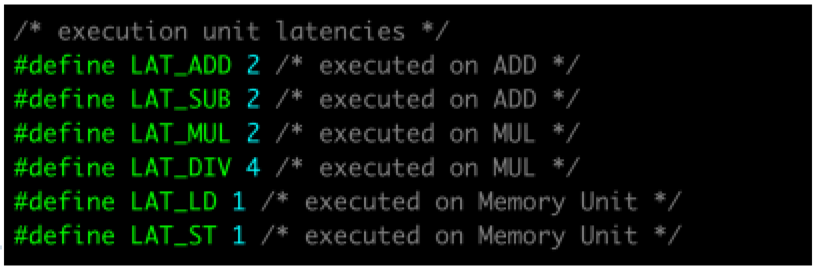
Laurence Kim
ELEN 511 ADV COMP ARCH
DR. YANG

HW #2 Report

Part 0: Understand the given architecture

Instructions
Ld r6,32(r12)
Ld r2, 44(r13)
Mul r0,r2,r4
Sub r8,r2,r6
Div r10,r0,r6
Add r11,r0,r6



```
/* execution unit latencies */
#define LAT_ADD 2 /* executed on ADD */
#define LAT_SUB 2 /* executed on ADD */
#define LAT_MUL 2 /* executed on MUL */
#define LAT_DIV 4 /* executed on MUL */
#define LAT_LD 1 /* executed on Memory Unit */
#define LAT_ST 1 /* executed on Memory Unit */
```

Cycles	LD1	LD2	ADD1	ADD2	MULT1	MULT2
1	I					
2	E	I				
3	W	E			I	
4		W	I		E	
5			E		E	I
6			E	I	W	E
7			W	E		E
8				E		E
9				W		E
10						W

I estimate this sequence of instructions to take **10** cycles. My reasoning is depicted in the table above. The table visualizes the sequence of the instructions fed into the reservation stations with the step of the reservation station during its respective cycle.
I is the issue step, E is the execute step, and W is the write step.

Part 1: Implement the basic Tomasulo Algorithm

Why do they appear in a reverse order?

They appear in a reverse order because if we have issue, execute, and write sequentially, then we could have issue, execute, and write all happen in the same cycle. Due to the nature of the CDB and data dependencies, finishing write before execute actually enables the features of tomasulo's algorithm.

Cycles	LD1	LD2	ADD1	ADD2	MULT1	MULT2
1	I					
2	E	I				
3	E	-			I	
4	W	E	I		-	
5		E	-		-	I
6		W	E	I	E	-
7			E	-	E	-
8			W	E	W	E
9				E		E
10				W		E
11						E
12						W

12 cycles estimated.

My set of instructions that I use for 3 data dependencies are as show below

```
 inst[0].num=1; inst[0].op=LD; inst[0].rd=6; inst[0].rs=12; inst[0].rt=32;
// ld r6,32(r12)
 inst[1].num=2; inst[1].op=LD; inst[1].rd=2; inst[1].rs=13; inst[1].rt=44;
// ld r2,44(r13)
 inst[2].num=3; inst[2].op=DIV; inst[2].rd=0; inst[2].rs=2; inst[2].rt=4;
// div r0, r2, r4
 inst[3].num=4; inst[3].op=ADD; inst[3].rd=8; inst[3].rs=0; inst[3].rt=3;
// add r8, r0, r3
 inst[4].num=5; inst[4].op=SUB; inst[4].rd=7; inst[4].rs=0; inst[4].rt=1;
// sub r7, r0, r1
```

```
 inst[5].num=6; inst[5].op=MUL; inst[5].rd=9; inst[5].rs=7; inst[5].rt=6;
// mul r9, r7, r6
 inst[6].num=7; inst[6].op=ST; inst[6].rd=8; inst[6].rs=8; inst[6].rt=0;
// st r8,0(r8)
```

The console output can be shown at the end of the report.

Part 3: HW speculation – 4 points
The sequence of instructions are stated as seen in my code below.

```
void init_inst()
{
 inst[0].num=1; inst[0].op=LD; inst[0].rd=0; inst[0].rs=1; inst[0].rt=0;
// ld r0,0(r1)
 inst[1].num=2; inst[1].op=MUL; inst[1].rd=4; inst[1].rs=0; inst[1].rt=2;
// mul r4,r0, r2
 inst[2].num=3; inst[2].op=ST; inst[2].rd=4; inst[2].rs=1; inst[2].rt=0;
// st r4, 0(r1)
 inst[3].num=4; inst[3].op=ADDI; inst[3].rd=1; inst[3].rs=1; inst[3].rt=1;
// addi r1, r1, 1
 inst[4].num=5; inst[4].op=BNE; inst[4].rd=1; inst[4].rs=2; inst[4].rt=6;
// bne r1, r2, loop
 return;
}
```

We are expecting the reorder buffer to implemented as an additional hardware speculation that will enhance parallelism within our preexisting processor that implements tomasulo's algorithm. The reorder buffers is a circular queue with a head and tail pointers. The instruction would be assigned to the entry of the buffer at the tail which is the destination register. At the end of the execution step, we expect that that the value is put into the instruction reorder buffer's position and then this value is stored in the register after deciding whether to be committed or not. The reorder buffer always holds the instructions that are expected to be ran if the the branch is taken. This means that there needs to be more memory and power taken in order to implement the reorder buffer, but we can save latency with loops by feeding in the instruction immediately from the buffer by having it ready. In order completion is done by including the type of instruction, destination, and value. The previous rs tags are not the ids of the entries in the recorder buffer. The re order buffer encompasses 4 different stages being the issue, execute, write, and commit, in comparison to the previous version not having a commit.

I have attached my code and results as can be seen in the attachments below.

The attachments below are labeled with headers that follow..

- Part 1 code
- Part 1 Given Instruction Console Output (LDLAT=1)
- Part 1 Given Instruction Console Output (LDLAT=2)
- PART 1 CODE PERSONAL INSTRUCTIONS
- Part 3 inst.c
- Part 3 inst.h
- Part 3 arch.c
- Part 3 arch.h
- Part 3 tomasulo.c
- Part 3 console output

