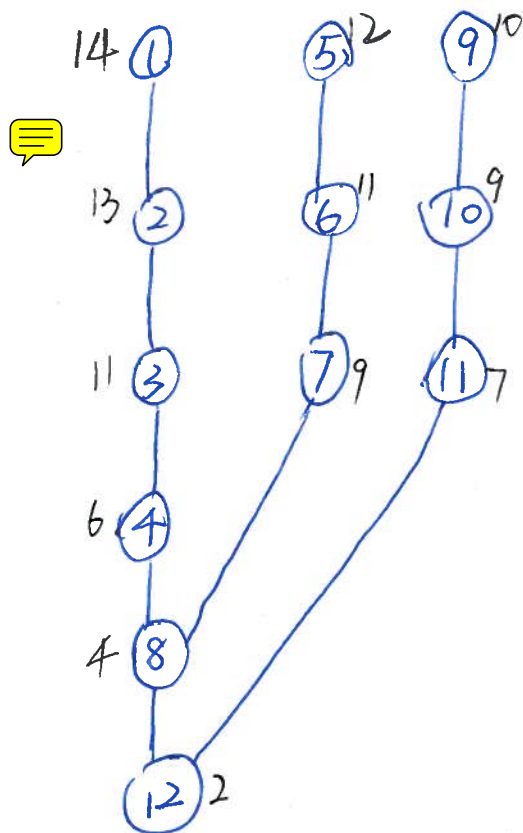
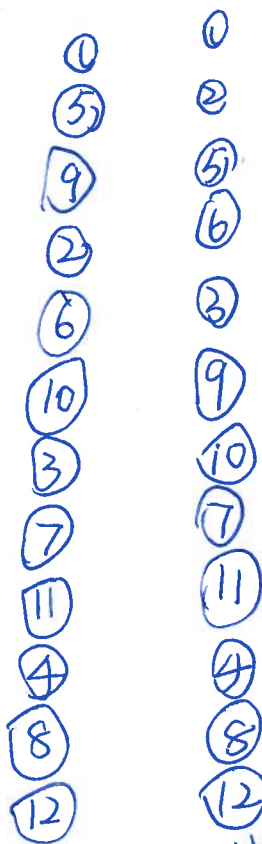


1. DDD with scheduling priority



b. Scheduling



original

	INT1	INT2	MEM
0	1		
1	2		
2			
3			3
4			
5			
6			
7			
8	4	5	
9	6		
10			7
11			
12			
13			
14			
15			
16	8	9	
17	10		
18			
19			11
20			
21			
22			
23			
24			12

Forward:

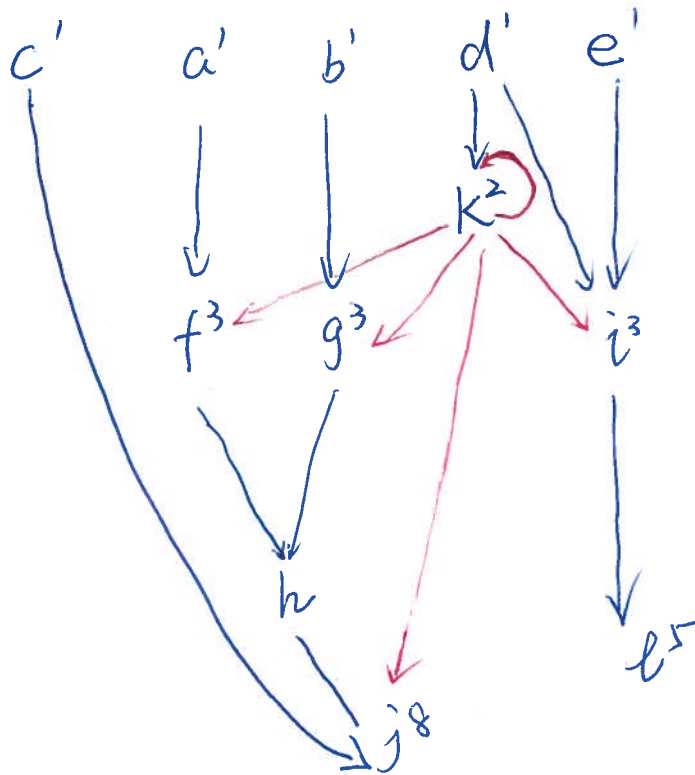
	INT1	INT2	MEM
0	1	5	
1	9	2	
2	6	10	
3			3
4			7
5			11
6			
7			
8	4		
9	8		
10	12		
11			

Since $12 < 24$
It's optimal.

