Dipartimento di Elettronica e Informazione

Politecnico di Milano

20133 Milano (Italia)
Piazza Leonardo da Vinci, 32
Tel. (+39) 02-2399.3400
Fax (+39) 02-2399.3411

Advanced Computer Architecture

June 29, 2022

Prof. Donatella Sciuto and Marco Santambrogio

Name
Last Name
Professor:

You have 90’ to complete the exam

Problem 1 (6pts)	
Problem 2 (6pts)	
Problem 3 (6pts)	
Question 1 (5pts)	
Question 2 (5pts)	
Question 3 (5pts)	
Total (32pts)	

To pass the exam (the following conditions are in AND):

- The overall score has to be equal or higher than 18
- At least 1 problem and 1 Question has to be correct

Problem 1

Design and describe (all the decisions have to be motivated) a 1-BHT and a 2-BHT able to execute the following assembly code.
(R0 is set to 100, R1 is set to 0)

```
LOOP:   LD      F3   0     R0
        ADDD    F1   F3    F3
        MULTD   F2   F3    F1
        ADDI    R1   R1    1000
LOOP2:  LD      F3   0     R1
        MULTD   F2   F2    F3
        SUBI    R1   R1    10
        BNEZ    R1   LOOP2
        SUBI    R0   R0    10
        BNE     R0   R1    LOOP
```

The obtained result, in terms of miss predictions, is inline with theoretical characteristics of the two predictors? Please effectively support your answer.

Answer

LOOP 10 times, LOOP2 steady 100

Worst case:
1BHT, not colliding initialized with NT and NT. 1+1 misprediction for LOOP and (1+1) * 10 for LOOP2. Total of 22 miss.

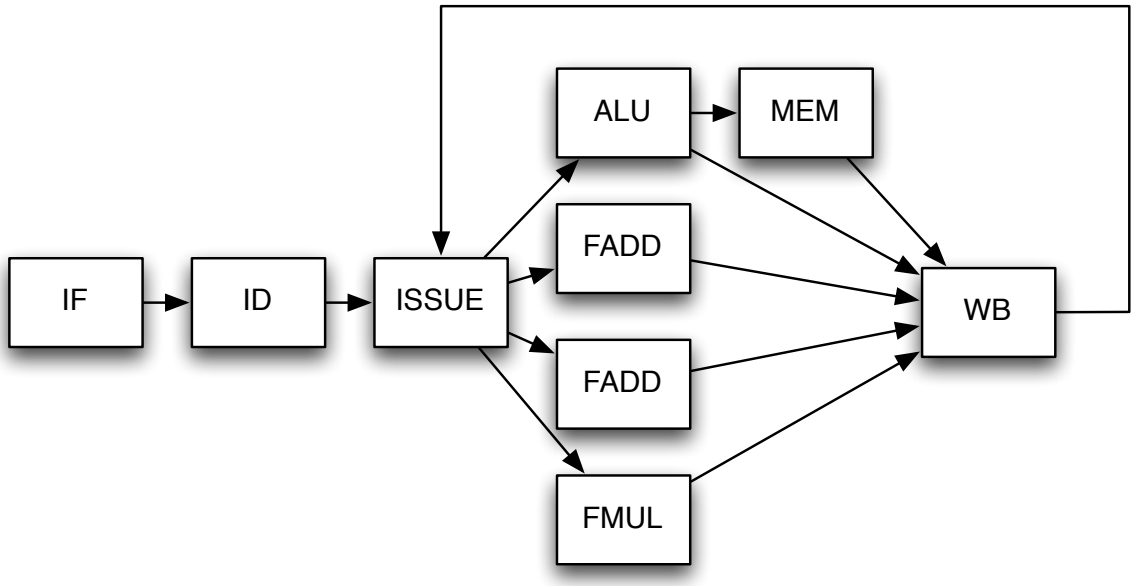
2BHT, NT strong and NT strong. 2 + 1 misprediction@LOOP and 2 + (1)*10@LOOP2. Total of 15 miss.

Best case:
1BHT. T and T: 1@LOOP and 1 + 2*9@LOOP2. Total of 20 miss.

Best of 1BHT is worst than worst-case of 2BHT

Problem 2

In this problem we will examine the execution of a code segment on the following single-issue out-of-order processor:



You can assume that:

- All functional units are pipelined
- ALU operations take 1 cycle
- Memory operations take 3 cycles (includes time in ALU)
- Floating-point add instructions take 3 cycles
- Floating-point multiply instructions take 5 cycles
- There is no register renaming
- Instructions are fetched, decoded and issued in order
- The issue stage is a buffer of unlimited length that holds instructions waiting to start execution
- An instruction will only enter the issue stage if it does not cause a WAR or WAW hazard
- Only one instruction can be issued at a time, and in the case multiple instructions are ready, the oldest one will go first

To simplify book-keeping for this problem we will only track instructions during a few critical stages:

- I - When an instruction enters the issue stage
- E - When an instruction starts execution (enters FU)
- C - When an instruction completes execution (enters WB, results available this cycle)

For this problem we will be using the following code:

```
I1      L.D    F2, 0(R1)
I2      ADD.D  F1, F1, F2
I3      MUL.D  F4, F3, F3
I4      ADDI   R1, R1, 8
I5      L.D    F2, 0(R1)
I6      ADD.D  F1, F2, F4
```

Fill in the following table to indicate how the given code will execute. Assume all register values are available at the start of execution.

