SECURITY ALARM SYSTEM

NICK DENEAU

MATI GORO

KEVIN JOU

RAMI MAROGY

PURPOSE

- Protect and secure your home
- An estimated 3.7 million household burglaries occur each year
- Manage electricity with a home security system

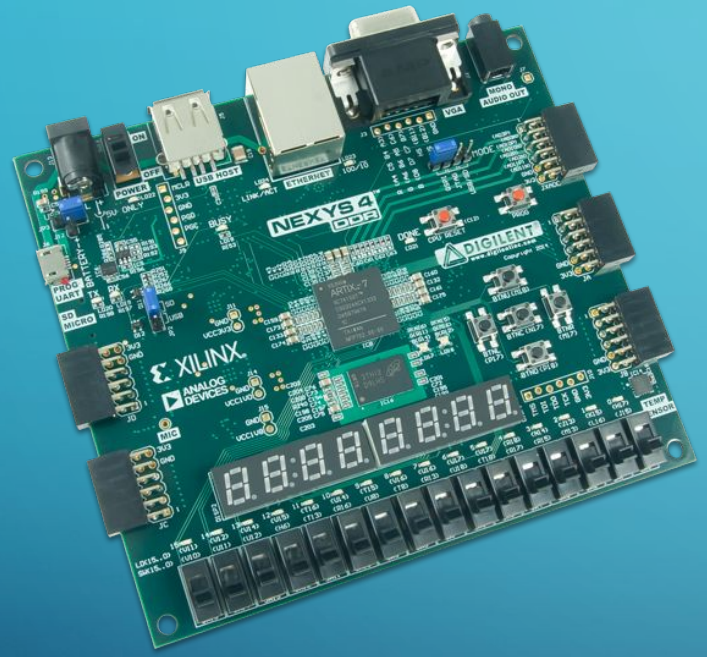
STATISTICS

- In the US, there is a home burglary that takes place every 13 seconds, 4 a minute, 240 an hour, and approximately 6000 a day.
- The estimated amount of households that have some sort of home security is about 30%.
- 1 in 3 homes without a security system will fall victim to a burglary as compared to 1 in 250 homes that do have a security system.

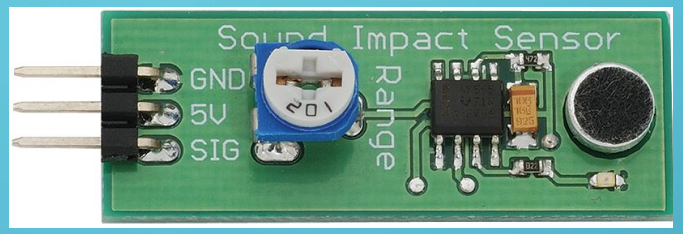
INSPIRATION

- Able to track movement with inexpensive sensors at a given space rather than expensive equipment.
- Easy programmable system for plug and play ability of the consumer.
- Being able to reach more people due to relatively low cost of system to put together, install, and maintain.

COMPONENTS

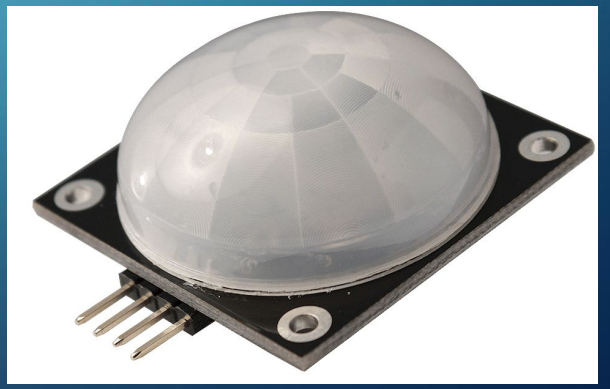


Nexys-4 DDR
Artix-7 FPGA Board

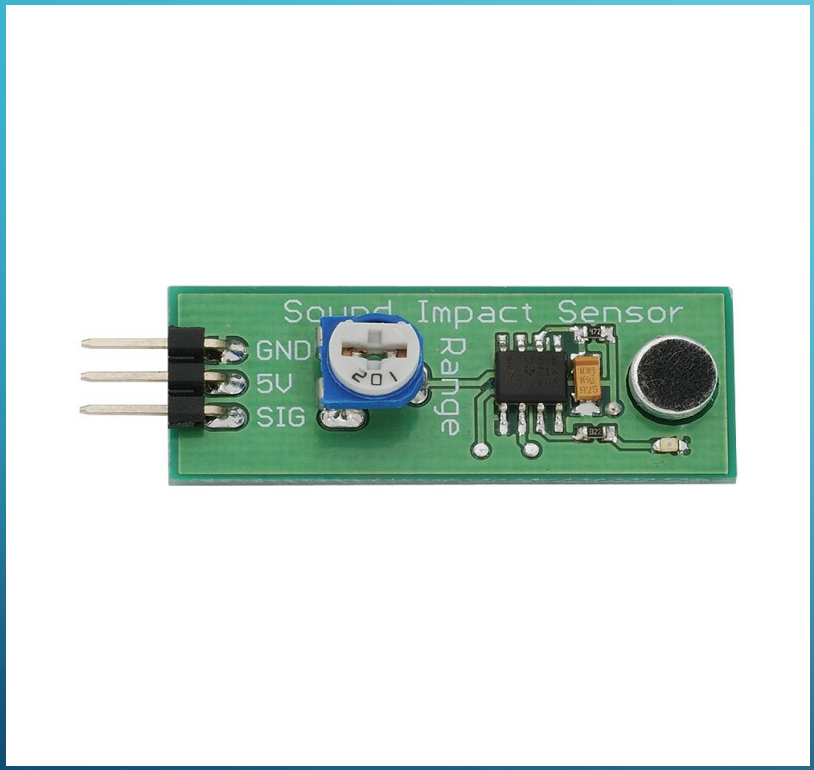


Parallax Sound
Impact Sensor

Parallax Wide Angle
PIR Sensor

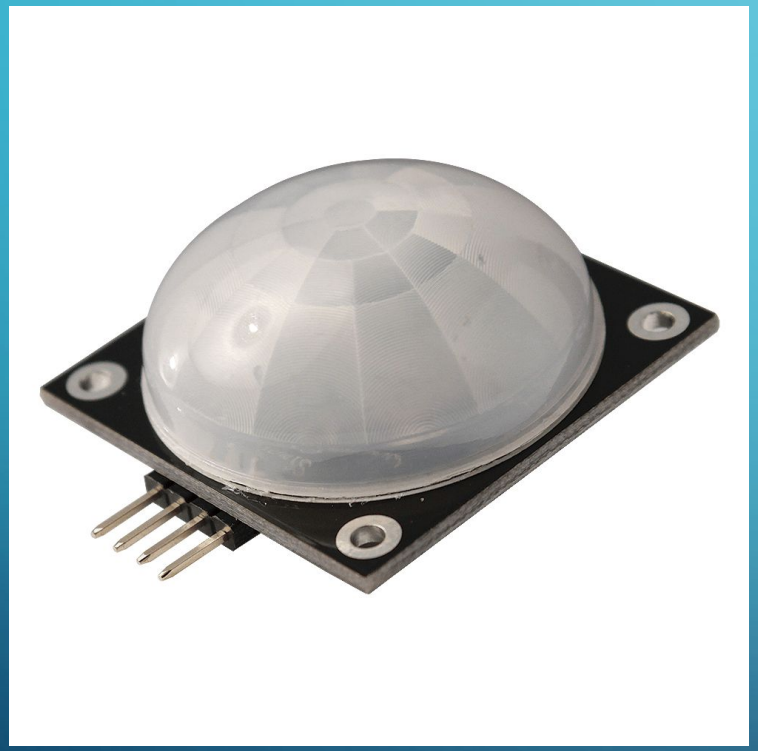


PARALLAX SOUND IMPACT SENSOR



- Takes incoming pressure from sound due to the wave created to produce a signal.
- Sensor is a one bit producing sensor such that if pressure is detected within its 3 meter radius of sensing it will produce a high 1 bit output.
- Detects sounds such as glass breaking. Anything that produces high decibel change in a short period of time.

PARALLAX WIDE ANGLE PASSIVE IR SENSOR

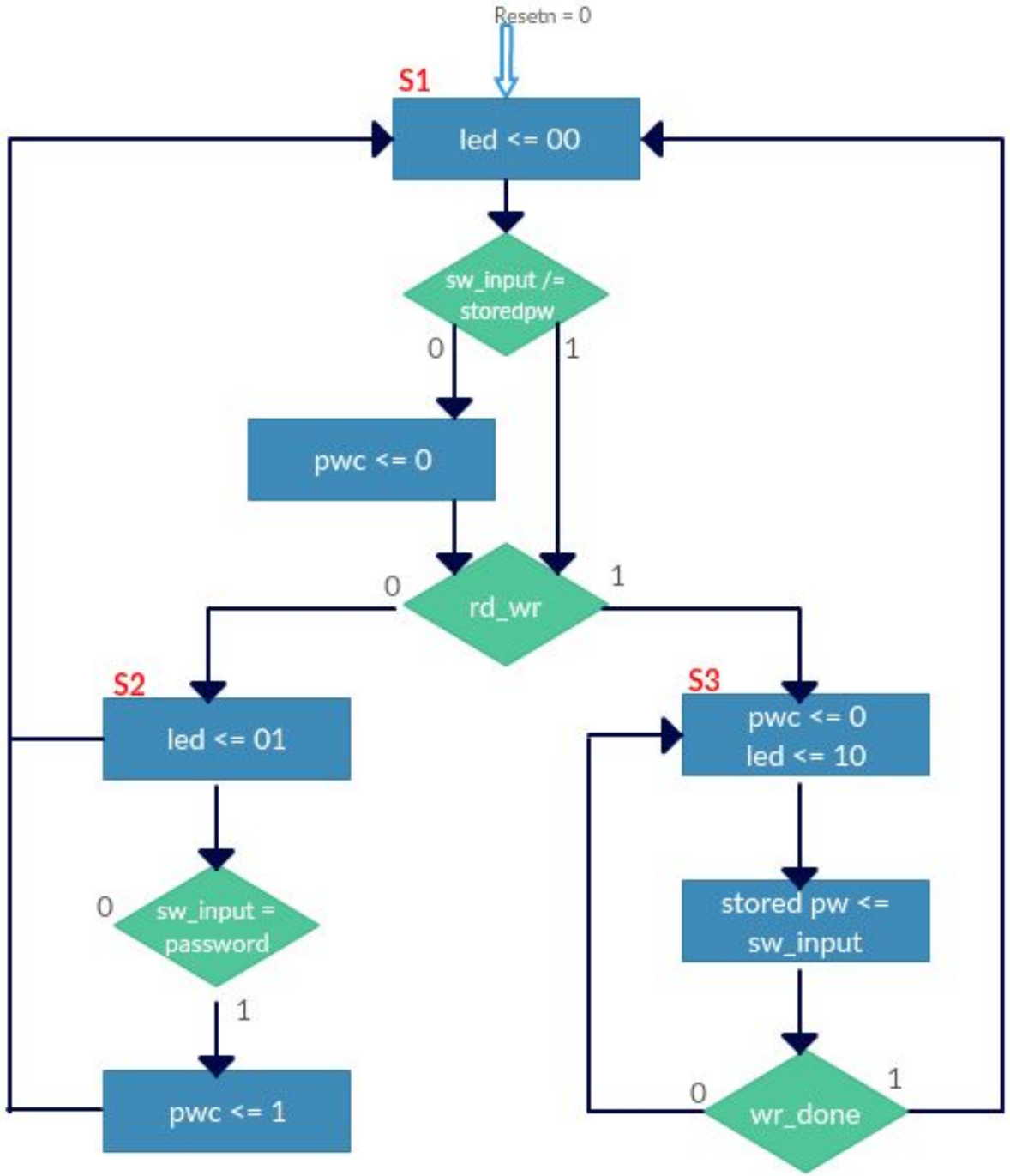


- A digital sensor that detects motion within a 180 Degree path of its detection area.
- When motion is detected the sensor outputs a 1 High bit that is used by the board to signal the system that the alarm has been tripped within its area of view.
- Typically detects people up to 30 feet away.

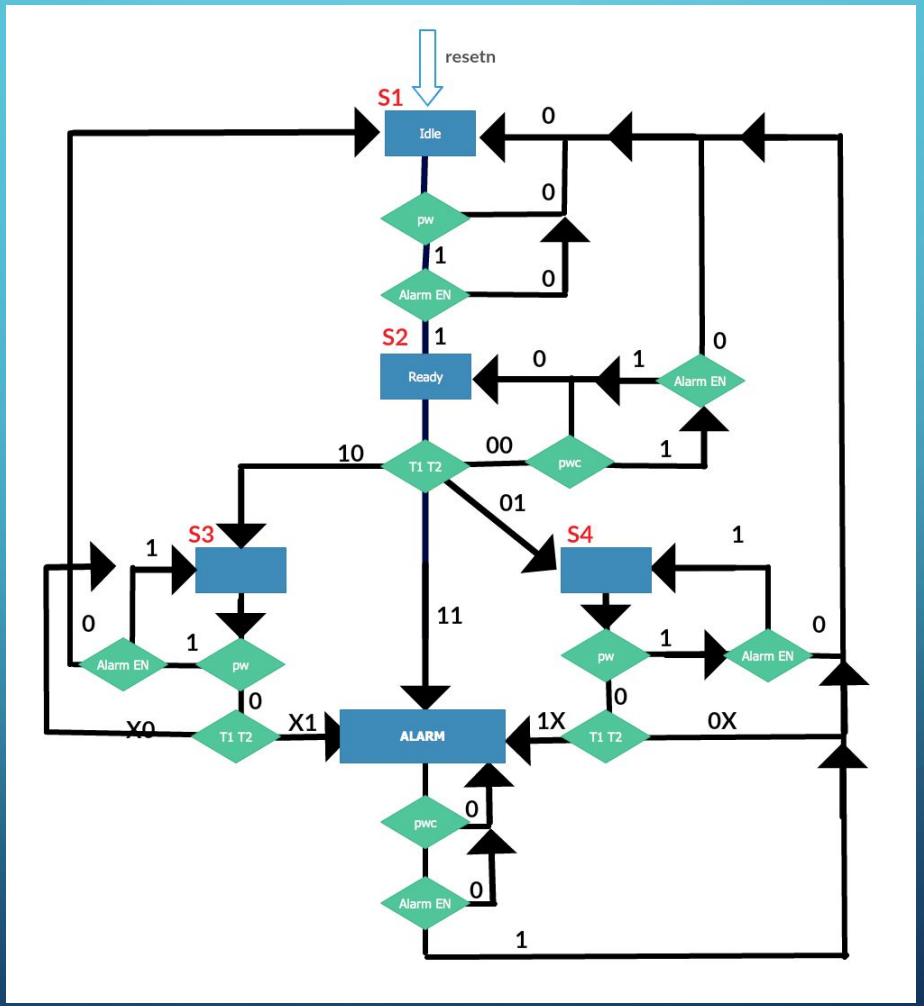
CURRENT LIMITATIONS

- The Parallax Wide Angle PIR Sensor was so sensitive that it would always output an active high.
- Both sensors were inexpensive which did not allow us to get accurate responses.
- The FPGA board only supports 3.3V, whereas the sensors require 5V for maximum accuracy.
- Unable to produce a sound from the sound impact sensor.

