Verilog assign statement

The assign statement where any wire or other similar wire like data-types can be driven continuously with a value.

Assign syntax :

```
[assign <net_expression>
      = [drive_strength] [delay]
           <expression of different signals or constant
              value>. ]
```

- Starts with keyword assign
- signal name which can be either a single signal or a concatenation of different signals
- drive strength and delay are optional and are mostly used for dataflow modeling than synthesizing into real flow.
- expression or signal on the right hand side is evaluated and assigned to net or expression of nets on the left hand side.

— Delay values are useful for specifying delays for gates and are used to model timing behaviour in real hardware because the values dicatate when the net should be assigned with the evaluated value.

Rules

- LHS should always be scalar or vector net or a concatenation of scalar or vector nets and never a scalar or vector register.
- RHS can contain scalar or vector registers and function calls.
- whenever any operand on the RHS changes in value, LHS will be updated with the new value
- assign statements are also called continuous assignmets and are always active

Example - used for writing logic gates.
(Md file me dal dunga)

- It is illegal to drive or assign reg type variables with an assign statement. This is because a reg variable is capable of storing data and does not require to be driven continuously. reg signals can only be driven in procedural blocks like initial and always.

Implicit continuous assignment:

```
wire [1:0] a;
 assign a = x & y; // explicit.
wire [1:0] a = x & y; // implicit
```