



Problem

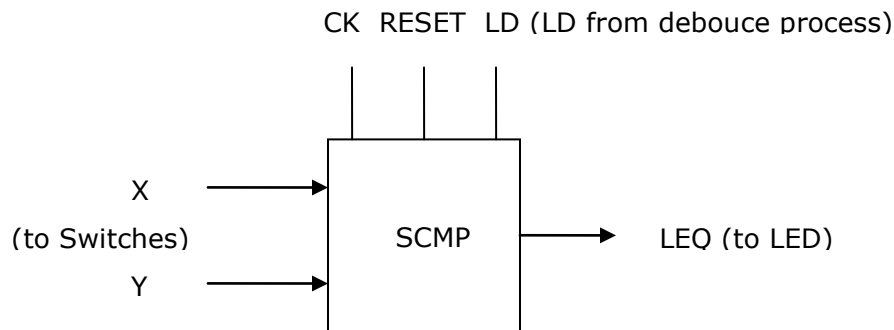


Fig. Sequential "Less Than or Equal" Compare

Implement a sequential "Less Than or Equal" comparator whose single-step enable LD is generated by a two push-buttons debounce process. The RESET is connected to the push button. The input X and Y are connected to the switches. The output LEQ is to the LED. The figure above shows a block diagram of the device.

Entity scmp is

```
Port (      X, Y : in  std_logic;
        LEQ : out std_logic;
        CK, RESET, LD : in  std_logic);
End scmp;
```

The device compares two unsigned vectors V1 and V2. The output LEQ indicates the condition if V1 is less than or equal to V2. Users first reset the device. The bits of V1 and V2 are applied at Port X and Y starting from the most significant bits. The output LEQ is a one when the vector V1 is less than or equal to V2. For example, let V1 = "1001" and V2 = "1010". After a reset the switches connected to Port X and Y set to '1' and '1' and the debounce button is pressed. The LEQ remains a zero. Next '0' and '0' are applied LEQ remains a zero. Next '0' and '1' are applied LEQ becomes a one. Last '1' and '0' are applied LEQ remains a one (since V1 is already found less than V2 on the previous application of the bits). Test also for V1 = "0010" and V2 = "0100" and V1 = "1010" and V2 = "1010".

Architecture beh of scmp is

Signal temp, done_flag: std_logic; -- temp wired to Z

Begin

Process(cK)

Begin

If ck='1' and ck'event then

If reset = '1' then done_flag <= '0'; else

If flag = '0' then

if a='1' and b='0' then temp <='1'; done_flag <= '1';

elsif a='0' and b='1' then temp <= '0'; done_flag <= '1';

else null;

end if;

end if;

end if;

end if;

end process;

z <= temp;

end beh;