EXERCISE 1 (H): TOMASULO

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 0.5 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
```

We assume the program be executed on a pipelined CPU with dynamic scheduling based on TOMASULO algorithm with:

- 2 RESERVATION STATIONS (RS1, RS2) + 1 LOAD/STORE FU (LDU1) with latency 2 cycles
- 2 RESERVATION STATIONS (RS3, RS4) + 1 ALU/BR FU (ALU1) with latency 1 cycle
- Check structural hazards for RS in ISSUE phase
- Check RAW hazards and Check structural hazards for FUs in START EXECUTE phase
- WRITE RESULT in RS and RF
- Static branch prediction for backward branches: branch always taken

Please complete the Tomasulo table by assuming all cache hits:

ISTRUZIONE	ISSUE	START EXEC	WRITE RESULT	Hazards Type	RSi	UNIT	OP1	OP2	STORE BUFFER
`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`									

- Instruction Count per iteration **(IC)**:
- **CPI =**

- Calculate the speedup with respect to the first version of Scoreboard (EX 1.F):

Speedup =

EXERCISE 1 (I): TOMASULO

We assume the original program be executed on CPU with dynamic scheduling based on TOMASULO algorithm with:

- 2 RESERVATION STATIONS (RS1, RS2) + 2 LOAD/STORE FU (LDU1, LDU2) with latency 2 cycles
- 2 RESERVATION STATIONS (RS3, RS4) + 2 ALU/BR FU (ALU1, ALU2) with latency 1 cycle
- Check structural hazards for RS in ISSUE phase
- Check RAW hazads and Check structural hazards for FUs in START EXECUTE phase
- WRITE RESULT in RS and RF
- Static branch prediction for backward branches: branch always taken

Please complete the Tomasulo table by assuming all cache hits:

ISTRUZIONE	ISSUE	START EXEC	WRITE RESULT	Hazards Type	RSi	UNIT	OP1	OP2	STORE BUFFER
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									

- Instruction Count per iteration **(IC)**:
- **CPI =**

- Calculate the speedup with respect to the previous version of Tomasulo (EX 1.H):

  Speedup =

EXERCISE 2 (F) - TOMASULO

Given the following loop expressed in a high level language:

```
for (i =0; i < N; i ++)
      vectC[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 N respectively. The symbols VECTA, VECTB and VECTC are 16-bit constant. The processor clock frequency is **2 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 $t2,VECTC($t6)	# VECTC[$t6] <- $t2;
addi $t6,$t6,4	# $t6 <- $t6 + 4;
j FOR1	# goto FOR1;
END:	

We assume the program be executed on a CPU with dynamic scheduling based on TOMASULO algorithm with:

- 2 RESERVATION STATIONS (RS1, RS2) + 1 LOAD/STORE FU (LDU1) with latency 4 cycles
- 2 RESERVATION STATIONS (RS3, RS4) + 1 ALU/BR FU (ALU1) with latency 2 cycles
- Check structural hazards for RS in ISSUE phase
- Check RAW hazards and Check structural hazards for FUs in START EXECUTE phase
- WRITE RESULT in RESERVATION STATIONS and RF
- Static Branch Prediction BTFNT (BACKWARD TAKEN FORWARD NOT TAKEN) with Branch Target Buffer

Please complete the Tomasulo table by assuming all cache hits:

ISTRUZIONE	PRED. T / NT	ISSUE	START EXEC	WRITE RESULTS	HAZARDS TYPE	RSi	UNIT	OP1	OP2	STORE BUFFER
`FOR1:beq $t6,$t7, FOR2`										
`lw $t2,VECTA($t6)`										
`lw $t3,VECTB($t6)`										
`add $t2,$t2,$t3`										
`sw $t2,VECTC($t6)`										
`addi $t6,$t6,4`										
`j FOR1`										

- **Given a clock frequency of 2 GHz, express the FORMULA then calculate the following metrics:**
  - **CPI =**
  - **IPC =**
  - **Throughput** (expressed in **MIPS**):

EXERCISE 2 (G) - TOMASULO

We assume the original program be executed on CPU with dynamic scheduling based on TOMASULO algorithm with:

- 2 RESERVATION STATIONS (RS1, RS2) + 2 LOAD/STORE FU (LDU1, LDU2) with latency 4 cycles
- 2 RESERVATION STATIONS (RS3, RS4) + 2 ALU/BR FU (ALU1, ALU2) with latency 2 cycles
- Check structural hazards for RS in ISSUE phase
- Check RAW hazards and Check structural hazards for FUs in START EXECUTE phase
- WRITE RESULT in RESERVATION STATIONS and RF
- Static Branch Prediction BTFNT (BACKWARD TAKEN FORWARD NOT TAKEN) with Branch Target Buffer

Please complete the Tomasulo table by assuming all cache hits:

ISTRUZIONE	PRED. T / NT	ISSUE	START EXEC	WRITE RESULTS	HAZARDS TYPE	RSi	UNIT	OP1	OP2	STORE BUFFER
`FOR1:beq $t6,$t7, FOR2`										
`lw $t2,VECTA($t6)`										
`lw $t3,VECTB($t6)`										
`add $t2,$t2,$t3`										
`sw $t2,VECTC($t6)`										
`addi $t6,$t6,4`										
`j FOR1`										

- **Given a clock frequency of 2 GHz, express the FORMULA then calculate the following metrics:**
  - **CPI =**
  - **IPC =**
  - **Throughput** (expressed in **MIPS**):

EXERCISE 2 (H) - TOMASULO

We assume the original program be executed on CPU with dynamic scheduling based on TOMASULO algorithm with:

- 2 RESERVATION STATIONS (RS1, RS2) + 2 LOAD/STORE FU (LDU1, LDU2) with latency 4 cycles
- 2 RESERVATION STATIONS (RS3, RS4) + 2 ALU/BR FU (ALU1, ALU2) with latency 2 cycles
- Check structural hazards for RS in ISSUE phase
- Check RAW hazards and Check structural hazards for FUs in START EXECUTE phase
- WRITE RESULT in RESERVATION STATIONS and RF
- Static Branch Prediction BTFNT (BACKWARD TAKEN FORWARD NOT TAKEN) with Branch Target Buffer

Please complete the Tomasulo table by assuming all instruction cache hits and all data cache write hits but all data cache read misses introducing 4 stalls cycles to WRITE RESULTS:

ISTRUZIONE	PRED. T / NT	ISSUE	START EXEC	WRITE RESULTS	HAZARDS TYPE	RSi	UNIT	OP1	OP2	STORE BUFFER
`FOR1:beq $t6,$t7, FOR2`										
`lw $t2,VECTA($t6)`										
`lw $t3,VECTB($t6)`										
`add $t2,$t2,$t3`										
`sw $t2,VECTC($t6)`										
`addi $t6,$t6,4`										
`j FOR1`										

- **Given a clock frequency of 2 GHz, express the FORMULA then calculate the following metrics:**
  - **CPI =**
  - **IPC =**
  - **Throughput** (expressed in **MIPS**):