

**UNIVERSIDADE FEDERAL DA FRONTEIRA SUL - UFFS**

**Ciência da Computação-2016/1**

**Professor Adriano Sanick Padilha**

## **Jogo de Par ou Ímpar**

Kadu Marcos Grando  
Adriel Schmitz J. de Paula  
Mayara Leticia Ogliari

**Chapecó-SC  
2016**

## **Resumo:**

Neste trabalho, o objetivo principal é a montagem de um jogo do tipo par ou ímpar. A princípio o trabalho era complexo pois, deveria tratar 6 variáveis na tabela verdade mas, como 2 variáveis não exerciam uma função importante tornou-se desnecessário a montagem da tabela verdade com essas 2 variáveis. Após a montagem da tabela com 4 expressões, iniciou-se a montagem mas, como a expressão, mesmo minimizadas ao máximo, gerariam muitos gastos portanto, a tabela verdade foi reformulada novamente mas, com somente 3 variáveis pois, removendo a 4ª variável não iria interferir no resultado final. Com a expressão final montada e minimizada iniciou-se a montagem. No princípio fora feita a montagem do circuito em simuladores online para depois montar o circuito na protoboard. Com o fim da montagem chegou-se a conclusão que é possível montar o circuito de um jogo do estilo par ou ímpar em uma protoboard de forma fácil.

**PALAVRAS-CHAVE.:** PROTOBOARD; CIRCUITO; JOGO;

## **SUMÁRIO:**

<b>1.Objetivo.....</b>	<b>6</b>
<b>2.introdução.....</b>	<b>7</b>
<b>3.Material.....</b>	<b>8</b>
<b>4.Jogo Par Ou Impar.....</b>	<b>9</b>
<b>5.Tempo Máximo De Atraso.....</b>	<b>19</b>
<b>6.Conclusão.....</b>	<b>20</b>
<b>7.Referências.....</b>	<b>21</b>
<b>8.Anexo.....</b>	<b>22</b>

## **Lista De Figuras:**

1.0 Simulação do circuito a partir das expressões booleanas.....	14
1.1 Imagem do Deep switch.....	14
2.0 Simulação em uma protoboard online.....	15
3.0 Circuito desligado após ser montado.....	15
3.1 Circuito informando uma jogada inválida.....	16
3.2 Jogador 1 vencedor.....	16
3.3 Jogador 2 vencedor.....	17

## **Lista de Tabelas:**

1.0 Primeira Tabela-Verdade.....	9
1.1 Mapa de Karnaugh da Primeira Saída.....	10
1.2 Mapa de Karnaugh da Segunda Saída.....	10
2.0 Segunda Tabela-Verdade.....	11
2.1 Tabela-Verdade Jogador 1.....	11
2.2 Tabela-Verdade Jogador 2.....	12
2.3 Tabela-Verdade Saída 1.....	12
2.4 Tabela-Verdade Saída Inválida.....	13

## **1. Objetivo**

Tem por objetivo a implementação de um jogo estilo par ou ímpar, através da utilização de uma protoboard e circuitos integrados.

## **2. Introdução**

O projeto a seguir utiliza-se da aplicação de conhecimentos teóricos na implementação de um jogo de par ou ímpar. Com a construção de uma tabela verdade obteve-se as possibilidades de jogadas e posteriormente encontrada as expressões simplificadas através da montagem do mapa de Karnaugh. Com a tabela verdade ordena-se que no momento em que o jogador 1 escolher a opção par e com a soma dos números jogados pelos dois jogadores for par, o LED do jogador 1 acenderá declarando-o vencedor, caso a soma dos números for ímpar o LED do jogador 2 será acesso o declarando como vencedor.

### 3. Material

<b>Materiais utilizados</b>	<b>Quantidade</b>
Fios de cabo Ethernet	
Protoboard	1
Deep Switch	1
CI 74LS86	1
CI 74LS04	1
CI 74LS32	1
LED's Vermelhos	2
Resistores de 1k	2
Resistores de 1.5k	6



#### 4. Jogo Par Ou Impar

No projeto em questão, foram feitas tabelas-verdade, mapas de karnaugh e a utilização da técnica de mintermos. Aplicou-se técnicas de simplificação algébrica, simulador de circuitos (Simulador Logisim) e simulador de protoboard online (Simulador TTL). Representou-se os números 0 para a opção par e 1 para a opção ímpar. Porém, para mostrar quem ganhou, utilizou-se o 1 (passagem de corrente elétrica) e 0 para o perdedor (ausência de corrente elétrica). No começo do trabalho foi feito uma tabela verdade para encontrar quem seria o vencedor, através da escolha entre par e ímpar e da escolha dos números.

