[Microprocessor Applications – Term Project]
FFT Accelerator with RISC-V

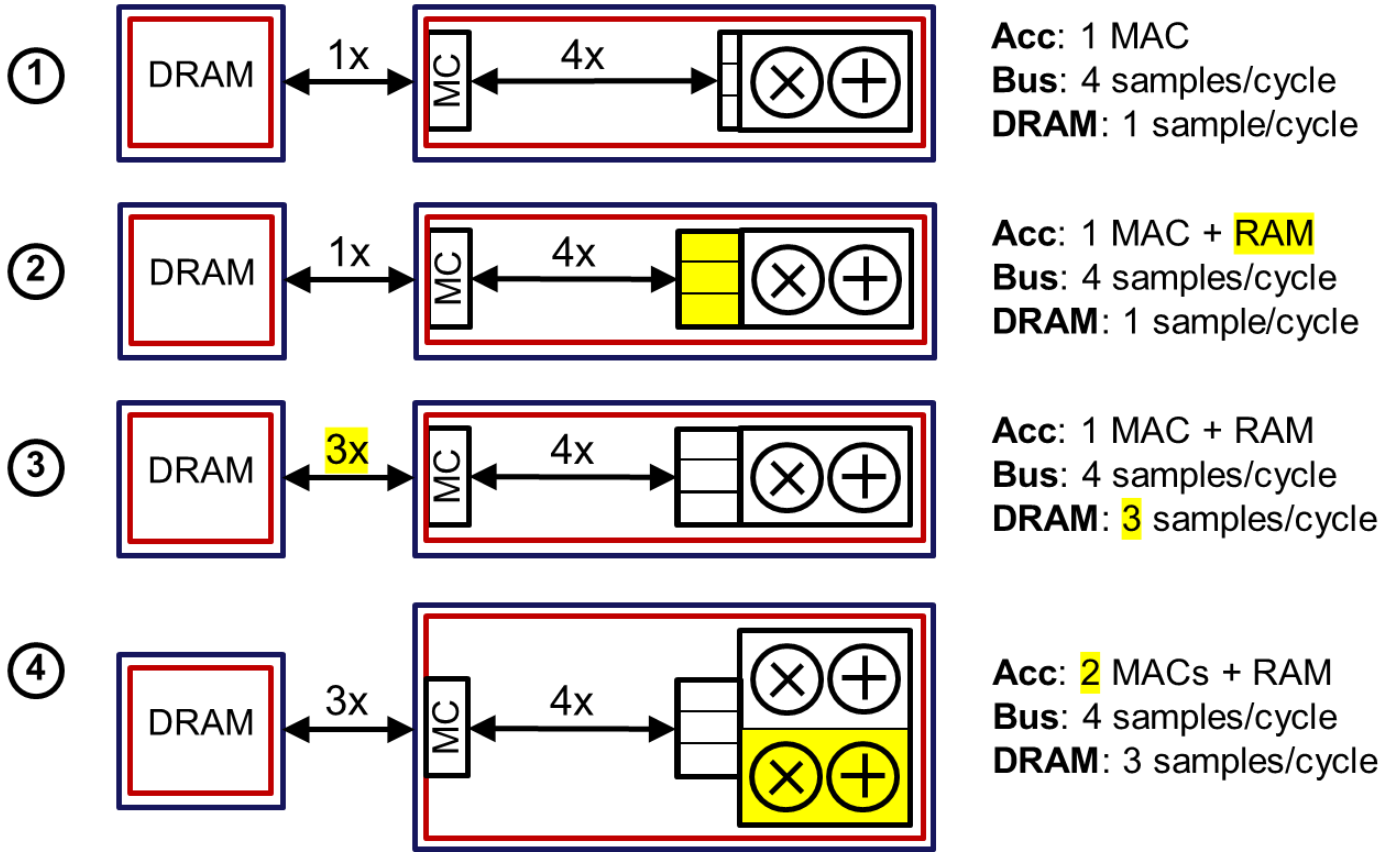
Chester Sungchung Park (박성정)
SoC Design Lab, Konkuk University
Webpage: http://soclab.konkuk.ac.kr

Outline

- Introduction
  - Accelerator dataflow
  - Transformer accelerator
- TP1: FFT accelerator w/ a single core
  - Target system
  - Reference project
  - Design constraints
  - Evaluation
  - Submission
- TP2: FFT accelerator w/ multiple cores
  - Target system
  - Reference project
  - Design constraints
  - Evaluation
  - Submission

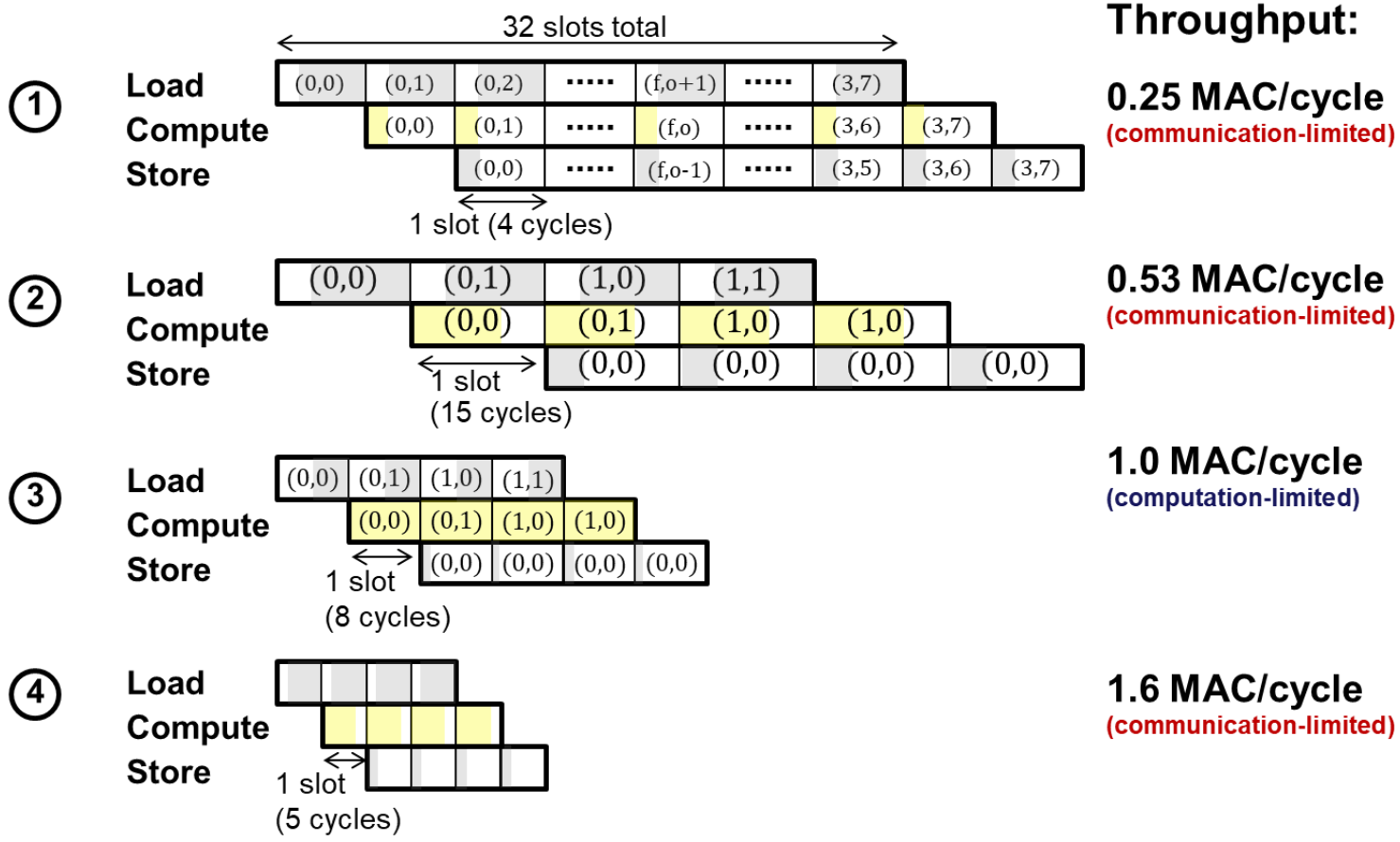
Accelerator Dataflow

- Toy example



Accelerator Dataflow

- 3-stage pipeline (load-compute-store)



Throughput:

0.25 MAC/cycle
(communication-limited)

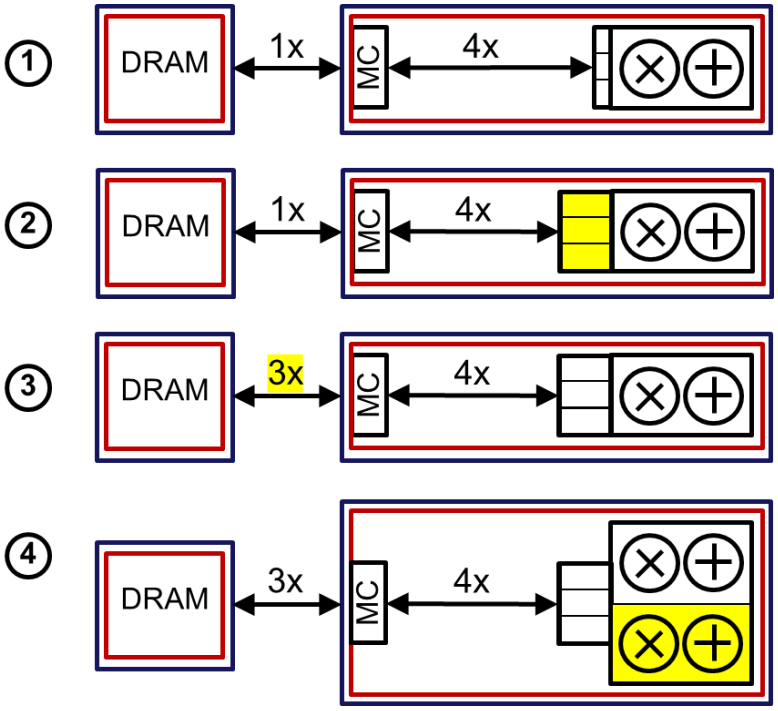
0.53 MAC/cycle
(communication-limited)

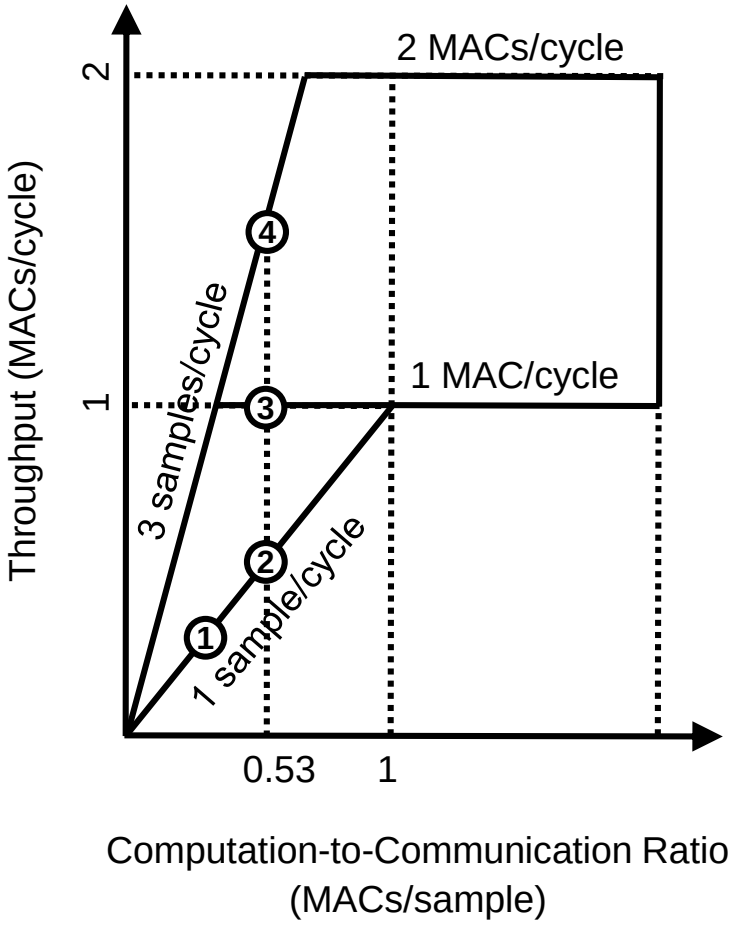
1.0 MAC/cycle
(computation-limited)

1.6 MAC/cycle
(communication-limited)

Accelerator Dataflow

- Roofline model

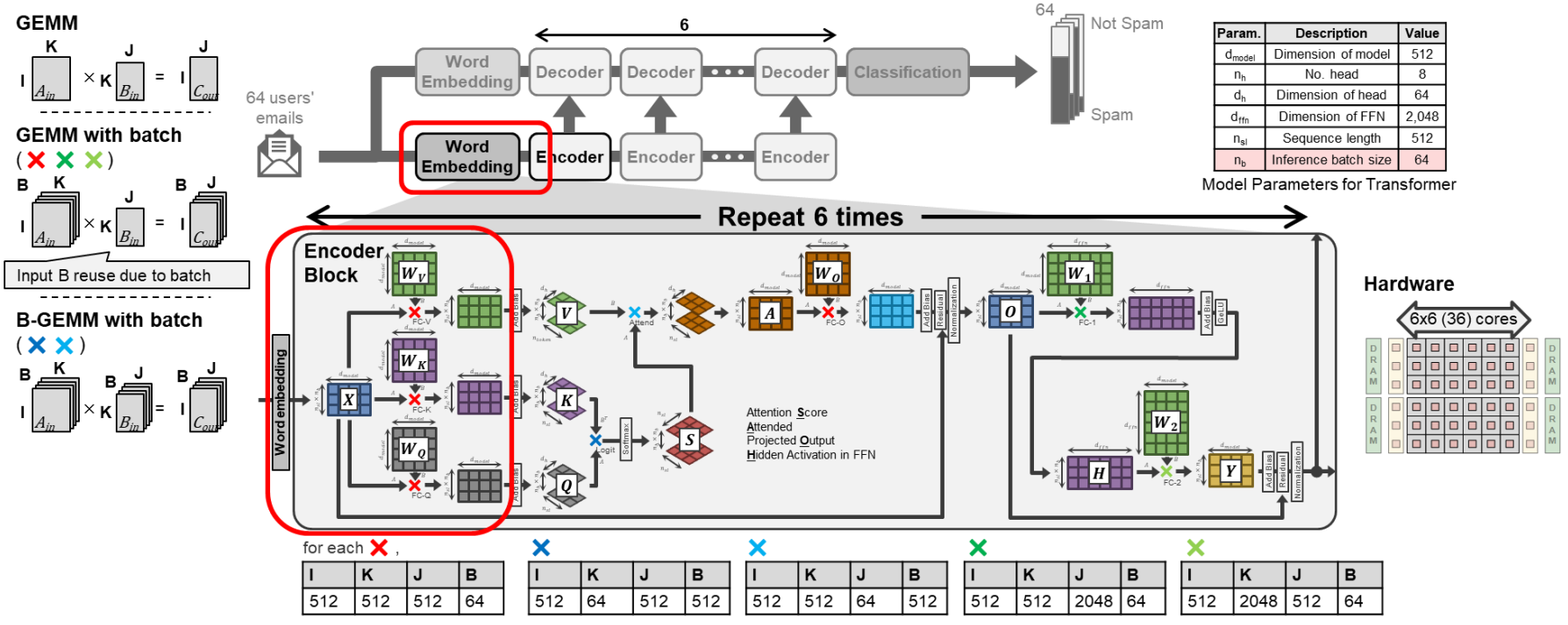




For more details, check the following presentation:
C. S. Park, *Accelerator dataflow simulations and interconnect optimizations*, HSN 2024

Transformer Accelerator

- GEMM-centric workload



Param.	Description	Value
d_{model}	Dimension of model	512
n_h	No. head	8
d_h	Dimension of head	64
d_{ffn}	Dimension of FFN	2,048
n_{sl}	Sequence length	512
n_b	Inference batch size	64

