THREE COLOR CHANGING DEVICE USING ACCELEROMETER

TIA SHERRARD

RYAN TOOHEY

ALEX DIETRICH

NATE ABRAHAM

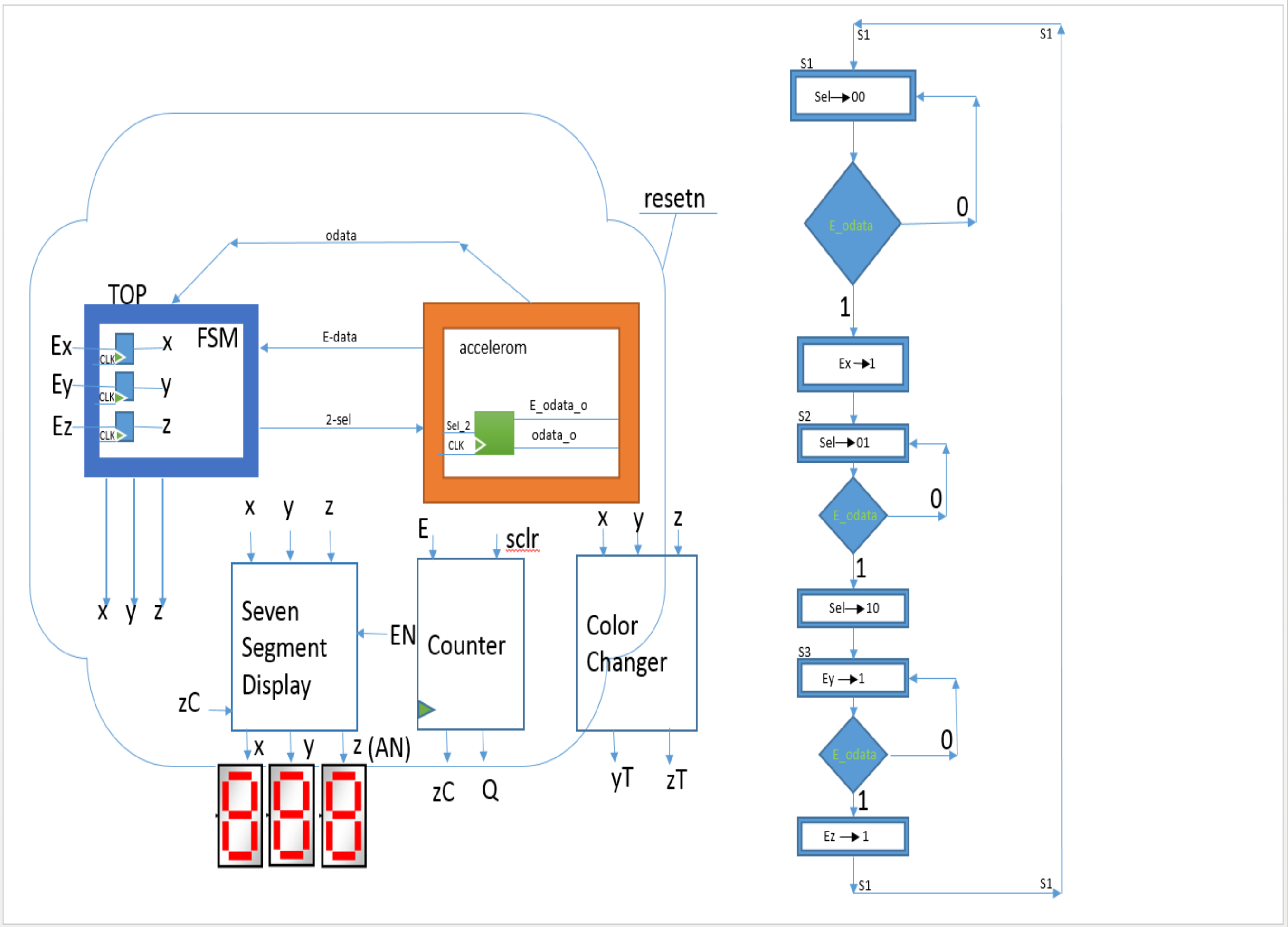
FINAL PROJECT: ECE 278

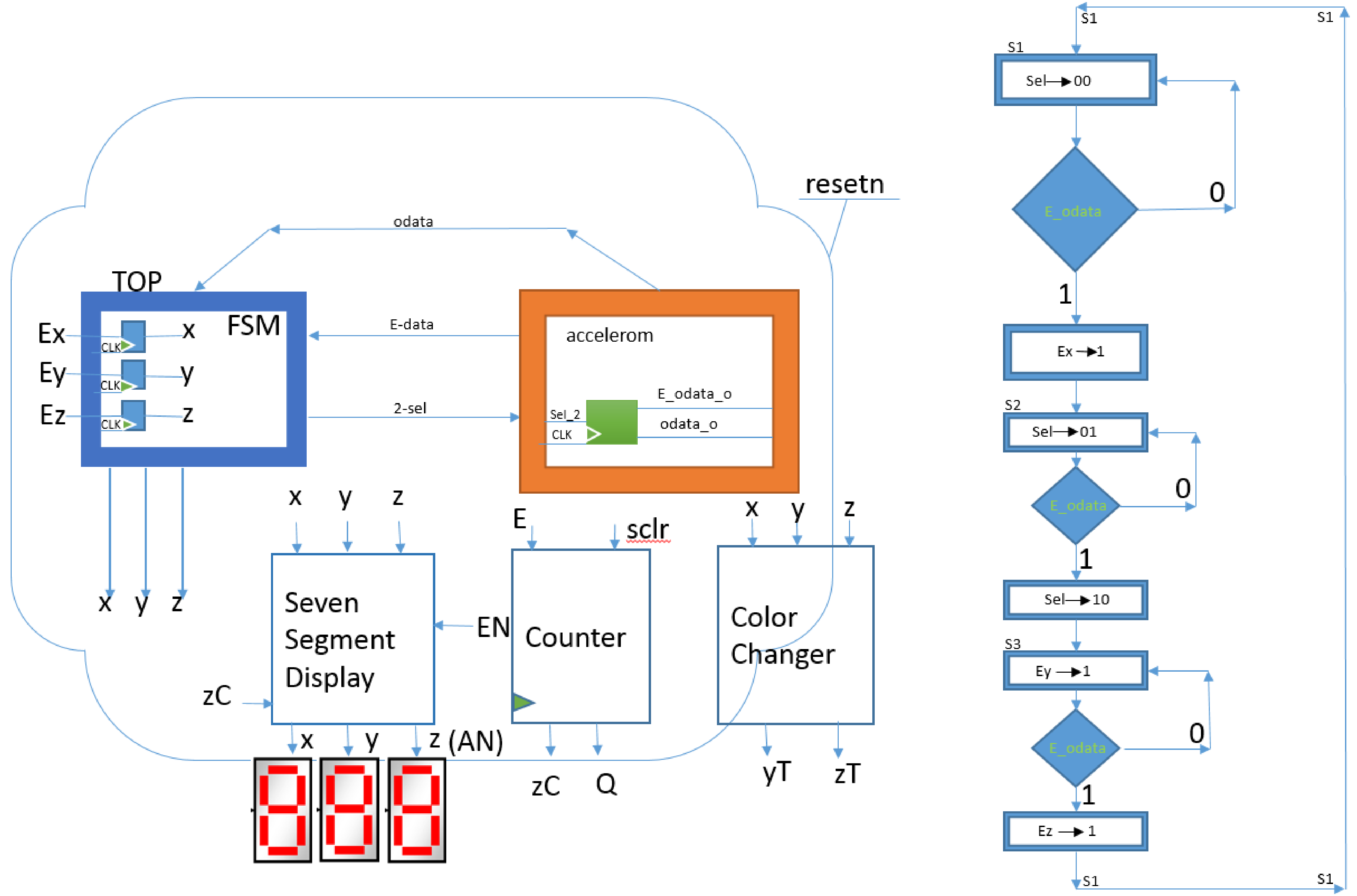
INSTRUCTOR: DANIEL LLAMOCCA (PH.D.)

