Surname (COGNOME)	
Name	
POLIMI Personal Code	
Signature	

Politecnico di Milano, 15 July, 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`	True data dependence with I0 for $F2 Output-dependence with I0 for $F2
I2	`FADD $F2, $F2, $F4`	True data dependence with I1 for $F2 Output-dependence with I1 for $F2 Anti-dependence with I1 for $F2
I3	`SD $F2, A($R1)`	True data dependence with I2 for $F2
I4	`ADDUI $R1, $R1, 8`	Anti-dependence with I0, I3 for $R1
I5	`BNE $R1, $R2, FOR`	True data dependence with I4 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 1 FP unit1 **(FPU1)** with latency **3***
- *2 RESERVATION STATION **(RS5, RS6)** with 1 INT_ALU/BR unit **(ALU1)** 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 $F2 solved	RS3	FPU1
`I2: FADD $F2, $F2, $F4`	3	11	14	Structuure ALU1 RAW $F2 WAW $F2 solved, WAR $F2 solved	RS4	FPU2
`I3: SD $F2, A($R1)`	4	15	19	RAW $F2	RS2	LDU2
`I4: ADDUI $R1, $R1, 8`	5	6	8	WAR $R1 solved	RS5	ALU1
`I5: BNE $R1, $R2, FOR`	6	9	11	Structure ALU1	RS6	ALU1

*Calculate the **CPI** =* _____

19 / 6 = 3.17

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 $F4, $F4, $F4
       FADD $F6, $F2, $F2
       FADD $F8, $F0, $F0
       SD $F6, 0 ($R1)
       SD $F8, 0 ($R2)
       ADDDUI $R1, $R1, 4
       ADDDUI $R2, $R2, 4
       BNE $R1, $R3, 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 – CACHE MEMORIES (5 points)

Let's consider a 32-block main memory with a **4-way set associative** 8-block cache based on a **write allocate** with **write-back** protocol with **LRU.**

The addresses are expressed as: Memory Addresses: [0, 1, 2, ……. 31] $_{10}$
Cache Addresses: [a, b, c, d, e, f, g, h]
and Cache Tags are expressed as binary numbers.

How many sets?

Answer 1: 2

Answer 2: 4

Answer 3: 8

Answer 3: 16

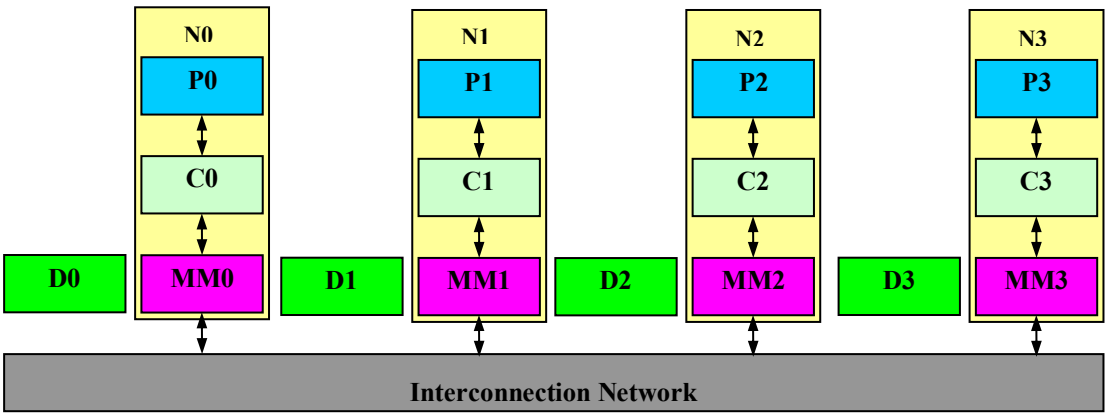
At cold start the cache is empty, then there is the following sequence of memory accesses.
Please complete the following table:

	Type of memory access	Memory Address	Hit/ Miss Type	Cache Tag	Set	Cache Address	Dirty bit	Write in memory
1	Write	[16] $_{10}$	Cold-start Miss	[1000] $_2$	[0] $_{10}$	a	1	No
2	Write	[14] $_{10}$	Cold-start Miss	**0111**	**0**	**b**	**1**	**No**
3	Write	[07] $_{10}$	**Cold-start Miss**	**0011**	**1**	**e**	**1**	**No**
4	Read	[14] $_{10}$	**Hit**	**0111**	**0**	**b**	**1**	**No**
5	Write	[16] $_{10}$	**Hit**	**1000**	**0**	**a**	**1**	**No**
6	Read	[12] $_{10}$	**Cold-start Miss**	**0110**	**0**	**c**	**0**	**No**
7	Write	[18] $_{10}$	**Cold-start Miss**	**1001**	**0**	**d**	**1**	**No**
8	Read	[06] $_{10}$	**Conflict**	**0011**	**0**	**b**	**0**	**14**
9	Write	[04] $_{10}$	**Confict**	**0010**	**0**	**a**	**1**	**Write 16**
10	Write	[30] $_{10}$	**Conflict**	**1111**	**0**	**c**	**1**	**No**

How much is the miss rate? _____

QUESTION 1: DIRECTORY-BASED PROTOCOL (5 points)

Consider a directory-based protocol for a distributed shared memory system with **4 nodes (N0, N1, N2, N3):**



Please answer to the following questions:

What is the definition of **home node**?	
What is the definition of **local node**?	
What is the definition of **remote node**?	
What are the possible messages sent from the **local node** to the **home node**?	
What are the possible messages sent from the **home node** to the **local node?**	

