Surname (COGNOME)	
Name	
POLIMI Personal Code	
Signature	

Politecnico di Milano, 11 September, 2024

Course on Advanced Computer Architectures

Prof. C. Silvano

EX1	(5 points)	
EX2	(5 points)	
EX3	(5 points)	
Q1	(5 points)	
Q2	(5 points)	
QUIZZES	(8 points)	
TOTAL	(33 points)	

EXERCISE 1 – DEPENDENCY ANALYSIS + TOMASULO (5 points)

1. *Let's consider the following assembly code containing multiple types of intra-loop dependences. Complete the following table by inserting all types of true-data-dependences, anti-dependences and output dependences for each instruction:*

I#	TYPE OF INSTRUCTION	ANALYSIS OF DEPENDECES: 1. True data dependence with I# for $Fx 2. Anti-dependence with I# for $Fy 3. Output-dependence with I# for $Fz
I0	`FOR:LD $F2, A($R1)`	None
I1	`FADD $F2, $F2, $F2`	Truedatadependencewith **I0** for $**F2** Output-dependence with I0 for $F2
I2	`FADD $F4, $F4, $F0`	None
I3	`SD $F2, A($R1)`	True data dependence with I1 for $F2
I4	`SD $F4, B($R1)`	True data dependence with I2 for $F4
I5	`ADDUI $R1, $R1, 8`	Anti-dependence with I0, I3, I4 for $R1
I6	`BNE $R1, $R2, FOR`	True data dependence with I5 for $R1

Let's consider the previous assembly code to be executed on a CPU with dynamic scheduling based on **TOMASULO algorithm** with all cache HITS, a single Common Data Bus and:

- *2 RESERV. STATIONS **(RS1, RS2)** with 2 LOAD/STORE units **(LDU1, LDU2)** with latency **4***
- *2 RESERVATION STATION **(RS3, RS4)** with 2 FP unit12 **(FPU1, FPU2)** with latency **3***
- *2 RESERVATION STATION **(RS5, RS6)** with 2 INT_ALU/BR units **(ALU1, ALU2)** with latency **2***

Please complete the following table:

INSTRUCTION	ISSUE	START EXEC	WRITE RESULT	Hazards Type	RSi	UNIT
`FOR:LD $F2, A($R1)`	1	2	6	None	RS1	LDU1
`I1: FADD $F2, $F2, $F2`	2	7	10	RAW $F2 WAW sloved	RS3	FPU1
`I2: FADD $F4, $F4, $F0`	3	4	7	None	RS4	FPU2
`I3: SD $F2, A($R1)`	4	11	15	RAW $F2	RS2	LDU2
`I4: SD $F4, B($R1)`	7	8	12	Structure RS1 nad LDU1 RAW $F4	RS1	LDU1
`I5: ADDUI $R1, $R1, 8`	8	9	11	WAR I0, I3, I4	RS5	ALU1
`I6: BNE $R1, $R2, FOR`	9	12	14	RAW $R1	RS6	ALU2

*Calculate the **CPI** =* ______________________________

15 / 7 = 2.14

EXERCISE 2 – VLIW SCHEDULING (5 points)

Let's consider the following LOOP code, where $**Ri** are integer registers and $**Fi** are floating-point registers.

```
LOOP:  LD $F2, 0 ($R1)
       LD $F4, 0 ($R2)
       FADD $F6, $F2, $F2
       FADD $F8, $F0, $F0
       FADD $F4, $F4, $F6
       FADD $F10, $F0, $F0
       SD $F6, 0 ($R1)
       SD $F4, 0 ($R2)
       SD $F10, 0 ($R3)
       ADDDUI $R1, $R1, 4
       ADDDUI $R2, $R2, 4
       ADDDUI $R3, $R3, 4
       BNE $R1, $R5, LOOP
```

Given a **3-issue VLIW** machine with **fully pipelined functional units:**

- **1 Memory Units with 3 cycles latency**
- **1 FP ALUs with 3 cycles latency**
- **1 Integer ALU with 1 cycle latency to next Int/FP/L/S & 2 cycle latency to next Branch**

The branch is completed with 1 cycle delay slot (branch solved in ID stage). **No branch prediction.** In the Register File, it is possible to read and write at the same address at the same clock cycle. *Considering one iteration of the loop, complete the following table by using the* ***list-based scheduling*** (do NOT introduce any software pipelining, loop unrolling and modifications to loop indexes) on the *4-issue VLIW machine including the BRANCH DELAY SLOT.* Please do not write in NOPs.

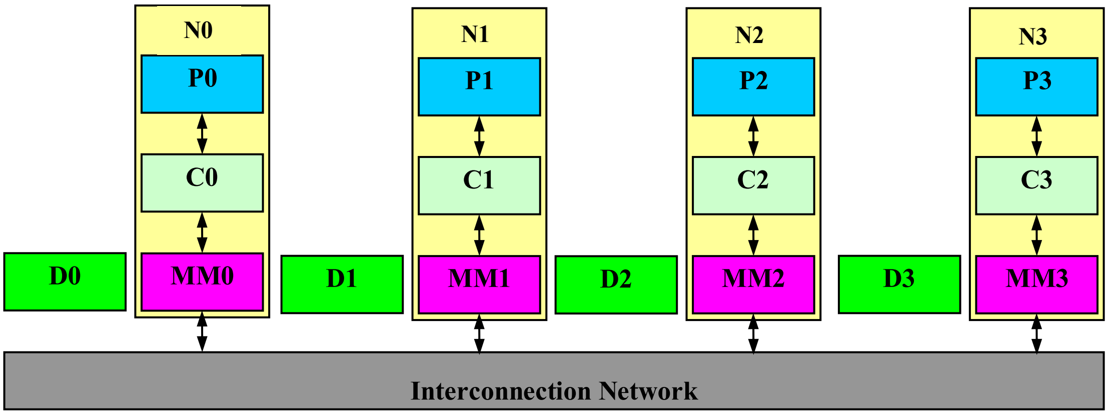
	Memory Unit	Floating Point Unit	Integer Unit
C1	LD $F2, 0 ($R1)		
C2			
C3			
C4			
C5			
C6			
C7			
C8			
C9			
C10			
C11			
C12			
C13			
C14			
C15			

