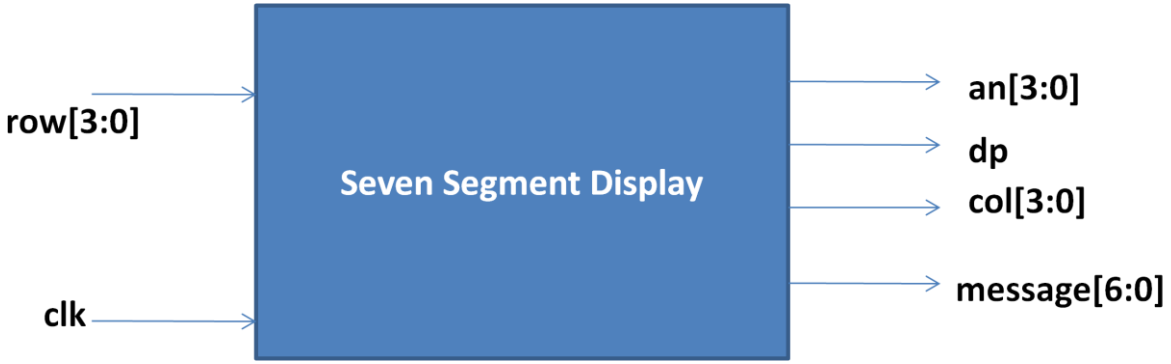
HDL code to display messages on the given seven segment display accepting Hex key pad input data.

External View



Theory

KEYPAD:

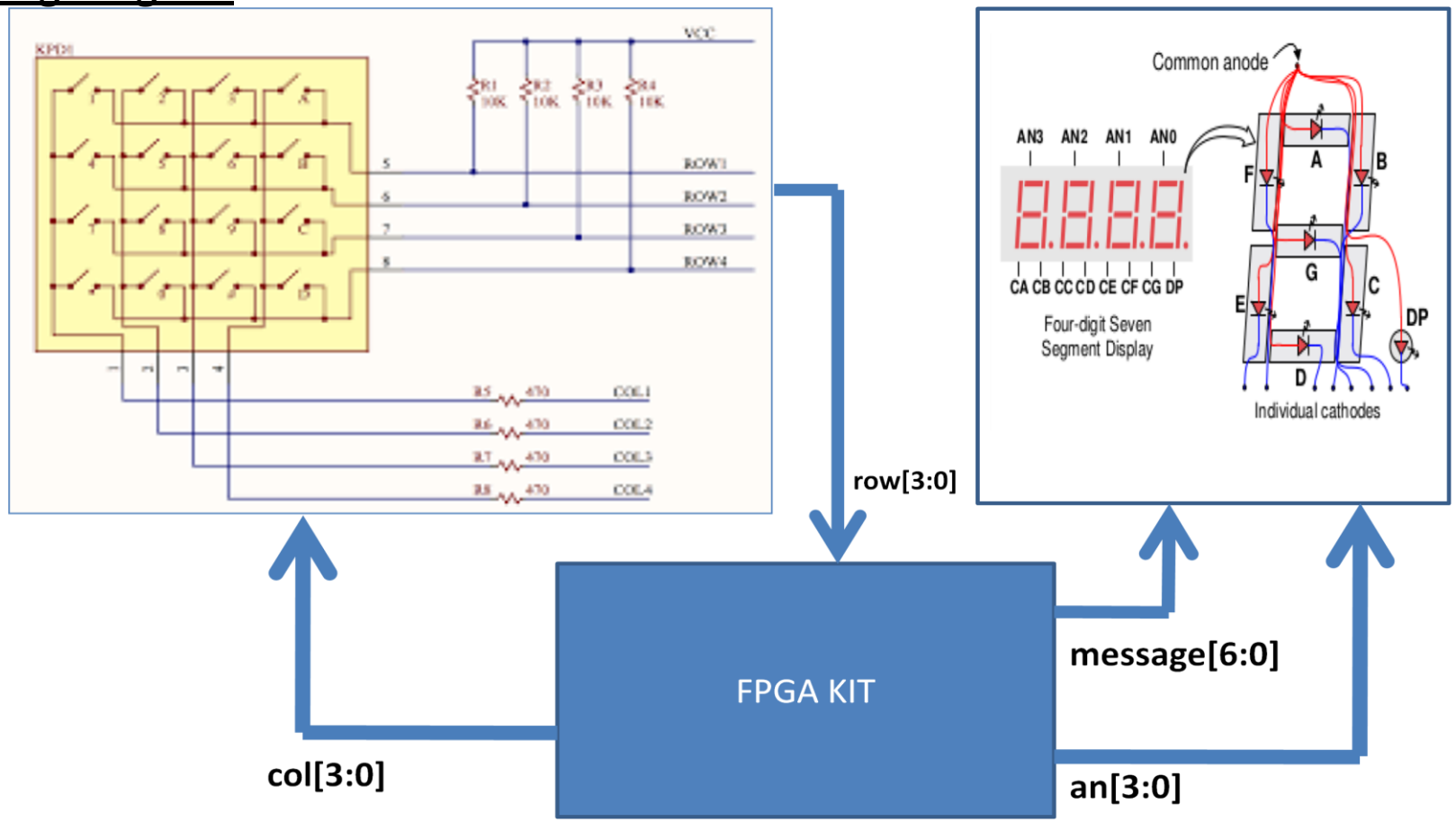
1 Functional Description The PmodKYPD utilizes 4 rows and columns to create an array of 16 momentary pushbuttons. By driving the column lines to a logic level low voltage one at a time, users may read the corresponding logic level voltage on each of the rows to determine which button, if any, is currently being pressed. Simultaneous button presses can also be recorded, although it is still required to step through each row and column separately in order to ensure that the pressed buttons do not interfere with each measurement.

