ENEE 446
Fall 2018
Handout #10

Scoreboarding Ex

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	…	55	56	57
L.D F6, 34(R2)	F	D	I	E	M	W																				
L.D F2, 45(R3)		F	D	I	E	M	W																			
MUL.D F0, F2, F4			F	D	I	-	E	E	E	E	E	E	E	E	E	E	W									
SUB.D F8, F6, F2				F	D	I	-	E	E	W																
DIV.D F10, F0, F6					F	D	I	-	-	-	-	-	-	-	-	-	E	E	E	E	E	E	…	E	E	W
ADD.D F6, F8, F2						F	D	I	-	E	E	-	-	-	-	-	-	W								

MUL.D - 9 cycles; DIV.D - 39 cycles; ADD.D, SUB.D - 1 cycle

<u>cycle 8:</u>

Busy bits:

F0	F2	F4	F6	F8	F10
1	0	0	0	1	1

	OPCODE	R_1	SR_1	R_2	SR_2	DR
ADD0	SUB.D	1	F6	1	F2	F8
ADD1						
MULT	MUL.D	1	F2	1	F4	F0
DIV	DIV.D	0	F0	1	F6	F10

Set busy bits [F6] : 1
allocate ADD1
  OPCODE ← ADD.D
  $R_1 \leftarrow 0$
  $SR_1 \leftarrow F8$
  $R_2 \leftarrow 1$
  $SR_2 \leftarrow F2$
  DR ← F6

<u>cycle 10:</u>

Busy bits:

F0	F2	F4	F6	F8	F10
1	0	0	1	1	1

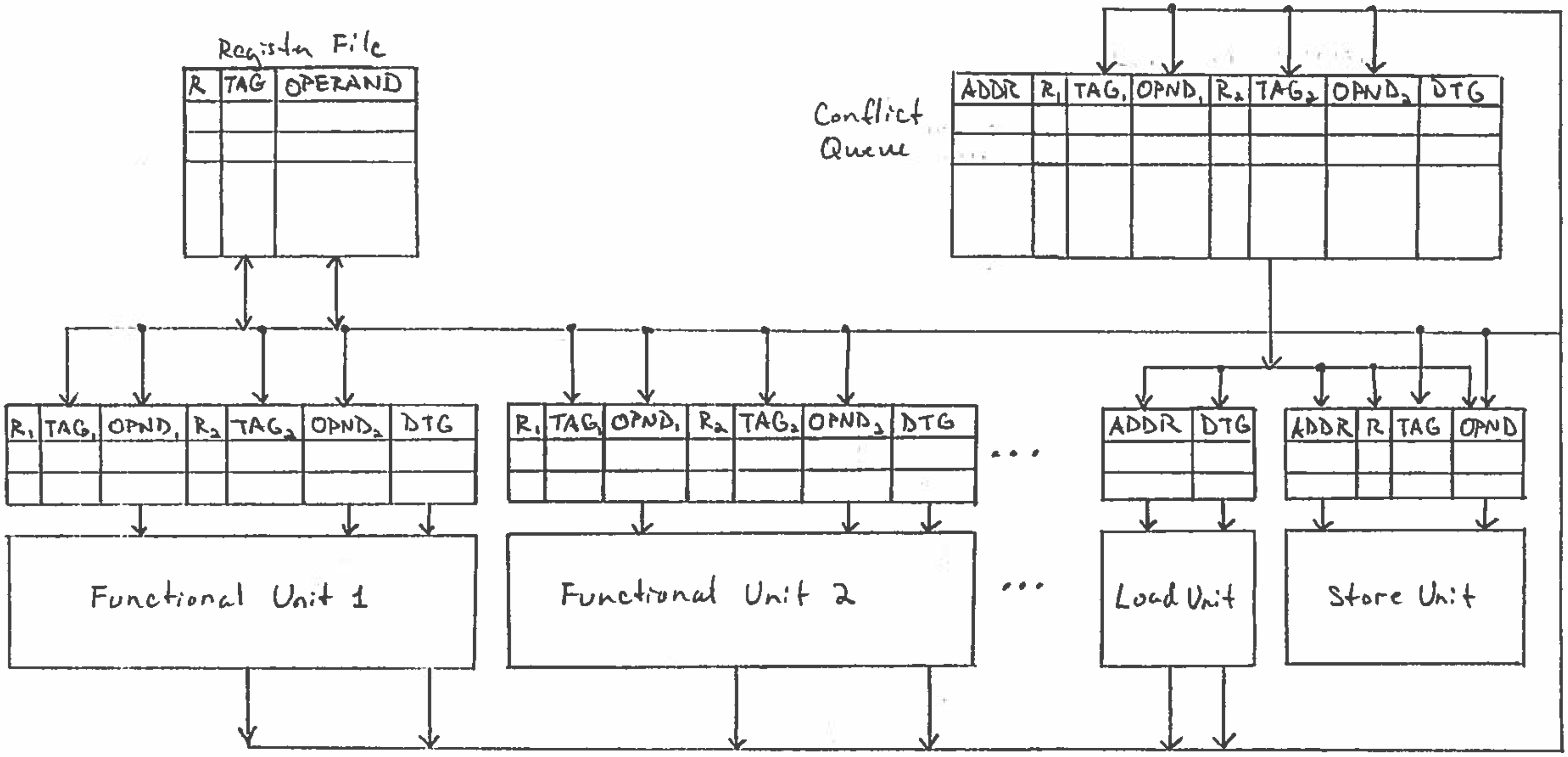
	OPCODE	R_1	SR_1	R_2	SR_2	DR
ADD0	~~SUB.D~~	1	F6	1	F2	F8
ADD1	ADD.D	0	F8	1	F2	F6
MULT	MUL.D	1	F2	1	F4	F0
DIV	DIV.D	0	F0	1	F6	F10