Part 1 Code

```
#include <stdio.h>
#include <stdlib.h>
#include "arch.h"


/******************************
main
******************************/
int main()
{
 int i,j;
 int done = 0;
 int cycle = 0;
 int num_issued_inst = 0;

 init_inst();
 init_fu();

 printf("============== TEST INSTRUCTION SEQUENCE ===========\n");
 print_program();

 init_rs();  // initialize RS entries
 init_regs();  // initalize registers
 init_mem(); // initialize memory
 printf("* CYCLE %d (initial state)\n",cycle);
 print_rs(); // print initial RS state
 print_regs(); // print initial register state

/*
1:ld1
2:ld2
3:st1
4:st2
5:add1
6:add2
7:add3
8:mul1
9:mul2
*/
 /* simulation loop main */
 while(!done){

   /* increment the cycle */
   cycle++;

   /*********************************
    *     Step III: Write result
    *********************************/
    for(i=0;i<NUM_RS_ENTRIES;i++) {
      // complete STEP III here
     if(rs_array[i].is_result_ready){//if execution is complete at R
       if((rs_array[i].op==ST)&& (rs_array[i].Qk==0)){ //if station
==store and RS[r].Qk==0
         set_mem(rs_array[i].A,rs_array[i].Vk);
         if(!is_mem_available){
           is_mem_available=true;
         }
         reset_rs_entry(&rs_array[i]);
       }
       else{//if station ==fp or load and cdb is available
         int j;
         for(j=0;j<16 ;j++){//j is num of reg
           if(regs[j].Qi==rs_array[i].id){
             regs[j].val=rs_array[i].result;
```

```
              regs[j].Qi=0;
            }
          }
          int k;
          for(k=0;k<NUM_RS_ENTRIES;k++){//j is num of reg
            if(rs_array[k].Qj==rs_array[i].id){
              rs_array[k].Vj=rs_array[i].result;
              rs_array[k].Qj=0;
            }
            if(rs_array[k].Qk==rs_array[i].id){
              rs_array[k].Vk=rs_array[i].result;
              rs_array[k].Qk=0;
            }
          }
          if((rs_array[i].op==ADD)||(rs_array[i].op==SUB))  {
            if(!is_add_available)
            is_add_available=true;
          }
          if((rs_array[i].op==MUL)||(rs_array[i].op==DIV))  {
            if(!is_mul_available)
            is_mul_available=true;
          }
          if((rs_array[i].op==LD))  {
            if(!is_mem_available)
            is_mem_available=true;
          }
          reset_rs_entry(&rs_array[i]);
        }
      }
    }

    /********************************
     *     Step II: Execute
     ********************************/
      for(i=0;i<NUM_RS_ENTRIES;i++) {
       // complete STEP II here
      bool eq0=(rs_array[i].Qj ==0) && (rs_array[i].Qk ==0);
      if((rs_array[i].op ==ADD)&& eq0){//if fp rs and RS[r].Qj ==0 and
RS[r].Qk ==0
        if(is_add_available){
          is_add_available=false;
          rs_array[i].in_exec=true;
        }
        if(rs_array[i].in_exec)
          rs_array[i].exec_cycles--;
        if(rs_array[i].exec_cycles==0){
          rs_array[i].result = rs_array[i].Vj + rs_array[i].Vk;
          rs_array[i].is_result_ready=true;
        }
       }

      if((rs_array[i].op ==SUB)&& eq0){//if fp rs and RS[r].Qj ==0 and
RS[r].Qk ==0
        if(is_add_available){
          is_add_available=false;
          rs_array[i].in_exec=true;
        }
        if(rs_array[i].in_exec)
          rs_array[i].exec_cycles--;
        if(rs_array[i].exec_cycles==0){
          rs_array[i].result = rs_array[i].Vj - rs_array[i].Vk;
          rs_array[i].is_result_ready=true;
        }
       }

      if((rs_array[i].op ==MUL)&& eq0){//if fp rs and RS[r].Qj ==0 and
RS[r].Qk ==0
        if(is_mul_available){
          is_mul_available=false;
          rs_array[i].in_exec=true;
        }
        if(rs_array[i].in_exec)
          rs_array[i].exec_cycles--;
        if(rs_array[i].exec_cycles==0){
          rs_array[i].result = rs_array[i].Vj * rs_array[i].Vk;
          rs_array[i].is_result_ready=true;
        }
       }
      if((rs_array[i].op ==DIV)&& eq0){//if fp rs and RS[r].Qj ==0 and
RS[r].Qk ==0
```

```
       if(is_mul_available){
         is_mul_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].result = rs_array[i].Vj / rs_array[i].Vk;
         rs_array[i].is_result_ready=true;
       }
      }
     if((rs_array[i].op ==LD)&& (rs_array[i].Qj==0)){//if fp rs and
RS[r].Qj ==0 and RS[r].Qk ==0
       if(is_mem_available){
         is_mem_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].A = rs_array[i].Vj +rs_array[i].A;
         rs_array[i].result = get_mem(rs_array[i].A);
         rs_array[i].is_result_ready=true;
       }
     }
     if((rs_array[i].op ==ST)&& (rs_array[i].Qj==0)){//if fp rs and
RS[r].Qj ==0 and RS[r].Qk ==0
       if(is_mem_available){
         is_mem_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].A = rs_array[i].Vj +rs_array[i].A;
         rs_array[i].is_result_ready=true;
       }
     }
   }
   /*******************************
    *     Step I: Issue
    *******************************/

   /*  wait if no RS entry is available */
   if(num_issued_inst < NUM_OF_INST) {
     int cand_rs_id;
     if(inst[num_issued_inst].op==ADD) cand_rs_id =
obtain_available_rs(ADD_RS);
     else if(inst[num_issued_inst].op==SUB) cand_rs_id =
obtain_available_rs(ADD_RS);
     else if(inst[num_issued_inst].op==MUL) cand_rs_id =
obtain_available_rs(MUL_RS);
     else if(inst[num_issued_inst].op==DIV) cand_rs_id =
obtain_available_rs(MUL_RS);
     else if(inst[num_issued_inst].op==LD) cand_rs_id =
obtain_available_rs(LD_BUF);
     else if(inst[num_issued_inst].op==ST) cand_rs_id =
obtain_available_rs(ST_BUF);

     /* if there is an available RS entry */
     if(cand_rs_id!=-1) {
       /* issue the instruction: See Fig. 3.13 */
       RS * curr_rs = get_rs(cand_rs_id);
       if(curr_rs ==NULL) {
         printf("NO RS found with the given id\n");
         exit(1);
       }

       /* normal ALU operations */
       if(inst[num_issued_inst].op==ADD || inst[num_issued_inst].op==SUB
||
         inst[num_issued_inst].op== MUL ||  inst[num_issued_inst].op==DIV)
{
         int rd, rs, rt;
         rd = inst[num_issued_inst].rd;
         rs = inst[num_issued_inst].rs;
         rt = inst[num_issued_inst].rt;

         /* Rs */
```

```
        if(regs[rs].Qi!=0) curr_rs->Qj = regs[rs].Qi;
        else curr_rs->Vj = regs[rs].val;

        /* Rt */
        if(regs[rt].Qi!=0) curr_rs->Qk = regs[rt].Qi;
        else curr_rs->Vk = regs[rt].val;

        /* set busy */
        curr_rs->is_busy = true;
        curr_rs->op = inst[num_issued_inst].op;

        /* register update */
        regs[rd].Qi = curr_rs->id;

        /* set exec cycles */
        if(inst[num_issued_inst].op==ADD) curr_rs->exec_cycles=LAT_ADD;
        else if(inst[num_issued_inst].op==SUB)
curr_rs->exec_cycles=LAT_SUB;
        else if(inst[num_issued_inst].op==MUL)
curr_rs->exec_cycles=LAT_MUL;
        else if(inst[num_issued_inst].op==DIV)
curr_rs->exec_cycles=LAT_DIV;

        /* num issued ++ */
        num_issued_inst++;

      } else if (inst[num_issued_inst].op==LD) {

        int rd, rs, imm;
        rd = inst[num_issued_inst].rd;
        rs = inst[num_issued_inst].rs;
        imm = inst[num_issued_inst].rt;

        /* Rs */
        if(regs[rs].Qi!=0) curr_rs->Qj = regs[rs].Qi;
        else curr_rs->Vj = regs[rs].val;

        /* addr */
        curr_rs->A = imm;
        /* set busy */
        curr_rs->is_busy = true;
        curr_rs->op = inst[num_issued_inst].op;

        /* set exec cycles */
        curr_rs->exec_cycles=LAT_LD;

        /* register update */
        regs[rd].Qi = curr_rs->id;

        /* num issued ++ */
        num_issued_inst++;

      } else if (inst[num_issued_inst].op==ST) {

        int rd, rs, imm;
        rd = inst[num_issued_inst].rd;
        rs = inst[num_issued_inst].rs;
        imm = inst[num_issued_inst].rt;

        /* Rs */
        if(regs[rs].Qi!=0) curr_rs->Qj = regs[rs].Qi;
        else curr_rs->Vj = regs[rs].val;

        /* Rd */
        if(regs[rd].Qi!=0) curr_rs->Qk = regs[rd].Qi;
        else curr_rs->Vk = regs[rd].val;

        /* addr */
        curr_rs->A = imm;

        /* set busy */
        curr_rs->is_busy = true;
        curr_rs->op = inst[num_issued_inst].op;

        /* set exec cycles */
        curr_rs->exec_cycles=LAT_ST;

        /* num issued ++ */
        num_issued_inst++;
```

```
       }
     }
   }



   /* print out the result */
   printf("* CYCLE %d\n",cycle);
   print_rs();
   print_regs();

   /* check the termination condition */
   if( (num_issued_inst>=NUM_OF_INST) && !is_rs_active())
     done =1;
 }
 return 0;
}
```

