Tomasulo Programming Project

Qingyuan Liu

The project simulates of the application of Tomasulo's algorithm for scheduling a block of MIPS-type Load/Store/Arithmetic instructions.

Tomasulo's algorithm

Since the scoreboard algorithm can only detect competition (WAR, WAW) and cannot eliminate these two kinds of competition, the scoreboard algorithm is improved to the Tomasulo algorithm.

Tomasulo algorithm adopts the register renaming method, replacing the registers in the scoreboard with a large set of virtual register names, that is, using a virtual register set to replace the real FP register set, because the virtual register set contains a huge number of registers Because of the real register set, you can use the virtual register set to realize register renaming.

The virtual register group consists of three parts:

1. Each functional unit (FU) has a reservation station
2. Load buffers-save the data and address of the accessed storage unit
3. Store buffers

The floating-point register (FP) is connected to each functional unit (FU) through a pair of buses. This pair of buses correspond to two operands and is connected to the load buffer through a bus.

The output of the functional unit (FU) and the output of the load buffer (Load Buffer) are summarized in the CDB and connected to the input of the floating-point register (FP).

The common data bus (Common data bus) CDB: CDB is connected to the input of FP, Reservation station, store buffer, etc. The only non-connection relationship is the input of the Load Buffer.

Since the reserved station and the buffer have corresponding identifiers, renaming is implemented here.

The instructions are sent from the instruction unit to the instruction queue, and then the instructions are issued from it in a first-in-first-out (FIFO) order.

The reserved station includes operands and actual operands, as well as information for detecting and resolving hazards. The load buffer has three functions: (1) keep the components of the effective address until it is calculated; (2) track the unfinished load waiting in the memory; (3) keep the result of the completed load waiting for the CDB.

The storage buffer has three functions: (1) Reserve the components of the effective address until it is calculated; (2) Reserve the target memory address of the unfinished storage waiting for the data value storage; (3) Reserve the component of the effective address. The address and value to be stored until the storage unit is available. All results of the FP unit or load unit are placed on the CDB, which enters the FP register file and the reserved station and storage buffer. The FP adder implements addition and subtraction, while the FP multiplier performs multiplication and division.

An instruction goes through three steps—issue, execute and write back.

Issue:

If it is an FP operation instruction, if the reserved station is available, it will be sent to the reserved station. If the operand of the instruction is already in the FP register, the value of the operand will be sent to the reserved station.

If it is a Load or Store instruction, if the Buffer is free, send it to the corresponding Buffer

If the reserved station or Buffer is not empty, there is a structural competition. Stop the instruction until the corresponding reserved station or Buffer is available.

This level is renamed because the operand in the reserved station is no longer using the register number.

Execute:

If one or several operands are not ready, wait for the operand and monitor CDB at the same time. Once the operand is ready, it is immediately stored in the corresponding reservation station. If both operands are ready, the operation is performed. This level checks whether there is RAW competition.

Write back:

After the result is calculated, it is written into CDB, and from CDB into the destination register and the reserved station waiting for the result. When the same register is written continuously, only the last time can be written, eliminating WAW competition.

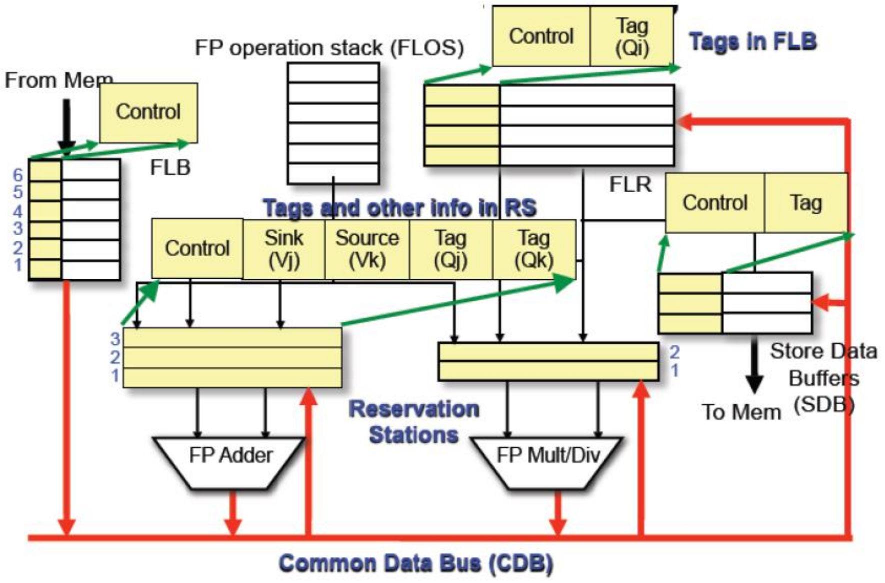


Fig1. IBM 360/91 FPU w/ Tomasulo Algorithm

The floating-point processing components of the Tomasulo algorithm include:

Reserved station (floating-point adder has 3 reserved stations: ADD1, ADD2, ADD3; floating-point multiplier has two reserved stations: MULT1, MULT2), common data bus CDB, load, and storage, floating-point register FP, instruction Axial, arithmetic components (floating-point adder and floating-point multiplier)

Structure of the reservation station:

Each reserved station has the following 6 fields:

① Op: The operation to be performed on the source operand.

② Qj, Qk: The reserved station number that will generate the source operand. Equal to 0 means that the operand is ready and in Vj or Vk, or no operand is needed.

③ Vj, Vk: the value of the source operand. For each operand, only one of the V or Q fields is valid. For load, the Vk field is used to store the offset.

④ Busy: "1" means the reserved station or buffer unit is "busy".

⑤ A: Only load and store buffers have this field. The beginning is to store the immediate number field in the instruction, and the effective address is stored after the address is calculated.

⑥ Qi: register status table. Each register has a corresponding entry in the table, which is used to store the station number of the reserved station that will write the result into the register. 0 means that there is no instruction currently being executed to write to this register, that is, the contents of this register are ready.

Compared with the scoreboard method, the Tomasulo algorithm does not have the function of checking WAW and WAR contention, because in the process of command transmission, the issue logic combined with the reservation station completes the rename process of register operands, which eliminates these two types of competition. CDB plays the role of broadcasting the result, without directly sending the result to all reservation stations and buffers that need the result through the register file. Load and Store buffers are equivalent to the basic functional unit.

Programming simulation

1 The programming language is C++.

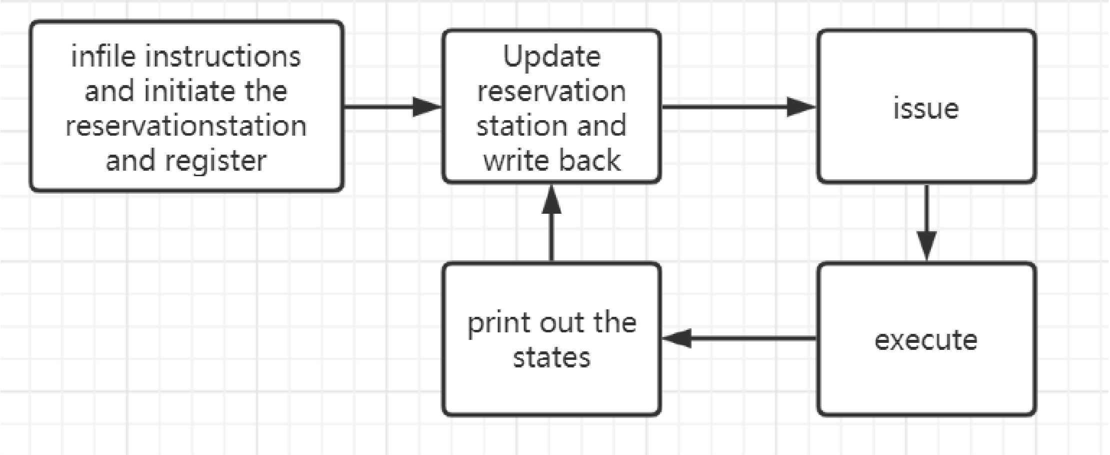
2 Input section

The input to the program is a short text file that lists up to 16 such instructions.
The right format of the input text file is as follows:
<instruction type> <store register> <register j > <register k>
Instruction types include LD, SD, MULTD, DIVD, ADDD and SUBD.

