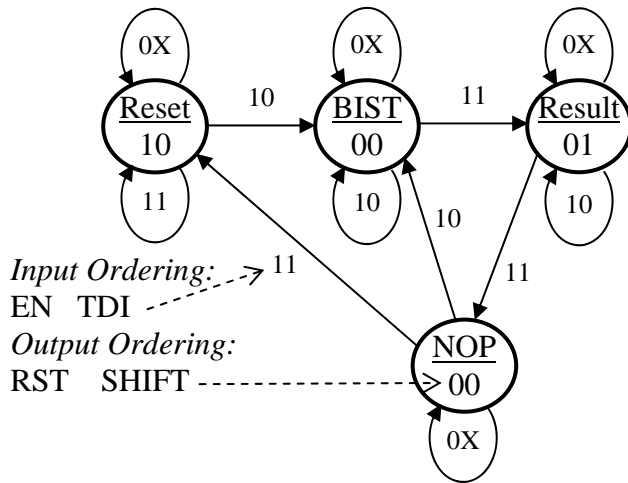


## FSMs in VHDL Using Enumeration Data Types

Consider a falling edge triggered FSM that performs the following state diagram functionality. We want to write a VHDL model using enumeration data types.



This FSM is a Moore machine since its outputs (RST and SHIFT) are a function of the current state only.

The four states in the state diagram have not been assigned and are represented by the variable names: Reset, BIST, Result, and NOP (all are valid VHDL names).

Since the FSM is falling edge triggered, all state transitions occur on the falling edge of the clock (which we will assume is named CLK).

Finally, recall that a signal will retain its value when not explicitly specified. As a result, we only need to address changes in states in our model.

```

entity FSM is
    port (CLK, EN, TDI: in bit;
          RST, SHIFT: out bit);
end entity FSM;
    
```

```

architecture RTL of FSM is
    type STATES is (Reset, BIST, Result, NOP);
    signal CS: STATES; -- current state
begin
    SYNC: process (CLK) begin
        if (CLK'event and CLK='0') then
            if (EN = '1') then
                if (CS = Reset) then
                    if (TDI='0') then CS <= BIST;
                    end if;
                elsif (CS = BIST) then
                    if (TDI='1') then CS <= Result;
                    end if;
                elsif (CS = Result) then
                    if (TDI='1') then CS <= NOP;
                    end if;
                elsif (CS = NOP) then
                    if (TDI='0') then CS <= BIST;
                    else CS <= Reset;
                    end if;
                end if;
            end if;
        end if;
    end process SYNC;
    
```

```

COMB: process (CS) begin
    if (CS = Reset) then
        RST <= '1';
        SHIFT <= '0';
    elsif (CS = Result) then
        RST <= '0';
        SHIFT <= '1';
    else
        RST <= '0';
        SHIFT <= '0';
    end if;
end process COMB;
end architecture RTL;
    
```