Logical Bit Shifts [Andy Chong Sam]

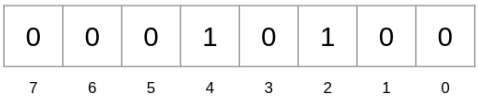
1 Logical Left Shift

(I) Consider the following line of Java code:

$$\text{int } i = 20 << 1;$$

The value of i is 40. A logical left shift by one doubles a number.

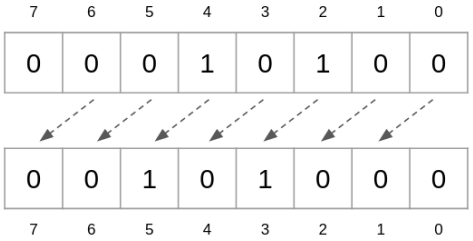
(II) In binary, 20 is represented as **10100**. The image below shows this representation along with the place numbers at the bottom:



We can confirm that this is correct since:

$$1(2)^4 + 1(2)^2 = 20$$

(III) If we do a left shift, the most significant bit is discarded and a 0 is placed in position zero:



The decimal value is now:

$$1(2)^5 + 1(2)^3 = 40$$

Let's see why this works.

(IV) Suppose that a decimal number is calculated from three binary digits. Let d be the binary digit. Let a, b, and c represent arbitrary bit positions. If x is the decimal representation of the number then:

$$x = d_a\,(2)^a + d_b\,(2)^b + d_c\,(2)^c$$

A left shift shifts the position of a, b, and c by one, so the new decimal y would be:

$$y = d_a\,(2)^{a+1} + d_b\,(2)^{b+1} + d_c\,(2)^{c+1}$$

By the power rule for products we can rewrite this as:

$$y = d_a\,(2^a)\,(2) + d_b\,(2^b)\,(2) + d_c\,(2^c)\,(2)$$

If we factor out the 2:

$$y = 2(\ d_a\,(2)^a + d_b\,(2)^b + d_c\,(2)^c\ )$$

We can see that $y = 2x$, so a left shift by one position has the effect of doubling a decimal number.

(V) Let's see what happens when we generalize this shift to n positions to the left:

$$y = d_a\,(2)^{a+n} + d_b\,(2)^{b+n} + d_c\,(2)^{c+n}$$

$$y = d_a\,(2^a)\,(2^n) + d_b\,(2^b)\,(2^n) + d_c\,(2^c)\,(2^n)$$

$$2^n(\ d_a\,(2)^a + d_b\,(2)^b + d_c\,(2)^c\ )$$

So a shift of n positions to the left is equivalent to multiplying the original number by 2^n.

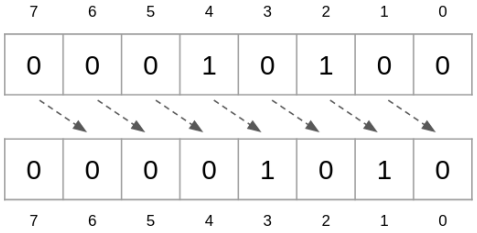
2 Logical Right Shift

(I) Now consider the following:

$$\text{int } j = 20 >>> 1;$$

The value of j is 10. A logical right shift by one halves a number.

In a right shift the least significant bit is discarded, and a zero is placed in the most significant bit.



We can see that the new value is 10:

$$1(2)^3 + 1(2)^1 = 10$$

Just like with the left shift, we can show how shifting right works mathematically. We'll use the same generic decimal number from the previous section:

$$x = d_a\,(2)^a + d_b\,(2)^b + d_c\,(2)^c$$

(II) A shift to the right would decrease the value of the exponents by one.

$$y = d_a\,(2)^{a-1} + d_b\,(2)^{b-1} + d_c\,(2)^{c-1}$$

Using the quotient rule:

$$y = d_a\,(2)^{a-1} + d_b\,(2)^{b-1} + d_c\,(2)^{c-1}$$

$$y = d_a\,\frac{2^a}{2} + d_b\,\frac{2^b}{2} + d_c\frac{2^c}{2}$$

If we factor out the $\frac{1}{2}$:

$$y = \frac{d_a\,(2)^a + d_b\,(2)^b + d_c\,(2)^c}{2}$$

Since $y = \frac{x}{2}$, a right shift by one position has the effect of halving a decimal number.

(III) Finally, let's generalize for n shifts to the right:

$$y = d_a\,(2)^{a-n} + d_b\,(2)^{b-n} + d_c\,(2)^{c-n}$$

$$y = d_a\,\frac{2^a}{2^n} + d_b\,\frac{2^b}{2^n} + d_c\frac{2^c}{2^n}$$

$$y = \frac{d_a\,(2)^a + d_b\,(2)^b + d_c\,(2)^c}{2^n}$$

So a shift of n to the right is equivalent to dividing the original number by 2^n.