Model Parameters for Transformer

I	K	J	B
512	512	512	64

I	K	J	B
512	64	512	512

I	K	J	B
512	512	64	512

I	K	J	B
512	512	2048	64

I	K	J	B
512	2048	512	64

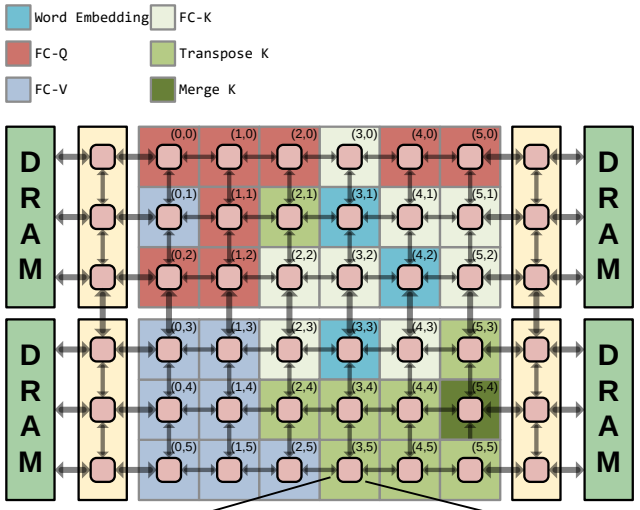
Transformer Accelerator

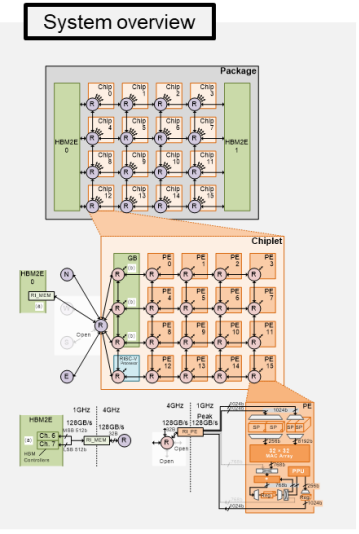
❑ NetTLMSim

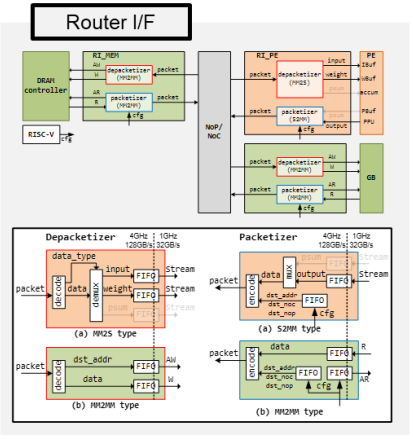
Assumed design parameters

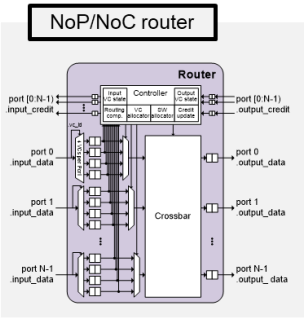
Space-Time Mapping	Target Hardware	16 chiplets per package [1], 16 PEs per chiplet [1, 2], 1024 MACs per PE [1, 3] Total 512 TOPS at 1 GHz freq.	Memory Architecture	DRAM Type	HBM2E, 2 devices Total 1024 GB/s
	PE Style	NVDLA-like vector MAC array		SRAM Type	Dual port [3]
	Loop Order	OS-LWS [4]		Global Buf. Size, Bandwidth	640 KB activation storage [4], 128 GB/s per NoC router
Network Architecture	Topology	Mesh [2]		Local Buf. Size, Bandwidth	36 KB x2 (double buffered) [4], 128 GB/s at 1 GHz freq.
	Routing	DOR YX [2]		Precision	8 bits (24 bits for partial sums) [4]
	Flow Ctrl	Cut-through [2]			
	Packet Len	1 head flit, 16 body flits [2]			

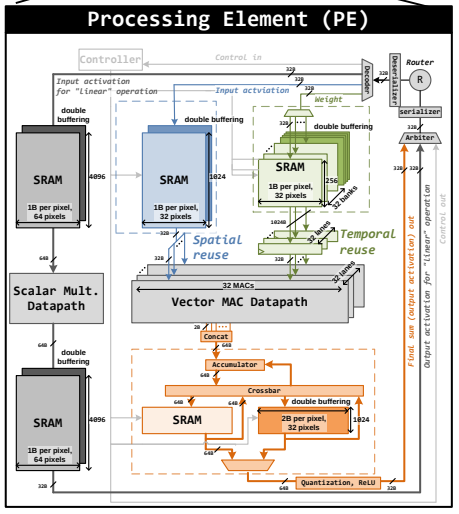
[1] Cai at el. 2024
[2] Zimmer at el. 2020
[3] Keller at el. 2023
[4] Venkatesan at el. 2019







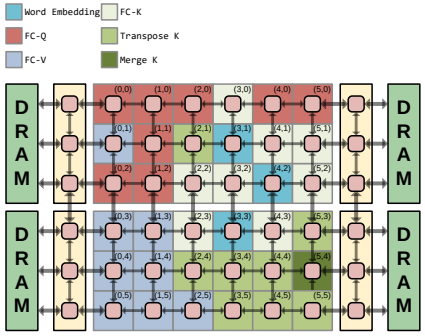


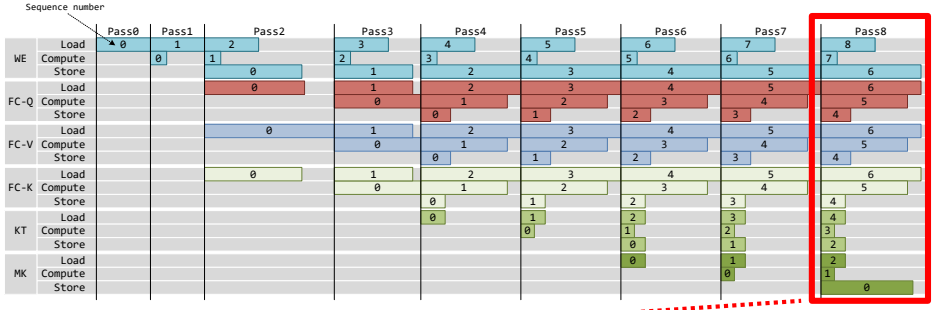


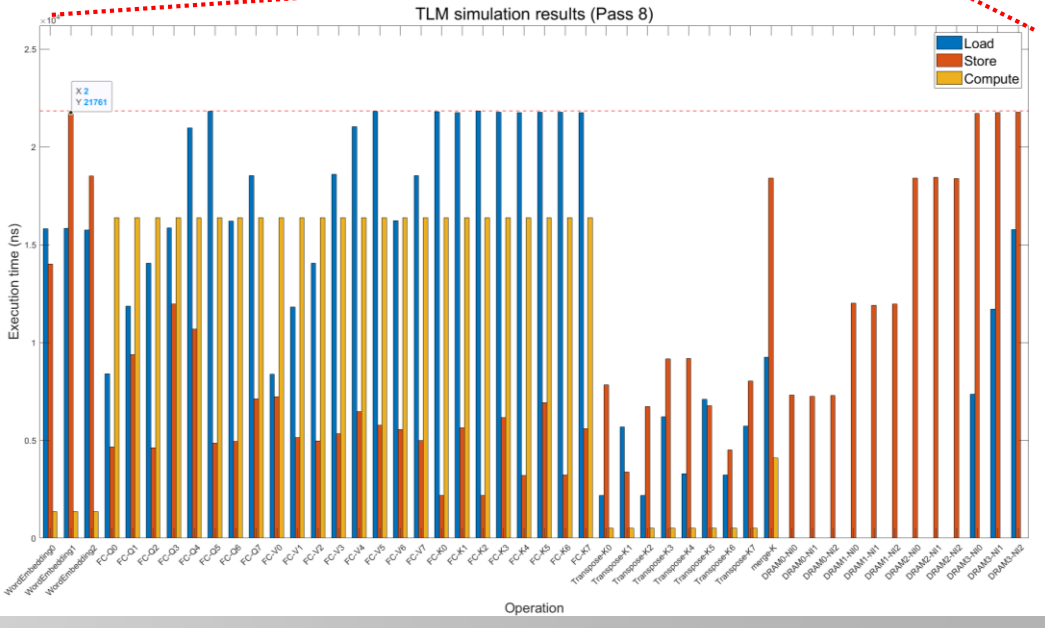
For more details, check the following presentation:
F. S. Park and C. S. Park, *Pre-RTL simulation based design space exploration for multi-chiplet dataflows*, HiPChips (MICRO 2024).

Transformer Accelerator

- NetTLMSim (cont'd)





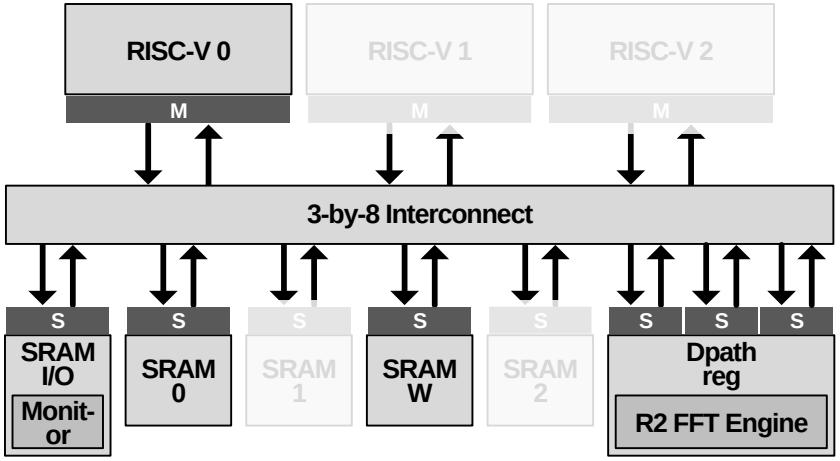


TP1: FFT Accelerator with a Single Core

Target System

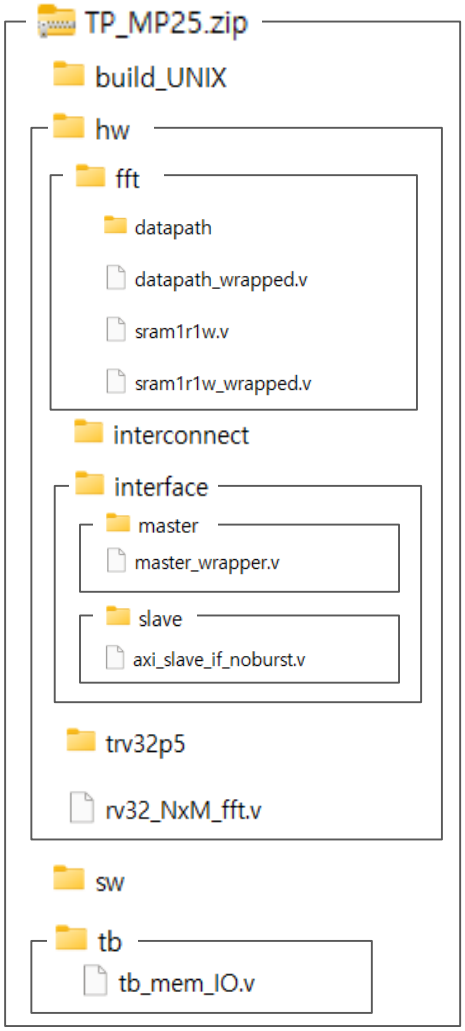
❑ Overview

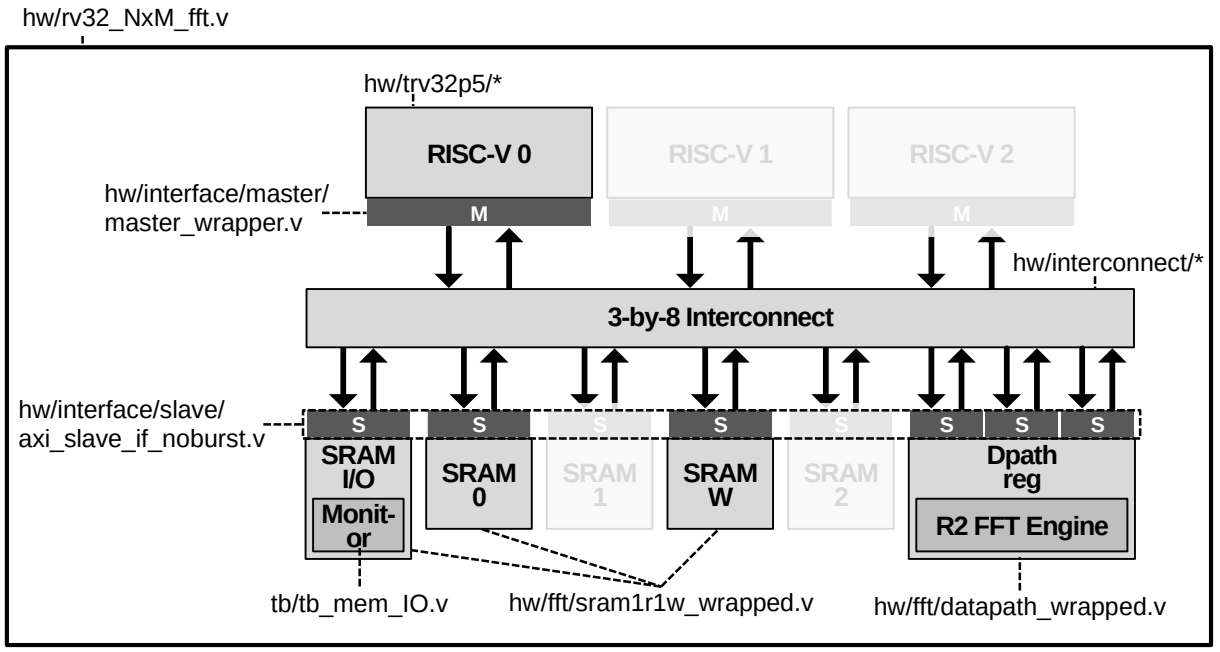
- 1 RISC-V core (RISC-V 0)
  - ✓ ***Not*** used for twiddle-factor multiplications & bufferflies
- 3-by-8 AXI interconnect
- SRAM
  - ✓ 1R1W
  - ✓ 3 banks (SRAM 0, I/O, W)
  - ✓ 16,384 lines/bank, 4 bytes/line
- R2 FFT engine
  - ✓ 5 32-bit I/O registers
  - ✓ Twiddle-factor multiplications & butterflies (done only here!)
- Monitor
  - ✓ Counts No. output sets (3)



