

Computer Architecture
DAT105
Exercise Session 6

M. WAQAR AZHAR
waqarm@chalmers.se

Chalmers University of Technology, Sweden

October 13, 2019

Agenda

- ▶ Problem 3.22
- ▶ Problem 3.25

Agenda

- ▶ Problem 3.22
- ▶ Problem 3.25

Problem 3.22

This problem is about a VLIW extension of the 5-stage pipeline shown in Figure 3.48. Pipeline registers between stages are not shown but are present and are named as usual, such as ID1/EX1. Conditional branches and unconditional jumps are delayed by one long instruction and are executed in the ID4 stage in all cases, so that the long instruction following the branch in the fetch stage is always executed, whether or not the branch is taken.

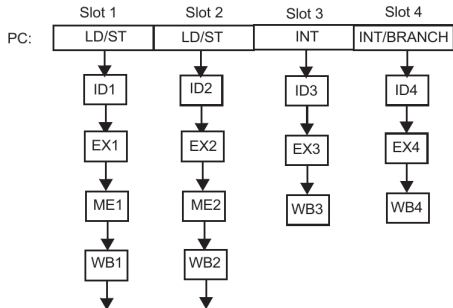
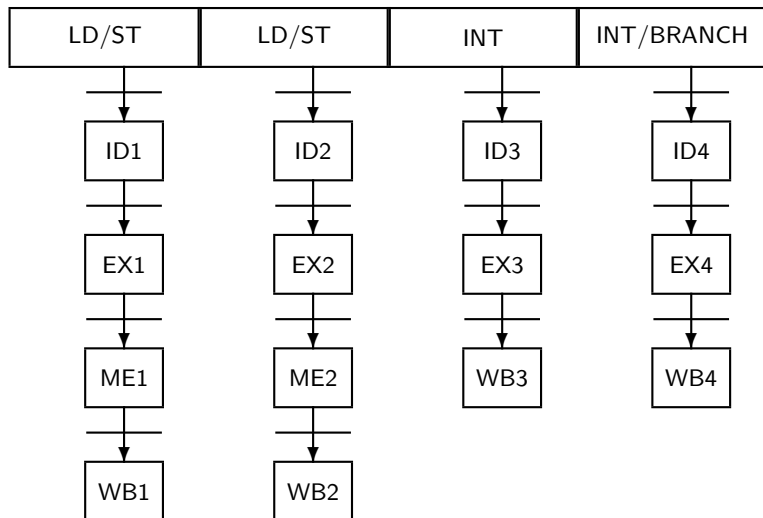


Figure 3.48. VLIW Integer pipeline

Problem 3.22



Problem 3.22 (a)

Compute the following operation latencies:

- ▶ L.W $\Rightarrow$ INT (on the register)
- ▶ L.W $\Rightarrow$ S.W (on the memory operand and on the address register)
- ▶ INT $\Rightarrow$ L.W or S.W (on the address register)
- ▶ INT $\Rightarrow$ BRANCH (on registers)
- ▶ L.W $\Rightarrow$ BRANCH (on registers)

Problem 3.22 (a)

Compute the following operation latencies:

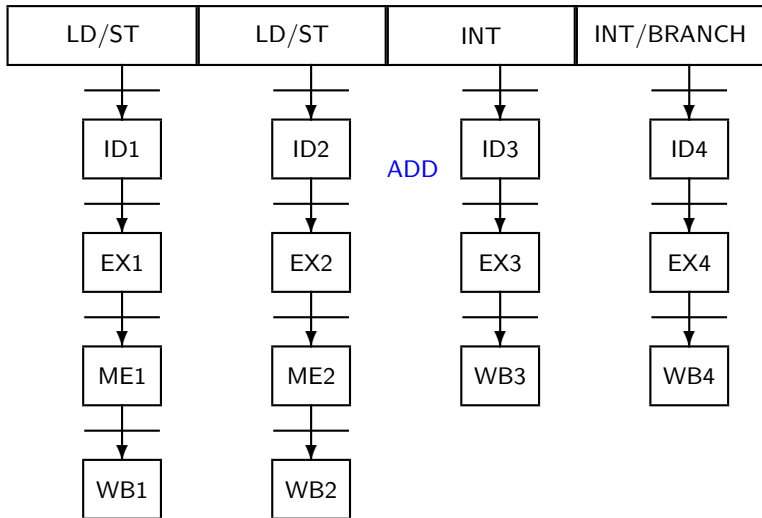
- ▶ L.W $\Rightarrow$ INT (on the register)
- ▶ L.W $\Rightarrow$ S.W (on the memory operand and on the address register)
- ▶ INT $\Rightarrow$ L.W or S.W (on the address register)
- ▶ INT $\Rightarrow$ BRANCH (on registers)
- ▶ L.W $\Rightarrow$ BRANCH (on registers)

For three cases:

- ▶ No forwarding at all
- ▶ Internal register forwarding only
- ▶ Full forwarding (includes internal register forwarding)

Problem 3.22 (a) - No forwarding at all

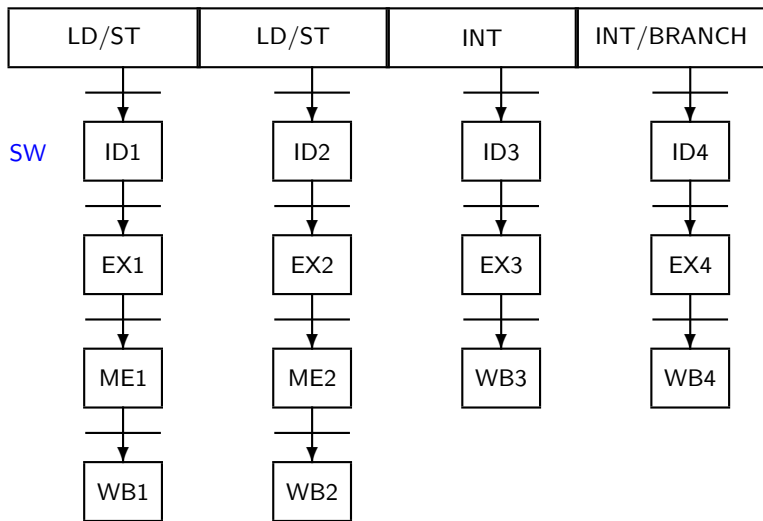
L.W => INT (on the register)