Instructions	Cycles
Load(LD)/Store(SD)	3
Add(ADDD)/Subtract(SUBD)	2
Multiply(MULTD)	10
Divide(DIVD)	40

Table 1. Typical Number of Clock Cycles

3 Main body section


infile instructions and initiate the reservationstation and register
Update reservation station and write back
issue
execute
print out the states

4 Output section

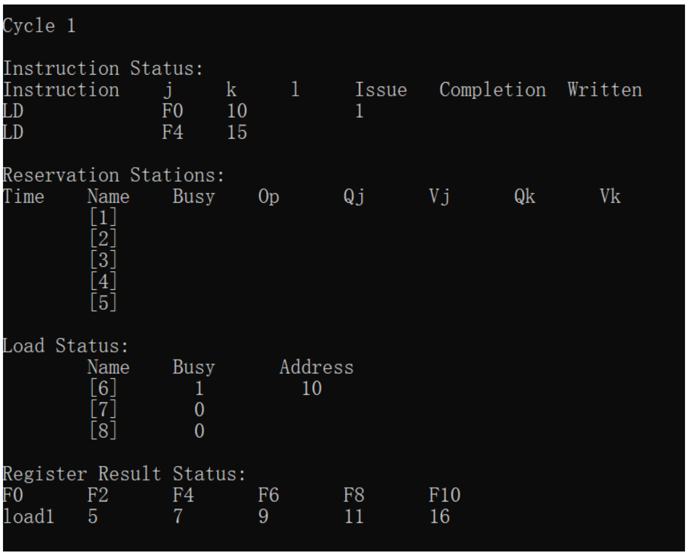
For each cycle, output the execution, completion, and write back of all code instructions, and the changes in the contents of the reservation station, instruction status table, and floating-point register status table.

Examples:

1. Load example

Instructions:

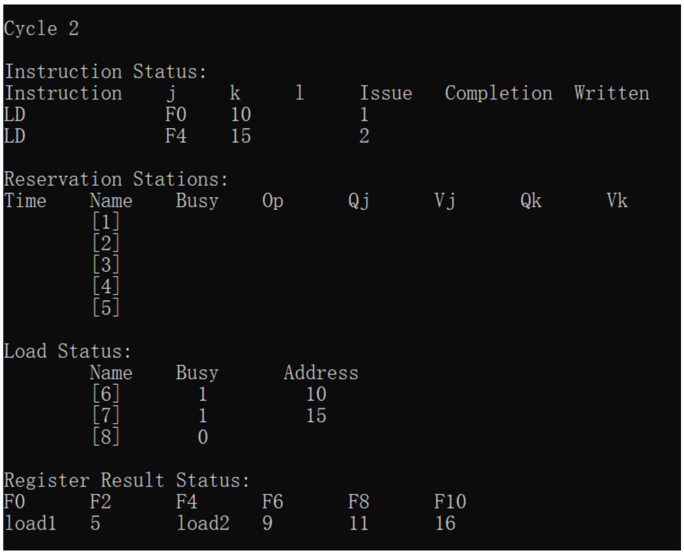
LD F0 10
LD F4 15



```
Cycle 1

Instruction Status:
Instruction     j     k     l      Issue   Completion  Written
LD              F0    10           1
LD              F4    15

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]      1          10
        [7]      0
        [8]      0

Register Result Status:
F0      F2      F4      F6      F8      F10
load1   5       7       9       11      16
```

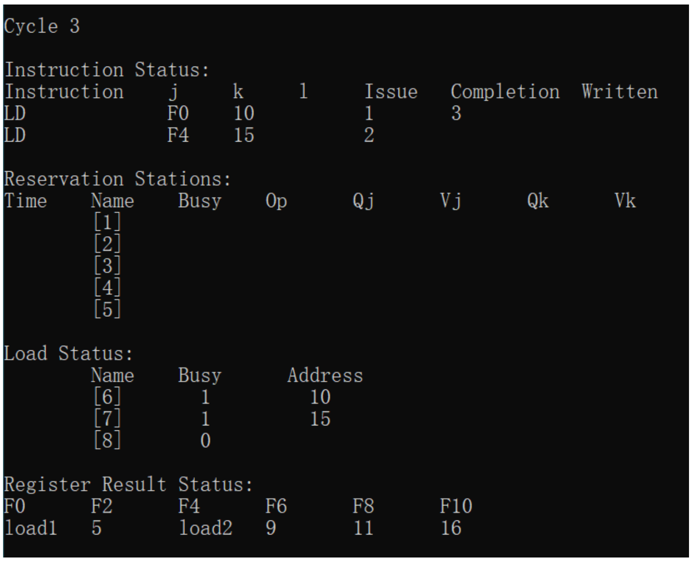
This program includes two load instructions. The cycle of load operation is 3.

In the first cycle, the first instruction starts issuing, and the register F0 becomes “load1”. And in load status, load1 which is [6], is busy.



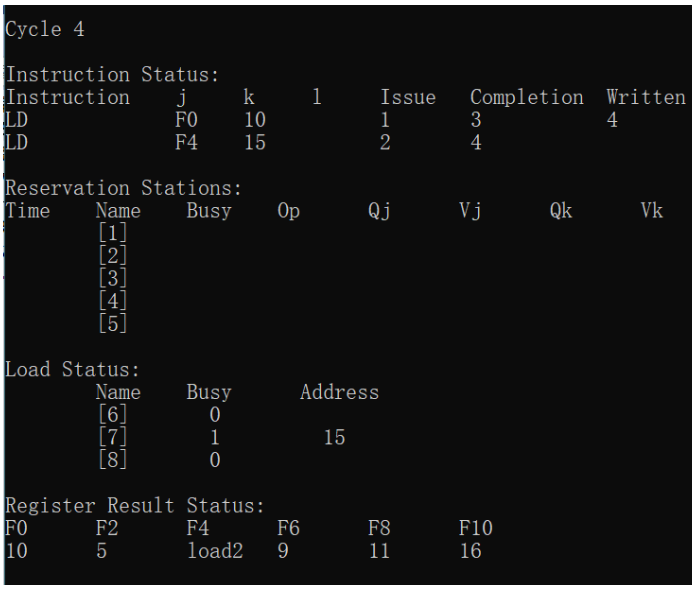
```
Cycle 2

Instruction Status:
Instruction     j     k     l      Issue   Completion  Written
LD              F0    10           1
LD              F4    15           2

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]      1          10
        [7]      1          15
        [8]      0

Register Result Status:
F0      F2      F4      F6      F8      F10
load1   5       load2   9       11      16
```

In the second cycle, the first instruction starts executing, and the second instruction starts issuing. The register F4 becomes “load2”. And in load status, load2 which is [7], is busy too.



```
Cycle 3

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F0    10          1       3
LD             F4    15          2

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]       1         10
        [7]       1         15
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
load1   5       load2   9       11      16
```

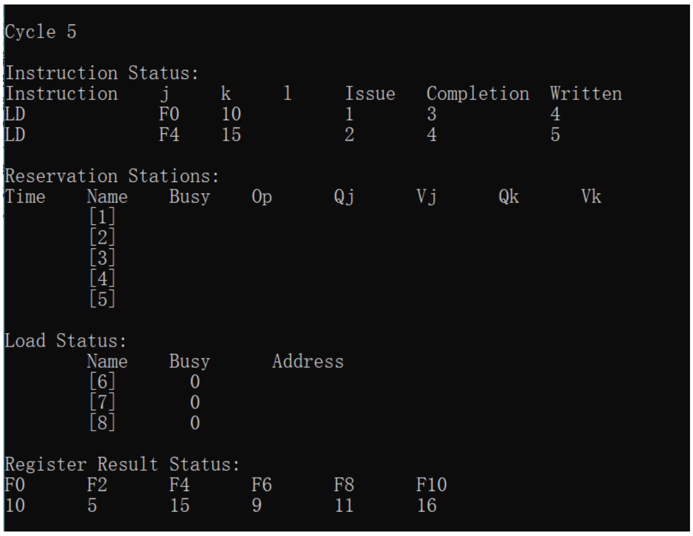
In the third cycle, the 1st instruction has been completed, and the second instruction is still being executed.



```
Cycle 4

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F0    10          1       3           4
LD             F4    15          2       4

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       1         15
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
10      5       load2   9       11      16
```

The first instruction has been written back then load1 is available, and register F0 becomes 10.

The second instruction has finished execution and the load2 is still busy.



```
Cycle 5

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F0    10          1       3           4
LD             F4    15          2       4           5

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
10      5       15      9       11      16
```

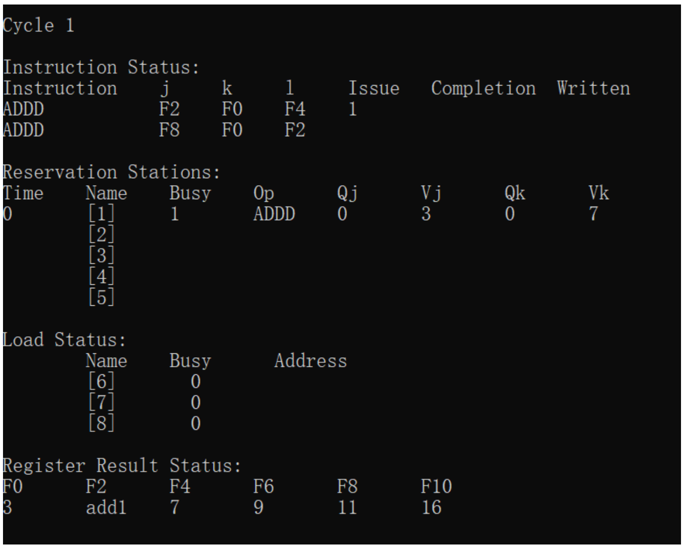
The second instruction has been written back and the load2 is not busy.

It takes 5 clock cycles to execute all the instructions.

2. RAW example

Instructions:

ADDD F2 F0 F4
ADDD F8 F0 F2



```
Cycle 1

Instruction Status:
Instruction     j     k     l     Issue   Completion  Written
ADDD            F2    F0    F4    1
ADDD            F8    F0    F2

Reservation Stations:
Time      Name     Busy     Op       Qj       Vj       Qk       Vk
0         [1]      1        ADDD     0        3        0        7
          [2]
          [3]
          [4]
          [5]

Load Status:
          Name     Busy       Address
          [6]        0
          [7]        0
          [8]        0

Register Result Status:
F0       F2       F4       F6       F8       F10
3        add1     7        9        11       16
```

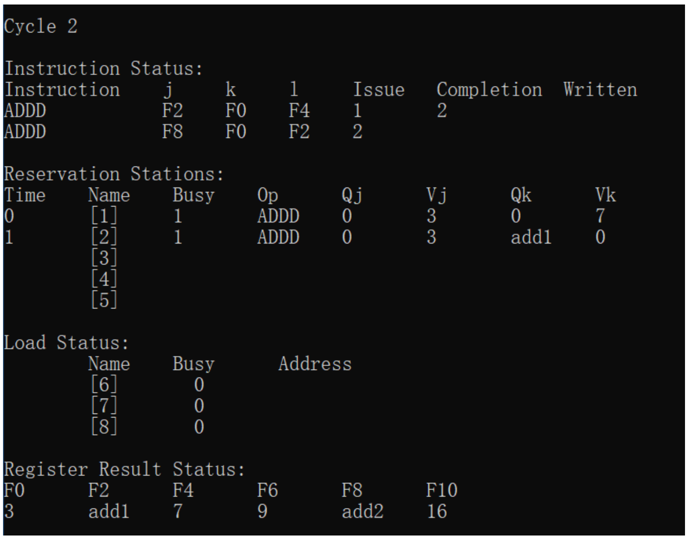
This program includes two add instructions.

For register F2, here is a RAW data hazard.

Cycle 1

The first instruction starts issuing.

In reservation station, the first one (Name [1]), which is add1, is busy. Register F2 is “add1”.

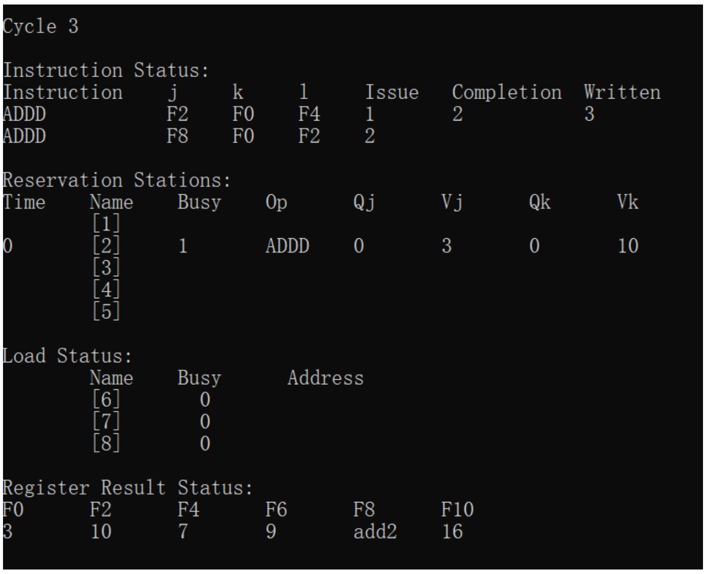


```
Cycle 2

Instruction Status:
Instruction     j     k     l     Issue   Completion  Written
ADDD            F2    F0    F4    1       2
ADDD            F8    F0    F2    2

Reservation Stations:
Time      Name     Busy     Op       Qj       Vj       Qk       Vk
0         [1]      1        ADDD     0        3        0        7
1         [2]      1        ADDD     0        3        add1     0
          [3]
          [4]
          [5]

Load Status:
          Name     Busy       Address
          [6]        0
          [7]        0
          [8]        0

Register Result Status:
F0       F2       F4       F6       F8       F10
3        add1     7        9        add2     16
```

Cycle 2

The second instruction starts issuing. The fir

In reservation station, the second one (Name [1]), which is add2, becomes busy. Qk becomes “add1”. Only the add1 has been written back can add2 starts executing.

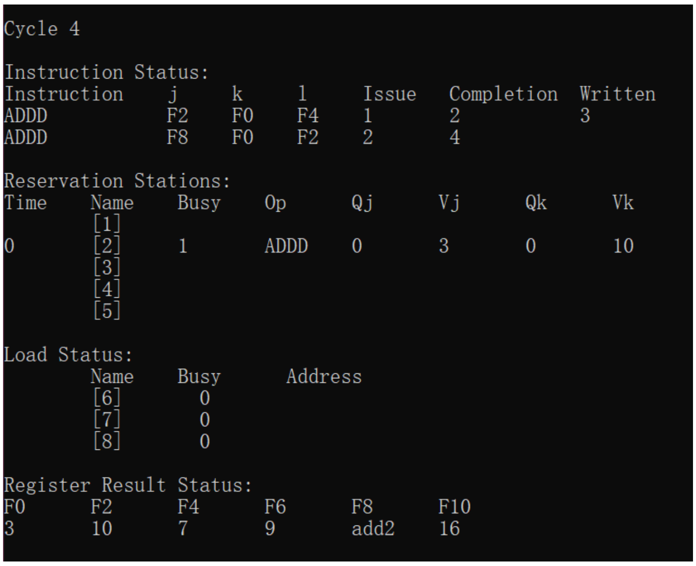


```
Cycle 3

Instruction Status:
Instruction     j     k     l     Issue    Completion   Written
ADDD            F2    F0    F4    1        2            3
ADDD            F8    F0    F2    2

Reservation Stations:
Time     Name     Busy     Op       Qj       Vj       Qk       Vk
         [1]
0        [2]      1        ADDD     0        3        0        10
         [3]
         [4]
         [5]

Load Status:
         Name     Busy       Address
         [6]        0
         [7]        0
         [8]        0

Register Result Status:
F0       F2       F4       F6       F8       F10
3        10       7        9        add2     16
```

Cycle 3

The first instruction has been finished. And register F2 changes from “add1” to the result 10.

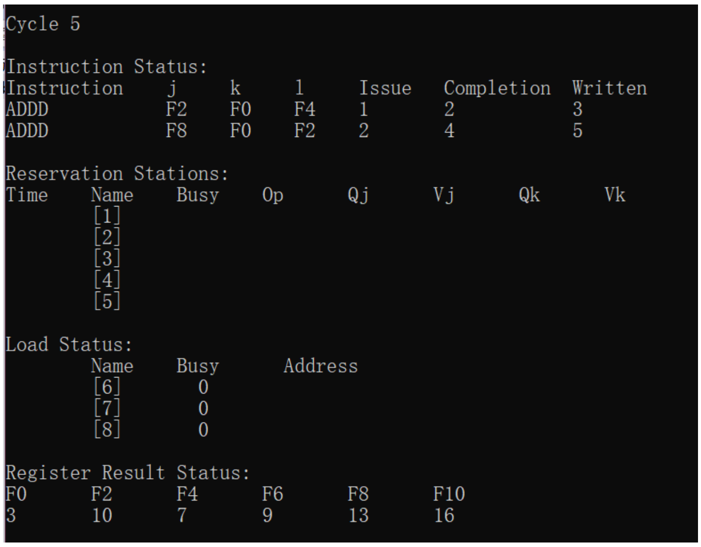
In reservation station, add1 RS is not busy now, and in add2 RS, “add1” also changes to 10, which is shown in Vk, Now the second instruction starts execution.



```
Cycle 4

Instruction Status:
Instruction     j     k     l     Issue    Completion   Written
ADDD            F2    F0    F4    1        2            3
ADDD            F8    F0    F2    2        4

Reservation Stations:
Time     Name     Busy     Op       Qj       Vj       Qk       Vk
         [1]
0        [2]      1        ADDD     0        3        0        10
         [3]
         [4]
         [5]

Load Status:
         Name     Busy       Address
         [6]        0
         [7]        0
         [8]        0

Register Result Status:
F0       F2       F4       F6       F8       F10
3        10       7        9        add2     16
```

