```
1   ----------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:    08:45:01 12/17/2015
6   -- Design Name:
7   -- Module Name:    sr_latch - Behavioral
8   -- Project Name:
9   -- Target Devices:
10  -- Tool versions:
11  -- Description:
12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity sr_latch is
33     Port( BTN0: in std_logic;
34           BTN1: in std_logic;
35           LED0: out std_logic);
36
37  end sr_latch;
38
39  architecture Behavioral of sr_latch is
40  signal notQ: std_logic;
41  signal Q2: std_logic;   -- copy of Q to avoid synthesis error
42
43  begin
44  -- logic design of the SR-Latch
45  LED0 <= Q2;
46  Q2 <= BTN0 nor notQ;       -- just having this statement throws and error during the
    synthesis as Q can not read. In order to avoid that error we can copy the signal to a
    second variable.
47  notQ <= BTN1 nor Q2;
48  -- remember that all statements here run concurrently, so the order of their
    placement does not matter
49
50  end Behavioral;
51
52
```