2 Interfacing with the Pmod The PmodKYPD communicates with the host board via the GPIO protocol. Each button is placed within a simple voltage divider circuit. When a button is not pressed, a large pull-up resistor maintains a logic level high voltage on each of the row pins. When a column pin is driven to a logic level low voltage and a corresponding

SEVEN SEGMENT DISPLAY:

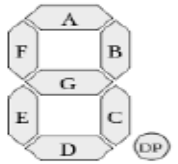
To illuminate a segment, the anode should be driven high while the cathode is driven low. However, since the Basys 3 uses transistors to drive enough current into the common anode point, the anode enables are inverted. Therefore, both the AN0..3 and the CA..G/DP signals are driven low when active.

Interfacing Diagram



Header J1					
Pin	Signal	Description	Pin	Signal	Description
1	COL4	Column 4	7	ROW4	Row 4
2	COL3	Column 3	8	ROW3	Row 3
3	COL2	Column 2	9	ROW2	Row 2
4	COL1	Column 1	10	ROW1	Row 1
5	GND	Power Supply Ground	11	GND	Power Supply Ground
6	VCC	Power Supply (3.3V/5V)	12	VCC	Power Supply (3.3V/5V)

Truth Table

col	row	message (GFEDCBA)	Display 
1110	1110	1111001	1
	1101	0011001	4
	1011	1111000	7
	0111	1111111	0
1101	1110	0100100	2
	1101	0010010	5
	1011	0000000	8
	0111	0001110	F
1011	1110	0110000	3
	1101	0000010	6
	1011	0010000	9
	0111	0000110	E
0111	1110	0001000	A
	1101	0000011	b
	1011	1000110	C
	0111	0100001	d
col= 1110(colomn 1)		row=1110(row 1)	
col= 1101(colomn 2)		row=1101(row 2)	
col= 1011(colomn 3)		row=1011(row 3)	
col= 0111(colomn 4)		row=0111(row 4)	

