course: CA
exercise: 2
date: 2024-12-05
author: Miko Osak, 2628565

Each part of the exercise has been completed, including the optional task. It compiles successfully and works successfully. The loadxi function was tested by using a modified version of the provided ArraySum example, and the bitwise operation functions were tested using an updated version of ALUrun as well as the using the provided test program.

In the archive there is a folder called ‘Exercise2’. Inside of folder is the status report you are reading as well as ‘Circuits’ and ‘Programs’ folders. You will find the programs I used in ‘./Programs/Sigma16/Core/UserPrograms/’, these are given as assembly code and object code. The circuits I modified during this assessed exercise are found in ‘./Circuits/M1/’

When using ‘M1run.hs’ to test, run ‘boot UserPrograms/TestArray’ for the program with loadxi and ‘boot UserPrograms/TestLogic’ for the program that uses the bitwise operations.

Task 1

My approach to task 1 (creating loadxi) was first to combine the ‘load’ and ‘add’ functions to make loadxi because loadxi was defined as ‘R1 := mem[12ab+R2], R2 := R2+1’ which is a combination of load and add. The only issue was that there was no control signal specifying that the output of the ALU should go to ir_sa instead of ir_d, and this functionality was needed in order to be able to automatically increment the given index register. So I added a signal ctl_rf_da (da = destination a). When this signal is given, it sets ir_d = ir_sa in the datapath.

Below is M1run’s output for all 4 states of loadxi, here are the part where the instructions are executing:

```
--------------------------------------------------------------------------------
Cycle 53.  Running

System control
  reset = 0  cpu = 1  ctl_start = 0

Input/Output
  io_DMA = 0  io_memStore = 0  io_memFetch = 0  io_regFetch = 0
  io_address = 0000  io_data = 0000

Control state
   st_instr_fet = 00  st_dispatch = 00       st_add = 00       st_sub = 00
        st_inv = 00       st_and = 00       st_or  = 00       st_xor = 00
       st_mul0 = 00      st_div0 = 00       st_cmp = 00     st_trap0 = 00
       st_lea0 = 00      st_lea1 = 00     st_load0 = 00
      st_load1 = 00     st_load2 = 00    st_store0 = 00    st_store1 = 00
     st_store2 = 00     st_jump0 = 00     st_jump1 = 00     st_jump2 = 00
    st_jumpc00 = 00   st_jumpc01 = 00   st_jumpc02 = 00   st_jumpc10 = 00
    st_jumpc11 = 00   st_jumpc12 = 00      st_jal0 = 00      st_jal1 = 00
       st_jal2 = 00   st_loadxi0 = 11   st_loadxi1 = 00   st_loadxi2 = 00
    st_loadxi3 = 00
```

Control signals
ctl_alu_a = 0 ctl_alu_b = 0 ctl_alu_c = 1 ctl_alu_d = 1
ctl_x_pc = 1 ctl_y_ad = 0 ctl_rf_ld = 0 ctl_rf_ldcc = 0
ctl_rf_pc = 0 ctl_pc_ld = 0 ctl_pc_ad = 0 ctl_rf_alu = 0
ctl_rf_sd = 0 ctl_rf_da = 0 ctl_ir_ld = 0 ctl_pc_ld = 1
ctl_ad_ld = 1 ctl_ad_alu = 0 ctl_ma_pc = 1 ctl_sto = 0

ALU
ALU inputs: operation = 0011 x = 000a y = 0000 cc = 0018 ir_d = 5
ALU outputs: r = 000b ccnew = 0003 condcc = 0

Datapath
ir = f537 pc = 000a ad = 000f cc = 0018
a = 0000 b = 0000 x = 000a y = 0000
p = 0014 q = 000b r = 000b
ma = 000a md = 0000

