $4, $4, 4`							IF	ID	**EX**	M	WB				
`bne $4,$7,L1`								IF	**ID**	EX	M	WB			**RAW $4**

- **Insert in the following pipeline scheme the stalls needed to solve the hazards by marking in GREEN the forwarding paths used:**

Instruction	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Forwarding path
`L1:lw$2,BASEA($4)`	IF	ID	EX	**M**	WB											
`addi $2,$2,INC1`		IF	ID	**EX-S**	**EX**	M	WB									**MEM-EX $2**
`lw $3,BASEB($4)`			IF	**ID-S**	ID	EX	**M**	WB								
`addi $3,$3,INC2`					IF	ID	**EX-S**	**EX**	M	WB						**MEM-EX $3**
`add $5,$2,$3`						IF	**ID-S**	ID	**EX**	M	WB					**EX-EX $3**
`sw $5,BASEC($4)`								IF	ID	**EX**	M	WB				**EX-EX $5**
`addi $4, $4, 4`									IF	ID	**EX**	M	WB			
`bne $4,$7, L1`										IF	**ID-S**	**ID**	EX	M	WB	**EX-ID $4**

- **Indicate synthetically in the first column the number of stalls to be inserted before (or during) each instruction to solve the hazards:**

Num. Stalli	Istruzione	Type of Hazards
	`L1: lw $2,BASEA($4)`	
1	`addi $2,$2,INC1`	RAW $2
	`lw $3,BASEB($4)`	
1	`addi $3,$3,INC2`	RAW $3
	`add $5,$2,$3`	
	`sw $5,BASEC($4)`	
	`addi $4, $4, 4`	
1	`bne $4,$7,L1`	RAW $4

- **Express the formulas, then calculate the following metrics:**
  - Instruction Count per iteration **(IC)** = **8**
  - Number of stalls per iteration = **3**
  - Asymptotic **CPI** (**N** cycles) : **CPI** $_{AS}$**=** **(IC + # stalls) / IC = (8 + 3) / 8 = 1,375**
  - Asymptotic **Throughput** (expressed in **MIPS**) (**N** cycles): **MIPS**$_{AS}$ **=** f_{CLOCK} **/ (CPI**$_{AS}$ *** 10**^{6}**) = (500 * 10**^{6}**) / (1,375 * 10**^{6}**) = 363,6**

EXERCISE 1 (D)

Besides the previous optimizations, we assume the **scheduling** of the assembly program has been optimized as follows:

```
L1:     lw   $2, BASEA ($4)
        lw   $3, BASEB ($4)
        addi $4, $4, 4
        addi $2, $2, INC1
        addi $3, $3, INC2
        add  $5, $2, $3
        bne  $4, $7, L1
        sw   $5, (BASEC-4) ($4)    # branch delay slot
```

- **Draw the pipeline scheme by marking in RED the RAW (Read After Write) data hazards and in BLUE the control hazards still remaining in the program (if any) and adding by GREEN ARROWs the forwarding paths used to solved the hazards:**

Istruzione	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	Forwarding Path
`L1: lw $2,BASEA($4)`	IF	ID	EX	M	WB								
`lw $3,BASEB($4)`		IF	ID	EX	M	WB							
`addi $4, $4, 4`			IF	ID	EX	M	WB						
`addi $2,$2,INC1`				IF	ID	EX	**M**	WB					
`addi $3,$3,INC2`					IF	ID	**EX**	M	WB				
`add $5,$2,$3`						IF	ID	**EX**	M	WB			MEM-EX$2 + EX-EX$3
`bne $4,$7,L1`							IF	ID	EX	M	WB		
`sw $5,BASEC-4($4)`								IF	**ID**	EX	M	WB	EX-ID $5

- **Express the formulas, then calculate the following metrics:**
  - Instruction Count per iteration **(IC)** = **8**
  - Number of stalls per iteration = **0**
  - Asymptotic **CPI** (**N** cycles) : **CPI** $_{AS}$ **= (IC + # stalls) / IC = 1 = CPI** $_{ideal}$
  - Asymptotic **Throughput** (expressed in **MIPS**) (**N** cycles): **MIPS**$_{AS}$ **=** MIPS $_{ideal}$ $= f_{CLOCK} / (CPI_{ideal} * 10^6) = (500 * 10^6) / (1 * 10^6) = 500$

EXERCISE 1 (E)

We assume that, in the previously scheduled and optimized program, each READ access in the **MEM** phase to the data cache generates a **DATA CACHE MISS** requiring **2 stalls** to access the memory:

```
L1:     lw   $2, BASEA ($4)
        lw   $3, BASEB ($4)
        addi $4, $4, 4
        addi $2, $2, INC1
        addi $3, $3, INC2
        add  $5, $2, $3
        bne  $4, $7, L1
        sw   $5, (BASEC-4) ($4)    # branch delay slot
```

- **Draw the pipeline scheme by inserting the stalls due to memory accesses and adding GREEN ARROWs for the forwarding paths used:**

Instruction	C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	C 9	C 10	C 11	C 12	C 13	C 14	C 15	C 16	Forwarding Path
`L1:lw $2,BASEA($4)`	IF	ID	EX	**M-S**	**M-S**	**M**	WB										
`lw $3,BASEB($4)`		IF	ID	**EX-S**	**EX_S**	EX	**M-S**	**M-S**	**M**	WB							
`addi $4, $4, 4`			IF	**ID-D**	**ID_S**	ID	**EX-S**	**EX_S**	EX	M	WB						
`addi $2,$2,INC1`				**IF-S**	**IF-S**	IF	**ID-D**	**ID_S**	ID	EX	**M**	WB					
`addi $3,$3,INC2`							**IF-S**	**IF-S**	IF	ID	**EX**	M	WB				
`add $5,$2,$3`										IF	ID	**EX**	M	WB			**MEM-EX$2 + EX-EX$3**
`bne $4,$7,L1`											IF	ID	EX	M	WB		
`sw $5,BASEC-4($4)`												IF	**ID**	EX	M	WB	**EX-ID $5**

- **Calculate the following metrics:**
  - Instruction Count per iteration **(IC)** = **8**
  - Number of stalls per iteration = **4 for the memory accesses (2 stalls for each data cache miss)**
  - Asymptotic **CPI** (**N** cycles) : **CPI** $_{AS}$ **= (IC + # stalls) / IC = (8 + 4) / 8 = 1,5**
  - Asymptotic **Throughput** (expressed in **MIPS**) (**N** cycles): **MIPS**$_{AS}$ **=** f_{CLOCK} **/ (CPI**$_{AS}$ *** 10**^{6}**) = (500 * 10**^{6}**) / (1,5 * 10**^{6}**) = 333,33**