1. (28 pts) Show the state of the reservation stations and register file for each of the first 6 clock cycles. You may assume that add/sub and load/store take 1 cycle for execution and mult/div instructions take 15 cycles for execution. For the register file and the Q fields, you may identify instruction that is waited on with the instruction number. (HINT – remember that instructions begin execution on the clock cycle following the cycle that all values are available)

The state column represents the current execution state that the instruction is in. Possible states are: (I) Issue, (W) Waiting for operands, (R) Ready to execute (all operands are available), (E) Executing, and (C) Commit results. Note that an instruction may be in both Issued and Ready at the same time if all operands are available upon instruction issue. That instruction can go directly from Issued to Executing if other constraints are met.

# The register file contains some starting values $-r^2 = 6$ , $r^3 = 8$ , $r^4 = 26$ . Please propagate all values through all subsequent clock cycles.

Inst #			
A1	add	r2, r2, r3	<i># OP result, j, k (same format for all ALU instructions)</i>
A2	load	r1, 0(r2)	# The value at memory location $0+r2$ is 10
A3	mult	r3, r2, r1	<i># multiple r2 and r1 and store result in r3</i>
A4	store	r1, 0(r2)	
A5	sub	r5, r3, r4	

#### Clock cycle 1:

Name	Inst. #	Vj	Vk	Qj	Qk	State
Add/Sub1	A1	6	8			Issue
Add/Sub2						
Add/Sub3						
Mult/Div1						
Mult/Div2						

Name	Inst #	Address	V (for stores)	Q (for stores)	State
Load/Store1					
Load/Store2					
Load/Store3					

r0	r1	r2	r3	r4	r5	r6
		A1	8	26		

### Clock cycle 2:

Name	Inst. #	Vj	Vk	Qj	Qk	State
Add/Sub1	A1	6	8			Execute
Add/Sub2						
Add/Sub3						
Mult/Div1						
Mult/Div2						

Name	Inst #	Address	V (for stores)	Q (for stores)	State
Load/Store1	A2	A1+0			Issue
Load/Store2					
Load/Store3					

r0	r1	r2	r3	r4	r5	r6
	A2	A1	8	26		

## Clock cycle 3:

	A1 A3	6	8			Commit
	A3	14				
	A3	14				1
	A3	14				
		14			A2	Issue
						15540
	I	I		<b>_</b>		
I	Inst #	Address	V (f	for stores)	O (for stores)	State
Т	<u>A2</u>	14	, ( <u>1</u>	01 5(0105)		Ready
	A2	14				Ktauy
+						
r1	rJ	r <sup>2</sup>	r1	r5	rh	
	12	13	26	15	10	
AZ	14	A3	26	<u> </u>		
:						
In	ıst. #	Vj	Vk	Qj	Qk	State
	A3	14			A2	Wait
-				_		
	I					
Ţ	Inst #	Address	V (f	for stores)	Q (for stores)	State
	A2	14		/		Execute
-	A4	14			A2	Issue
						15540
r1	r?	r3	r/	r5	r6	
11	14	13	26	15	10	
AZ	14	AJ	20			
:						
In	ıst. #	Vj	Vk	Qj	Qk	State
	A5		26	A3		Issue
						1
	A3	14	10			Ready
+ '			10			Licuuy
	1			I		1
]	Inst #	Address	V (f	for stores)	O (for stores)	State
	A2	14		/		Commit
	A4	14		10		Readv
+		<u> </u>				Liouuj
			I			
 r1	r2	r3	r4	r5	r6	
		A2    r1  r2    A2  14    Inst. #  A3    Inst #  A2    A3  Inst #    A2  14    Inst. #  A3    Inst. #  A2    A4  Inst. #    A2  14    Inst. #  A2    A3  Inst. #    A4  Inst. #    A3  Inst. #	A2  14    r1  r2  r3    A2  14  A3    r1  r2  r3    A2  14  A3    Inst. #  Vj    Inst. #  Vj    Inst #  Address    A2  14    A3  14    Inst #  Address    A2  14    A3  14    Inst. #  Vj    A3  14    A3  14    A4  14    Inst. #  Vj    A5  Inst. #    Inst #  Address    A3  14    A3  14    A3  14	A2  14    r1  r2  r3  r4    A2  14  A3  26    r1  r2  r3  r4    A2  14  A3  26    Inst. #  Vj  Vk    A3  14  A3  26    Inst. #  Vj  Vk    A3  14  A4    A4  14  A4    A2  14  A3  26    Inst #  Address  V (f    A2  14  A3  26    :  Inst. #  Vj  Vk    A3  14  10    Inst. #  Vj  Vk    A3  14  10    Inst #  Address  V (f    A2  14  14  10	A2  14    r1  r2  r3  r4  r5    A2  14  A3  26  26    r1  r2  r3  r4  r5    A2  14  A3  26  26    Inst. #  Vj  Vk  Qj    A3  14  40  40    A3  14  40  40  40    Inst #  Address  V (for stores)  41    A1  14  40  40  40    Inst. #  Vj  Vk  Qj  41    A3  14  10  41  41    Inst. #  Vj  Vk  Qj  A2    Inst. #  Vj  Vk  Qj  A3    Inst #  Address  V (for stores)  A2    Inst #  Address  V (for stores)  A2    Inst #  Address  V (for stores)    A2  14  10  10	A2  14  A2  14    r1  r2  r3  r4  r5  r6    A2  14  A3  26

## Clock cycle 6:

Name		Ins	st. #	Vj	Vk	Qj		Qk	State
Add/Sub1		A	15		26	A3			Wait
Add/Sub2									
Add/Sub3									
Mult/Div1		A	3	14	10				Execute
Mult/Div2									
Name		I	nst #	Address	s V	V (for stores)		res)	State
Load/Stor	e1								
Load/Store2			A4	14		10 Exe		te	
Load/Stor	e3								
r0	r1	l	r2	r3	r4	r5	r6	_	
	1(	0	14	A3	26	A5			