Write Result → REG[F8]
Reset busy bits [F8] = 0
Deallocate ADD0
Check scoreboard
  match waiting source opnd F8 for ADD1
  set ADD1.R_1 = 1

Issue ADD1

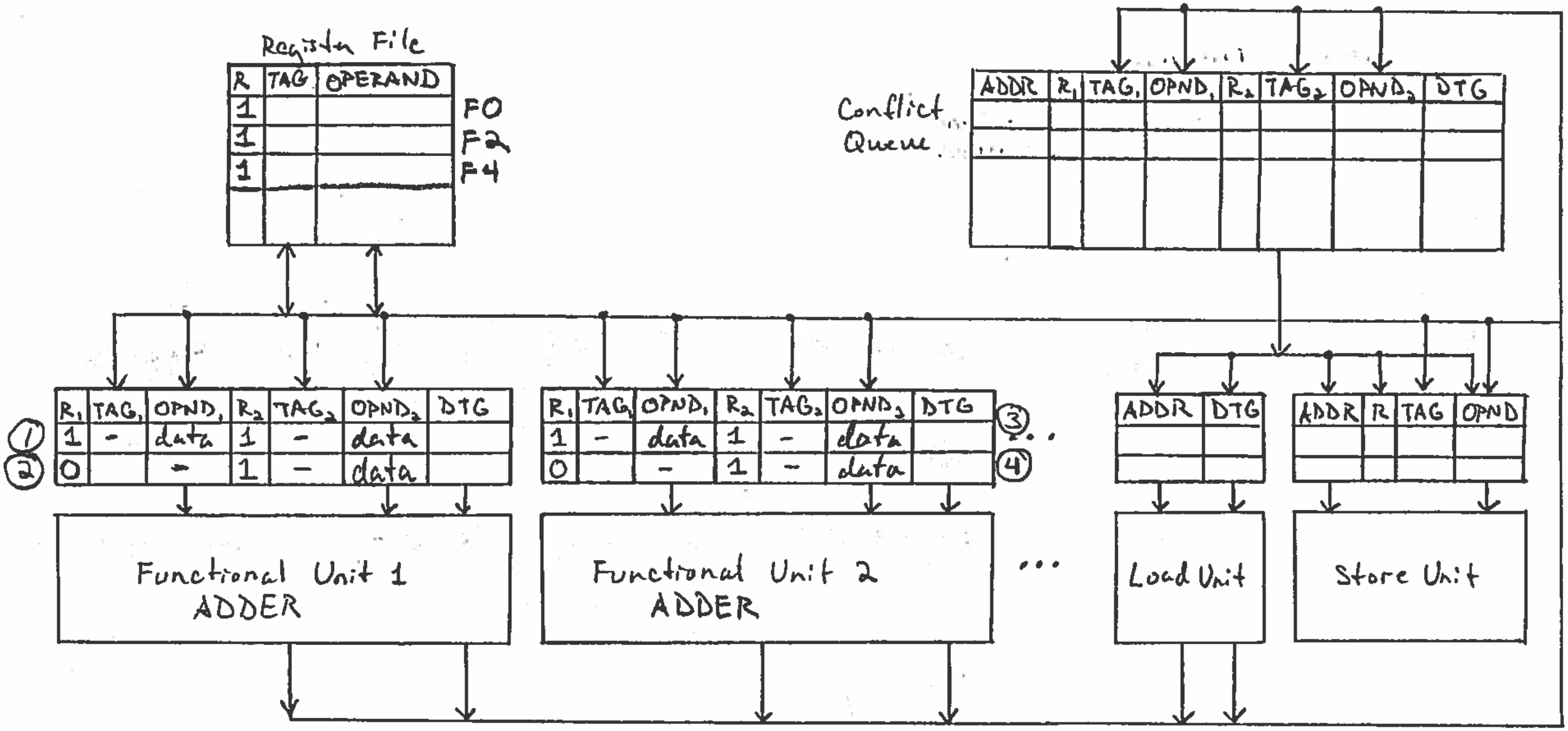
(1)