Memory
m_sto = 0 m_addr = 000a m_real_addr = 000a m_data = 0000 m_out = 0014
M1>

Cycle 54. Running

System control
reset = 0 cpu = 1 ctl_start = 0

Input/Output
io_DMA = 0 io_memStore = 0 io_memFetch = 0 io_regFetch = 0
io_address = 0000 io_data = 0000

Control state
st_instr_fet = 00 st_dispatch = 00 st_add = 00 st_sub = 00
st_inv = 00 st_and = 00 st_or = 00 st_xor = 00
st_mul0 = 00 st_div0 = 00 st_cmp = 00 st_trap0 = 00
st_lea0 = 00 st_lea1 = 00 st_load0 = 00
st_load1 = 00 st_load2 = 00 st_store0 = 00 st_store1 = 00
st_store2 = 00 st_jump0 = 00 st_jump1 = 00 st_jump2 = 00
st_jumpc00 = 00 st_jumpc01 = 00 st_jumpc02 = 00 st_jumpc10 = 00
st_jumpc11 = 00 st_jumpc12 = 00 st_jal0 = 00 st_jal1 = 00
st_jal2 = 00 st_loadxi0 = 00 st_loadxi1 = 11 st_loadxi2 = 00
st_loadxi3 = 00

Control signals
ctl_alu_a = 0 ctl_alu_b = 0 ctl_alu_c = 0 ctl_alu_d = 0
ctl_x_pc = 0 ctl_y_ad = 1 ctl_rf_ld = 0 ctl_rf_ldcc = 0
ctl_rf_pc = 0 ctl_pc_ld = 0 ctl_pc_ad = 0 ctl_rf_alu = 0
ctl_rf_sd = 0 ctl_rf_da = 0 ctl_ir_ld = 0 ctl_pc_ld = 0
ctl_ad_ld = 1 ctl_ad_alu = 1 ctl_ma_pc = 0 ctl_sto = 0

