EE533 Lab6

Quad HW-Threaded ARM ISA Compatible Processor Core in NetFPGA

Group19

Ya-Yi Lin

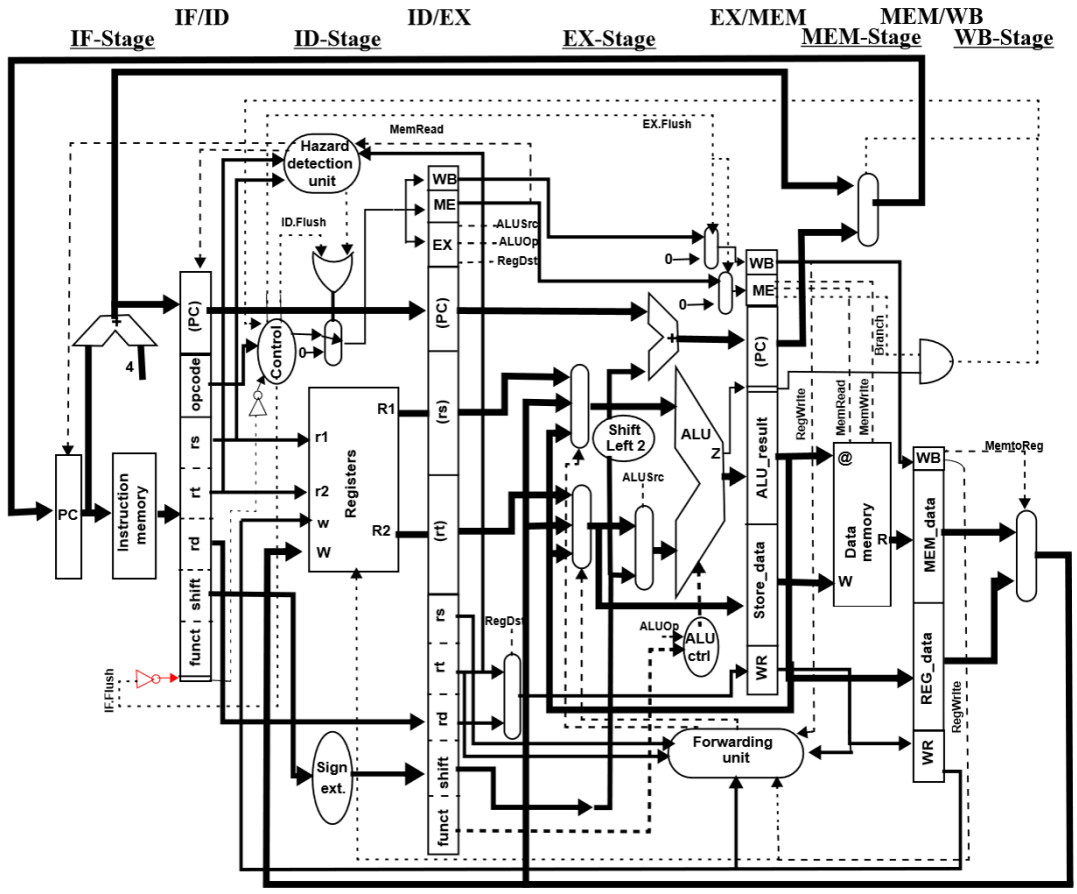
Pin-Chieh Yen

Yung-Chun Yang

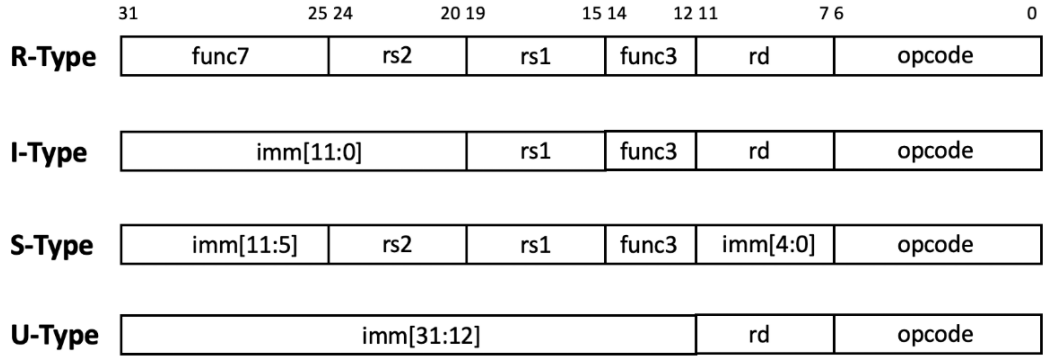
Github Link: https://github.com/yungchun1007/EE533_Lab6

- **High-level Design of Datapath**
- **Instruction Set Architecture**
- **Bubble Sort in C and Assembly**
- **Machine Code**
- **NetFPGA**

