

Microprocessor Applications

(Term Project 1)

TP (1)

2025. 5. 30.

Students

202110410 조민우

202110365 김상만

- **64pt FFT Accelerator with RISC-V**

- **Result of TP_1**
- **Code Review**
- **# of lw/sw instructions**
- **Row-Wise Calculation**

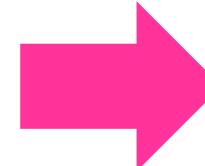
Result of TP_1

■ Accuracy & Performance

```
Type: [3] Severity: [3] Code: All
Chronologic WCS simulator copyright 1991-2023
Contains Synopsys proprietary information.
Compiler version U-2023.03-SP1_Full64; Runtime version U-2023.03-SP1_Full64; May 30 12:05 2025
VCD+ Writer U-2023.03-SP1_Full64 Copyright (c) 1991-2023 by Synopsys Inc.
The design has assertions or cover properties.
The assertion browser can be used to view them. Click on the assertion toolbar button or use the menu 'Window->Panes->Assertion' to open it.
The file '/home/student_1/workspace/TP_MP25/build_UNIX/sim/interv.vpd' was opened successfully.

0 rv32_NxM_fft.Uaxi_switch_mMsN.Uaxi_slave_default Error expecting WLAST
0 rv32_NxM_fft.Uaxi_switch_mMsN.Uaxi_slave_default Error AWID(0xxx):WID(0xxx) mismatch
0INFO: While loading memory 'DM      2': Content file not found, skipping.
0INFO: While loading memory 'DM      3': Content file not found, skipping.
0INFO: While loading memory 'DM      5': Content file not found, skipping.
2INFO: While loading memory 'DM      2': Content file not found, skipping.
2INFO: While loading memory 'DM      3': Content file not found, skipping.
2INFO: While loading memory 'DM      5': Content file not found, skipping.
4INFO: While loading memory 'DM      2': Content file not found, skipping.
4INFO: While loading memory 'DM      3': Content file not found, skipping.
4INFO: While loading memory 'DM      5': Content file not found, skipping.

Output count has reached the total number of samples.
Validating bit-accurate output...
- Total mismatches: 0 / 192
Checking SRAM occupancy...
- SRAM 2: occupancy = 64
Test passed. Ending simulation.
$finish called from file "../../tb/tb_mem_IO.v", line 132.
$finish at simulation time 14190000
Simulation complete, time is 14190000 ps.
```



• Accuracy

- Total mismatches : 0/192

• Performance

- Simulation Time : 14,190,000 [ps]

■ Implement POV

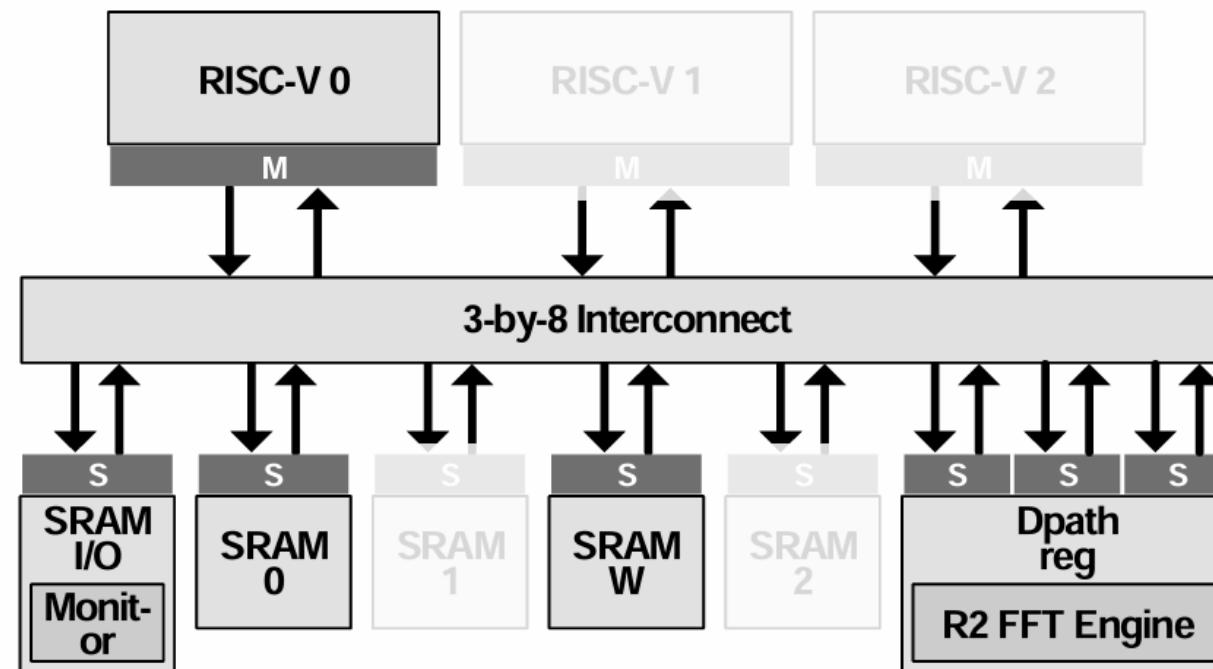
- Row Wise Operation (stage 0~3)
- Reuse Twiddle Factor (stage 1~4)
- FFT Reordering (stage 4)
- Register Pipelining (stage 4)
- Loop Unrolling (stage 0~5)