Backward

	INT1	INT2	MEM
0	1		
1	2		
2	5		
3	6		3
4	9		
5	10		7
6			
7			11
8	4		
9	8		
10	12		
11			

2.



BC

$$\lceil 7/2 \rceil = 4$$

for load and store

$$\lceil 3/1 \rceil = 3$$

$$\text{So } BC = 4$$

$$DC = 1/1 = 1$$

$$\text{So } \min II = 4$$

Since k is antidependence with

f, g, i, j . It need to wait ~~for~~ them all done.

	Func 1	Func 2
k_1	f	i
k_2	g	i
k_3		
k_4		
k_5	h	
k_6		
k_7	j	k
k_8	l	

since all include in loop.
no neessary to generate prologue and epilogue.

- a addI $r_{arp} @x \Rightarrow r_{@x}$
- b addI $r_{arp} @y \Rightarrow r_{@y}$
- c addI $r_{arp} @z \Rightarrow r_{@z}$
- d loadI $0 \Rightarrow r_{ctr}$
- e loadI $792 \Rightarrow r_{ub}$
- f $L1 = \text{loadA0 } r_{ctr}, r_{@x} \Rightarrow r_x$
- g $\text{loadA0 } r_{ctr} r_{@y} \Rightarrow r_y$
- h mult $r_x, r_y \Rightarrow r_z$
- i cmpLT $r_{ctr} r_{ub} \Rightarrow r_{cc}$
- j store $A0, r_z \Rightarrow r_{ctr} r_{@i}$
- k addI $r_{ctr} 4 \Rightarrow r_{ctr}$
- l cbr $r_{cc} \rightarrow L1, L2$
- m $L2 \dots$

but if can. change j. store A0 $r_z \Rightarrow r_{ctr}, r_{@z}$

to store A0 $r_z \Rightarrow r_{ctr-4}, r_{@z}$, the kernel scheduling would be:

	Fun1	Fun2
P ₁	f	
P ₂	g	i
P ₃		
P ₄	K add1 $r_{ctr-4} \Rightarrow r_{ctr}$	
	L. if $r_{ctr} \geq r_{ub}$ goto e1	
K ₁	f	h
K ₂	g	i
K ₃	j	K
	$r_z \Rightarrow r_{ctr-4}, r_{@z}$	
K ₄	L	nop
	if $r_{ctr} \geq r_{ub}$ goto e1	
e ₁	nop	h
e ₂	nop	nop
e ₃	j	nop
	$r_z \Rightarrow r_{ctr-4}, r_{@z}$	
e ₄	goto L2	

3.

