Sistema de Alarme Residencial FSM: Conceitos e Implementação em VHDL

Explorando os fundamentos de sistemas digitais através da implementação prática de um controlador de alarme residencial baseado em máquina de estados finitos.

Estados do Sistema de Alarme Residencial

DESARMADO

Sistema inativo, sensores não acionam o alarme. O usuário pode entrar e sair livremente sem preocupações.

ARMADO

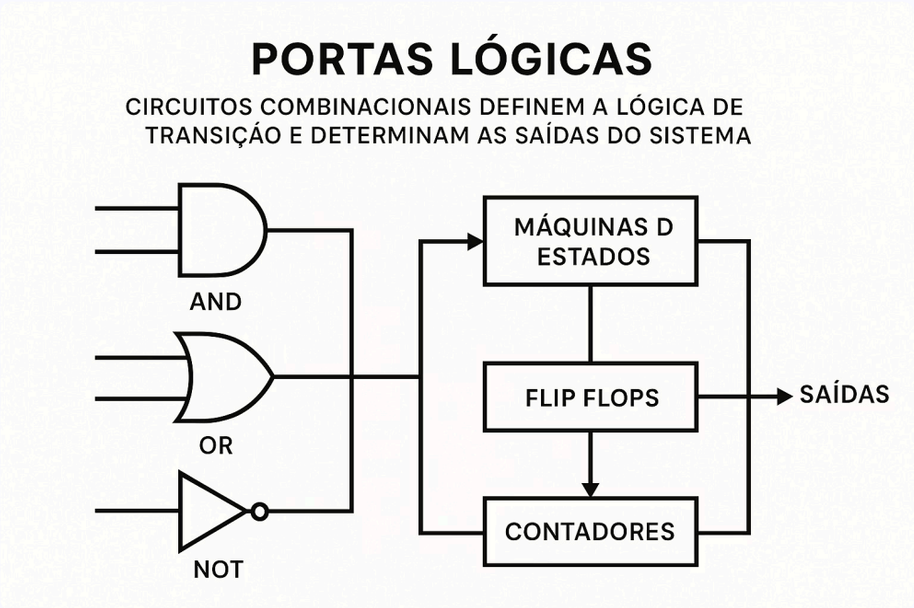
Sistema ativo, monitorando continuamente todos os sensores com retardo configurável para permitir saída segura.

DISPARO

Alarme acionado após detecção de intrusão e término do contador de retardo, ativando sirene e notificações.

A FSM garante transições seguras entre estados, evitando falsos alarmes e proporcionando operação confiável do sistema de segurança residencial.

Componentes Digitais Fundamentais



01

Portas Lógicas

Circuitos combinacionais definem a lógica de transição e determinam as saídas do sistema.

02

Flip-Flops

Elementos de memória sequencial que armazenam o estado atual da FSM entre ciclos de clock.

03

Contadores

Implementam o retardo configurável necessário antes do disparo efetivo do alarme.

04

Máquina de Estados

FSM controla o fluxo completo entre estados conforme as entradas dos sensores e comandos do usuário.

Implementação em VHDL: Estrutura Básica

1

Definição dos Estados

Tipo enumerado contendo DESARMADO, ARMADO e DISPARO como valores possíveis.

2

Processo Sequencial

Atualização sincronizada do estado com base nas entradas e sinal de clock do sistema.

3

Lógica Combinacional

Determina transições de estado e controla as saídas do alarme de forma eficiente.

4

Contador de Retardo

Implementa temporização configurável antes do disparo efetivo do alarme residencial.

```
-- Exemplo simplificado de FSM em VHDL
type estado_tipo is (DESARMADO, ARMADO, DISPARO);
signal estado_atual, proximo_estado : estado_tipo;

process(clock, reset)
begin
 if reset = '1' then
 estado_atual <= DESARMADO;
 elsif rising_edge(clock) then
 estado_atual <= proximo_estado;
 end if;
end process;
```