■ Code Review

■ Implement POV

- Row Wise Operation (How?)

- Utilize Register for do not store & load output to SRAM_0 in stage[0:3]

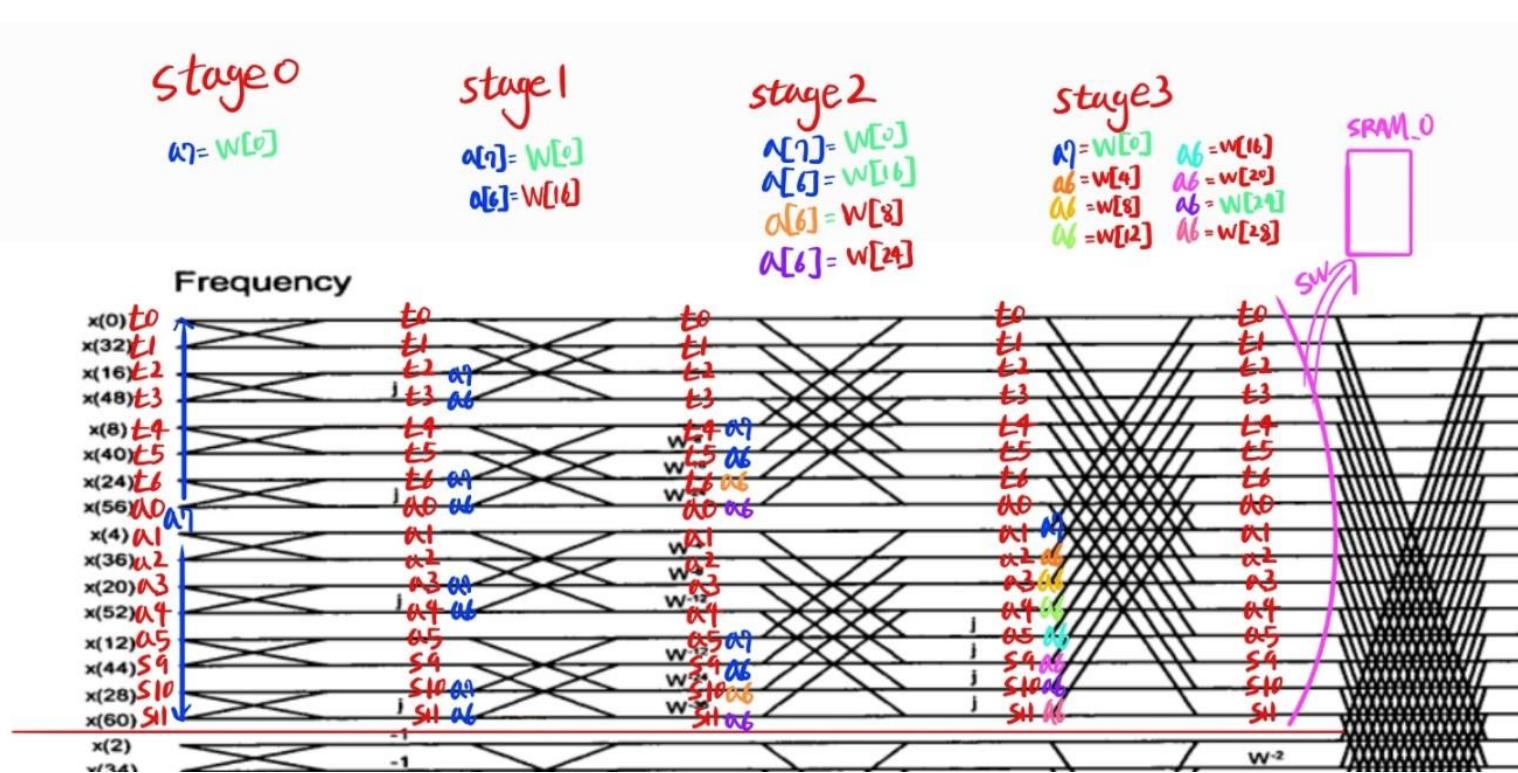


Code Review (2/7)

■ Implement POV-cont'd

- Row Wise Operation (How?)

- Utilize Register for do not store & load output to SRAM_0 in stage[0:3]



Name	Register Number	Usage	Saver
zero	x0	constant 0 (hardwired)	—
ra	x1	return address (link register)	Caller
sp	x2	stack pointer	Callee
gp	x3	global pointer	—
tp	x4	thread pointer	—
t0 – t2	x5 – x7	temporaries	Caller
s0 / fp	x8	saved register / frame pointer	Callee
s1	x9	saved register	Callee
a0 – a1	x10 – x11	function arguments / return values	Caller
a2 – a7	x12 – x17	function arguments	Caller
s2 – s11	x18 – x27	saved registers	Callee
t3 – t6	x28 – x31	temporaries	Caller

Code Review (3/7)