Tabela 1.0:

Jogador 1		Jogador 2		Saída	
A	C	D	F	J1	J2
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	1	1
1	1	1	1	1	1

Foram usadas 4 variáveis para se chegar a um resultado. As variáveis **A** e **D** representavam a escolha entre par e ímpar (o que os jogadores iriam pedir), enquanto as variáveis **C** ou **F** representavam os números escolhidos pelos jogadores. Na saída, quando os dois jogadores escolhem a mesma opção de jogada (jogada inválida), as duas saídas recebem

1, ligando os dois LEDs e declarando a jogada inválida. Quando os jogadores escolhem par ou ímpar as variáveis **C** e **F** decidem quem é o vencedor.

Logo depois foi feito os dois mapas de Karnaugh dessa tabela.

Tabela 1.1:

Jogador 1	$\sim A * \sim C$	$\sim A * C$	$A * C$	$A * \sim C$
$\sim D * \sim F$	1	1	1	0
$\sim D * F$	1	1	0	1
$D * F$	0	1	1	1
$D * \sim F$	1	0	1	1

Onde foi encontrado a seguinte expressão:

$$J1 = \overline{A} * \overline{D} + \overline{A} * D + C * \overline{D} * \overline{F} + C * D * F + \overline{C} * \overline{D} * F + \overline{C} * D * \overline{F}$$

Tabela 1.2:

Jogador 2	$\sim A * \sim C$	$\sim A * C$	$A * C$	$A * \sim C$
$\sim D * \sim F$	1	1	0	1
$\sim D * F$	1	1	1	0
$D * F$	1	0	1	1
$D * \sim F$	0	1	1	1

Originando a seguinte expressão:

$$J2 = \overline{A} * \overline{D} + A * D + C * \overline{D} * F + C * D * \overline{F} + \overline{C} * \overline{D} * \overline{F} + \overline{C} * D * F$$

Como o circuito estava extenso e com isso, gastando muitos recursos para ser produzido, o projeto foi repensado adotando uma nova tabela verdade onde, foi utilizado três variáveis. Segue imagem da tabela:

Tabela 2.0:

A(J1)	D(J2)	S	J1	J2
0	0	0	X	X
0	0	1	X	X
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	X	X
1	1	1	X	X

Nessa tabela utilizou-se as variáveis **A** e **D** para representar a escolha dos jogadores e a variável **S** como opção de escolha de números feita pelos jogadores ou seja, entre par e ímpar, os jogadores terão 2 opções de jogada que são, 0 ou 1. Depois com a aplicação de minitermo nas duas tabelas verdades foram encontradas as expressões minimizadas do circuito, já que, o que o objetivo principal fora encontrar o vencedor, então a saída que continha 1 era a que interessava. O **X** na tabela representa que a entradas dos dois jogadores resultará em uma jogada inválida.

Na tabela 2.1, o jogador 1 é representado pela variável **A**, onde são representadas as possíveis combinações entre jogadas. A variável **S** representa a possibilidade de escolha dos números feitas pelo jogador e **J1** representa o resultado das jogadas. Conforme a tabela, se o jogador escolher 0 no caso, par, e escolher o número 0 como saída, o resultado será 1, acendendo o LED e considerando-o vencedor. Da mesma forma, acontecerá a vitória do jogador se ele escolher ímpar e jogar 1. Do contrario, se o jogador escolher par e jogar o número 1, o resultado será 0 indicando que o jogador não obteve vitória em sua jogada. O mesmo acontecerá se ele escolher ímpar e jogar 0. Abaixo, a imagem da tabela:

Tabela 2.1:

A	S	J1
0	0	1
0	1	0
1	0	0
1	1	1

Com o uso dos *mintermos*, encontra-se a seguinte expressão minimizada que simboliza uma *XNOR* (XOR com a saída negada):

$$J1 = (\bar{A} * \bar{S}) + (A * S)$$

A tabela 2.2 representa o jogador 2 e suas opções de jogo. A variável **D** representa as opções de escolha do jogador (par ou ímpar) enquanto a variável **S** representa as possíveis jogadas que o mesmo fará ou seja, quando o jogador escolher 0 ou 1 essa variável irá representá-lo. Já J2 é a variável que definirá o resultado final ou seja, se o jogador 2 escolher par e jogar o número 0, o resultado final será 1, ligando o LED e considerando-o vencedor. O mesmo acontece quando o jogador escolher ímpar e jogar 1. Já no caso de o jogador escolher par e jogar 1 o LED não acenderá considerando-o perdedor. O mesmo acontecerá se o jogador escolher ímpar e jogar 0.