THE CAPTURE TO THE RIGHT REPRESENTS THE DESIGN OF THE DEVICE. ALSO, TO THE RIGHT OF THE CAPTURE IS THE BLOCK DIAGRAM WHICH CHARACTERIZES THE MAIN STATE MACHINE (FSM).

The components used include:

- The accelerometer
- 3 registers for 8 bit data x, y, and z.
- Seven Segment Display (with a state machine inside of module).
- Counter
- Color Changing Module.





odata
resetn
TOP
FSM
Ex
Ey
Ez
CLK
x
y
z
E-data
2-sel
accelerom
E_odata_o
odata_o
Sel_2
CLK
x y z
Seven Segment Display
EN
zC
x y z (AN)
E
sclr
Counter
zC Q
x y z
Color Changer
yT zT
S1
Sel→00
E_odata
0
1
Ex →1
S2
Sel→01
E_odata
0
1
Sel→10
S3
Ey →1
E_odata
0
1
Ez → 1
S1

THE NEXYS 4 DDR HAS AN ANALOG DEVICE ADXL362 ACCELEROMETER BUILT IN. THIS DEVICE USES (2 X 10^(-6))A AT 100HZ. THIS DEVICE BEHAVES AS A SLAVE DEVICE THAT USES THE SPI INTERFACE TO COMMUNICATE BETWEEN THE USER (OR MASTER) AND THE ACCELEROMETER (OR SLAVE).

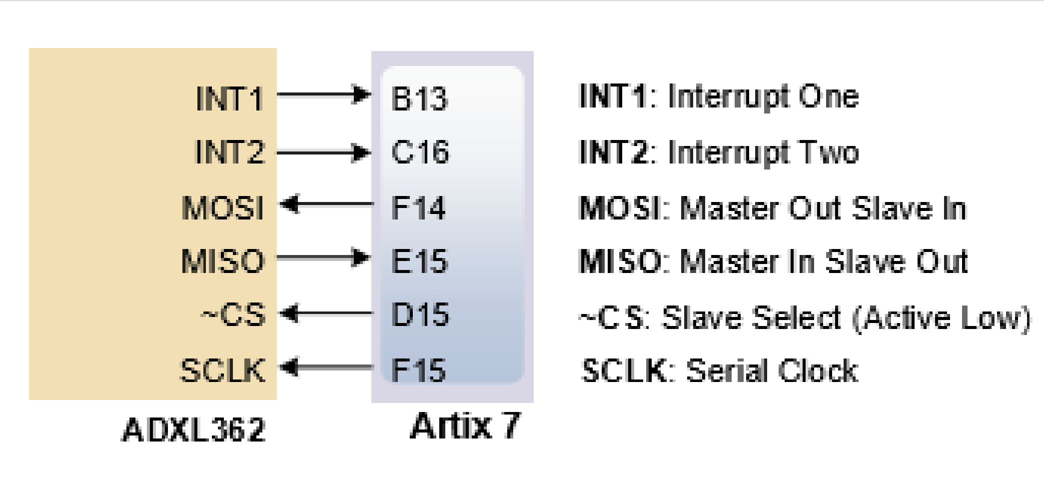


Figure 23. Accelerometer interface.