1. loadAl. Rarp -4 \Rightarrow ~~Vr1~~^{r3}

	r ₁	r ₂	r ₃
name	1	1	1 Vr ₁
next	∞	∞	5 5
Free	T	T	F F

~~r₃~~
r₂
r₁

5. add. ~~Vr1~~^{r3}, ~~Vr4~~^{r1} \Rightarrow ~~Vr5~~^{r1}

	r ₁	r ₂	r ₃
name	1 Vr1 ^{r3}	Vr ₂	1 1
next	5 5	9	5 ∞
free	F T	F	T T

~~r₁~~
r₁
r₃

2. loadAl. Rarp -8 \Rightarrow ~~Vr2~~^{r2}

	r ₁	r ₂	r ₃
name	1	1 Vr ₂	Vr ₁
next	∞	4 5	5
Free	T	F F	F

~~r₂~~
r₁

6. loadli 10 \Rightarrow ~~Vr6~~^{r1}

	r ₁	r ₂	r ₃
name	1 Vr ₆	Vr ₂	1
next	∞ 7	9	∞
free	T F	F	T

~~r₁~~
r₃

3. loadli 4 \Rightarrow ~~Vr3~~^{r1}

	r ₁	r ₂	r ₃
name	1 Vr ₃	Vr ₂	Vr ₁
next	∞ 4	4	5
Free	T F	F	F

~~r₁~~

7. store. ~~Vr6~~^{r1} \Rightarrow ~~Vr5~~^{r3}

	r ₁	r ₂	r ₃
name	1 Vr ₆	Vr ₂	1 Vr ₅
next	∞ 7	9	∞ 5
free	T F	F	T T

~~r₃~~
r₃
r₃

4. mult. ~~Vr2~~^{r2}, ~~Vr3~~^{r1} \Rightarrow ~~Vr4~~^{r1}

	r ₁	r ₂	r ₃
name	1 Vr ₃	Vr ₂	Vr ₁
next	5 5	9 9	5
Free	T F	F	F

~~r₁~~

Spill store Vr₁ \Rightarrow Rarp. 1h

8. loadAl Rarp -12 \Rightarrow ~~Vr7~~^{r3}

	r ₁	r ₂	r ₃
name	1	Vr ₂	1 Vr ₇
next	∞	9	∞ 9
free	T	F	T F

~~r₃~~
r₁

9. add $\overset{r_3}{VR8} \overset{r_2}{VR2} \Rightarrow \overset{r_2}{VR8}$

11. store $\overset{r_3}{VR3} \Rightarrow \overset{r_2}{VR9}$

	r_1	r_2	r_3
name	1	VR2 ^{VR8}	VR7 1
next	∞	VR10 10	VR0 00
free	T	ATF A T	AT T

~~r_2~~
 r_3
 r_1

	r_1	r_2	r_3
name	1	VR9 1	VR3 1
next	∞	VR0 00	VR 00
free	T	AT A T	AT T

r_2
 r_3
 r_1

10.

~~spin~~ first. 

Done

loadAl $\overset{r_3}{Var, -16} \Rightarrow \overset{r_3}{VR3}$

	r_1	r_2	r_3
name	1	VR8	VR3
next	∞	10	VR 11
free	T	F	AT F

~~r_2~~
 r_1

mult $\overset{r_2}{VR8}, \overset{r_3}{VR3} \Rightarrow \overset{r_2}{VR9}$

	r_1	r_2	r_3
name	1	VR8 ^{VR9}	VR3
next	∞	VR11 11	VR 11
free	T	ATF A F F	AT F

~~r_2~~
 r_1

4. a) live range

Do not include the def. instruction in the live range

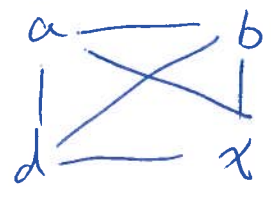
interference graph

a: (1-8, 9-10)

b: (2-10)

d: (5-6, 8-9)

x: (3-4, 6, 7-9)



b) spill cost

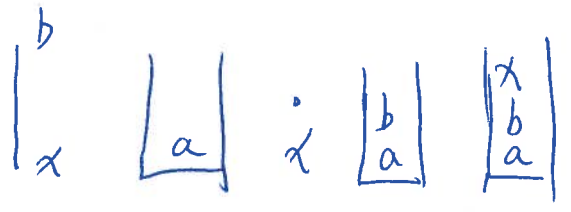
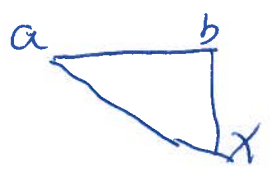
spill d = r4

