## 1. Behavioral and Data Flow Designs

* Assign statements can never be inside the always block
* If we write data flow design using assign statement then outputs have to be declared as wires
* If we write behavioral design using always block then output has to be defined as reg as the output value needs to be held continuously.
* Behavioral

module mux2x1(i0, i1, sel, y);  
 input i0, i1, sel;  
 output reg y; // reg stands for register, because the register has to hold the value till the output changes, this occurs only when i0, i1 or sel changes (in always block)  
   
 // Behavioral implementation of mux2x1  
 // anytime i0, i1, or sel changes  
 always @ (i0 or i1 or sel) begin   
 // If sel=0, y should be i0, else y should be i1  
 if (sel == 0) y = i0;  
 else y = i1;  
 end  
   
   
endmodule

Data Flow

module mux2x1(i0, i1, sel, y);  
 input i0, i1, sel;  
 output y; // reg is not required here because assign is also "continuous" assignment  
 // even wire is not required because by default every variable is a wire  
   
 // Dataflow implementation of mux2x1  
 assign y = (sel == 1) ? i1: i0; // Whenever you use assign it is called data flow type of implementation  
 // assign y = (i0 & ~sel) | (i1 & sel);  
   
endmodule

## 2. Concatenation of Signals

