HIGH PERFORMANCE COMPUTER ARCHITECTURE midterm 31-10-2018 **MATR.NO.**_____

SURNAME_____

FIRST NAME_____

1) (POINTS 25/30)
Consider the following snippet of code running on 4-ways out-of-order superscalar processor.
Initially, R1=0x1000, R3=0x3000, R7=0x0003 and the other registers contain zero.

```
lab1:   LW   R2,0(R1)
        MUL  R4,R2,R2
        SW   R4,0(R1)
        ADDI R1,R1,4
        BNE  R2,R0,lab1
```

Working hypothesis:
* the fetch, decode and commit stages are 4 instructions wide
* the instruction window has 8 slots
* we have 8 physical registers in the free pool
* the reorder buffer has 10 entries
* the integer multiplier has 4 stages
* the load/store queues have 3 slots each and a common effective-address calculation unit
* there are 4 ALUs for arithmetic and logic operations and for branching
* an ALU/BRANCH performs its operation in the same cycle when the operation is issued
* reads require 1 clock cycle (after the addressing phase)
* the register file has 4 input- and 4 output-ports
* there are 9 logical registers (including R0 which is hardwired to 0)
* the store operation leaves the issue stage as it is inserted in the store queue
* when the X stage finds a free slot in the LQ/SQ, also a slot in the I stage is reserved
* branches are predicted taken

In order to calculate the total cycles needed to execute three iterations of the above loop on such machine, complete the following chart until the end of the THIRD iteration of the code fragment above, including the renamed stream the precise evolution of the free pool of the physical registers (the register map), the Instruction Window (IW), the Reorder Buffer (ROB) and the Load Queue (LQ) and Store Queue (SQ).
Note: Ci, Cj, Ck are the cycles when the corresponding physical registers are available in IW.

Iter.	Instruction	F Fetch (clock)	D Decode (clock)	P disPatch (clock)	I Issue (clock)	X eXecute (clock)	W Write-back (clock)	C Commit (clock)	Renamed Stream	Instruction Pi Pj Pk	Window I Ci Cj Ck	Reorder Buffer Ri Pi,old Cplt	Load Queue	Store Queue
1	**LW R2,0(R1)**	0							P2,0(P1)	P2 P1 -	0 0 0	R2 - -		
…	…													
…	…													

2) (POINTS 5/30) On a Linux system, write the SINGLE command line to perform at the BASH shell prompt the following operation (please note that no intermediate files should be used:
- The file ‘data1.txt’ contains a list of alpha-numerical values to be used as input
- The file ‘data2.txt’ should contain a list of the lines which start with “al”
- The extracted list should be directed to a file data3.txt

EXERCIZE 1

```
================================================================================================================================================================
PHYSICAL REGS:  1  2  3  4  5  6  7  8
                *  *                 *
          qi:   0  0  1  1  1  1  1  0
          vi:  00 04 00 04 00 00 00 00
================================================================================================================================================================
REG.FILE: Ri:         1        2        3        4        5        6        7        8
          Pi:         2        8        -        1        -        -        -        -
          Qi:         0        0        0        0        0        0        0        0
          Vi:  00001000 00000000 00003000 00000000 00000000 00000000 00000003 00000000
================================================================================================================================================================
STAGES:                 F  D  P  I  X  W  C RENAMED-STR   INSTRUCTION-WINDOW                  REORDER-BUFFER          A  M  L  S  B  F  X
TOTAL SLOTS:            4  4  8  4 12  4  4 8            8                                   10                      4  1  1  0  1  4  1
BUSY SLOTS:             0  0  0  0  0  1  0 3            0                                   0                       0  0  0  0  0  0  0
STALLS:                 0  8  1  8  0  0 10 8            0                                   1                       0  0  0  2  0  0  0
================================================================================================================================================================
PC   INSTRUCTION        F  D  P  I  X  W  C Pi,Pj Pk Pl   IW#  OPCD Pi  Pj  Pk I/Pl  Cj Ck Cl  ROB# PC  Ri  oPi x s c  +--------------------+
000] LW   R2,0(R1)      0  1  2  3  4  6  7 P2,0(P1)     ----   LW P2  P1  -   0    2  -  -   ---- 000 R2  -   0 0 1  |LQ(0 )              |
001] MUL  R4,R2,R2      0  1  2  6  6 11 12 P3,P2,P2     ----  MUL P3  P2  P2  -    6  6  -   ---- 001 R4  -   0 0 1  |PC   OP Pi  EFAD Ci|
002] SW   R4,0(R1)      0  1  2  4  5 11 12 P3,0(P1)     ----   SW -   P3  P1  0    -  2  -   ---- 002 -   -   1 0 1  |---- LW P2  1000  6|
003] ADDI R1,R1,4       0  1  2  3  3  4 12 P4,P1,4      ---- ADDI P4  P1  -   4    2  -  -   ---- 003 R1  P1  0 0 1  |---- LW P5  1004  8|
004] BNE  R2,R0,-5      1  2  3  4  4  5 12 P2,P0,-5     ----  BNE -   P2  P0  -5   -  3  -   ---- 004 -   -   0 0 1  |---- LW P8  1008 11|
005] LW   R2,0(R1)      2  3  4  5  6  8 13 P5,0(P4)     ----   LW P5  P4  -   0    4  -  -   ---- 000 R2  P2  0 0 1  +--------------------+
006] MUL  R4,R2,R2      2  3  4  8  8 13 14 P6,P5,P5     ----  MUL P6  P5  P5  -    8  8  -   ---- 001 R4  P3  0 0 1
007] SW   R4,0(R1)      2  3  4  6  7 13 14 P6,0(P4)     ----   SW -   P6  P4  0    -  4  -   ---- 002 -   -   1 0 1  +--------------------+
008] ADDI R1,R1,4       2  3  4  5  5  6 14 P7,P4,4      ---- ADDI P7  P4  -   4    4  -  -   ---- 003 R1  P4  0 0 1  |SQ(0 )              |
009] BNE  R2,R0,-5      3  4  5  6  6  7 14 P5,P0,-5     ----  BNE -   P5  P0  -5   -  5  -   ---- 004 -   -   0 0 1  |PC   OP Pi  EFAD Cl|
010] LW   R2,0(R1)      4  5  7  8  9 11 15 P8,0(P7)     ----   LW P8  P7  -   0    7  -  -   ---- 000 R2  P5  0 0 1  |---- SW P0  1000 11|
011] MUL  R4,R2,R2      4 12 13 14 14 19 20 P1,P8,P8     ----  MUL P1  P8  P8  -   13 13  -   ---- 001 R4  P6  0 0 1  |---- SW P0  1004 13|
012] SW   R4,0(R1)      4 12 13 14 15 19 20 P1,0(P7)     ----   SW -   P1  P7  0    - 13  -   ---- 002 -   -   1 0 1  |---- SW P0  1000 19|
013] ADDI R1,R1,4       4 13 14 15 15 16 20 P2,P7,4      ---- ADDI P2  P7  -   4   14  -  -   ---- 003 R1  P7  0 0 1  +--------------------+
014] BNE  R2,R0,-5      5 13 14 15 15 16 20 P8,P0,-5     ----  BNE -   P8  P0  -5   - 14  -   ---- 004 -   -   0 0 1
```

Therefore 21 cycles are needed for this configuration.

EXERCIZE 2

The requested command line is:

```
grep "^al" data1.txt  |  tee data2.tx  > data3.txt
```