1

In this exercise, we analyze how the instructions of a Vector-Add kernel function run on SIMT pipelines. The following is the instruction sequence in the PTX format:

Using the pipeline timing diagram to show how these instructions are executed on one SM. The following assumptions are made:

- 1. The SM has 16 SPs
- 2. The load/store latency is 5 cycles (i.e., 5 MEM stages) with AGEN included in the 5 stages, and the add latency is 3 cycles (i.e., 3 EX stages)
- 3. The SM can run 6 warps concurrently and the warp scheduling policy is round robin
- 4. There is no scoreboard to support more than 1 instruction from the same warp to be issued to the pipeline.
- 5. IF and ID are 2 cycles each
- 6. WB can handle 16 threads at a time.

How many cycles does it take to execute all the 6 warps, counting from when the first instruction enters the IF stage to when the last instruction enters the WB stage?

There are some additional assumptions that we must consider that the problem statement didn't mention before:

- The pipeline is diversified, meaning that ALU will follow a pipeline of IF, ID, RR, EX[123], WB and load/store will follow a pipeline of IF, ID, RR, MEM[12345], WB
- Also assume that the ALU and MEM stages are fully pipelined.

It takes 58 cycles to execute all all warps. The work is shown below. Attached to this homework, you will find a separate image for better readability and viewing for the work shown.

```
Instr
                                                         3
                                                                              17
    (t0-t15): ld.global.f32 %f1, [%r14+0]
                                                         ID |
                                              IF |
                                                                        MEM1 | MEM2 |
w0 (t16-t32): ld.global.f32
                             %f1, [%r14+0]
                                                                              I MEM1 I
    (t0-t15): ld.global.f32 %f1, [%r14+0] |
                                                  1
                                                       | IF |
                                                                 | ID |
                                                                              I RR
w1 (t16-t32): ld.global.f32 %f1, [%r14+0] |
                                                 - 1
                                                                                     1
```

w2	(±0-±15)·	ld.global.f32	%f1 [%r14+0]	1	1	1	1	IF	ı	ID	ı l
		ld.global.f32	%f1, [%r14+0]	i I	i	i	i			10	'
w3		ld.global.f32	%f1, [%r14+0]	i i	i	i	i	i I	I	   IF	i
		ld.global.f32	%f1, [%r14+0]	i	i	i	i	i I	i I	<del> </del>	i
w4		ld.global.f32	%f1, [%r14+0]	i	i	i	i	i	I	I I	i
		ld.global.f32	%f1, [%r14+0]	i	i	i	i	i I	I	I I	i
w5		ld.global.f32	%f1, [%r14+0]	i	i	i	i	i I	I	' 	i
		•	%f1, [%r14+0]	i	i	i	i	i I	i I	' 	i I
wO		•	%f2, [%r16+0]	i	i	i	i	i I	i I	' 	i I
		ld.global.f32	%f2, [%r16+0]	i	i	i	i	i	i I	I I	i
w1		ld.global.f32	%f2, [%r16+0]	i	i	i	i	i	I	I I	i
		ld.global.f32	%f2, [%r16+0]	i	i I	i	i	i I	İ		i I
w2		ld.global.f32	%f2, [%r16+0]	i	İ	i	i	i I	I		i I
		ld.global.f32	%f2, [%r16+0]	i	i	i	i	i	I	I I	i
w3		ld.global.f32	%f2, [%r16+0]	i	i	i	i	i	I	I I	i
		•	%f2, [%r16+0]	i	i I	i	i	i I	İ		i I
w4		ld.global.f32	%f2, [%r16+0]	İ	İ	i	İ	İ	I		l
w4		ld.global.f32	%f2, [%r16+0]	İ	İ	i	İ	İ	I		l
w5		ld.global.f32	%f2, [%r16+0]	Ī	İ	i	Ì	ĺ	l		ı
w5		ld.global.f32	%f2, [%r16+0]	Ī	İ	i	Ì	ĺ	l		ı
wO	(t0-t15):	add.f32	%f3, %f1, %f2	1	1	1	1	1	1		ı
wO			%f3, %f1, %f2		Ī	ĺ	Ì	ĺ	1		l
w1	(t0-t15):	add.f32	%f3, %f1, %f2		I	1	1	1	1		I
w1	(t16-t32):	add.f32	%f3, %f1, %f2		I	1		1	1		l
w2	(t0-t15):	add.f32	%f3, %f1, %f2			1	1		1		
w2	(t16-t32):	add.f32	%f3, %f1, %f2		1	1	1		1		
wЗ	(t0-t15):	add.f32	%f3, %f1, %f2		1	1	1		1		1
w3	(t16-t32):	add.f32	%f3, %f1, %f2		1	1	1		1		
w4	(t0-t15):	add.f32	%f3, %f1, %f2		1	1	1		1		1
w4	(t16-t32):	add.f32	%f3, %f1, %f2		1	1			1		l
w5	(t0-t15):	add.f32	%f3, %f1, %f2			1	1		1		
w5	(t16-t32):	add.f32	%f3, %f1, %f2			1			1		
ωO	(t0-t15):	st.global.f32	[%r18+0], %f3			1	1		1		
wO	(t16-t32):	st.global.f32	[%r18+0], %f3			1			1		
w1	(t0-t15):	st.global.f32	[%r18+0], %f3			1	1		1		
w1	(t16-t32):	st.global.f32	[%r18+0], %f3		1		1		1	<b> </b>	l
w2	(t0-t15):	st.global.f32	[%r18+0], %f3		1			1	1		l
w2	(t16-t32):	st.global.f32	[%r18+0], %f3		1				1		
w3	(t0-t15):	st.global.f32	[%r18+0], %f3		1				1		
wЗ	(t16-t32):	st.global.f32	[%r18+0], %f3		1				I		l
w4	(t0-t15):	st.global.f32	[%r18+0], %f3	1	1				1		I

ECE 786 Spring 2023	Advanced C Huiyang Zho	1	rchite	cture:	Data	a Pa	ralle	l Proc		SHW 2
w5 (t0-t15):	st.global.f32 st.global.f32 st.global.f32	[%r18+0], %	5f3		1	l	1	1	1	1 1