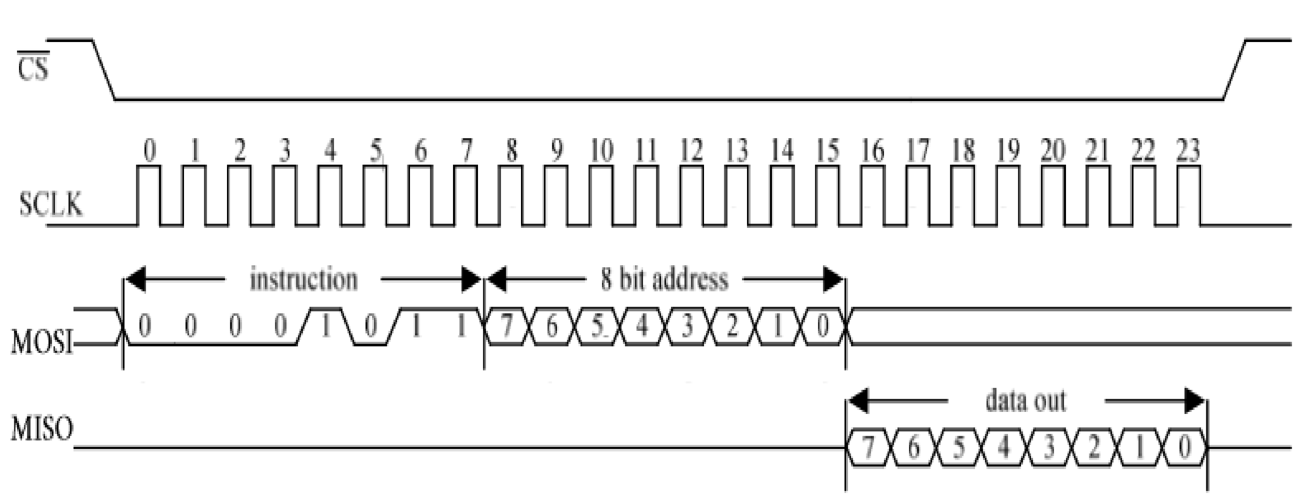
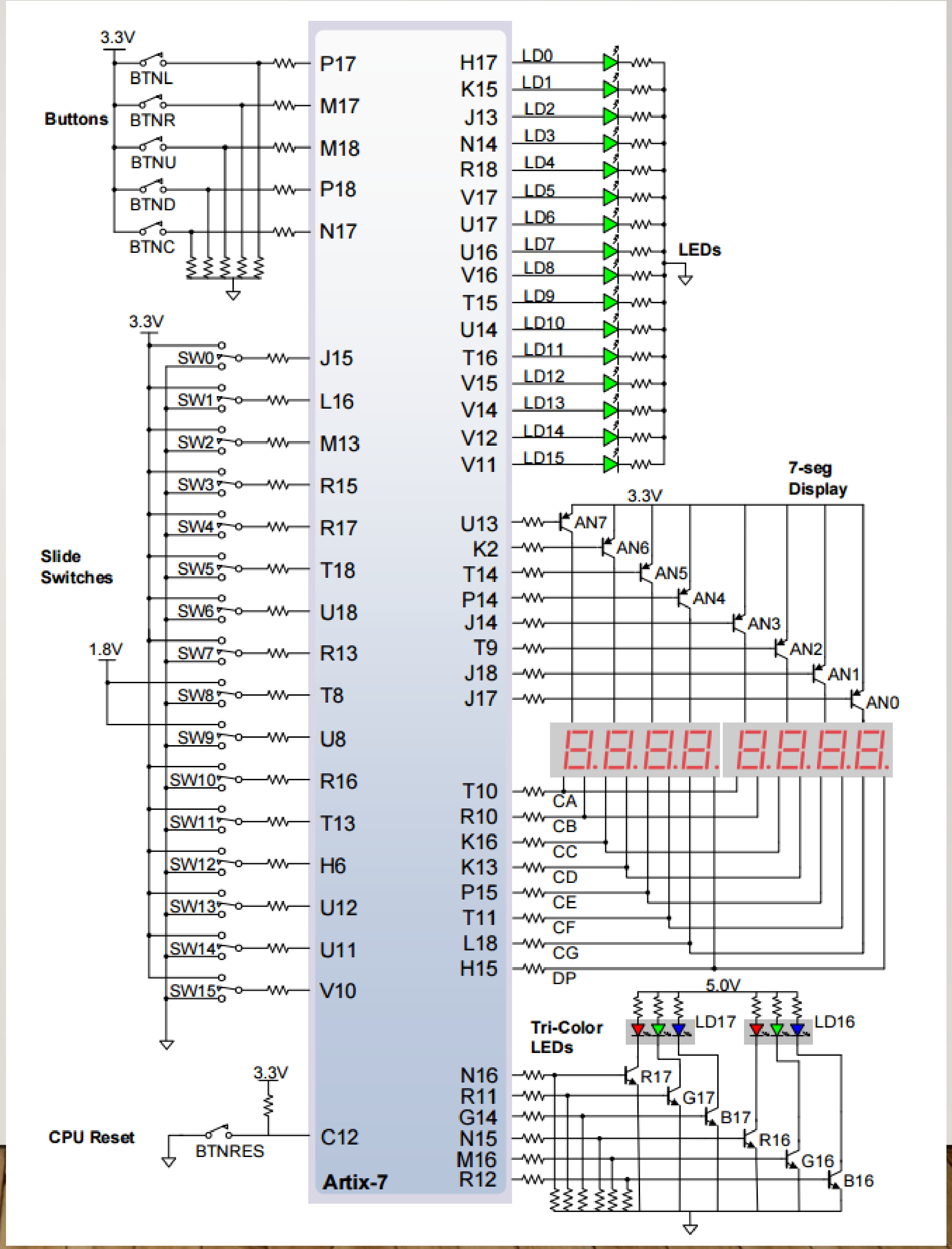


Figure 7. Register Read

NEXYS4 FPGA INPUTS AND OUTPUTS

The Nexys4 DDR board includes two tri-color LEDs, and an eight-digit seven-segment display which are controlled by the SevenSeg and ColorChanger modules.



SEVEN SEGMENT DISPLAY.

This if statement is used to change the led lights on the 7 segment display according to what range the data from mux signal is in. The numbers are represented in binary(8 bits), but the integer range is 0 to 255.

```
entity SevenSeg is
    Port ( clock, resetn : in STD_LOGIC;
            x, y, z : in STD_LOGIC_VECTOR (7 downto 0);
              zC: in std_logic;
           seg : out  STD_LOGIC_VECTOR (6 downto 0);
              AN : out STD_LOGIC_VECTOR (7 downto 0);
              EN: in std_logic);
end SevenSeg;

architecture Behavioral of SevenSeg is
signal leds: std_logic_vector (6 downto 0 );
signal mux: std_logic_vector (7 downto 0);

type state is (S1, S2, S3);
signal currentState: state;

begin

process (mux)

begin

-- | a | b | c | d | e | f | g |
-- |leds6|leds5|leds4|leds3|leds2|leds1|leds0|


    if (mux >= "00000000" and mux < "00010000")  then
     leds <=  "1000111" ;

    elsif (mux >= "00010000" and mux < "00100000") then
     leds <=   "1001111";

    elsif (mux >= "00100000" and mux < "00110000") then
     leds <= "0111101";

    elsif (mux >= "00110000" and mux < "01000000") then
     leds <= "1001110";

    elsif (mux >= "01000000" and mux < "01010000") then
     leds <= "0011111";

    elsif (mux >= "01010000" and mux < "01100000") then
     leds <= "1110111";

    elsif (mux >= "01100000" and mux < "01110000") then
```

SEVEN SEGMENT DISPLAY STATE MACHINE.

This FSM within the seven segment display, sets the mux signal to the input values x, y, and z. It also sets the location of the display (AN).

```
Transitions: process (resetn, clock)
    begin
        if resetn = '0' then
            currentState <= S1;

         elsif (clock'event and clock = '1') then
            if zC = '1' then
                case currentState is
                    when S1 =>
                        currentState <= S2;
                    when S2 =>
                        currentState <= S3;
                    when S3 =>
                        currentState <= S1;
                end case;
            end if;
        end if;

    end process;

    Outputs: process (currentState,x,y,z)
    begin
    mux <= "00000000";
    case currentState is

        when S1 =>
            mux <= x;
            AN <= "11111011";
        when S2 =>
            mux <= y;
            AN <= "11111101";
        when S3 =>
            mux <= z;
            AN <= "11111110";
    end case;


    end process;

with EN select
seg <= not leds when '1',
"1111111" when others;

--SevenSegDisp <= not(leds);
```

COLOR CHANGING MODULE.

8 bit numbers in Hex, and the Colors that mark the different positions:

X = C, Color = Green

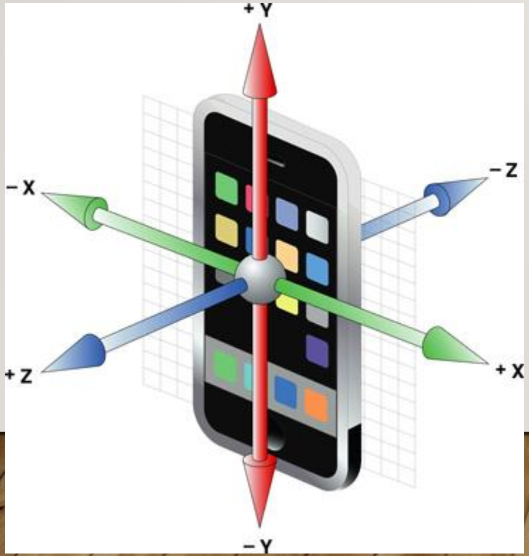
Y = 4, Color = Red

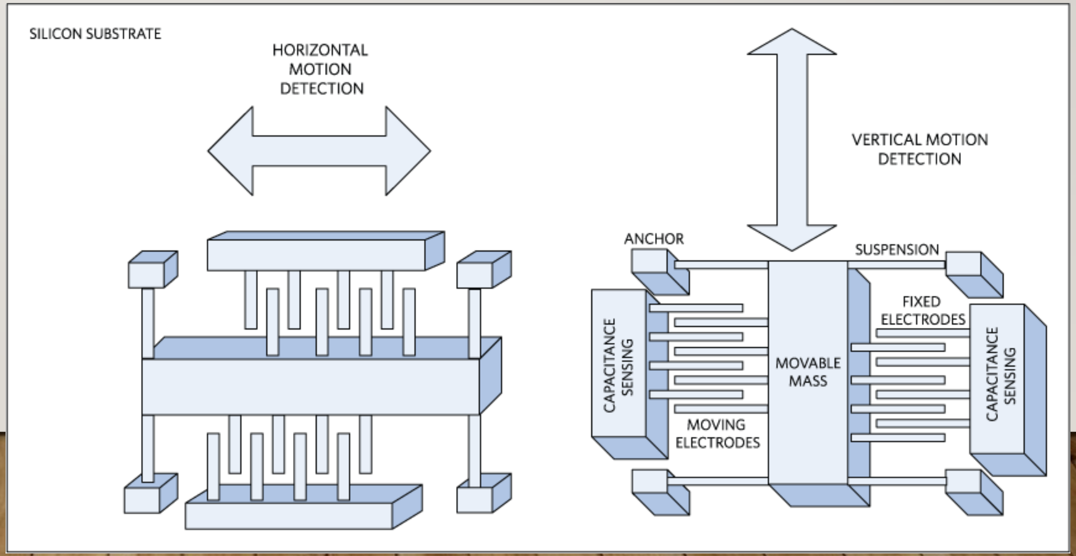
Z = 3, Color = Blue

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity ColorChanger is
    Port ( x : in  STD_LOGIC_VECTOR (7 downto 0);
            y : in  STD_LOGIC_VECTOR (7 downto 0);
           z : in  STD_LOGIC_VECTOR (7 downto 0);
           yT : out  STD_LOGIC_VECTOR (2 downto 0);
           zT : out  STD_LOGIC_VECTOR (2 downto 0));
end ColorChanger;

architecture Behavioral of ColorChanger is

begin

    process (x, y, z)

     begin
        if (z < x and z < y) then
            yT <= "100";
            zT <= "100";
        elsif (x < y and x < z) then
            yT <= "010";
            zT <= "010";
        elsif (y < x and y < z) then
            yT <= "001";
            zT <= "001";
        else
            yT <= "000";
            zT <= "000";
        end if;

    end process;

end Behavioral;
```

COMMON AND UNCOMMON USES OF ACCELEROMETERS:

- Roller Coasters: Accelerometers are used for testing.
- Weapons (Missiles, and Aircraft): Accelerometers can be used for
- Fall Detection Devices "Help! I've fallen and I can't get up!: Devices to detect a fall.
- Smart phone and Devices: Screen rotation, compasses, and fall detection, etc.





THREE COLOR CHANGING DEVICE USING ACCELEROMETER

YAY!