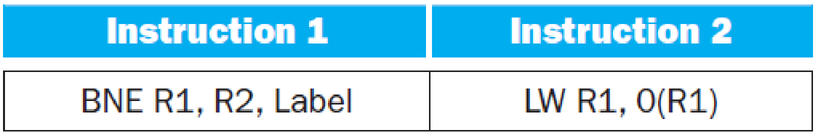
Cairo University | **Faculty of Engineering – Credit hour system**
Computer Engineering Department | **CMP N 301 – Computer Architecture**

Sheet (4)
Chapter 4 – Exception & ILP

1) [4.17] This exercise explores how exception handling affects pipeline design. The problems in this exercise refer to the following two instructions:



Instruction 1	Instruction 2
BNE R1, R2, Label	LW R1, 0(R1)

a. Which exceptions can each of these instructions trigger? For each of these exceptions, specify the pipeline stage in which it is detected.

b. If there is a separate handler address for each exception, show how the pipeline organization must be changed to be able to handle this exception. You can assume that the addresses of these handlers are known when the processor is designed.

c. If the second instruction is fetched right after the first instruction, describe what happens in the pipeline when the first instruction causes the first exception you listed in (a). Show the pipeline execution diagram from the time the first instruction is fetched until the time the first instruction of the exception handler is completed.

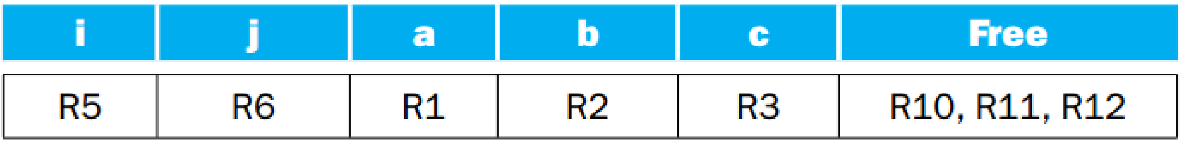
2) [4.18] In this exercise we compare the performance of 1-issue and 2-issue processors, taking into account program transformations that can be made to optimize for 2-issue execution. Problems in this exercise refer to the following loop (written in C):

```
for( i=0; i!=j; i+=2)
        b[i]=a[i]–a[i+1];
```

MIPS Code:

```
 1.  Loop: BEQ   R5, R6 , Finish
 2.  ADD   R10, R5,R1  ;R10 =&a[i]
 3.  LW    R11, 0(R10)
 4.  LW    R10, 1(R10)
 5.  SUB   R11, R11,R10
 6.  ADD   R12,R5,R2
 7.  SW    R11 ,0(R12)
 8.  ADDI  R5,R5,2
 9.  BEQ   R0,R0, Loop
10.  Finish: …
```

When writing MIPS code, assume that variables are kept in registers as follows, and that all registers except those indicated as Free are used to keep various variables, so they cannot be used for anything else.



i	j	a	b	c	Free
R5	R6	R1	R2	R3	R10, R11, R12

The Processor with 5-stages pipeline, full-forwarding, has 2 Parallel ALU Unit, and only One Memory Unit, Branch Executed in Execution stage. **Assuming that the compiler is the one responsible for generating the packets.**

a. Assume the processor has perfect branch prediction and can fetch two instructions in the same cycle. If the loop exits after executing only two iterations,

   i. Write the packets generated by the compilers. Don't reorder, add no operation whenever necessary. Take care that the within the same packet there should be no hazard.

   ii. draw a pipeline diagram for your packets executed on a 2-issue processor.

b. Rearrange/Rename your code to achieve better performance on a 2-issue statically scheduled processor.

c. Repeat a.i), but this time use your MIPS code from (b).

d. Try to use loop unrolling, how much speedup?

e. What is the speedup of going from a 1-issue processor to a 2-issue processor? Use your code from (a) for both 1-issue and 2-issue, and assume that 1,000,000 iterations of the loop are executed. assume that the processor has perfect branch predictions, and that a 2-issue processor can fetch any two instructions in the same cycle.