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

Instr	8	9	10	11	12	13
w0 (t0-t15): ld.global.f32	%f1, [%r14+0]   MEM3	MEM4	MEM5	WB	1	1
w0 (t16-t32): ld.global.f32	%f1, [%r14+0]   MEM2	MEM3	MEM4	MEM5	WB	1
w1 (t0-t15): ld.global.f32	%f1, [%r14+0]   MEM1	MEM2	MEM3	MEM4	MEM5	WB
w1 (t16-t32): ld.global.f32	%f1, [%r14+0]   RR	MEM1	MEM2	MEM3	MEM4	MEM5
w2 (t0-t15): ld.global.f32	%f1, [%r14+0]	RR	MEM1	MEM2	MEM3	MEM4
w2 (t16-t32): ld.global.f32	%f1, [%r14+0]		RR	MEM1	MEM2	MEM3
w3 (t0-t15): ld.global.f32	%f1, [%r14+0]	ID	- 1	RR	MEM1	MEM2
w3 (t16-t32): ld.global.f32	%f1, [%r14+0]		- 1	- 1	RR	MEM1
w4 (t0-t15): ld.global.f32	%f1, [%r14+0]	IF	- 1	ID		RR
w4 (t16-t32): ld.global.f32	%f1, [%r14+0]		- 1	1		1
w5 (t0-t15): ld.global.f32	%f1, [%r14+0]		- 1	IF		ID
w5 (t16-t32): ld.global.f32	%f1, [%r14+0]		- 1	1		1
w0 (t0-t15): ld.global.f32	%f2, [%r16+0]		- 1	1		IF
w0 (t16-t32): ld.global.f32	%f2, [%r16+0]		- 1	1		1
w1 (t0-t15): ld.global.f32	%f2, [%r16+0]		- 1	1		1
w1 (t16-t32): ld.global.f32	%f2, [%r16+0]		1	1		1
w2 (t0-t15): ld.global.f32	%f2, [%r16+0]		1	1		1
w2 (t16-t32): ld.global.f32	%f2, [%r16+0]		1	1		1
w3 (t0-t15): ld.global.f32	%f2, [%r16+0]		1	1		1
w3 (t16-t32): ld.global.f32	%f2, [%r16+0]		1	1		1
w4 (t0-t15): ld.global.f32	%f2, [%r16+0]		1	1		1
w4 (t16-t32): ld.global.f32	%f2, [%r16+0]		- 1	- 1		1
w5 (t0-t15): ld.global.f32	%f2, [%r16+0]		- 1	- 1		
w5 (t16-t32): ld.global.f32	%f2, [%r16+0]		- 1	- 1		
w0 (t0-t15): add.f32	%f3, %f1, %f2		- 1	- 1		
w0 (t16-t32): add.f32	%f3, %f1, %f2		- 1	1		
w1 (t0-t15): add.f32	%f3, %f1, %f2		- 1	1		
w1 (t16-t32): add.f32	%f3, %f1, %f2		- 1	- 1		
w2 (t0-t15): add.f32	%f3, %f1, %f2		- 1	- 1		
w2 (t16-t32): add.f32	%f3, %f1, %f2		- 1	1		
w3 (t0-t15): add.f32	%f3, %f1, %f2		- 1	1		
w3 (t16-t32): add.f32	%f3, %f1, %f2		- 1	1		
w4 (t0-t15): add.f32	%f3, %f1, %f2	l I	- 1	1		1
w4 (t16-t32): add.f32	%f3, %f1, %f2		1	1		1
w5 (t0-t15): add.f32	%f3, %f1, %f2	I I	- 1	1		1
w5 (t16-t32): add.f32	%f3, %f1, %f2	I I	- 1	1		1
w0 (t0-t15): st.global.f32	[%r18+0], %f3	I I	- 1	1		1
w0 (t16-t32): st.global.f32	[%r18+0], %f3	I I	- 1	1		1
w1 (t0-t15): st.global.f32	[%r18+0], %f3	I I	- 1	- 1	I	1
w1 (t16-t32): st.global.f32	[%r18+0], %f3	I I	- 1	1		1

ECE 786	Advanced Computer Architecture	
Spring 2023	Huiyang Zhou	HW 2

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

Instr	14	15   16	17   18	19
w0 (t0-t15): ld.global.f32	%f1, [%r14+0]	1	1	1 1
w0 (t16-t32): ld.global.f32	%f1, [%r14+0]	I I	1	1 1
w1 (t0-t15): ld.global.f32	%f1, [%r14+0]	1	1	1 1
w1 (t16-t32): ld.global.f32	%f1, [%r14+0]   WB	1	1	1 1
w2 (t0-t15): ld.global.f32	%f1, [%r14+0]   MEM5	WB	1	1 1
w2 (t16-t32): ld.global.f32	%f1, [%r14+0]   MEM4	MEM5   WB	1 1	1 1
w3 (t0-t15): ld.global.f32	%f1, [%r14+0]   MEM3	MEM4   MEM5	WB	1 1
w3 (t16-t32): ld.global.f32	%f1, [%r14+0]   MEM2	MEM3   MEM4	MEM5   WB	1 1
w4 (t0-t15): ld.global.f32	%f1, [%r14+0]   MEM1	MEM2   MEM3	MEM4   MEM5	WB
w4 (t16-t32): ld.global.f32	%f1, [%r14+0]   RR	MEM1   MEM2	MEM3   MEM4	MEM5
w5 (t0-t15): ld.global.f32	%f1, [%r14+0]	RR   MEM1	MEM2   MEM3	MEM4
w5 (t16-t32): ld.global.f32	%f1, [%r14+0]	RR	MEM1   MEM2	MEM3
w0 (t0-t15): ld.global.f32	%f2, [%r16+0]	ID	RR   MEM1	MEM2
w0 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	RR	MEM1
w1 (t0-t15): ld.global.f32	%f2, [%r16+0]	IF	ID	RR
w1 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	1 1	1 1
w2 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	IF	ID
w2 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	1 1	1 1
w3 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	1 1	IF
w3 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	1 1	1 1
w4 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	1 1	1 1
w4 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	1 1	1 1
w5 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	1 1	1 1
w5 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	1 1	1 1
w0 (t0-t15): add.f32	%f3, %f1, %f2	I I		1 1
w0 (t16-t32): add.f32	%f3, %f1, %f2	I I		1 1
w1 (t0-t15): add.f32	%f3, %f1, %f2	I I		1 1
w1 (t16-t32): add.f32	%f3, %f1, %f2	1 1	1 1	1 1
w2 (t0-t15): add.f32	%f3, %f1, %f2	I I		1 1
w2 (t16-t32): add.f32	%f3, %f1, %f2	I I		1 1
w3 (t0-t15): add.f32	%f3, %f1, %f2	I I	1 1	1 1
w3 (t16-t32): add.f32	%f3, %f1, %f2	I I	1 1	1 1
w4 (t0-t15): add.f32	%f3, %f1, %f2	I I	1	1 1
w4 (t16-t32): add.f32	%f3, %f1, %f2	I I	1	1 1
w5 (t0-t15): add.f32	%f3, %f1, %f2	I I	1 1	1 1
w5 (t16-t32): add.f32	%f3, %f1, %f2	I I	1 1	1 1
w0 (t0-t15): st.global.f32	[%r18+0], %f3	I I	1 1	1 1
w0 (t16-t32): st.global.f32	[%r18+0], %f3	I I	1 1	1 1
w1 (t0-t15): st.global.f32	[%r18+0], %f3	I I	1 1	1 1
w1 (t16-t32): st.global.f32	[%r18+0], %f3	l I	1 1	1 1

ECE 786	Advanced Computer Architecture	
Spring 2023	Huiyang Zhou	HW 2

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

w0 (t0-t15): ld.global.f32	Instr	1	20	21	22	23	24	25 I
w1 (t0-t15): ld.global.f32 %f1, [%r14+0]	w0 (t0-t15): ld.global.f32	%f1, [%r14+0]	1					I
w1 (t16-t32): ld.global.f32	w0 (t16-t32): ld.global.f32	%f1, [%r14+0]	1					I
w2 (t0-t15): ld.global.f32         %f1, [%r14+0]	w1 (t0-t15): ld.global.f32	%f1, [%r14+0]	1					I
w2 (t16-t32): ld.global.f32         %f1, [%r14+0]	w1 (t16-t32): ld.global.f32	%f1, [%r14+0]	[					
w3 (t0-t15): ld.global.f32  %f1, [%r14+0]	w2 (t0-t15): ld.global.f32	%f1, [%r14+0]	1					
w3 (t16-t32): ld.global.f32         %f1, [%r14+0]	w2 (t16-t32): ld.global.f32	%f1, [%r14+0]	1					
w4 (t0-t15): ld.global.f32         %f1, [%r14+0]   WB	w3 (t0-t15): ld.global.f32	%f1, [%r14+0]	[					
w4 (t16-t32): ld.global.f32         %f1, [%r14+0]   WB	w3 (t16-t32): ld.global.f32	%f1, [%r14+0]	[					
w5 (t0-t15): ld.global.f32         %f1, [%r14+0]   MEM5   WB	w4 (t0-t15): ld.global.f32	%f1, [%r14+0]	1					I
w5 (t16-t32): ld.global.f32         %f1, [%r14+0]   MEM4   MEM5   WB	w4 (t16-t32): ld.global.f32	%f1, [%r14+0]	WB					I
w0 (t0-t15): ld.global.f32         %f2, [%r16+0]   MEM3   MEM4   MEM5   WB   <td< td=""><td>w5 (t0-t15): ld.global.f32</td><td>%f1, [%r14+0]  </td><td>MEM5</td><td>WB</td><td></td><td></td><td></td><td>   </td></td<>	w5 (t0-t15): ld.global.f32	%f1, [%r14+0]	MEM5	WB				
w0 (t16-t32): ld.global.f32         %f2, [%r16+0]   MEM2   MEM3   MEM4   MEM5   WB	w5 (t16-t32): ld.global.f32	%f1, [%r14+0]	MEM4	MEM5	WB			I
w1 (t0-t15): ld.global.f32         %f2, [%r16+0]   MEM1   MEM2   MEM3   MEM4   MEM5   WB             w1 (t16-t32): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4   MEM5             w2 (t0-t15): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4             w2 (t16-t32): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4             w3 (t0-t15): ld.global.f32         %f2, [%r16+0]   ID   RR   MEM1   MEM2   MEM3             w3 (t16-t32): ld.global.f32         %f2, [%r16+0]   ID   RR   MEM1   MEM2             w4 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   ID   RR   MEM1             w4 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   ID   RR   MEM1             w5 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   ID   IF   ID             w5 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   IF   ID             w6 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   IF   ID   IF             w0 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   ID             w1 (t16-t32): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF             w1 (t16-t32): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF             w2 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF   IF   IF             w3 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF   IF   IF   I	w0 (t0-t15): ld.global.f32	%f2, [%r16+0]	MEM3	MEM4	MEM5	WB		1
w1 (t16-t32): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4   MEM5             w2 (t0-t15): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4             w2 (t16-t32): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4             w3 (t0-t15): ld.global.f32         %f2, [%r16+0]   ID   RR   MEM1   MEM2   MEM3             w3 (t16-t32): ld.global.f32         %f2, [%r16+0]   ID   RR   MEM1   MEM2             w4 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   ID   RR   MEM1             w4 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   ID   RR             w5 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   IF   ID             w5 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   IF   ID             w5 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   IF   IF   ID             w0 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   ID             w0 (t16-t32): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF   IF             w1 (t16-t32): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF   IF   IF   I	w0 (t16-t32): ld.global.f32	%f2, [%r16+0]	MEM2	MEM3	MEM4	MEM5	WB	1
w2 (t0-t15): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4   w2 (t16-t32): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4   w3 (t0-t15): ld.global.f32         %f2, [%r16+0]   RR   MEM1   MEM2   MEM3   MEM4   w3 (t16-t32): ld.global.f32         %f2, [%r16+0]   ID   RR   MEM1   MEM2   MEM3   WEM4   w4 (t0-t15): ld.global.f32         %f2, [%r16+0]   ID   RR   MEM1   MEM2   MEM1   MEM2   WEM1   W4 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   ID   RR   MEM1   MEM2   WEM1   W5 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   ID   RR   MEM1   MEM2   MEM1   W5 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   ID   IF   ID   IF   ID   W5 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   IF   ID   IF   ID   IF   ID   IF   ID   W6 (t16-t32): ld.global.f32         %f3, %f1, %f2   IF   IF   ID   IF   IF   ID   IF   IF	w1 (t0-t15): ld.global.f32	%f2, [%r16+0]	MEM1	MEM2	MEM3	MEM4	MEM5	WB
w2 (t16-t32): ld.global.f32       %f2, [%r16+0]	w1 (t16-t32): ld.global.f32	%f2, [%r16+0]	RR	MEM1	MEM2	MEM3	MEM4	MEM5