■ Implement POV-cont'd

- Row Wise Operation (Stage 0 for example)

```
41 # i = 0
42 # n = 0
43 # stage[0][0]
44
45 # stage[0][0]의 연산 결과를 stage[0][1]로 전달하는데
46 # SRAM 0의 sw/lw 과정을 제하기위해 가용한 reg를 최대한 사용하였다.
47
48 # Input
49 # 아래는 DIT 형식에 맞추어 SRAM I/O으로부터의 Input을 받는 과정이다.
50 # e.g. x[0], x[32], x[6] ... 순서...
51 lw t0, 0(s0)
52 lw t1, 128(s0)
53 lw t2, 64(s0)
54 lw t3, 192(s0)
55 lw t4, 32(s0)
56 lw t5, 160(s0)
57 lw t6, 96(s0)
58 lw a0, 224(s0)
59
60 lw a1, 16(s0)
61 lw a2, 144(s0)
62 lw a3, 80(s0)
63 lw a4, 208(s0)
64 lw s9, 176(s0)
65 lw s10, 112(s0)
66 lw s11, 240(s0)
67
68
69 # BF 연산
70 # reg 활용을 극대화 하기위해 연산을 위해 sw한 reg의 값을
71 # 더 이상 필요가 없으므로 연산의 결과값으로 바로 대체하였다.
72
73 # BF 0
74 sw t0, 0(s4)#Input1
75 sw t1, 0(s5)#Input2
76 sw a7, 0(s6)#Twiddle 0
77 lw t0, 4(s4)# BitSet 결과
78 lw t1, 4(s5)# BitSet 결과
79
80
```