Target System

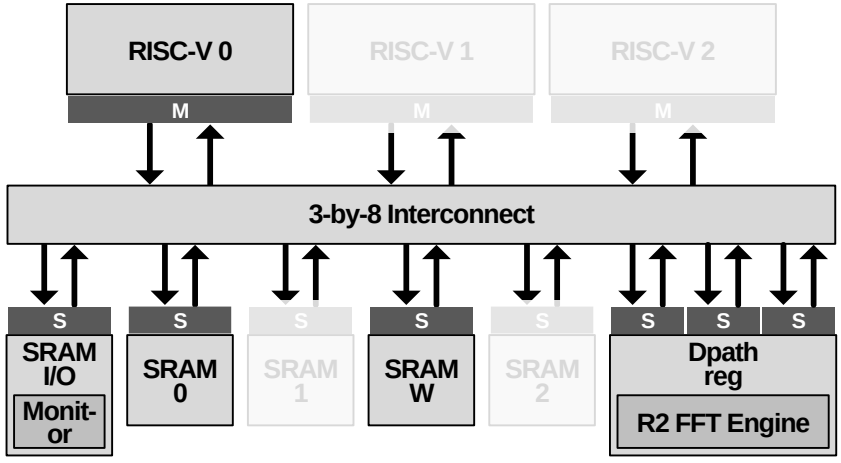
❑ File Tree





Target System

❑ Address map



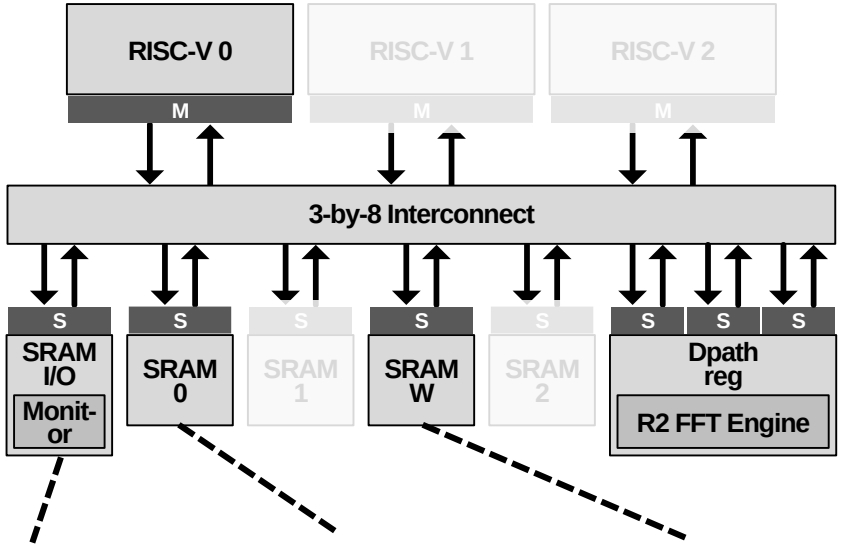
Name	31 … Description … 0	Address	Note
Input a	real number | imaginary number	0x0500_0000	write only
Output x	real number | imaginary number	0x0500_0004	read only
Input b	real number | imaginary number	0x0600_0000	write only
Output y	real number | imaginary number	0x0600_0004	read only
Input w	real number | imaginary number	0x0700_0000	write only

Address	Name
0xFFFF_FFFF – 0x0700_0004	*RESERVED*
0x0700_0003 – 0x0700_0000	Dpath reg (port 2)
0x06FF_FFFF – 0x0600_0008	*RESERVED*
0x0600_0007 – 0x0600_0000	Dpath reg (port 1)
0x05FF_FFFF – 0x0500_0008	*RESERVED*
0x0500_0007 – 0x0500_0000	Dpath reg (port 0)
0x04FF_FFFF – 0x0401_0000	*RESERVED*
0x0400_FFFF – 0x0400_0000	SRAM 2
0x03FF_FFFF – 0x0301_0000	*RESERVED*
0x0300_FFFF – 0x0300_0000	SRAM W
0x02FF_FFFF – 0x0201_0000	*RESERVED*
0x0200_FFFF – 0x0200_0000	SRAM 1
0x01FF_FFFF – 0x0101_0000	*RESERVED*
0x0100_FFFF – 0x0100_0000	SRAM 0
0x00FF_FFFF – 0x0001_0000	*RESERVED*
0x0000_FFFF – 0x0000_0000	SRAM I/O

Target System

❑ Data layout

- Initial condition



Address	Values in SRAM I/O
0x0000_FFFF	*Zeros*
0x0000_0B00	
0x0000_0AFF	Golden Output
0x0000_0800	
0x0000_07FF	*Zeros*
0x0000_0700	
0x0000_06FF	Output
0x0000_0400	
0x0000_03FF	*Zeros*
0x0000_0300	
0x0000_02FF	Input
0x0000_0000	

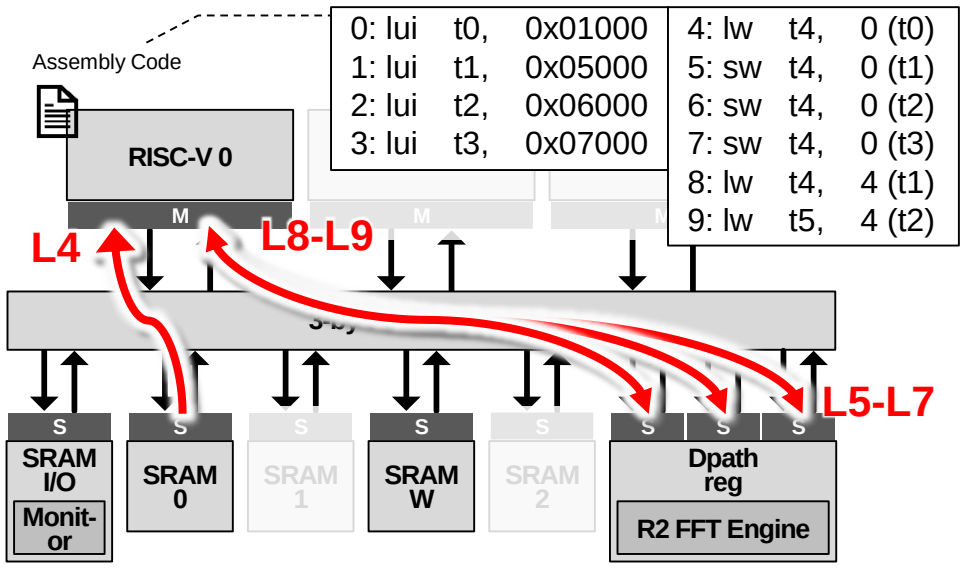
Address	Values in SRAM 0
0x01FF_FFFF	*Zeros*
0x0100_0080	

Address	Values in SRAM W
0x03FF_FFFF	*Zeros*
0x0300_0080	
0x0300_007F	Twiddle Factor
0x0300_0000	

Address	Name
0xFFFF_FFFF	*RESERVED*
0x0700_0004	
0x0700_0003	Dpath reg (port 2)
0x0700_0000	
0x06FF_FFFF	*RESERVED*
0x0600_0008	
0x0600_0007	Dpath reg (port 1)
0x0600_0000	
0x05FF_FFFF	*RESERVED*
0x0500_0008	
0x0500_0007	Dpath reg (port 0)
0x0500_0000	
0x04FF_FFFF	*RESERVED*
0x0401_0000	
0x0400_FFFF	SRAM 2
0x0400_0000	
0x03FF_FFFF	*RESERVED*
0x0301_0000	
0x0300_FFFF	SRAM W
0x0300_0000	
0x02FF_FFFF	*RESERVED*
0x0201_0000	
0x0200_FFFF	SRAM 1
0x0200_0000	
0x01FF_FFFF	*RESERVED*
0x0101_0000	
0x0100_FFFF	SRAM 0
0x0100_0000	
0x00FF_FFFF	*RESERVED*
0x0001_0000	
0x0000_FFFF	SRAM I/O
0x0000_0000	

Target System

❑ Dataflow (example)

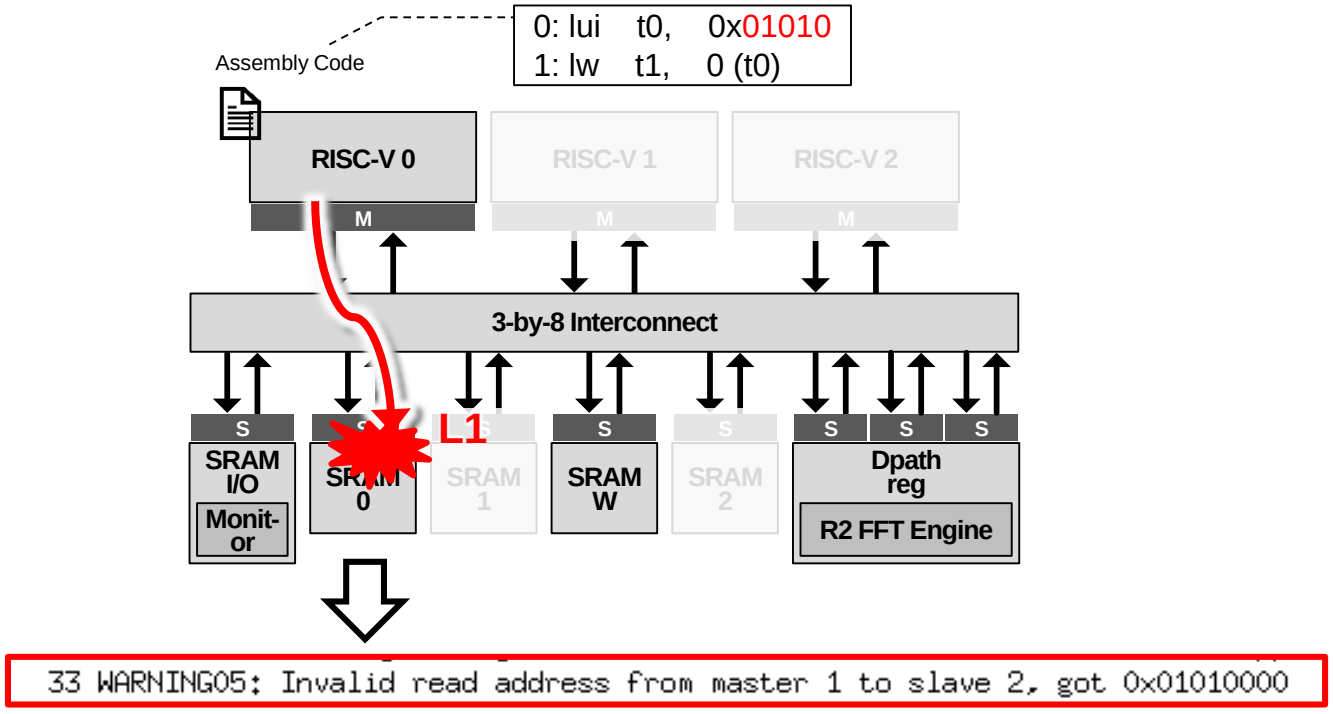


Name	31 Description 0	Address	Note
Input a	real number / imaginary number	0x0500_0000	write only
Output x	real number / imaginary number	0x0500_0004	read only
Input b	real number / imaginary number	0x0600_0000	write only
Output y	real number / imaginary number	0x0600_0004	read only
Input w	real number / imaginary number	0x0700_0000	write only

Address	Name
0xFFFF_FFFF – 0x0700_0004	*RESERVED*
0x0700_0003 – 0x0700_0000	Dpath reg (port 2)
0x06FF_FFFF – 0x0600_0008	*RESERVED*
0x0600_0007 – 0x0600_0000	Dpath reg (port 1)
0x05FF_FFFF – 0x0500_0008	*RESERVED*
0x0500_0007 – 0x0500_0000	Dpath reg (port 0)
0x04FF_FFFF – 0x0401_0000	*RESERVED*
0x0400_FFFF – 0x0400_0000	SRAM 2
0x03FF_FFFF – 0x0301_0000	*RESERVED*
0x0300_FFFF – 0x0300_0000	SRAM W
0x02FF_FFFF – 0x0201_0000	*RESERVED*
0x0200_FFFF – 0x0200_0000	SRAM 1
0x01FF_FFFF – 0x0101_0000	*RESERVED*
0x0100_FFFF – 0x0100_0000	SRAM 0
0x00FF_FFFF – 0x0001_0000	*RESERVED*
0x0000_FFFF – 0x0000_0000	SRAM I/O

Target System

- Dataflow (example) (cont'd)
  - Warning messages for the *RESERVED* region (not mapped to SRAM/Dpath)



Address	Name
0xFFFF_FFFF	*RESERVED*
0x0700_0004	
0x0700_0003	Dpath reg (port 2)
0x0700_0000	
0x06FF_FFFF	*RESERVED*
0x0600_0008	
0x0600_0007	Dpath reg (port 1)
0x0600_0000	
0x05FF_FFFF	*RESERVED*
0x0500_0008	
0x0500_0007	Dpath reg (port 0)
0x0500_0000	
0x04FF_FFFF	*RESERVED*
0x0401_0000	
0x0400_FFFF	SRAM 2
0x0400_0000	
0x03FF_FFFF	*RESERVED*
0x0301_0000	
0x0300_FFFF	SRAM W
0x0300_0000	
0x02FF_FFFF	*RESERVED*
0x0201_0000	
0x0200_FFFF	SRAM 1
0x0200_0000	
0x01FF_FFFF	*RESERVED*
0x0101_0000	
0x0100_FFFF	SRAM 0
0x0100_0000	
0x00FF_FFFF	*RESERVED*
0x0001_0000	
0x0000_FFFF	SRAM I/O
0x0000_0000	

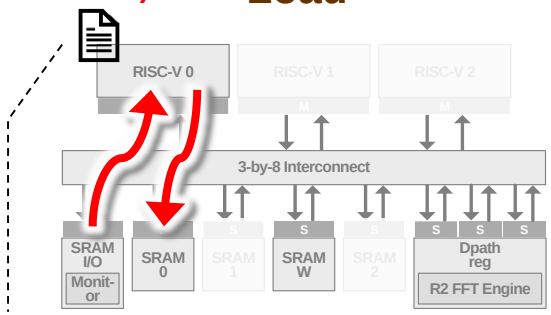
Target System

❑ Scenario

- 3-stage pipeline (load-compute-store)
- A total of 3 input/output sets

Program (ASM)
(Handwritten)

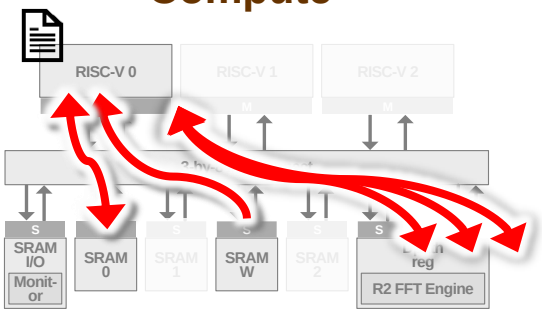
Load



Reference Code (*'L_MP_9.pdf'*)

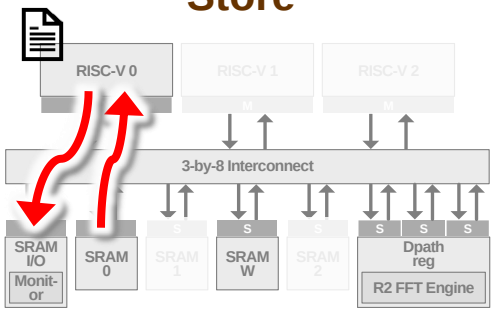
```
95          // load_input
96          for (j = 0; j < L_FFT; j++)
97          {
98              int rev = 0;
99              for (b = 0; b < N_STAGE; b++)
100             {
101                 rev |= ((j >> b) & 1) << (N_STAGE - 1 - b);
102             }
103             SRAM_0[rev] = input_region_of_SRAM_IO[set * L_FFT + j];
104         }
105         //
```

Compute




```
106         // compute_fft
107         n = 1 << N_STAGE;
108         l = 1;
109         k = N_STAGE - 1;
110         while (l < n)
111         {
112             istep = l << 1;
113             for (m = 0; m < l; ++m)
114             {
115                 j = m << k;
116                 twiddle = SRAM_W[j];
117                 for (i = m; i < n; i += istep)
118                 {
119                     j = i + l;
120                     dpath_reg_0 = SRAM_0[i];
121                     dpath_reg_1 = SRAM_0[j];
122                     dpath_reg_2 = twiddle;
123
124                     R2_FFT_Engine();
125
126                     SRAM_0[i] = dpath_reg_0;
127                     SRAM_0[j] = dpath_reg_1;
128                 }
129             }
130
131 #ifdef DEBUG 비활성 전처리기 블록
140 #endif
141             --k;
142             l = istep;
143         }
144         //
```