ALU
ALU inputs: operation = 0000 x = 0000 y = 0014 cc = 0018 ir_d = 5

```
 ALU outputs:   r = 0014  ccnew = 0003  condcc = 0

Datapath
     ir = f537    pc = 000b    ad = 0014    cc = 0018
      a = 0000     b = 0000     x = 0000     y = 0014
      p = 0012     q = 0014     r = 0014
     ma = 0014    md = 0000

Memory
   m_sto = 0  m_addr = 0014  m_real_addr = 0014  m_data = 0000  m_out = 0012
*** Fetched displacement = 0014
M1>

--------------------------------------------------------------------------------
Cycle 55.  Running

System control
 reset = 0  cpu = 1  ctl_start = 0

Input/Output
 io_DMA = 0  io_memStore = 0  io_memFetch = 0  io_regFetch = 0
 io_address = 0000  io_data = 0000

Control state
  st_instr_fet = 00  st_dispatch = 00       st_add = 00       st_sub = 00
       st_inv = 00       st_and = 00       st_or  = 00       st_xor = 00
      st_mul0 = 00      st_div0 = 00       st_cmp = 00     st_trap0 = 00
      st_lea0 = 00      st_lea1 = 00     st_load0 = 00
     st_load1 = 00     st_load2 = 00    st_store0 = 00    st_store1 = 00
    st_store2 = 00     st_jump0 = 00     st_jump1 = 00     st_jump2 = 00
   st_jumpc00 = 00   st_jumpc01 = 00   st_jumpc02 = 00   st_jumpc10 = 00
   st_jumpc11 = 00   st_jumpc12 = 00      st_jal0 = 00      st_jal1 = 00
      st_jal2 = 00   st_loadxi0 = 00   st_loadxi1 = 00   st_loadxi2 = 11
   st_loadxi3 = 00

Control signals
     ctl_alu_a = 0    ctl_alu_b = 0    ctl_alu_c = 0    ctl_alu_d = 0
      ctl_x_pc = 0     ctl_y_ad = 0    ctl_rf_ld = 1  ctl_rf_ldcc = 0
     ctl_rf_pc = 0    ctl_pc_ld = 0    ctl_pc_ad = 0   ctl_rf_alu = 0
     ctl_rf_sd = 0    ctl_rf_da = 0    ctl_ir_ld = 0    ctl_pc_ld = 0
     ctl_ad_ld = 0   ctl_ad_alu = 0    ctl_ma_pc = 0      ctl_sto = 0

ALU
 ALU inputs:   operation = 0000  x = 0000  y = 0000  cc = 0018  ir_d = 5
 ALU outputs:   r = 0000  ccnew = 0004  condcc = 0

Datapath
     ir = f537    pc = 000b    ad = 0014    cc = 0018
      a = 0000     b = 0000     x = 0000     y = 0000
      p = 0012     q = 0000     r = 0000
     ma = 0014    md = 0000
```

```
Memory
   m_sto = 0  m_addr = 0014  m_real_addr = 0014  m_data = 0000  m_out = 0012
Register file update: R5 := 0012
M1>

--------------------------------------------------------------------------------
Cycle 56.  Running

System control
  reset = 0  cpu = 1  ctl_start = 1

Input/Output
  io_DMA = 0  io_memStore = 0  io_memFetch = 0  io_regFetch = 0
  io_address = 0000  io_data = 0000

Control state
   st_instr_fet = 00  st_dispatch = 00       st_add = 00       st_sub = 00
        st_inv = 00       st_and = 00       st_or  = 00       st_xor = 00
       st_mul0 = 00      st_div0 = 00       st_cmp = 00     st_trap0 = 00
       st_lea0 = 00      st_lea1 = 00     st_load0 = 00
      st_load1 = 00     st_load2 = 00    st_store0 = 00    st_store1 = 00
     st_store2 = 00     st_jump0 = 00     st_jump1 = 00     st_jump2 = 00
    st_jumpc00 = 00   st_jumpc01 = 00   st_jumpc02 = 00   st_jumpc10 = 00
    st_jumpc11 = 00   st_jumpc12 = 00      st_jal0 = 00      st_jal1 = 00
       st_jal2 = 00   st_loadxi0 = 00   st_loadxi1 = 00   st_loadxi2 = 00
    st_loadxi3 = 11

Control signals
     ctl_alu_a = 0    ctl_alu_b = 0    ctl_alu_c = 1    ctl_alu_d = 1
      ctl_x_pc = 0     ctl_y_ad = 0    ctl_rf_ld = 1  ctl_rf_ldcc = 1
     ctl_rf_pc = 0    ctl_pc_ld = 0    ctl_pc_ad = 0   ctl_rf_alu = 1
     ctl_rf_sd = 0    ctl_rf_da = 1    ctl_ir_ld = 0    ctl_pc_ld = 0
     ctl_ad_ld = 0   ctl_ad_alu = 0    ctl_ma_pc = 0      ctl_sto = 0

ALU
  ALU inputs:   operation = 0011  x = 0000  y = 0000  cc = 0018  ir_d = 5
  ALU outputs:   r = 0001  ccnew = 0003  condcc = 0

Datapath
     ir = f537    pc = 000b    ad = 0014    cc = 0018
      a = 0000     b = 0000     x = 0000     y = 0000
      p = 0001     q = 0001     r = 0001
     ma = 0014    md = 0000

Memory
   m_sto = 0  m_addr = 0014  m_real_addr = 0014  m_data = 0000  m_out = 0012
Register file update: R3 := 0001

********************************************************************************
Executed instruction:  loadxi  R5,0014[R3]
R5 := 0012 was loaded in cycle 55
R3 := 0001 was loaded in cycle 56
```

```
Processor state:    pc = 000b  ir = f537  ad = 0014
********************************************************************************
M1>
```

Task 2

For task 2, I first focused on adding functionality to the ALU. The ability to inverse a word of bits was already implemented in Hydra in 'Hydra-3.5.7/src/HDL/Hydra/Circuits/Combinational.hs' as invw. Using that as a guide, I created bitwise operations for words using map2 (and2w, or2w and xor2w).