a 7/3

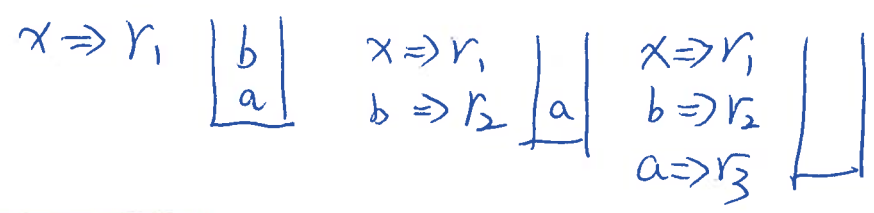
b 4/3

d 4/3

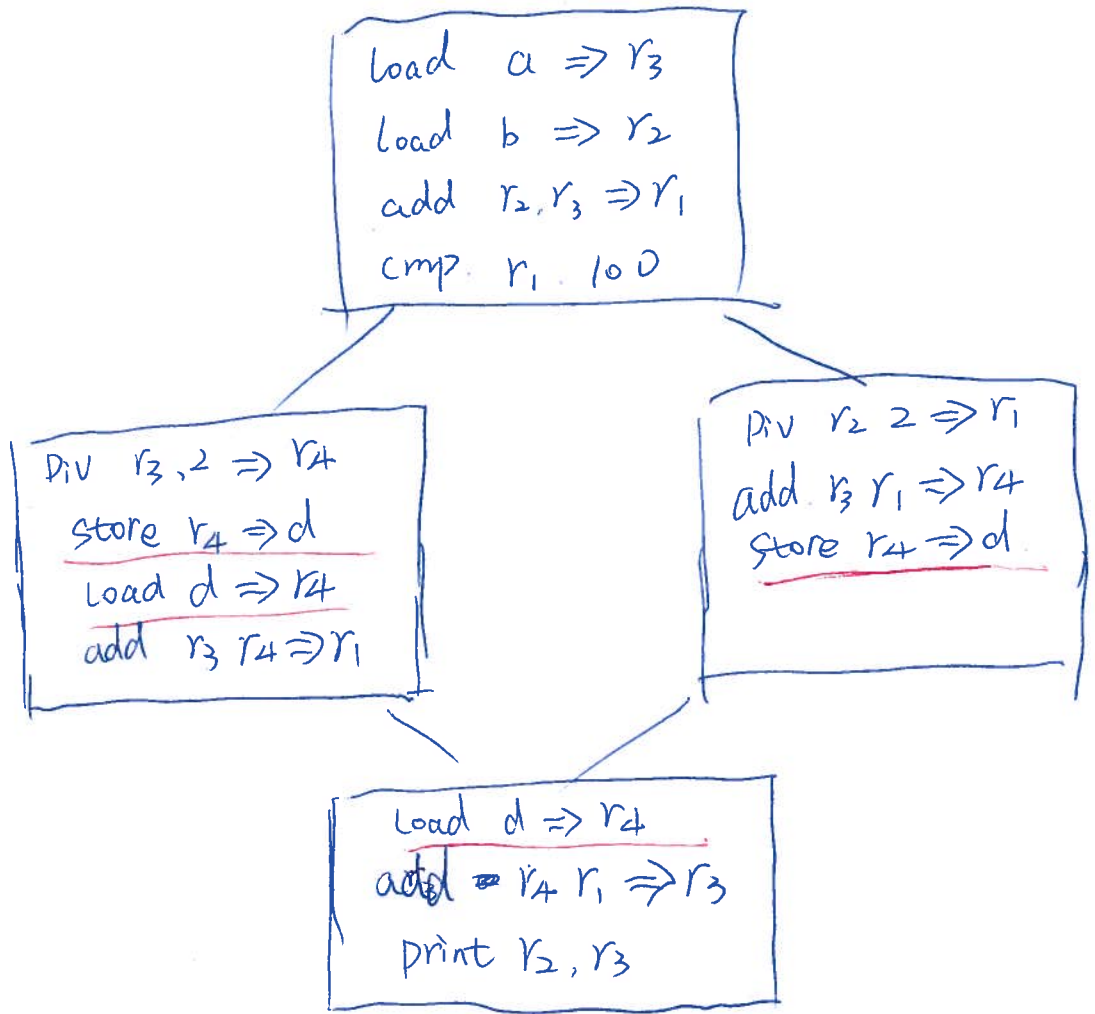
x 6/3



$b = d < x < a$



C.



d.

top down live range

$$a = \{B_1 \ B_2 \ B_3 \ B_4\}$$

$$b = \{B_1 \ B_3 \ B_4\}$$

$$d = \{B_2 \ B_3 \ B_4\}$$

$$x = \{B_1 \ B_2 \ B_3 \ B_4\}$$