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity seven_segment_display is
Port ( col : inout std_logic_vector(3 downto 0);
row : in std_logic_vector(3 downto 0);
clk : in std_logic;
an: out std_logic_vector(3 downto 0);
dp: out std_logic;
message: out std_logic_vector(6 downto 0));
end seven_segment_display;

architecture seven_segment_display of seven_segment_display is
signal clk_div : std_logic_vector( 25 downto 0);
signal cnt_2bit : std_logic_vector(1 downto 0);
begin

process(clk)
begin
if clk='1' and clk'event then
clk_div <= clk_div + '1';
end if;
end process;

process(clk_div(25))
begin
if clk_div(25) = '1' and clk_div(25)'event then
cnt_2bit <= cnt_2bit + '1';
end if;
end process;

process(cnt_2bit)
begin
case cnt_2bit is
when "00" => col<= "1110";
when "01" => col<= "1101";
when "10" => col <= "1011";
when "11" => col<= "0111";
when others => null;
end case;
end process;
```

```
process(col,row)
begin
case col is
when "1110" =>
case row is
when "1110" => message<= "1111001";
when "1101" => message <= "0011001";
when "1011" => message <= "1111000";
when "0111" => message <= "1111111";
when others=> message <= "1111111";
end case;

when "1101" =>
case row is
when "1110" => message<= "0100100";
when "1101" => message<= "0010010";
when "1011" => message <= "0000000";
when "0111" => message <= "0001110";
when others=> message<= "1111111";
end case;

when "1011" =>
case row is
when "1110" => message<= "0110000";
when "1101" => message<= "0000010";
when "1011" => message<= "0010000";
when "0111" => message<= "0000110";
when others=> message<= "1111111";
end case;

when "0111" =>
case row is
when "1110" => message<= "0001000";
when "1101" => message<= "0000011";
when "1011" => message<= "1000110";
when "0111" => message<= "0100001";
when others=> message<= "1111111";
end case;
when others=> null;
end case;
end process;

an <= "0000";
dp <= '0';
end seven_segment_display;
```

XDC file:

```
## Clock signal
set_property PACKAGE_PIN W5 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]


#7 segment display
set_property PACKAGE_PIN W7 [get_ports {message[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {message[0]}]
set_property PACKAGE_PIN W6 [get_ports {message[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {message[1]}]
set_property PACKAGE_PIN U8 [get_ports {message[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {message[2]}]
set_property PACKAGE_PIN V8 [get_ports {message[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {message[3]}]
set_property PACKAGE_PIN U5 [get_ports {message[4]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {message[4]}]
set_property PACKAGE_PIN V5 [get_ports {message[5]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {message[5]}]
set_property PACKAGE_PIN U7 [get_ports {message[6]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {message[6]}]

set_property PACKAGE_PIN V7 [get_ports dp]
        set_property IOSTANDARD LVCMOS33 [get_ports dp]

set_property PACKAGE_PIN U2 [get_ports {an[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {an[0]}]
set_property PACKAGE_PIN U4 [get_ports {an[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {an[1]}]
set_property PACKAGE_PIN V4 [get_ports {an[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {an[2]}]
set_property PACKAGE_PIN W4 [get_ports {an[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {an[3]}]
```

```
##Pmod Header JC
##Sch name = JC1
set_property PACKAGE_PIN K17 [get_ports {col[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {col[3]}]
##Sch name = JC2
set_property PACKAGE_PIN M18 [get_ports {col[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {col[2]}]
##Sch name = JC3
set_property PACKAGE_PIN N17 [get_ports {col[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {col[1]}]
##Sch name = JC4
set_property PACKAGE_PIN P18 [get_ports {col[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {col[0]}]
##Sch name = JC7
set_property PACKAGE_PIN L17 [get_ports {row[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {row[3]}]
##Sch name = JC8
set_property PACKAGE_PIN M19 [get_ports {row[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {row[2]}]
##Sch name = JC9
set_property PACKAGE_PIN P17 [get_ports {row[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {row[1]}]
##Sch name = JC10
set_property PACKAGE_PIN R18 [get_ports {row[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {row[0]}]
```