The ALU has a 3-bit opcode and already supports 5 operations. A 3-bit opcode can only support up to 8 operations. So to add 4 (total 9), I had to add an opcode, resulting in a 4-bit opcode. With this I can have each logic operation have an opcode which has a leftmost bit of 1. And so, INV = 1000, AND = 1001, OR = 1010 and XOR = 1011. Based off the two rightmost values I can see what the exact operation is. Then I can pick whether to return the arithmetic or logic result based off the leftmost bit. I then tested this using ALUrun.

The rest of the task consisted of adding the ctl_alu_d control signal and adding the inv, and, or & xor instructions (similar to task 1). After updating M1run and running the provided program, here are the part where the instructions are executing:

```
--------------------------------------------------------------------------------
Cycle 21.  Running

System control
  reset = 0  cpu = 1  ctl_start = 1

Input/Output
  io_DMA = 0  io_memStore = 0  io_memFetch = 0  io_regFetch = 0
  io_address = 0000  io_data = 0000

Control state
   st_instr_fet = 00  st_dispatch = 00       st_add = 00       st_sub = 00
         st_inv = 11       st_and = 00       st_or  = 00       st_xor = 00
        st_mul0 = 00      st_div0 = 00       st_cmp = 00     st_trap0 = 00
        st_lea0 = 00      st_lea1 = 00     st_load0 = 00
       st_load1 = 00     st_load2 = 00    st_store0 = 00    st_store1 = 00
      st_store2 = 00     st_jump0 = 00     st_jump1 = 00     st_jump2 = 00
     st_jumpc00 = 00   st_jumpc01 = 00   st_jumpc02 = 00   st_jumpc10 = 00
     st_jumpc11 = 00   st_jumpc12 = 00      st_jal0 = 00      st_jal1 = 00
        st_jal2 = 00   st_loadxi0 = 00   st_loadxi1 = 00   st_loadxi2 = 00
     st_loadxi3 = 00

Control signals
      ctl_alu_a = 1    ctl_alu_b = 0    ctl_alu_c = 0    ctl_alu_d = 0
       ctl_x_pc = 0     ctl_y_ad = 0    ctl_rf_ld = 1  ctl_rf_ldcc = 0
      ctl_rf_pc = 0    ctl_pc_ld = 0    ctl_pc_ad = 0   ctl_rf_alu = 1
      ctl_rf_sd = 0    ctl_rf_da = 0    ctl_ir_ld = 0    ctl_pc_ld = 0
      ctl_ad_ld = 0   ctl_ad_alu = 0     ctl_ma_pc = 0      ctl_sto = 0

ALU
```

```
 ALU inputs:   operation = 1000  x = 0003  y = 0005  cc = 0000  ir_d = 3
 ALU outputs:   r = fffc  ccnew = 0018  condcc = 0

Datapath
     ir = 5312    pc = 0005    ad = 6412    cc = 0000
      a = 0003     b = 0005     x = 0003     y = 0005
      p = fffc     q = fffc     r = fffc
     ma = 6412    md = 0003

Memory
   m_sto = 0  m_addr = 6412  m_real_addr = 6412  m_data = 0003  m_out = 0000
Register file update: R3 := fffc

*************************************************************************
Executed instruction:  inv  R3,0005[R1]
R3 := fffc was loaded in cycle 21
Processor state:    pc = 0005  ir = 5312  ad = 6412
*************************************************************************
M1>

------------------------------------------------------------------------
Cycle 24.  Running

System control
 reset = 0  cpu = 1  ctl_start = 1

