SECURITY ALARM SYSTEM

NICK DENEAU MATI GORO KEVIN JOU RAMI MAROGY

 \bigcirc

 \frown

Ó

 \bigcirc

 \bigcirc

 \bigcirc

 \bigcirc

 \bigcirc

 \bigcirc

PURPOSE

• Protect and secure your home

• An estimated 3.7 million household burglaries occur each year

Manage electricity with a home security system

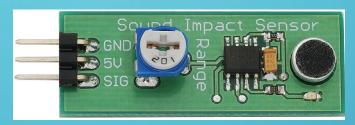
STATISTICS

- 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



Parallax Sound Impact Sensor

Nexys-4 DDR Artix-7 FPGA Board 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



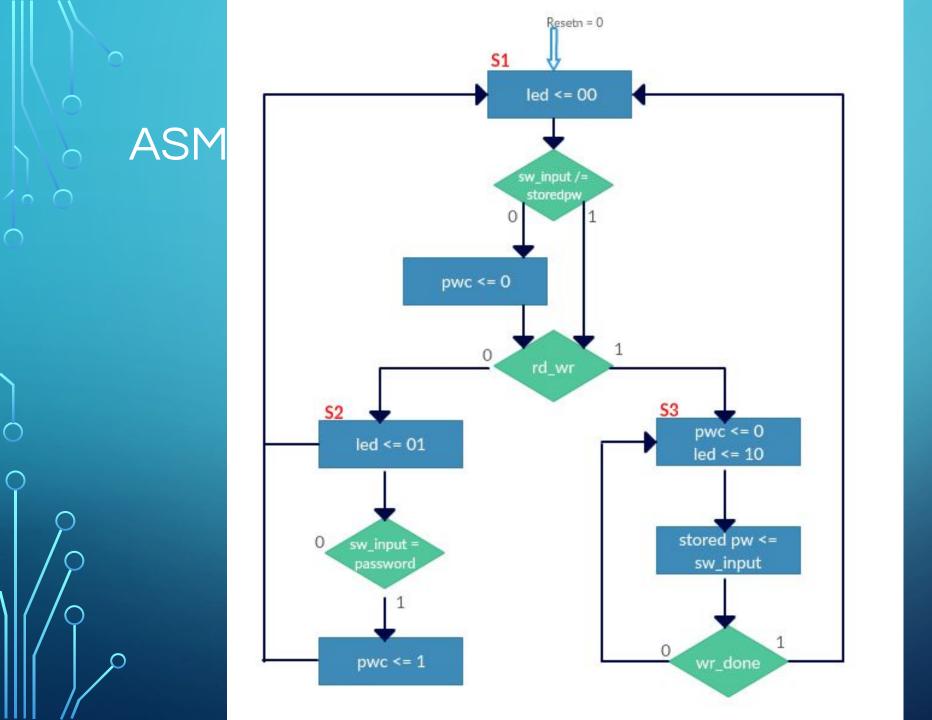
• A digital sensor that detects motion within a 180 Degree path of its detection area.

• 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.

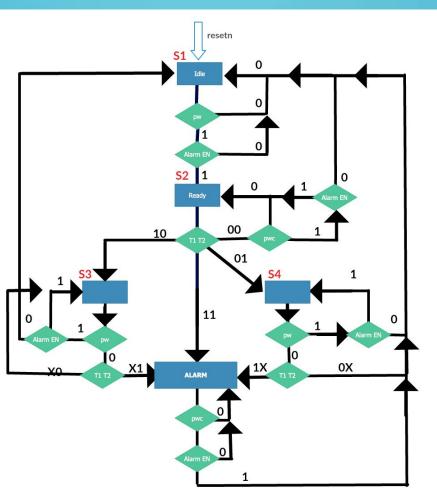
CURRENT LIMITATIONS

- 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.