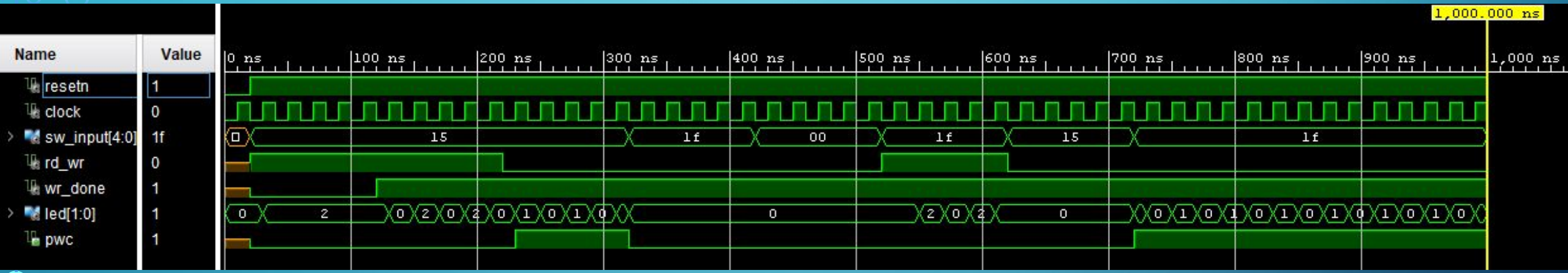
ASM

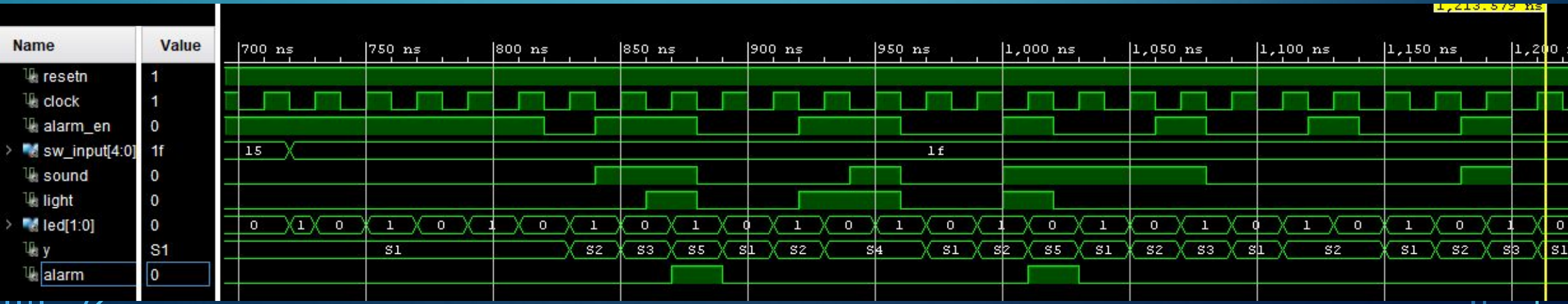

Resetn = 0
S1
led <= 00
sw_input /= storedpw
0
1
pwc <= 0
0
rd_wr
1
S2
led <= 01
S3
pwc <= 0
led <= 10
0
sw_input = password
1
stored pw <= sw_input
pwc <= 1
0
wr_done
1

MAIN ASM FORMAT

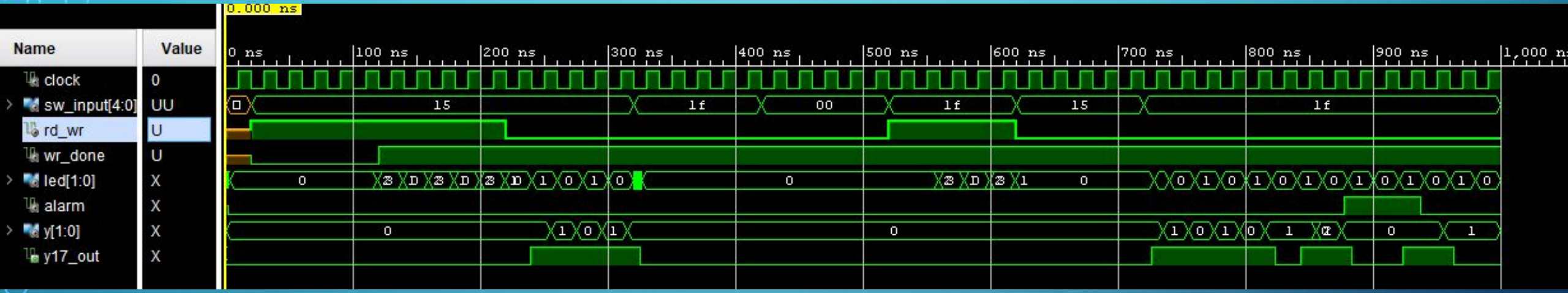


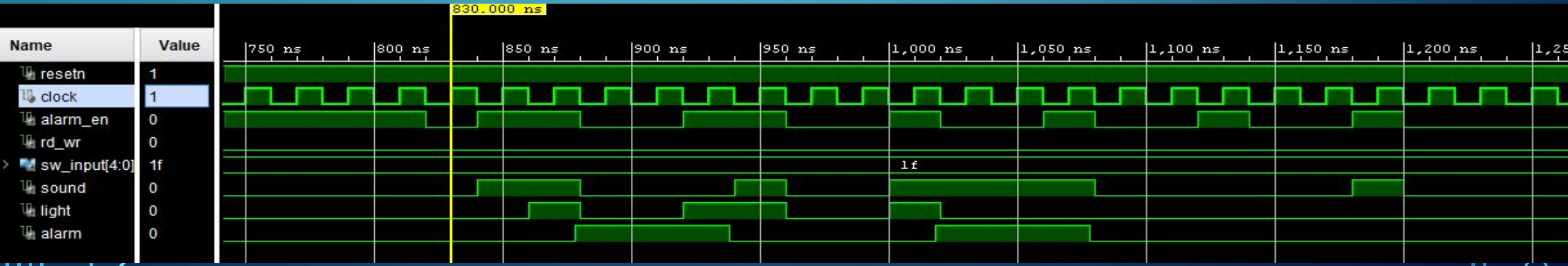
BEHAVIORAL SIMULATION





TIMING SIMULATION





SAMPLE CODE

```
begin
    if resetn = '0' then
        y <= s1; -- initial state
        else if (clock'event and clock = '1') then
            case y is
                when s1 =>
                    if pwc = '0' then y<= s1; end if;
                    if pwc = '1' and alarm_en = '1' then y<=s2; end if;
                when s2 =>
                    if t1 = '0' and t2 = '0' and pwc = '0' then y <= s2; end if;
                    if t1 = '0' and t2 = '0' and pwc = '1' and alarm_en = '1' then y <= s1; end if;
                    if t1 = '0' and t2 = '1' then y <= s4; end if;
                    if t1 = '1' and t2 = '0' then y <= s3; end if;
                    if t1 = '1' and t2 = '1' then y <= s5; end if;
                when s3 =>
                    if pwc = '1' and alarm_en = '1' then y <= s1; end if;
                    if t2 = '0' then y <= s3; end if;
                    if t2 = '1' then y <= s5; end if;
                when s4 =>
                    if pwc = '1' and alarm_en = '1' then y <= s1; end if;
                    if t1 = '0' then y <= s4; end if;
                    if t1 = '1' then y <= s5; end if;
                when s5 =>
                    if pwc = '1' and alarm_en = '1'then y <= s1; end if;
            end case;
        end if;
    end if;
end process;
Outputs: process (y)
begin
    case y is
        when s1 => alarm <= '0';
        when s2 => alarm <= '0';
        when s3 => alarm <= '0';
        when s4 => alarm <= '0';
        when s5 => alarm <= '1';
    end case;
end process;
```

LETS TEST IT!