High-level Design of Datapath



Instruction Set Architecture



	31 25	24 20	19 15	14 12	11 7	6 0	
lui	imm[31:12]				rd	0110111	U-Type
auipc	imm[31:12]				rd	0010111	U-Type
addi	imm[11:0]		rs1	000	rd	0010011	I-Type
slti	imm[11:0]		rs1	010	rd	0010011	I-Type
sltiu	imm[11:0]		rs1	011	rd	0010011	I-Type
xori	imm[11:0]		rs1	100	rd	0010011	I-Type
ori	imm[11:0]		rs1	110	rd	0010011	I-Type
andi	imm[11:0]		rs1	111	rd	0010011	I-Type
slli	0000000	shamt	rs1	001	rd	0010011	I-Type
srli	0000000	shamt	rs1	101	rd	0010011	I-Type
srai	0100000	shamt	rs1	101	rd	0010011	I-Type
add	0000000	rs2	rs1	000	rd	0110011	R-Type
sub	0100000	rs2	rs1	000	rd	0110011	R-Type
sll	0000000	rs2	rs1	001	rd	0110011	R-Type
slt	0000000	rs2	rs1	010	rd	0110011	R-Type
sltu	0000000	rs2	rs1	011	rd	0110011	R-Type
xor	0000000	rs2	rs1	100	rd	0110011	R-Type
srl	0000000	rs2	rs1	101	rd	0110011	R-Type
sra	0100000	rs2	rs1	101	rd	0110011	R-Type
or	0000000	rs2	rs1	110	rd	0110011	R-Type
and	0000000	rs2	rs1	111	rd	0110011	R-Type

	31 25	24 20	19 15	14 12	11 7	6 0	
jal	imm[20], imm[10:1], imm[11], imm[19:12]				rd	1101111	J-Type
jalr	imm[11:0]		rs1	000	rd	1100111	I-Type
beq	imm[12,10:5]	rs2	rs1	000	imm[4:1,11]	1100011	B-Type
bne	imm[12,10:5]	rs2	rs1	001	imm[4:1,11]	1100011	B-Type
blt	imm[12,10:5]	rs2	rs1	100	imm[4:1,11]	1100011	B-Type
bge	imm[12,10:5]	rs2	rs1	101	imm[4:1,11]	1100011	B-Type
bltu	imm[12,10:5]	rs2	rs1	110	imm[4:1,11]	1100011	B-Type
bgeu	imm[12,10:5]	rs2	rs1	111	imm[4:1,11]	1100011	B-Type

	31 25	24 20	19 15	14 12	11 7	6 0	
lb	imm[11:0]		rs1	000	rd	0000011	I-Type
lh	imm[11:0]		rs1	001	rd	0000011	I-Type
lw	imm[11:0]		rs1	010	rd	0000011	I-Type
lbu	imm[11:0]		rs1	100	rd	0000011	I-Type
lhu	imm[11:0]		rs1	101	rd	0000011	I-Type
sb	imm[11:5]	rs2	rs1	000	imm[4:0]	0100011	S-Type
sh	imm[11:5]	rs2	rs1	001	imm[4:0]	0100011	S-Type
sw	imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	S-Type

Bubble Sort in C

C: > EE533 > Lab6 > C sort.c > main()

```c
int main() {
    int array[10] = {323, 123, -455, 2, 98, 125, 10, 65, -56, 0};
    int i, j, swap;
    for (i = 0 ; i < 10; i++) {
        for (j = i+1 ; j < 10 ; j++) {
            if (array[j] < array[i]) {
                swap = array[j];
                array[j] = array[i];
                array[i] = swap;
            }
        }
    }
}
```

Bubble Sort in Assembly

C: > EE533 > Lab6 > sort_rv32.s

```
    .file   "sort.c"
    .option nopic
    .attribute arch, "rv32i2p1_m2p0_zmmul1p0"
    .attribute unaligned_access, 0
    .attribute stack_align, 16
    .text
    .section    .rodata
    .align  2
.LC0:
    .word   323
    .word   123
    .word   -455
    .word   2
    .word   98
    .word   125
    .word   10
    .word   65
    .word   -56
    .word   0
    .text
    .align  2
    .globl  main
    .type   main, @function
main:
    addi    sp,sp,-80
    sw  ra,76(sp)
    sw  s0,72(sp)
    addi    s0,sp,80
    lui a5,%hi(.LC0)
    addi    a5,a5,%lo(.LC0)
    lw  t3,0(a5)
    lw  t1,4(a5)
    lw  a7,8(a5)
    lw  a6,12(a5)
    lw  a0,16(a5)
    lw  a1,20(a5)
    lw  a2,24(a5)
    lw  a3,28(a5)
    lw  a4,32(a5)
    sw  t3,-68(s0)
    sw  t1,-64(s0)
    sw  a7,-60(s0)
    sw  a6,-56(s0)
    sw  a0,-52(s0)
    sw  a1,-48(s0)
```