Dynamic Scheduling Using Reservation Stations
(Tomasulo's Algorithm)



Dynamic Scheduling Using Reservation Stations

(Tomasulo's Algorithm)



Register File

R	TAG	OPERAND	
1			F0
1			F2
1			F4

Conflict Queue

ADDR	R_1	TAG_1	$OPND_1$	R_2	TAG_2	$OPND_2$	DTG
.							
...							

	R_1	TAG_1	$OPND_1$	R_2	TAG_2	$OPND_2$	DTG
①	1	-	data	1	-	data	
②	0		-	1	-	data	

Functional Unit 1 ADDER

R_1	TAG_1	$OPND_1$	R_2	TAG_2	$OPND_2$	DTG	
1	-	data	1	-	data		③
0		-	1	-	data		④

Functional Unit 2 ADDER

...

ADDR	DTG

Load Unit

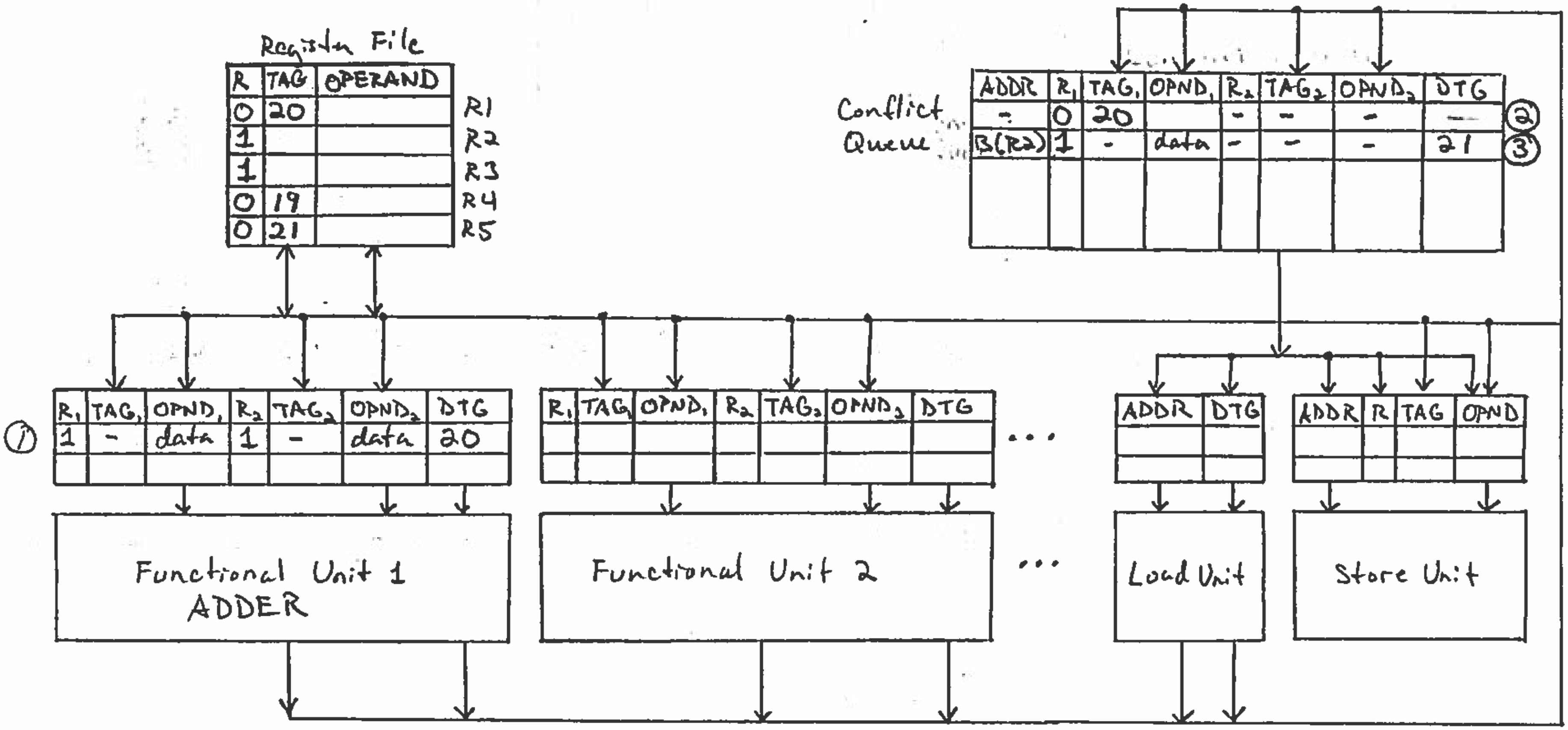
ADDR	R	TAG	OPND

Store Unit

Register Renaming Example

① ADD F0 F4 F4
② ADD F2 F0 F4
③ ADD F0 F4 F4
④ ADD F2 F0 F4

Dynamic Scheduling Using Reservation Stations
(Tomasulo's Algorithm)



Register File

R	TAG	OPERAND	
0	20		R1
1			R2
1			R3
0	19		R4
0	21		R5

Conflict Queue

ADDR	R₁	TAG₁	OPND₁	R₂	TAG₂	OPND₂	DTG	
-	0	20		-	-	-	—	②
B(R2)	1	-	data	-	-	-	21	③

	R₁	TAG₁	OPND₁	R₂	TAG₂	OPND₂	DTG
①	1	-	data	1	-	data	20

Functional Unit 1
ADDER

R₁	TAG₁	OPND₁	R₂	TAG₂	OPND₂	DTG

Functional Unit 2

. . .

ADDR	DTG

Load Unit

ADDR	R	TAG	OPND

Store Unit

Memory Disambiguation Example

① ADD R1 R2 R3
② SW R4 A(R1)
③ LW R5 B(R2)

Tomasulo's Algorithm Example

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
iteration i	1	L.D F0, A(R1)	F	D	I	E	M	W																								
	2	L.D F2, B(R1)		F	D	I	E	M	W																							
	3	MUL.D F4, F0, F2			F	D	I	-	E	E	E	E	E	E	E	E	E	E	W													
	4	ADD.D F6, F4, F8				F	D	I	-	-	-	-	-	-	-	-	-	-	E	E	E	E	W									
	5	S.D F6, D(R1)					F	D	I	(E)	-	-	-	-	-	-	-	-	-	-	-	-	E	M	W							
	6	DADDI R1, R1, #8						F	D	I	E	M	W																			
	7	SLT R3, R1, R2							F	D	I	E	M	W																		
	8	BNEZ R3, LOOP								F	D	I	E	M	W																	
iteration i+1	9	L.D F0, A(R1)									-	-	-	F	D	I	E	M	W													
	10	L.D F2, B(R1)													F	D	I	E	M	W												
	11	MUL.D F4, F0, F2														F	D	I	-	E	E	E	E	E	E	E	E	E	E	W		
	12	ADD.D F6, F4, F8															F	D	I	-	-	-	-	-	-	-	-	-	-	E	E	W
	13	S.D F6, D(R1)																F	D	I	(E)	-	-	-	-	-	-	-	-	-	-	E
	14	DADDI R1, R1, #8																	F	D	I	E	M	W								
	15	SLT R3, R1, R2																		F	D	I	E	M	W							
	16	BNEZ R3, LOOP																			F	D	I	E	M	W						