16 Register

```
102 # BF 4:
103 sw a1, 0(s4)
104 sw a2, 0(s5)
105 sw a7, 0(s6)
106 lw a1, 4(s4)
107 lw a2, 4(s5)
108
109 # BF 5:
110 sw a3, 0(s4)
111 sw a4, 0(s5)
112 sw a7, 0(s6)
113 lw a3, 4(s4)
114 lw a4, 4(s5)
115
116 # BF 6:
117 sw a5, 0(s4)
118 sw s9, 0(s5)
119 sw a7, 0(s6)
120 lw a5, 4(s4)
121 lw s9, 4(s5)
122
123 # BF 7:
124 sw s10, 0(s4)
125 sw s11, 0(s5)
126 sw a7, 0(s6)
127 lw s10, 4(s4)
128 lw s11, 4(s5)
129
130 # 이렇게 하면 stage[0][0]의 연산이 종료되었고 결과는 reg에 저장되어 있다.
131 # 이 결과를 stage[0][1]에 전달하여 다시 BF 연산을 진행한다.
132
133 #n=1
134 #stage[0][1]
```

Go to Next Stage

```
321 # BF 7:
322 sw a0, 0(s4)
323 sw s11, 0(s5)
324 sw a6, 0(s6)#Twiddle 20
325 lw a0, 4(s4)
326 lw s11, 4(s5)
327
328 #stage 3로 넘겨 주기 위해 SRAM에 순서대로 저장한다.
329 #stage 3부터는 reg 개수의 제한이 있어 SRAM을 활용한다.
330 sw t0, 0(s2)
331 sw t1, 4(s2)
332 sw t2, 8(s2)
333 sw t3, 12(s2)
334 sw t4, 16(s2)
335 sw t5, 20(s2)
336 sw t6, 24(s2)
337 sw a0, 28(s2)
338
339 sw a1, 32(s2)
340 sw a2, 36(s2)
341 sw a3, 40(s2)
342 sw a4, 44(s2)
343 sw a5, 48(s2)
344 sw s9, 52(s2)
345 sw s10, 56(s2)
346 sw s11, 60(s2)
347
348
349
350
351
```

Store Output !!

#이렇게 stage[0][0]~stage[0][3]의 연산 및 SRAM0로의 저장이 종료되었다.
#이러한 방식으로 stage[1][0]~stage[1][3] 연산을 i=0~3 까지 진행한 후에
#stage 4, 5 연산을 시작하여 메모리 접근을 최소화하고
#더욱 효율적인 시간 단축 효과를 확인하였다.

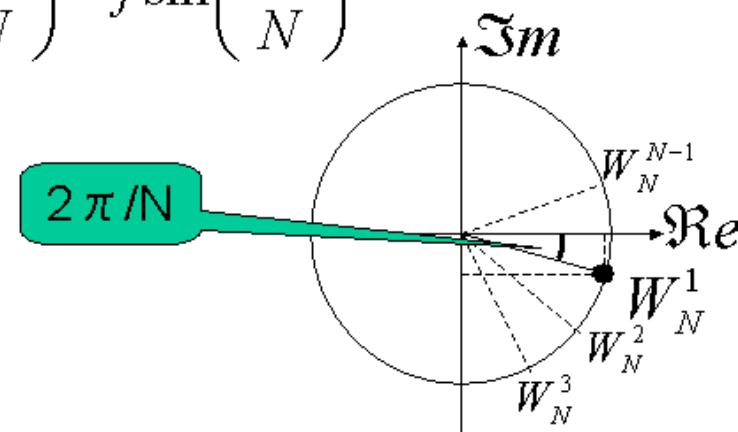
[data_m1.s]

■ Implement POV-cont'd

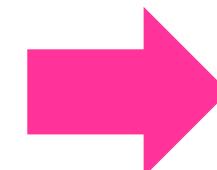
- Reuse Twiddle Factor

$$W_N = e^{-j\left(\frac{2\pi}{N}\right)}$$

$$= \cos\left(\frac{2\pi}{N}\right) - j \sin\left(\frac{2\pi}{N}\right)$$



[Twiddle Factor]