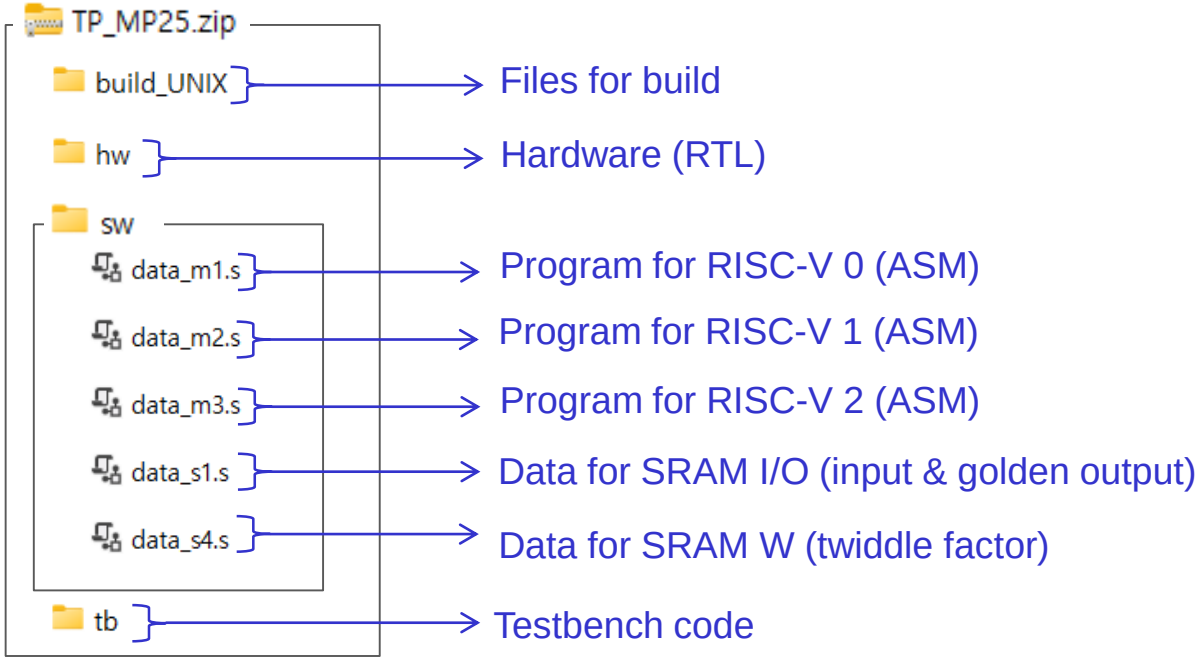
Store




```
145         // store_output
146         for (j = 0; j < L_FFT; j++)
147             output_region_of_SRAM_IO[set * L_FFT + j] = SRAM_0[j];
148         //
```

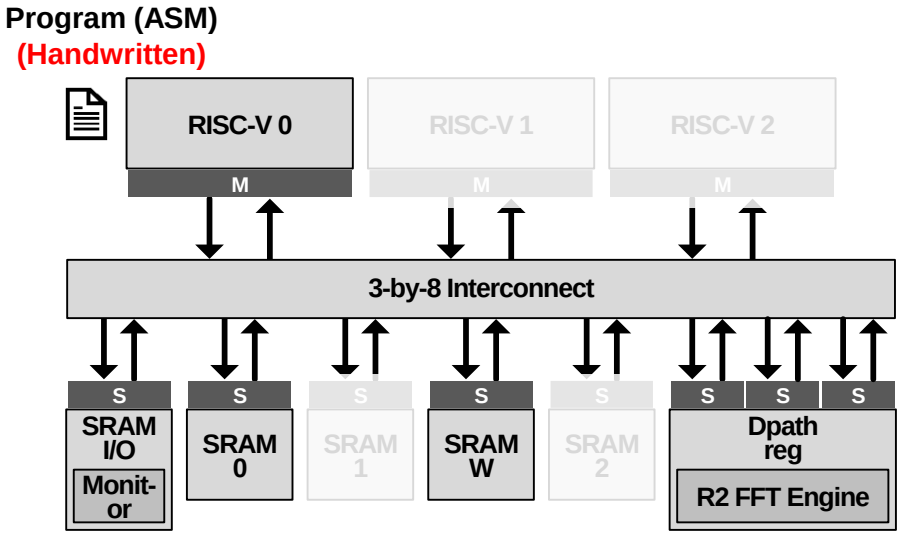
Reference Project

- Unzip ***'TP_MP25.zip'***
  - The input, golden output and twiddle factor values in ***'data_s1.s'*** and ***'data_s4.s'*** are set to be identical to those of the reference code (See ***'L_MP_9.pdf'***)
- Build project and run simulation (See ***Appendix B***)



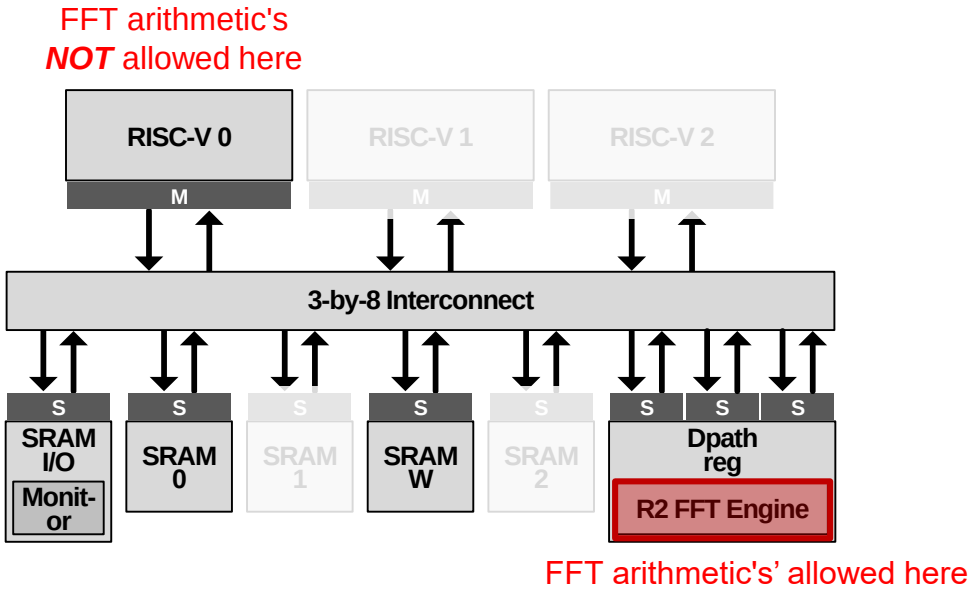
Design Constraints

- Design only the **software** part (ASM program)
  - You are ***not*** allowed to modify any other parts, e.g., the hardware part (RTL) of the target system
    - ✓ Otherwise, it won't be considered for evaluation



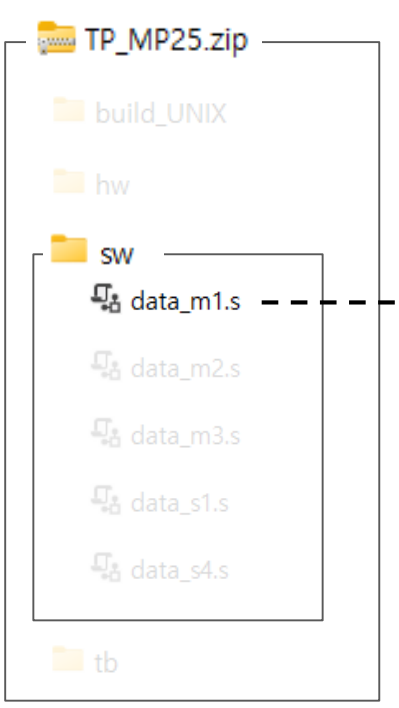
Design Constraints

- FFT arithmetic's **only** in datapath
  - FFT arithmetic's defined as bufferfly (additions/subtractions) and twiddle factor multiplications
  - You are ***not*** allowed to use any other parts, e.g., the ALU inside RISC-V cores
    - ✓ Otherwise, it won't be considered for evaluation



Design Constraints

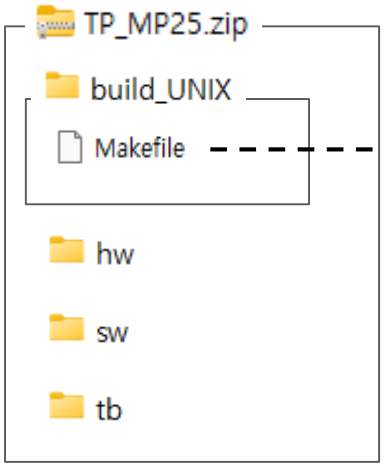
- Modify only (a part of) the body of ***data_m1.s***
  - You are allowed to change ***only*** the paragraphs between ***"Edit code below"*** and ***"Edit code above"***




```
28  start_fft:
29
30  /* Edit Code Below */
31
32      mv      s11,    x0
33  input_loop:
34      bge     s11,    x0,     exit_input_loop
35  exit_input_loop:
36
37      mv      s11,    x0
38  dummy_loop:
39      bge     s11,    s7,     exit_dummy_loop
40      sw      x0,     0       (s4)
41      sw      x0,     0       (s5)
42      sw      x0,     0       (s6)
43      lw      t0,     4       (s4)
44      lw      t1,     4       (s5)
45      slli    s11,    s11,    2
46      add     s9,     s2,     s11
47      srli    s11,    s11,    2
48      sw      t0,     0       (s9)
49      addi    s11,    s11,    1
50      jal     s11,    dummy_loop
51  exit_dummy_loop:
52
53  /* Edit Code Above */
54
55      mv      s11,    x0
56  store_output:
```

Design Constraints

- Modify only (a part of) the body of ***data_m1.s*** (cont'd)
  - You should keep any other lines unchanged, for example, ***Makefile*** (that sets the instruction set to RV32I)
    - ✓ Otherwise, it won't be considered for evaluation




```
35  THISDIR = $(shell dirname $(realpath $(lastword $(MAKEFILE_LIST))))
36  VPATH = $(THISDIR)/../hw
37  APPDIR = $(THISDIR)/../sw
38  SIMDIR = $(THISDIR)/sim
39
40  AS       := /tools/binutils/riscv/riscv64-unknown-elf/bin/as
41  ASFLAGS  := -march=rv32i -mabi=ilp32
42  OBJCOPY  := /tools/binutils//riscv/riscv64-unknown-elf/bin/objcopy
43  PMFLAGS  := -O verilog --verilog-data-width=1 -j .text
44  DMFLAGS  := -O verilog --verilog-data-width=4 -j .data
45  APP = data
46
47  SRCS     := $(wildcard $(APPDIR)/*.s)
48  OBJS     := $(SRCS:.s=.o)
49  PMS      := $(SRCS:.s=.PMb)
50  DMS      := $(SRCS:.s=.eDM)
51
52  SIMFLAGS := +fft_size=64 +num_sets=3
53
```

Evaluation

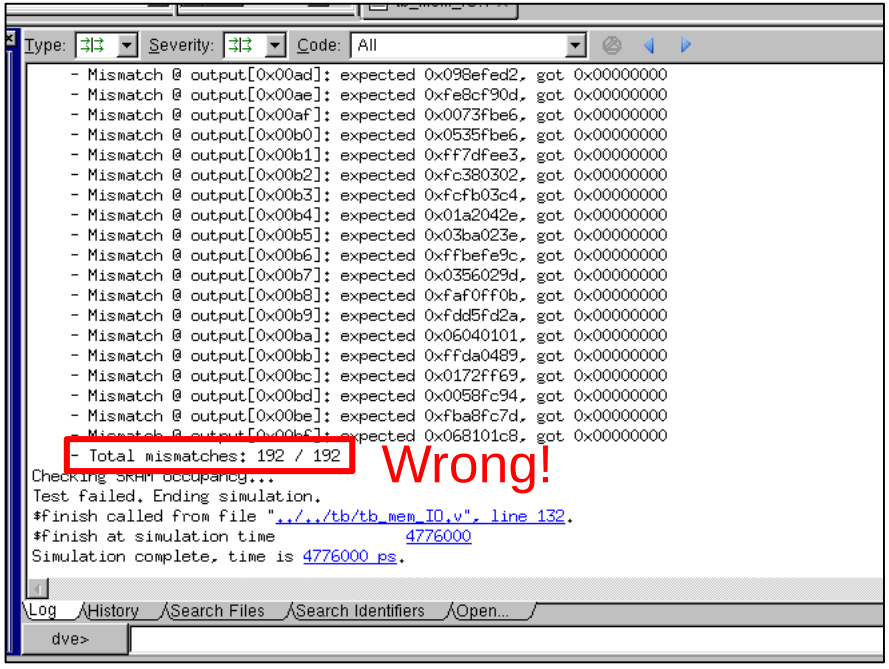
- Submission completeness
  - Reproducibility (20pt)
- Accuracy
  - Functionality (40pt)
    - ✓ In comparison to the golden output (provided in the reference project)
- Performance
  - Execution time (20pt)
    - ✓ **Won't** be considered for evaluation, if functionality fails
- Documentation
  - Explanation of the handwritten ASM code (20pt)

Reproducibility

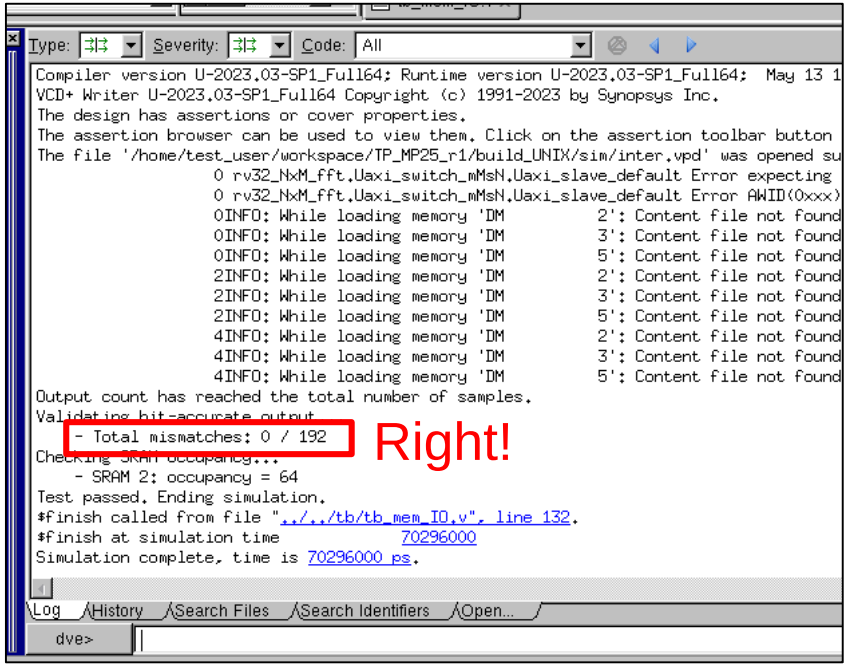
- Make sure that your submission is **complete**
  - In other words, it should be possible to reproduce your design together with the claimed accuracy and performance using **only** the files that you submitted by the submission deadline

Functionality

- Run the simulation to check functionality (see ***Appendix B***)
  - For the **3** output sets, **all** the **192** output samples must exactly match those in the golden output.



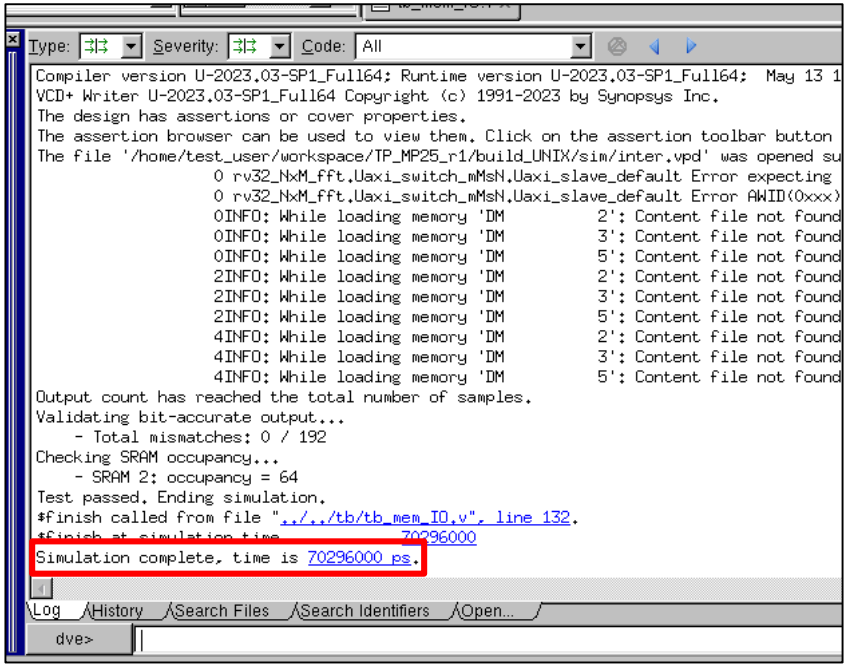
Result of the code in the reference project



Result of the corrected code

Execution Time

- Run the target system to measure the execution time (see ***Appendix A***)



Documentation

- Explain your handwritten ASM code
  - Only the paragraphs that you modified, i.e., those between ***"Edit code below"*** and ***"Edit code above"***
  - Plus, if you provide as much code detail as possible

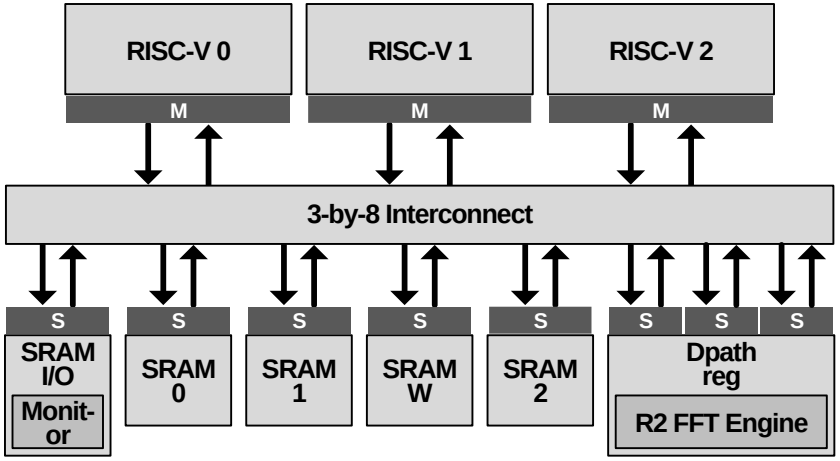
Submission

- Deadline
  - **May 30 (Fri)**, **15:00** GMT+9
- **Only one zip file** submission **per team** including the following files:
  - **ASM code (*data_m1.s*)** (needed to reproduce your design)
  - **Slideset file** (PPT) (explaining the handwritten ASM code)
- Upload the zip file to the Ecampus
  - Send to chesterku2013@gmail.com as a *backup*
- You can post questions in the Ecampus (Q&A)
- Delayed submission will result in penalty!

TP2: FFT Accelerator with Multiple Cores

Target System

- Overview
  - ***3 RISC-V cores*** (RISC-V 0, 1, 2)
    - ✓ ***Not*** used for twiddle-factor multiplications & bufferflies
  - 3-by-8 AXI interconnect
  - SRAM
    - ✓ 1R1W
    - ✓ ***5 banks*** (SRAM 0, 1, 2, I/O, W)
    - ✓ 16,384 lines/bank, 4 bytes/line
  - R2 FFT engine
    - ✓ 5 32-bit I/O registers
    - ✓ Twiddle-factor multiplications & butterflies (done only here!)
  - Monitor
    - ✓ Counts No. output sets (3)

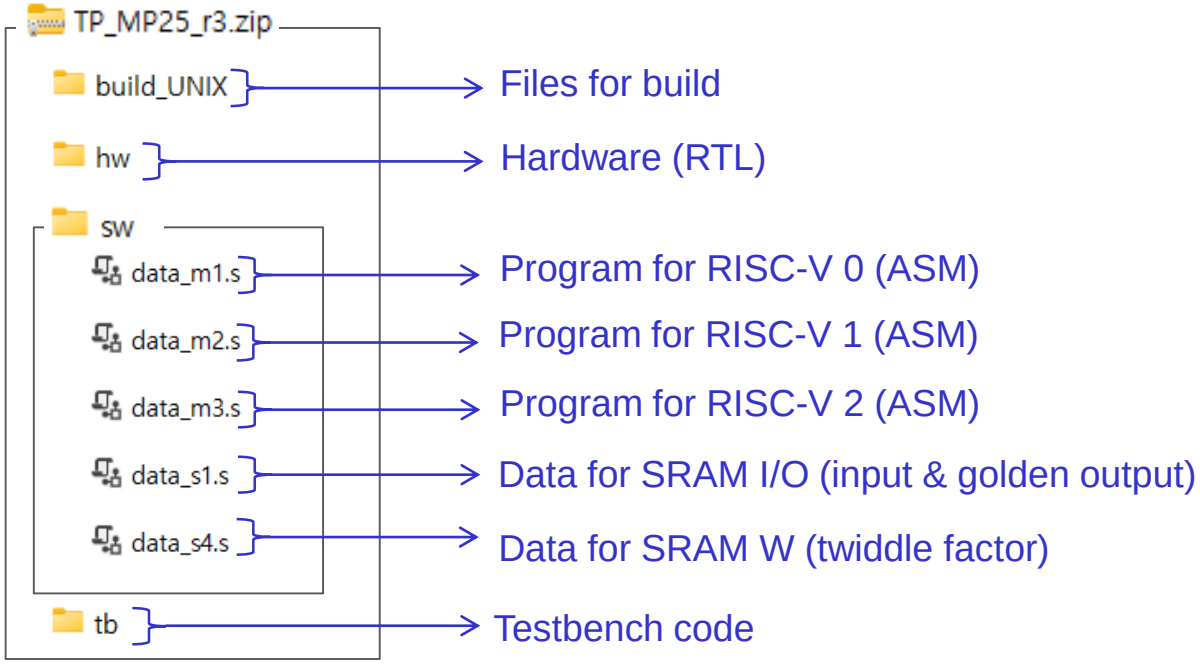


Target System

- File Tree
  - The same as for ***TP1***
- Address map
  - The same as for ***TP1***
- Data layout
  - The same as for ***TP1***

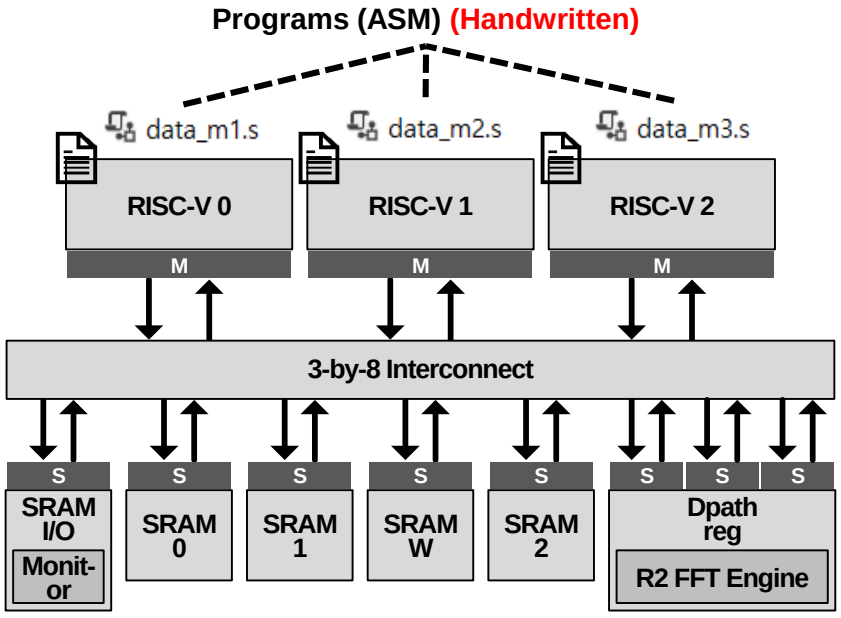
Reference Project

- Unzip ***'TP_MP25_r3.zip'***
  - The input, golden output and twiddle factor values in ***'data_s1.s'*** and ***'data_s4.s'*** are set to be identical to those of the reference code (See ***'L_MP_9.pdf'***)
- Build project and run simulation (See ***Appendix B***)



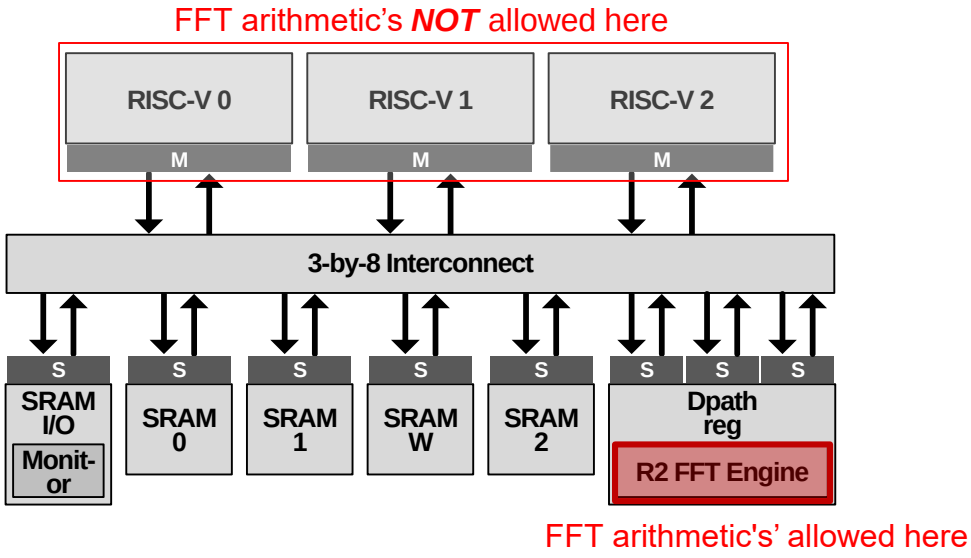
Design Constraints

- Design only the **software** parts (ASM programs)
  - You are ***not*** allowed to modify any other parts, e.g., the hardware part (RTL) of the target system
    - ✓ Otherwise, it won't be considered for evaluation



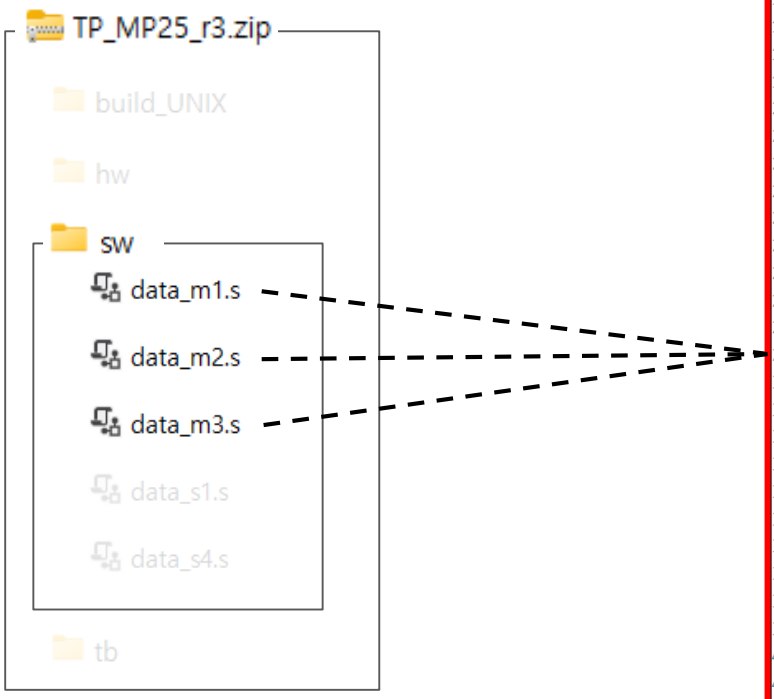
Design Constraints

- FFT arithmetic's **only** in datapath
  - FFT arithmetic's defined as bufferfly (additions/subtractions) and twiddle factor multiplications
  - You are ***not*** allowed to use any other parts, e.g., the ALU inside RISC-V cores
    - ✓ Otherwise, it won't be considered for evaluation



Design Constraints

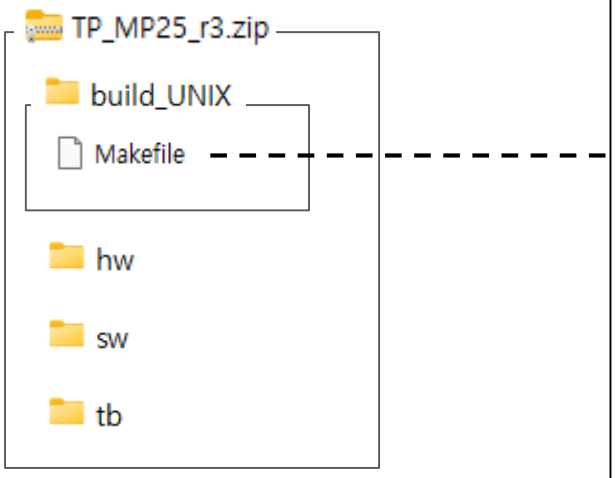
- Modify only (a part of) the body of ***data_m1.s, data_m2.s***, and ***data_m3.s***
  - You are allowed to change ***only*** the paragraphs between ***"Edit code below"*** and ***"Edit code above"***




```
16  _main_init:
17
18  /* Edit Code Below */
19
20      lui     s2,     0x00000
21      lui     s4,     0x05000
22      lui     s5,     0x06000
23      lui     s6,     0x07000
24      li      s7,     64
25
26      mv      s11,    x0
27  dummy_loop:
28      bge     s11,    s7,     exit_dummy_loop
29      sw      x0,     0       (s4)
30      sw      x0,     0       (s5)
31      sw      x0,     0       (s6)
32      lw      t0,     4       (s4)
33      lw      t1,     4       (s5)
34      slli    s11,    s11,    2
35      add     s9,     s2,     s11
36      srli    s11,    s11,    2
37      sw      t0,     0       (s9)
38      addi    s11,    s11,    1
39      jal     x0,     dummy_loop
40  exit_dummy_loop:
41
42  /* Edit Code Above */
```

Design Constraints

- Modify only (a part of) the body of ***data_m1.s***, ***data_m2.s***, and ***data_m3.s*** (cont'd)
  - You should keep any other lines unchanged, for example, ***Makefile*** (that sets the instruction set to RV32I)
    - ✓ Otherwise, it won't be considered for evaluation




```
35  THISDIR = $(shell dirname $(realpath $(lastword $(MAKEFILE_LIST))))
36  VPATH = $(THISDIR)/../hw
37  APPDIR = $(THISDIR)/../sw
38  SIMDIR = $(THISDIR)/sim
39
40  AS       := /tools/binutils/riscv/riscv64-unknown-elf/bin/as
41  ASFLAGS  := -march=rv32i -mabi=ilp32
42  OBJCOPY  := /tools/binutils//riscv/riscv64-unknown-elf/bin/objcopy
43  PMFLAGS  := -O verilog --verilog-data-width=1 -j .text
44  DMFLAGS  := -O verilog --verilog-data-width=4 -j .data
45  APP = data
46
47  SRCS     := $(wildcard $(APPDIR)/*.s)
48  OBJS     := $(SRCS:.s=.o)
49  PMS      := $(SRCS:.s=.PMb)
50  DMS      := $(SRCS:.s=.eDM)
51
52  SIMFLAGS := +fft_size=64 +num_sets=3
53
```

Design Constraints

- Do not skip the communication part
  - For each input/output set, load from/store to SRAM I/O
  - Compute on SRAM 0/1/2/W, **not** directly on SRAM I/O