Cycle 4

The operation add takes 2 cycles. The second instruction has finished execution.



```
Cycle 5

Instruction Status:
Instruction     j     k     l     Issue    Completion   Written
ADDD            F2    F0    F4    1        2            3
ADDD            F8    F0    F2    2        4            5

Reservation Stations:
Time     Name     Busy     Op       Qj       Vj       Qk       Vk
         [1]
         [2]
         [3]
         [4]
         [5]

Load Status:
         Name     Busy       Address
         [6]        0
         [7]        0
         [8]        0

Register Result Status:
F0       F2       F4       F6       F8       F10
3        10       7        9        13       16
```

Cycle 5

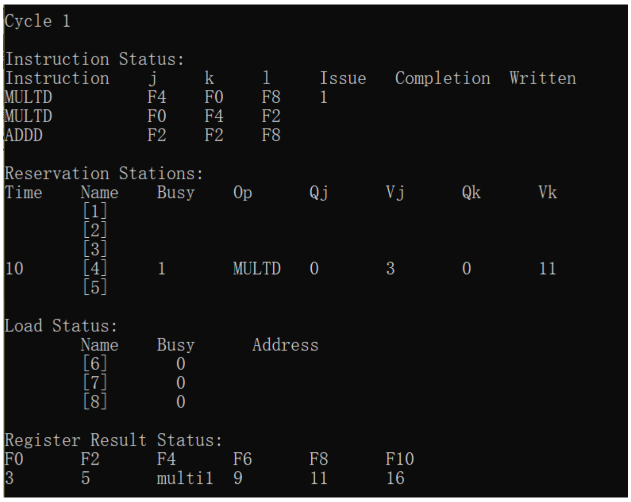
The result of the second instruction has been written back. All the reservation stations and load buffers are available now.

It takes 5 clock cycles to execute all the instructions.

3. WAR example

Instructions:

MULTD F4 F0 F8
MULTD F0 F4 F2
ADDD F2 F2 F8



```
Cycle 1

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
MULTD          F4    F0    F8    1
MULTD          F0    F4    F2
ADDD           F2    F2    F8

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
10      [4]     1       MULTD   0       3       0       11
        [5]

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
3       5       multi1  9       11      16
```

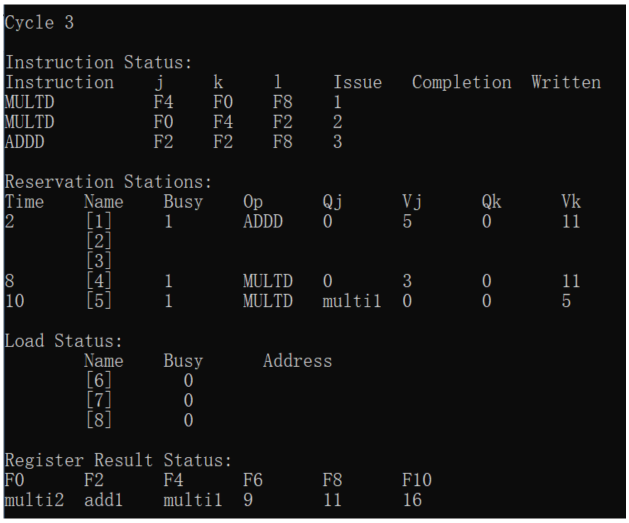
This program includes three instructions.

For register F4, in instruction 1 and 2, here is a WAR data hazard.

Cycle 1

The first instruction starts issuing.

In reservation station, the forth one (Name [4]), which is multi1, is busy. Register F4 is “multi1”.



```
Cycle 3

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
MULTD          F4    F0    F8    1
MULTD          F0    F4    F2    2
ADDD           F2    F2    F8    3

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
2       [1]     1       ADDD    0       5       0       11
        [2]
        [3]
8       [4]     1       MULTD   0       3       0       11
10      [5]     1       MULTD   multi1  0       0       5

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi2  add1    multi1  9       11      16
```

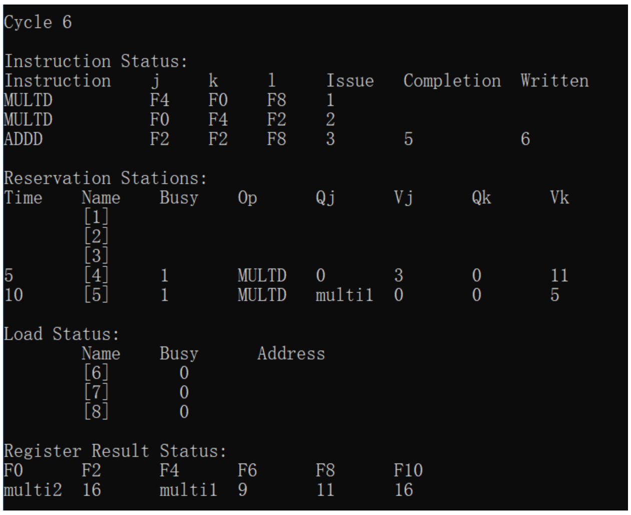
Cycle 3

All the instructions start issuing.

In register, F0 is “multi2”, F2 is “add1”, F4 is “multi1”.

In reservation station, the forth RS which is multi1, has been executed for 2 cycles, there are 8 cycles left.

The fifth RS is multi2, in which Qj is multi1.

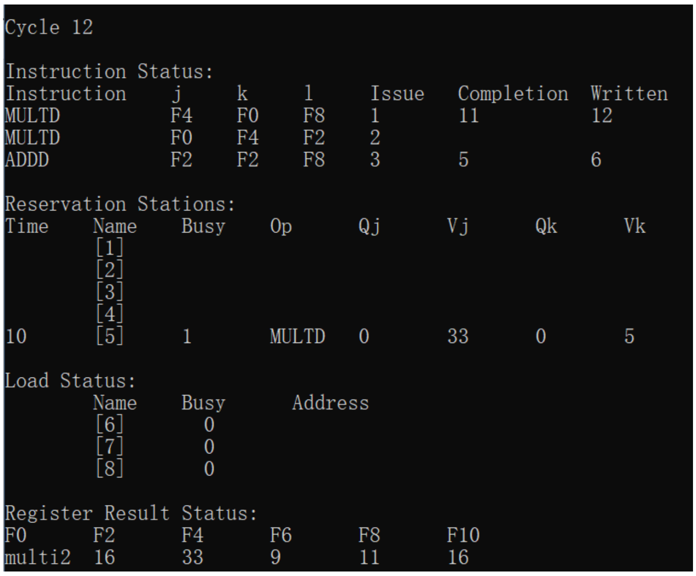


```
Cycle 6

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
MULTD          F4    F0    F8    1
MULTD          F0    F4    F2    2
ADDD           F2    F2    F8    3       5           6

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
5       [4]     1       MULTD   0       3       0       11
10      [5]     1       MULTD   multi1  0       0       5

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi2  16      multi1  9       11      16
```

Cycle 6

The add operation has been finished. And register F2 becomes 16.

In reservation station, multi1 operation has 5 cycles left.



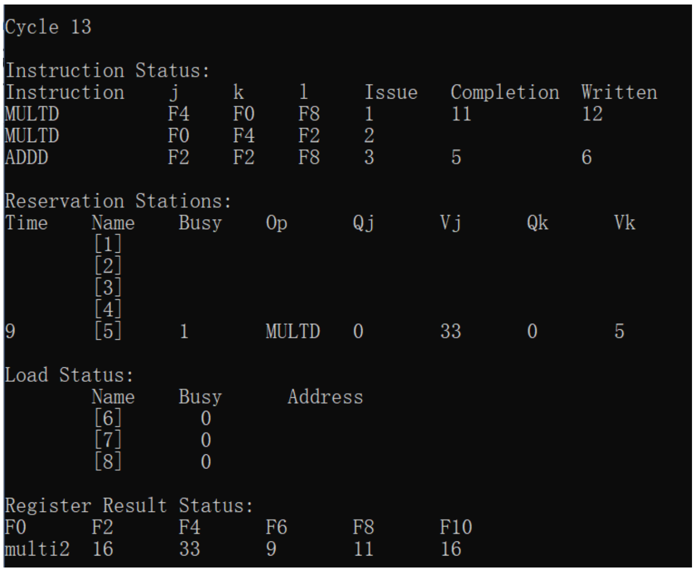
```
Cycle 12

Instruction Status:
Instruction     j     k     l     Issue    Completion   Written
MULTD           F4    F0    F8    1        11           12
MULTD           F0    F4    F2    2
ADDD            F2    F2    F8    3        5            6

Reservation Stations:
Time     Name     Busy     Op       Qj       Vj       Qk       Vk
         [1]
         [2]
         [3]
         [4]
10       [5]      1        MULTD    0        33       0        5

Load Status:
         Name     Busy       Address
         [6]        0
         [7]        0
         [8]        0

Register Result Status:
F0       F2       F4       F6       F8       F10
multi2   16       33       9        11       16
```

Cycle 12

The first instruction has finished. The F4 register changes from "multi1" to the result 33.

In multi2 reservation station, which is [5], Qj ("multi1") changes to Vj (33).

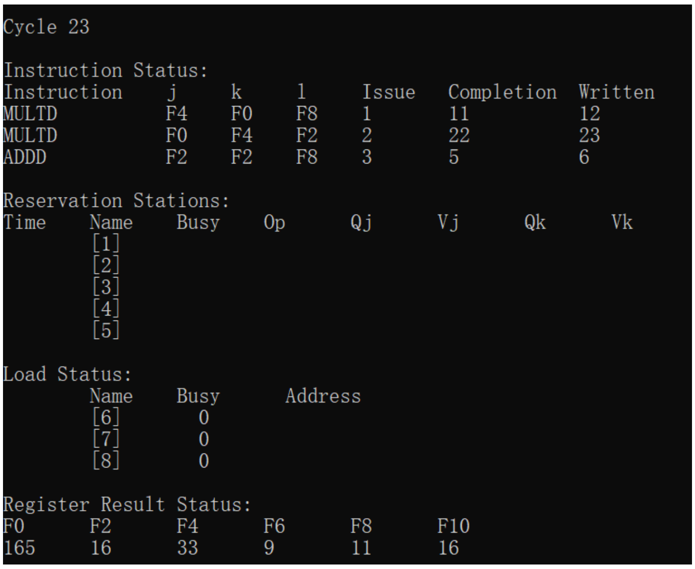
The second instruction starts execution.



```
Cycle 13

Instruction Status:
Instruction     j     k     l     Issue    Completion   Written
MULTD           F4    F0    F8    1        11           12
MULTD           F0    F4    F2    2
ADDD            F2    F2    F8    3        5            6

Reservation Stations:
Time     Name     Busy     Op       Qj       Vj       Qk       Vk
         [1]
         [2]
         [3]
         [4]
9        [5]      1        MULTD    0        33       0        5

Load Status:
         Name     Busy       Address
         [6]        0
         [7]        0
         [8]        0

Register Result Status:
F0       F2       F4       F6       F8       F10
multi2   16       33       9        11       16
```

Cycle 13

The second instruction executes for 1 cycle, and there are 9 cycles left.



```
Cycle 23

Instruction Status:
Instruction     j     k     l     Issue    Completion   Written
MULTD           F4    F0    F8    1        11           12
MULTD           F0    F4    F2    2        22           23
ADDD            F2    F2    F8    3        5            6

Reservation Stations:
Time     Name     Busy     Op       Qj       Vj       Qk       Vk
         [1]
         [2]
         [3]
         [4]
         [5]

Load Status:
         Name     Busy       Address
         [6]        0
         [7]        0
         [8]        0

Register Result Status:
F0       F2       F4       F6       F8       F10
165      16       33       9        11       16
```

Cycle 23

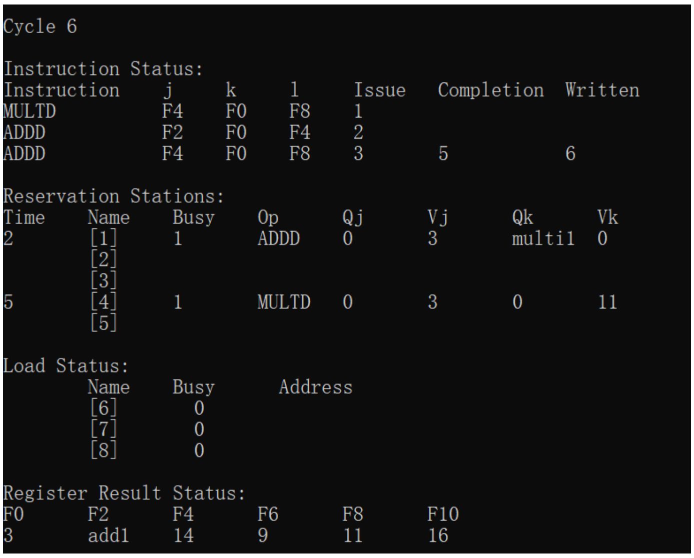
The second instruction finished.

It takes 23 clock cycles to execute all the instructions.

4. WAW example

Instructions:

MULTD F4 F0 F8
ADDD F2 F0 F4
ADDD F4 F0 F8

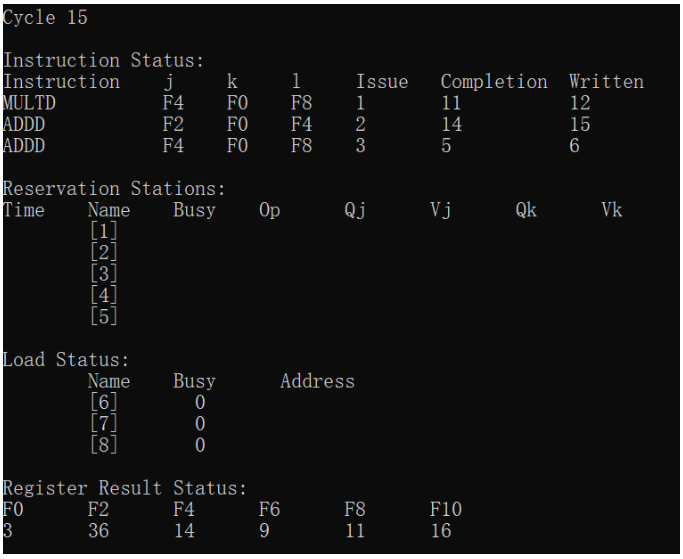


```
Cycle 6

Instruction Status:
Instruction     j      k      l      Issue    Completion  Written
MULTD           F4     F0     F8     1
ADDD            F2     F0     F4     2
ADDD            F4     F0     F8     3        5           6

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
2       [1]     1       ADDD    0       3       multi1  0
        [2]
        [3]
5       [4]     1       MULTD   0       3       0       11
        [5]

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
3       add1    14      9       11      16
```

Cycle 6

The third instruction has finished, and register F4 is the result of the 3rd instruction.

The 1st instruction is still executing. The 2nd instruction is waiting for the Qk (multi1) to be finished.



```
Cycle 15

Instruction Status:
Instruction     j      k      l      Issue    Completion  Written
MULTD           F4     F0     F8     1        11          12
ADDD            F2     F0     F4     2        14          15
ADDD            F4     F0     F8     3        5           6

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
3       36      14      9       11      16
```

Cycle 15

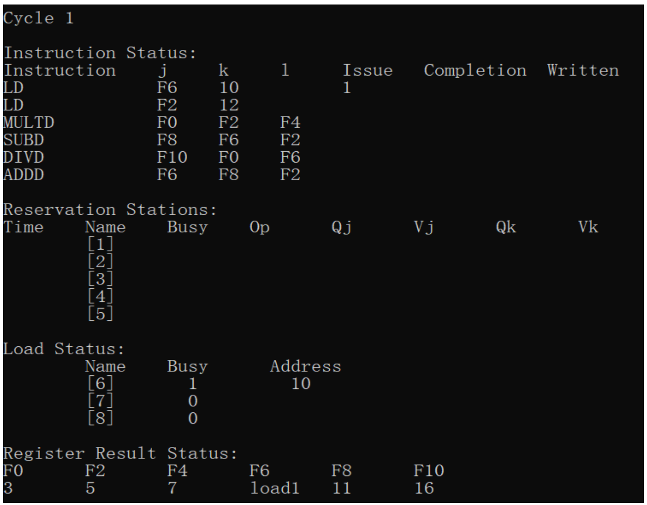
After the multi1 has been written, add1 starts execution. In cycle 15, add1 is finished, too.

It takes 15 clock cycles to execute all the instructions.

5. A long (57-cycle) example in the ppt chap3

Instructions:

LD F6 10
LD F2 12
MULTD F0 F2 F4
SUBD F8 F6 F2
DIVD F10 F0 F6
ADDD F6 F8 F2

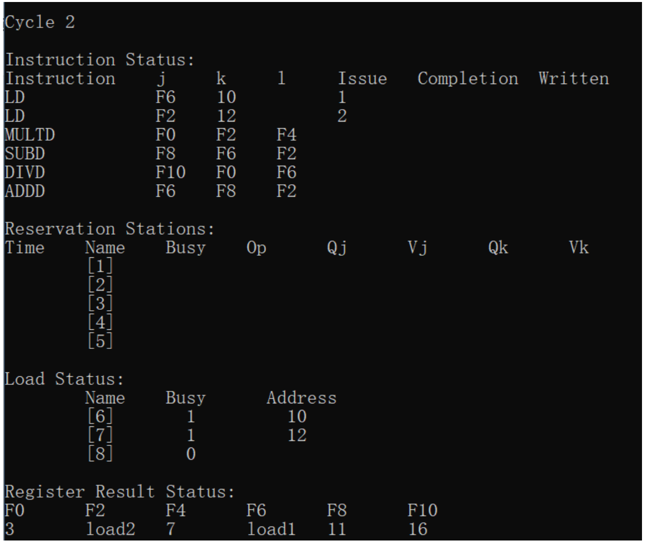


```
Cycle 1

Instruction Status:
Instruction     j     k     l     Issue   Completion  Written
LD              F6    10          1
LD              F2    12
MULTD           F0    F2    F4
SUBD            F8    F6    F2
DIVD            F10   F0    F6
ADDD            F6    F8    F2

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]       1         10
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
3       5       7       load1   11      16
```

This program includes 6 instructions.

Cycle 1

The 1st instruction starts issuing. Register F6 is “load1”. Load1 is busy now.



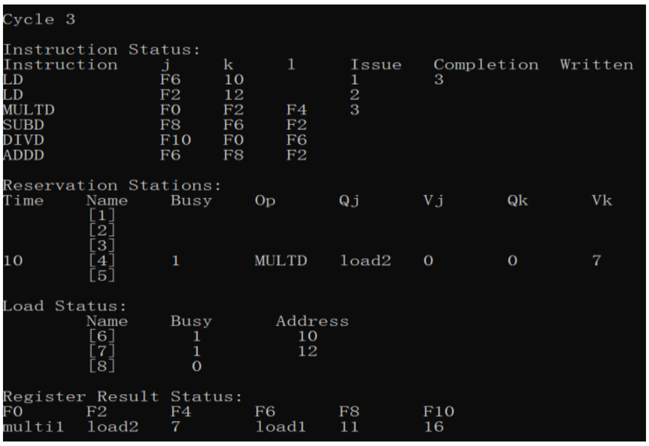
```
Cycle 2

Instruction Status:
Instruction     j     k     l     Issue   Completion  Written
LD              F6    10          1
LD              F2    12          2
MULTD           F0    F2    F4
SUBD            F8    F6    F2
DIVD            F10   F0    F6
ADDD            F6    F8    F2

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]       1         10
        [7]       1         12
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
3       load2   7       load1   11      16
```

Cycle 2

The 1st instruction starts executing. And the 2nd instruction starts issuing.

Load1 and load2 are both busy.

Register F2 is “load2” now.



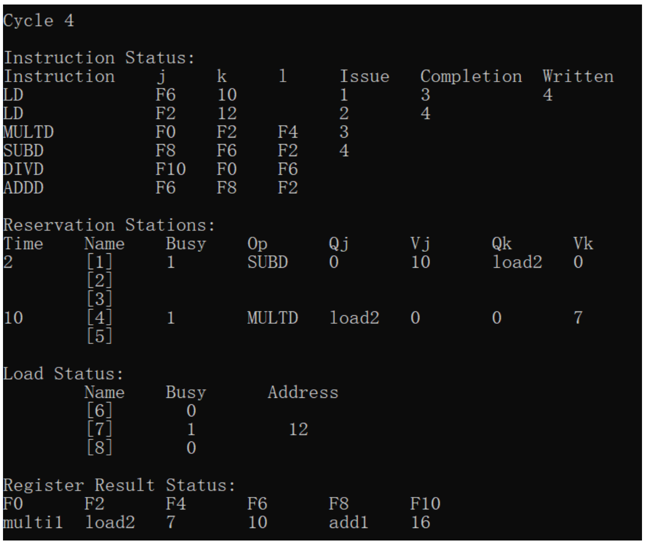
```
Cycle 3

Instruction Status:
Instruction     j     k     l     Issue   Completion  Written
LD              F6    10          1       3
LD              F2    12          2
MULTD           F0    F2    F4    3
SUBD            F8    F6    F2
DIVD            F10   F0    F6
ADDD            F6    F8    F2

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
10      [4]     1       MULTD   load2   0       0       7
        [5]

Load Status:
        Name    Busy      Address
        [6]       1         10
        [7]       1         12
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  load2   7       load1   11      16
```

Cycle 3

The 1st instruction has been executed.

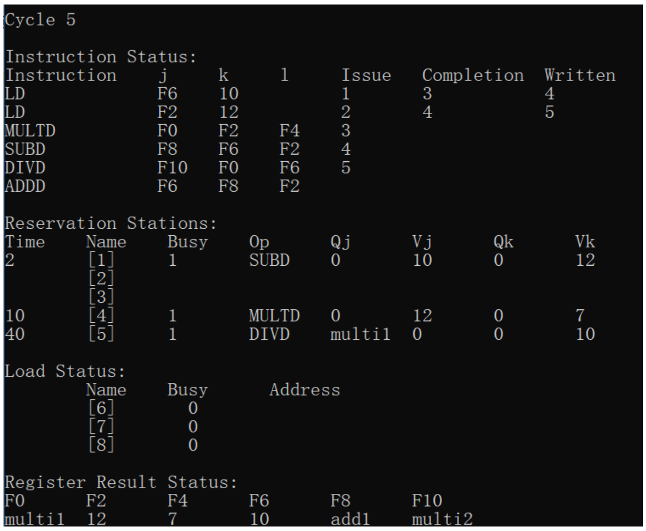
The 3rd instruction start issuing.

In the reservation station, multi1 RS([4]) is waiting for the load2 to be finished.



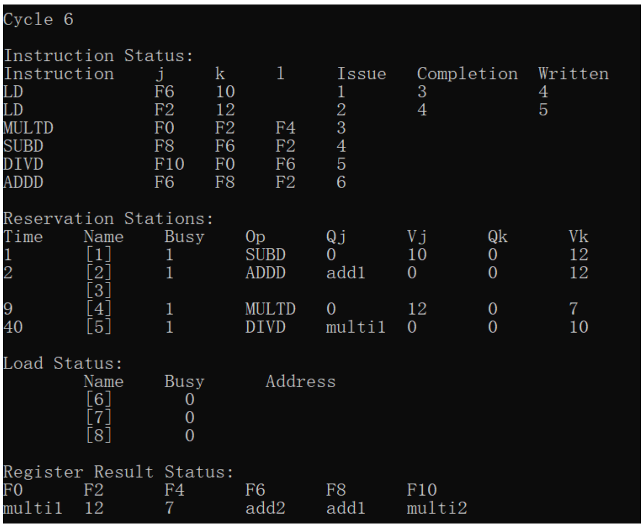
```
Cycle 4

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4
MULTD          F0    F2    F4    3
SUBD           F8    F6    F2    4
DIVD           F10   F0    F6
ADDD           F6    F8    F2

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
2       [1]     1       SUBD    0       10      load2   0
        [2]
        [3]
10      [4]     1       MULTD   load2   0       0       7
        [5]

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       1         12
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  load2   7       10      add1    16
```

Cycle 4
The 4th instruction starts issuing.
The 1st instruction is finished, register F6 is 10, and load1 is available now.
In reservation station, aad1 and multi1 are both waiting for the load2.