C: > EE533 > Lab6 > ASM sort_rv32.s

```
24  main:
46      sw  a2,-44(s0)
47      sw  a3,-40(s0)
48      sw  a4,-36(s0)
49      lw  a5,36(a5)
50      sw  a5,-32(s0)
51      sw  zero,-20(s0)
52      j   .L2
53  .L6:
54      lw  a5,-20(s0)
55      addi    a5,a5,1
56      sw  a5,-24(s0)
57      j   .L3
58  .L5:
59      lw  a4,-24(s0)
60      addi    a5,s0,-68
61      slli    a4,a4,2
62      add a5,a4,a5
63      lw  a4,0(a5)
64      lw  a3,-20(s0)
65      addi    a5,s0,-68
66      slli    a3,a3,2
67      add a5,a3,a5
68      lw  a5,0(a5)
69      bge a4,a5,.L4
70      lw  a4,-24(s0)
71      addi    a5,s0,-68
72      slli    a4,a4,2
73      add a5,a4,a5
74      lw  a5,0(a5)
75      sw  a5,-28(s0)
76      lw  a4,-20(s0)
77      addi    a5,s0,-68
78      slli    a4,a4,2
79      add a5,a4,a5
80      lw  a4,0(a5)
81      lw  a3,-24(s0)
82      addi    a5,s0,-68
83      slli    a3,a3,2
84      add a5,a3,a5
85      sw  a4,0(a5)
86      lw  a4,-20(s0)
87      addi    a5,s0,-68
88      slli    a4,a4,2
89      add a5,a4,a5
```

```
90      lw  a4,-28(s0)
91      sw  a4,0(a5)
92  .L4:
93      lw  a5,-24(s0)
94      addi    a5,a5,1
95      sw  a5,-24(s0)
96  .L3:
97      lw  a4,-24(s0)
98      li  a5,9
99      ble a4,a5,.L5
100     lw  a5,-20(s0)
101     addi    a5,a5,1
102     sw  a5,-20(s0)
103 .L2:
104     lw  a4,-20(s0)
105     li  a5,9
106     ble a4,a5,.L6
107     li  a5,0
108     mv  a0,a5
109     lw  ra,76(sp)
110     lw  s0,72(sp)
111     addi    sp,sp,80
112     jr  ra
113     .size   main, .-main
114     .ident  "GCC: (xPack GNU RISC-V Embedded GCC x86_64) 15.2.0"
115     .section    .note.GNU-stack,"",@progbits
116
```

Bubble Sort in Machine Code

1. **Compile C code to RV64I binary (ELF format)**
   riscv64-unknown-elf-gcc -march=rv64i -mabi=lp64 sort.c -o sort.elf

2. **Then use objdump to view the disassembled code**
   riscv64-unknown-elf-objdump -d sort.elf > sort.dump