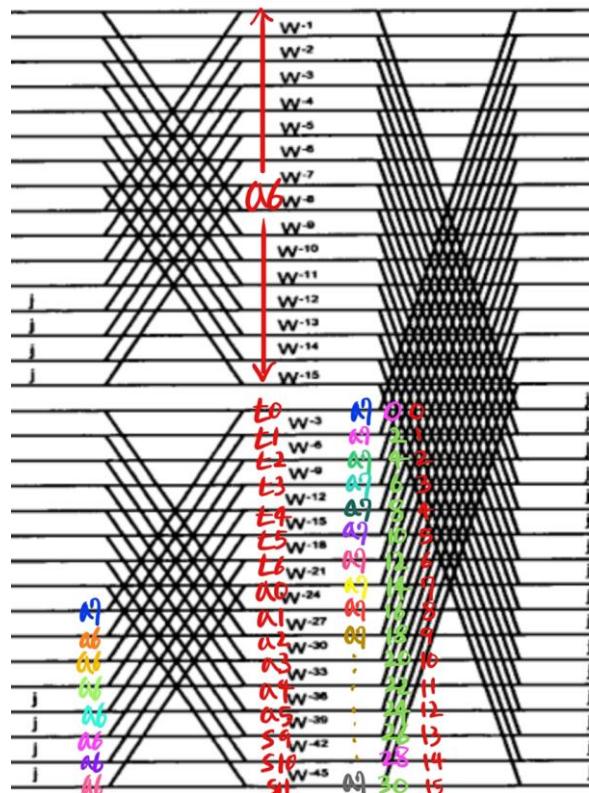


37 # reg에 Twiddle Factor을 담아두고, 재사용하려한다.
38 lw a7, 0(\$3) # a7 = twiddle[0]
39 lw a6, 64(\$3) # a6 = twiddle[16]

Code Review (5/7)

■ Implement POV-cont'd

- FFT Reordering



```

1093
1094 # 버터플라이 16: i=32, j=48 (twiddle[0] 사용 - 고정 레지스터)
1095 lw a6, 128(s2)      # SRAM_0[32]
1096 sw a6, 0(s4)
1097 sw t0, 0(s5)
1098 sw a7, 0(s6)
1099 lw a6, 4(s4)
1100 lw t0, 4(s5)
1101 sw a6, 128(s2)
1102 sw t0, 192(s2)
1103
1104 # 버터플라이 17: i=33, j=49 (twiddle[2] 사용) 데이터 디펜던시 확인
1105 lw t0, 8(s3)        # twiddle[2] 로드
1106 lw a6, 132(s2)      # SRAM_0[33]
1107 sw a6, 0(s4)
1108 sw t1, 0(s5)
1109 sw t0, 0(s6)
1110 lw a6, 4(s4)
1111 lw t1, 4(s5)
1112 sw a6, 132(s2)
1113 sw t1, 196(s2)
1114
1115 # 버터플라이 18: i=34, j=50 (twiddle[4] 사용)
1116 lw t1, 16(s3)       # twiddle[4] 로드
1117 lw a6, 136(s2)      # SRAM_0[34]
1118 sw a6, 0(s4)
1119 sw t2, 0(s5)
1120 sw t1, 0(s6)
1121 lw a6, 4(s4)
1122 lw t2, 4(s5)
1123 sw a6, 136(s2)
1124 sw t2, 200(s2)
1125

