Mux:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

  ENTITY mux IS
    PORT ( w0, w1, w2, w3 : IN STD_LOGIC ;
    s : IN STD_LOGIC_VECTOR(1 DOWNTO 0) ;
    f : OUT STD_LOGIC ) ;
END mux ;

ARCHITECTURE Behavior OF mux IS
BEGIN
  WITH s SELECT
    f <= w0 WHEN "00",
       w1 WHEN "01",
       w2 WHEN "10",
       w3 WHEN OTHERS ;
END Behavior ;

LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
PACKAGE mux_package IS
  COMPONENT mux
    PORT ( w0, w1, w2, w3 : IN STD_LOGIC ;
    s : IN STD_LOGIC_VECTOR(1 DOWNTO 0) ;
    f : OUT STD_LOGIC ) ;
  END COMPONENT ;
END mux_package ;
```

Decode:

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY decode IS
   PORT ( w : IN STD_LOGIC_VECTOR(1 DOWNTO 0) ;
         En : IN STD_LOGIC ;
          y : OUT STD_LOGIC_VECTOR(0 TO 3) ) ;
END decode ;

ARCHITECTURE Behavior OF decode IS
   SIGNAL Enw : STD_LOGIC_VECTOR(2 DOWNTO 0) ;
BEGIN
   Enw <= En & w ;
   WITH Enw SELECT
   y <= "1000" WHEN "100",
         "0100" WHEN "101",
         "0010" WHEN "110",
         "0001" WHEN "111",
         "0000" WHEN OTHERS ;
END Behavior ;
```

Encod:

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY encod IS
   PORT ( w : IN STD_LOGIC_VECTOR(3 DOWNTO 0) ;
          y : OUT STD_LOGIC_VECTOR(1 DOWNTO 0) ;
          z : OUT STD_LOGIC ) ;
END encod ;

ARCHITECTURE Behavior OF encod IS
BEGIN
   PROCESS ( w )
   BEGIN
   y <= "00" ;
   IF w(1) = '1' THEN y <= "01" ; END IF ;
   IF w(2) = '1' THEN y <= "10" ; END IF ;
   IF w(3) = '1' THEN y <= "11" ; END IF ;

   z <= '1' ;
   IF w = "0000" THEN z <= '0' ; END IF ;
   END PROCESS ;
END Behavior ;
```

Johns:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY johns IS PORT(Clrn, E, Clkn
          : IN STD_LOGIC;
          --clrn is your reset button STUDENT_ID
          : out std_logic_vector(3 downto 0);
          Q
          : OUT STD_LOGIC_VECTOR(0 TO 2));
END johns;
ARCHITECTURE Behavior OF johns IS
signal Qreg : STD_LOGIC_VECTOR (0 TO 2);
BEGIN
  PROCESS (Clrn, Clkn)
   BEGIN
    IF Clrn = '0' THEN
      Qreg <= "000";
    ELSIF (Clkn'EVENT AND Clkn = '0') THEN
      IF E = '1' THEN
      Qreg(1) <= Qreg(0);
      Qreg(2) <= Qreg(1);
      Qreg(0) <= NOT Qreg(2);
    ELSE
      Qreg <= Qreg;
    END IF;
  END IF;
-- STUDENT_ID variable represents the last 6 digits of your student ID
hence d4 is the fourth digit of your
--student ID in four bits, d5 is the fifth and so on. For example, for
```

Student ID 501 137659,

```
--d4 is 0100, d5 is 0011 and so on
 CASE Qreg IS
        WHEN "000" =>
        STUDENT_ID <= "0001"; --d1
        WHEN "100" =>
        STUDENT_ID <= "0011"; --d2
        WHEN "110" =>
        STUDENT_ID <= "0111";--d3
        WHEN "111" =>
        STUDENT_ID <= "0110";--d4
        WHEN "010" =>
        STUDENT_ID <= "0101";--d5
        WHEN "011" =>
        STUDENT_ID <= "1001";--d6
            WHEN OTHERS => STUDENT_ID <= "----";--error
        END CASE;
END PROCESS;
Q <= Qreg;
END Behavior;
```

State	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1