

CONTROLE DE ELEVADOR

Aluno.....: Leonardo Rafael
Coordenador: Prof. Engº Luiz Antonio Vargas Pinto
vargasp@uol.com.br

Escola Técnica Rubens de Faria e Souza



Dedicatória e Agradecimentos

Dedico aos meus pais pelo carinho e compreensão, a minha escola sem onde não teria onde adquirir conhecimentos e aos meus professores pelo apoio.

Agradeço aos meus professores pelo incentivo e em especial ao Prof. Vargas pelo estímulo ao projeto

Índice

Dedicatória e Agradecimentos	2
Índice	3
Resumo	4
Diagrama Geral	5
Introdução	5
Objetivo	6
15/08/2003	6
1º Andar	6
2º andar	7
05/09/2003	7
a) Parado	7
b) Desce	8
17/11/2003	9
c) Sobe	10
16/09/2003	10
3º Andar	10
Botões	11
22/09/2003	11
26/09/2003	11
14/10/03	13
20/10/2003	14
23/10/2003	18
28/12/03	19
Problema do comando de SOBE	19

Resumo

Este é um projeto para ter o controle de um elevador de um prédio com três andares, onde cada andar tem apenas um botão de chamada do elevador.

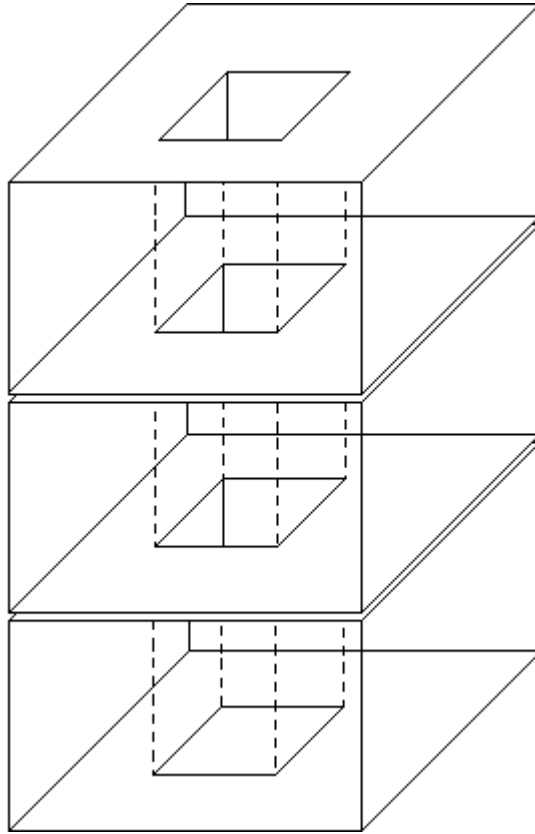
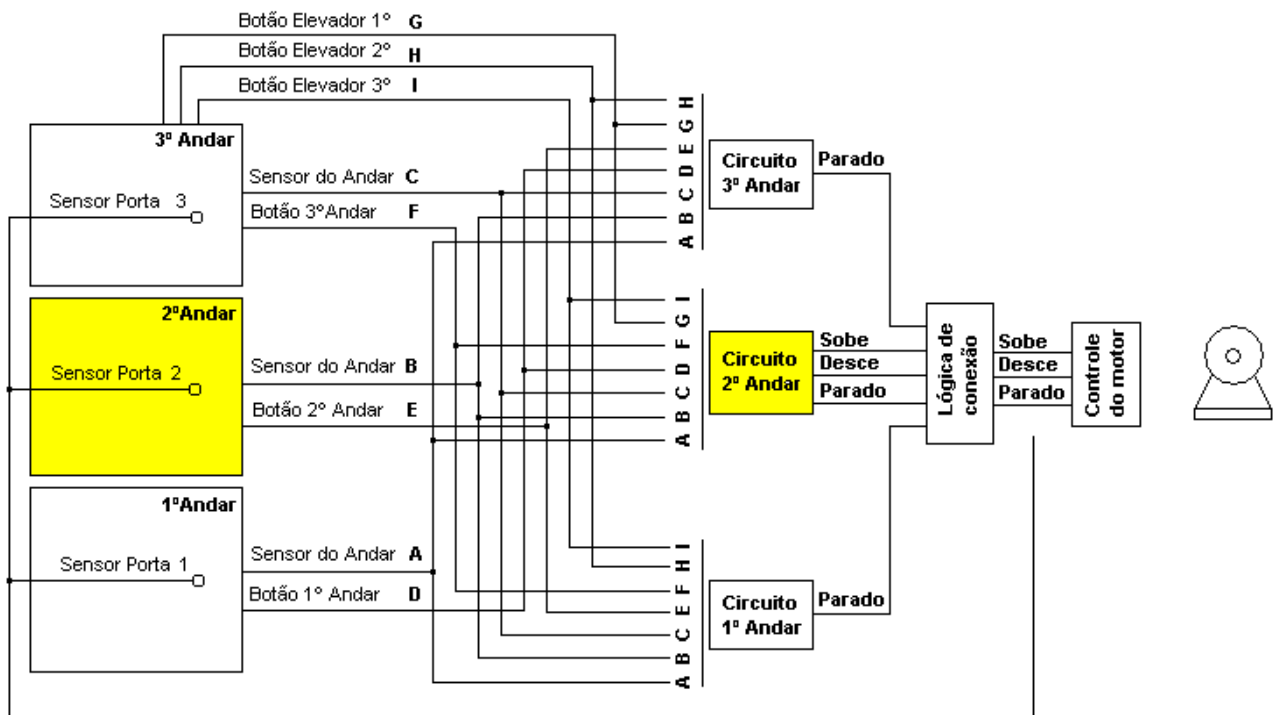


Diagrama Geral



Introdução

O projeto nasceu da idéia de aproveitar o conhecimento anterior de Sistemas Digitais Microprocessados nos cursos da área de elétrica da ETE Rubens de Faria e Souza.

O primeiro passo da concepção foi estabelecer o envolvimento entre a realidade de um elevador, seu funcionamento, suas limitações e passar isso á variáveis de sistema.