LW

Problem 3.22 (a) - No forwarding at all

L.W => S.W

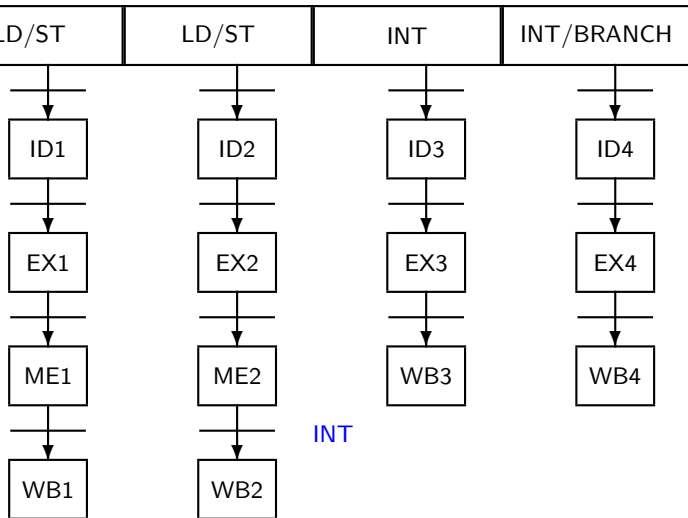


LW

Problem 3.22 (a) - No forwarding at all

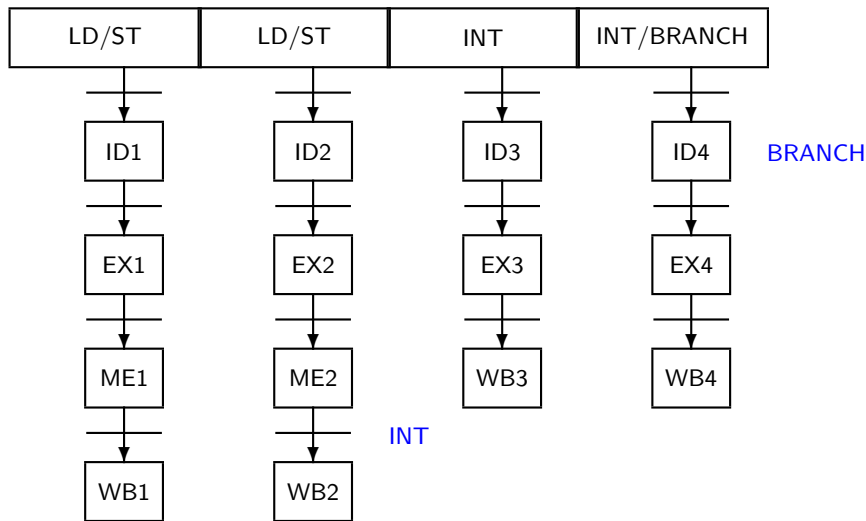
INT => L.W/S.W

S.W/L.W



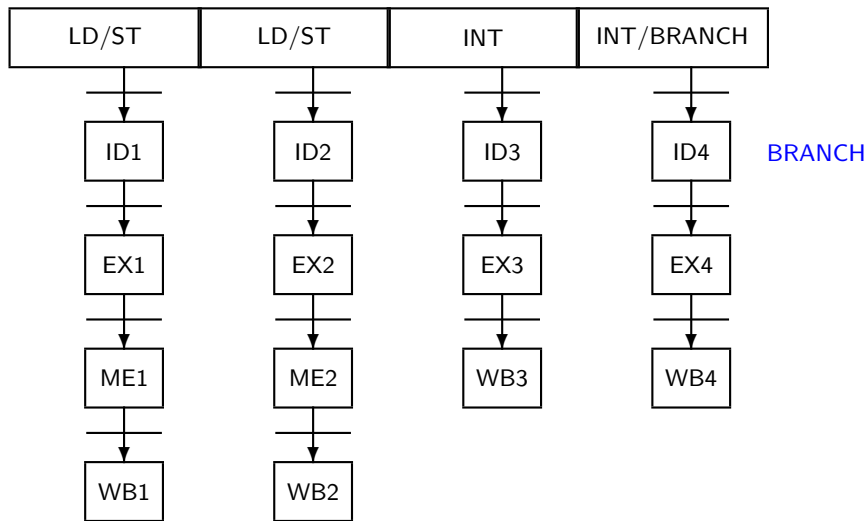
Problem 3.22 (a) - No forwarding at all

INT => BRANCH



Problem 3.22 (a) - No forwarding at all

L.W => BRANCH

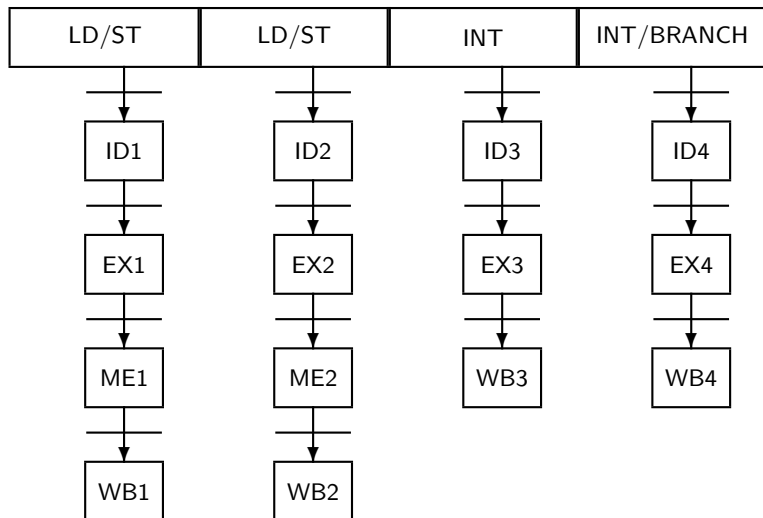


LW

Problem 3.22 (a)

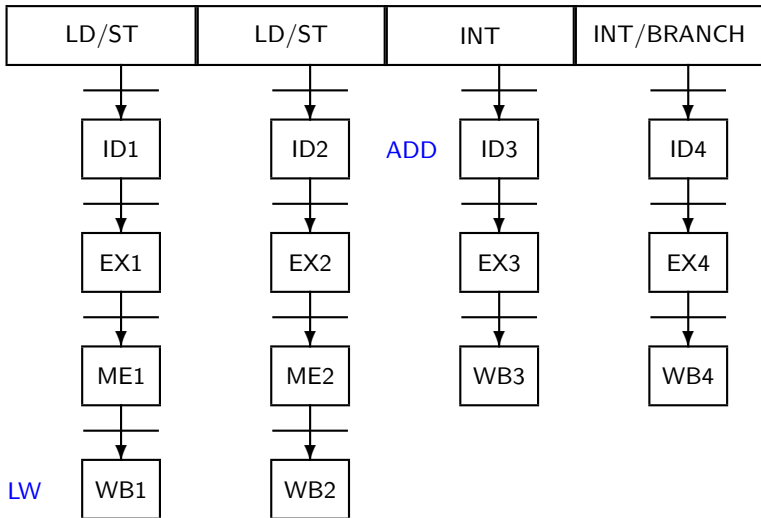
	No Forwarding	Register Forwarding	Full Forwarding
L.W => INT	3	-	-
L.W => S.W	3	-	-
INT => L.W or S.W	2	-	-
INT => BRANCH	2	-	-
L.W => BRANCH	3	-	-

Problem 3.22 (a) - Register forwarding



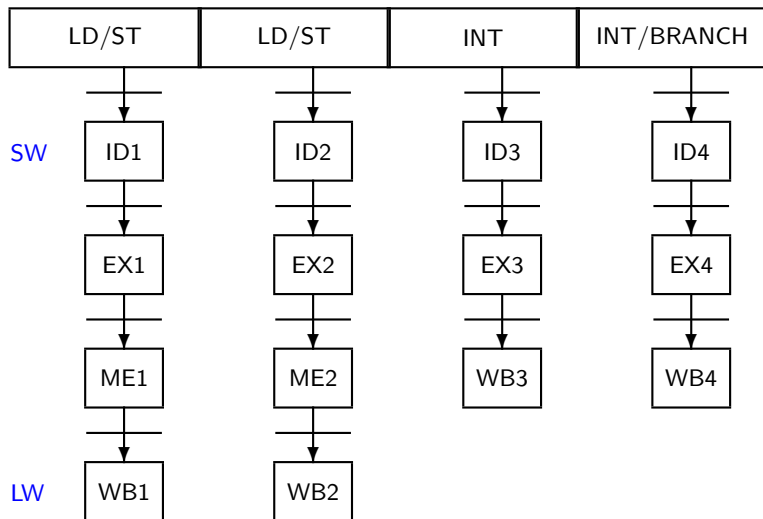
Problem 3.22 (a)- Register forwarding

L.W => INT (on the register)



Problem 3.22 (a) - Register forwarding

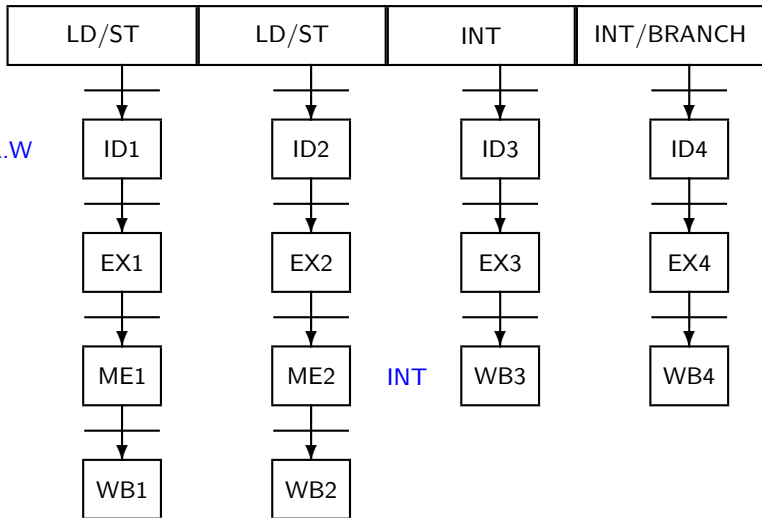
L.W => S.W



Problem 3.22 (a) - Register forwarding

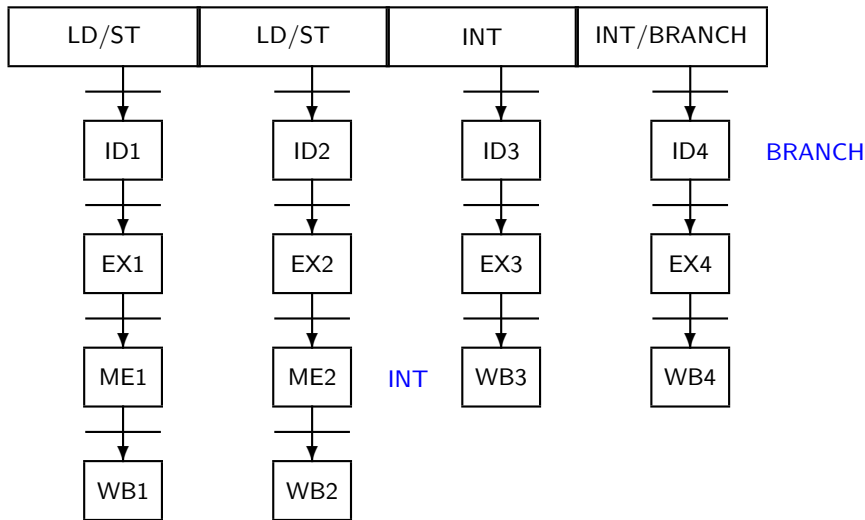
INT $\Rightarrow$ L.W/S.W

S.W/L.W



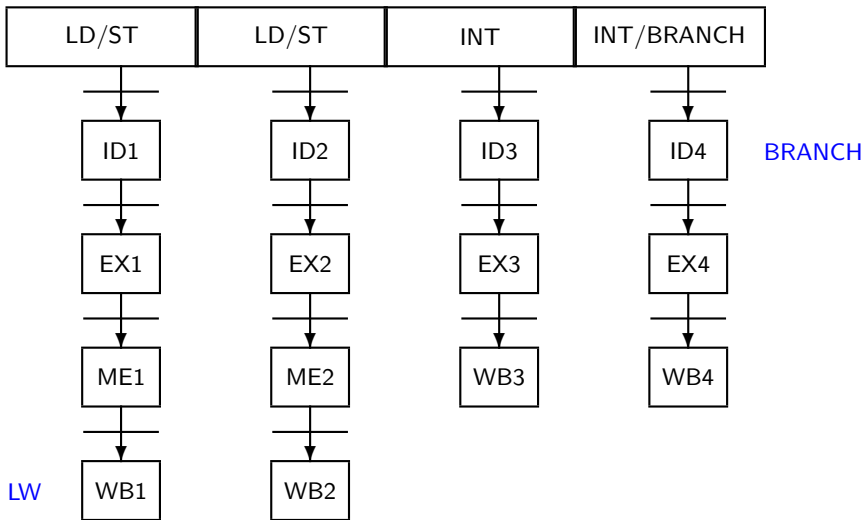
Problem 3.22 (a) - Register forwarding

INT => BRANCH



Problem 3.22 (a) - Register forwarding

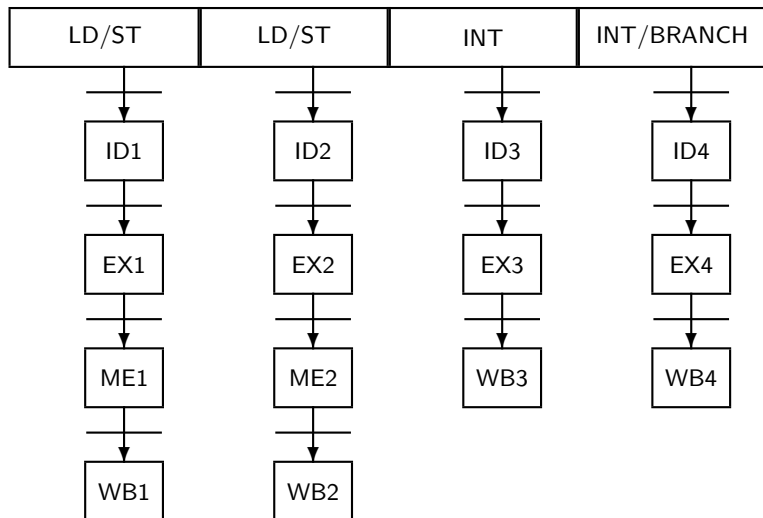
L.W => BRANCH



Problem 3.22

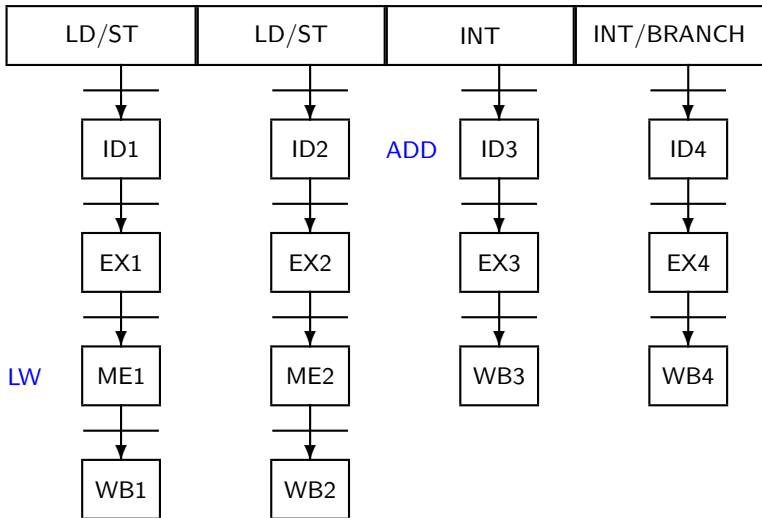
	No Forwarding	Register Forwarding	Full Forwarding
L.W => INT	3	2	-
L.W => S.W	3	2	-
INT => L.W or S.W	2	1	-
INT => BRANCH	2	1	-
L.W => BRANCH	3	2	-

Problem 3.22 (a) - Full Forwarding



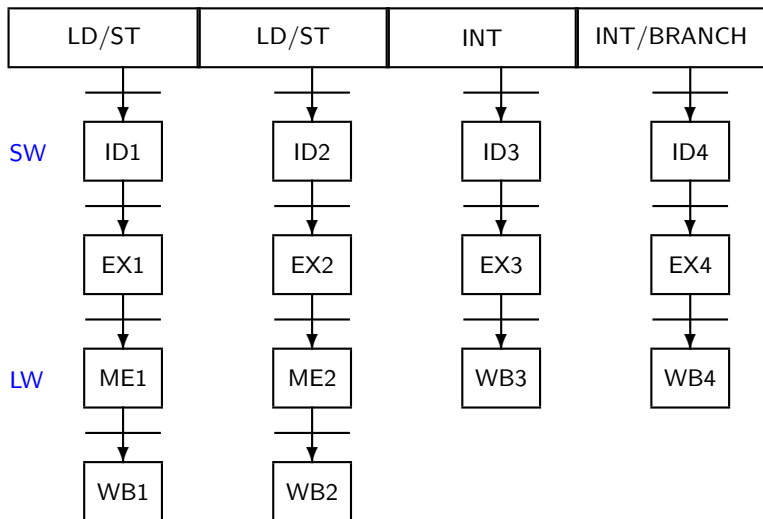
Problem 3.22 (a) - Full Forwarding

L.W => INT (on the register)



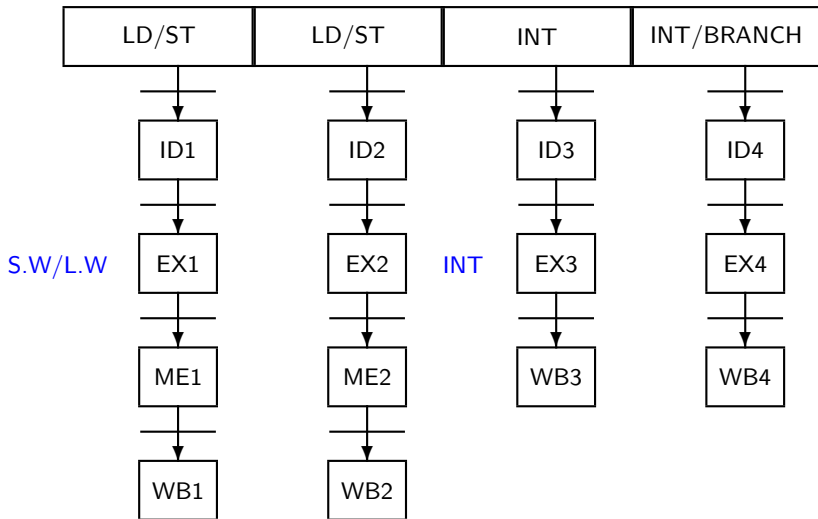
Problem 3.22 (a) - Full forwarding

L.W => S.W



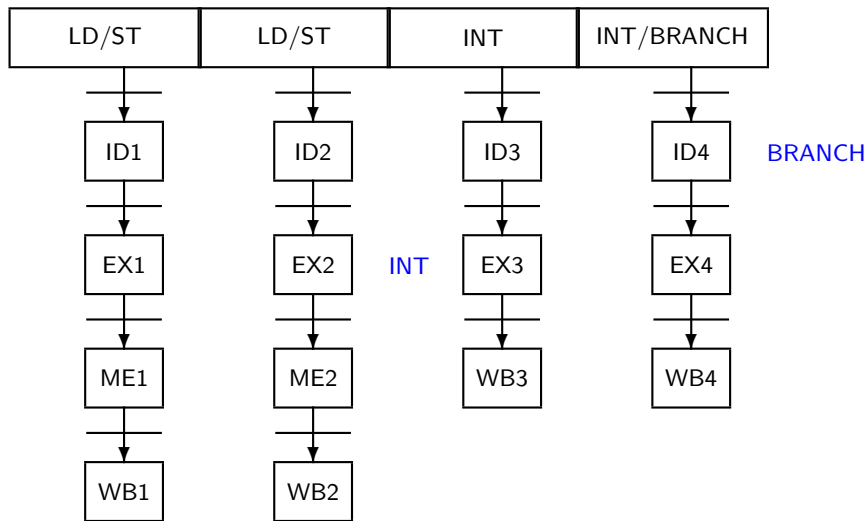
Problem 3.22 (a) - Full forwarding

INT $\Rightarrow$ L.W/S.W



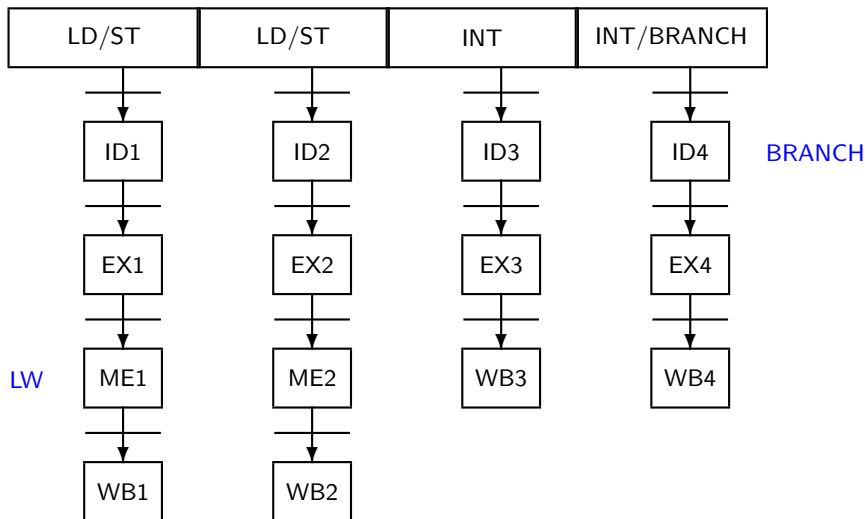
Problem 3.22 (a) - Full forwarding

INT => BRANCH



Problem 3.22 (a) - Full Forwarding

L.W => BRANCH



Problem 3.22 (a)

	No Forwarding	Register Forwarding	Full Forwarding
L.W => INT	3	2	1
L.W => S.W	3	2	1
INT => L.W or S.W	2	1	0
INT => BRANCH	2	1	1
L.W => BRANCH	3	2	1

Problem 3.22 (Part -b)

Consider again the code counting matches to a key in a vector of values:

SEARCH:	LW R5,0(R3)	I1 Load item
	SUB R6,R5,R2	I2 Compare with key
	BNEZ R6,NOMATCH	I3 Check for match
	ADDI R1,R1,# 1	I4 Count matches
NOMATCH:	ADDI R3,R3,# 4	I5 Next item
	BNE R4,R3,SEARCH	I6 Continue until all items

Because the branch inside the loop is an impediment to parallelization of the code, we replace it with a conditional move instruction CMOVZ:

Problem 3.22 (Part -b)

SEARCH:	LW R5,0(R3)	I1 Load item
	SUB R6,R5,R2	I2 Compare with key
	ADDI R7,R1,# 1	I3 Assume a match
	CMOVZ R1,R7,R6	I4 If match, then R1 is increased
	ADDI R3,R3,# 4	I5 Next item
	BNE R4,R3,SEARCH	I6 Continue until all items

To further enhance parallelism the compiler unroll the loop three times. Show the best possible VLIW code for the three forwarding options. Use the same format as in Table 3.19.

Problem 3.22 (Part -b)

```
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
ADDI R3,R3,# 12
BNE R4,R3,SEARCH
```

Problem 3.22 (Part -b)

```
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
ADDI R3,R3,# 12
BNE R4,R3,SEARCH
```

```
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
LW R8,4(R3)
SUB R9,R8,R2
ADDI R10,R1,# 1
CMOVZ R1,R10,R9
LW R11,8(R3)
SUB R12,R11,R2
ADDI R13,R1,# 1
CMOVZ R1,R13,R12
ADDI R3,R3,# 12
BNE R4,R3,SEARCH
```

Problem 3.22 (Part -b)

LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
ADDI R3,R3,# 12
BNE R4,R3,SEARCH

LW R5,0(R3)
SUB R6,R5,R2
ADDI R7,R1,# 1
CMOVZ R1,R7,R6
LW R8,4(R3)
SUB R9,R8,R2
ADDI R10,R1,# 1
CMOVZ R1,R10,R9
LW R11,8(R3)
SUB R12,R11,R2
ADDI R13,R1,# 1
CMOVZ R1,R13,R12
ADDI R3,R3,# 12
BNE R4,R3,SEARCH

LW R5,0(R3)
LW R8,4(R3)
LW R11,8(R3)
ADDI R7,R1,# 1
SUB R6,R5,R2
SUB R9,R8,R2
SUB R12,R11,R2
CMOVZ R1,R7,R6
ADDI R10,R1,# 1
ADDI R3,R3,# 12
CMOVZ R1,R10,R9
ADDI R13,R1,# 1
BNE R4,R3,SEARCH
CMOVZ R1,R13,R12

Problem 3.22 No Forwarding

	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	LW R5,0(R3)	LW R8,4(R3)	NOOP	NOOP
2	LW R11,8(R3)	NOOP	NOOP	NOOP
3	NOOP	NOOP	ADDI R7,R1,#1	NOOP
4	NOOP	NOOP	NOOP	NOOP
5	NOOP	NOOP	SUB R6,R5,R2	SUB R9,R8,R2
6	NOOP	NOOP	SUB R12,R11,R2	NOOP
7	NOOP	NOOP	NOOP	NOOP
8	NOOP	NOOP	CMOVZ R1,R7,R6	NOOP
9	NOOP	NOOP	NOOP	NOOP
10	NOOP	NOOP	NOOP	NOOP
11	NOOP	NOOP	ADDI R10,R1,#1	ADDI R3,R3,#12
12	NOOP	NOOP	NOOP	NOOP
13	NOOP	NOOP	NOOP	NOOP
14	NOOP	NOOP	CMOVZ R1,R10,R9	NOOP
15	NOOP	NOOP	NOOP	NOOP
16	NOOP	NOOP	NOOP	NOOP
17	NOOP	NOOP	ADDI R13,R1,#1	NOOP
18	NOOP	NOOP	NOOP	NOOP
19	NOOP	NOOP	NOOP	BNE R4,R3,SEARCH
20	NOOP	NOOP	CMOVZ R1,R13,R12	NOOP