w3 (t0-t15): ld.global.f32         %f2, [%r16+0]   ID   RR   MEM1   MEM2             w3 (t16-t32): ld.global.f32         %f2, [%r16+0]   ID   RR   MEM1   MEM2             w4 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   ID   RR   MEM1             w4 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   ID   IF   ID             w5 (t0-t15): ld.global.f32         %f2, [%r16+0]   IF   IF   ID             w5 (t16-t32): ld.global.f32         %f2, [%r16+0]   IF   IF   ID             w6 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF             w0 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF             w1 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF             w1 (t16-t32): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF             w2 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF             w3 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF   IF             w3 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF   IF   IF             w4 (t0-t15): add.f32         %f3, %f1, %f2   IF   IF   IF   IF   IF   IF   IF   I	w2 (t0-t15): ld.global.f32	%f2, [%r16+0]		RR	MEM1	MEM2	MEM3	MEM4
w3 (t16-t32): ld.global.f32       %f2, [%r16+0]	w2 (t16-t32): ld.global.f32	%f2, [%r16+0]			RR	MEM1	MEM2	MEM3
w4 (t0-t15): ld.global.f32       %f2, [%r16+0]   IF   ID   RR           w4 (t16-t32): ld.global.f32       %f2, [%r16+0]   I   IF   ID   IF   ID           w5 (t0-t15): ld.global.f32       %f2, [%r16+0]   I   IF   ID           w5 (t16-t32): ld.global.f32       %f2, [%r16+0]   I   IF   ID           w0 (t0-t15): add.f32       %f3, %f1, %f2   I   I   IF           w0 (t16-t32): add.f32       %f3, %f1, %f2   I   I   I   IF           w1 (t0-t15): add.f32       %f3, %f1, %f2   I   I   I   I   I           w1 (t16-t32): add.f32       %f3, %f1, %f2   I   I   I   I   I           w2 (t0-t15): add.f32       %f3, %f1, %f2   I   I   I   I   I           w2 (t16-t32): add.f32       %f3, %f1, %f2   I   I   I   I   I           w3 (t0-t15): add.f32       %f3, %f1, %f2   I   I   I   I   I           w4 (t0-t15): add.f32       %f3, %f1, %f2   I   I   I   I   I           w4 (t16-t32): add.f32       %f3, %f1, %f2   I   I   I   I   I           w5 (t0-t15): add.f32       %f3, %f1, %f2   I   I   I   I   I           w5 (t16-t32): add.f32       %f3, %f1, %f2   I   I   I   I   I           w6 (t0-t15): st.global.f32       [%r18+0], %f3   I   I   I   I   I   I           w1 (t0-t15): st.global.f32       [%r18+0], %f3   I   I   I   I   I   I   I           w1 (t0-t15): st.global.f32       [%r18+0], %f3   I   I   I   I   I   I   I   I   I	w3 (t0-t15): ld.global.f32	%f2, [%r16+0]		ID		RR	MEM1	MEM2
w4 (t16-t32): ld.global.f32       %f2, [%r16+0]	w3 (t16-t32): ld.global.f32	%f2, [%r16+0]					RR	MEM1
w5 (t0-t15): ld.global.f32       %f2, [%r16+0]	_	%f2, [%r16+0]		IF		ID		RR
w5 (t16-t32): ld.global.f32       %f2, [%r16+0]	_	%f2, [%r16+0]	I					I
w0 (t0-t15): add.f32       %f3, %f1, %f2	w5 (t0-t15): ld.global.f32	•	1			IF		ID
w0 (t16-t32): add.f32       %f3, %f1, %f2	w5 (t16-t32): ld.global.f32		1					I
w1 (t0-t15): add.f32       %f3, %f1, %f2			1					IF
w1 (t16-t32): add.f32       %f3, %f1, %f2								I
w2 (t0-t15): add.f32       %f3, %f1, %f2		%f3, %f1, %f2						I
w2 (t16-t32): add.f32       %f3, %f1, %f2			1					I
w3 (t0-t15): add.f32       %f3, %f1, %f2								I
w3 (t16-t32): add.f32       %f3, %f1, %f2								I
w4 (t0-t15): add.f32       %f3, %f1, %f2                               w4 (t16-t32): add.f32       %f3, %f1, %f2                               w5 (t0-t15): add.f32       %f3, %f1, %f2                               w5 (t16-t32): add.f32       %f3, %f1, %f2                               w0 (t0-t15): st.global.f32       [%r18+0], %f3                               w0 (t16-t32): st.global.f32       [%r18+0], %f3                               w1 (t0-t15): st.global.f32       [%r18+0], %f3								
w4 (t16-t32): add.f32       %f3, %f1, %f2                             w5 (t0-t15): add.f32       %f3, %f1, %f2                               w5 (t16-t32): add.f32       %f3, %f1, %f2                               w0 (t0-t15): st.global.f32       [%r18+0], %f3                             w0 (t16-t32): st.global.f32       [%r18+0], %f3                             w1 (t0-t15): st.global.f32       [%r18+0], %f3	(							
w5 (t0-t15): add.f32       %f3, %f1, %f2                             w5 (t16-t32): add.f32       %f3, %f1, %f2                             w0 (t0-t15): st.global.f32 [%r18+0], %f3                           w0 (t16-t32): st.global.f32 [%r18+0], %f3                           w1 (t0-t15): st.global.f32 [%r18+0], %f3			I					I
w5 (t16-t32): add.f32       %f3, %f1, %f2                             w0 (t0-t15): st.global.f32 [%r18+0], %f3                           w0 (t16-t32): st.global.f32 [%r18+0], %f3                           w1 (t0-t15): st.global.f32 [%r18+0], %f3			I					I
w0 (t0-t15): st.global.f32 [%r18+0], %f3			1					I
w0 (t16-t32): st.global.f32 [%r18+0], %f3			l					I
w1 (t0-t15): st.global.f32 [%r18+0], %f3	_		l					I
<u> </u>	_		I					I
w1 (t16-t32): st.global.f32 [%r18+0], %f3	<u> </u>		l	  -				l
	w1 (t16-t32): st.global.f32	[%r18+0], %f3	I					I