```

```

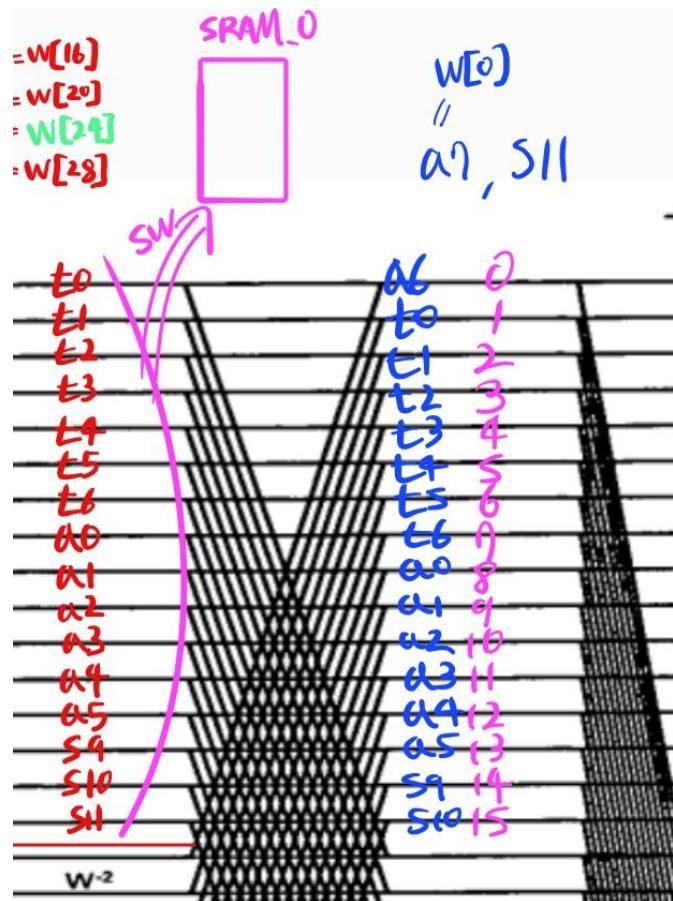
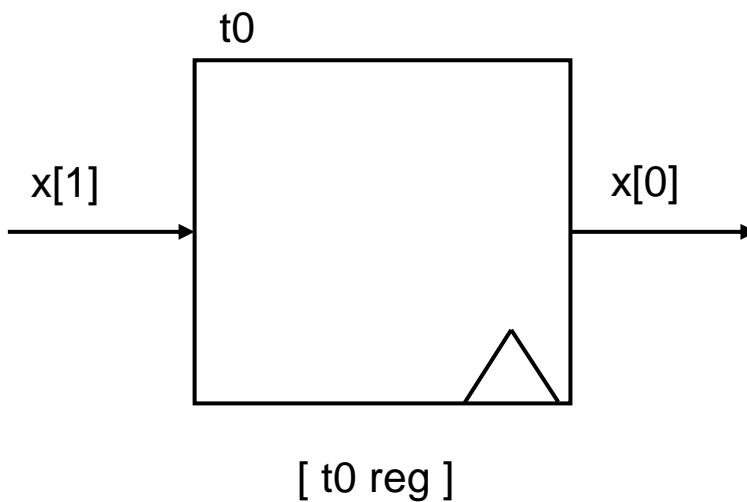
1269 # 버터플라이 1: i=1, j=17 (twiddle[2] 사용)
1270 lw a6, 4(s2)          # SRAM_0[1]
1271 lw s11, 68(s2)        # SRAM_0[17]
1272 sw a6, 0(s4)
1273 sw s11, 0(s5)
1274 sw t0, 0(s6)
1275 lw t0, 4(s4)
1276 lw s11, 4(s5)
1277 sw s11, 68(s2)
1278
1279 # 버터플라이 2: i=2, j=18 (twiddle[4] 사용)
1280 lw a6, 8(s2)          # SRAM_0[2]
1281 lw s11, 72(s2)        # SRAM_0[18]
1282 sw a6, 0(s4)
1283 sw s11, 0(s5)
1284 sw t1, 0(s6)
1285 lw t1, 4(s4)
1286 lw s11, 4(s5)
1287 sw s11, 72(s2)
1288
1289 # 버터플라이 3: i=3, j=19 (twiddle[6] 사용)
1290 lw a6, 12(s2)         # SRAM_0[3]
1291 lw s11, 76(s2)        # SRAM_0[19]
1292 sw a6, 0(s4)
1293 sw s11, 0(s5)
1294 sw t2, 0(s6)
1295 lw t2, 4(s4)

```

Code Review (6/7)

Implement POV-cont'd

- Register Pipelining



```

1280
1281 lw a6, 4(s2) # SRAM_0[1]
1282 lw s11, 68(s2) # SRAM_0[17]
1283 sw a6, 0(s4)
1284 sw s11, 0(s5)
1285 sw t0, 0(s6)
1286 lw t0, 4(s4) //
1287 lw s11, 4(s5)
1288 sw s11, 68(s2)

# 버터플라이 2: i=2, j=18 (twiddle[4] 사용)
1290 lw a6, 8(s2) # SRAM_0[2]
1291 lw s11, 72(s2) # SRAM_0[18]
1292 sw a6, 0(s4)
1293 sw s11, 0(s5)
1294 sw t1, 0(s6)
1295 lw t1, 4(s4)
1296 lw s11, 4(s5)
1297 sw s11, 72(s2)

# 버터플라이 3: i=3, j=19 (twiddle[6] 사용)
1300 lw a6, 12(s2) # SRAM_0[3]
1301 lw s11, 76(s2) # SRAM_0[19]
1302 sw a6, 0(s4)
1303 sw s11, 0(s5)
1304 sw t2, 0(s6)
1305 lw t2, 4(s4)
1306 lw s11, 4(s5)
1307 sw s11, 76(s2)

# 버터플라이 4: i=4, j=20 (twiddle[8] 사용)
1310 lw a6, 16(s2) # SRAM_0[4]
1311 lw s11, 80(s2) # SRAM_0[20]
1312 sw a6, 0(s4)
1313 sw s11, 0(s5)
1314 sw t3, 0(s6)
1315 lw t3, 4(s4)
1316 lw s11, 4(s5)
1317 sw s11, 80(s2)
1318

```

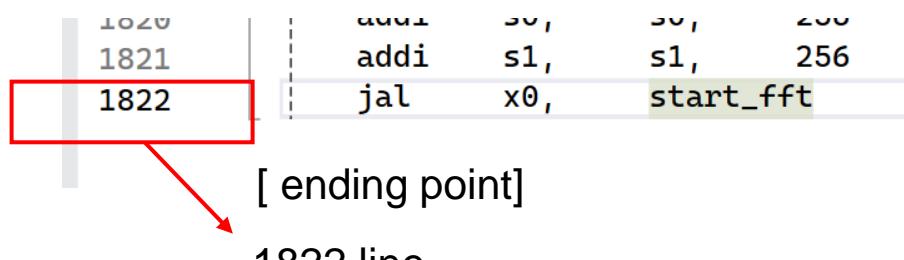
[data_m1.s]

Code Review (7/7)

■ Implement POV-cont'd

- Loop Unrolling

1 branch inst → 4000ps



[Some unrolled code]

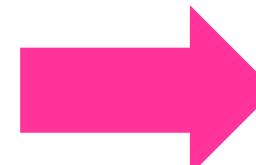
686	sw	t1, 0(\$4)	746	sw	a3, 0(\$4)
687	sw	t3, 0(\$5)	747	sw	s10, 0(\$5)
688	sw	a6, 0(\$6)	748	sw	a6, 0(\$6)
689	lw	t1, 4(\$4)	749	lw	a3, 4(\$4)
690	lw	t3, 4(\$5)	750	lw	s10, 4(\$5)
691			751		
692	sw	t4, 0(\$4)	752	lw	a6, 32(\$3)
693	sw	t6, 0(\$5)	753		
694	sw	a7, 0(\$6)	754	sw	t1, 0(\$4)
695	lw	t4, 4(\$4)	755	sw	t5, 0(\$5)
696	lw	t6, 4(\$5)	756	sw	a6, 0(\$6)
697			757	lw	t1, 4(\$4)
698	sw	t5, 0(\$4)	758	lw	t5, 4(\$5)
699	sw	a0, 0(\$5)	759		
700	sw	a6, 0(\$6)	760	sw	a2, 0(\$4)
701	lw	t5, 4(\$4)	761	sw	s9, 0(\$5)
702	lw	a0, 4(\$5)	762	sw	a6, 0(\$6)
703			763	lw	a2, 4(\$4)
704	sw	a1, 0(\$4)	764	lw	s9, 4(\$5)
705	sw	a3, 0(\$5)	765		
706	sw	a7, 0(\$6)	766	lw	a6, 96(\$3)
707	lw	a1, 4(\$4)	767		
708	lw	a3, 4(\$5)	768	sw	t3, 0(\$4)
709			769	sw	a0, 0(\$5)
710	sw	a2, 0(\$4)	770	sw	a6, 0(\$6)
711	sw	a4, 0(\$5)	771	lw	t3, 4(\$4)
712	sw	a6, 0(\$6)	772	lw	a0, 4(\$5)
713	lw	a2, 4(\$4)	773		
714	lw	a4, 4(\$5)	774	sw	a4, 0(\$4)
715			775	sw	s11, 0(\$5)
716	sw	a5, 0(\$4)	776	sw	a6, 0(\$6)
717	sw	s10, 0(\$5)	777	lw	a4, 4(\$4)
718	sw	a7, 0(\$6)	778	lw	s11, 4(\$5)
719	lw	a5, 4(\$4)	779		
720	lw	s10, 4(\$5)	780	sw	t0, 0(\$4)
721			781	sw	a1, 0(\$5)
722	sw	s9, 0(\$4)	782	sw	a7, 0(\$6)
723	sw	s11, 0(\$5)	783	lw	t0, 4(\$4)
724	sw	a6, 0(\$6)	784	lw	a1, 4(\$5)
725	lw	s9, 4(\$4)	785		
726	lw	s11, 4(\$5)	786	sw	t6, 0(\$4)
			787	sw	s10, 0(\$5)