Part 1 Given Instruction Console Output (LDLAT=1)

```
============== TEST INSTRUCTION SEQUENCE ===========
I#1    ld    r6,32(r12)
I#2    ld    r2,44(r13)
I#3    mul    r0,r2,r4
I#4    sub    r8,r2,r6
I#5    div    r10,r0,r6
I#6    add    r11,r0,r6
* CYCLE 0 (initial state)
=============================================================================
==========================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
----------------------------------------------------------------------------------------------------
#1    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#2    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#3    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#4    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#5    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#6    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#7    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#8    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#9    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
```

```
=================================================================================
===========================
=================================================================================
Registers
-----------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------
val   0   1   2   3   4   5   6   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------
=================================================================================
* CYCLE 1
=================================================================================
===========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
---------------------------------------------------------------------------------------------
#1   LD   Yes   I#-1   ld   12   -1   #0   #0   32   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
=================================================================================
===========================
=================================================================================
Registers
-----------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------
val   0   1   2   3   4   5   6   7
Qi   #0   #0   #0   #0   #0   #0   #1   #0
-----------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
```

```
--------------------------------------------------------------------
========================================================================
* CYCLE 2
=============================================================================
========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
---------------------------------------------------------------------------------------------
#1   LD   Yes   I#-1   ld   12   -1   #0   #0   44   Yes   Yes
#2   LD   Yes   I#-1   ld   13   -1   #0   #0   44   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
=============================================================================
========================
========================================================================
Registers
--------------------------------------------------------------------
          r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   0   1   2   3   4   5   6   7
Qi   #0   #0   #2   #0   #0   #0   #1   #0
--------------------------------------------------------------------
          r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
========================================================================
* CYCLE 3
=============================================================================
========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
---------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   Yes   I#-1   ld   13   -1   #0   #0   57   Yes   Yes
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
```

```
#8    MUL    Yes    I#-1    mul    -1    4    #2    #0    -1    No    No
#9    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
============================================================================
===========================
=========================================================================
Registers
------------------------------------------------------------------
        r0    r1    r2    r3    r4    r5    r6    r7
------------------------------------------------------------------
val    0    1    2    3    4    5    12    7
Qi    #8    #0    #2    #0    #0    #0    #0    #0
------------------------------------------------------------------
        r8    r9    r10    r11    r12    r13    r14    r15
------------------------------------------------------------------
val    8    9    10    11    12    13    14    15
Qi    #0    #0    #0    #0    #0    #0    #0    #0
------------------------------------------------------------------
=========================================================================
* CYCLE 4
============================================================================
===========================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
------------------------------------------------------------------------------------------
#1    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#2    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#3    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#4    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#5    ADD    Yes    I#-1    sub    9    12    #0    #0    -1    No    No
#6    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#7    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#8    MUL    Yes    I#-1    mul    9    4    #0    #0    -1    Yes    No
#9    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
============================================================================
===========================
=========================================================================
Registers
------------------------------------------------------------------
        r0    r1    r2    r3    r4    r5    r6    r7
------------------------------------------------------------------
val    0    1    9    3    4    5    12    7
Qi    #8    #0    #0    #0    #0    #0    #0    #0
------------------------------------------------------------------
        r8    r9    r10    r11    r12    r13    r14    r15
------------------------------------------------------------------
```

```
val    8    9    10    11    12    13    14    15
Qi    #5    #0    #0    #0    #0    #0    #0    #0
----------------------------------------------------------------------
=============================================================
* CYCLE 5
================================================================
=======================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
----------------------------------------------------------------------------------------------------
#1    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#2    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#3    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#4    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#5    ADD    Yes    I#-1    sub    9    12    #0    #0    -1    Yes    No
#6    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#7    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#8    MUL    Yes    I#-1    mul    9    4    #0    #0    -1    Yes    Yes
#9    MUL    Yes    I#-1    div    -1    12    #8    #0    -1    No    No
================================================================
=======================
=============================================================
Registers
----------------------------------------------------------------------
        r0    r1    r2    r3    r4    r5    r6    r7
----------------------------------------------------------------------
val    0    1    9    3    4    5    12    7
Qi    #8    #0    #0    #0    #0    #0    #0    #0
----------------------------------------------------------------------
        r8    r9    r10    r11    r12    r13    r14    r15
----------------------------------------------------------------------
val    8    9    10    11    12    13    14    15
Qi    #5    #0    #9    #0    #0    #0    #0    #0
----------------------------------------------------------------------
=============================================================
* CYCLE 6
================================================================
=======================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
----------------------------------------------------------------------------------------------------
#1    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#2    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#3    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#4    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#5    ADD    Yes    I#-1    sub    9    12    #0    #0    -1    Yes    Yes
```

```
#6   ADD   Yes   I#-1   add   36   12   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   div   36   12   #0   #0   -1   Yes   No
================================================================================
==========================
================================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   36   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #0   #9   #6   #0   #0   #0   #0
--------------------------------------------------------------------
================================================================================
* CYCLE 7
================================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   Yes   I#-1   add   36   12   #0   #0   -1   Yes   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   div   36   12   #0   #0   -1   Yes   No
================================================================================
==========================
================================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   36   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
```

```
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------------
val   -3   9   10   11   12   13   14   15
Qi   #0   #0   #9   #6   #0   #0   #0   #0
-----------------------------------------------------------------------
==========================================================================
* CYCLE 8
================================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   Yes   I#-1   add   36   12   #0   #0   -1   Yes   Yes
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   div   36   12   #0   #0   -1   Yes   No
================================================================================
==========================
==========================================================================
Registers
-----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------------
val   36   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------------
val   -3   9   10   11   12   13   14   15
Qi   #0   #0   #9   #6   #0   #0   #0   #0
-----------------------------------------------------------------------
==========================================================================
* CYCLE 9
================================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
```

```
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   div   36   12   #0   #0   -1   Yes   Yes
================================================================================
=========================
============================================================================
Registers
--------------------------------------------------------------------
         r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   36   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
         r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   -3   9   10   48   12   13   14   15
Qi   #0   #0   #9   #0   #0   #0   #0   #0
--------------------------------------------------------------------
============================================================================
* CYCLE 10
================================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
================================================================================
=========================
============================================================================
Registers
--------------------------------------------------------------------
         r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   36   1   9   3   4   5   12   7
```

```
Qi   #0   #0   #0   #0   #0   #0   #0   #0
------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
------------------------------------------------------------------
val   -3   9   3   48   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
------------------------------------------------------------------
====================================================================
```

