

## Module Instantiation

```
// Define the top-level module called ripple carry
// counter. It instantiates 4 T-flipflops. Interconnections are
// shown in Section 2.2, 4-bit Ripple Carry Counter.
module ripple_carry_counter(q, clk, reset);
```

```
    output [3:0] q; //I/O signals and vector declarations
                    //will be explained later.
    input clk, reset; //I/O signals will be explained later.
```

```
    //Four instances of the module T_FF are created. Each has a unique
    //name. Each instance is passed a set of signals. Notice, that
    //each instance is a copy of the module T_FF.
```

```
    T_FF tff0(q[0], clk, reset);
```

```
    T_FF tff1(q[1], q[0], reset);
```

```
    T_FF tff2(q[2], q[1], reset);
```

```
    T_FF tff3(q[3], q[2], reset);
```

```
endmodule
```

```
// Define the module T_FF. It instantiates a D-flipflop. We assumed
// that module D-flipflop is defined elsewhere in the design. Refer
// to Figure 2-4 for interconnections.
module T_FF(q, clk, reset);
```

```
    //Declarations to be explained later
    output q;
    input clk, reset;
    wire d;
```

```
    D_FF dff0(q, d, clk, reset); // Instantiate D_FF. Call it dff0.
    not n1(d, q); // not gate is a Verilog primitive. Explained later.
```

```
endmodule
```

instan.

4 instances

template