ECE 786	Advanced Computer Architecture	
Spring 2023	Huiyang Zhou	HW 2

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

Instr	2	26   27	28	29	30	31
w0 (t0-t15): ld.global.f32	%f1, [%r14+0]	I	I			
w0 (t16-t32): ld.global.f32	%f1, [%r14+0]	1	I			
w1 (t0-t15): ld.global.f32	%f1, [%r14+0]	1	I			
w1 (t16-t32): ld.global.f32	%f1, [%r14+0]	1	I			
w2 (t0-t15): ld.global.f32	%f1, [%r14+0]	1	I			
w2 (t16-t32): ld.global.f32	%f1, [%r14+0]	1	I			
w3 (t0-t15): ld.global.f32	%f1, [%r14+0]	1	I			
w3 (t16-t32): ld.global.f32	%f1, [%r14+0]	1	I			
w4 (t0-t15): ld.global.f32	%f1, [%r14+0]	I	I			
w4 (t16-t32): ld.global.f32	%f1, [%r14+0]	1	1			
w5 (t0-t15): ld.global.f32	%f1, [%r14+0]	1	1			
w5 (t16-t32): ld.global.f32	%f1, [%r14+0]	1	1			
w0 (t0-t15): ld.global.f32	%f2, [%r16+0]	1	1			
w0 (t16-t32): ld.global.f32	%f2, [%r16+0]	1	I			
w1 (t0-t15): ld.global.f32	%f2, [%r16+0]	1	I			
w1 (t16-t32): ld.global.f32	%f2, [%r16+0]   W	IB	I			
w2 (t0-t15): ld.global.f32	%f2, [%r16+0]   M	MEM5   WB	I			
w2 (t16-t32): ld.global.f32	%f2, [%r16+0]   M	MEM4   MEM5	WB			
w3 (t0-t15): ld.global.f32	%f2, [%r16+0]   M	MEM3   MEM4	MEM5	WB		
w3 (t16-t32): ld.global.f32	%f2, [%r16+0]   M	MEM2   MEM3	MEM4	MEM5	WB	
w4 (t0-t15): ld.global.f32	%f2, [%r16+0]   M	MEM1   MEM2	MEM3	MEM4	MEM5	WB
w4 (t16-t32): ld.global.f32	%f2, [%r16+0]   R	RR   MEM1	MEM2	MEM3	MEM4	MEM5
w5 (t0-t15): ld.global.f32	%f2, [%r16+0]	RR	MEM1	MEM2	MEM3	MEM4
w5 (t16-t32): ld.global.f32	%f2, [%r16+0]	I	RR	MEM1	MEM2	MEM3
w0 (t0-t15): add.f32	%f3, %f1, %f2	ID	I	RR	EX1	EX2
w0 (t16-t32): add.f32	%f3, %f1, %f2	1	I		RR	EX1
w1 (t0-t15): add.f32	%f3, %f1, %f2	IF	I	ID		RR
w1 (t16-t32): add.f32	%f3, %f1, %f2	1	I			
w2 (t0-t15): add.f32	%f3, %f1, %f2	I	I	IF		ID
w2 (t16-t32): add.f32	%f3, %f1, %f2	1	1			
w3 (t0-t15): add.f32	%f3, %f1, %f2	1	1			IF
w3 (t16-t32): add.f32	%f3, %f1, %f2	1	1			
w4 (t0-t15): add.f32	%f3, %f1, %f2	1	1			
w4 (t16-t32): add.f32	%f3, %f1, %f2	l .	1			
w5 (t0-t15): add.f32	%f3, %f1, %f2	l .	1			
w5 (t16-t32): add.f32	%f3, %f1, %f2	l .	1			
9	[%r18+0], %f3	l .	1			
w0 (t16-t32): st.global.f32	[%r18+0], %f3	1	1			
w1 (t0-t15): st.global.f32	[%r18+0], %f3	l .	1			
w1 (t16-t32): st.global.f32	[%r18+0], %f3	I	I			

ECE 786	Advanced Computer Architecture	
Spring 2023	Huiyang Zhou	HW 2

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

Instr	1	32	33	34	35	36	37
w0 (t0-t15): ld.global.f32	%f1, [%r14+0]						i
w0 (t16-t32): ld.global.f32	%f1, [%r14+0]						i
w1 (t0-t15): ld.global.f32	%f1, [%r14+0]						i
w1 (t16-t32): ld.global.f32	%f1, [%r14+0]						i
w2 (t0-t15): ld.global.f32	%f1, [%r14+0]						i
w2 (t16-t32): ld.global.f32	%f1, [%r14+0]						i
w3 (t0-t15): ld.global.f32	%f1, [%r14+0]						i
w3 (t16-t32): ld.global.f32	%f1, [%r14+0]						i
w4 (t0-t15): ld.global.f32	%f1, [%r14+0]						i
w4 (t16-t32): ld.global.f32	%f1, [%r14+0]						i
w5 (t0-t15): ld.global.f32	%f1, [%r14+0]						
w5 (t16-t32): ld.global.f32	%f1, [%r14+0]						
w0 (t0-t15): ld.global.f32	%f2, [%r16+0]						i
w0 (t16-t32): ld.global.f32	%f2, [%r16+0]			l		l i	ı il
w1 (t0-t15): ld.global.f32	%f2, [%r16+0]						
w1 (t16-t32): ld.global.f32	%f2, [%r16+0]						
w2 (t0-t15): ld.global.f32	%f2, [%r16+0]						
w2 (t16-t32): ld.global.f32	%f2, [%r16+0]						
w3 (t0-t15): ld.global.f32	%f2, [%r16+0]						
w3 (t16-t32): ld.global.f32	%f2, [%r16+0]						
w4 (t0-t15): ld.global.f32	%f2, [%r16+0]						
w4 (t16-t32): ld.global.f32	%f2, [%r16+0]	WB					1
w5 (t0-t15): ld.global.f32	%f2, [%r16+0]	MEM5	WB				1
w5 (t16-t32): ld.global.f32	%f2, [%r16+0]	MEM4	MEM5	WB			1
w0 (t0-t15): add.f32	%f3, %f1, %f2	EX3			WB		1
w0 (t16-t32): add.f32	%f3, %f1, %f2	EX2	EX3			WB	1
w1 (t0-t15): add.f32	%f3, %f1, %f2	EX1	EX2	EX3			WB
w1 (t16-t32): add.f32	%f3, %f1, %f2	RR	EX1	EX2	EX3		1
w2 (t0-t15): add.f32	%f3, %f1, %f2		RR	EX1	EX2	EX3	1
w2 (t16-t32): add.f32	%f3, %f1, %f2			RR	EX1	EX2	EX3
w3 (t0-t15): add.f32	%f3, %f1, %f2		ID		RR	EX1	EX2
w3 (t16-t32): add.f32	%f3, %f1, %f2		l			RR	EX1
w4 (t0-t15): add.f32	%f3, %f1, %f2		IF		ID		RR
w4 (t16-t32): add.f32	%f3, %f1, %f2						
w5 (t0-t15): add.f32	%f3, %f1, %f2		l		IF		ID
w5 (t16-t32): add.f32	%f3, %f1, %f2						
w0 (t0-t15): st.global.f32	[%r18+0], %f3						IF
w0 (t16-t32): st.global.f32	[%r18+0], %f3						1
w1 (t0-t15): st.global.f32	[%r18+0], %f3						
w1 (t16-t32): st.global.f32	[%r18+0], %f3						

