

# Circuito de Senha de Cofre

## Esboço de Descrição:

### 1. Descrição:

Neste presente trabalho, implementamos um circuito lógico presente em um cofre empresarial, cujo acesso é concedido através de uma das possíveis combinações de senha e identificador de funcionário. São permitidas 5 tentativas de abrir o cofre, e caso essas sejam todas inválidas, o cofre irá bloquear, e para desbloquear, um gerente terá que reiniciar o sistema da tranca.

O circuito consiste em várias portas lógicas usadas de forma a verificar se a combinação de bits inserida é correta, tanto para o funcionário, quanto para o código do cofre em si. As lâmpadas que sinalizam as tentativas e o bloqueio do cofre são controladas com o uso de flip-flops, que sequenciam o acionamento das luzes. O sistema de tentativas é reiniciado cada vez que uma combinação código-funcionário é inserida corretamente.

### 2. Mapa de Karnaugh

Senha				
	C'D'	C'D	CD	CD'
A'B'	0	0	0	0
A'B	0	0	0	0
AB	0	1	0	0
AB'	0	0	0	0

Funcionário				
	G'H'	G'H	GH	GH'
E'F'	0	0	0	1
E'F	1	0	0	1
EF	1	0	0	1
EF'	0	1	0	0

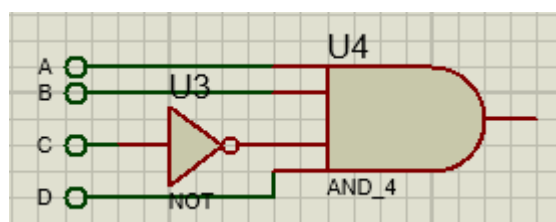
### 3. Circuito Lógico Simplificado

A entrada do projeto se dá por 8 Bits (A, B, C, D, E, F, G, H), nos quais de A até D se referem a senha, e de E até H para funcionário.

Em relação a senha, não há o que simplificar, já que é uma única possibilidade para tal, logo o circuito fica da seguinte forma:

Senha				
A	B	C	D	S
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

ABC'D

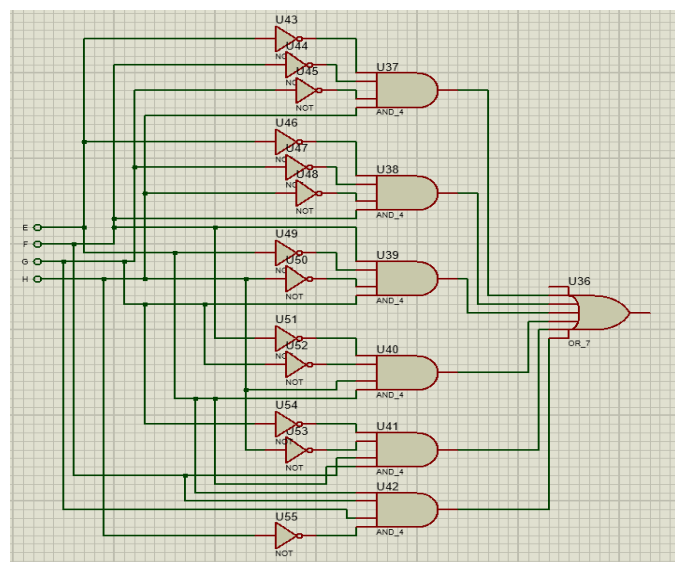


O mesmo não se pode afirmar com os usuários, já que utilizando o mapa de karnaugh é possível reduzir algumas entradas, tornando mais fácil de compreender o que está acontecendo. Vejamos nas imagens abaixo:

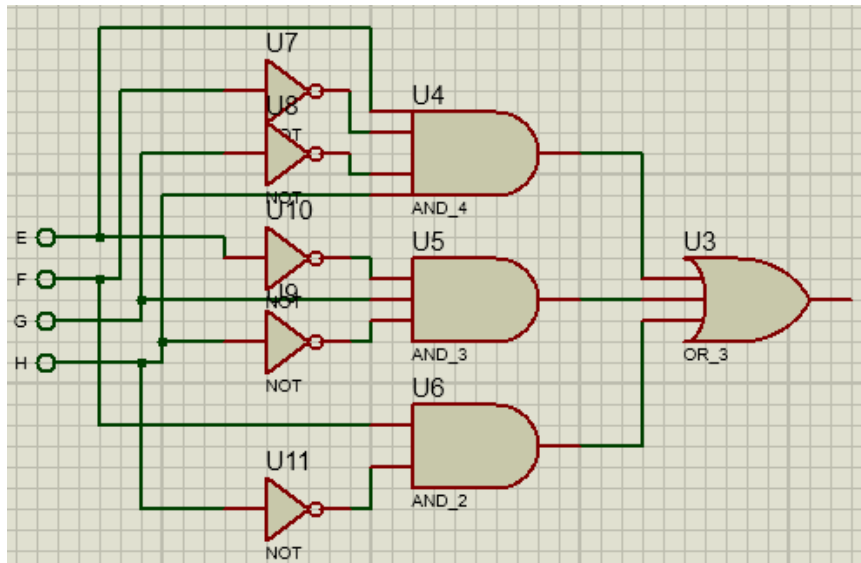
Usuário				
E	F	G	H	S
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1
1	1	1	1	0

Completo:

$$E'F'GH' + E'FG'H' + E'FGH' + EF'G'H + EFG'H' + EFGH'$$



**Simplificado:**  
 $EF'G'H + E'GH' + FH'$



Como é possível ver, a forma reduzida pelo mapa de karnaugh torna muito mais simples e compacto, além de tornar o circuito mais fácil de se compreender.

#### 4. Pseudocódigo:

O pseudocódigo abaixo foi feito baseado na pseudolinguagem Portugal:

##### Var

```
inteiro bitsUser;
inteiro bitsSenha,
inteiro Senha=1101
inteiro usersPermitidos[] = {0010, 0100, 0110, 1001, 1100, 1110};
inteiro i;
booleano userEncontrado = falso;
```

##### Inicio

```
escreva("Digite o usuario: ")
leia (bitsUser)

escreva("Digite a senha: ")
```

**leia** (bitsSenha)

**para** i **de** 0 **ate** 6 **faça**:

**se** bitsUser == usersPermitidos[i] **entao**

        userEncontrado = verdadeiro;

**fimse**

**fimpara**

**se** Senha == bitsSenha **e** userEncontrado == verdadeiro **entao**

**escreva**("Acesso permitido! (Luz verde acende)") // Abre o Cofre

**senao**

    // Mensagem de Erro e Ativação do Alarme

**escreva**("Acesso negado! Senha e/ou usuario incorreto! (Uma luz amarela acende. Caso seja a 5ª, a vermelha se acende junto)")

**fimse**

**Fim**

## 5. Diagrama ASM

A partir do pseudocódigo, é possível gerar um diagrama que modela o comportamento do circuito, que é chamado de diagrama ASM de alto nível. Esse diagrama pode ser visualizado na figura abaixo:

