```
entity MUX IS
    PORT(
        A : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
        B : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
        C : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
        D : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
        SEL : IN STD_LOGIC_VECTOR (1 DOWNTO 0);
        Y : OUT STD_LOGIC_VECTOR (3 DOWNTO 0)
    );
END MUX;

architecture BEHAVE1 of MUX_CASE is
begin

    process1(A,B,C,D,SEL)
    begin
        case SEL is
            when "00" =>
                Y <= A;
            when "01" =>
                Y <= B;
            when "10" =>
                Y <= C;
            when other =>
                Y <= D;
        end case;

    end process1;
end BEHAVE1;


architechture BEHAVE2 of MUX_IF is
begin

    process2(A,B,C,D,SEL)
    begin
        IF (SEL = "00") then
            Y <= A;
        ELSIF (SEL = "01") then
            Y <= B;
        ELSIF (SEL = "10") then
            Y <= C;
        ELSE
            Y <= D;
        END IF;
    end process2;
end BEHAVE2;

architecture BEHAVE3 of MUX_ELSE is
begin

    process3(A,B,C,D,SEL)
```

```
    begin
        Y <= A when (SEL = "00") else
             B when (SEL = "00") else
             C when (SEL = "00") else
             D;
    end process3;
end BEHAVE3;

architecture BEHAVE4 of MUX_IF is
begin

    process4(A,B,C,D,SEL)
    begin
        if SEL = "00" then
            Y <= A;
        elsif SEL = "01" then
            Y <= B;
        elsif SEL = "10" then
            Y <= C;
        else
            Y <=D;
        end if;
    end process4;
end BEHAVE4;
```