Evaluation

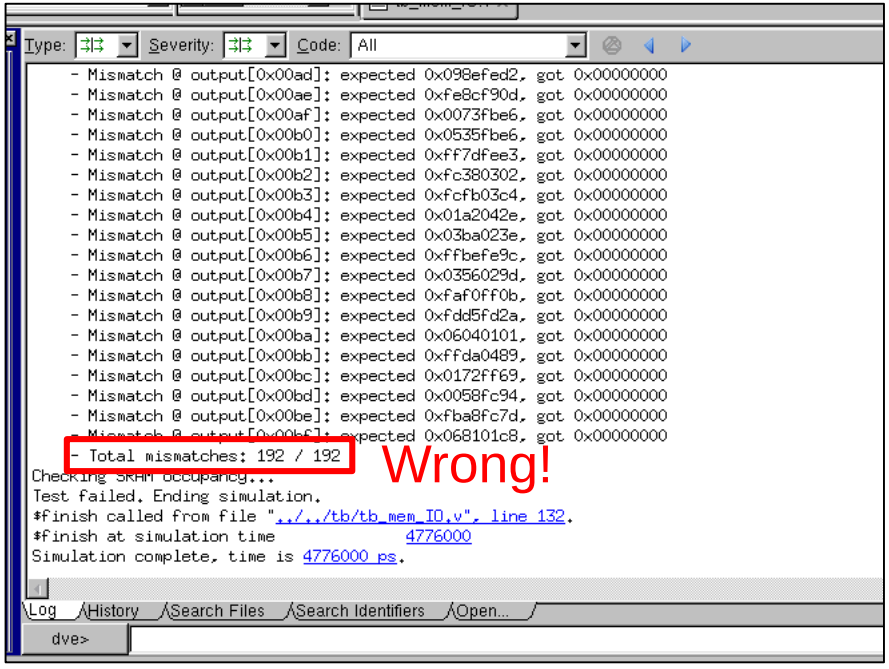
- Submission completeness
  - Reproducibility (20pt)
- Accuracy
  - Functionality (20pt)
    - ✓ In comparison to the golden output (provided in the reference project)
    - ✓ Accompanied by code review (Q&A), if necessary
- Performance
  - Execution time (20pt)
    - ✓ **Won't** be considered for evaluation, if functionality fails
    - ✓ Accompanied by code review (Q&A), if necessary
- Cost
  - Effective memory capacity for SRAM 0~2 (20pt)
    - ✓ Defined as the **minimum** address region required for 3 input/output sets (see ***Appendix D***)
    - ✓ The less capacity you propose, the better score you will get
- Documentation
  - Explanation of the handwritten ASM codes (20pt)

Reproducibility

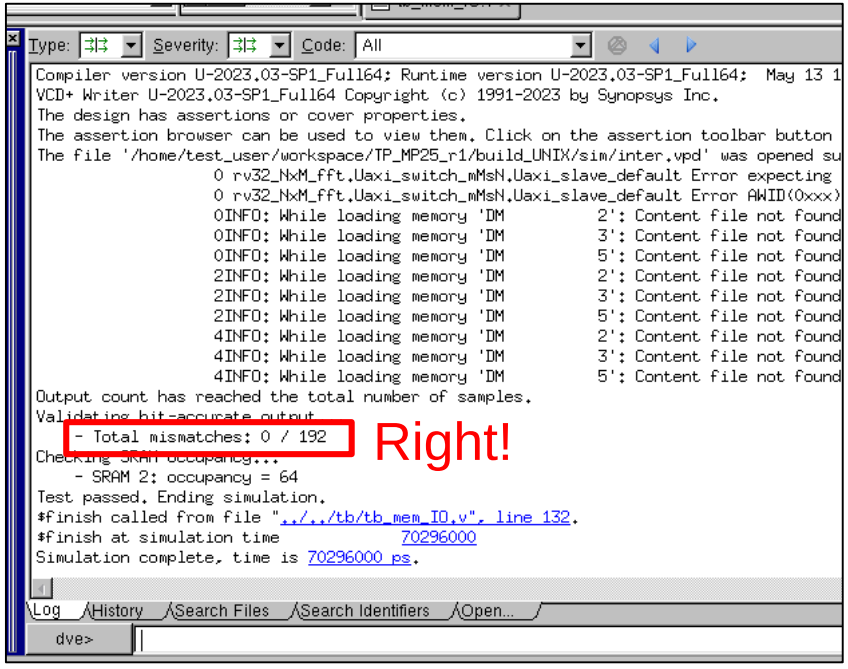
- Make sure that your submission is **complete**
  - In other words, it should be possible to reproduce your design together with the claimed accuracy and performance using **only** the files that you submitted by the submission deadline

Functionality

- Run the simulation to check functionality (see ***Appendix B***)
  - For the **3** output sets, **all** the **192** output samples must exactly match those in the golden output.



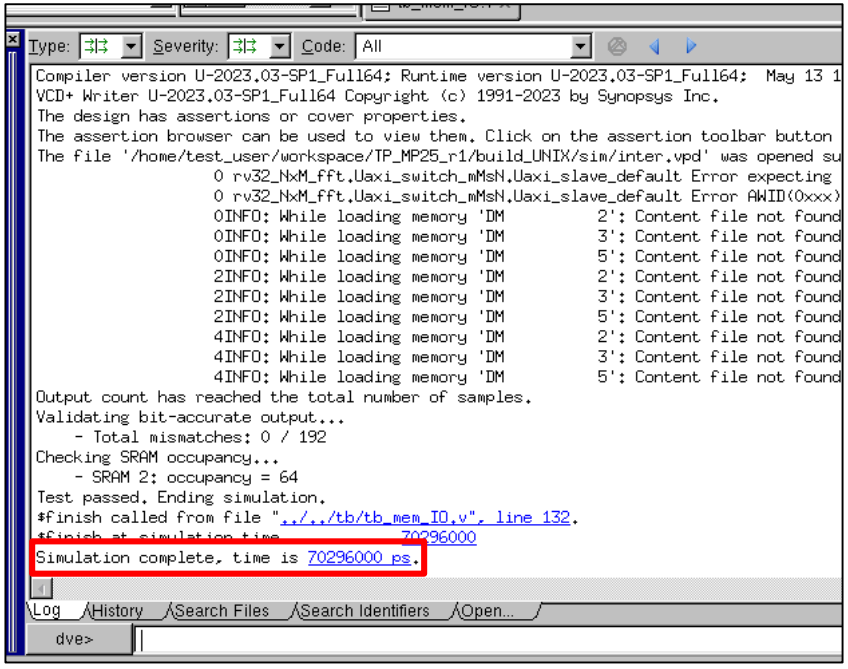
Result of the code in the reference project



Result of the corrected code

Execution Time

- Run the target system to measure the execution time (see ***Appendix A***)



Documentation

- **Screen capture** of the simulation results
  - Show (1) the functionality (p. 39) and (2) the execution time (p. 40)
  - Plus, if the relevant **roofline model** is correctly provide
    - ✓ Samuel Williams, Andrew Waterman, and David Patterson. 2009. Roofline: an insightful visual performance model for multicore architectures. Commun. ACM 52, 4 (April 2009)

Documentation

- Clear explanation your handwritten ASM codes
  - Only the paragraphs that you modified, i.e., those between ***"Edit code below"*** and ***"Edit code above"***
- Plus, if the relevant **ablation study** is well-documented
  - See ***Appendix C***

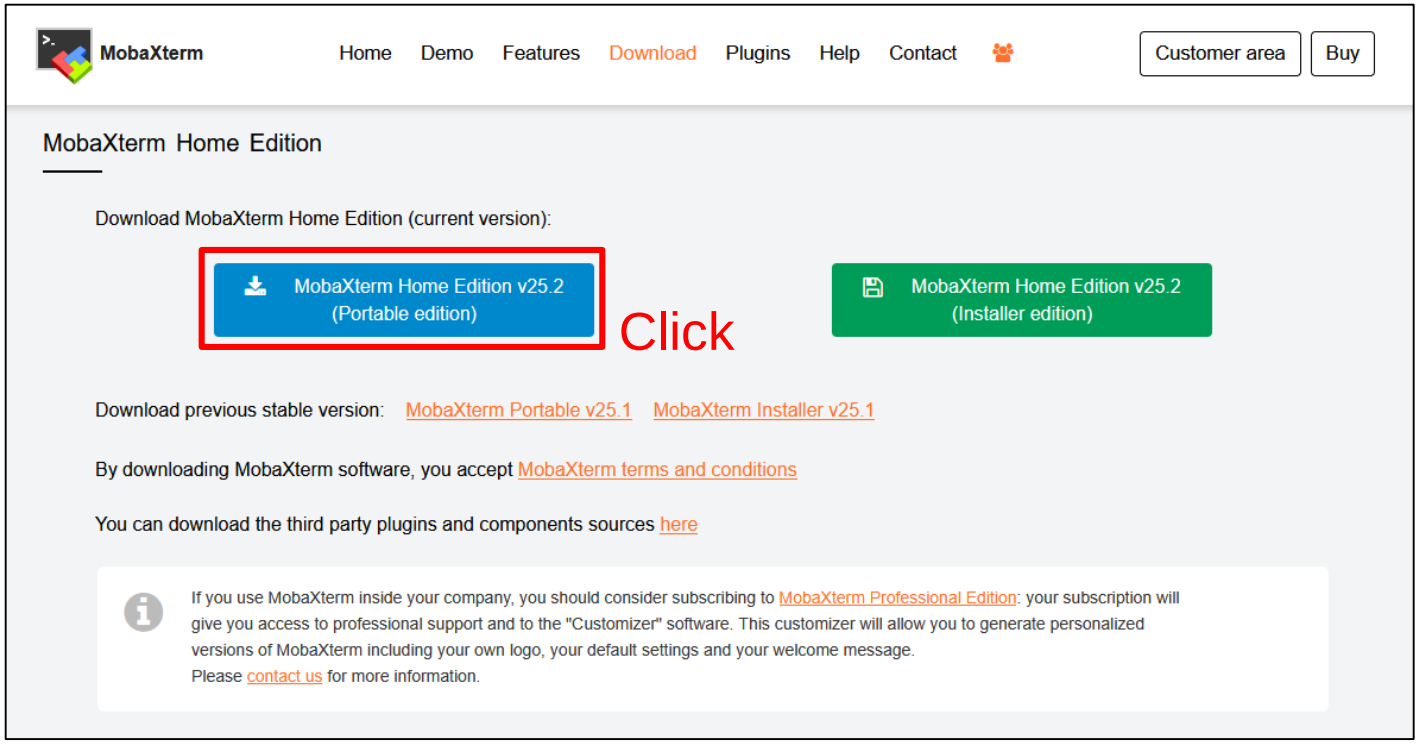
Submission

- Deadline
  - **June 12 (Thu)**, **15:00** GMT+9
- **Only one zip file** submission **per team** including the following files:
  - **ASM codes (*data_m1.s*, *data_m2.s*, and *data_m3.s*)** (needed to reproduce your design)
  - **Slideset file** (PPT) (explaining the handwritten ASM code)
- Upload the zip file to the Ecampus
  - Send to chesterku2013@gmail.com as a *backup*
- You can post questions in the Ecampus (Q&A)
- Delayed submission will result in penalty!

Appendix A:
Installing MobaXterm and Connecting to the Server

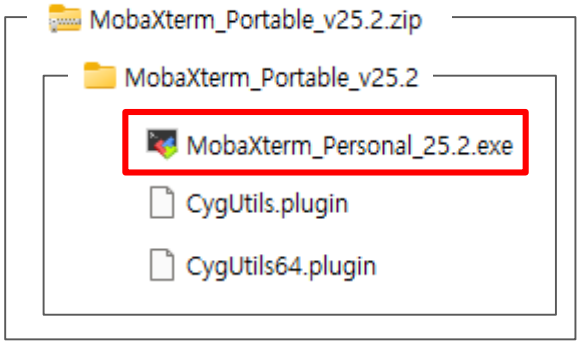
Installing MobaXterm

- Enter the link and download the file
  - https://mobaxterm.mobatek.net/download-home-edition.html

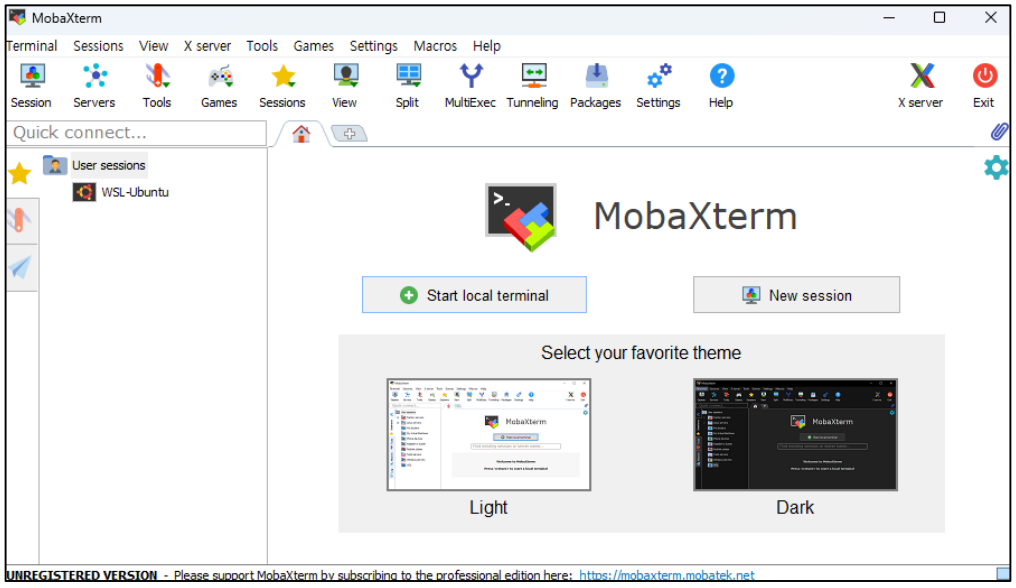


Installing MobaXterm

- Unzip the file and double-click the program

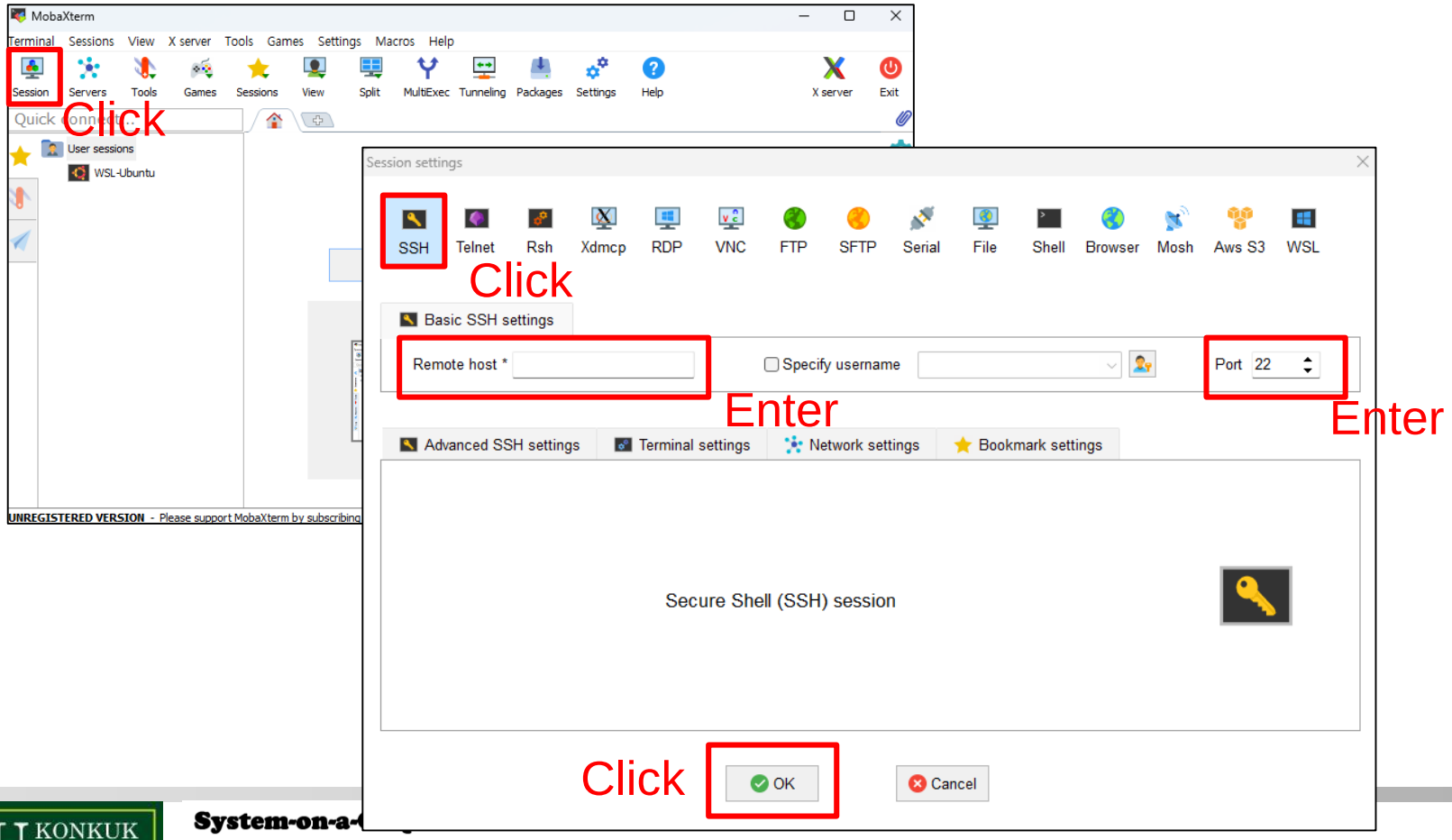


Double Click



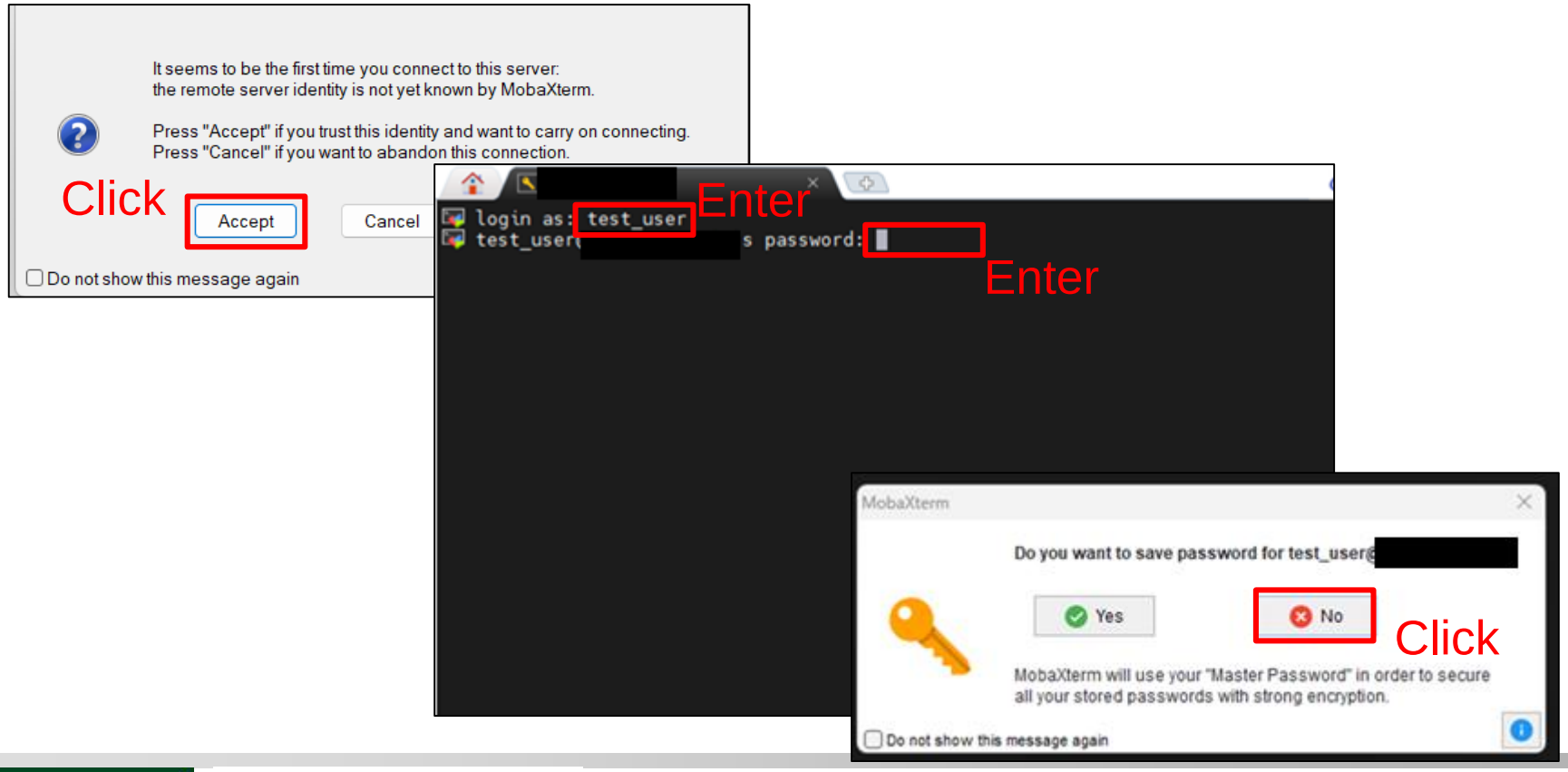
Connecting to the Server

- Click the icons and enter ***'Remote host'*** and ***'Port'*** fields
  - Refer to the Notice page in the Ecampus



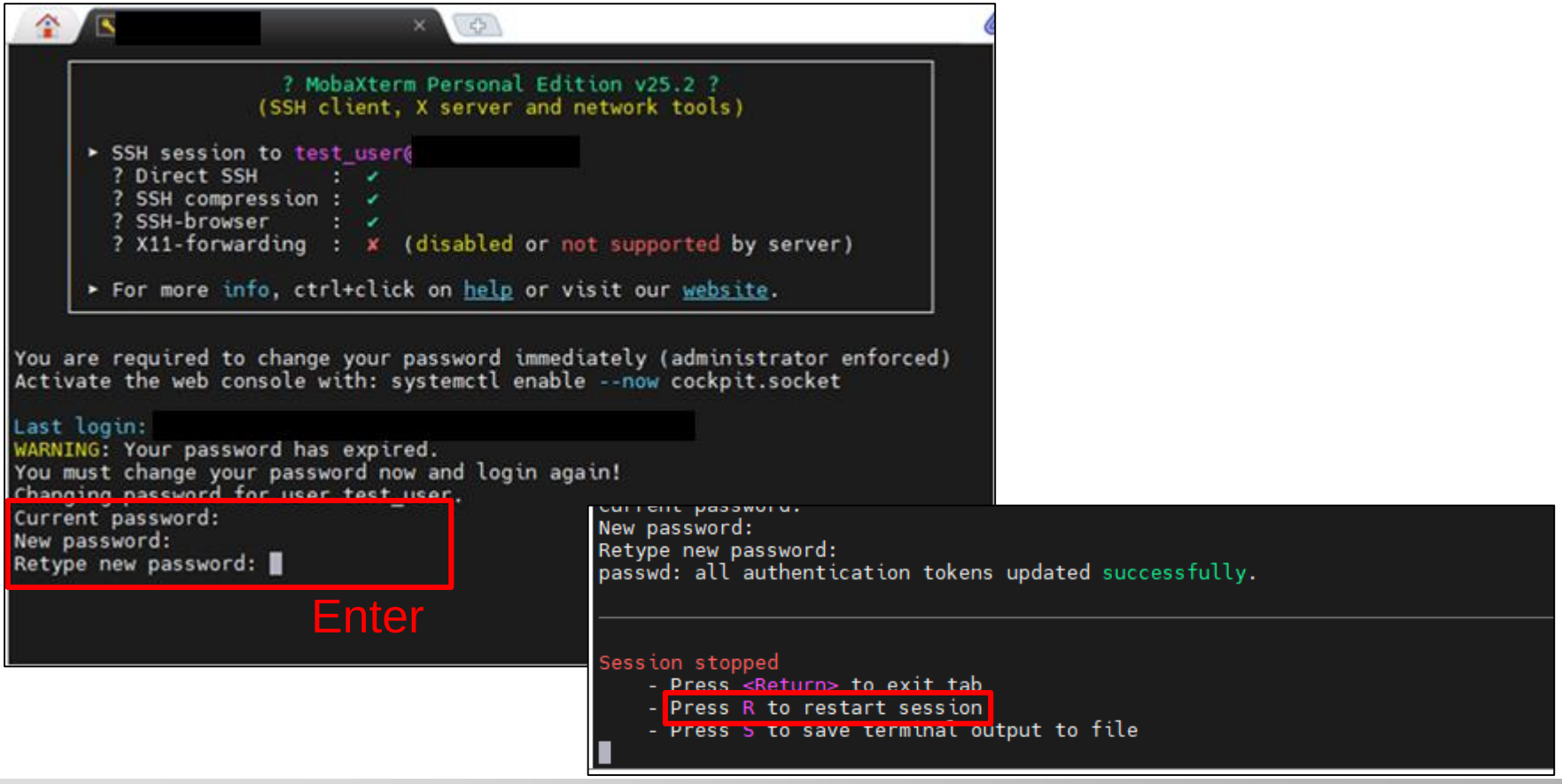
Connecting to the Server

- Click the icons and enter ***the username*** and ***the initial password***
  - Refer to the Notice page in the Ecampus



Connecting to the Server

- Enter ***a new password*** of your choice and press ***R*** to reconnect to the server

