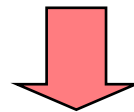




Rotating Register Files

Problems: Scheduled loops require lots of registers,
Lots of duplicated code in prolog, epilog

ld r1, ()		
add r2, r1, #1	ld r1, ()	
st r2, ()	add r2, r1, #1	ld r1, ()
	st r2, ()	add r2, r1, #1
		st r2, ()

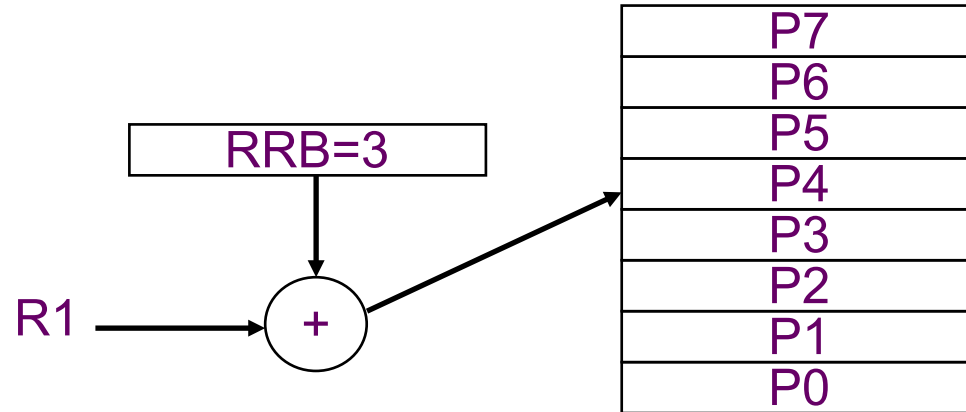


Prolog	ld r1, ()		
	ld r1, ()	add r2, r1, #1	
Loop	ld r1, ()	add r2, r1, #1	st r2, ()
		add r2, r1, #1	st r2, ()
Epilog			st r2, ()
			st r2, ()

Solution: Allocate new set of registers for each loop iteration



Rotating Register File



Rotating Register Base (RRB) register points to base of current register set. Value added on to logical register specifier to give physical register number. Usually, split into rotating and non-rotating registers.

Prolog	ld r1, ()			dec RRB
	ld r1, ()	add r3, r2, #1		dec RRB
Loop	ld r1, ()	add r3, r2, #1	st r4, ()	bloop
		add r2, r1, #1	st r4, ()	dec RRB
Epilog			st r4, ()	dec RRB

Loop closing branch decrements RRB



Rotating Register File (Previous Loop Example)

Three cycle load latency
encoded as difference of 3
in register specifier
number ($f4 - f1 = 3$)

Four cycle fadd latency
encoded as difference of 4
in register specifier
number ($f9 - f5 = 4$)

ld f1, ()	fadd f5, f4, ...	sd f9, ()	bloop
-----------	------------------	-----------	-------

ld P9, ()	fadd P13, P12,	sd P17, ()	bloop	RRB=8
ld P8, ()	fadd P12, P11,	sd P16, ()	bloop	RRB=7
ld P7, ()	fadd P11, P10,	sd P15, ()	bloop	RRB=6
ld P6, ()	fadd P10, P9,	sd P14, ()	bloop	RRB=5
ld P5, ()	fadd P9, P8,	sd P13, ()	bloop	RRB=4
ld P4, ()	fadd P8, P7,	sd P12, ()	bloop	RRB=3
ld P3, ()	fadd P7, P6,	sd P11, ()	bloop	RRB=2
ld P2, ()	fadd P6, P5,	sd P10, ()	bloop	RRB=1