

Storage multiple data in a word

We can store multiple items in a word assuming we have enough bits in the word to store the sum of all the bits in all the multiple items.

For example, let's say we want to store 2 integer values in one word. The first integer value ranges from 0 - 7, and the second integer value ranges 0 - 12. For our example let's assume we have a 32 bit word size. This is the number of bits that an *int* type would be allocated to us.

The first integer value, ranging from 0-7, would only need 3 bits. With 3 bits we can generate the values 0-7 exactly.

For the second integer value, ranging from 0-12, we would need 4 bits. With 4 bits we can generate the numbers 0-15, so this will cover our second integer.

We are going to store the first integer in the bits labeled 0 - 2, and second integer value in the bits labeled 3 - 6. Bits 7 - 31 will not be used (ah the waste...) Below is a picture of the word bit pattern in binary.

— — — • • — — — — — — — — — —
31 30 29 6 5 4 3 2 1 0

To place the second integer value in bits labeled 3-6, we need to move the value over 3 binary place values to the left, which is multiplying by 2^3 . Therefore to store the data:

```
wordValue = firstInt + 8 * secondInt;
```

To get the two values back out of the word, knowing the placement of them inside the word:

```
firstInt = wordValue % 8;  
secondInt = wordValue / 8;
```