Inst \ Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I_1																				
I_2																				
I_3																				
I_4																				
I_5																				
I_6																				

Problem 3

Consider the following access pattern on a two-processor system with a direct-mapped, write-back cache with one cache block and a two cache block memory. Assume the MESI protocol is used, with write-back caches, write-allocate, and invalidation of other caches on write (instead of updating the value in the other caches).

Time	After Operation	P1 cache block state	P2 cache block state	Memory at block 0 up to date?	Memory at block 1 up to date?
0	P1: read block 0				
1	P2: write block 1				
2	P1: write block 0				
3	P2: read block 1				
4	P1: read block 1				
5	P2 : write block 1				
6	P1: read block 0				
7	P2: write block 1				
8	P1: read block 0				
9	P2: read block 0				

Question 1

The following 5 questions are True/False.
Each of them will count 1 point

Question 1.1

A superscalar architecture can let instructions from different threads to be executed at the same clock cycle.
Given the previous statement, confirm if it is TRUE or FALSE and **effectively support** your answer.

Circle the **right** answer: True False

Question 1.2

The ARM big.LITTLE architecture, considering the fact that is a multicore architecture where the cores are sharing the same ISA, cannot be considered a MIMD architecture but a SIMD one.

Circle the **right** answer: True False

Question 1.3

A physically centralized memory can be used to implement a shared-memory architecture

Circle the **right** answer: True False

Question 1.4

A VLIW Architecture has to have multiple Program Counters to load the necessary Multiple Data

Circle the **right** answer: True False

Question 1.5

In a 5-stage pipeline MIPS architecture, it is possible to forward a data twice from stage-i to stage-j.
As an example, given the following code:

I1 ADDD F2, F4, F6
I2 ADDD F4, F2, F26
I3 ADDD F6, F2, F4

F2 will be forwarder from I1 to both I2 and I3 by using the EX->EX forwarding path

Circle the **right** answer: True False

Question 2

Explain the difference among fine-grained multithreading, coarse-grained multithreading and simultaneous multithreading.

Question 3

1. Explain the main differences between a VLIW architecture and a superscalar architecture, in terms of instructions scheduling, hardware complexity, and code size.
2. In real superscalar processors, some architectural limits do not allow fully exploitation of the ILP present in the program. Specify these architectural limits and specify how they impact on the potential ILP

1

R0=100 R1=0

```
LOOP:   LD      F3   0    R0
        ADDD    F1   F3   F3
        MULTD   F2   F3   F1
        ADDI    R1   R1   1000
LOOP2:  LD      F3   0    R1
        MULTD   F2   F2   F3
        SUBI    R1   R1   10
        BNEZ    R1   LOOP2
        SUBI    R0   R0   10
        BNE     R0   R1   LOOP
```

100 ITERATIONS

10 ITERATIONS

1-BHT NO COLLISION

```
LOOP:   LD      F3   0    R0
        ADDD    F1   F3   F3
        MULTD   F2   F3   F1
        ADDI    R1   R1   1000
LOOP2:  LD      F3   0    R1
        MULTD   F2   F2   F3
        SUBI    R1   R1   10
T       BNEZ    R1   LOOP2
        SUBI    R0   R0   10
T       BNE     R0   R1   LOOP
```

T→NT NT→T T→NT

$1 + (1+1)\cdot(10-1) +$

$+ \ 1 = 20$

WITH COLLISION T, WE WOULD HAVE GET

T→NT NT→T T→NT

$1 + (1+1)\cdot(10-1) = 19$

2-BHT NO COLLISION

```
LOOP:   LD      F3   0    R0
        ADDD    F1   F3   F3
        MULTD   F2   F3   F1
        ADDI    R1   R1   1000
LOOP2:  LD      F3   0    R1
        MULTD   F2   F2   F3
        SUBI    R1   R1   10
NT,T    BNEZ    R1   LOOP2
        SUBI    R0   R0   10
NT,T    BNE     R0   R1   LOOP
```