A definição inicial foi estabelecer letras para cada peça envolvida, assim sugerimos:

A	B	C	D	E	F	G	H	I
1º	2º	3º	1º	2º	3º	1º	2º	3º
Sensor dos andares			Botões da Parede do andar			Botões de dentro do elevador		

E, ainda considerando o sensor da porta de cada andar teríamos $2^{12} = 4096$ condições possíveis. Numa primeira análise isto pareceu uma situação impossível de estudar.

Daí a idéia de separar os sensores das portas de cada andar conforme o esquema anterior, daí reduziríamos á $2^9 = 512$. O que sem dúvida já viabiliza o projeto.

Objetivo

15/08/2003

Uma simples lembrança ajudou a melhorar a situação do projeto: Os sensores de andar não podem estar indicando mais de um andar em uso. Ora, isso os coloca em 3 casos diferentes:

- a) Estudo do primeiro andar;
- b) Estudo do segundo andar;
- c) Estudo do terceiro andar.

1º Andar

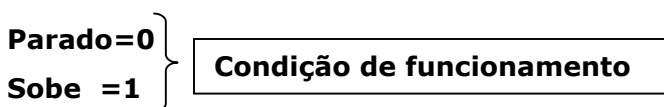
O estudo do 1º andar pode ser efetuado analisando-se somente a condição de **parado**, porque, ou o elevador está em movimento de **subida** ou ele está **parado**. Porém uma análise detalhada auxiliada pelo excel nos leva a conclusão de que existem somente 4 condições onde estamos **parados**.

1º Andar										
Sensor do andar			Botão andar			Botão elevador			Saída	
1º	2º	3º	1º	2º	3º	1º	2º	3º	Sobe	Desce
1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	0

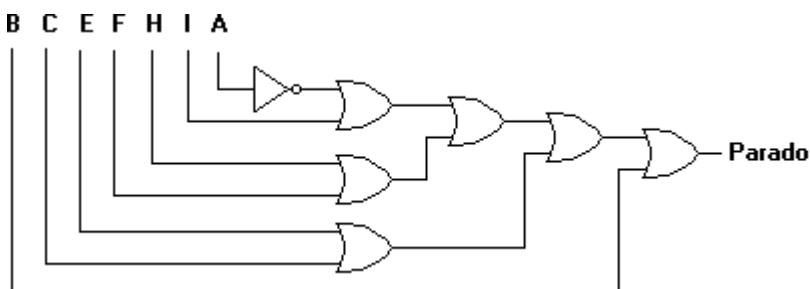
que muda muito pouco em relação ao segundo andar, resultando em:

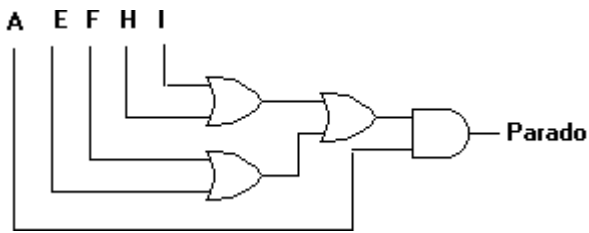
$$Parado = \overline{A \cdot B \cdot C \cdot E \cdot F \cdot H \cdot I} = \overline{A} + B + C + E + F + H + I$$

Sendo que consideramos:



O que nos leva ao seguinte circuito:





Corrigido para atender apenas ao 1° andar

(26/12/03 20:56:56)

2° andar

05/09/2003

a) Parado

A	B	C	D	E	F	G	H	I		
1°	2°	3°	1°	2°	3°	1°	2°	3°	Sobe	Desce
0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0	0
0	1	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0

Logo, a equação que resolve quando o elevador **não deve partir** é dada por:

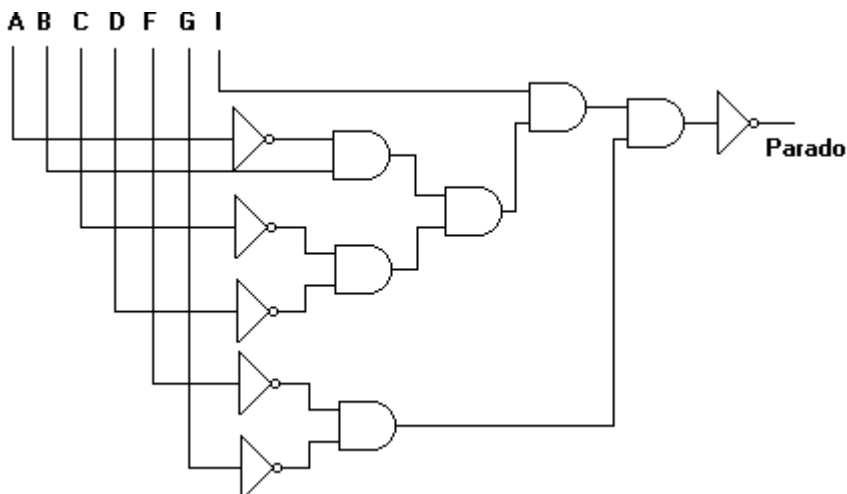
$$Parado = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{F} \cdot \overline{G} \cdot \overline{I} = A + \overline{B} + C + D + F + G + I$$

Considerando que:

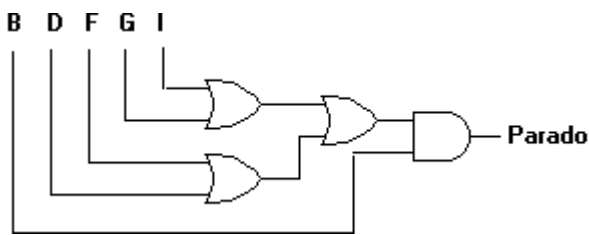
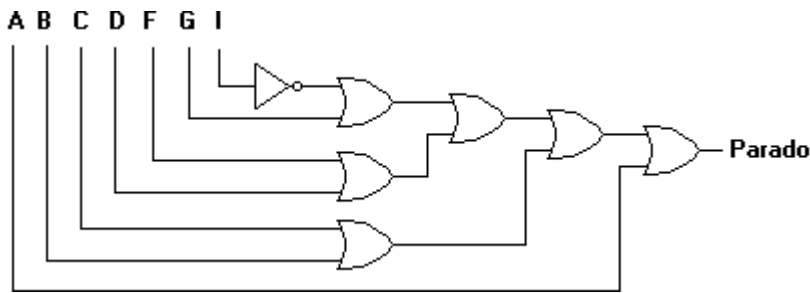
Sobe=0
Desce=0

} **Condição de parada**

O que resulta em:



Ou ainda, utilizando o teorema de Morgan, também poderíamos projetar da seguinte forma:



Corrigido para atender apenas ao 2° andar

(26/12/03 20:56:56)

b) Desce

Sensor			Andar			Elevador			Ação	
A	B	C	D	E	F	G	H	I	Sobe	Desce
1	2	3	1	2	3	1	2	3	Sobe	Desce
0	1	0	0	0	0	1	0	0	0	1
0	1	0	0	0	0	1	1	0	0	1
0	1	0	0	1	0	1	0	0	0	1
0	1	0	0	1	0	1	1	0	0	1
0	1	0	1	0	0	0	0	0	0	1
0	1	0	1	0	0	0	1	0	0	1
0	1	0	1	0	0	1	0	0	0	1
0	1	0	1	0	0	1	1	0	0	1
0	1	0	1	1	0	0	0	0	0	1
0	1	0	1	1	0	0	1	0	0	1
0	1	0	1	1	0	1	0	0	0	1
0	1	0	1	1	0	1	1	0	0	1

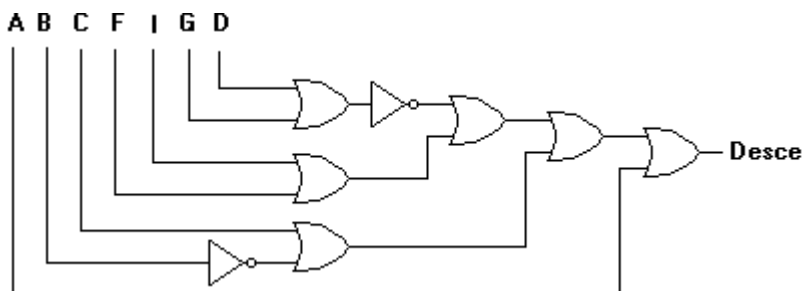
Observamos que as variações ocorrem apenas em D, E, G e H, um estudo deles sob o mapa de Karnaugh nos mostra que:

	DE			
GH	00	01	11	10
00			1	1
01			1	1
11	1	1	1	1
10	1	1	1	1

Cuja análise resulta em:

$$Desce = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{F} \cdot \overline{I} \cdot (D+G)} = A + \overline{B} + C + F + I + \overline{(D+G)}$$

Cujo circuito pode ser resumido a:

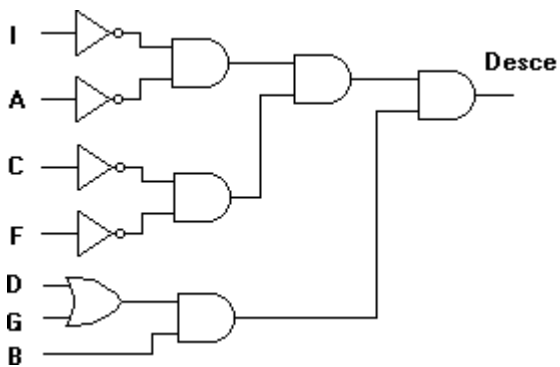


17/11/2003

Um novo problema precisou rever alguns conceitos, na verdade, a equação para que o elevador desça do 2º Andar resulta:

$$Desce = \overline{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{F} \cdot \overline{I} \cdot (D+G)}$$

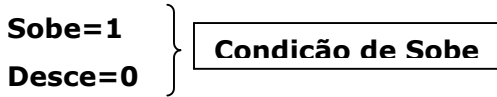
Cujo circuito pode ser resumido a:



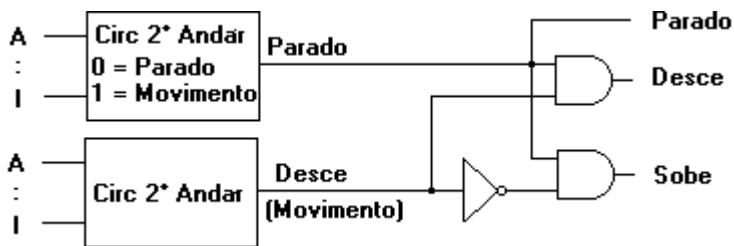
c) Sobe

16/09/2003

Considerando que:



E pudemos observar que, se o elevador para ou se o elevador desce, evidentemente ele não poderia estar subindo, logo consideramos o seguinte circuito:



3º Andar

O estudo do 3º andar é análogo aos outros.

3º Andar										
Sensor do andar			Botão andar			Botão elevador			Saída	
1º	2º	3º	1º	2º	3º	1º	2º	3º	Sobe	Desce
0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	1	0	0

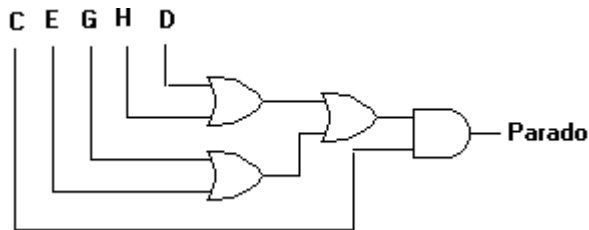
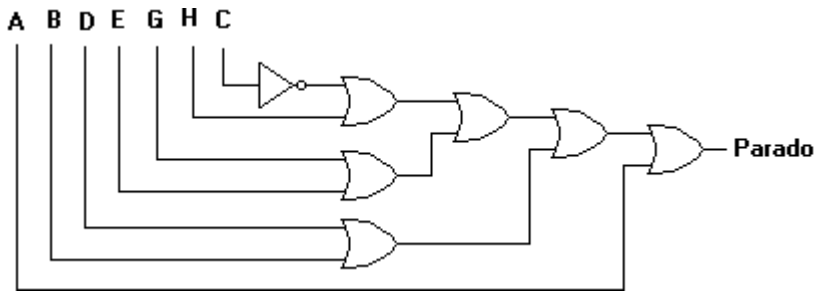
que também muda muito pouco em relação ao segundo andar, resultando em:

$$Parado = \overline{A.B.C.D.E.G.H}$$

Considerando que:



Isto nos leva ao seguinte circuito:



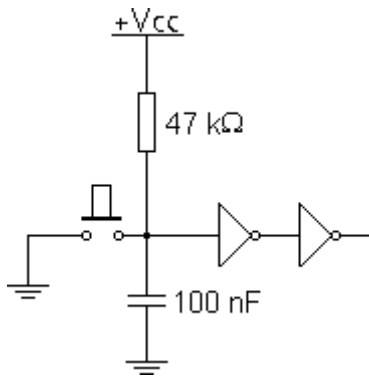
Corrigido para atender apenas ao 3° andar

(26/12/03 20:56:56)

Botões

22/09/2003

Cada botão empregado é mecânico, logo gera ruído. Para controle deste problema é necessário empregar um circuito de debounce. No nosso caso específico optamos pelo seguinte:

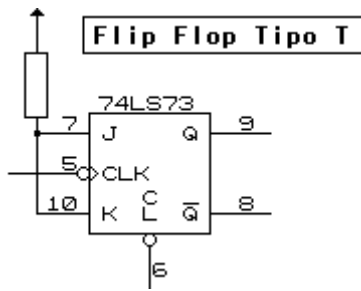


Quanto a esse problema, foi um dos mais simples.

26/09/2003

Um dos piores problemas enfrentados em projetos é desenvolver uma tese e ser surpreendido por algum detalhe que escapou ao raciocínio.

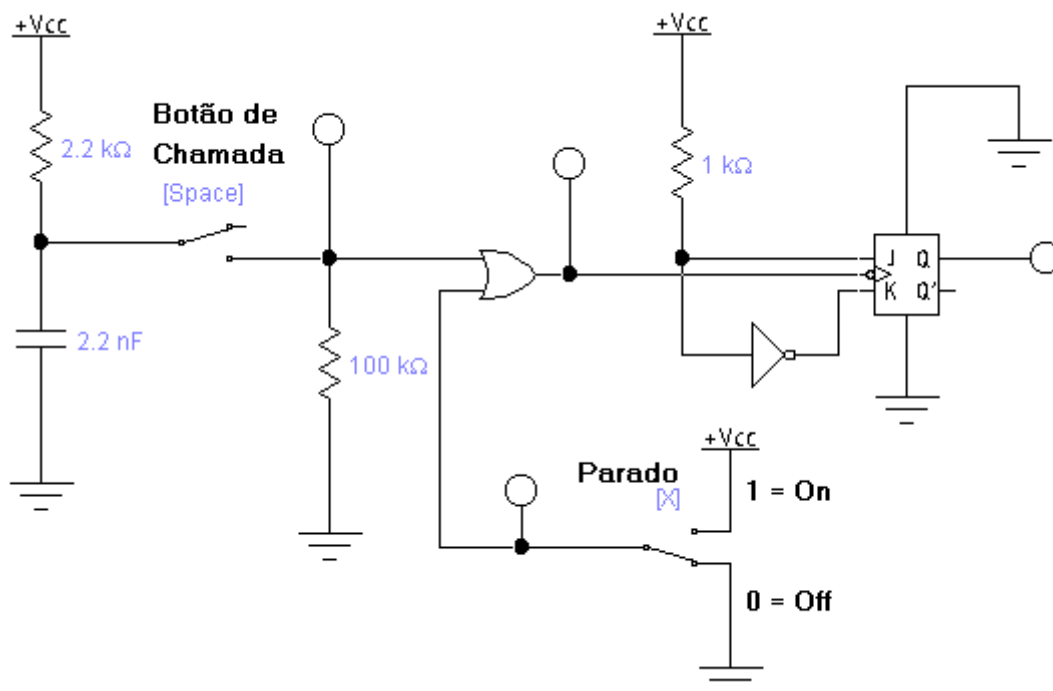
Na análise dos botões foi aventado o uso de um Flip-flop tipo T com um Flip-Flop JK:



Assim, se o usuário apertasse o botão do andar ou elevador, este colocaria nível "1" na saída Q e isto pode acender um LED. O inconveniente seria que se o usuário apertasse novamente o LED apaga, se novamente, acende e assim sucessivamente.

Até aí não há problema. Porém isto faria com que o usuário ficasse ligando e desligando o motor do elevador, além de acender o LED. Claro, isto o queimaria.

Na verdade o problema era complexo. Criar um botão de memória e que só disparasse e não mais alterasse sua saída até a chegada do elevador. Uma outra tentativa usando o Electronic WorkBench resultou em:



E simulamos o sinal de parada com uma chave. O maior fracasso foi o fato de que se o elevador estivesse parado não seria possível chamá-lo.

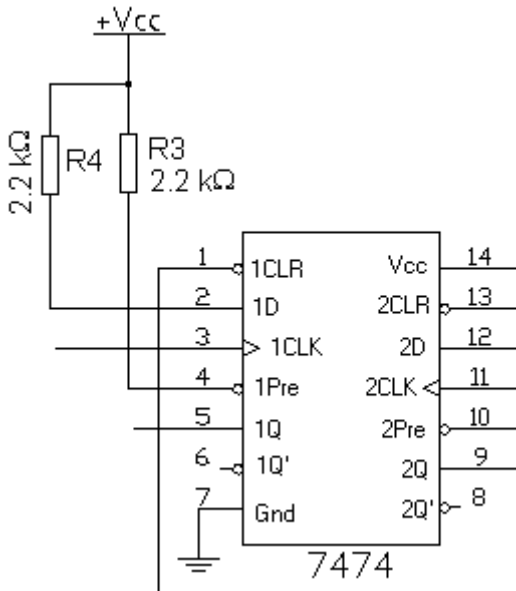
Outro candidato era o flip-flop tipo D. Conseguimos finalmente. O maior de todos os erros foi inserir o sinal **Parado** como referência. Na verdade o ideal seria considerar uma combinação que nos permitisse saber que o elevador chegou ao destino, assim, propusemos os seguinte:

14/10/03

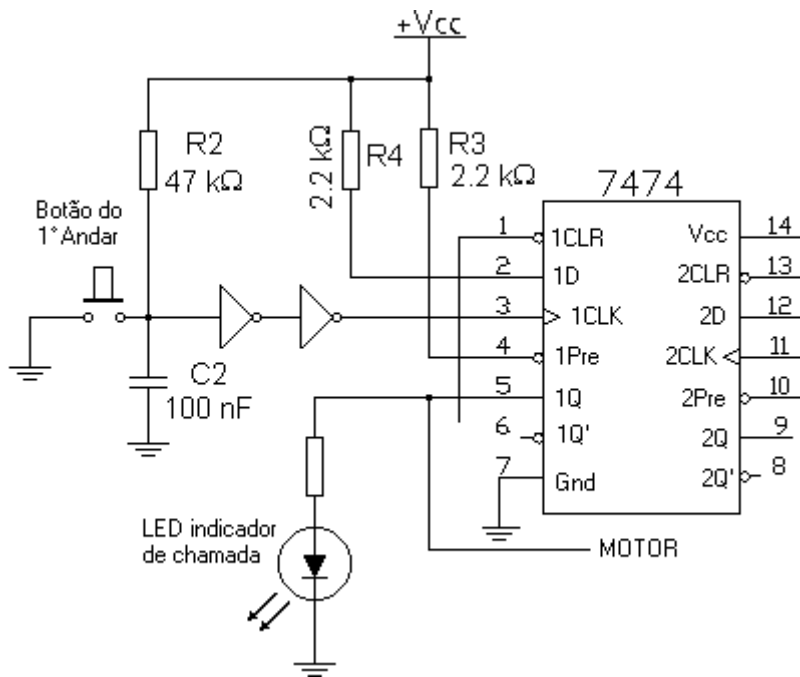
Observando a True table do CI 74LS74 temos:

74LS74						
	Entrada				Saída	
	Assíncrona		Síncrona		Q	\bar{Q}
	PR	CLR	CK	D		
Set Assinc.	0	1	X	X	1	0
Reset Assinc	1	0	X	X	0	1
instável	0	0	X	X	1	1
Set	1	1	↑	1	1	0
Reset	1	1	↑	0	0	1
	1	1	0	X	não varia	

Se PR=1 e CLR=1 podemos controlar Q=1 permitindo o disparo com CLK desde que coloquemos D em Vcc (Pull-up). Isto resolve o problema de colocar um nível "1" permanentemente na saída Q.



O sinal de CLK vem de qualquer chave de Chamada (Andar ou Elevador), da forma:



20/10/2003

Mas, e como garantir que a chegada no andar desejado coloque a saída em zero?

Considerando que:

Sensor dos andares			Botões da Parede do andar			Botões de dentro do elevador		
1°	2°	3°	1°	2°	3°	1°	2°	3°
A	B		D	E	F	G	H	I
					0			0
					0			1
					1			0

} Possíveis

Um breve estudo de uma condição fatalmente nos levaria á uma solução mais global. Consideramos o caso onde o elevador se encontrasse em qualquer andar menos no 3°. Logo, este somente pararia no 3° se uma das três situações acima ocorresse:

- 1) O botão da parede do 3° andar foi pressionada e o sensor do andar indica que o elevador está no 3° andar - condição de parada.
- 2) O botão do 3° andar no elevador foi pressionado e o sensor do andar indica que o elevador está no 3° andar - condição de parada.

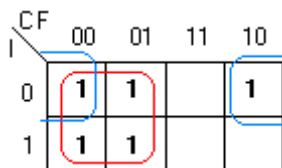
3) O botão da parede do 3° andar e o Botão do 3° andar no elevador foi pressionado e o sensor do andar indica que o elevador está no 3° andar - condição de parada.

Isto propiciou a criação da seguinte tabela verdade:

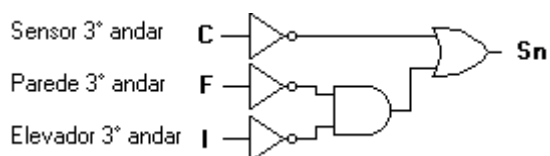
C	F	I	Sn
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