Part 1 Given Instruction Console Output (LDLAT=2)

```
============= TEST INSTRUCTION SEQUENCE ===========
I#1   ld   r6,32(r12)
I#2   ld   r2,44(r13)
I#3   mul   r0,r2,r4
I#4   sub   r8,r2,r6
I#5   div   r10,r0,r6
I#6   add   r11,r0,r6
* CYCLE 0 (initial state)
========================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
---------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
========================================================================
=========================
====================================================================
Registers
------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
------------------------------------------------------------------
val   0   1   2   3   4   5   6   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
```

```
--------------------------------------------------------------------
========================================================================
* CYCLE 1
=============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------
#1   LD   Yes   I#-1   ld   12   -1   #0   #0   32   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
=============================================================================
==========================
========================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   0   1   2   3   4   5   6   7
Qi   #0   #0   #0   #0   #0   #0   #1   #0
--------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
========================================================================
* CYCLE 2
=============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------
#1   LD   Yes   I#-1   ld   12   -1   #0   #0   32   Yes   No
#2   LD   Yes   I#-1   ld   13   -1   #0   #0   44   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
```

```
#8    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#9    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
================================================================================
==========================
============================================================================
Registers
-------------------------------------------------------------------
        r0    r1    r2    r3    r4    r5    r6    r7
-------------------------------------------------------------------
val    0    1    2    3    4    5    6    7
Qi    #0    #0    #2    #0    #0    #0    #1    #0
-------------------------------------------------------------------
        r8    r9    r10    r11    r12    r13    r14    r15
-------------------------------------------------------------------
val    8    9    10    11    12    13    14    15
Qi    #0    #0    #0    #0    #0    #0    #0    #0
-------------------------------------------------------------------
============================================================================
* CYCLE 3
================================================================================
==========================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
-----------------------------------------------------------------------------------------------
#1    LD    Yes    I#-1    ld    12    -1    #0    #0    44    Yes    Yes
#2    LD    Yes    I#-1    ld    13    -1    #0    #0    44    No    No
#3    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#4    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#5    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#6    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#7    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#8    MUL    Yes    I#-1    mul    -1    4    #2    #0    -1    No    No
#9    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
================================================================================
==========================
============================================================================
Registers
-------------------------------------------------------------------
        r0    r1    r2    r3    r4    r5    r6    r7
-------------------------------------------------------------------
val    0    1    2    3    4    5    6    7
Qi    #8    #0    #2    #0    #0    #0    #1    #0
-------------------------------------------------------------------
        r8    r9    r10    r11    r12    r13    r14    r15
-------------------------------------------------------------------
```

```
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
==================================================================
* CYCLE 4
=====================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   Yes   I#-1   ld   13   -1   #0   #0   44   Yes   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   sub   -1   12   #2   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   mul   -1   4   #2   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
=====================================================================
=========================
==================================================================
Registers
--------------------------------------------------------------------
          r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   0   1   2   3   4   5   12   7
Qi   #8   #0   #2   #0   #0   #0   #0   #0
--------------------------------------------------------------------
          r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
==================================================================
* CYCLE 5
=====================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   Yes   I#-1   ld   13   -1   #0   #0   57   Yes   Yes
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   sub   -1   12   #2   #0   -1   No   No
```

```
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   mul   -1   4   #2   #0   -1   No   No
#9   MUL   Yes   I#-1   div   -1   12   #8   #0   -1   No   No
============================================================================
=========================
=========================================================================
Registers
-------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-------------------------------------------------------------------
val   0   1   2   3   4   5   12   7
Qi   #8   #0   #2   #0   #0   #0   #0   #0
-------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #0   #9   #0   #0   #0   #0   #0
-------------------------------------------------------------------
=========================================================================
* CYCLE 6
============================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   sub   9   12   #0   #0   -1   Yes   No
#6   ADD   Yes   I#-1   add   -1   12   #8   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   mul   9   4   #0   #0   -1   Yes   No
#9   MUL   Yes   I#-1   div   -1   12   #8   #0   -1   No   No
============================================================================
=========================
=========================================================================
Registers
-------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-------------------------------------------------------------------
val   0   1   9   3   4   5   12   7
Qi   #8   #0   #0   #0   #0   #0   #0   #0
-------------------------------------------------------------------
```

```
        r8   r9   r10   r11   r12   r13   r14   r15
-------------------------------------------------------------------
val   8    9    10    11    12    13    14    15
Qi   #5   #0   #9   #6   #0   #0   #0   #0
-------------------------------------------------------------------
=========================================================================
* CYCLE 7
============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
--------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   sub   9   12   #0   #0   -1   Yes   Yes
#6   ADD   Yes   I#-1   add   -1   12   #8   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   mul   9   4   #0   #0   -1   Yes   Yes
#9   MUL   Yes   I#-1   div   -1   12   #8   #0   -1   No   No
============================================================================
==========================
=========================================================================
Registers
-------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-------------------------------------------------------------------
val   0    1    9    3    4    5    12    7
Qi   #8   #0   #0   #0   #0   #0   #0   #0
-------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-------------------------------------------------------------------
val   8    9    10    11    12    13    14    15
Qi   #5   #0   #9   #6   #0   #0   #0   #0
-------------------------------------------------------------------
=========================================================================
* CYCLE 8
============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
--------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
```

```
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   Yes   I#-1   add   36   12   #0   #0   -1   Yes   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   div   36   12   #0   #0   -1   Yes   No
================================================================================
==========================
============================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   36   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   -3   9   10   11   12   13   14   15
Qi   #0   #0   #9   #6   #0   #0   #0   #0
--------------------------------------------------------------------
============================================================================
* CYCLE 9
================================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   Yes   I#-1   add   36   12   #0   #0   -1   Yes   Yes
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   div   36   12   #0   #0   -1   Yes   No
================================================================================
==========================
============================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   36   1   9   3   4   5   12   7
```

```
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------------
val   -3   9   10   11   12   13   14   15
Qi   #0   #0   #9   #6   #0   #0   #0   #0
-----------------------------------------------------------------------
================================================================================
* CYCLE 10
====================================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   div   36   12   #0   #0   -1   Yes   No
====================================================================================
==========================
================================================================================
Registers
-----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------------
val   36   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------------
val   -3   9   10   48   12   13   14   15
Qi   #0   #0   #9   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
================================================================================
* CYCLE 11
====================================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
```

```
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   div   36   12   #0   #0   -1   Yes   Yes
==========================================================================
==========================
======================================================================
Registers
----------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------
val   36   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
----------------------------------------------------------------
val   -3   9   10   48   12   13   14   15
Qi   #0   #0   #9   #0   #0   #0   #0   #0
----------------------------------------------------------------
======================================================================
* CYCLE 12
==========================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
==========================================================================
==========================
======================================================================
Registers
----------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
```

```
----------------------------------------------------------------------
val    36    1    9    3    4    5    12    7
Qi    #0    #0    #0    #0    #0    #0    #0    #0
----------------------------------------------------------------------
          r8    r9    r10    r11    r12    r13    r14    r15
----------------------------------------------------------------------
val    -3    9    3    48    12    13    14    15
Qi    #0    #0    #0    #0    #0    #0    #0    #0
----------------------------------------------------------------------
================================================================================
```

PART 1 CODE PERSONAL INSTRUCTIONS

```
============== TEST INSTRUCTION SEQUENCE ===========
I#1    ld    r6,32(r12)
I#2    ld    r2,44(r13)
I#3    div    r0,r2,r4
I#4    add    r8,r0,r3
I#5    sub    r7,r0,r1
I#6    mul    r9,r7,r6
I#7    st    r8,0(r8)
* CYCLE 0 (initial state)
===================================================================================
==========================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
----------------------------------------------------------------------------------------------
#1    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#2    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#3    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#4    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#5    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#6    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#7    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#8    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#9    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
===================================================================================
==========================
================================================================================
Registers
----------------------------------------------------------------------
          r0    r1    r2    r3    r4    r5    r6    r7
----------------------------------------------------------------------
val    0    1    2    3    4    5    6    7
Qi    #0    #0    #0    #0    #0    #0    #0    #0
```

```
----------------------------------------------------------------------
        r8    r9    r10    r11    r12    r13    r14    r15
----------------------------------------------------------------------
val    8    9    10    11    12    13    14    15
Qi    #0    #0    #0    #0    #0    #0    #0    #0
----------------------------------------------------------------------
==========================================================================
* CYCLE 1
==============================================================================
==========================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
----------------------------------------------------------------------------------------------
#1    LD    Yes    I#-1    ld    12    -1    #0    #0    32    No    No
#2    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#3    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#4    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#5    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#6    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#7    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#8    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#9    MUL    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
==============================================================================
==========================
==========================================================================
Registers
----------------------------------------------------------------------
        r0    r1    r2    r3    r4    r5    r6    r7
----------------------------------------------------------------------
val    0    1    2    3    4    5    6    7
Qi    #0    #0    #0    #0    #0    #0    #1    #0
----------------------------------------------------------------------
        r8    r9    r10    r11    r12    r13    r14    r15
----------------------------------------------------------------------
val    8    9    10    11    12    13    14    15
Qi    #0    #0    #0    #0    #0    #0    #0    #0
----------------------------------------------------------------------
==========================================================================
* CYCLE 2
==============================================================================
==========================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
----------------------------------------------------------------------------------------------
#1    LD    Yes    I#-1    ld    12    -1    #0    #0    44    Yes    Yes
#2    LD    Yes    I#-1    ld    13    -1    #0    #0    44    No    No
```

```
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
==================================================================================================
==================================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   0   1   2   3   4   5   6   7
Qi   #0   #0   #2   #0   #0   #0   #1   #0
--------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
==================================================================================
* CYCLE 3
==================================================================================================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   Yes   I#-1   ld   13   -1   #0   #0   57   Yes   Yes
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   div   -1   4   #2   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
==================================================================================================
==================================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
```

```
val   0   1   2   3   4   5   12   7
Qi   #8   #0   #2   #0   #0   #0   #0   #0
---------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
---------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
---------------------------------------------------------------------
=========================================================================
* CYCLE 4
==============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   add   -1   3   #8   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   div   9   4   #0   #0   -1   Yes   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
==============================================================================
==========================
=========================================================================
Registers
---------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
---------------------------------------------------------------------
val   0   1   9   3   4   5   12   7
Qi   #8   #0   #0   #0   #0   #0   #0   #0
---------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
---------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #0   #0   #0   #0   #0   #0   #0
---------------------------------------------------------------------
=========================================================================
* CYCLE 5
==============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
```

```
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   add   -1   3   #8   #0   -1   No   No
#6   ADD   Yes   I#-1   sub   -1   1   #8   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   div   9   4   #0   #0   -1   Yes   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
==============================================================================
============================
==========================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   0   1   9   3   4   5   12   7
Qi   #8   #0   #0   #0   #0   #0   #0   #6
--------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
==========================================================================
* CYCLE 6
==============================================================================
============================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   add   -1   3   #8   #0   -1   No   No
#6   ADD   Yes   I#-1   sub   -1   1   #8   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   div   9   4   #0   #0   -1   Yes   No
#9   MUL   Yes   I#-1   mul   -1   12   #6   #0   -1   No   No
==============================================================================
============================
==========================================================================
Registers
--------------------------------------------------------------------
```

```
        r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------------
val   0   1   9   3   4   5   12   7
Qi   #8   #0   #0   #0   #0   #0   #0   #6
----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
----------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #9   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------------
=======================================================================
* CYCLE 7
============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   Yes   I#-1   st   -1   -1   #5   #5   0   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   add   -1   3   #8   #0   -1   No   No
#6   ADD   Yes   I#-1   sub   -1   1   #8   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   div   9   4   #0   #0   -1   Yes   Yes
#9   MUL   Yes   I#-1   mul   -1   12   #6   #0   -1   No   No
============================================================================
==========================
=======================================================================
Registers
----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------------
val   0   1   9   3   4   5   12   7
Qi   #8   #0   #0   #0   #0   #0   #0   #6
----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
----------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #9   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------------
=======================================================================
* CYCLE 8
============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   Yes   I#-1   st   -1   -1   #5   #5   0   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   add   2   3   #0   #0   -1   Yes   No
#6   ADD   Yes   I#-1   sub   2   1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   mul   -1   12   #6   #0   -1   No   No
=============================================================================
===========================
=============================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   2   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #6
--------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #9   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
=============================================================================
* CYCLE 9
=============================================================================
===========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   Yes   I#-1   st   -1   -1   #5   #5   0   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   add   2   3   #0   #0   -1   Yes   Yes
#6   ADD   Yes   I#-1   sub   2   1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   mul   -1   12   #6   #0   -1   No   No
=============================================================================
===========================
=============================================================================
Registers
-----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------------
val   2   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #6
-----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #5   #9   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
=================================================================
* CYCLE 10
====================================================================
========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   Yes   I#-1   st   5   5   #0   #0   5   Yes   Yes
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   Yes   I#-1   sub   2   1   #0   #0   -1   Yes   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   mul   -1   12   #6   #0   -1   No   No
====================================================================
========================
=================================================================
Registers
-----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------------
val   2   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #6
-----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------------
val   5   9   10   11   12   13   14   15
Qi   #0   #9   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
=================================================================
* CYCLE 11
```

```
================================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
--------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   Yes   I#-1   sub   2   1   #0   #0   -1   Yes   Yes
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   mul   -1   12   #6   #0   -1   No   No
================================================================================
=========================
============================================================================
Registers
-------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-------------------------------------------------------------------
val   2   1   9   3   4   5   12   7
Qi   #0   #0   #0   #0   #0   #0   #0   #6
-------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-------------------------------------------------------------------
val   5   9   10   11   12   13   14   15
Qi   #0   #9   #0   #0   #0   #0   #0   #0
-------------------------------------------------------------------
============================================================================
* CYCLE 12
================================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
--------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   mul   1   12   #0   #0   -1   Yes   No
```

```
==================================================================================
========================
==============================================================================
Registers
-----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------------
val   2   1   9   3   4   5   12   1
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------------
val   5   9   10   11   12   13   14   15
Qi   #0   #9   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
==============================================================================
* CYCLE 13
==================================================================================
========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   Yes   I#-1   mul   1   12   #0   #0   -1   Yes   Yes
==================================================================================
========================
==============================================================================
Registers
-----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------------
val   2   1   9   3   4   5   12   1
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------------
val   5   9   10   11   12   13   14   15
Qi   #0   #9   #0   #0   #0   #0   #0   #0
```

```
----------------------------------------------------------------------
=================================================================
* CYCLE 14
====================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
====================================================================
=========================
=================================================================
Registers
----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------------
val   2   1   9   3   4   5   12   1
Qi   #0   #0   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
----------------------------------------------------------------------
val   5   12   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------------
=================================================================
```

Part 3 inst.c

```
#include "inst.h"
#include <stdio.h>

INST inst[NUM_OF_INST]; /* instruction array */
/******************************
* instruction initialization
******************************/
void init_inst()
```

```
{
 inst[0].num=1; inst[0].op=LD; inst[0].rd=0; inst[0].rs=1; inst[0].rt=0;
// ld r0,0(r1)
 inst[1].num=2; inst[1].op=MUL; inst[1].rd=4; inst[1].rs=0; inst[1].rt=2;
// mul r4,r0, r2
 inst[2].num=3; inst[2].op=ST; inst[2].rd=4; inst[2].rs=1; inst[2].rt=0;
// st r4, 0(r1)
 inst[3].num=4; inst[3].op=ADDI; inst[3].rd=1; inst[3].rs=1; inst[3].rt=1;
// addi r1, r1, 1
 inst[4].num=5; inst[4].op=BNE; inst[4].rd=1; inst[4].rs=2; inst[4].rt=6;
// bne r1, r2, loop
 return;
}

/* print an instruction */
void print_inst(INST ins) {
 printf("I#%d\t",ins.num);
 if(ins.op==ADD) printf("add\tr%d,r%d,r%d\n",ins.rd,ins.rs,ins.rt);
 else if(ins.op==SUB) printf("sub\tr%d,r%d,r%d\n",ins.rd,ins.rs,ins.rt);
 else if(ins.op==MUL) printf("mul\tr%d,r%d,r%d\n",ins.rd,ins.rs,ins.rt);
 else if(ins.op==DIV) printf("div\tr%d,r%d,r%d\n",ins.rd,ins.rs,ins.rt);
 else if(ins.op==LD) printf("ld\tr%d,%d(r%d)\n",ins.rd,ins.rt,ins.rs);
 else if(ins.op==ST) printf("st\tr%d,%d(r%d)\n",ins.rd,ins.rt,ins.rs);
 else if(ins.op==ADDI) printf("addi\tr%d,%d(r%d)\n",ins.rd,ins.rt,ins.rs);
 else if(ins.op==BNE) printf("bne\tr%d,%d(r%d)\n",ins.rd,ins.rt,ins.rs);
 else printf("unknown\n");
}

void print_program() {
 int i=0;
 for(i=0;i<NUM_OF_INST;i++) print_inst(inst[i]);
}
```

Part 3 inst.h

```
/*********************************************
 operations
*********************************************/
```

```
#define NUM_OF_OP_TYPES 8
enum op_type {ADD, SUB, MUL, DIV, LD, ST, ADDI, BNE };

/* data structure for an instruction */
typedef struct instruction {
 int num; /* number: starting from 1 */
 enum op_type op; /* operation type */
 int rd; /* destination register id */
 int rs; /* source regsiter id or base register for ld/st */
 int rt; /* target regsiter id or addr offset for ld/st */
} INST;


#define NUM_OF_INST 5
extern INST inst[NUM_OF_INST]; /* instruction array */

void init_inst();
void print_inst(INST ins);
void print_program();
```

