```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

--This is the VHDL for a 4 to 1 mux. S1 and S0 will select which of the
--4 inputs get passed to output Y
--The mux will be implemented using 2 concurrent (selected signal and
--conditional assignments) and 2 sequential (case and if/then/else)VHDL
--structures

ENTITY mux4to1 IS
    PORT (A, B, C, D  : IN STD_LOGIC;
          S1, S0      : IN STD_LOGIC;
           Y_sel       : OUT STD_LOGIC;
           Y_cond      : OUT STD_LOGIC;
           Y_case      : OUT STD_LOGIC;
           Y_if        : OUT STD_LOGIC);
END mux4to1;

ARCHITECTURE behave OF mux4to1 IS
    SIGNAL selects : STD_LOGIC_VECTOR(1 DOWNTO 0);  --vector for the select signals
BEGIN
    selects <= S1 & S0;  --concatenate two signals into a bus

--This is the selected signal assignment
    WITH selects SELECT
        Y_sel <= A WHEN "00",
                 B WHEN "01",
                 C WHEN "10",
                 D WHEN OTHERS;

--This is the conditional signal assignment
        Y_cond <= A WHEN selects = "00" ELSE
                  B WHEN selects = "01" ELSE
                  C WHEN selects = "10" ELSE
                  D;

--This is the case. Case requires a process
    PROCESS (selects, A, B, C, D ) is  --sensitivity list should contain all signals
    that are read in process
        BEGIN
            CASE selects IS
                WHEN "00"   => Y_case <= A;
                WHEN "01"   => Y_case <= B;
                WHEN "10"   => Y_case <= C;
                WHEN OTHERS => Y_case <= D;
            END CASE;
        END PROCESS;

--This is the if/then/else.  IF requires a process
    PROCESS (selects, A, B, C, D ) is  --sensitivity list should contain all signals
    that are read in process
        BEGIN
            IF    (selects = "00") THEN Y_if <= A;
            ELSIF (selects = "01") THEN Y_if <= B;
            ELSIF (selects = "10") THEN Y_if <= C;
            ELSE   Y_if <= D;
            END IF;
        END PROCESS;
END behave;
```