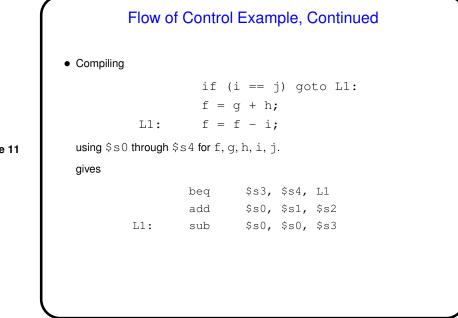
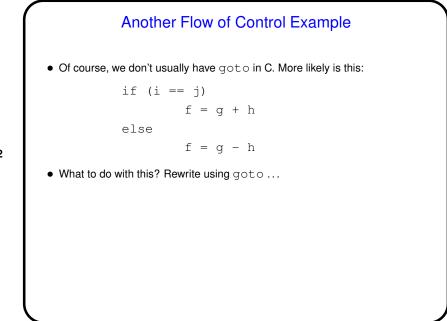
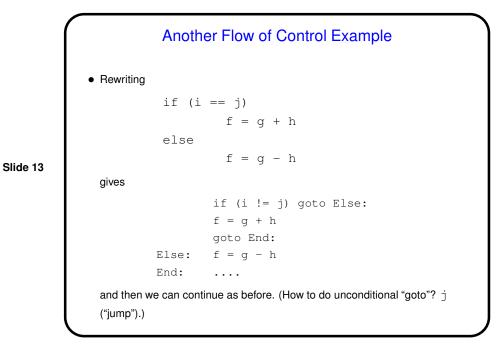


Slide 10





Slide 12



Do we have enough to do (some kinds of) loops? Yes — example:
Loop: g = g + A[i];
i = i + j;

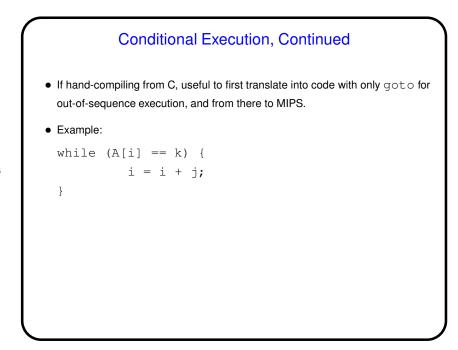
```
Slide 14
```

assuming we're using s1 through s4 for $g,\,h,\,i,\,j,$ and s5 for the address of A.

(This time we'll use sll rather than two adds to multiply i by 4.)

if (i != h) goto Loop:

	Loops — Example Continued								
	Result								
	Loop:	sll	\$t1,	\$s3,	2	#	\$t1 <- 4*i		
		add	\$t1,	\$t1,	\$s5	#	\$t1 <- & of A[i]		
		lw	\$t0,	0(\$t1)		#	\$t0 <- A[i]		
		add	\$s1,	\$s1,	\$t0	#	g = h + A[i]		
Slide 15		add	\$s3,	\$s3,	\$s4	#	i = i + j		
		bne	\$s3,	\$s2,	Loop	#	if (i!=h) goto Loop		



	•			-					l assuming \$s0 ha i, j, and k):
Loc		of 11 and 1	egiotore	, , , , , , , , , , , , , , , , , , ,	niougii ç	00		uve	±,), and x/.
	-	i] != k) goto	o End	:				
		sll	\$t0,	\$s1,	2	#	i	*	4
		add	\$t0,	\$s0,	\$t1	#	&	A[i	L]
		lw	\$t0,	0(\$t1	L)	#	A	[i]	
		bne	\$t0,	\$s3,	End				
#	i =	i + j							
		add	\$s1,	\$s1,	\$s2				
#	goto	Loop:							
		i	Loop						

More Flow of Control
• With what we have now we can do if/then/else and loops, but only if condition being tested is equals / not equals.
• So, we need instructions such as blt, ble, right?
 But those are apparently difficult to implement well; instead MIPS has "set on less than":
slt r1, r2, r3
which compares the contents of registers $\tt r2$ and $\tt r3$ and sets $\tt r1-1$ if $\tt r2$ is smaller, else 0.
• Example — compile the following C:
if (a < b) goto Less:
assuming we're using \$s0, \$s1 for a, b.