What are the possible messages sent from the **home node** to the **remote node?**	_____ _____ _____ _____ _____ _____ _____
What are the possible messages sent from the **remote node** to the **home node?**	_____ _____ _____ _____ _____
What are the possible **coherence states** of a block in the local cache?	1) _____ 2) _____ 3) _____
What are the possible **coherence states** of a block in the home directory?	1) _____ 2) _____ 3) _____
What is the meaning of the **sharer bits** of a block in the home directory?	_____ _____ _____ _____

QUESTION 2: MULTITHREADING (5 points)

Let's consider the fine-grained and simultaneous multithreading techniques used to manage thread-level parallelism by hardware processors. *Answer to the following questions:*

	Fine-grained Multithreading	**Simultaneous Multithreading**
*Explain the **main concepts** for each technique.*		
*For each technique, make **an example** with up to 4 threads (T1, T2, T3, T4) on a quad-issue processor*		
Let's consider a quad-issue SMT processor that can manage up to 4 threads		*How much is the **ideal CPI?*** *How much is the **ideal per-thread CPI?***

*What are the main **benefits** for each technique?*		
*What are the main **drawbacks** for each technique?*		

QUIZZES

Question 1 (format Multiple Choice – Single answer)

Let's consider the following loop:

```
for (i=1; i<=100, i++) {
      X[i] = X[i] + X[i-1];     /*S1*/
      Z[i] = X[i] + Y[i-1]      /*S2*/
}
```

How many loop-carried dependencies are in the code?
(SINGLE ANSWER)
1 point

Answer 1: Only one in S2 because Z[i] depends on Y[i-1];

Answer 2: Only one in S1 because X[i] depends on X[i-1];

Answer 3: Both of them

Question 2 (format Multiple Choice – Single answer)

Let's consider the following memory hierarchy addressed at word-level (32-bit):
- main memory size = 4 Giga word
- 4-way set-associative cache with cache size = 1 Mega word and block size = 256 word

Please indicate the structure of the 32-bit memory address:

(SINGLE ANSWER)
1 point

Answer 1: |12-bit tag | 12-bit index | 8-bit offset |

Answer 2: | 24-bit tag | 8-bit offset |

Answer 3: |14-bit tag | 10-bit index | 8-bit offset |

Answer 4: |13-bit tag | 11-bit index | 8-bit offset |

Question 3 (format Multiple Choice – Single answer)

Let's consider a dual-issue SMT processor that can manage up to 4 simultaneous threads.
What are the values of the ideal CPI and the ideal per-thread CPI?

(SINGLE ANSWER)
1 point

Answer 1: Ideal CPI = 1 & Ideal per-thread CPI = 0.25

Answer 2: Ideal CPI = 0.25 & Ideal per-thread CPI =0.25

Answer 3: Ideal CPI = 0.5 & Ideal per-thread CPI = 2

Answer 4: Ideal CPI = 0.5 & Ideal per-thread CPI = 1

Answer 5: Ideal CPI = 0.25 & Ideal per-thread CPI = 4

Question 4 (format Multiple Choice – Single answer)

Let's consider a directory-based protocol for a distributed shared memory system with 4 Nodes (N0, N1, N2, N3) where: **Directory N1 Block B1 | State: Shared | Sharer Bits: 1001**
Which is the state of the block B1 in N1 after this sequence:
Read Miss B1 from local cache N1;
Read Miss B1 from local cache N2;
Write Hit B1 from local cache N1 ;

(SINGLE ANSWER)
1 point

Answer 1: Directory N1 Block B1 | State: Shared | Sharer Bits: 1111

Answer 2: Directory N1 Block B1 | State: Modified | Sharer Bits: 0100

Answer 3: Directory N1 Block B1 | State: Shared | Sharer Bits: 0110

Answer 4: Directory N1 Block B1 | State: Modified | Sharer Bits: 0110

Question 5 (format Multiple Choice – Single answer)

Let's consider the following code:

```
for (i=0; i<260; i++)
Y[i] = X[i] + Y[i];
```

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

(SINGLE ANSWER)
1 point

Answer 1: Trace scheduling

Answer 2: Software pipeling

Answer 3: Strip mining

Answer 4: Apply a mask register

Motivate your answers:
1 point

Question 6 (format Multiple Choice – Single answer)

Let's consider the following code executed by a Vector Processor with:

- *Vector Register File composed of 32 vectors of 16 elements per 64 bits/element;*
- *Scalar FP Register File composed of 32 registers of 64 bits;*
- *One Load/Store Vector Unit with operation chaining and memory bandwidth 64 bits;*
- *One ADD/SUB Vector Unit with operation chaining.*
- *One MUL/DIV Vector Unit with operation chaining.*

```
L.V V1, R1              # Load vector from mem. address R1 into V1
L.V V3, R2              # Load vector from mem. address R2 into V3
ADDVS.D V2, V1, S1      # FP add vector V1 to scalar S1
MULVS.D V2, V2, S2      # FP multiply vector V2 to scalar S2
ADDVS.D V3, V3, S3      # FP add vector V3 to scalar S3
S.V V2, R1              # Store vector V2 into memory address R1
S.V V3, R2              # Store vector V3 into memory address R2
```

How many convoys? How many clock cycles to execute the code?

(SINGLE ANSWER)
1 point

Answer 1: 3 convoys; 48 clock cycles

Answer 2: 2 convoys; 32 clock cycles

Answer 3: 4 convoys; 64 clock cycles

Motivate your answer by completing the following figure:
1 point

0	*1*	*… L.V V1, R1 …*	*15*