Problem 3.22 Register Forwarding

	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	LW R5,0(R3)	LW R8,4(R3)	NOOP	NOOP
2	LW R11,8(R3)	NOOP	NOOP	NOOP
3	NOOP	NOOP	ADDI R7,R1,#1	NOOP
4	NOOP	NOOP	SUB R6,R5,R2	SUB R9,R8,R2
5	NOOP	NOOP	SUB R12,R11,R2	NOOP
6	NOOP	NOOP	CMOVZ R1,R7,R6	NOOP
7	NOOP	NOOP	NOOP	NOOP
8	NOOP	NOOP	ADDI R10,R1,#1	NOOP
9	NOOP	NOOP	NOOP	NOOP
10	NOOP	NOOP	CMOVZ R1,R10,R9	NOOP
11	NOOP	NOOP	NOOP	ADDI R3,R3,#12
12	NOOP	NOOP	ADDI R13,R1,#1	NOOP
13	NOOP	NOOP	NOOP	BNE R4,R3,SEARCH
14	NOOP	NOOP	CMOVZ R1,R13,R12	NOOP

Problem 3.22 Full Forwarding

	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	LW R5,0(R3)	LW R8,4(R3)	NOOP	NOOP
2	LW R11,8(R3)	NOOP	ADDI R7,R1,#1	NOOP
3	NOOP	NOOP	SUB R6,R5,R2	SUB R9,R8,R2
4	NOOP	NOOP	SUB R12,R11,R2	CMOVZ R1,R7,R6
5	NOOP	NOOP	ADDI R10,R1,#1	ADDI R3,R3,#12
6	NOOP	NOOP	CMOVZ R1,R10,R9	NOOP
7	NOOP	NOOP	ADDI R13,R1,#1	BNE R4,R3,SEARCH
8	NOOP	NOOP	CMOVZ R1,R13,R12	NOOP

Agenda

- ▶ Problem 3.22
- ▶ Problem 3.25

Problem 3.25 Problem Definition

We revisit the code of Problem 3.24 above with branches and jumps.

```
l1      LW R5,0(R1)      /load A
l2      LW R7,0(R3)      /load C
l3      SLT R8,R5,R7      /test A<C
l4      LW R6,0(R2)      /load B
l5      BNEZ R8, else
l6      SLT R9,R5,R6
l7      BNEZ R9, else      /test A<B
l8      ADD R10,R6,R7
l9      SW R10,0(R1)      /execute if clause
l10     J exit
l11  else  LW R11,0(R4)      /load D
l12     SLT R12,R11,R6
l13     BEZ R12,else1      /test D<B
l14     ADD R13,R7,R11
l15     BNE R5,R13,exit      /test A==C+D
l16  else1 SUB R14,R5,R7
l17     SW R14,0(R2)
l18  exit
```

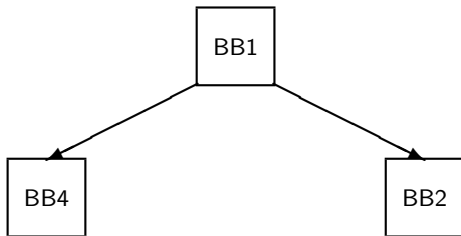
Problem 3.25 (Part-a)

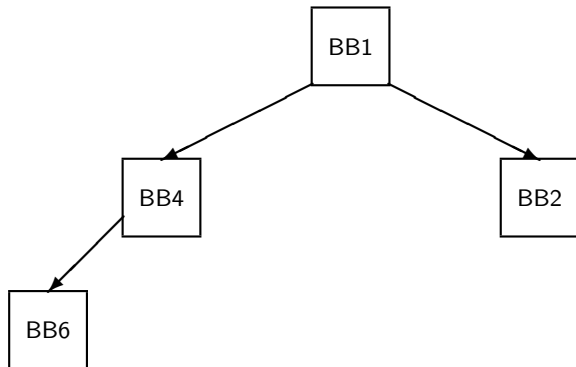
Identify all basic blocks in the code

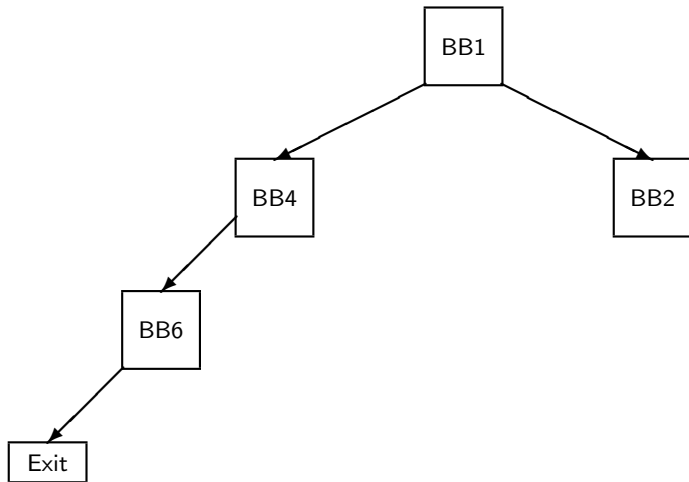
Problem 3.25

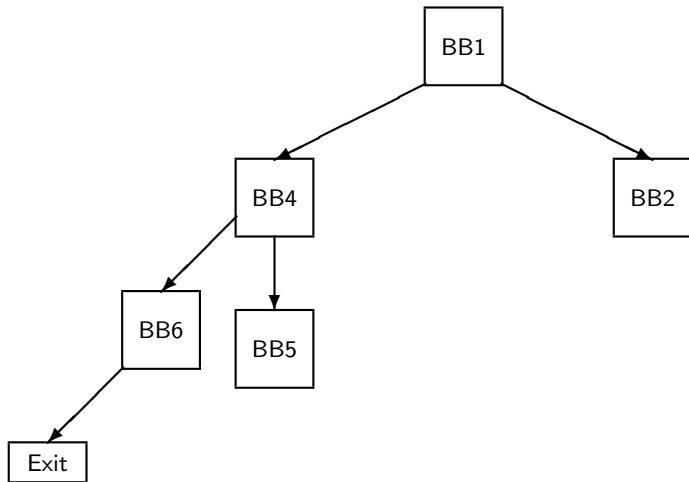
I1		LW R5,0(R1)	/load A	
I2		LW R7,0(R3)	/load C	
I3		SLT R8,R5,R7	/test $A < C$	BB1
I4		LW R6,0(R2)	/load B	
I5		BNEZ R8, else		
I6		SLT R9,R5,R6		BB2
I7		BNEZ R9, else	/test $A < B$	
I8		ADD R10,R6,R7		
I9		SW R10,0(R1)	/execute if clause	BB3
I10		J exit		
I11	else	LW R11,0(R4)	/load D	
I12		SLT R12,R11,R6		BB4
I13		BEZ R12,else1	/test $D < B$	
I14		ADD R13,R7,R11		BB5
I15		BNE R5,R13,exit	/test $A == C + D$	
I16	else1	SUB R14,R5,R7		BB6
I17		SW R14,0(R2)		
I18	exit			

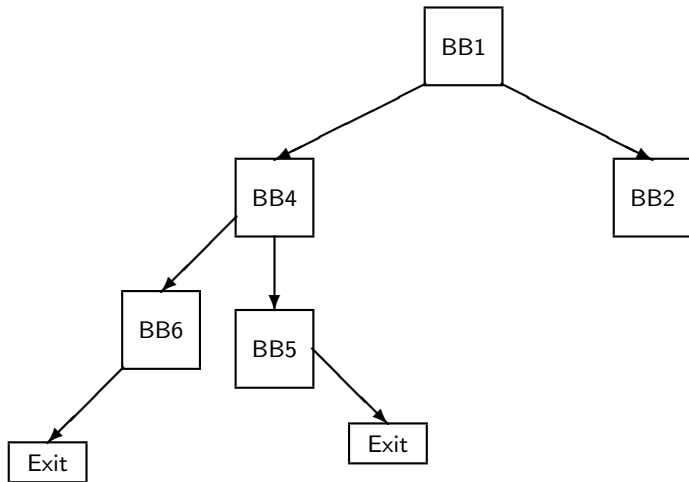
BB1

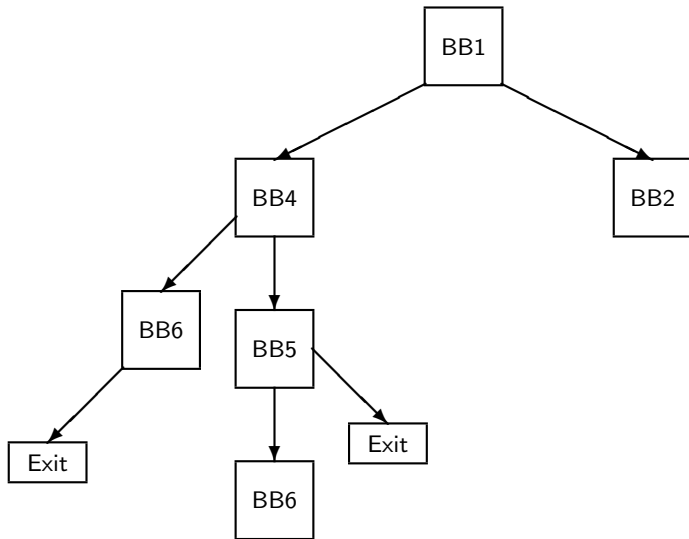


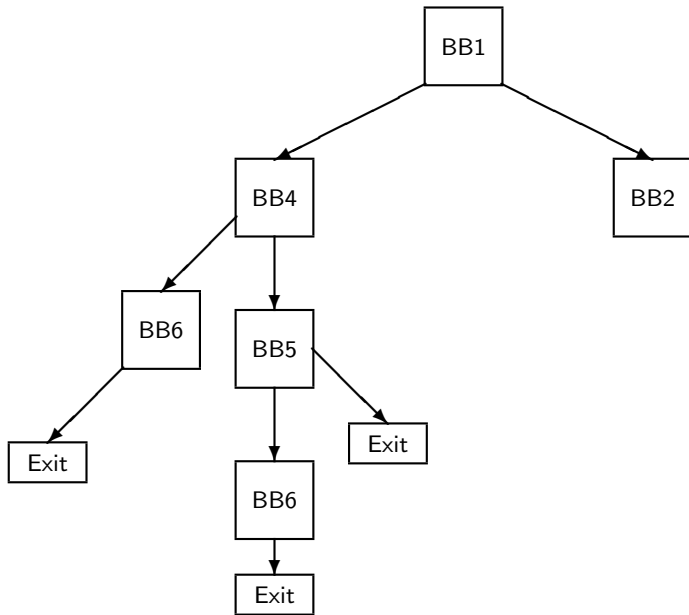


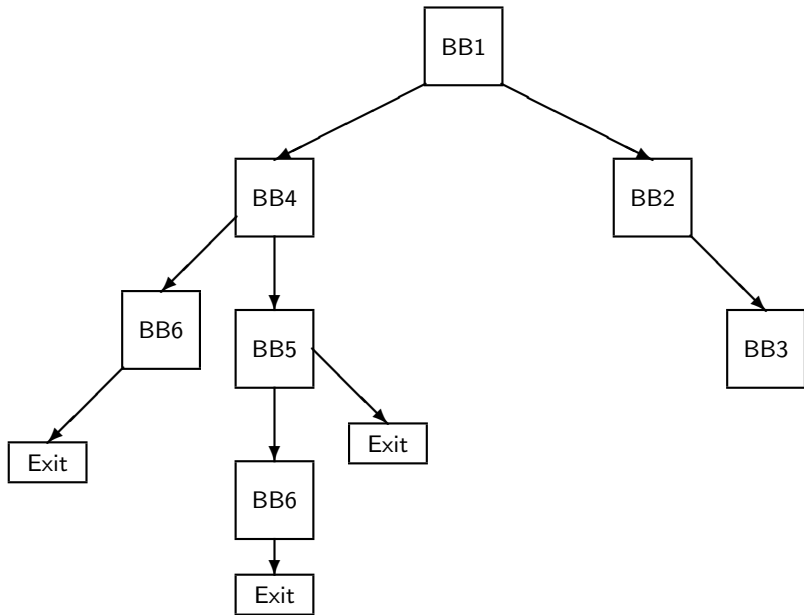


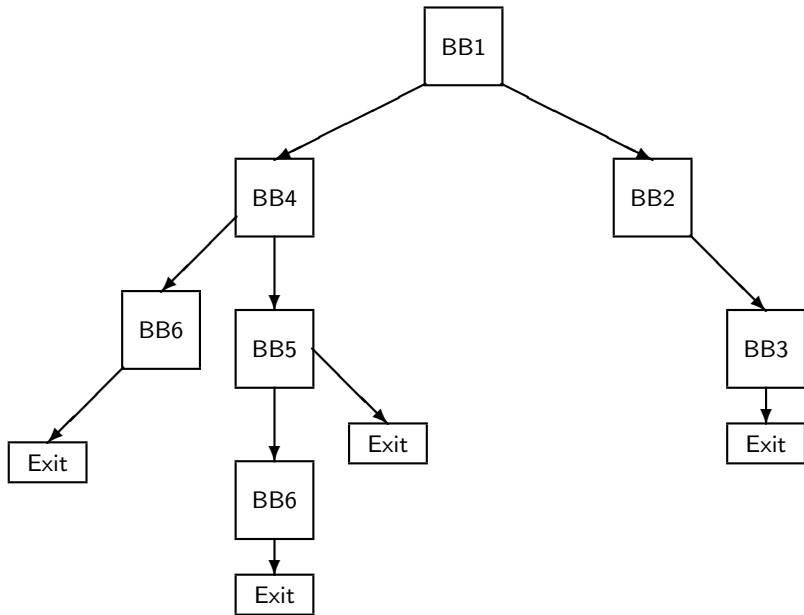


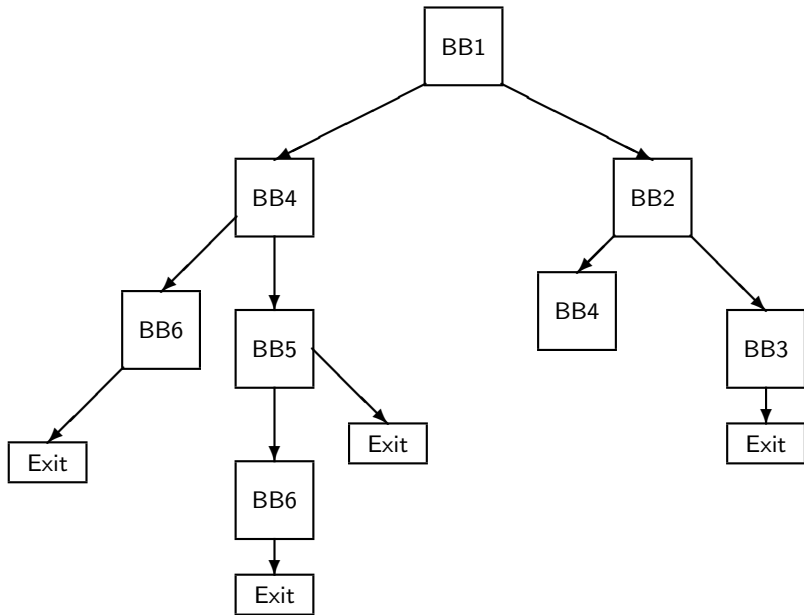


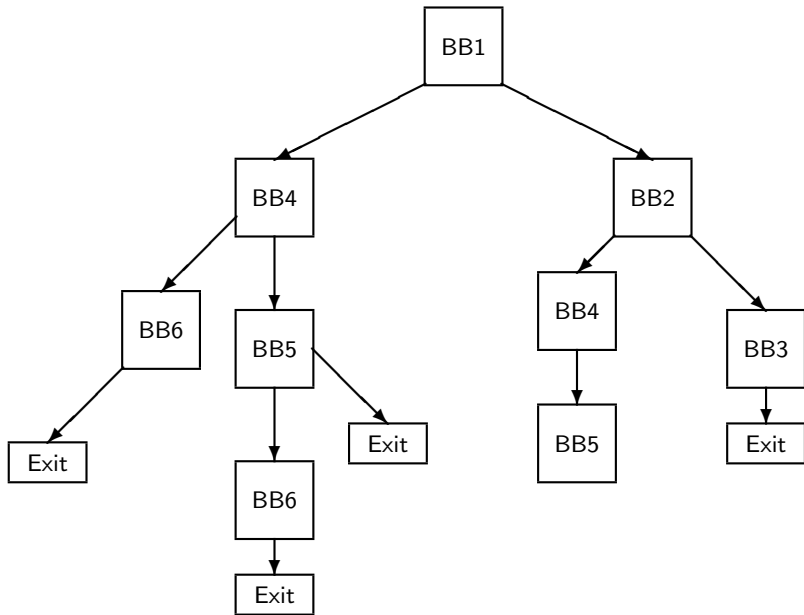


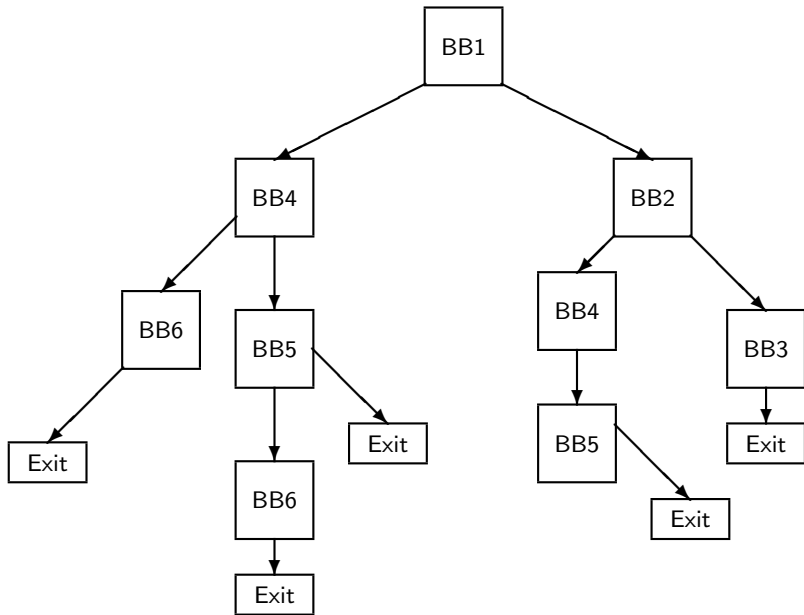


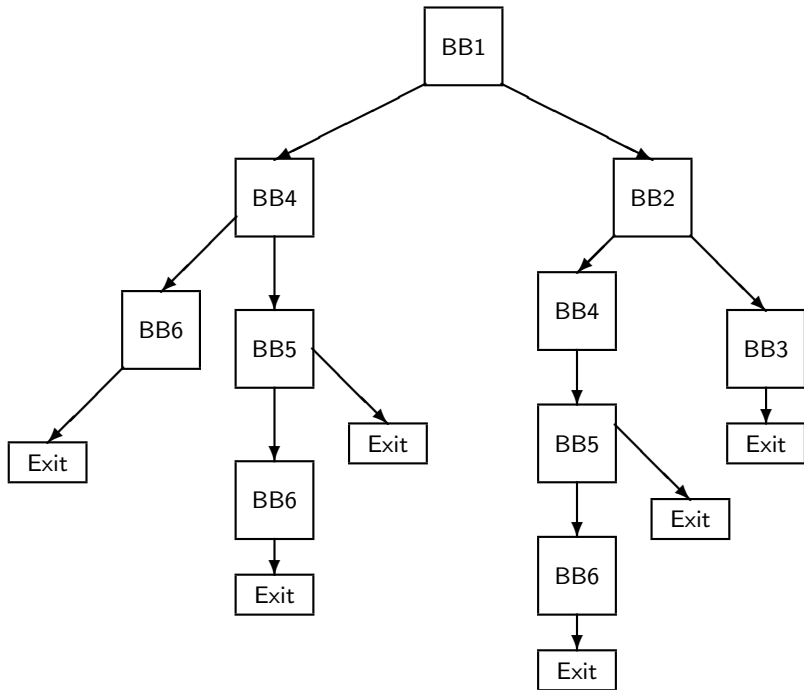












Problem 3.25 (Part-b)

Schedule the code the best you can by using local scheduling only (i.e., within basic blocks) on the VLIW machine of Problem 3.22. Note that the branch is delayed by one cycle. This delay has not been scheduled in the code.

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit

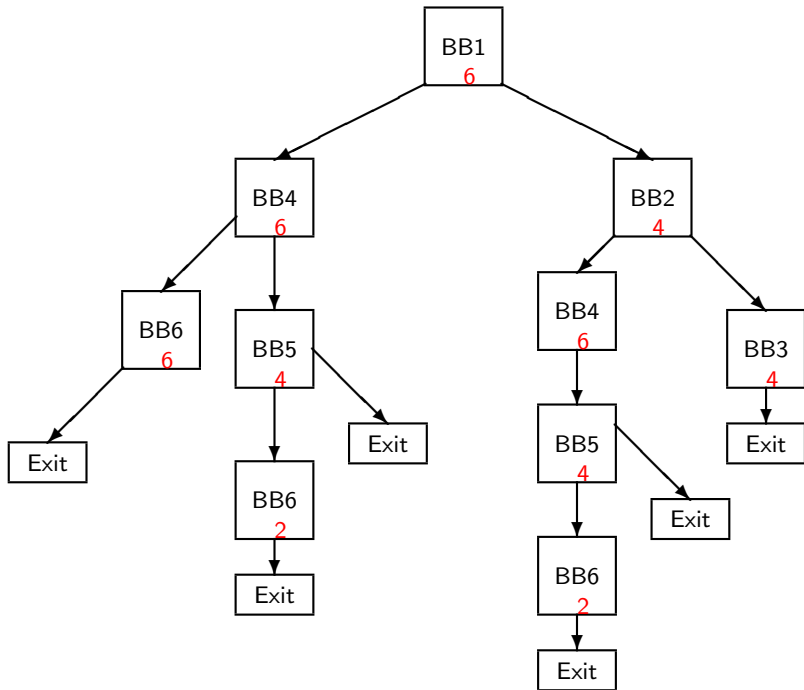
BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	24		NOOP	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	24		NOOP	NOOP	NOOP	NOOP
6	25	else1	NOOP	NOOP	SUB R14,R5,R7	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	24		NOOP	NOOP	NOOP	NOOP
6	25	else1	NOOP	NOOP	SUB R14,R5,R7	NOOP
6	26		SW R14,0(R2)	NOOP	NOOP	NOOP

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
3	13		NOOP	NOOP	NOOP	J exit
3	14		NOOP	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	24		NOOP	NOOP	NOOP	NOOP
6	25	else1	NOOP	NOOP	SUB R14,R5,R7	NOOP
6	26		SW R14,0(R2)	NOOP	NOOP	NOOP
I27	exit					

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH	Cycles
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP	6
1	2		NOOP	NOOP	NOOP	NOOP	
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP	
1	4		NOOP	NOOP	NOOP	NOOP	
1	5		NOOP	NOOP	NOOP	BNEZ R8,else	
1	6		NOOP	NOOP	NOOP	NOOP	
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP	4
2	8		NOOP	NOOP	NOOP	NOOP	
2	9		NOOP	NOOP	NOOP	BNEZ R9,else	
2	10		NOOP	NOOP	NOOP	NOOP	
3	11		NOOP	NOOP	ADD R10,R6,R7	NOOP	4
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP	
3	13		NOOP	NOOP	NOOP	J exit	
3	14		NOOP	NOOP	NOOP	NOOP	
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP	6
4	16		NOOP	NOOP	NOOP	NOOP	
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP	
4	18		NOOP	NOOP	NOOP	NOOP	
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1	
4	20		NOOP	NOOP	NOOP	NOOP	
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP	4
5	22		NOOP	NOOP	NOOP	NOOP	
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit	
5	24		NOOP	NOOP	NOOP	NOOP	
6	25	else1	NOOP	NOOP	SUB R14,R5,R7	NOOP	2
6	26		SW R14,0(R2)	NOOP	NOOP	NOOP	
	27	exit					



Problem 3.25 (Part-b)

Trace	Basic Blocks	Cycles
Trace1	BB1, BB2, BB3	14
Trace2	BB1, BB2, BB4, BB5, BB6	22
Trace3	BB1, BB2, BB4, BB5	20
Trace4	BB1, BB2, BB4, BB6	18
Trace5	BB1, BB4, BB5, BB6	18
Trace6	BB1, BB4, BB5	16
Trace7	BB1, BB4, BB6	14

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT / BRANCH

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	24		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	24		NOOP	NOOP	NOOP	NOOP
6	25	else1	NOOP	NOOP	SUB R14,R5,R7	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	24		NOOP	NOOP	NOOP	NOOP
6	25	else1	NOOP	NOOP	SUB R14,R5,R7	NOOP
6	26		SW R14,0(R2)	NOOP	NOOP	NOOP