Effect of Registers Utilizing (1/4)

■ Loop Unrolling

- # of lw/sw Instructions
- Execution time

of lw: 961
of sw: 833



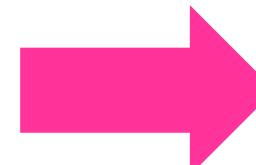
```
Output count has reached the total number of samples.  
Validating bit-accurate output...  
- Total mismatches: 0 / 192  
Checking SRAM occupancy...  
- SRAM 2: occupancy = 64  
Test passed. Ending simulation.  
$finish called from file "../../../../tb/tb\_mem\_IO.v", line 132.  
$finish at simulation time 16566000  
Simulation complete, time is 16566000 ps.
```

Effect of Registers Utilizing (2/4)

▪ Loop Unrolling + Registers in Stage 0,1,2

- # of lw/sw Instructions
- Execution time

of lw: 833
of sw: 702



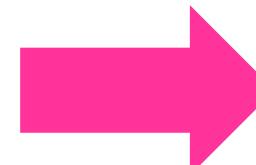
```
Output count has reached the total number of samples.  
Validating bit-accurate output...  
- Total mismatches: 0 / 192  
Checking SRAM occupancy...  
- SRAM 2: occupancy = 64  
Test passed. Ending simulation.  
$finish called from file "../../../../tb/tb\_mem\_I0.v", line 132.  
$finish at simulation time 15012000  
Simulation complete, time is 15012000 ps.
```

Effect of Registers Utilizing (3/4)

▪ Loop Unrolling + Registers in Stage 0,1,2,3

- # of lw/sw Instructions
- Execution time

of lw: 769
of sw: 663



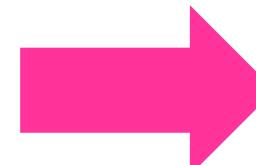
Output count has reached the total number of samples.
Validating bit-accurate output...
- Total mismatches: 0 / 192
Checking SRAM occupancy...
- SRAM 2: occupancy = 64
Test passed. Ending simulation.
\$finish called from file "[../../tb/tb_mem_IO.v](#)", line 132
\$finish at simulation time [14394000](#)
Simulation complete, time is [14394000 ps](#).

Effect of Registers Utilizing (4/4)

- **(3/4) + Reordering & Pipelining in stage 4 : Completed Version**

- # of lw/sw Instructions
- Execution time

of lw: 737
of sw: 629



```
Output count has reached the total number of samples.  
Validating bit-accurate output...  
- Total mismatches: 0 / 192  
Checking SRAM occupancy...  
- SRAM 2: occupancy = 64  
Test passed. Ending simulation.  
$finish called from file "../../../../tb/tb\_mem\_IO.v", line 132.  
$finish at simulation time 14190000  
Simulation complete, time is 14190000 ps.
```