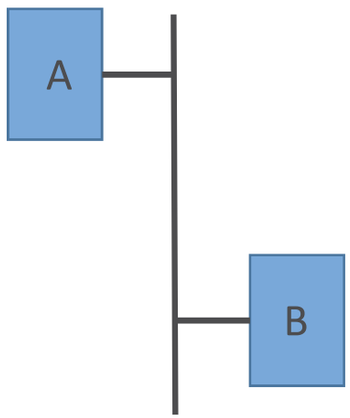
Data types

- **Nets** connect components and are continuously assigned values
  - wire is main net type (tri also used, and is identical)
- **Variables** store values between assignments
  - reg is main variable type
  - Also integer, real, time variables

- **Scalar** is a single value (usually one bit)
- **Vector** is a set of values of a given type
  - **reg** [7:0] v1,v2; //8-bit vectors, MSB is highest bit #
  - **wire** [1:4] v3; //4-bit vector, MSB is lowest bit #
  - **reg** [31:0] memory [0:127]; //array of 128 32-bit values
  - **{v1,v2}** // 16-bit vector: concatenate bits/vectors into larger vector

Logic values

- Logic values: 0, 1, x, z x = undefined state
  z = tri-state/floating/high impedance



Multiple drivers of one wire

wire	B: 0	1	x	z
A: 0	0	x	x	0
1	x	1	x	1
x	x	x	x	x
z	0	1	x	z

State of the net

Analagous to VHDL std_logic values
‘0’ ‘1’ ‘X’ ‘Z’

How to Express numbers ?

N’ Bxx

```
8’b0000_0001
```

- **(N) Number of bits**
  - Expresses how many bits will be used to store the value
- **(B) Base**
  - Can be b (binary), h (hexadecimal), d (decimal), o (octal)
- **(xx) Number**
  - The value expressed in base, apart from numbers it can also have X and Z as values.
  - Underscore _ can be used to improve readability

Number Representation in Verilog

Verilog	Stored Number	Verilog	Stored Number
4’b1001	1001	4’d5	0101
8’b1001	0000 1001	12’hFA3	1111 1010 0011
8’b0000_1001	0000 1001	8’o12	00 001 010
8’bxX0X1zZ1	XX0X 1ZZ1	4’h7	0111
‘b01	0000 .. 0001	12’h0	0000 0000 0000