**I, Giới thiệu**  
**- Verilog**chỉ hỗ trợ những loại dữ liệu đã được định nghĩa trước. Những loại dữ liệu này bao gồm dữ liệu bit, mảng bit, vùng nhớ, số nguyên, số thực, sự kiện, và độ mạnh của dữ liệu. Những loại này định nghĩa trong phần lớn mô tả của **Verilog**.  
**- Verilog**chủ yếu xử lí trên bit và byte khi mô tả mạch điện tử. Loại số thực thì hữu dụng trong việc mô tả độ trì hoãn và định thời, nó cũng rất hữu dụng trong việc mô hình hóa ở mức cao như là phân tích xác suất kết nối mạch trong hệ thống và những giải thuật xử lí tín hiệu số. Loại dữ liệu phần cứng bao gồm ***net***và ***reg***. Thông thường những loại này có thể được xem như là dây kết nối và thanh ghi. Dữ liệu ***net***có thể được khai báo chi tiết hơn để tạo ra những loại dữ liệu khác như ***tri***-stated hay non-***tri***-stated và phụ thuộc vào các xử lí nhiều kết nối sẽ tạo ra những phép and, or hoặc dùng giá trị trước đó.  
  
**II. Net**  
- Mô tả kết nối dây dẫn trong một mạch điện và được dùng để kết nối các cổng hay các module. Giá trị của **Net** có thể đọc, nhưng không được gán trong hàm (function) hoặc khối (block). Một ***net***sẽ không lưu giữ giá trị của nó. Nó phải được điều khiển bởi một trong hai cách sau:  
+ Bằng việc kết nối ***net***đến ngõ ra của một cổng hay một module.  
+ Bằng việc gán một giá trị đến ***net***trong một phép gán nối tiếp.  
**WIRE** là một kiểu dữ liệu sử dụng phổ biến thuộc kiểu Net. ***wire***là một kiểu dữ liệu đơn giản để kết nối giữa liệu hai cổng,module hay trong phép gán nối tiếp….Ngoài ra còn có nhiều kiểu dữ liễu khác như wand, wor, tri….  
  
1. Cú pháp

MÃ:

Wire [msb:lsb] tên biến wire.

Wand [msb:lsb] tên biến wand.

Wor [msb:lsb] tên biến wor.

Tri [msb:lsb] tên biến tri.

2. Ví dụ

MÃ:

Wire c;

Wand d;

Assign d= a; // d có giá trị là a

Assign d= a&b; // giá trị d là mức logic của phép And a và b.

Wire [9:0] A; // vectơ A có 10 bit.

**III, REG**  
- Khai báo ***reg***được thực hiện cho tất cả những tín hiệu mà được điều khiển từ những mô tả hành vi. Loại dữ liệu ***reg***lưu giữ một giá trị được cho đến khi nó được gán một giá trị mới trong một mô tả tuần tự (khối ***initial***hoặc ***always***). Có thể hiểu nôm na **Reg** là một kiểu **net** có khả năng lưu trữ giá trị và có thể xem là một register trong phần cứng.  
  
**1,Cú pháp:  
Reg [msb:lsb]** tên biến reg.  
  
**2,Ví dụ:**

MÃ:

Reg a; // biến thanh ghi đơn giản 1 bit.

Reg [7:0] A; // một thanh ghi A 8bit.

Reg [5:0]b, c; // hai biến thanh ghi 6 bit.

**IV. Khai báo port**  
- Những từ khoá Input, Output, Inout biểu thì đầu vào, đầu ra, và port hai chiều của một module hoặc task. Một port đầu ra có thể được cấu hình từ các dạng: wire, reg, wand, wor, hoặc tri. Mặc định là wire. Kiểu reg chỉ xuất hiện ở cổng Output hay phét gán nối tiếp có nhớ.  
  
1,Cú pháp:

MÃ:

Input [msb:lsb] port đầu vào.

Output [msb:lsb] port đầu ra.

Inout [msb:lsb] port đầu vào,ra hai chiều.

2,Ví dụ:

MÃ:

Module sample (b, e, c, a);

Input a; // một đầu vào mặc định là kiểu wire.

Output b, e; // hai đầu ra mặc định là kiểu wire.

Output [1:0] c; // đầu ra c 2 bit

Output Reg [1:0] c; // đầu ra c được khai báo như một reg 2 bit.

**V. Integer (Số nguyên)**  
- Integer là một biến đa năng. Trong tổng hợp chúng được dùng chủ yếu cho vòng lặp, tham số, và hằng số. Chúng hoàn toàn là reg. Tuy nhiên chúng chứa dữ liệu bằng những số có dấu, trong khi đó khai báo dạng reg chứa chung bằng số không dấu. Nếu chúng chứa những số mà không định nghĩa thời gian biên dịch thì kích thước mặc định là 32 bit. Nếu chúng chứa hằng, sự tổng hợp điều chỉnh các số có kích thước nhỏ nhất cần thiết cho sự biên dịch.  
  
1. Cú pháp:   
**Integer** tên biến nguyên;  
…tên hằng nguyên…;  
  
2. Ví dụ:  
Integer a; // số nguyên đơn giản 32bit.  
Assign b= 63; // mặc định là một biến 7 bit  
  
**VI. Supply 0, Supply1**  
Xác định chổ đường dẫn lên mức logic 0 ( đất), logic 1( nguồn) theo thứ tự định sẵn.  
  
**VII. Time**  
- Time là một lượng 64 bit mà được sử dụng cùng với $time, hệ thống thao tác chứa lượng thời gian mô phỏng. Time không được hỗ trợ tổng hợp và và thế chỉ được dùng trong mục đích mô phỏng.  
  
1,Cú pháp:  
**Time**biến time;  
  
2,Ví dụ:  
**Time c;  
c = $time;** // c = thời gian mô phỏng.  
  
**VIII. Parameter (Tham số)**  
- Trong **Verilog**HDL, loại dữ liệu tham số (***parameter***) không thuộc loại dữ liệu biến (variables: ***reg***, ***integer***, ***time***, ***real***, ***realtime***) cũng như loại dữ liệu ***net***. Dữ liệu tham số không phải là biến mà chúng là hằng số. Có hai loại tham số trong **Verilog**đó là:  
++ Tham số module (module parameter): ***parameter***và ***localparam.***  
++ Tham số đặc tả (specify parameter): ***specparam.***  
  
- Cả hai loại tham số trên đều được phép khai báo độ rộng. Mặc định, ***parameter***và  
***specparam***sẽ có độ rộng đủ để chứa giá trị của hằng số, ngoại trừ khi tham số đó có khai báo độ rộng. Việc khai báo trùng tên giữa net , biến hay tham số là không được phép.  
  
1,Cú pháp:  
**Parameter** par\_1= giá trị, par\_2= giá trị, …;  
**Parameter**[msb:lsb] par\_3 = giá trị;  
  
2,Ví dụ:  
**Parameter add = 2b’00, sub = 3b’111;  
Parameter n = 4;  
Parameter [3:0] par\_2 = 4b’1010;**