FSMD Design

- **Combinational logic**
- **State register**
- **Controller**
- **Datapath**

**State Transition Diagram:**
- **Off** transitions to **On1** when **x=0** and **b'**.
- **On1** transitions to **On2** when **x=1**.
- **On2** transitions to **On3** when **x=1**.
- **On3** transitions back to **Off** when **b'**.

**Countdown Logic:**
- **cnt=2** when **x=0**.
- **cnt=cnt-1** when **x=1**.
- **cnt=0** when **cnt=0'**.

**FSMD Diagram:**
- **Inputs:** **b**, **x**
- **Outputs:** **x**
- **Controller** receives **b** and **x**.
- **Datapath** processes data and sends **x**.
ARCHITECTURE Timer_Beh OF Timer IS

TYPE Statetype IS (S_Off, S_On);
SIGNAL State: Statetype;
SIGNAL Cnt: std_logic_vector(1 DOWNTO 0);
BEGIN
PROCESS (Clk)
BEGIN
IF (Clk = '1' AND Clk'EVENT) THEN
IF (Rst = '1') THEN
State <= S_Off;
Cnt <= "00";
ELSE
CASE State IS
WHEN S_Off =>
x <= '0';
Cnt <= "10";
IF (b = '0') THEN
State <= S_Off;
ELSE
State <= S_On;
END IF;
END CASE;
END IF;
END IF;
END PROCESS;
END Timer_Beh;
ARCHITECTURE Timer_Beh2 OF Timer IS
  TYPE Statetype IS (S_Off, S_On);
  SIGNAL Currstate, Nextstate: Statetype;
  SIGNAL Cnt,Cnt_Next:std_logic_vector(1 DOWNTO 0);
BEGIN
  PROCESS (Clk, Currstate, b)
  BEGIN
    CASE Currstate IS
      WHEN S_Off =>
        x <= '0';
        Cnt_Next <= "10";
        IF (b = '0') THEN
          Nextstate <= S_Off;
        ELSE
          Nextstate <= S_On;
        END IF;
      WHEN S_On =>
        x <= '1';
        Cnt_Next <= Cnt - "01";
        IF (Cnt = "00") THEN
          Nextstate <= S_Off;
        ELSE
          Nextstate <= S_On;
        END IF;
    END CASE;
    IF (Clk = '1' AND Clk'EVENT) THEN
      IF (Rst = '1') THEN
        Currstate <= S_Off;
        Cnt <= "00";
      ELSE
        Currstate <= Nextstate;
        Cnt <= Cnt_Next;
      END IF;
    END IF;
  END PROCESS;
END Timer_Beh2;