C: > EE533 > Lab6 > machine_code.txt

```
sort.o:     file format elf32-littleriscv


Disassembly of section .text:

00000000 <main>:
   0: fb010113            addi  sp,sp,-80
   4: 04112623            sw  ra,76(sp)
   8: 04812423            sw  s0,72(sp)
   c: 05010413            addi  s0,sp,80
  10: 000007b7            lui a5,0x0
  14: 00078793            mv  a5,a5
  18: 0007ae03            lw  t3,0(a5) # 0 <main>
  1c: 0047a303            lw  t1,4(a5)
  20: 0087a883            lw  a7,8(a5)
  24: 00c7a803            lw  a6,12(a5)
  28: 0107a503            lw  a0,16(a5)
  2c: 0147a583            lw  a1,20(a5)
  30: 0187a603            lw  a2,24(a5)
  34: 01c7a683            lw  a3,28(a5)
  38: 0207a703            lw  a4,32(a5)
  3c: fbc42e23            sw  t3,-68(s0)
  40: fc642023            sw  t1,-64(s0)
  44: fd142223            sw  a7,-60(s0)
  48: fd042423            sw  a6,-56(s0)
  4c: fca42623            sw  a0,-52(s0)
  50: fcb42823            sw  a1,-48(s0)
  54: fcc42a23            sw  a2,-44(s0)
  58: fcd42c23            sw  a3,-40(s0)
  5c: fce42e23            sw  a4,-36(s0)
  60: 0247a783            lw  a5,36(a5)
  64: fef42023            sw  a5,-32(s0)
  68: fe042623            sw  zero,-20(s0)
  6c: 0bc0006f            j 128 <.L2>

00000070 <.L6>:
  70: fec42783            lw  a5,-20(s0)
  74: 00178793            addi  a5,a5,1
  78: fef42423            sw  a5,-24(s0)
  7c: 0940006f            j 110 <.L3>

00000080 <.L5>:
  80: fe842703            lw  a4,-24(s0)
  84: fbc40793            addi  a5,s0,-68
```

```
 88: 00271713            slli  a4,a4,0x2
 8c: 00f707b3            add a5,a4,a5
 90: 0007a703            lw  a4,0(a5)
 94: fec42683            lw  a3,-20(s0)
 98: fbc40793            addi  a5,s0,-68
 9c: 00269693            slli  a3,a3,0x2
 a0: 00f687b3            add a5,a3,a5
 a4: 0007a783            lw  a5,0(a5)
 a8: 04f75e63            bge a4,a5,104 <.L4>
 ac: fe842703            lw  a4,-24(s0)
 b0: fbc40793            addi  a5,s0,-68
 b4: 00271713            slli  a4,a4,0x2
 b8: 00f707b3            add a5,a4,a5
 bc: 0007a783            lw  a5,0(a5)
 c0: fef42223            sw  a5,-28(s0)
 c4: fec42703            lw  a4,-20(s0)
 c8: fbc40793            addi  a5,s0,-68
 cc: 00271713            slli  a4,a4,0x2
 d0: 00f707b3            add a5,a4,a5
 d4: 0007a703            lw  a4,0(a5)
 d8: fe842683            lw  a3,-24(s0)
 dc: fbc40793            addi  a5,s0,-68
 e0: 00269693            slli  a3,a3,0x2
 e4: 00f687b3            add a5,a3,a5
 e8: 00e7a023            sw  a4,0(a5)
 ec: fec42703            lw  a4,-20(s0)
 f0: fbc40793            addi  a5,s0,-68
 f4: 00271713            slli  a4,a4,0x2
 f8: 00f707b3            add a5,a4,a5
 fc: fe442703            lw  a4,-28(s0)
100: 00e7a023            sw  a4,0(a5)

00000104 <.L4>:
104: fe842783            lw  a5,-24(s0)
108: 00178793            addi  a5,a5,1
10c: fef42423            sw  a5,-24(s0)

00000110 <.L3>:
110: fe842703            lw  a4,-24(s0)
114: 00900793            li  a5,9
118: f6e7d4e3            bge a5,a4,80 <.L5>
11c: fec42783            lw  a5,-20(s0)
120: 00178793            addi  a5,a5,1
124: fef42623            sw  a5,-20(s0)
```

```
90
91    00000128 <.L2>:
92      128: fec42703           lw  a4,-20(s0)
93      12c: 00900793           li  a5,9
94      130: f4e7d0e3           bge a5,a4,70 <.L6>
95      134: 00000793           li  a5,0
96      138: 00078513           mv  a0,a5
97      13c: 04c12083           lw  ra,76(sp)
98      140: 04812403           lw  s0,72(sp)
99      144: 05010113           addi  sp,sp,80
100     148: 00008067           ret
101
```

Use python script to extract the machine code generated

C: > EE533 > Lab6 > code > script.py

```python
import re

def generate_verilog(mem_file, asm_file):
    with open(asm_file, 'r') as asm_f, open(mem_file, 'w') as mem_f:
        lines = asm_f.readlines()

        index = 0

        for line in lines:
            line = line.strip()
            if not line or line.startswith('file format'):
                continue

            match = re.match(r'^([0-9a-fA-F]+):\s+([0-9a-fA-F]{8})\s+(.+)$', line)
            if match:
                instruction = match.group(2)
                comment = match.group(3) if match.group(3) else ''
                mem_f.write(f"inst_mem[{index}] = 32'h{instruction};        // {comment}\n")
                index += 1
            else:
                print(f"Invalid instruction format || {line}")


asm_file = 'sort_machine.s'
mem_file = 'instruction.text'


generate_verilog(mem_file, asm_file)

```

The transcript of a sequence of commands typed to the interface

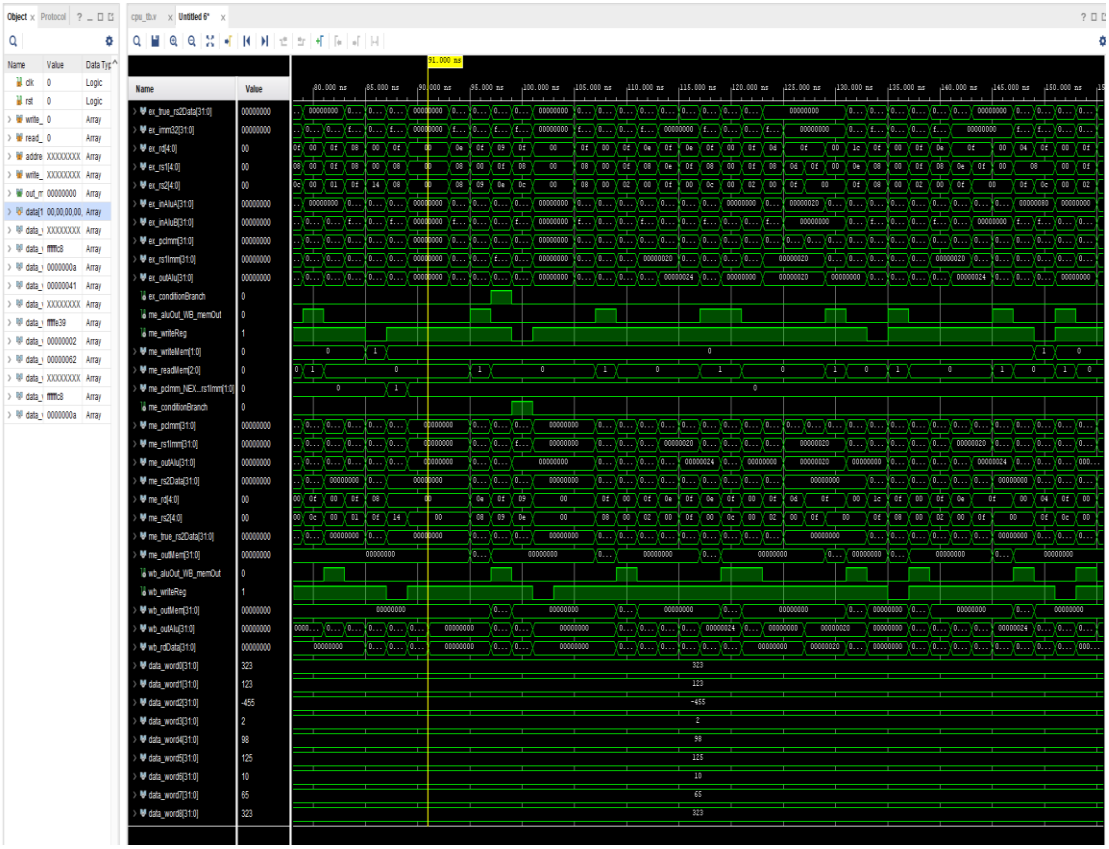
- SW/HW interface in pipeline module

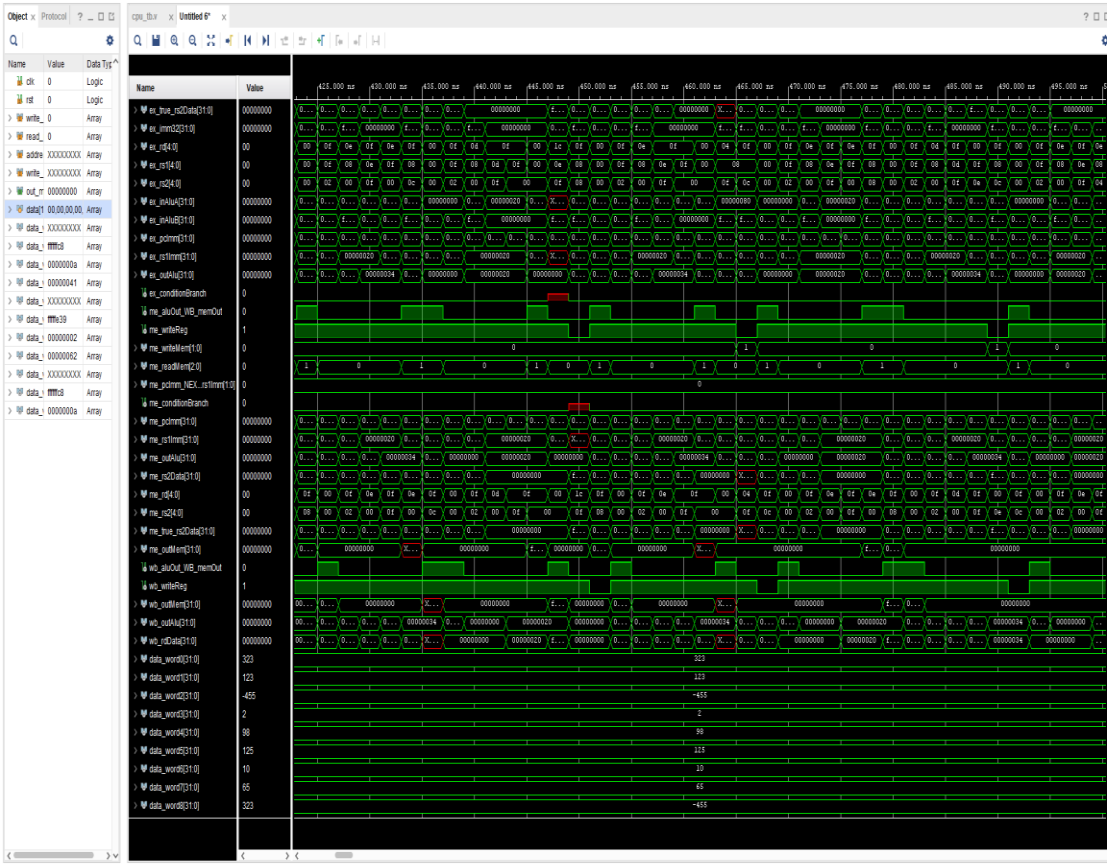
```
// pipeline_netfpga.v

// Software registers
wire [7:0]                                  mem_raddr_ver;

// Hardware registers
wire [63:0]                                 mem_rdata_ver;
wire [31:0]                                 mem_rdata_ver_high;
wire [31:0]                                 mem_rdata_ver_low;

// Verification(D-MEM)
assign mem_rdata_ver = dmem_rdata;
assign mem_rdata_ver_low = mem_rdata_ver[63:32];
assign mem_rdata_ver_high = mem_rdata_ver[31:0];

// ......

generic_regs
#(
   .UDP_REG_SRC_WIDTH   (UDP_REG_SRC_WIDTH),
   .TAG                 (`PIPELINE_BLOCK_ADDR),          // Tag -- eg. MODULE_TAG
   .REG_ADDR_WIDTH      (`PIPELINE_REG_ADDR_WIDTH),     // Width of block addresses -- eg. MODULE_REG_ADDR_WIDTH
   .NUM_COUNTERS        (0),                 // Number of counters
   .NUM_SOFTWARE_REGS   (1),                 // Number of sw regs
   .NUM_HARDWARE_REGS   (2)                  // Number of hw regs
) module_regs (
   .reg_req_in       (reg_req_in),
   .reg_ack_in       (reg_ack_in),
   .reg_rd_wr_L_in   (reg_rd_wr_L_in),
   .reg_addr_in      (reg_addr_in),
   .reg_data_in      (reg_data_in),
   .reg_src_in       (reg_src_in),

   .reg_req_out      (reg_req_out),
   .reg_ack_out      (reg_ack_out),
   .reg_rd_wr_L_out  (reg_rd_wr_L_out),
   .reg_addr_out     (reg_addr_out),
   .reg_data_out     (reg_data_out),
   .reg_src_out      (reg_src_out),

   // --- counters interface
   .counter_updates  (),
   .counter_decrement(),

   // --- SW regs interface
   .software_regs    (mem_raddr_ver),

   // --- HW regs interface
   .hardware_regs    ({mem_rdata_ver_high, mem_rdata_ver_low}),

   .clk              (clk),
   .reset            (reset)
);
```

- Registers in pipeline.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<nf:module xmlns:nf="http://www.NetFPGA.org/NF2_register_system" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
    <nf:name>pipeline</nf:name>
    <nf:prefix>pipeline</nf:prefix>
    <nf:location>udp</nf:location>
    <nf:description>Registers for PIPELINE</nf:description>
    <nf:blocksize>64</nf:blocksize>
    <nf:registers>
        <nf:register>
          <nf:name>mem_raddr_ver</nf:name>
            <nf:description>D-MEM Read Address</nf:description>
            <nf:type>generic_software32</nf:type>
        </nf:register>
        <nf:register>
          <nf:name>mem_rdata_ver_high</nf:name>
            <nf:description>Upper 32 bits of D-MEM Read Data</nf:description>
            <nf:type>generic_hardware32</nf:type>
        </nf:register>
            <nf:register>
          <nf:name>mem_rdata_ver_low</nf:name>
            <nf:description>Lower 32 bits of D-MEM Read Data</nf:description>
            <nf:type>generic_hardware32</nf:type>
        </nf:register>
    </nf:registers>
</nf:module>
```

The internal memory dump that shows the array data before and after sort program execution

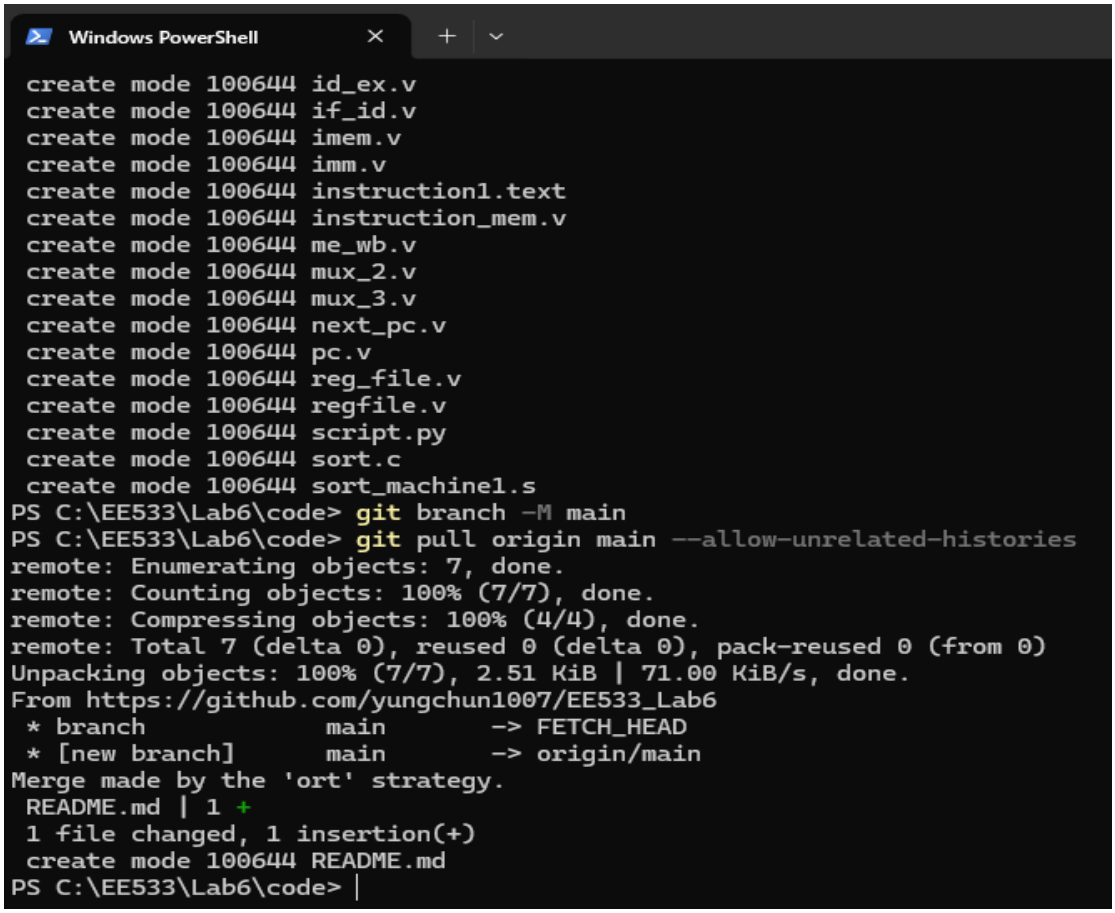


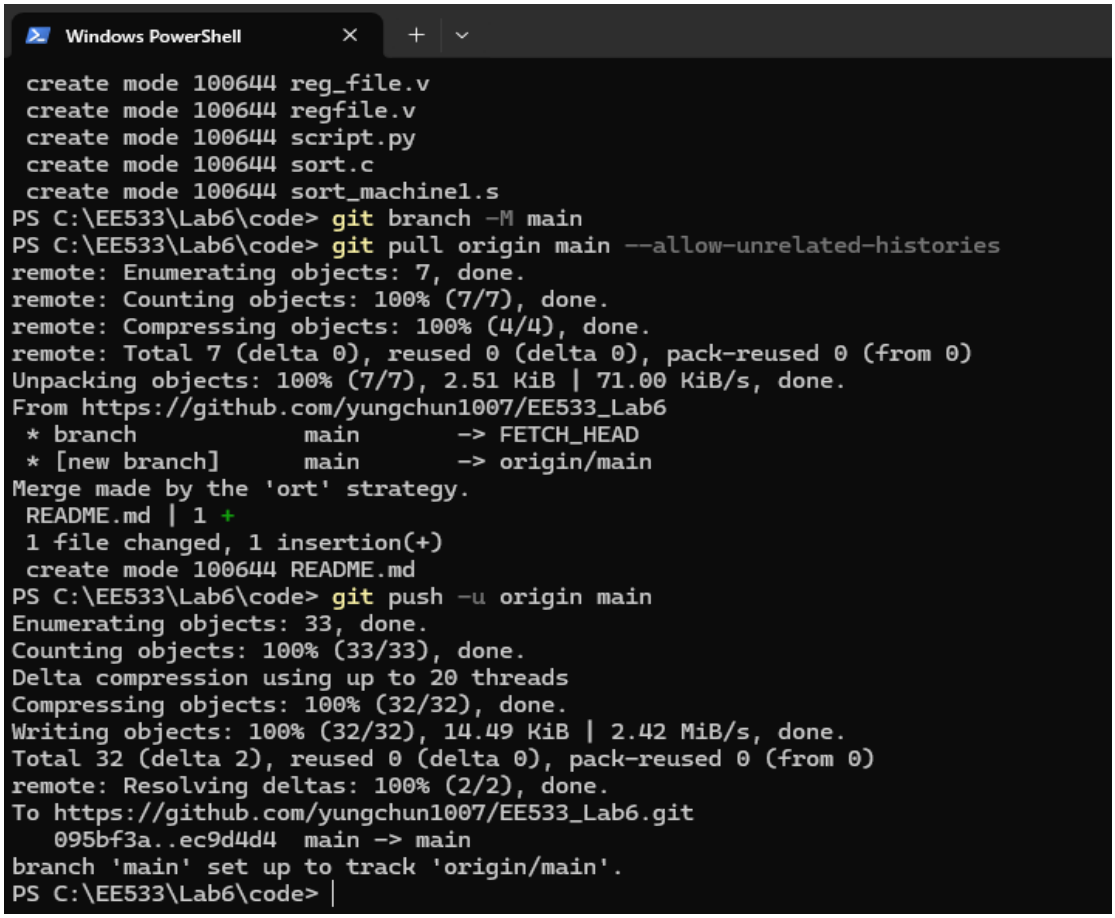


The last line is swapped to -455:

Commit log

```
PS C:\EE533\Lab6\code> git commit -m "Commit code"
[master (root-commit) a3c8c83] Commit code
 28 files changed, 1953 insertions(+)
 create mode 100644 add_4.v
 create mode 100644 add_pc.v
 create mode 100644 alu.v
 create mode 100644 bubble_sort_asm.txt
 create mode 100644 controller.v
 create mode 100644 cpu.v
 create mode 100644 data_mem.v
 create mode 100644 datapath.v
 create mode 100644 ex_me.v
 create mode 100644 forward_unit.v
 create mode 100644 hazard_detection_unit.v
 create mode 100644 id.v
 create mode 100644 id_ex.v
 create mode 100644 if_id.v
 create mode 100644 imem.v
 create mode 100644 imm.v
 create mode 100644 instruction1.text
 create mode 100644 instruction_mem.v
 create mode 100644 me_wb.v
 create mode 100644 mux_2.v
 create mode 100644 mux_3.v
 create mode 100644 next_pc.v
 create mode 100644 pc.v
 create mode 100644 reg_file.v
 create mode 100644 regfile.v
 create mode 100644 script.py
 create mode 100644 sort.c
 create mode 100644 sort_machine1.s
```


Windows PowerShell
create mode 100644 id_ex.v
create mode 100644 if_id.v
create mode 100644 imem.v
create mode 100644 imm.v
create mode 100644 instruction1.text
create mode 100644 instruction_mem.v
create mode 100644 me_wb.v
create mode 100644 mux_2.v
create mode 100644 mux_3.v
create mode 100644 next_pc.v
create mode 100644 pc.v
create mode 100644 reg_file.v
create mode 100644 regfile.v
create mode 100644 script.py
create mode 100644 sort.c
create mode 100644 sort_machine1.s
PS C:\EE533\Lab6\code> git branch -M main
PS C:\EE533\Lab6\code> git pull origin main --allow-unrelated-histories
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (7/7), 2.51 KiB | 71.00 KiB/s, done.
From https://github.com/yungchun1007/EE533_Lab6
* branch main -> FETCH_HEAD
* [new branch] main -> origin/main
Merge made by the 'ort' strategy.
README.md | 1 +
1 file changed, 1 insertion(+)
create mode 100644 README.md
PS C:\EE533\Lab6\code>


Windows PowerShell
create mode 100644 reg_file.v
create mode 100644 regfile.v
create mode 100644 script.py
create mode 100644 sort.c
create mode 100644 sort_machine1.s
PS C:\EE533\Lab6\code> git branch -M main
PS C:\EE533\Lab6\code> git pull origin main --allow-unrelated-histories
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (7/7), 2.51 KiB | 71.00 KiB/s, done.
From https://github.com/yungchun1007/EE533_Lab6
* branch main -> FETCH_HEAD
* [new branch] main -> origin/main
Merge made by the 'ort' strategy.
README.md | 1 +
1 file changed, 1 insertion(+)
create mode 100644 README.md
PS C:\EE533\Lab6\code> git push -u origin main
Enumerating objects: 33, done.
Counting objects: 100% (33/33), done.
Delta compression using up to 20 threads
Compressing objects: 100% (32/32), done.
Writing objects: 100% (32/32), 14.49 KiB | 2.42 MiB/s, done.
Total 32 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/yungchun1007/EE533_Lab6.git
095bf3a..ec9d4d4 main -> main
branch 'main' set up to track 'origin/main'.
PS C:\EE533\Lab6\code>