Tabela 2.2:

D	S	J2
0	0	1
0	1	0
1	0	0
1	1	1

Sabendo que a variável **D** representa o jogador 2 e a variável **S** representa o número escolhido pelo jogador, com a aplicação dos *mintermos* encontra-se a expressão:

$$J2 = (\bar{D} * \bar{S}) + (D * S)$$

Uma *XNOR* (XOR negada a saída):

Depois de definir as opções de escolha, par ou ímpar, outra tabela foi feita para tratar os números que os jogadores escolherão, gerando a possível saída que está sendo representada por S, segue abaixo a tabela:

Tabela 3.0:

C(J1)	F(J2)	S
0	0	0
0	1	1
1	0	1
1	1	0

Nesta tabela usou-se 2 variáveis, a **C** que representa o jogador 1 e a variável **F** que representa o jogador 2, após a aplicação dos *mintermos* foi encontrado a seguinte expressão:

$$S = (\overline{C} * F) + (C * \overline{F}) \text{ Uma porta XOR.}$$

Como era preciso acender os LED's toda vez que os jogadores fizessem uma jogada inválida, uma tabela-verdade foi desenvolvida para fazer a relação entre as escolhas iniciais que cada jogador iria pedir. Toda vez que fosse uma combinação inválida, a saída é 1 fazendo os dois LED's acenderem. Com essa tabela é possível mostrar aos jogadores. Segue a tabela 4.0:

Tabela 4.0:

A	D	Saída
0	0	1
0	1	0
1	0	0
1	1	1

A letra **A** representa o pedido do jogador 1, a letra **D** o pedido do jogador 2 e a possível saída de acordo com a combinação entre os dois pedidos. Usando os *mintermos*, se obteve a expressão:

$$S = (\overline{D} * \overline{A}) + (A * D) \text{ Uma XNOR ( XOR negada a saída ) .}$$

Depois de feito todas estas tabelas verdades e encontrado todas essas expressões, uniu-se todas formando essas duas expressões:

$$\text{Jogador 1: } \overline{A \oplus (C \oplus F)} + \overline{A \oplus D}$$

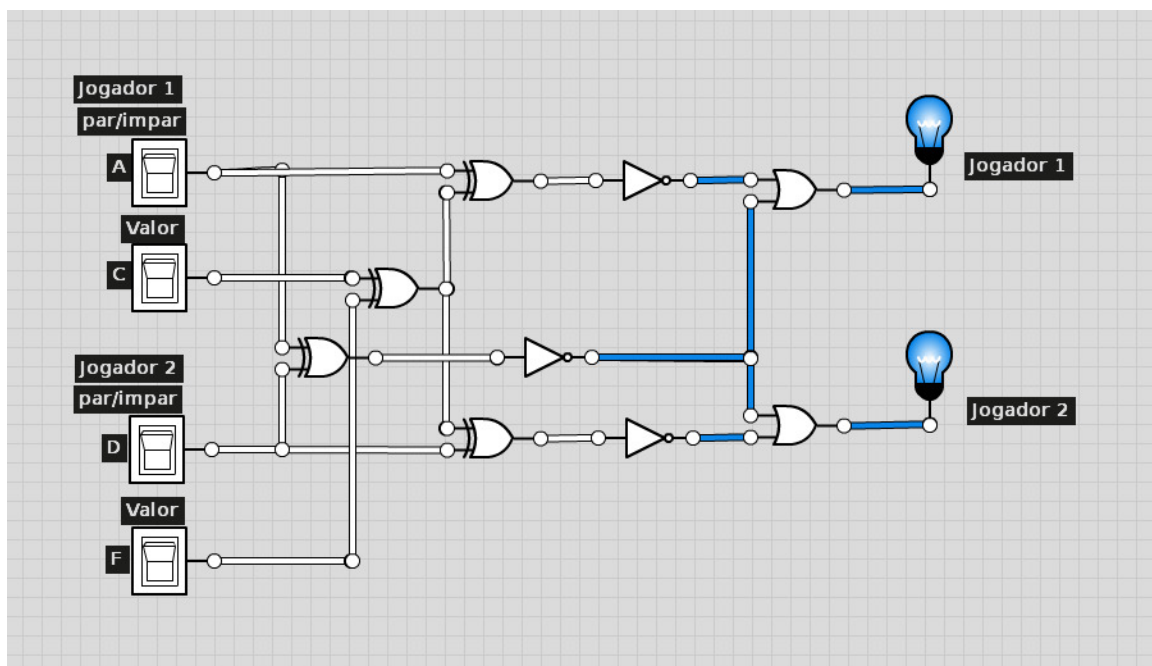
$$\text{Jogador 2: } \overline{D \oplus (C \oplus F)} + \overline{A \oplus D}$$

Como o pedido dos jogadores dependia diretamente da combinação dos números que os iriam jogar em seguida, foi feito **A** (jogador 1) multiplicando a possível combinação entre **C** e **F** ( **C** numero informado pelo jogador 1 e **D** número informado pelo jogador 2). O mesmo foi feito com o jogador 2.