} Interessantes

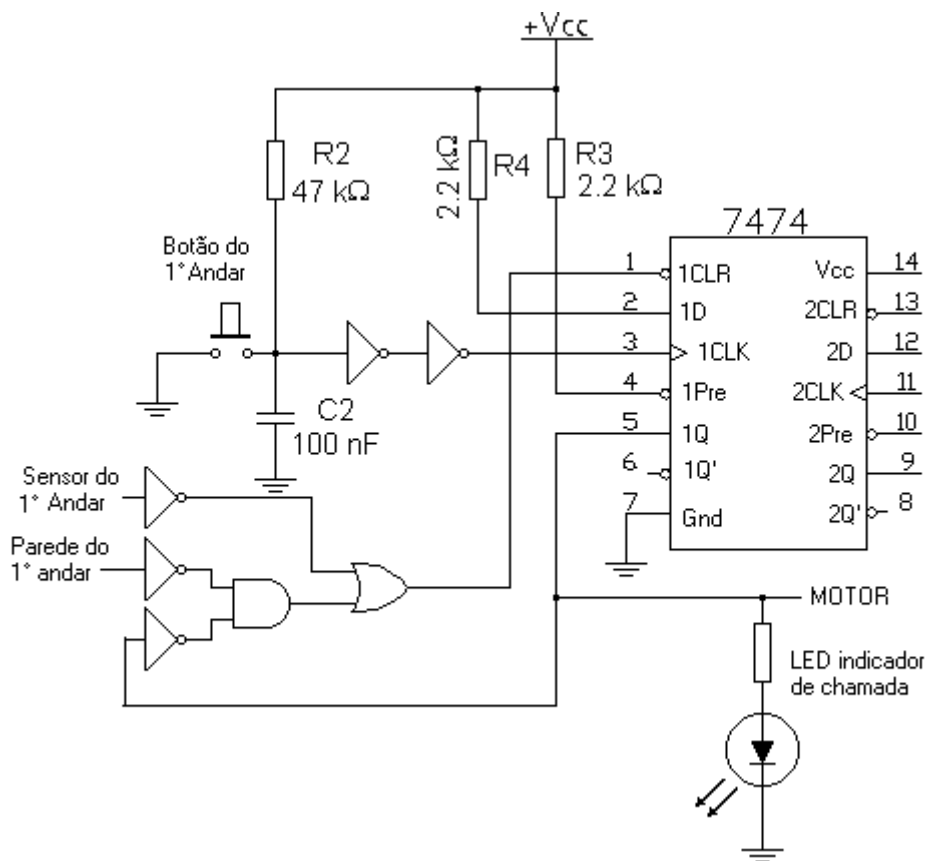
Cujo Mapa de Karnaugh resulta em:



Cujo circuito correspondente é:



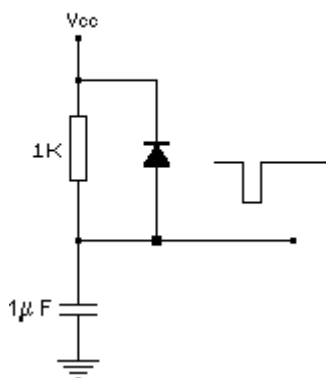
E o circuito se resume a:

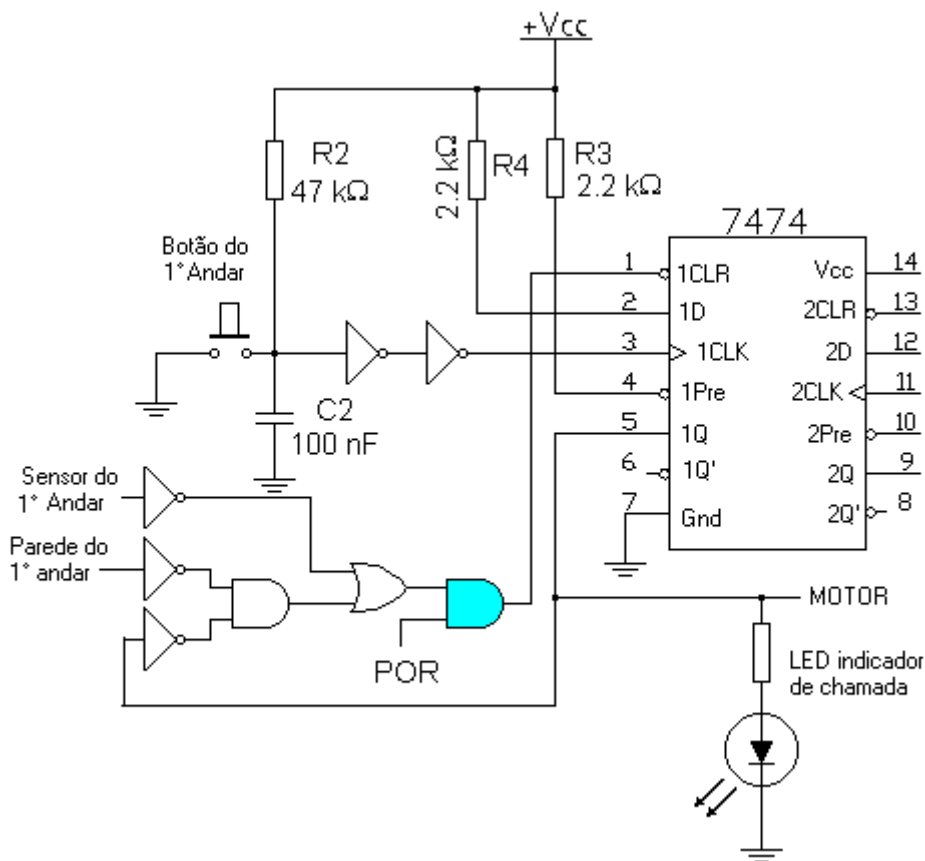


Ainda restavam dois problemas a serem considerados:

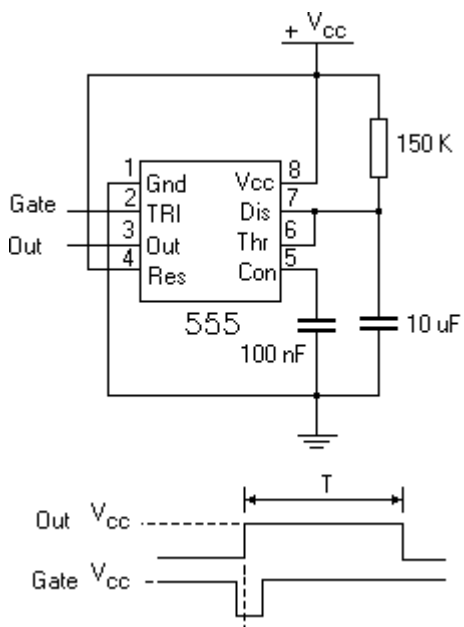
1º) E quando ligarmos o circuito ou ao religarmos após uma falta de energia? Como ele responde à energização? É possível que ao religarmos todos os andares e elevador acendam ao mesmo tempo, o que não seria interessante. Assim optamos por acrescentar um circuito de POR (Power On Reset) para sincronizar a energização.