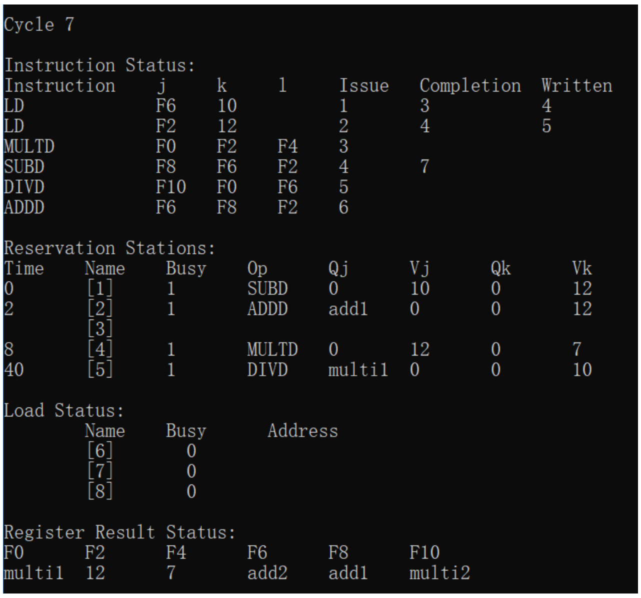
```
Cycle 5

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4           5
MULTD          F0    F2    F4    3
SUBD           F8    F6    F2    4
DIVD           F10   F0    F6    5
ADDD           F6    F8    F2

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
2       [1]     1       SUBD    0       10      0       12
        [2]
        [3]
10      [4]     1       MULTD   0       12      0       7
40      [5]     1       DIVD    multi1  0       0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  12      7       10      add1    multi2
```

Cycle 5
The 2nd instruction has been finished.
In reservation station, load2 (Qj Qk) becomes 12(Vk Vj).
Load2 is emptied.



```
Cycle 6

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4           5
MULTD          F0    F2    F4    3
SUBD           F8    F6    F2    4
DIVD           F10   F0    F6    5
ADDD           F6    F8    F2    6

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
1       [1]     1       SUBD    0       10      0       12
2       [2]     1       ADDD    add1    0       0       12
        [3]
9       [4]     1       MULTD   0       12      0       7
40      [5]     1       DIVD    multi1  0       0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  12      7       add2    add1    multi2
```

Cycle 6
The 6th instruction starts issuing.
The 3rd and 4th instruction starts execution, the corresponding Qj Qk Vj Vk is in the reservation station.

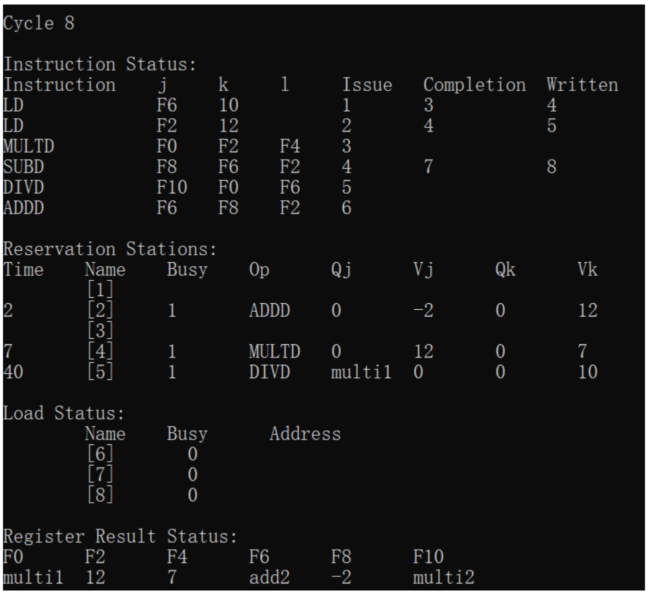


```
Cycle 7

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4           5
MULTD          F0    F2    F4    3
SUBD           F8    F6    F2    4       7
DIVD           F10   F0    F6    5
ADDD           F6    F8    F2    6

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
0       [1]     1       SUBD    0       10      0       12
2       [2]     1       ADDD    add1    0       0       12
        [3]
8       [4]     1       MULTD   0       12      0       7
40      [5]     1       DIVD    multi1  0       0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  12      7       add2    add1    multi2
```

Cycle 7

The 4th instruction is executed.

RS[4] is still executing.



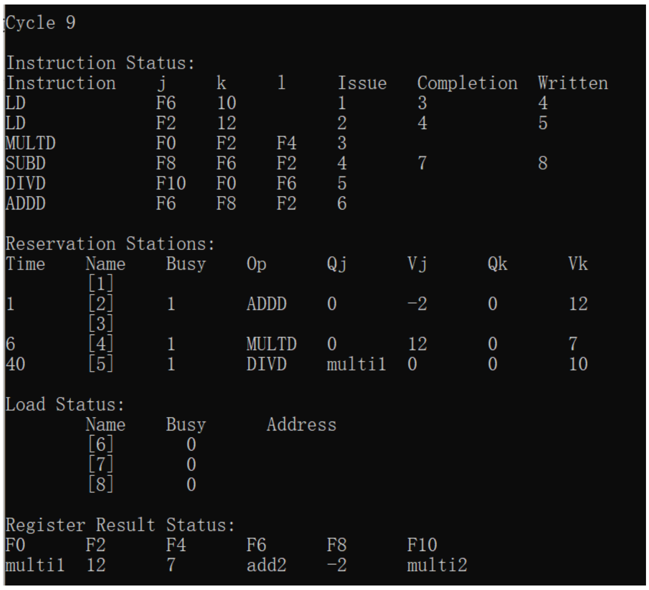
```
Cycle 8

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4           5
MULTD          F0    F2    F4    3
SUBD           F8    F6    F2    4       7           8
DIVD           F10   F0    F6    5
ADDD           F6    F8    F2    6

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
2       [2]     1       ADDD    0       -2      0       12
        [3]
7       [4]     1       MULTD   0       12      0       7
40      [5]     1       DIVD    multi1  0       0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  12      7       add2    -2      multi2
```

Cycle 8

The result of the 4th instruction is written back.

RS[1](add1) is emptied.

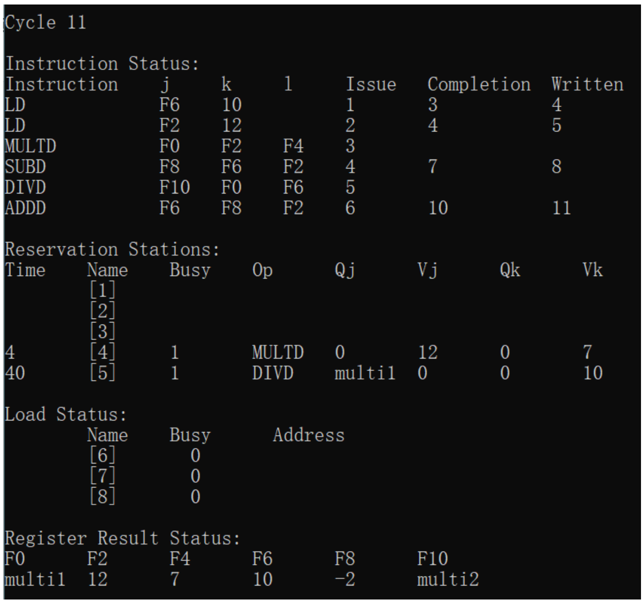
RS[4](multi1) is still executing.



```
Cycle 9

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4           5
MULTD          F0    F2    F4    3
SUBD           F8    F6    F2    4       7           8
DIVD           F10   F0    F6    5
ADDD           F6    F8    F2    6

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
1       [2]     1       ADDD    0       -2      0       12
        [3]
6       [4]     1       MULTD   0       12      0       7
40      [5]     1       DIVD    multi1  0       0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  12      7       add2    -2      multi2
```

Cycle 9

The 6th instruction(RS[2]) starts executing.

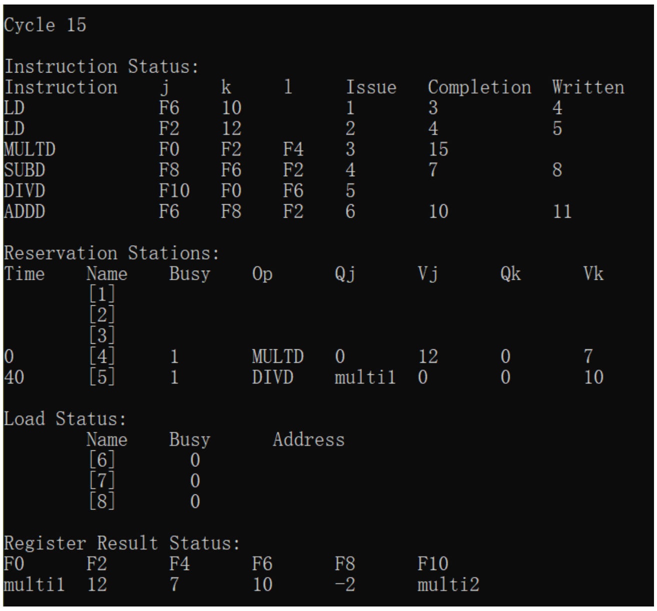


```
Cycle 11

Instruction Status:
Instruction     j      k      l      Issue   Completion  Written
LD              F6     10            1       3           4
LD              F2     12            2       4           5
MULTD           F0     F2     F4     3
SUBD            F8     F6     F2     4       7           8
DIVD            F10    F0     F6     5
ADDD            F6     F8     F2     6       10          11

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
4       [4]     1       MULTD   0       12      0       7
40      [5]     1       DIVD    multi1  0       0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  12      7       10      -2      multi2
```

Cycle 11

The result of the 6th instruction has been written back. RS[2] is emptied.