ECE 786	Advanced Computer Architecture	
Spring 2023	Huiyang Zhou	HW 2

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

Instr	38	39	40	41	42	43
w0 (t0-t15): ld.global.f32	%f1, [%r14+0]			1		
w0 (t16-t32): ld.global.f32	%f1, [%r14+0]			1		1
w1 (t0-t15): ld.global.f32	%f1, [%r14+0]			1		
w1 (t16-t32): ld.global.f32	%f1, [%r14+0]			1		
w2 (t0-t15): ld.global.f32	%f1, [%r14+0]			1	1	
w2 (t16-t32): ld.global.f32	%f1, [%r14+0]			1	1	
w3 (t0-t15): ld.global.f32	%f1, [%r14+0]			1		
w3 (t16-t32): ld.global.f32	%f1, [%r14+0]			1		
w4 (t0-t15): ld.global.f32	%f1, [%r14+0]					
w4 (t16-t32): ld.global.f32	%f1, [%r14+0]	1	1	1		1 1
w5 (t0-t15): ld.global.f32	%f1, [%r14+0]	1	1	1		1 1
w5 (t16-t32): ld.global.f32	%f1, [%r14+0]					
w0 (t0-t15): ld.global.f32	%f2, [%r16+0]	1	1	1		1 1
w0 (t16-t32): ld.global.f32	%f2, [%r16+0]	1			1	1 1
w1 (t0-t15): ld.global.f32	%f2, [%r16+0]			1		
w1 (t16-t32): ld.global.f32	%f2, [%r16+0]			1		
w2 (t0-t15): ld.global.f32	%f2, [%r16+0]			1		
w2 (t16-t32): ld.global.f32	%f2, [%r16+0]		1	1	1	l I
w3 (t0-t15): ld.global.f32	%f2, [%r16+0]		1	1	1	l I
w3 (t16-t32): ld.global.f32	%f2, [%r16+0]		1	1	1	l I
w4 (t0-t15): ld.global.f32	%f2, [%r16+0]			1		
w4 (t16-t32): ld.global.f32	%f2, [%r16+0]			1		
w5 (t0-t15): ld.global.f32	%f2, [%r16+0]			1		
w5 (t16-t32): ld.global.f32	%f2, [%r16+0]			1		1 1
w0 (t0-t15): add.f32	%f3, %f1, %f2			1		1 1
w0 (t16-t32): add.f32	%f3, %f1, %f2			1		1 1
w1 (t0-t15): add.f32	%f3, %f1, %f2	l		1		
w1 (t16-t32): add.f32	%f3, %f1, %f2   WB			1		1 1
w2 (t0-t15): add.f32	%f3, %f1, %f2	WB		1		
w2 (t16-t32): add.f32	%f3, %f1, %f2	l	WB	1		
w3 (t0-t15): add.f32	%f3, %f1, %f2   EX3	3		WB	1	1 1
w3 (t16-t32): add.f32	%f3, %f1, %f2   EX2		1	1	WB	1 1
w4 (t0-t15): add.f32	%f3, %f1, %f2   EX		EX3	1	I	WB
w4 (t16-t32): add.f32	%f3, %f1, %f2   RR	EX1	EX2	EX3	I	I I
w5 (t0-t15): add.f32	%f3, %f1, %f2	RR	EX1	EX2	EX3	I I
w5 (t16-t32): add.f32	%f3, %f1, %f2	I	RR	EX1	EX2	EX3
w0 (t0-t15): st.global.f32	[%r18+0], %f3	ID		RR	MEM1	MEM2
w0 (t16-t32): st.global.f32	[%r18+0], %f3	I			RR	MEM1
w1 (t0-t15): st.global.f32	[%r18+0], %f3	IF		ID	I	RR
w1 (t16-t32): st.global.f32	[%r18+0], %f3	I			I	I I

ECE 786	Advanced Computer Architecture	
Spring 2023	Huiyang Zhou	HW 2

w2 (t0-	-t15): st.g	lobal.f32	[%r18+0],	%f3			IF		ID
w2 (t16	-t32): st.g	lobal.f32	[%r18+0],	%f3			I		
w3 (t0-	-t15): st.g	lobal.f32	[%r18+0],	%f3			I		IF
w3 (t16	-t32): st.g	lobal.f32	[%r18+0],	%f3		1	1		1
w4 (t0	-t15): st.g	lobal.f32	[%r18+0],	%f3			I		1
w4 (t16	-t32): st.g	lobal.f32	[%r18+0],	%f3			1		1
w5 (t0-	-t15): st.g	lobal.f32	[%r18+0],	%f3			1		1
w5 (t16	-t32): st.g	lobal.f32	[%r18+0],	%f3			1		1
	_								

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

Instr	44	45	46	47	48	49
w0 (t0-t15): ld.global.f32	%f1, [%r14+0]	1			l	1
w0 (t16-t32): ld.global.f32	%f1, [%r14+0]	1			l	1
w1 (t0-t15): ld.global.f32	%f1, [%r14+0]	1			l	1
w1 (t16-t32): ld.global.f32	%f1, [%r14+0]	1			l	1
w2 (t0-t15): ld.global.f32	%f1, [%r14+0]	1			I	1
	%f1, [%r14+0]	1			I	1
w3 (t0-t15): ld.global.f32	%f1, [%r14+0]	1			l	1
w3 (t16-t32): ld.global.f32	%f1, [%r14+0]	1			l	1
w4 (t0-t15): ld.global.f32	%f1, [%r14+0]	1			l	1
w4 (t16-t32): ld.global.f32	%f1, [%r14+0]	1			l	
w5 (t0-t15): ld.global.f32	%f1, [%r14+0]	1			l	1
w5 (t16-t32): ld.global.f32	%f1, [%r14+0]	1			l	1
=	%f2, [%r16+0]	1			l	
w0 (t16-t32): ld.global.f32	%f2, [%r16+0]	1			l	1
w1 (t0-t15): ld.global.f32	%f2, [%r16+0]	1			l	
w1 (t16-t32): ld.global.f32	%f2, [%r16+0]	1			l	1
w2 (t0-t15): ld.global.f32	%f2, [%r16+0]	1			l	
w2 (t16-t32): ld.global.f32	%f2, [%r16+0]	1			l	
w3 (t0-t15): ld.global.f32	%f2, [%r16+0]	1			l	
w3 (t16-t32): ld.global.f32	%f2, [%r16+0]	1			l	
w4 (t0-t15): ld.global.f32	%f2, [%r16+0]	1			l	
w4 (t16-t32): ld.global.f32	%f2, [%r16+0]	1			l	
w5 (t0-t15): ld.global.f32	%f2, [%r16+0]	1			l	1
w5 (t16-t32): ld.global.f32	%f2, [%r16+0]	1			l	1
w0 (t0-t15): add.f32	%f3, %f1, %f2	1			l	
w0 (t16-t32): add.f32	%f3, %f1, %f2	1			l	
w1 (t0-t15): add.f32	%f3, %f1, %f2	1				1
w1 (t16-t32): add.f32	%f3, %f1, %f2	1			l	
w2 (t0-t15): add.f32	%f3, %f1, %f2	1				
w2 (t16-t32): add.f32	%f3, %f1, %f2	1				
w3 (t0-t15): add.f32	%f3, %f1, %f2	1			l	
w3 (t16-t32): add.f32	%f3, %f1, %f2	I			l	
w4 (t0-t15): add.f32	%f3, %f1, %f2	1			l	1
w4 (t16-t32): add.f32	%f3, %f1, %f2   WB	1			l	1
w5 (t0-t15): add.f32	%f3, %f1, %f2	WB			l	I I
w5 (t16-t32): add.f32	%f3, %f1, %f2	1	WB		l	I I
w0 (t0-t15): st.global.f32	[%r18+0], %f3   MEM3	MEM4	MEM5	WB	l	1
w0 (t16-t32): st.global.f32	[%r18+0], %f3   MEM2	MEM3	MEM4	MEM5	WB	1
w1 (t0-t15): st.global.f32	[%r18+0], %f3   MEM1		MEM3	MEM4	MEM5	WB
w1 (t16-t32): st.global.f32	[%r18+0], %f3   RR	MEM1	MEM2	MEM3	MEM4	MEM5

w2	(t0-t15):	st.global.f32	[%r18+0], %	%f3		RR		MEM1	١	MEM2		MEM3		MEM4	ı
w2	(t16-t32):	st.global.f32	[%r18+0], %	√f3				RR		MEM1		MEM2		MEM3	I
wЗ	(t0-t15):	st.global.f32	[%r18+0], %	%f3		ID				RR		MEM1		MEM2	I
wЗ	(t16-t32):	st.global.f32	[%r18+0], %	%f3								RR		MEM1	١
w4	(t0-t15):	st.global.f32	[%r18+0], %	%f3	1	IF				ID				RR	I
w4	(t16-t32):	st.global.f32	[%r18+0], %	%f3											١
w5	(t0-t15):	st.global.f32	[%r18+0], %	%f3						IF				ID	١
w5	(t16-t32):	st.global.f32	[%r18+0], %	%f3											١

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

Instr	50	51	52	53	54   55	ıl
	%f1, [%r14+0]		 	i	i	il
w0 (t16-t32): ld.global.f32	%f1, [%r14+0]	i	I I	i	i	i
w1 (t0-t15): ld.global.f32	%f1, [%r14+0]	i		i	i	i
w1 (t16-t32): ld.global.f32	%f1, [%r14+0]	i		i	i	i
w2 (t0-t15): ld.global.f32	%f1, [%r14+0]	i		i	i	i
w2 (t16-t32): ld.global.f32	%f1, [%r14+0]	i		i	i	i
w3 (t0-t15): ld.global.f32		İ		İ	i	i
w3 (t16-t32): ld.global.f32	%f1, [%r14+0]			- 1	I	1
w4 (t0-t15): ld.global.f32	%f1, [%r14+0]	1		1	I	1
w4 (t16-t32): ld.global.f32	%f1, [%r14+0]	1		- 1	1	1
w5 (t0-t15): ld.global.f32	%f1, [%r14+0]			1	1	1
w5 (t16-t32): ld.global.f32	%f1, [%r14+0]			1	1	1
w0 (t0-t15): ld.global.f32	%f2, [%r16+0]			- 1	I	- 1
w0 (t16-t32): ld.global.f32	%f2, [%r16+0]			- 1	I	- 1
w1 (t0-t15): ld.global.f32	%f2, [%r16+0]			- 1	I	- 1
w1 (t16-t32): ld.global.f32	%f2, [%r16+0]			- 1	I	1
w2 (t0-t15): ld.global.f32	%f2, [%r16+0]				I	- 1
w2 (t16-t32): ld.global.f32	%f2, [%r16+0]				I	- 1
w3 (t0-t15): ld.global.f32	%f2, [%r16+0]			I	I	1
w3 (t16-t32): ld.global.f32	%f2, [%r16+0]			I	I	1
w4 (t0-t15): ld.global.f32	%f2, [%r16+0]			I	I	1
w4 (t16-t32): ld.global.f32	%f2, [%r16+0]			I	I	- 1
w5 (t0-t15): ld.global.f32	%f2, [%r16+0]			I	I	1
w5 (t16-t32): ld.global.f32	%f2, [%r16+0]				l	- 1
w0 (t0-t15): add.f32	%f3, %f1, %f2				ļ	
w0 (t16-t32): add.f32	%f3, %f1, %f2			l	!	I
w1 (t0-t15): add.f32	%f3, %f1, %f2				l	l l
w1 (t16-t32): add.f32	%f3, %f1, %f2	!		l	!	l l
w2 (t0-t15): add.f32	%f3, %f1, %f2			ļ	!	l l
w2 (t16-t32): add.f32	%f3, %f1, %f2			ļ	!	
w3 (t0-t15): add.f32	%f3, %f1, %f2			ļ	!	
w3 (t16-t32): add.f32	%f3, %f1, %f2	l l		l		
w4 (t0-t15): add.f32	%f3, %f1, %f2			I	!	l I
w4 (t16-t32): add.f32	%f3, %f1, %f2			I	1	
w5 (t0-t15): add.f32	%f3, %f1, %f2			I	1	1
w5 (t16-t32): add.f32	%f3, %f1, %f2	 		l I		1
w0 (t0-t15): st.global.f32	[%r18+0], %f3			l I	l I	
w0 (t16-t32): st.global.f32 w1 (t0-t15): st.global.f32	[%r18+0], %f3			l I	 	 
w1 (t0-t15): st.global.f32 w1 (t16-t32): st.global.f32	[%r18+0], %f3   [%r18+0], %f3   WB	 		l I	 	
wi (010 002). St.giobai.132	[/0110.0] , /010   MD	ı	ı I	'	I	ı

```
(t0-t15): st.global.f32
                             [%r18+0], %f3 | MEM5 | WB
w2 (t16-t32): st.global.f32
                             [%r18+0], %f3 | MEM4 | MEM5 | WB
   (t0-t15): st.global.f32
                             [%r18+0], %f3 | MEM3 | MEM4
                                                           MEM5
w3 (t16-t32): st.global.f32
                             [%r18+0], %f3 | MEM2 | MEM3
    (t0-t15): st.global.f32
                             [%r18+0], %f3 | MEM1 | MEM2
                                                         MEM3 MEM4
w4 (t16-t32): st.global.f32
                             [%r18+0], %f3 | RR
                                                  | MEM1 | MEM2 | MEM3
   (t0-t15): st.global.f32
                             [%r18+0], %f3 |
                                                    RR
                                                         | MEM1 |
                                                                  MEM2
                                                                         MEM3 |
w5 (t16-t32): st.global.f32
                             [%r18+0], %f3 |
                                                  | RR
                                                                | MEM1 | MEM2 | MEM3 |
```

ECE 786 Advanced Computer Architecture: Data Parallel Processors
Spring 2023 Huiyang Zhou HW 2

Instr	5	56   57	58
w0 (t0-t15): ld.global.f32	%f1, [%r14+0]	i i	İ
w0 (t16-t32): ld.global.f32	%f1, [%r14+0]	i	i
w1 (t0-t15): ld.global.f32	%f1, [%r14+0]	i i	İ
_	%f1, [%r14+0]	i	Ì
w2 (t0-t15): ld.global.f32	%f1, [%r14+0]	i i	Ì
w2 (t16-t32): ld.global.f32	%f1, [%r14+0]	i i	Ì
G	%f1, [%r14+0]	1 1	I
•	%f1, [%r14+0]	1 1	I
w4 (t0-t15): ld.global.f32	%f1, [%r14+0]	1 1	1
w4 (t16-t32): ld.global.f32	%f1, [%r14+0]	1 1	I
w5 (t0-t15): ld.global.f32	%f1, [%r14+0]	1 1	I
w5 (t16-t32): ld.global.f32	%f1, [%r14+0]	1 1	I
w0 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	I
w0 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	I
w1 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	I
w1 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	I
w2 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	I
w2 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	I
w3 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	I
w3 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	I
w4 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	1
w4 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	1
w5 (t0-t15): ld.global.f32	%f2, [%r16+0]	1 1	1
w5 (t16-t32): ld.global.f32	%f2, [%r16+0]	1 1	1
w0 (t0-t15): add.f32	%f3, %f1, %f2		1
w0 (t16-t32): add.f32	%f3, %f1, %f2		I
w1 (t0-t15): add.f32	%f3, %f1, %f2		I
w1 (t16-t32): add.f32	%f3, %f1, %f2		I
w2 (t0-t15): add.f32	%f3, %f1, %f2		I
w2 (t16-t32): add.f32	%f3, %f1, %f2		I
w3 (t0-t15): add.f32	%f3, %f1, %f2		I
w3 (t16-t32): add.f32	%f3, %f1, %f2		I
w4 (t0-t15): add.f32	%f3, %f1, %f2		I
w4 (t16-t32): add.f32	%f3, %f1, %f2	1 1	I
w5 (t0-t15): add.f32	%f3, %f1, %f2	1 1	I
w5 (t16-t32): add.f32	%f3, %f1, %f2	1 1	I
w0 (t0-t15): st.global.f32	[%r18+0], %f3	1 1	I
w0 (t16-t32): st.global.f32	[%r18+0], %f3	1 1	I
w1 (t0-t15): st.global.f32	[%r18+0], %f3	1 1	I
w1 (t16-t32): st.global.f32	[%r18+0], %f3	1 1	I

```
w2 (t0-t15): st.global.f32
                             [%r18+0], %f3 |
w2 (t16-t32): st.global.f32
                             [%r18+0], %f3 |
   (t0-t15): st.global.f32
                             [%r18+0], %f3 |
                             [%r18+0], %f3 |
w3 (t16-t32): st.global.f32
                             [%r18+0], %f3 |
   (t0-t15): st.global.f32
w4 (t16-t32): st.global.f32
                             [%r18+0], %f3 | WB
   (t0-t15): st.global.f32
                             [%r18+0], %f3 | MEM5 | WB
w5 (t16-t32): st.global.f32
                             [%r18+0], %f3 | MEM4 | MEM5 | WB
```