How long is the critical path for a single iteration? ________________________________

How much is the code efficiency for a single iteration? ______________________________

EXERCISE 3: DIRECTORY-BASED PROTOCOL (*5 points)*

Consider a directory-based protocol for a distributed shared memory system with 4 nodes (N0, N1, N2, N3) where **Directory N1 Block B1 | State: Modified | Sharer Bits 1000 |**



After a ***Write Miss on B1 from Node 2,*** *please answer to the following questions:*

Which is the **home node**? What is the definition of **home node**?	
Which is the **local node**? What is the definition of **local node**?	
Which is the **remote node**? What is the definition of **remote node**?	
What is the message sent from the **home node** in reply to the Write Miss?	

What is the **coherence state** of the block B1 in the remote cache?	
What is the next message sent from the remote node to the home node?	
What is the next message sent from the home node to the local node?	
What is the **coherence state** of the block B1 in the Directory of the home node?	
What is the **coherence state** of the block B1in the local cache?	
Which is the **new owner** of the block B1?	

QUESTION 1: BRANCH PREDICTION (5 points)

Let’s consider the STATIC and DYNAMIC branch prediction techniques used in modern microprocessors. *Answer to the following questions:*

	STATIC BRANCH PREDICTION	**DYNAMIC BRANCH PREDICTION**
Explain the main concepts for these two branch prediction techniques.		
What are the main benefits of each branch prediction technique?		

What are the main drawbacks of each branch prediction technique?		
Explain the possible ways how to schedule an instruction in the branch delay slot		
Explain the main concepts of the correlating branch prediction technique		

QUESTION 2: REORDER BUFFER (5 points)

Let's consider the ReOrder Buffer used in modern microprocessors. *Answer to the following questions:*

Explain the main purpose to introduce the ReOrder buffer in a dynamically scheduled processor.	
How can the Reorder Buffer support the speculation in the Tomasulo architecture?	

List the fields of each row entry in a RoB	1.Busy field:
Explain what are the four stages of the Speculative Tomasulo pipelined architecture.	
Explain what happens in a Speculative Tomasulo architecture in the case of a ***branch misprediction***	

QUIZZES

Question 1 (format Multiple Choice – Single answer)

Let's consider a fully associative write-back cache with many cache entries that at cold start is empty and receives the following sequence of 5 memory accesses:

Write Mem[AAAA]
Write Mem[AAAA]
Read Mem[BBBB]
Write Mem[BBBB]
Write Mem[AAAA]

What are the number of cache hits and misses when using a "write allocate" versus a "nowrite allocate" policy?

(SINGLE ANSWER)
1 point

Answer 1: Write allocate has 2 hits & 3 misses | No-write allocate has 1 hit & 4 misses

Answer 2: Write allocate has 3 hits & 2 misses | No-write allocate has 1 hit & 4 misses

Answer 3: Write allocate has 1 hit & 4 misses | No-write allocate has 3 hits & 2 misses

Answer 4: Write allocate has 4 hits & 1 miss | No-write allocate has 1 hit & 4 misses

Answer 5: Write allocate has 1 hit & 4 misses | No-write allocate has 2 hits & 3 misses

Motivate your answer:
1 point

____

____

____

____

Question 2 (format Multiple Choice – Single answer)

Let's consider the following code:

```
    for (i=0; i<255; i++)
    if (X[i] != 0)
    Y[i] = X[i] + Y[i];
```

Which code transformation can be applied to be executed by the VMIPS Vector Processor with a Vector Register File composed of 8 registers of 32 elements and 64 bits/element?

(SINGLE ANSWER)
1 point

Answer 1: Trace scheduling

Answer 2: Software pipeling

Answer 3: Vector strip mining

Answer 4: Vector mask registers

Answer 5: Memory striding

Motivate your answers:
1 point

Question 3 (format Multiple Choice – Multiple answer)

*How does a MESI write-invalidate write-back protocol manage a **Write Hit** on an Exclusive cache block?*

(MULTIPLE ANSWERS)
1 point
Answer 1: The status of the cache block becomes Modified

Answer 2: The cache block is retrieved from memory

Answer 3: An invalidate is broadcasted on the bus to the other copies of the block

Answer 4: The cache block is retrieved from another cache

Answer 5: The cache block is overwritten in the processor's cache

Question 4 (format Multiple Choice – Multiple answer)

*How does a MESI write-invalidate write-back protocol manage a **Write Miss** on a cache block which is Exclusive in another cache?*

(MULTIPLE ANSWERS)
1 point

Answer 1: The status of the cache block becomes Modified

Answer 2: The cache block is retrieved from memory

Answer 3: An invalidate is broadcasted on the bus to the other copies of the block

Answer 4: The cache block is retrieved from another cache

Answer 5: The cache block is overwritten in the processor's cache

Question 5 (format True/False)

To obtain a loop unrolling version of a code, we need to use register renaming if there are some true data dependences in the original code.
(format True/False)
1 point
Answer 1: True / False

Motivate your answer:
1 point