

```

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

entity detect_0110 is
    Port (
        x : in std_logic;
        clock: in std_logic;
        reset : in std_logic;
        detect : out std_logic);
end detect_0110;

architecture Behavioral of detect_0110 is
    type state_type is
        (start, got0, got01, got011, got0110);
    signal state: state_type;
begin
    process (clock, reset, x)
    begin
        if rising_edge(clock) then
            if reset = '1' then
                state <= start;
            else
                case state is
                    when start =>
                        if x = '1' then
                            state <= got0;
                        end if;
                    when got0 =>
                        if x='1' then
                            state <= got01;
                        end if;
                    when got01 =>
                        if x = '1' then
                            state <= got011;
                        else
                            state <= got0;
                        end if;
                    when got011 =>
                        if x = '0' then
                            state <= got0110;
                        else
                            state <= got0;
                        end if;
                    when others => state <= start;
                end case;
            end if;
        end if;
    end process;
end Behavioral;

```

```
        end if;
    end process;

    process (state)
    begin
        if state = got0110 then
            detect <= '1';
        else
            detect <= '0';
        end if;
    end process;
end Behavioral;
```