Part 3 arch.c

```
#include <stdio.h>
#include "arch.h"

REG regs[NUM_REGS];
int mem_arr[MEM_SIZE];
RS rs_array[NUM_RS_ENTRIES];
ROB rob_array[ROB_SIZE];

bool is_add_available;
bool is_mul_available;
bool is_mem_available;
bool is_bne_available;

bool is_rs_active()
{
 int i;
 for(i=0;i<NUM_RS_ENTRIES;i++)
```

```
   if(rs_array[i].is_busy) return true;
 return false;
}

void set_mem(int addr, int val)
{
 mem_arr[addr] = val;
 return;
}

int get_mem(int addr)
{
 return mem_arr[addr];
}

void init_mem()
{
 int i;
 for(i=0;i<MEM_SIZE;i++) mem_arr[i] = i%16;
 return;
}

void init_fu()
{
 is_add_available=true;
 is_mul_available=true;
 is_mem_available=true;
 is_bne_available=true;
 return;
}

void init_regs()
{
 int i=0;
 for(i=0;i<NUM_REGS;i++)
 {
   regs[i].num = i;
   regs[i].val = i;
   regs[i].Qi = 0;
 }
```

```
}

void init_rob()
{
 int i=0;
 for(i=0;i<ROB_SIZE;i++)
 {
   rob_array[i].op = i;
   rob_array[i].destination = i;
   rob_array[i].value = i;
   rob_array[i].ready = true;
 }
}
bool is_rob_available(){
 int i=0;
 for(i=0;i<ROB_SIZE;i++){
   if (rob_array[i].ready = true) return true;
 }
 return false;
}

void print_regs()
{
 int i=0;

printf("===================================================================\n");
 printf("Registers\n");

printf("-------------------------------------------------------------------\n");
 for(i=0;i<NUM_REGS;i++)
 {
   if(i%8==7) {
     printf(" \tr%d\tr%d\tr%d\tr%d\tr%d\tr%d\tr%d\tr%d\n",
             regs[i-7].num,
             regs[i-6].num,
             regs[i-5].num,
             regs[i-4].num,
             regs[i-3].num,
             regs[i-2].num,
             regs[i-1].num,
             regs[i].num);

printf("---------------------------------------------------------------------
---\n");
     printf("val\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n",
             regs[i-7].val,
             regs[i-6].val,
             regs[i-5].val,
             regs[i-4].val,
             regs[i-3].val,
             regs[i-2].val,
             regs[i-1].val,
             regs[i].val);
     printf("Qi\t#%d\t#%d\t#%d\t#%d\t#%d\t#%d\t#%d\t#%d\n",
             regs[i-7].Qi,
             regs[i-6].Qi,
             regs[i-5].Qi,
             regs[i-4].Qi,
             regs[i-3].Qi,
             regs[i-2].Qi,
             regs[i-1].Qi,
             regs[i].Qi);

printf("---------------------------------------------------------------------
---\n");
   }
 }

printf("=====================================================================
===\n");
 return;
 }

void reset_rs_entry(RS * t)
{
   t->is_busy = false;
   t->op = -1;
   t->Qj = 0;
   t->Qk = 0;
   t->Vj = -1;
   t->Vk = -1;
   t->A = -1;
   t->exec_cycles = -1;
   t->result = -1;
   t->in_exec = false;
   t->is_result_ready = false;
   t->inst_num = -1;
   return;
}


/**************************************
* initialize RS
***************************************/
void init_rs()
{
 int i;
 int curr_ind=0;
 RS * curr_entry;
  /* common for all */
 for(i=0;i<NUM_RS_ENTRIES;i++) {
   curr_entry = &(rs_array[i]);
   curr_entry->id = i+1;
   curr_entry->is_busy = false;
   curr_entry->op = -1;
   curr_entry->Qj = 0;
   curr_entry->Qk = 0;
   curr_entry->Vj = -1;
   curr_entry->Vk = -1;
   curr_entry->A = -1;
   curr_entry->exec_cycles = -1;
   curr_entry->result = -1;
   curr_entry->in_exec = false;
   curr_entry->is_result_ready = false;
   curr_entry->inst_num = -1;
 }
 /* types */
 for(i=0;i<NUM_LD_BUF;i++) rs_array[curr_ind++].type=LD_BUF;
```

```c
 for(i=0;i<NUM_ST_BUF;i++) rs_array[curr_ind++].type=ST_BUF;
 for(i=0;i<NUM_ADD_RS;i++) rs_array[curr_ind++].type=ADD_RS;
 for(i=0;i<NUM_MUL_RS;i++) rs_array[curr_ind++].type=MUL_RS;
 for(i=0;i<NUM_BNE_RS;i++) rs_array[curr_ind++].type=BNE_RS;

}

/*************************************
* print RS
**************************************/
void print_rs()
{
 int i;


printf("========================================================================================================\n");
 printf("RS_id\ttype\tBusy\tinst#\tOp\tVj\tVk\tQi\tQj\tA\tExec\tDone\n");

printf("--------------------------------------------------------------------------------------------------------\n");
 for(i=0;i<NUM_RS_ENTRIES;i++) {
   printf("#%d\t",rs_array[i].id); // id
   if(rs_array[i].type==LD_BUF) printf("LD\t"); // RS buff type
   else if(rs_array[i].type==ST_BUF) printf("ST\t");
   else if(rs_array[i].type==ADD_RS) printf("ADD\t");
   else if(rs_array[i].type==MUL_RS) printf("MUL\t");
   else if(rs_array[i].type==BNE_RS) printf("BNE\t");
   else printf("NONE\t");
   if(rs_array[i].is_busy) printf("Yes\t"); else printf("No\t"); // Busy
   printf("I#%d\t",rs_array[i].inst_num);
   if(rs_array[i].op==ADD) printf("add\t"); // instr type
   else if(rs_array[i].op==SUB) printf("sub\t");
   else if(rs_array[i].op==MUL) printf("mul\t");
   else if(rs_array[i].op==DIV) printf("div\t");
   else if(rs_array[i].op==LD) printf("ld\t");
   else if(rs_array[i].op==ST) printf("st\t");
   else if(rs_array[i].op==ADDI) printf("st\t");
   else if(rs_array[i].op==BNE) printf("st\t");
   else printf("NONE\t");

printf("%d\t%d\t#%d\t#%d\t%d\t",rs_array[i].Vj,rs_array[i].Vk,rs_array[i].
Qj,rs_array[i].Qk,rs_array[i].A);
   if(rs_array[i].in_exec) printf("Yes\t"); else printf("No\t"); // in
Execution?
   if(rs_array[i].is_result_ready) printf("Yes\n"); else printf("No\n");
// done Execution?

 }

printf("===================================================================
=================================\n");
  return;
}


int obtain_available_rs(enum rs_type t)
{
 int i;
 for(i=0;i<NUM_RS_ENTRIES;i++) {
   if((rs_array[i].type==t) && !rs_array[i].is_busy) return
rs_array[i].id;
 }
 return -1; /* no available RS entriy */
}

RS * get_rs(int id)
{
 int i=0;
 for(i=0;i<NUM_RS_ENTRIES;i++) {
   if(rs_array[i].id == id) return &(rs_array[i]);
 }
 return NULL;
}
ROB * get_ROB(int index)
{
 int i=0;
 for(i=0;i<NUM_RS_ENTRIES;i++) {
   if(rob_array[i].destination == index) return &(rs_array[i]);
 }
 return NULL;
}
```

Part 3 arch.h

```
#include <stdbool.h>
#include "inst.h"
/*********************************************
  architecture definition
*********************************************/
/* num of RS or load/store buffer entries  */
#define ROB_SIZE 16

enum rs_type {LD_BUF, ST_BUF, ADD_RS, MUL_RS, BNE_RS};
enum rob_type {BRANCH,MEM,FP};
#define NUM_LD_BUF 2
#define NUM_ST_BUF 2
#define NUM_ADD_RS 3
#define NUM_MUL_RS 2
#define NUM_BNE_RS 2

#define NUM_RS_ENTRIES
(NUM_LD_BUF+NUM_ST_BUF+NUM_ADD_RS+NUM_MUL_RS+NUM_BNE_RS)

extern bool is_add_available;
extern bool is_mul_available;
extern bool is_mem_available;
extern bool is_bne_available;


#define MEM_SIZE 1024 /* memory size in word */
extern int mem_arr[MEM_SIZE];

/* registers */
#define NUM_REGS 16

/* execution unit latencies */
```

```
#define LAT_ADD 2 /* executed on ADD */
#define LAT_SUB 2 /* executed on ADD */
#define LAT_MUL 2 /* executed on MUL */
#define LAT_DIV 4 /* executed on MUL */
#define LAT_LD 1 /* executed on Memory Unit */
#define LAT_ST 1 /* executed on Memory Unit */

#define LAT_ADDI 2 /* executed on ADDI */
#define LAT_BNE 4 /* executed on BNE */


typedef struct a_reg {
 int num; /* register id starting from 0 */
 int val; /* value */
 int reorder;
 bool busy;
 int Qi; /* the number of the RS entry that contains the operation whose
result should be stored into this reg */
} REG;

typedef struct reorder_buffer
{
 enum op_type op; //ADD, SUB, MUL, DIV, LD, ST, ADDI, BNE
 int destination; /*register number or memory address*/
 int value;
 bool ready;
} ROB;

extern ROB rob_array[ROB_SIZE];
extern REG regs[NUM_REGS];


/* data structure for an RS/LB/SB entry */
typedef struct reservation_station {
 int id;   /* RS id starting from 1 */
 bool is_busy;   /* is busy? */
 enum rs_type type;    /* 0:LD, 1:ST, 2:ADD, 3:MUL 4:BNE*/
 enum op_type op;  /* ADD, SUB, MUL, DIV, LD, ST, ADDI, BNE */
 int Qj, Qk;     /* ids of the RS entries */
 int Vj, Vk;     /* values */
 int A;     /* address for LD/ST */
 int exec_cycles;  /* remaining execution cycles */
 int result;   /* result */
 bool in_exec;   /* is in execution? */
 bool is_result_ready; /* is result ready? */
 int inst_num;   /* instruction number */
 int destination;
} RS;

extern RS rs_array[NUM_RS_ENTRIES];


void set_mem(int addr, int val);
int get_mem(int addr);
void init_mem();

void init_regs();
void print_regs();

void init_fu();

void init_rs();
void print_rs();

void init_rob();

bool is_rob_available();
int obtain_available_rs(enum rs_type t);
RS * get_rs(int id);
bool is_rs_active();
void reset_rs_entry(RS * t);
```

Part 3 tomasulo.c

```
#include <stdio.h>
#include <stdlib.h>
#include "arch.h"
```

```
/*******************************
main
*******************************/
int main()
{
 int i,j;
 int done = 0;
 int cycle = 0;
 int num_issued_inst = 0;
 int num_rob_entries=0;
 int h=0;

 init_inst();
 init_fu();

 printf("============== TEST INSTRUCTION SEQUENCE ===========\n");
 print_program();

 init_rs();  // initialize RS entries
 init_regs();  // initalize registers
 init_mem(); // initialize memory

 printf("* CYCLE %d (initial state)\n",cycle);
 print_rs(); // print initial RS state
 print_regs(); // print initial register state

/*
1:ld1
2:ld2
3:st1
4:st2
5:add1
6:add2
7:add3
8:mul1
9:mul2
*/
 /* simulation loop main */
 while(!done){
   /* increment the cycle */
   cycle++;
   /********************************
    *     Step IV: Commit
    ********************************/




    /********************************
    *     Step III: Write result
    ********************************/
    for(i=0;i<NUM_RS_ENTRIES;i++) {
      // complete STEP III here
     if(rs_array[i].is_result_ready){//if execution is complete at R
       if((rs_array[i].op==ST)&& (rs_array[i].Qk==0)){ //if station
==store and RS[r].Qk==0
         set_mem(rs_array[i].A,rs_array[i].Vk);
         if(!is_mem_available){
           is_mem_available=true;
         }
         reset_rs_entry(&rs_array[i]);
       }
       else{//if station ==fp or load and cdb is available
         int j;
         for(j=0;j<16 ;j++){//j is num of reg
           if(regs[j].Qi==rs_array[i].id){
             regs[j].val=rs_array[i].result;
             regs[j].Qi=0;
           }
         }
         int k;
         for(k=0;k<NUM_RS_ENTRIES;k++){//j is num of reg
           if(rs_array[k].Qj==rs_array[i].id){
             rs_array[k].Vj=rs_array[i].result;
             rs_array[k].Qj=0;
           }
           if(rs_array[k].Qk==rs_array[i].id){
             rs_array[k].Vk=rs_array[i].result;
             rs_array[k].Qk=0;
           }
         }

if((rs_array[i].op==ADD)||(rs_array[i].op==SUB)||(rs_array[i].op==BNE)||(r
s_array[i].op==ADDI))  {
           if(!is_add_available)
           is_add_available=true;
         }
         if((rs_array[i].op==MUL)||(rs_array[i].op==DIV))  {
           if(!is_mul_available)
           is_mul_available=true;
         }
         if((rs_array[i].op==LD))  {
           if(!is_mem_available)
           is_mem_available=true;
         }
         reset_rs_entry(&rs_array[i]);
       }
     }
   }

   /********************************
    *     Step II: Execute
    ********************************/
     for(i=0;i<NUM_RS_ENTRIES;i++) {
      // complete STEP II here
     bool eq0=(rs_array[i].Qj ==0) && (rs_array[i].Qk ==0);
     if(((rs_array[i].op ==ADD)||(rs_array[i].op ==ADDI))&& eq0){//if fp
rs and RS[r].Qj ==0 and RS[r].Qk ==0
       if(is_add_available){
         is_add_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].result = rs_array[i].Vj + rs_array[i].Vk;
         rs_array[i].is_result_ready=true;
       }
      }
     if(((rs_array[i].op ==SUB)||(rs_array[i].op ==BNE))&& eq0){//if fp rs
and RS[r].Qj ==0 and RS[r].Qk ==0
       if(is_add_available){
         is_add_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].result = rs_array[i].Vj - rs_array[i].Vk;
         rs_array[i].is_result_ready=true;
       }
      }

     if((rs_array[i].op ==MUL)&& eq0){//if fp rs and RS[r].Qj ==0 and
RS[r].Qk ==0
       if(is_mul_available){
         is_mul_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].result = rs_array[i].Vj * rs_array[i].Vk;
         rs_array[i].is_result_ready=true;
       }
      }
     if((rs_array[i].op ==DIV)&& eq0){//if fp rs and RS[r].Qj ==0 and
RS[r].Qk ==0
       if(is_mul_available){
         is_mul_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].result = rs_array[i].Vj / rs_array[i].Vk;
         rs_array[i].is_result_ready=true;
       }
      }
     if((rs_array[i].op ==LD)&& (rs_array[i].Qj==0)){//if fp rs and
RS[r].Qj ==0 and RS[r].Qk ==0
       if(is_mem_available){
         is_mem_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].A = rs_array[i].Vj +rs_array[i].A;
         rs_array[i].result = get_mem(rs_array[i].A);
         rs_array[i].is_result_ready=true;
       }
     }
     if((rs_array[i].op ==ST)&& (rs_array[i].Qj==0)){//if fp rs and
RS[r].Qj ==0 and RS[r].Qk ==0
       if(is_mem_available){
         is_mem_available=false;
         rs_array[i].in_exec=true;
       }
       if(rs_array[i].in_exec)
         rs_array[i].exec_cycles--;
       if(rs_array[i].exec_cycles==0){
         rs_array[i].A = rs_array[i].Vj +rs_array[i].A;
         rs_array[i].is_result_ready=true;
       }
     }

     }

   /********************************
    *     Step I: Issue
    ********************************/

   /*  wait if no RS entry is available */
   if(num_issued_inst < NUM_OF_INST) {
     int cand_rs_id;
     if(inst[num_issued_inst].op==ADD) cand_rs_id =
obtain_available_rs(ADD_RS);
```

```
     else if(inst[num_issued_inst].op==SUB) cand_rs_id =
obtain_available_rs(ADD_RS);
     else if(inst[num_issued_inst].op==MUL) cand_rs_id =
obtain_available_rs(MUL_RS);
     else if(inst[num_issued_inst].op==DIV) cand_rs_id =
obtain_available_rs(MUL_RS);
     else if(inst[num_issued_inst].op==LD) cand_rs_id =
obtain_available_rs(LD_BUF);
     else if(inst[num_issued_inst].op==ST) cand_rs_id =
obtain_available_rs(ST_BUF);
     else if(inst[num_issued_inst].op==ADDI) cand_rs_id =
obtain_available_rs(ADD_RS);
     else if(inst[num_issued_inst].op==BNE) cand_rs_id =
obtain_available_rs(BNE_RS);
     if(num_rob_entries<ROB_SIZE){
       int h=0;
     }

     /* if there is an available RS entry AND ROB ENTRY AVAILABLE*/
     if(cand_rs_id!=-1 && is_rob_available()) {//ROB ARRAY CHEKC
       /* issue the instruction: See Fig. 3.13 */
       RS * curr_rs = get_rs(cand_rs_id);

       if(curr_rs ==NULL) {
         printf("NO RS found with the given id\n");
         exit(1);
       }

       /* normal ALU operations */
       if(inst[num_issued_inst].op==ADD || inst[num_issued_inst].op==SUB
|| inst[num_issued_inst].op==BNE||
         inst[num_issued_inst].op== MUL ||  inst[num_issued_inst].op==DIV
|| inst[num_issued_inst].op==ADDI) {
         int rd, rs, rt;
         rd = inst[num_issued_inst].rd;
         rs = inst[num_issued_inst].rs;
         rt = inst[num_issued_inst].rt;

         /* Rs */
         if(regs[rs].Qi!=0) curr_rs->Qj = regs[rs].Qi;
         else curr_rs->Vj = regs[rs].val;

         /* Rt */
         if(regs[rt].Qi!=0) curr_rs->Qk = regs[rt].Qi;
         else curr_rs->Vk = regs[rt].val;

         /* set busy */
         curr_rs->is_busy = true;
         curr_rs->op = inst[num_issued_inst].op;

         /* register update */
         regs[rd].Qi = curr_rs->id;

         /* set exec cycles */
         if(inst[num_issued_inst].op==ADD) curr_rs->exec_cycles=LAT_ADD;
         else if(inst[num_issued_inst].op==SUB)
curr_rs->exec_cycles=LAT_SUB;
         else if(inst[num_issued_inst].op==MUL)
curr_rs->exec_cycles=LAT_MUL;
         else if(inst[num_issued_inst].op==DIV)
curr_rs->exec_cycles=LAT_DIV;
         else if(inst[num_issued_inst].op==ADDI)
curr_rs->exec_cycles=LAT_ADDI;
         else if(inst[num_issued_inst].op==BNE)
curr_rs->exec_cycles=LAT_BNE;

         /* num issued ++ */
         num_issued_inst++;

       } else if (inst[num_issued_inst].op==LD) {

         int rd, rs, imm;
         rd = inst[num_issued_inst].rd;
         rs = inst[num_issued_inst].rs;
         imm = inst[num_issued_inst].rt;

         /* Rs */
         if(regs[rs].Qi!=0) curr_rs->Qj = regs[rs].Qi;
         else curr_rs->Vj = regs[rs].val;
```

```
      /* addr */
      curr_rs->A = imm;

      /* set busy */
      curr_rs->is_busy = true;
      curr_rs->op = inst[num_issued_inst].op;

      /* set exec cycles */
      curr_rs->exec_cycles=LAT_LD;

      /* register update */
      regs[rd].Qi = curr_rs->id;

      /* num issued ++ */
      num_issued_inst++;

    } else if (inst[num_issued_inst].op==ST) {

      int rd, rs, imm;
      rd = inst[num_issued_inst].rd;
      rs = inst[num_issued_inst].rs;
      imm = inst[num_issued_inst].rt;

      /* Rs */
      if(regs[rs].Qi!=0) curr_rs->Qj = regs[rs].Qi;
      else curr_rs->Vj = regs[rs].val;

      /* Rd */
      if(regs[rd].Qi!=0) curr_rs->Qk = regs[rd].Qi;
      else curr_rs->Vk = regs[rd].val;

      /* addr */
      curr_rs->A = imm;

      /* set busy */
      curr_rs->is_busy = true;
      curr_rs->op = inst[num_issued_inst].op;

      /* set exec cycles */
      curr_rs->exec_cycles=LAT_ST;
```

```
         /* num issued ++ */
         num_issued_inst++;
       }
     }
   }


   /* print out the result */
   printf("* CYCLE %d\n",cycle);
   print_rs();
   print_regs();

   /* check the termination condition */
   if( (num_issued_inst>=NUM_OF_INST) && !is_rs_active())
     done =1;
 }
 return 0;
}
```