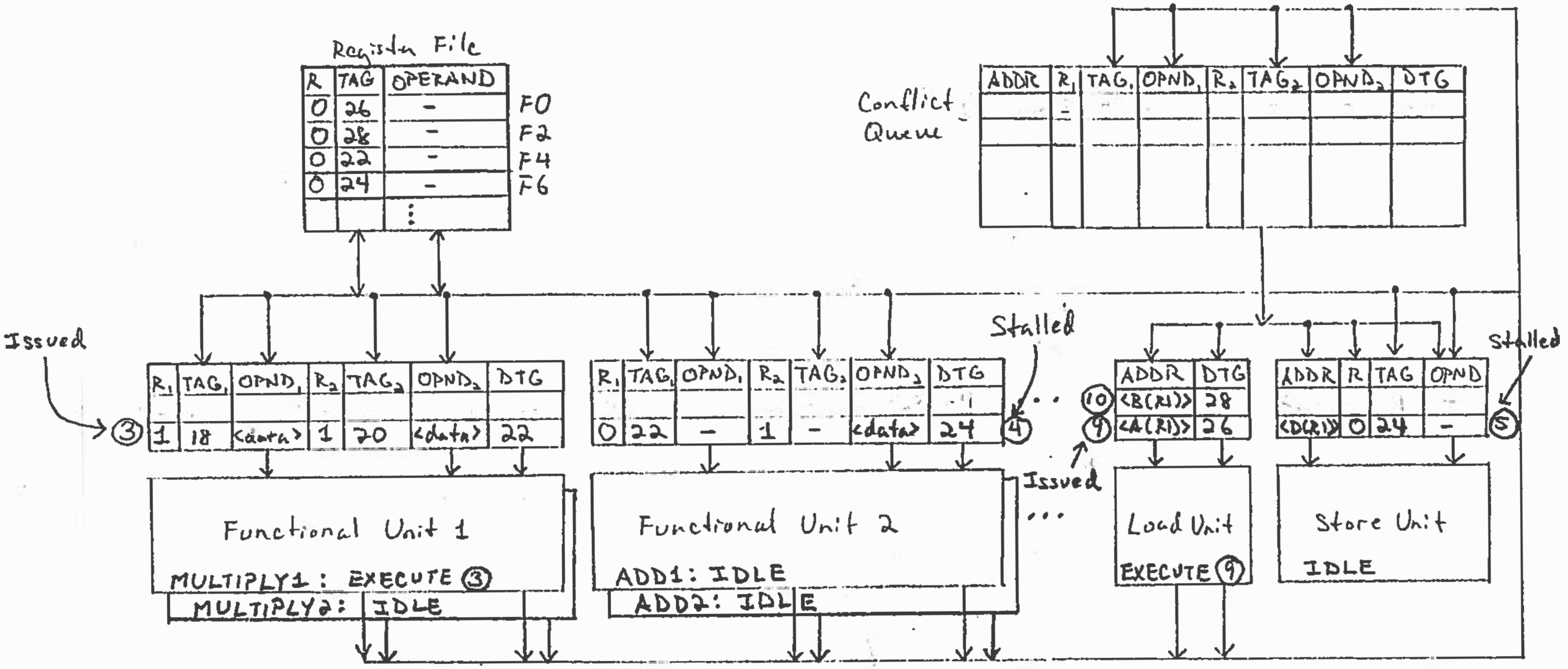
```
for ( i=0; i<n; i++) {
    D[i] = A[i] · B[i] + C;
}
```

i → R1, C → F8, n·8 → R2

MUL.D - 9 cycles
ADD.D - 3 cycles

Dynamic Scheduling Using Reservation Stations

(Tomasulo's Algorithm)