Circuito de Power-On-Reset





2º) O tempo de resposta de circuitos digitais é muito rápido, levando determinadas condições ao limite. Consideramos o seguinte situação: Estou no 1º andar e simultaneamente o elevador é chamado ao 2º e 3º andar. Ambos tem a mesma prioridade e por conseguinte por ambos o elevador é acionado. Ao passar pelo 2º andar o motor deve parar para posteriormente seguir para o 3º andar. Claro, eu sei disso, mas e os circuitos. Nesse caso, o motor pararia por uma fração de segundos e detectaria que a situação de chamada é do 3º andar e simplesmente seguiria para lá praticamente não parando no 2º andar.



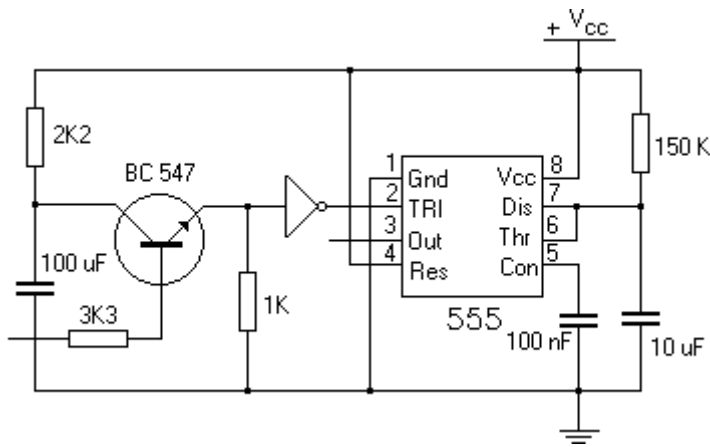
Na verdade o problema é TEMPO. Seria preciso parar algum tempo no 2º andar para posteriormente seguir. A solução proposta seria inserir um controle de tempo (Timer) no andar. Quando o elevador para no 2º andar, um timer é disparado por cerca de 15 segundos impossibilitando a re-ligação do motor, liberando-o mas apenas após 15 segundos.

segundos.

Ora, isso seria o tempo suficiente para que o SOLICITANTE do elevador abrisse a porta do andar impossibilitando que o motor de ligar (O projeto prevê que

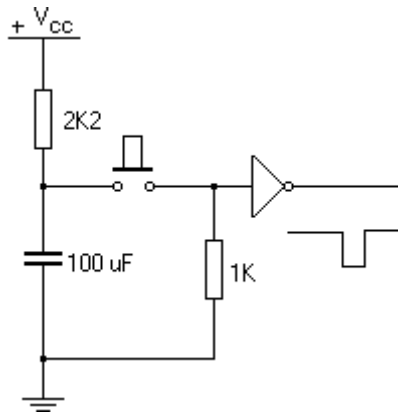
caso alguma das portas seja aberta ou não feche, sensores das portas seguindo um caminho separado impede o motor de ligar).

O timer será feito com um CI LM555 conectado na forma Monoestável conforme a figura seguinte:



Em uma primeira análise parecia simples, porém, de onde conseguir o pulso, uma vez que todo o sistema funciona em níveis e transição ?

A resposta consiste em um circuito gerador de pulso (Trigger).



23/10/2003

Bom. Porém ainda não encerra a questão. Quem é a chave push-botton que gera o pulso? Como de fato isso não existe, a solução foi a colocação de um transistor para a função. Mas isso ainda seria mais uma tentativa, não fosse o fato do circuito ter funcionado completamente no simulador.

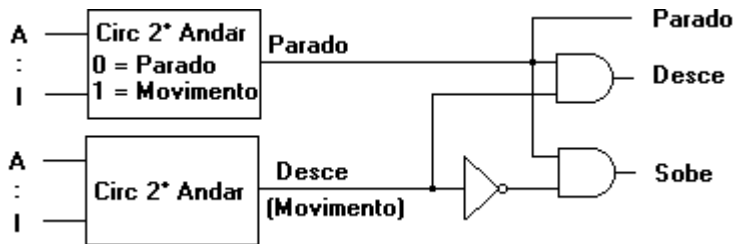
(Segue em anexo o diagrama esquemático completo do elevador.)

Agora o próximo passo é definir e testar o Motor.

28/12/03

Problema do comando de SOBE

Muitos problemas foram relacionados a lógica de SOBE para o 2º andar. Desde o princípio pareceu suspeita a simplicidade do circuito proposto. Cada andar, e o 2º andar não foge a regra, é composto na primeira análise de 64 possibilidades. Destas, 4 referem-se à condição de **não funcionar o motor**, 12 referem-se à condição de **DESCE** e as 48 restantes se referem à condição de **SOBE**. Partindo do circuito proposto inicialmente:

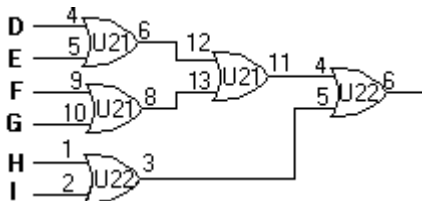


Embora o *timing* entre as mudanças de estado na saída de **sobe** e **desce** não fosse muito claro em momento de projeto, ficou provado, na prática, que ocorre um "impulso" de curtíssima duração na saída de **desce** durante um processo de subida, e mesmo muito pequeno, dispara o sinal de **sobe** criando uma situação anômala com **sobe** e **desce** ativos simultaneamente.

Além deste problema incômodo, havia também o problema de "como manter a condição de movimento". Isto porque em sistemas digitais, cada condição de entrada produz uma condição de saída *sui generis* (única) e portanto, assim que o elevador sai do repouso no andar, as condições dos sensores mudam e conseqüentemente as saídas. Em outras palavras, assim que o elevador sai, ele para.

A solução proposta para o sistema consiste em memorizar a condição de movimento e "resetar" a condição de parar.

Até aí, foi solucionado com a colocação de um flip flop tipo D (U13 - 7474) um canal para **sobe** e outro para **desce**. E para resolver o problema do "reset" dessa memória, adotamos o seguinte circuito:



Assim, sempre que uma tecla é pressionada, esse sensor tem nível **1** na saída. Isto também resolve outro problema: Quando o elevador vai, por exemplo, do 1º para o 3º com direito a parado no 2º, o timer paralisa o motor e 16 segundos depois ele libera o motor. Ora, como a memória ainda informa que uma tecla ainda está ativa, o sistema prossegue até o 3º andar.