Part 3 console output

```
[lkim@linuxvdi14 tomasulo p3]$ ./tomasulo
============== TEST INSTRUCTION SEQUENCE ===========
I#1    ld    r0,0(r1)
I#2    mul    r4,r0,r2
I#3    st    r4,0(r1)
I#4    addi    r1,1(r1)
I#5    bne    r1,6(r2)
* CYCLE 0 (initial state)
======================================================================
==========================
RS_id    type    Busy    inst#    Op    Vj    Vk    Qi    Qj    A    Exec    Done
----------------------------------------------------------------------------------------------------
#1    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#2    LD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#3    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#4    ST    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#5    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
#6    ADD    No    I#-1    NONE    -1    -1    #0    #0    -1    No    No
```

```
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
================================================================================
========================
============================================================================
Registers
-------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-------------------------------------------------------------------
val   0   1   2   3   4   5   6   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-------------------------------------------------------------------
============================================================================
* CYCLE 1
================================================================================
========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   Yes   I#-1   ld   1   -1   #0   #0   0   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
================================================================================
========================
============================================================================
Registers
-------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-------------------------------------------------------------------
```

```
val   0   1   2   3   4   5   6   7
Qi   #1   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
========================================================================
* CYCLE 2
============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-------------------------------------------------------------------------------------------
#1   LD   Yes   I#-1   ld   1   -1   #0   #0   1   Yes   Yes
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   mul   -1   2   #1   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
============================================================================
==========================
========================================================================
Registers
--------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   0   1   2   3   4   5   6   7
Qi   #1   #0   #0   #0   #8   #0   #0   #0
--------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
========================================================================
* CYCLE 3
============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   Yes   I#-1   st   1   -1   #0   #8   0   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   mul   1   2   #0   #0   -1   Yes   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
===============================================================================
==========================
======================================================================
Registers
-------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-------------------------------------------------------------------
val   1   1   2   3   4   5   6   7
Qi   #0   #0   #0   #0   #8   #0   #0   #0
-------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-------------------------------------------------------------------
======================================================================
* CYCLE 4
===============================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   Yes   I#-1   st   1   -1   #0   #8   1   Yes   Yes
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   st   1   1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   Yes   I#-1   mul   1   2   #0   #0   -1   Yes   Yes
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
```

```
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
================================================================================
========================
============================================================================
Registers
----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------------
val   1   1   2   3   4   5   6   7
Qi   #0   #5   #0   #0   #8   #0   #0   #0
----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
----------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------------
============================================================================
* CYCLE 5
================================================================================
========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   st   1   1   #0   #0   -1   Yes   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   Yes   I#-1   st   2   6   #0   #0   -1   No   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
================================================================================
========================
============================================================================
Registers
----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------------
val   1   1   2   3   2   5   6   7
Qi   #0   #10   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
```

```
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
========================================================================
* CYCLE 6
===========================================================================
===========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   Yes   I#-1   st   1   1   #0   #0   -1   Yes   Yes
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   Yes   I#-1   st   2   6   #0   #0   -1   No   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
===========================================================================
===========================
========================================================================
Registers
--------------------------------------------------------------------
          r0   r1   r2   r3   r4   r5   r6   r7
--------------------------------------------------------------------
val   1   1   2   3   2   5   6   7
Qi   #0   #10   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
          r8   r9   r10   r11   r12   r13   r14   r15
--------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
--------------------------------------------------------------------
========================================================================
* CYCLE 7
===========================================================================
===========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-----------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
```

```
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   Yes   I#-1   st   2   6   #0   #0   -1   Yes   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
================================================================================
=========================
================================================================================
Registers
------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
------------------------------------------------------------------
val   1   1   2   3   2   5   6   7
Qi   #0   #10   #0   #0   #0   #0   #0   #0
------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
------------------------------------------------------------------
================================================================================
* CYCLE 8
================================================================================
=========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   Yes   I#-1   st   2   6   #0   #0   -1   Yes   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
================================================================================
=========================
================================================================================
Registers
----------------------------------------------------------------
         r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------
val   1    1    2    3    2    5    6    7
Qi   #0   #10   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------
         r8   r9   r10   r11   r12   r13   r14   r15
----------------------------------------------------------------
val   8    9    10    11    12    13    14    15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------
================================================================================
* CYCLE 9
==========================================================================================================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   Yes   I#-1   st   2   6   #0   #0   -1   Yes   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
==========================================================================================================
================================================================================
Registers
----------------------------------------------------------------
         r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------
val   1    1    2    3    2    5    6    7
Qi   #0   #10   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------
         r8   r9   r10   r11   r12   r13   r14   r15
----------------------------------------------------------------
val   8    9    10    11    12    13    14    15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------
```

```
=======================================================================
* CYCLE 10
==========================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   Yes   I#-1   st   2   6   #0   #0   -1   Yes   Yes
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
==========================================================================
==========================
=======================================================================
Registers
-----------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
-----------------------------------------------------------------
val   1   1   2   3   2   5   6   7
Qi   #0   #10   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
-----------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
-----------------------------------------------------------------
=======================================================================
* CYCLE 11
==========================================================================
==========================
RS_id   type   Busy   inst#   Op   Vj   Vk   Qi   Qj   A   Exec   Done
-------------------------------------------------------------------------------------------
#1   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#2   LD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#3   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#4   ST   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#5   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#6   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
```

```
#7   ADD   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#8   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#9   MUL   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#10   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
#11   BNE   No   I#-1   NONE   -1   -1   #0   #0   -1   No   No
==============================================================================
=========================
=========================================================================
Registers
----------------------------------------------------------------------
        r0   r1   r2   r3   r4   r5   r6   r7
----------------------------------------------------------------------
val   1   -4   2   3   2   5   6   7
Qi   #0   #0   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------------
        r8   r9   r10   r11   r12   r13   r14   r15
----------------------------------------------------------------------
val   8   9   10   11   12   13   14   15
Qi   #0   #0   #0   #0   #0   #0   #0   #0
----------------------------------------------------------------------
=========================================================================
```