Input/Output
 io_DMA = 0  io_memStore = 0  io_memFetch = 0  io_regFetch = 0
 io_address = 0000  io_data = 0000

Control state
   st_instr_fet = 00  st_dispatch = 00       st_add = 00       st_sub = 00
        st_inv = 00       st_and = 11       st_or  = 00       st_xor = 00
       st_mul0 = 00      st_div0 = 00       st_cmp = 00     st_trap0 = 00
       st_lea0 = 00      st_lea1 = 00     st_load0 = 00
      st_load1 = 00     st_load2 = 00    st_store0 = 00    st_store1 = 00
     st_store2 = 00     st_jump0 = 00     st_jump1 = 00     st_jump2 = 00
    st_jumpc00 = 00   st_jumpc01 = 00   st_jumpc02 = 00   st_jumpc10 = 00
    st_jumpc11 = 00   st_jumpc12 = 00      st_jal0 = 00      st_jal1 = 00
       st_jal2 = 00   st_loadxi0 = 00   st_loadxi1 = 00   st_loadxi2 = 00
    st_loadxi3 = 00

Control signals
     ctl_alu_a = 1    ctl_alu_b = 0    ctl_alu_c = 0    ctl_alu_d = 1
      ctl_x_pc = 0     ctl_y_ad = 0    ctl_rf_ld = 1  ctl_rf_ldcc = 0
     ctl_rf_pc = 0    ctl_pc_ld = 0    ctl_pc_ad = 0   ctl_rf_alu = 1
     ctl_rf_sd = 0    ctl_rf_da = 0    ctl_ir_ld = 0    ctl_pc_ld = 0
     ctl_ad_ld = 0   ctl_ad_alu = 0    ctl_ma_pc = 0      ctl_sto = 0

ALU
 ALU inputs:   operation = 1001  x = 0003  y = 0005  cc = 0000  ir_d = 4
 ALU outputs:   r = 0001  ccnew = 0018  condcc = 0
```

```
Datapath
      ir = 6412    pc = 0006    ad = 6412    cc = 0000
       a = 0003     b = 0005     x = 0003     y = 0005
       p = 0001     q = 0001     r = 0001
      ma = 6412    md = 0003

Memory
    m_sto = 0  m_addr = 6412  m_real_addr = 6412  m_data = 0003  m_out = 0000
Register file update: R4 := 0001

*************************************************************************
Executed instruction:  and  R4,0005[R1]
R4 := 0001 was loaded in cycle 24
Processor state:    pc = 0006  ir = 6412  ad = 6412
*************************************************************************
M1>

-------------------------------------------------------------------------
Cycle 27.  Running

System control
  reset = 0  cpu = 1  ctl_start = 1

Input/Output
  io_DMA = 0  io_memStore = 0  io_memFetch = 0  io_regFetch = 0
  io_address = 0000  io_data = 0000

Control state
   st_instr_fet = 00  st_dispatch = 00       st_add = 00       st_sub = 00
        st_inv = 00       st_and = 00       st_or  = 11       st_xor = 00
       st_mul0 = 00      st_div0 = 00       st_cmp = 00     st_trap0 = 00
       st_lea0 = 00      st_lea1 = 00     st_load0 = 00
      st_load1 = 00     st_load2 = 00    st_store0 = 00    st_store1 = 00
     st_store2 = 00     st_jump0 = 00     st_jump1 = 00     st_jump2 = 00
    st_jumpc00 = 00   st_jumpc01 = 00   st_jumpc02 = 00   st_jumpc10 = 00
    st_jumpc11 = 00   st_jumpc12 = 00      st_jal0 = 00      st_jal1 = 00
       st_jal2 = 00   st_loadxi0 = 00   st_loadxi1 = 00   st_loadxi2 = 00
    st_loadxi3 = 00

Control signals
      ctl_alu_a = 1    ctl_alu_b = 0    ctl_alu_c = 1    ctl_alu_d = 0
       ctl_x_pc = 0     ctl_y_ad = 0    ctl_rf_ld = 1  ctl_rf_ldcc = 0