VLIW program with local scheduling

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		NOOP	NOOP	NOOP	NOOP
1	3		LW R6,0(R2)	NOOP	SLT R8,R5,R7	NOOP
1	4		NOOP	NOOP	NOOP	NOOP
1	5		NOOP	NOOP	NOOP	BNEZ R8,else
1	6		NOOP	NOOP	NOOP	NOOP
2	7		NOOP	NOOP	SLT R9,R5,R6	NOOP
2	8		NOOP	NOOP	NOOP	NOOP
2	9		NOOP	NOOP	NOOP	BNEZ R9,else
2	10		NOOP	NOOP	NOOP	NOOP
3	11		NOOP	NOOP	ADD R10,R6,R7	J exit
3	12		SW R10,0(R1)	NOOP	NOOP	NOOP
4	15	else	LW R11,0(R4)	NOOP	NOOP	NOOP
4	16		NOOP	NOOP	NOOP	NOOP
4	17		NOOP	NOOP	SLT R12,R11,R6	NOOP
4	18		NOOP	NOOP	NOOP	NOOP
4	19		NOOP	NOOP	NOOP	BEQZ R12,else1
4	20		NOOP	NOOP	NOOP	NOOP
5	21		NOOP	NOOP	ADD R13,R7,R11	NOOP
5	22		NOOP	NOOP	NOOP	NOOP
5	23		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	24		NOOP	NOOP	NOOP	NOOP
6	25	else1	NOOP	NOOP	SUB R14,R5,R7	NOOP
6	26		SW R14,0(R2)	NOOP	NOOP	NOOP
	27	exit				

Problem 3.25 (Part-b)

Tarce	Baisc Blocks	Cycles Before	Cycles New	Speedup
Trace1	BB1, BB2, BB3	14	12	1.17
Trace2	BB1, BB2, BB4, BB5, BB6	22	22	1
Trace3	BB1, BB2, BB4, BB5	20	20	1
Trace4	BB1, BB2, BB4, BB6	18	18	1
Trace5	BB1, BB4, BB5, BB6	18	18	1
Trace6	BB1, BB4, BB5	16	16	1
Trace7	BB1, BB4, BB6	14	14	1

Problem 3.25 (Part-c)

I1		LW R5,0(R1)	/load A
I2		LW R7,0(R3)	/load C
I3		SLT R8,R5,R7	
I11		LW R11,0(R4)	/load D
I12		SLT R12,R11,R6	
I14		ADD R13,R7,R11	
I4		LW R6,0(R2)	/load B
I6		SLT R9,R5,R6	
I8		ADD R10,R6,R7	
I16		SUB R14,R5,R7	
I5		BNEZ R8, then	/test $A < C$
I7		BNEZ R9, then	/test $A < B$
I9		SW R10,0(R1)	/execute if clause
I10		J exit	
I13	then	BNEZ R12,then1	/test $D < B$
I15		BNE R5,R13,exit	/test $A == C + D$
I17	then1	SW R14,0(R2)	/execute Then clause
I18	exit		

Problem 3.25

The student decided not to move stores up across branches. Why is this a good thing?

Problem 3.25

The student decided not to move stores up across branches. Why is this a good thing?

Store instructions modify the value in memory.

Problem 3.25

The student decided not to move stores up across branches. Why is this a good thing?

Store instructions modify the value in memory.

Store instructions cannot be moved up across a branch.

Problem 3.25

Add the instructions in the new code to check for mispeculations (both memory and exceptions).

Problem 3.25 (Part-c)

I1	LW R5,0(R1)	/load A
I2	LW R7,0(R3)	/load C
I3	SLT R8,R5,R7	
I4	LW R6,0(R2)	/load B
I11	LW.s R11,0(R4)	/load D
I12	SLT R12,R11,R6	
I14	ADD R13,R7,R11	
I6	SLT R9,R5,R6	
I8	ADD R10,R6,R7	
I16	SUB R14,R5,R7	
I5	BNEZ R8, then	/test A<C
I7	BNEZ R9, then	/test A<B
I9	SW R10,0(R1)	/execute if clause
I10	J exit	
then	check.s R11,repair	/inserted where I11 was
I13	BNEZ R12,then1	/test D<B
I15	BNE R5,R13,exit	/test A==C+D
I17	then1 SW R14,0(R2)	/execute Then clause
I18	exit	

Problem 3.25

Schedule the new code locally, taking advantage of the branch and jump delay slots.

Schedule the code on the VLIW machine of Problem 3.22.

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP
3	9		SW R10,0(R1)	NOOP	NOOP	J exit

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP
3	9		SW R10,0(R1)	NOOP	NOOP	J exit
3	10		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP
3	9		SW R10,0(R1)	NOOP	NOOP	J exit
3	10		NOOP	NOOP	NOOP	NOOP
4	11	then	NOOP	NOOP	check.s R11,repair	BEQZ R12,then1

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP
3	9		SW R10,0(R1)	NOOP	NOOP	J exit
3	10		NOOP	NOOP	NOOP	NOOP
4	11	then	NOOP	NOOP	check.s R11,repair	BEQZ R12,then1
4	12		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP
3	9		SW R10,0(R1)	NOOP	NOOP	J exit
3	10		NOOP	NOOP	NOOP	NOOP
4	11	then	NOOP	NOOP	check.s R11,repair	BEQZ R12,then1
4	12		NOOP	NOOP	NOOP	NOOP
5	13		NOOP	NOOP	NOOP	BNE R5,R13,exit

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP
3	9		SW R10,0(R1)	NOOP	NOOP	J exit
3	10		NOOP	NOOP	NOOP	NOOP
4	11	then	NOOP	NOOP	check.s R11,repair	BEQZ R12,then1
4	12		NOOP	NOOP	NOOP	NOOP
5	13		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	14		NOOP	NOOP	NOOP	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP
3	9		SW R10,0(R1)	NOOP	NOOP	J exit
3	10		NOOP	NOOP	NOOP	NOOP
4	11	then	NOOP	NOOP	check.s R11,repair	BEQZ R12,then1
4	12		NOOP	NOOP	NOOP	NOOP
5	13		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	14		NOOP	NOOP	NOOP	NOOP
6	15	then1	SW R14,0(R2)	NOOP	NOOP	NOOP

VLIW program with local scheduling + compiler

BB	Cycle	Label	LD/ST 1	LD/ST 2	INT	INT/BRANCH
1	1		LW R5,0(R1)	LW R7,0(R3)	NOOP	NOOP
1	2		LW R6,0(R2)	LW.s R11,0(R4)	NOOP	NOOP
1	3		NOOP	NOOP	SLT R8,R5,R7	SUB R14,R5,R7
1	4		NOOP	NOOP	SLT R12,R11,R6	ADD R13,R7,R11
1	5		NOOP	NOOP	SLT R9,R5,R6	BNEZ R8,then
1	6		NOOP	NOOP	ADD R10,R6,R7	NOOP
2	7		NOOP	NOOP	NOOP	BNEZ R9,then
2	8		NOOP	NOOP	NOOP	NOOP
3	9		SW R10,0(R1)	NOOP	NOOP	J exit
3	10		NOOP	NOOP	NOOP	NOOP
4	11	then	NOOP	NOOP	check.s R11,repair	BEQZ R12,then1
4	12		NOOP	NOOP	NOOP	NOOP
5	13		NOOP	NOOP	NOOP	BNE R5,R13,exit
5	14		NOOP	NOOP	NOOP	NOOP
6	15	then1	SW R14,0(R2)	NOOP	NOOP	NOOP
	16	exit				

Problem 3.25 (Part-c)

The execution times of basic blocks are:

- ▶ BB1: 6 clocks
- ▶ BB2: 2 clocks
- ▶ BB3: 2 clocks
- ▶ BB4: 2 clocks
- ▶ BB5: 2 clocks
- ▶ BB6: 1 clocks

Problem 3.25 (Part-c)

What is the speedup obtained by this new code on the VLIW machine, for all possible cases?

Problem 3.25 (Part-c)

What is the speedup obtained by this new code on the VLIW machine, for all possible cases?

When is the new code with speculative loads better?

Problem 3.25 (Part-c)

Trace	Basic Blocks	Cycles Before	Cycles New	Speedup
Trace1	BB1, BB2, BB3	14	10	1.4
Trace2	BB1, BB2, BB4, BB5, BB6	22	13	1.69
Trace3	BB1, BB2, BB4, BB5	20	12	1.67
Trace4	BB1, BB2, BB4, BB6	18	11	1.64
Trace5	BB1, BB4, BB5, BB6	18	11	1.64
Trace6	BB1, BB4, BB5	16	10	1.6
Trace7	BB1, BB4, BB6	14	9	1.56