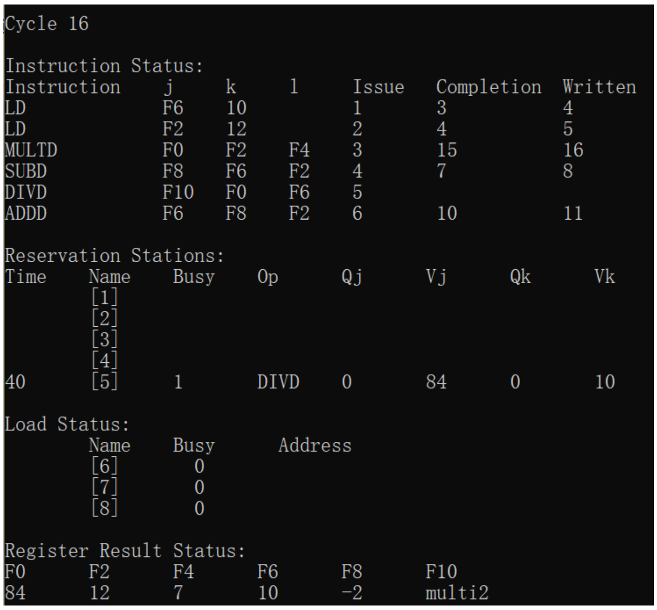
The 3rd instruction (RS[4] multi1) is still executing.



```
Cycle 15

Instruction Status:
Instruction     j      k      l      Issue   Completion  Written
LD              F6     10            1       3           4
LD              F2     12            2       4           5
MULTD           F0     F2     F4     3       15
SUBD            F8     F6     F2     4       7           8
DIVD            F10    F0     F6     5
ADDD            F6     F8     F2     6       10          11

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
0       [4]     1       MULTD   0       12      0       7
40      [5]     1       DIVD    multi1  0       0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
multi1  12      7       10      -2      multi2
```

Cycle 12~15

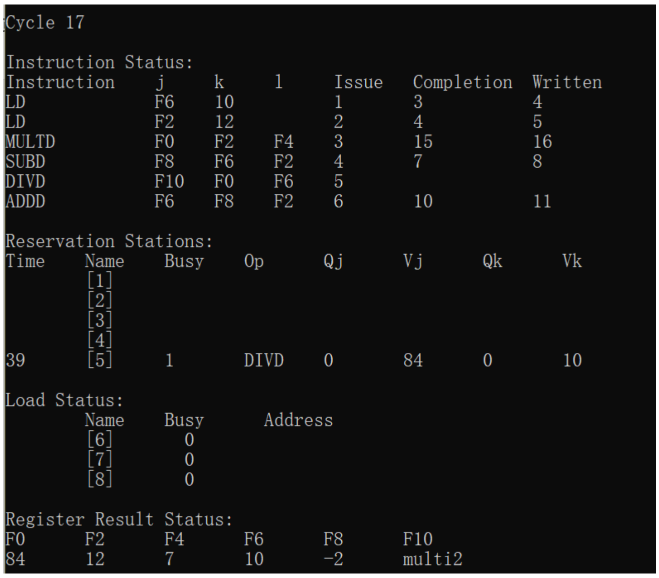
The 3rd instruction goes on executing until completed.



```
Cycle 16

Instruction Status:
Instruction     j      k      l      Issue   Completion  Written
LD              F6     10            1       3           4
LD              F2     12            2       4           5
MULTD           F0     F2     F4     3       15          16
SUBD            F8     F6     F2     4       7           8
DIVD            F10    F0     F6     5
ADDD            F6     F8     F2     6       10          11

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
40      [5]     1       DIVD    0       84      0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
84      12      7       10      -2      multi2
```

Cycle 16

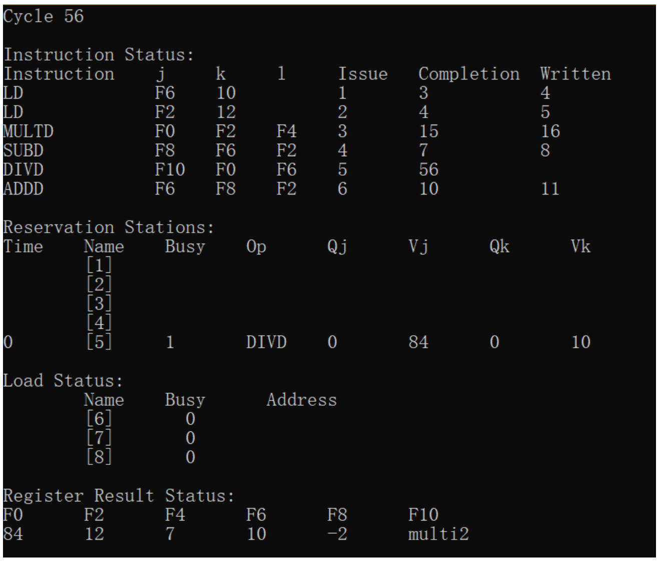
The result of 3rd instruction is written back. In reservation station, “multi1” is 84 right now. RS[4] is emptied.



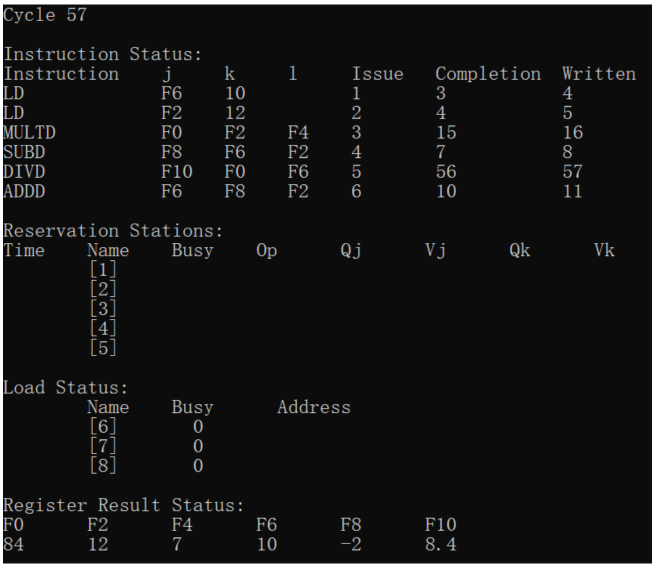
```
Cycle 17

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4           5
MULTD          F0    F2    F4    3       15          16
SUBD           F8    F6    F2    4       7           8
DIVD           F10   F0    F6    5
ADDD           F6    F8    F2    6       10          11

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
39      [5]     1       DIVD    0       84      0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
84      12      7       10      -2      multi2
```

Cycle 17~56

The 5th instruction starts execution till finished.



```
Cycle 56

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4           5
MULTD          F0    F2    F4    3       15          16
SUBD           F8    F6    F2    4       7           8
DIVD           F10   F0    F6    5       56
ADDD           F6    F8    F2    6       10          11

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
0       [5]     1       DIVD    0       84      0       10

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
84      12      7       10      -2      multi2
```



```
Cycle 57

Instruction Status:
Instruction    j     k     l     Issue   Completion  Written
LD             F6    10          1       3           4
LD             F2    12          2       4           5
MULTD          F0    F2    F4    3       15          16
SUBD           F8    F6    F2    4       7           8
DIVD           F10   F0    F6    5       56          57
ADDD           F6    F8    F2    6       10          11

Reservation Stations:
Time    Name    Busy    Op      Qj      Vj      Qk      Vk
        [1]
        [2]
        [3]
        [4]
        [5]

Load Status:
        Name    Busy      Address
        [6]       0
        [7]       0
        [8]       0

Register Result Status:
F0      F2      F4      F6      F8      F10
84      12      7       10      -2      8.4
```

Cycle 57

The last writeback completed.

It takes 57 clock cycles to execute all the instructions.

Appendix A

Tomasulo.pdf

Appendix B Input.txt

1. RAWExample.txt

```
ADDD F2 F0 F4
ADDD F8 F0 F2
```

2. WARExample.txt

```
MULTD F4 F0 F8
MULTD F0 F4 F2
ADDD F2 F2 F8
```

3. WAWExample.txt

```
MULTD F4 F0 F8
ADDD F2 F0 F4
ADDD F4 F0 F8
```

4. LoadExample.txt

```
LD F0 10
LD F4 15
```

5. LongExample.txt

```
LD F6 10
LD F2 12
MULTD F0 F2 F4
SUBD F8 F6 F2
DIVD F10 F0 F6
ADDD F6 F8 F2
```