$T_{2} \rightarrow T_{w}$ $T_{2} \rightarrow T_{w}$ LAST

$1 \cdot 10 + 1 = 11$

T→NT $T_{2} \rightarrow T_{w}$ T→NT $T_{2} \rightarrow T_{w}$

$2 + 1 \cdot 10 + 2 + 1 = 15$

RESULTS MAKE SENSE, AS WORST 2-BHT OUTPERFORMS 1-BHT'S BEST CASE

2

I1	L.D	F2, 0(R1)
I2	ADD.D	F1, F1, F2
I3	MUL.D	F4, F3, F3
I4	ADDI	R1, R1, 8
I5	L.D	F2, 0(R1)
I6	ADD.D	F1, F2, F4

ALU 1cc FP+ 3cc
MEM 3cc FP. 5cc

RAW F2 11-12

WAW F2 11-15

WAR F2 12-15 RAW F2 15-16

WAW F1 12-16 WAR F1 12-16 RAW F4 13-16

WAR R1 11-14 RAW R1 14-15

Inst \ Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I_1	F	D	I	E	E	E	W													
I_2		F	D	↓	↓	↓	I	E	E	E	W									
I_3			F	D	↓	↓	↓	I	E	E	E	E	E	W						
I_4				F	↓	D	↓	↓	I	E	↓	W								
I_5						F	↓	↓	↓	D	↓	I	E	E	E	W				
I_6										F	D	↓	↓	↓	↓	I	E	E	E	W

3

Time	After Operation	P1 cache block state	P2 cache block state	Memory at block 0 up to date?	Memory at block 1 up to date?
0	P1: read block 0	E(0)	I	YES	YES
1	P2: write block 1	E(0)	M(1)	YES	NO
2	P1: write block 0	M(0)	M(1)	NO	NO
3	P2: read block 1	M(0)	M(1)	NO	NO
4	P1: read block 1	S(1)	S(1)	YES	YES
5	P2 : write block 1	I	M(1)	YES	NO
6	P1: read block 0	E(0)	M(1)	YES	NO
7	P2: write block 1	E(0)	M(1)	YES	NO
8	P1: read block 0	E(0)	M(1)	YES	NO
9	P2: read block 0	S(0)	S(0)	YES	YES

THEORY 1

Question 1.1
A superscalar architecture can let instructions from different threads to be executed at the same clock cycle.
Given the previous statement, confirm if it is TRUE or FALSE and **effectively support** your answer.

Circle the **right** answer: ✓ True False

THANKS TO EXPLOITATION OF MULTIPLE P.C.

Question 1.2
The ARM big.LITTLE architecture, considering the fact that is a multicore architecture where the cores are sharing the same ISA, cannot be considered a MIMD architecture but a SIMD one.

Circle the **right** answer: True ✓ False

DIFFERENT P.C.

Question 1.3
A physically centralized memory can be used to implement a shared-memory architecture

Circle the **right** answer: ✓ True False

Question 1.4
A VLIW Architecture has to have multiple Program Counters to load the necessary Multiple Data

Circle the **right** answer: True ✓ False

VLIW, BY DEFINITION, IS A SINGLE INSTRUCTION ARCHITECTURE

Question 1.5
In a 5-stage pipeline MIPS architecture, it is possible to forward a data twice from stage-i to stage-j.
As an example, given the following code:

I1 ADDD F2, F4, F6
I2 ADDD F4, F2, F26
I3 ADDD F6, F2, F4

F2 will be forwarder from I1 to both I2 and I3 by using the EX->EX forwarding path

Circle the **right** answer: True ✓ False

E→E TO I3 LEADS TO RECEIVE F4

THEORY 2

FINE GRAINED PARALLELISM

SWITCH BETWEEN THREADS AT EACH INSTRUCTIONS

- HIDE LONG STALLS BUT SLOWS OVERALL EXECUTION

COARSE GRAINED MULTITHREADING

SWITCH BETWEEN THREADS WHEN LONG STALLS OCCUR

- OVERALL EXECUTION NOT AFFECTED, BUT FOR SHORT STALLS NOTHING IS MANAGED

SIMULTANEOUS MULTITHREADING

DYNAMICALLY ADAPT AND "LEARN" WHEN TO STALL A THREAD, START ANOTHER ONE AND MOVE BACK TO THE "WAITING LIST THREADS"

3

VLIW	SUPERSCALAR
STATIC SCHEDULING	DYNAMIC SCHEDULING
IN-ORDER ISSUE, PARALLEL EXECUTION	OUT-OF-ORDER ISSUE/EXECUTION
COMPILE-TIME MANAGEMENT	DYNAMIC, RUN-TIME MANAGEMENT
HIGHER COMPLEXITY BECAUSE OF CONTINUOUS ADAPTATION	
LONGER CODE, AS INSTRUCTIONS INCLUDES MULTIPLE OPERATIONS AND NOP WHERE "NOTHING HAPPENS" THAT MAKES CODE HEAVIER	SIMPLER, PARALLELIZATION HANDLED AT RUNTIME

REAL WORLD SUPERSCALAR ARCHITECTURES ARE LIMITED BY:

- FINITE NUMBER OF F.U.
- RAW, WAR, WAW HAZARDS
- BRANCH PREDICTION AND CONTROL HAZARDS
- LIMITED INSTRUCTION WINDOW SIZES