Register File

R	TAG	OPERAND	
0	26	–	F0
0	28	–	F2
0	22	–	F4
0	24	–	F6
		⋮	

Conflict Queue

ADDR	R₁	TAG₁	OPND₁	R₂	TAG₂	OPND₂	DTG

Functional Unit 1

	R₁	TAG₁	OPND₁	R₂	TAG₂	OPND₂	DTG
③	1	18	<data>	1	20	<data>	22

Functional Unit 2

R₁	TAG₁	OPND₁	R₂	TAG₂	OPND₂	DTG	
0	22	–	1	–	<data>	24	④

Load Unit

	ADDR	DTG
⑩	<B(R1)>	28
⑨	<A(R1)>	26

Store Unit

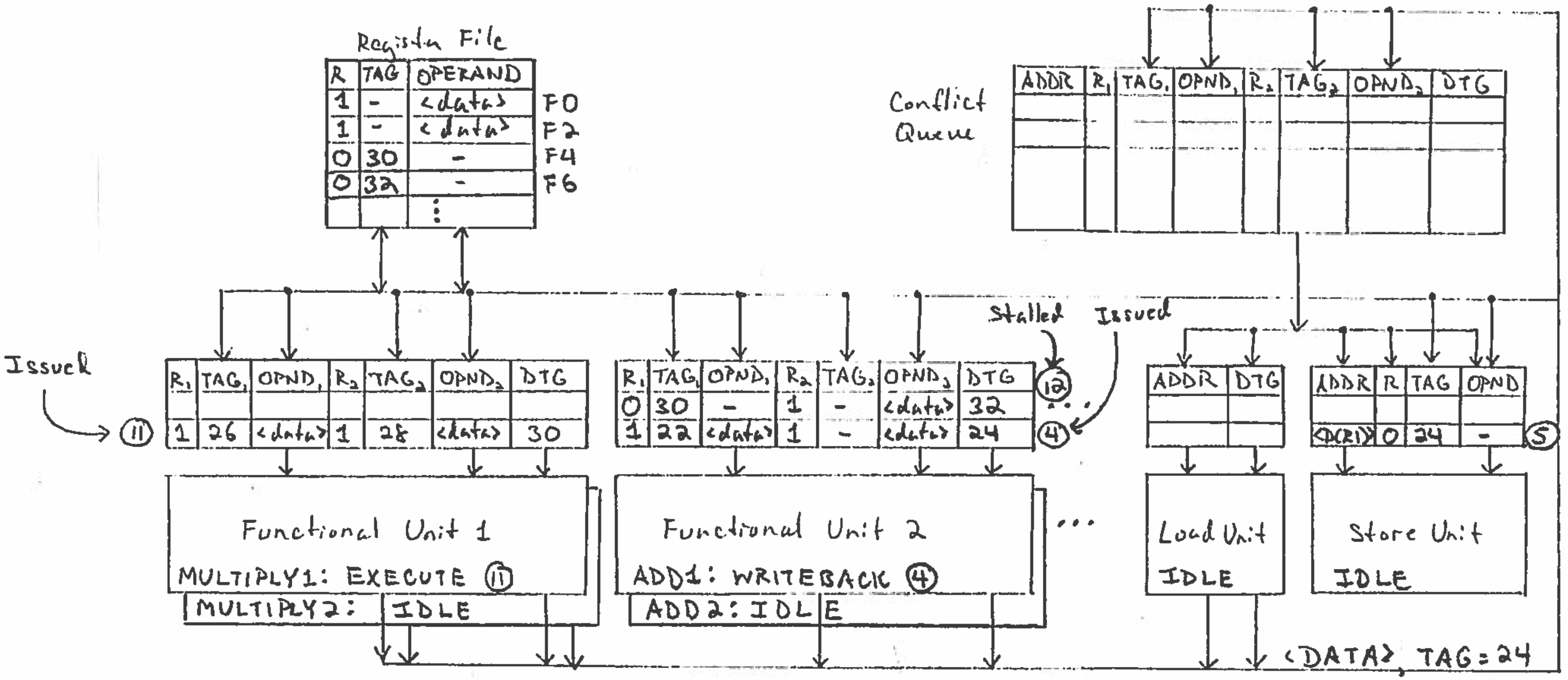
ADDR	R	TAG	OPND	
<D(R1)>	0	24	–	⑤

Cycle 16: Issue MULTD (instruction ⑪)

Rename F0 → 26
Rename F2 → 28
Rename F4 → 30

$R_1 = 0$
$TAG_1 = 26$
$OPND_1 = -$
$R_2 = 0$
$TAG_2 = 28$
$OPND_2 = -$
$DTG = 30$

Dynamic Scheduling Using Reservation Stations
(Tomasulo's Algorithm)



Register File

R	TAG	OPERAND	
1	-	<data>	F0
1	-	<data>	F2
0	30	-	F4
0	32	-	F6
		⋮	

Conflict Queue

ADDR	R_1	TAG_1	$OPND_1$	R_2	TAG_2	$OPND_2$	DTG

Issued → ⑪

R_1	TAG_1	$OPND_1$	R_2	TAG_2	$OPND_2$	DTG
1	26	<data>	1	28	<data>	30

Functional Unit 1

MULTIPLY1: EXECUTE ⑪

MULTIPLY2: IDLE

Stalled → ⑫ Issued → ④

R_1	TAG_1	$OPND_1$	R_2	TAG_2	$OPND_2$	DTG
0	30	-	1	-	<data>	32
1	22	<data>	1	-	<data>	24

Functional Unit 2

ADD1: WRITEBACK ④

ADD2: IDLE

...

ADDR	DTG

Load Unit

IDLE

ADDR	R	TAG	OPND
<DCR1>	0	24	-

⑤

Store Unit

IDLE

<DATA>, TAG=24

Cycle 21: ADDD Completes (Instruction ④)
Assert <DATA>, TAG=24 onto result bus
TAG matches instruction ⑤ in store unit
TAG does not match in register file