MAIN ASM FORMAT



 \bigcirc

ρ

 \bigcap

BEHAVIORAL SIMULATION

											1,000.00	/00 ns
Name		0 ns	100 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns	900 ns	1,000 ns
🕼 resetn	1											
1/2 clock	0											
> 🦋 sw_input[4:0]	1f		15) 1f	X00	X 1f	15		lf		
₩ rd_wr	0											
₩ wr_done	1											
> 📢 led[1:0]	1				фX	0	2/0/2	<u>ه م</u> لغ				
1 pwc	1											

																														-/-	10.01		1
Name	Value		700 ns		750 1	ns	_	800 n	is I I	100 11 - 10	850 n	IS I I	26	900	ns	a 10	95	0 ns	e		1,00	0 ns		1,05	0 ns	 1,1	00 n:	5	1,15	50 ns	i si	1,2	00
🕼 resetn	1							1		î	1									_													
🖫 clock	1			17		11													17											11			
🕼 alarm_en	0								1								1																
🛛 🔣 sw_input[4:0]	1f		15																lf														
🕼 sound	0																																
🕼 light	0											-				1	2				1										20 		
📲 led[1:0]	0		0)	1 0	1	Xo			0)	1	0	$\langle 1$	\mathcal{T}		1	χo	X	1 X	0	X		0			χ_1	•X	1	χo					0
u∰a y	S1				S1					S2)	S 3	X 85) s	īχ	S2	X	S4		S1	X s	2)(S5	S1	S2	X SS	sı X		S2	S1	Xs	2 (3	53 X	SI
🖫 alarm	0	_																														_	

TIMING SIMULATION

		0.000 ns										
Name	Value	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns	900 ns 1,	.000 n:
🕼 clock	0											
> 🔣 sw_input[4:0]	UU		15			X 00		15	EX.	lf		
🔓 rd_wr	U											
₩ wr_done	U											
> 🔣 led[1:0]	Х	0	XZ XD XZ XD	<u>Z D 1 0 1</u>	0	0	X2 XD	X23X1 0		10101		
🕼 alarm	Х											
> 🔣 y[1:0]	Х		0	X1X0X	1)		0				0 (1)	
🖫 y17_out	Х											

				830.000 ns								
Name	Value	750 ns	800 ns	850 ns	 900 ns	950 ns	1,000 ns	1,050 ns	1,100 ns	1,150 ns	1,200 ns	1,2
∿a resetn	1											
16 clock	1											
14 alarm_en	0											
₩ rd_wr	0											
> 1 sw_input[4:0]	1f						lf					
🕼 sound	0											
내 light	0											
🕼 alarm	0											
												\mathbf{Q}

SAMPLE CODE

```
begin
   if resetn = '0' then
       y <= sl; -- initial state
        else if (clock'event and clock = '1') then
           case y is
                when sl =>
                    if pwc = '0' then y<= sl; end if;
                    if pwc = 'l' and alarm en = 'l' then y<=s2; end if;
                when s_2 =>
                    if tl = 0' and t2 = 0' and pwc = 0' then y \le s2; end if;
                    if tl = 0' and t2 = 0' and pwc = 1' and alarm en = 1' then y \le sl; end if;
                    if t1 = '0' and t2 = '1' then y \le s4; end if;
                    if tl = 'l' and t2 = '0' then y \le s3; end if;
                    if tl = 'l' and t2 = 'l' then y \leq s5; end if;
                when s3 =>
                    if pwc = 'l' and alarm en = 'l' then y \leq sl; end if;
                    if t_2 = 0' then y \le s_3; end if;
                    if t_2 = '1' then y \leq s_5; end if;
                when s4 =>
                    if pwc = 'l' and alarm en = 'l' then y <= sl; end if;
                    if tl = '0' then v \le s4; end if;
                    if tl = 'l' then y \leq s5; end if;
                when s5 =>
                    if pwc = 'l' and alarm en = 'l'then y <= sl; end if;
           end case:
       end if;
   end if;
end process;
Outputs: process (y)
begin
   case y is
        when sl => alarm <= '0';
        when s2 => alarm <= '0';
        when s3 => alarm <= '0';
        when s4 => alarm <= '0';
        when s5 => alarm <= '1';
   end case;
end process;
```

 \mathcal{O}

LETS TEST IT!

 \cap

 \bigcirc

 \bigcap