Shift Instructions

- "Move" bits in a register
  - Left shift: Moves bits ***from a lower position to a higher position***
  - Right shift: Moves bits ***from a higher position to a lower position***
- Two directions: Left and Right
- Shift instructions

sll $t0,$t1,5	shift left logical
srl $t0,$t1,5	shift right logical
sra $t0,$t1,5	shift right arithmetic
sllv $t0,$t1,$t2	shift left logical value (register)
srlv $t0,$t1,$t2	shift right logical value (register)
srav $t0,$t1,$t2	shift right arithmetic value (register)

Shift left logical (sll)

- sll $t0,$t1,5 moves the bits left (lower to higher)

suppose $t1 is
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1010

Shift left logical (sll)

- sll $t0,$t1,**5** moves the bits left (lower to higher)

suppose $t1 is
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 **0 1010**

logical shift: ***fill in 0s*** when value moved to the left by number of positions

Shift left logical (sll)

- sll $t0,$t1,**5** moves the bits left (lower to higher)

suppose $t1 is
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 **0 1010**
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 00**01 0100 0000**

logical shift: ***fill in 0s*** when value moved to the left by number of positions

Shift left logical (sll)

- sll $t0,$t1,**5** moves the bits left (lower to higher)

suppose $t1 is
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1010
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0100 0000

logical shift: ***fill in 0s*** when value moved to the left by number of positions
what mathematical operation is this?

	$t1	**shift amount**	**$t0**
consider:	0001	1	0010
	0001	2	0100
	0001	3	1000

Shift left logical (sll)

- sll $t0,$t1,**5** moves the bits left (lower to higher)

suppose $t1 is
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 1010
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0100 0000

logical shift: ***fill in 0s*** when value moved to the left by number of positions
what mathematical operation is this?

	$t1	**shamt**	**$t0**	
consider:	0001	1	0010	***2**
	0001	2	0100	***4**
	0001	3	1000	***8**

shift left logical is multiplication by 2^{shamt}

Shift right logical (srl)

- srl $t0,$t1,**5** moves the bits right (higher to lower)

suppose $t1 is
1011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0101 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

logical shift: ***fill in 0s*** when value moved to the right by number of positions
what mathematical operation is this?

	$t1	**shamt**	**$t0**	
consider:	1000	1	0100	**/2**
	1000	2	0010	**/4**
	1000	3	0001	**/8**

shift right logical is division (integer) by 2^{shamt}

Shift right arithmetic (sra)

- srl $t0,$t1,**5** moves the bits right (higher to lower)

suppose $t1 is
1011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
1111 1101 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

0011 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0001 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

logical shift: ***fill in MSB value*** when value moved to the right by number of positions
what mathematical operation is this?
division by power of 2, preserving sign (neg/pos)

Shifts – value forms

- sllv, srlv, srav: shift left/right by a value in register
  - same operation as sll, srl, sra
- e.g., sllv $t0,$t1,$t2
  - shifts $t1 by value in $t2, placing result in $t0
  - what range is allowed for $t2? (i.e., min and max value?)
- what instruction format?
  - sll, srl, sra **R-format (shamt)**
  - sllv, srlv, srav **R-format**