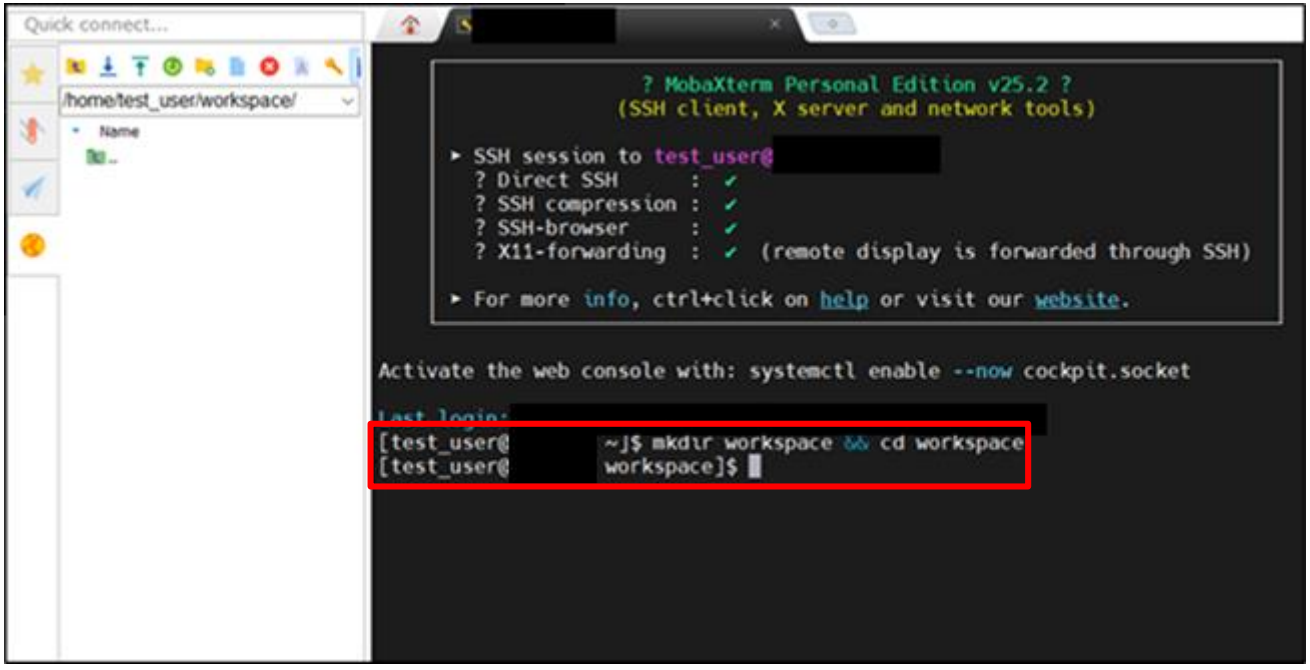


Appendix B:
Building Project and Running Simulation

Building Project

- Connect to the server (see ***Appendix A***) and enter the following commands

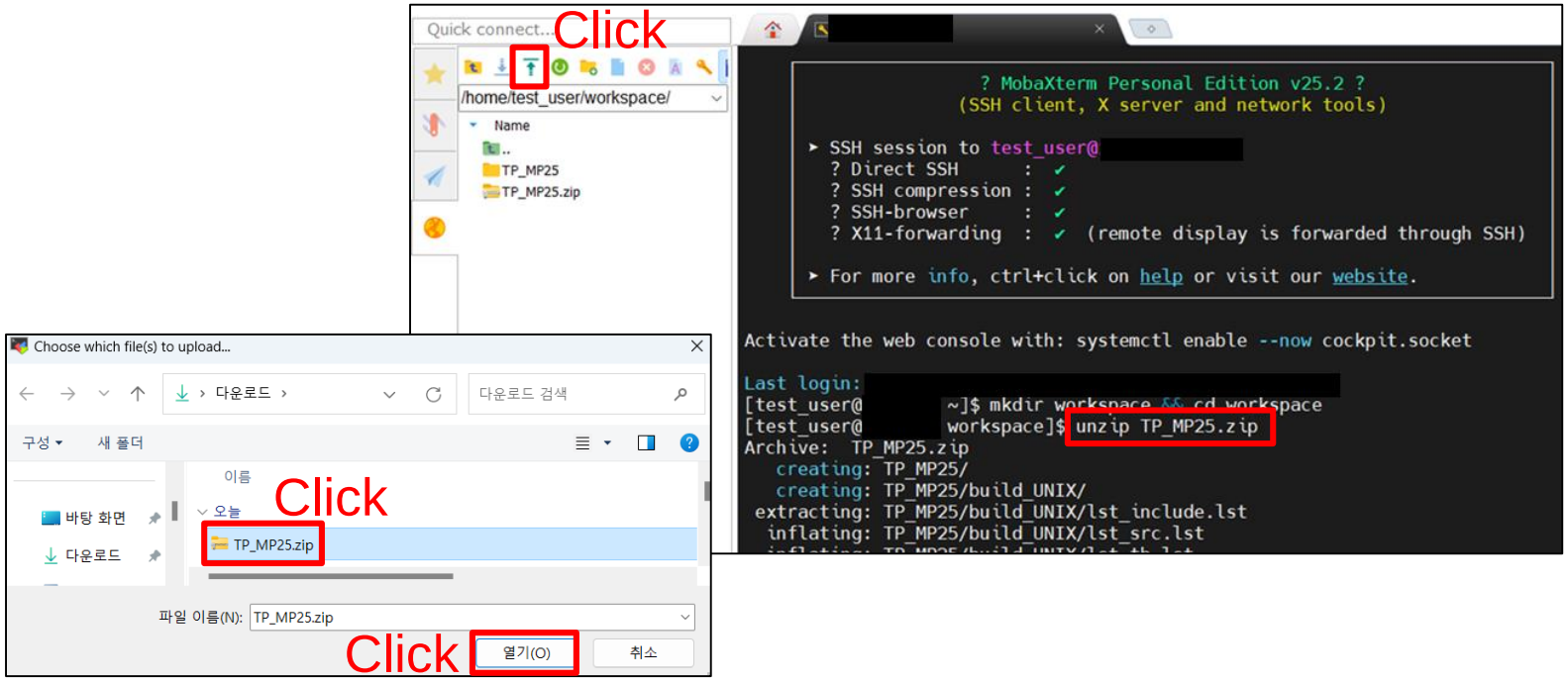
mkdir workspace && cd workspace



Building Project

❑ Upload ***'TP_MP25.zip'*** and unzip the file

unzip TP_MP25.zip

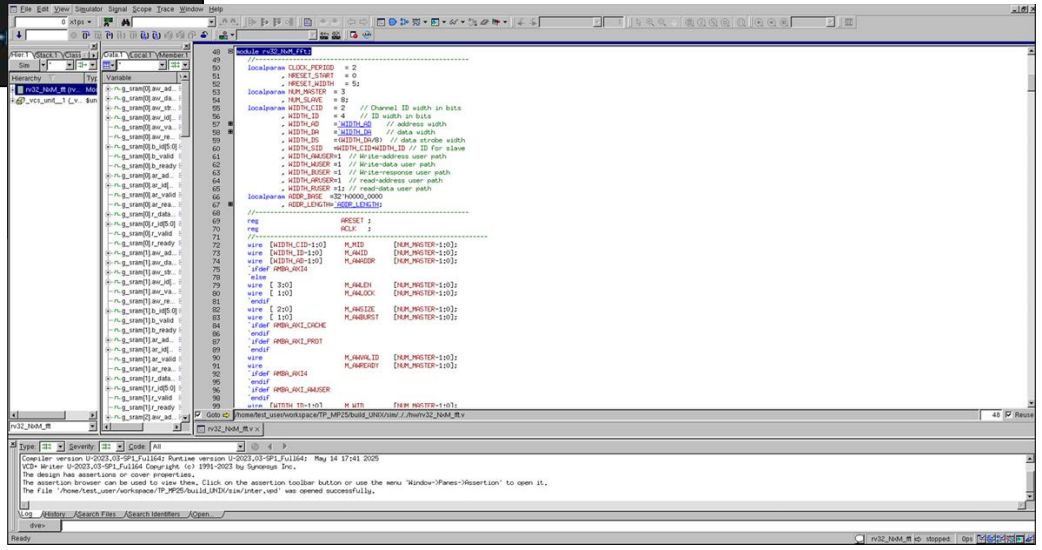


Building Project

- Change directory and build project

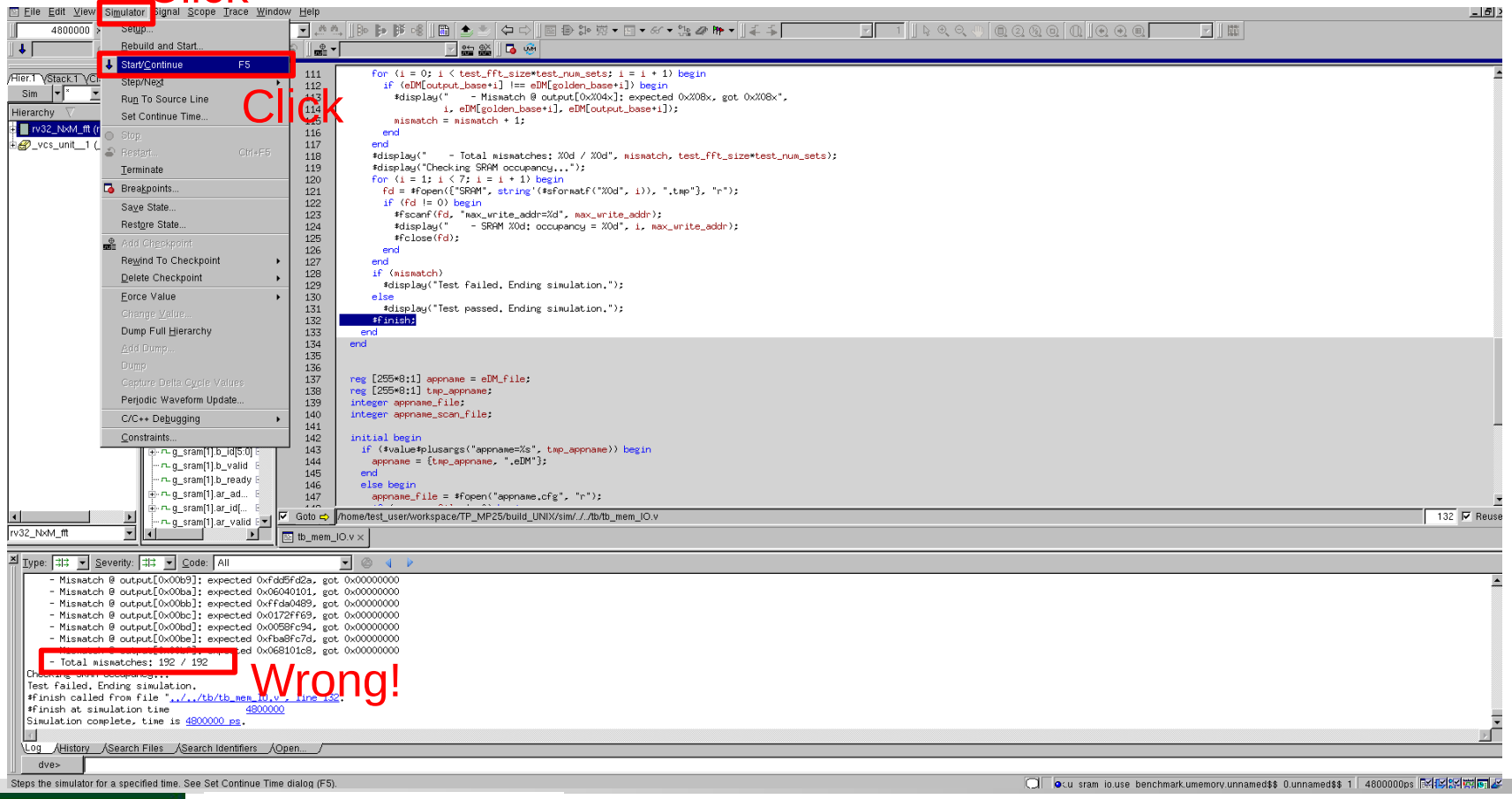
cd TP_MP25/build_UNIX/
make

```
  inflating: TP_MP25/tb/par_emulator.v
  inflating: TP_MP25/tb/tb_mem_IO.v
  inflating: TP_MP25/tb/tb_mem_PMb.v
[test_user@          workspace]$ cd TP_MP25/build_UNIX/
[test_user@          build_UNIX]$ make
/tools/binutils/riscv/riscv64-unknown-elf/bin/as -march=rv32i -
space/TP_MP25/build_UNIX/../sw/data_m1.s
/tools/binutils/riscv/riscv64-unknown-elf/bin/as -
space/TP_MP25/build_UNIX/../sw/data_m2.s
```



Running Simulation

- Press F5 or click the icons to run simulation
  - Write your own ASM code and run again

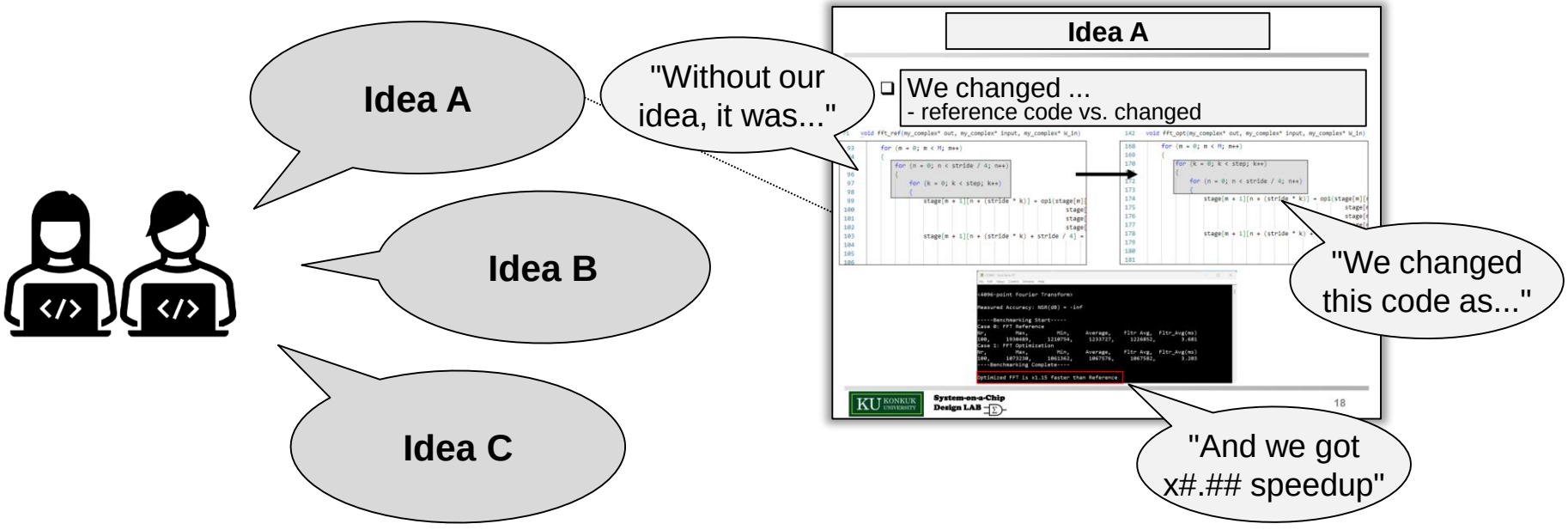


Result of the code provided in the reference project

Appendix C:
Documenting The Ablation Study

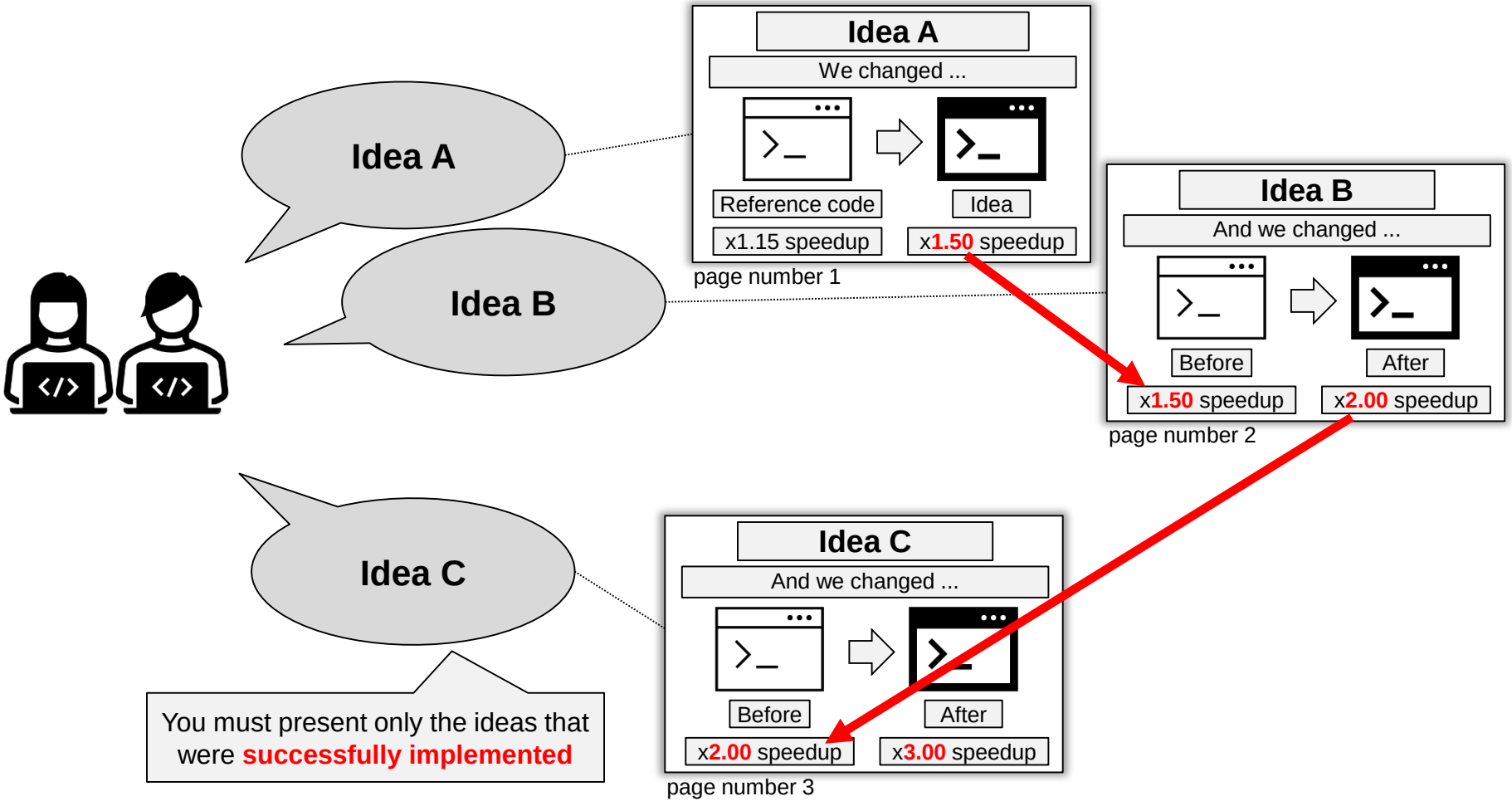
Ablation Study

- Propose your ideas with the **metrics**(execution time)
  - For each idea you've implemented,
    - ✓explain the idea briefly
    - ✓and show the measurement of the speedup achieved with it.



Ablation Study

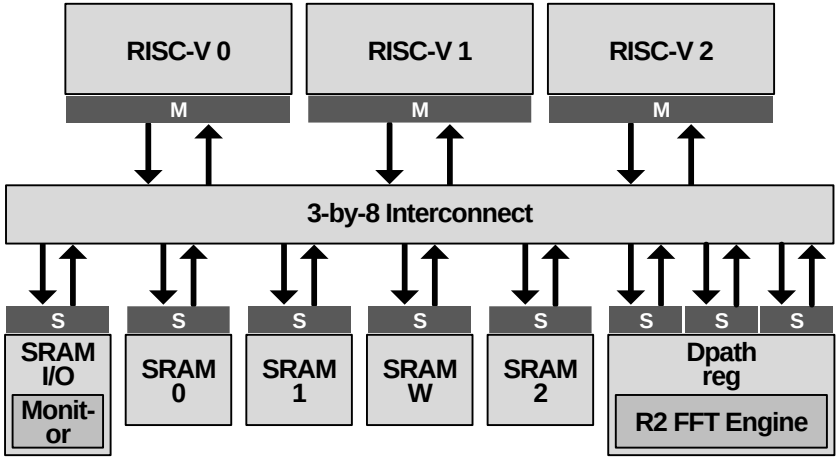
- Present **all successfully implemented** ideas

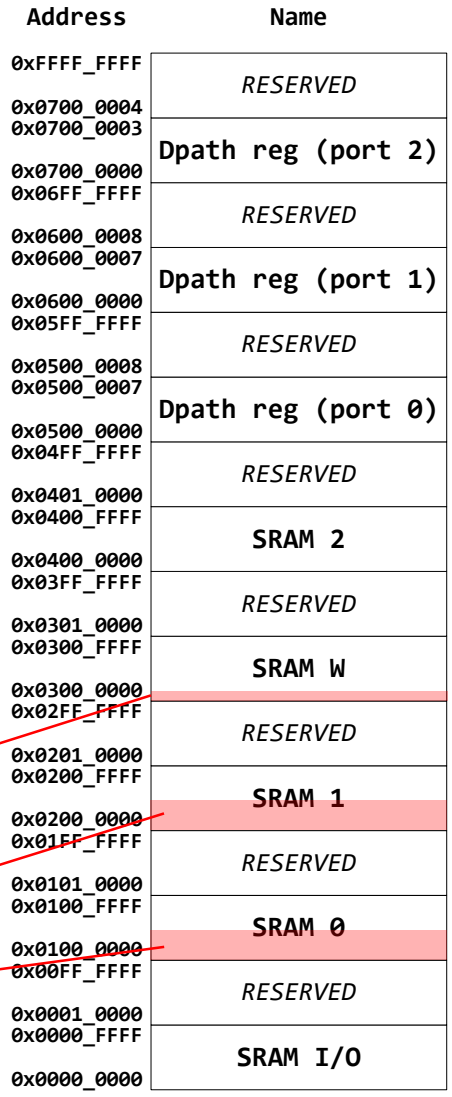


Appendix D:
Effeictive Memory Capacity

Effective Memory Capacity

- Example*
  - Only lower 256/256/8-byte address region for SRAM 0/1/2 required for 3 input/output sets





* The effective memory capacity may be proposed to be more or less than in this example (the less capacity you propose, the better score you will get!)