      ctl_rf_pc = 0    ctl_pc_ld = 0    ctl_pc_ad = 0   ctl_rf_alu = 1
      ctl_rf_sd = 0    ctl_rf_da = 0    ctl_ir_ld = 0    ctl_pc_ld = 0
      ctl_ad_ld = 0   ctl_ad_alu = 0    ctl_ma_pc = 0      ctl_sto = 0

ALU
  ALU inputs:   operation = 1010  x = 0003  y = 0005  cc = 0000  ir_d = 5
  ALU outputs:   r = 0007  ccnew = 0018  condcc = 0

Datapath
```

```
      ir = 7512    pc = 0007    ad = 6412    cc = 0000
       a = 0003     b = 0005     x = 0003     y = 0005
       p = 0007     q = 0007     r = 0007
      ma = 6412    md = 0003

Memory
   m_sto = 0  m_addr = 6412  m_real_addr = 6412  m_data = 0003  m_out = 0000
Register file update: R5 := 0007

*************************************************************************
Executed instruction:  or  R5,0005[R1]
R5 := 0007 was loaded in cycle 27
Processor state:    pc = 0007  ir = 7512  ad = 6412
*************************************************************************
M1>

-------------------------------------------------------------------------
Cycle 30.  Running

System control
  reset = 0  cpu = 1  ctl_start = 1

Input/Output
  io_DMA = 0  io_memStore = 0  io_memFetch = 0  io_regFetch = 0
  io_address = 0000  io_data = 0000

Control state
   st_instr_fet = 00  st_dispatch = 00       st_add = 00       st_sub = 00
        st_inv = 00       st_and = 00       st_or  = 00       st_xor = 11
       st_mul0 = 00      st_div0 = 00       st_cmp = 00     st_trap0 = 00
       st_lea0 = 00      st_lea1 = 00     st_load0 = 00
      st_load1 = 00     st_load2 = 00    st_store0 = 00    st_store1 = 00
     st_store2 = 00     st_jump0 = 00     st_jump1 = 00     st_jump2 = 00
    st_jumpc00 = 00   st_jumpc01 = 00   st_jumpc02 = 00   st_jumpc10 = 00
    st_jumpc11 = 00   st_jumpc12 = 00      st_jal0 = 00      st_jal1 = 00
       st_jal2 = 00   st_loadxi0 = 00   st_loadxi1 = 00   st_loadxi2 = 00
    st_loadxi3 = 00

Control signals
      ctl_alu_a = 1    ctl_alu_b = 0    ctl_alu_c = 1    ctl_alu_d = 1
       ctl_x_pc = 0     ctl_y_ad = 0    ctl_rf_ld = 1  ctl_rf_ldcc = 0
      ctl_rf_pc = 0    ctl_pc_ld = 0    ctl_pc_ad = 0   ctl_rf_alu = 1
      ctl_rf_sd = 0    ctl_rf_da = 0    ctl_ir_ld = 0    ctl_pc_ld = 0
      ctl_ad_ld = 0   ctl_ad_alu = 0    ctl_ma_pc = 0      ctl_sto = 0

ALU
  ALU inputs:   operation = 1011  x = 0003  y = 0005  cc = 0000  ir_d = 6
  ALU outputs:   r = 0006  ccnew = 0018  condcc = 0

Datapath
      ir = 8612    pc = 0008    ad = 6412    cc = 0000
       a = 0003     b = 0005     x = 0003     y = 0005
```

```
      p = 0006     q = 0006     r = 0006
     ma = 6412    md = 0003

Memory
   m_sto = 0  m_addr = 6412  m_real_addr = 6412  m_data = 0003  m_out = 0000
Register file update: R6 := 0006

*********************************************************************************
Executed instruction:  xor  R6,0005[R1]
R6 := 0006 was loaded in cycle 30
Processor state:    pc = 0008  ir = 8612  ad = 6412
*********************************************************************************
M1>
```