Junção das duas expressões.

$$S = (\overline{A \oplus (C \oplus F)} + \overline{A \oplus D}) + (\overline{D \oplus (C \oplus F)} + \overline{A \oplus D})$$

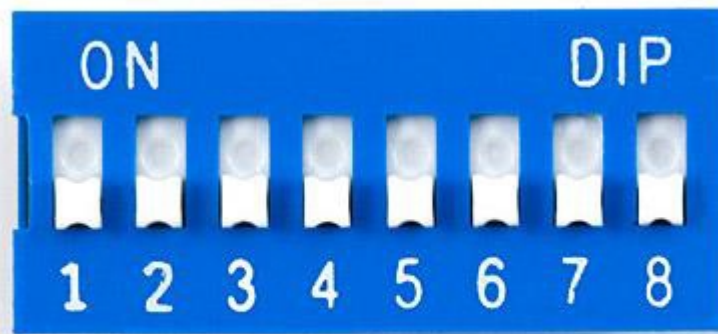
Figura 1.0. Fonte: Autoria própria.



**Nota:** Simulação do circuito a partir das expressões booleanas.

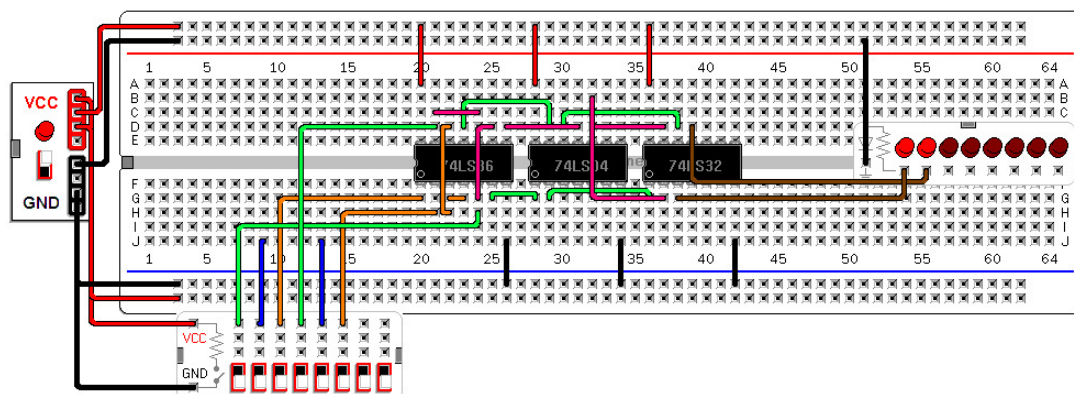
Para ajudar na hora da escolha das jogadas, utilizou-se um Deep Switch, nele os números 1 e 8 ficaram para a escolha dos jogadores (par ou ímpar) e os números 2 e 3 foram usados pelo jogador 1 para escolher o seu número e os números 6 e 7 foram usados para escolher os números do jogador 2.

Figura 1.1.



Fonte: <<http://corp-energysolutions.com/pontos-de-instalacao.html>>

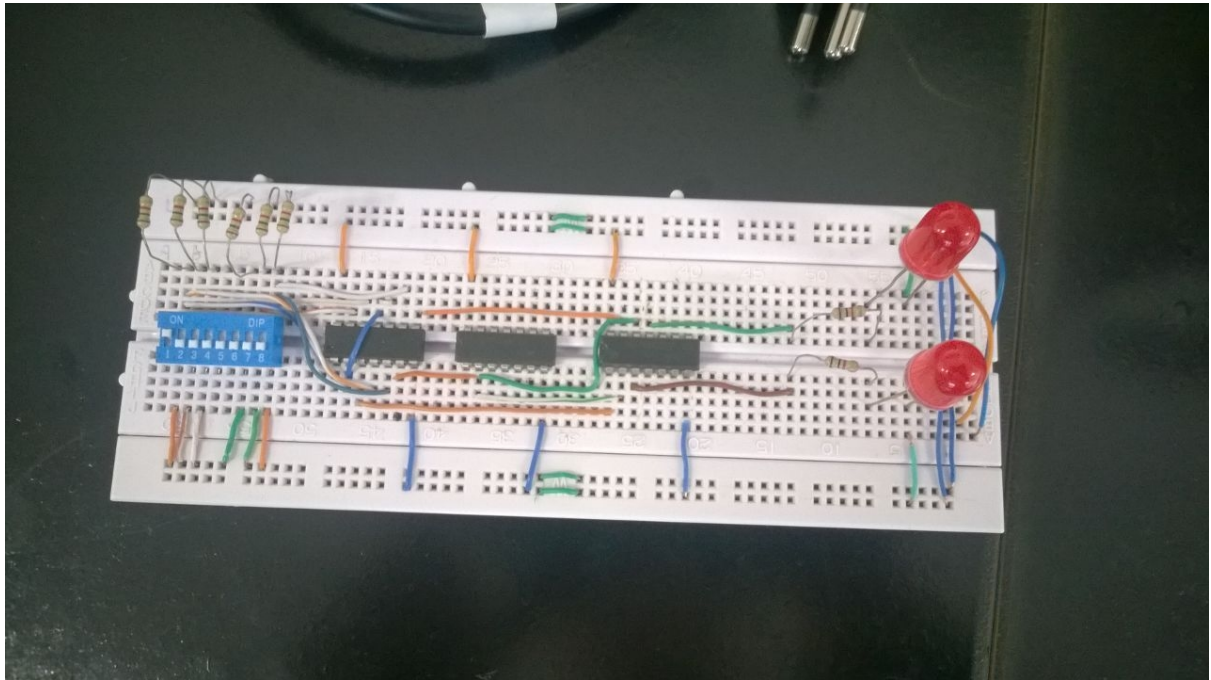
Figura 2.0. Fonte: Autoria própria.



**Nota:** Simulação em uma protoboard online.

Após a simulação online e feito testes, enfim o circuitos foi montado e testado. A montagem do circuito não exigiu muitos esforços. Para essa etapa do projeto contou-se com a ajuda de monitores para fazer a montagem do Deep Switch sem causar danos.

Figura 3.0. Fonte: Autoria própria.

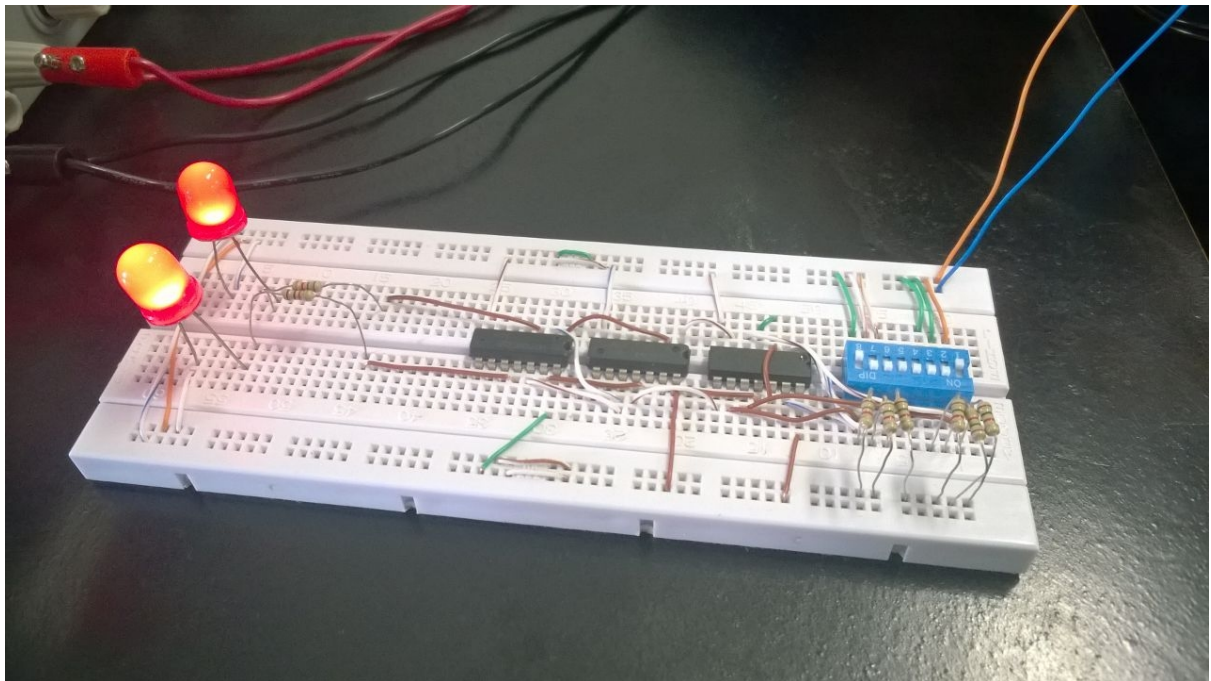


**Nota:** Circuito desligado após ser montado.

*Obs:* Foi recebida ajuda de um monitor da matéria de Circuitos Digitais que auxiliou em relação a utilização do Deep Switch, orientando sobre a utilização de resistores, quantidade e valores deles etc.

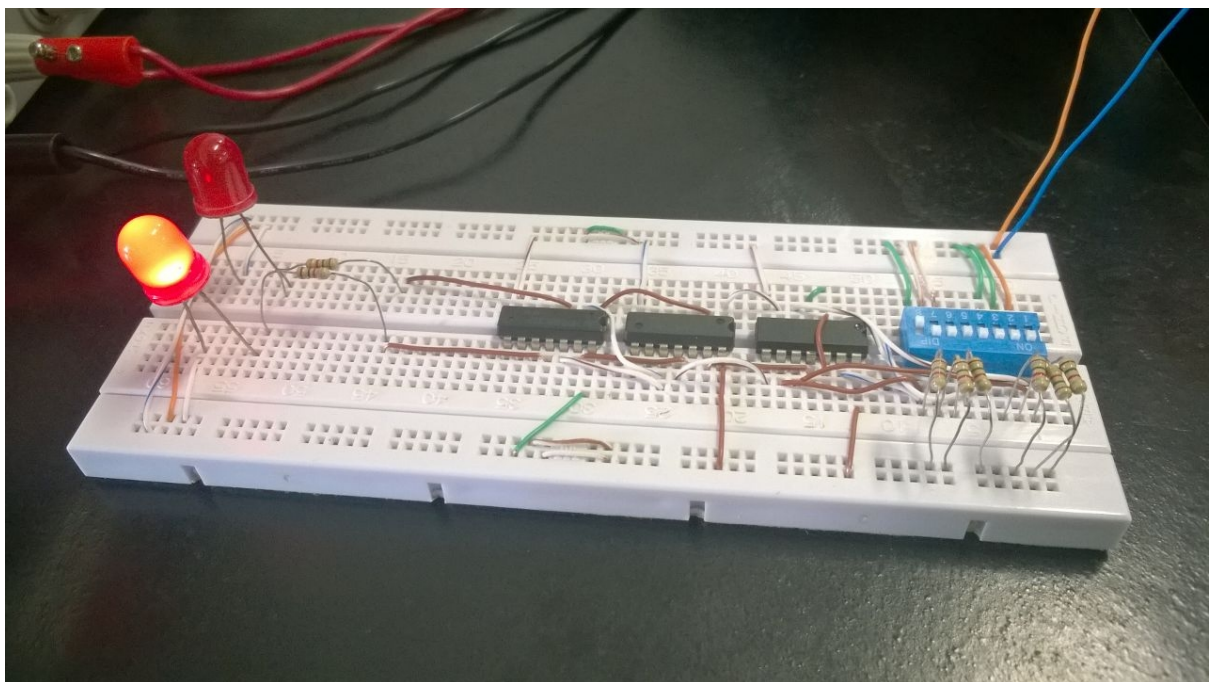


Figura 3.1. Fonte: Autoria própria.



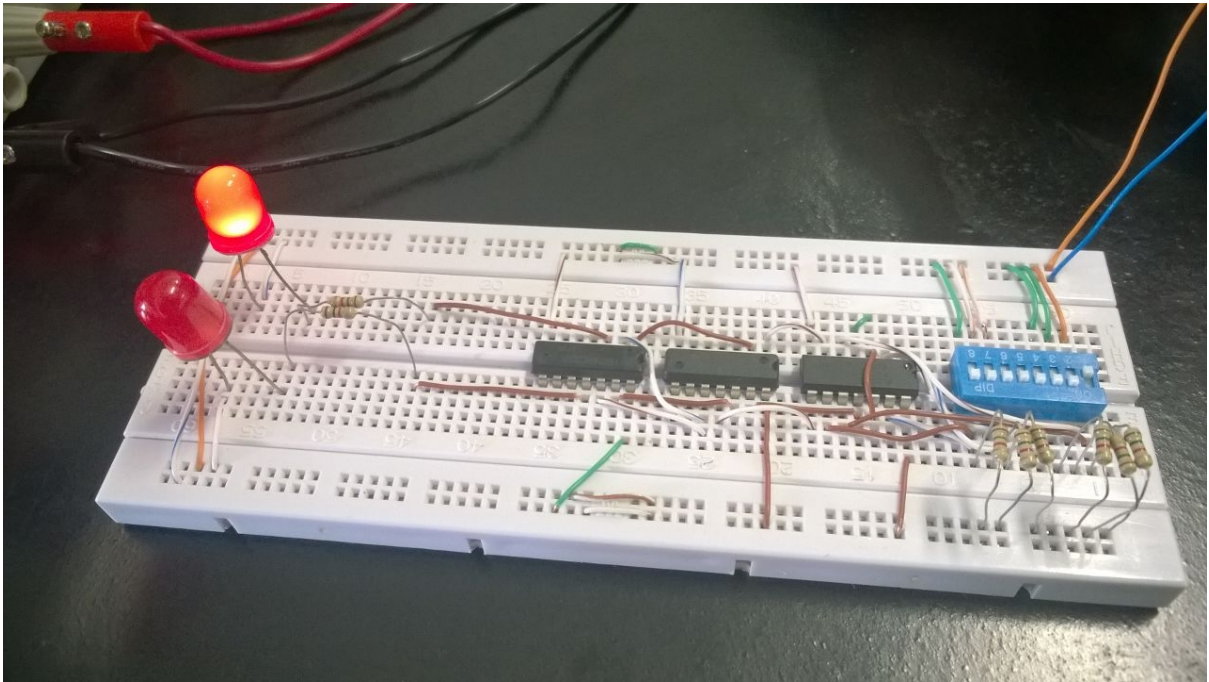
**Nota:** Circuito informando uma jogada inválida, pois os dois jogadores escolheram ímpar.

Figura 3.2. Fonte: Autoria própria.



**Nota:** O jogador 2 escolheu ímpar e o jogador 1 escolheu par, a soma dos números foi par declarando o jogador 1 como vencedor.

Figura 3.3.Fonte: Autoria própria.



**Nota:** O jogador 1 escolheu ímpar e o jogador 2 escolheu par, a soma dos números foi par declarando o jogador 2 como vencedor.

## 5. Tempo Máximo de atraso

Tendo em vista que foram usadas apenas portas *XORs*, *ORs* e *NOTs*. Sabendo que na porta XOR se tem um atraso máximo de 23 nanosegundos, na porta OR e NOT o atraso é de 15 nanosegundos. Conclui-se que, o caminho mais critico é das saídas C e F, pois o atraso máximo será de 76 nanosegundos.

## **6. Conclusão:**

Após testar muitas opções de tabelas verdade, solucionar muitos problemas, conhecer novas formas de montagem de circuito, chegou-se a uma tabela-verdade aonde o que importava era a escolha entre par e ímpar, escolha feita pelos dois jogadores e com uma terceira variável responsável pelo número escolhido pelos jogadores. Depois de feitas as tabelas verdade, tratar as entradas para obter as saídas e minimizar algebricamente as expressões obtidas através das saídas dos jogadores, a expressão algébrica foi montada em um simulador virtual de circuitos e posteriormente montada em um simulador online de protoboard. Após fazer esse processo o circuito foi montado na protoboard sem muita dificuldade mas, com a ajuda de monitores para esclarecimento. Finalizada a montagem conseguiu se comprovar que é possível implementar o jogo do tipo par ou ímpar em uma protoboard sem muita dificuldade.

## 7. Referências:

**Boolean Expression Calculator.** Disponível em: < <http://electronics-course.com/boolean-algebra>>. Acesso em: 18 Maio 2016.


**Futurlec.** Disponível em: < <http://www.futurlec.com/IC74LS00Series.shtml>>. Acesso em: 18 de Maio de 2016.

**Normas para Elaboração de um Relatório Técnico.** Disponível em <[http://portal.ifrn.edu.br/antigos/campi/copy\\_of\\_natalcentral/arquivos/links-de-noticias/NORMAS%20PARA%20ELABORACAO%20DE%20UM%20RELATORIO%20TECNICO.pdf](http://portal.ifrn.edu.br/antigos/campi/copy_of_natalcentral/arquivos/links-de-noticias/NORMAS%20PARA%20ELABORACAO%20DE%20UM%20RELATORIO%20TECNICO.pdf)>. Acesso em: 01 de Junho de 2016.

Soldán, A. J. M. de P. P. **Simulador de Construcción de Circuitos Digitales con Escenarios Virtuales y Tutoriales Interactivos.** Disponível em: < <http://www.tourdigital.net/SimuladorTTLconEscenarios.htm>>. Acesso em: 17 de Maio de 2016.

## 8.Anexos

### Datashet 74LS86 (XOR)



**DM74LS86**  
**Quad 2-Input Exclusive-OR Gate**

**General Description**  
This device contains four independent gates each of which performs the logic exclusive-OR function.

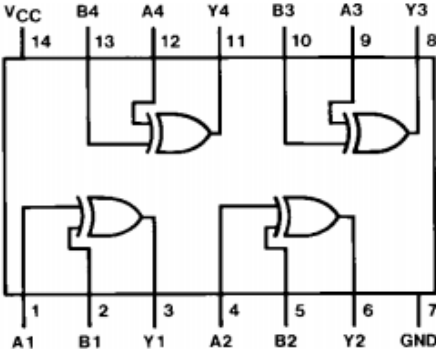
August 1986  
Revised March 2000

**Ordering Code:**

Order Number	Package Number	Package Description
DM74LS86M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS86SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS86N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

**Connection Diagram**



**Function Table**

$$Y = A \oplus B = \bar{A}B + A\bar{B}$$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level  
L = LOW Logic Level


### Switching Characteristics

at  $V_{CC} = 5V$  and  $T_A = 25^\circ C$

at $V_{CC} = 5V$ and $T_A = 25^\circ C$							
Symbol	Parameter	Conditions	$R_L = 2\text{ k}\Omega$				Units
			$C_L = 15\text{ pF}$		$C_L = 50\text{ pF}$		
			Min	Max	Min	Max	
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	Other Input		18		23	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	Low		17		21	ns
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	Other Input		10		15	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	High		12		15	ns

Disponível em: <<http://www.futurlec.com/74LS/74LS86.shtml>>. Acesso em: 23 de maio de 2016.

## Datashet 74LS32(OR)



**FAIRCHILD**  
SEMICONDUCTOR™

June 1986  
Revised March 2000

# DM74LS32

## Quad 2-Input OR Gate

### General Description

This device contains four independent gates each of which performs the logic OR function.

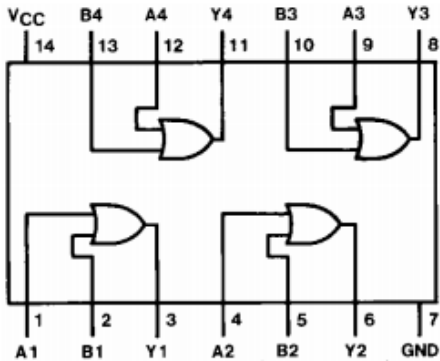
---

### Ordering Code:

Order Number	Package Number	Package Description
DM74LS32M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS32SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS32N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

### Connection Diagram



### Function Table

$Y = A + B$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	H

H = HIGH Logic Level  
L = LOW Logic Level

## Switching Characteristics


at  $V_{CC} = 5V$  and  $T_A = 25^\circ C$

Symbol	Parameter	R <sub>L</sub> = 2 kΩ				Units
		C <sub>L</sub> = 15 pF		C <sub>L</sub> = 50 pF		
		Min	Max	Min	Max	
t <sub>PLH</sub>	Propagation Delay Time LOW-to-HIGH Level Output	3	11	4	15	ns
t <sub>PHL</sub>	Propagation Delay Time HIGH-to-LOW Level Output	3	11	4	15	ns



Disponível em: <<http://www.futurlec.com/74LS/74LS32.shtml>>. Acesso em: 23 de maio de 2016.

## Datashet CI 74LS04(Not)



**FAIRCHILD**  
SEMICONDUCTOR™

August 1986  
Revised March 2000

# DM74LS04

## Hex Inverting Gates

### General Description

This device contains six independent gates each of which performs the logic INVERT function.

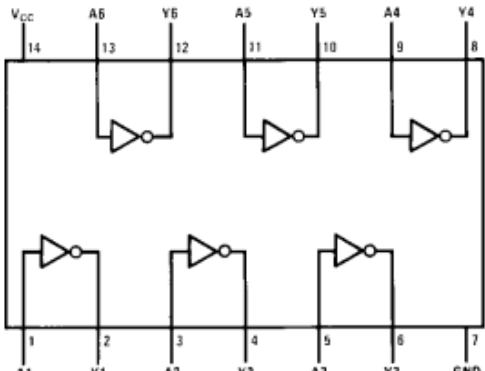
---

### Ordering Code:

Order Number	Package Number	Package Description
DM74LS04M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS04SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS04N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

### Connection Diagram



### Function Table

$Y = \bar{A}$

Input	Output
A	Y
L	H
H	L

H = HIGH Logic Level  
L = LOW Logic Level

Switching Characteristics						
at $V_{CC} = 5V$ and $T_A = 25^{\circ}C$						
Symbol	Parameter	$R_L = 2\text{ k}\Omega$				Units
		$C_L = 15\text{ pF}$		$C_L = 50\text{ pF}$		
		Min	Max	Min	Max	
$t_{PLH}$	Propagation Delay Time LOW-to-HIGH Level Output	3	10	4	15	ns
$t_{PHL}$	Propagation Delay Time HIGH-to-LOW Level Output	3	10	4	15	ns

Disponível em: <<http://www.futurlec.com/74LS/74LS04.shtml>>. Acesso em: 23 de maio de 2016.