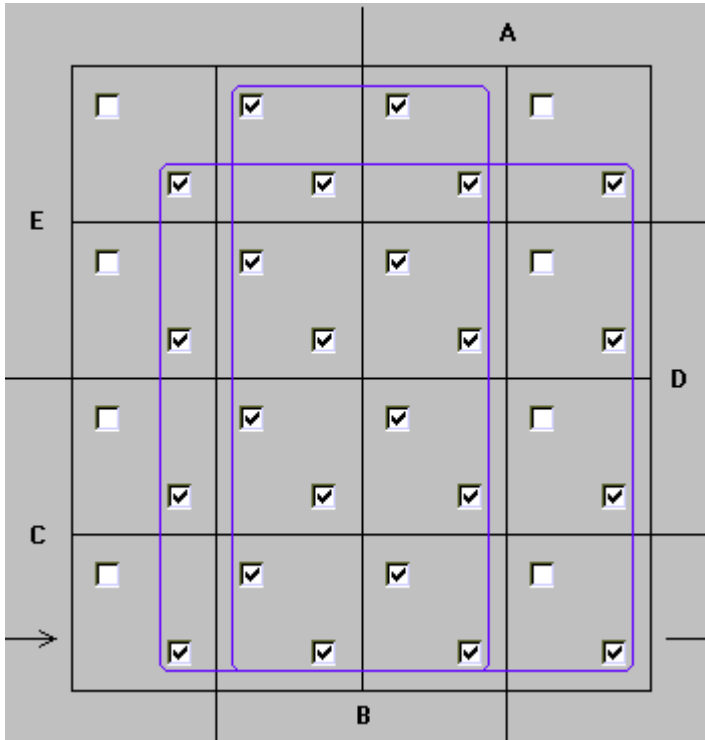
Agora, resolvido o problema de "dois andares solicitarem o elevador" restava o problema do ruído na lógica de **sobe** que disparava "acidentalmente".

Porém, para solucionar o problema, o único caminho parecia o impossível: analisar as 48 linhas lógicas da condição de **sobe**:

2º Andar										
Sensor do andar			Botão andar			Botão elevador			Saída	
A	B	C	D	E	F	G	H	I	Sobe	Desce
1º	2º	3º	1º	2º	3º	1º	2º	3º		
0	1	0	0	0	0	0	0	1	1	0
0	1	0	0	0	0	0	1	1	1	0
0	1	0	0	0	0	1	0	1	1	0
0	1	0	0	0	0	1	1	1	1	0
0	1	0	0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0	1	1	0
0	1	0	0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1	0	1	0
0	1	0	0	0	1	1	1	1	1	0
0	1	0	0	1	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	1	0
0	1	0	0	1	0	1	0	1	1	0
0	1	0	0	1	0	1	1	1	1	0
0	1	0	0	1	1	0	0	0	1	0
0	1	0	0	1	1	0	1	0	1	0
0	1	0	0	1	1	1	0	0	1	0
0	1	0	0	1	1	1	0	1	1	0
0	1	0	0	1	1	1	1	0	1	0
0	1	0	0	1	1	1	1	1	1	0
0	1	0	1	0	0	0	0	1	1	0
0	1	0	1	0	0	0	1	1	1	0
0	1	0	1	0	0	1	0	1	1	0
0	1	0	1	0	0	1	1	1	1	0
0	1	0	1	0	1	0	0	0	1	0
0	1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	1	1	0	1	0
0	1	0	1	0	1	1	1	1	1	0
0	1	0	1	1	0	0	0	1	1	0
0	1	0	1	1	0	0	1	1	1	0
0	1	0	1	1	0	1	0	1	1	0
0	1	0	1	1	0	1	1	1	1	0
0	1	0	1	1	1	0	0	0	1	0
0	1	0	1	1	1	0	0	1	1	0
0	1	0	1	1	1	1	0	0	1	0
0	1	0	1	1	1	1	1	0	1	0
0	1	0	1	1	1	1	1	1	1	0
0	1	0	1	1	1	0	0	0	1	0
0	1	0	1	1	1	0	0	1	1	0
0	1	0	1	1	1	1	1	1	1	0
0	1	0	1	1	1	0	0	0	1	0
0	1	0	1	1	1	0	0	1	1	0
0	1	0	1	1	1	1	1	1	1	0
0	1	0	1	1	1	0	0	0	1	0
0	1	0	1	1	1	0	0	1	1	0
0	1	0	1	1	1	1	1	1	1	0
0	1	0	1	1	1	0	0	0	1	0
0	1	0	1	1	1	0	0	1	1	0
0	1	0	1	1	1	1	1	1	1	0

0	1	0	1	1	1	0	1	1	1	0
0	1	0	1	1	1	1	0	0	1	0
0	1	0	1	1	1	1	0	1	1	0
0	1	0	1	1	1	1	1	0	1	0
0	1	0	1	1	1	1	1	1	1	0

Cujo mapa de Karnaugh, analisado pelo programa "KarnaugMap.exe Versão 4.4.5" resultou em:



Fazendo um paralelo entre E F G H I = A B C D E para adequá-lo ao programa, resultou em uma equação para a situação envolvendo essas colunas nas linhas onde a coluna referente ao Sensor da parede do 1º andar igual a 0 (D=0) resultou em:

$$W1 = \bar{D} \cdot (I + F)$$

e depois para as colunas (EFGHI) nas linhas onde D=1 resultou em:

$$W2 = D \cdot (I + F)$$

Ora, sendo a solução a soma das duas (W1+W2), então:

$$\bar{D} \cdot (I + F) + D \cdot (I + F) \Rightarrow (\bar{D} + D) \cdot (I + F) = I + F$$

E ainda considerando a ocorrência apenas no 2º Andar, teremos o seguinte circuito de **sobe**:

