



Circuitos Digitais (116351) – 1º Experimento

FAMILIARIZAÇÃO COM O PAINEL

OBJETIVO: Fornecer ao aluno um contato inicial com o painel. Os módulos periféricos básicos são apresentados e usados em alguns circuitos elementares. São, ainda, apresentadas as portas **AND**, **OR** e **NOT**.

1. INTRODUÇÃO TEÓRICA

Sistemas digitais são aqueles cujas variáveis assumem valores discretos, ou compreendidos entre certos níveis fixos. Esta noção deve ser familiar para a maioria de nós, pois ultimamente esse tipo de sistema vem ganhando importância crescente em todas as áreas tecnológicas e até mesmo em nossas atividades diárias.

1.1. CIRCUITOS DIGITAIS

Os sistemas digitais são implementados na prática principalmente por meio de circuitos eletrônicos. A informação é nesse caso representada por tensões que podem assumir apenas dois níveis. Tais sistemas são ditos binários.

Considere por exemplo o circuito da **Figura 1**, que tem duas entradas e uma saída, e se destina a apresentar como resposta uma função lógica dos sinais (tensões) nas entradas. Seu funcionamento é o seguinte:

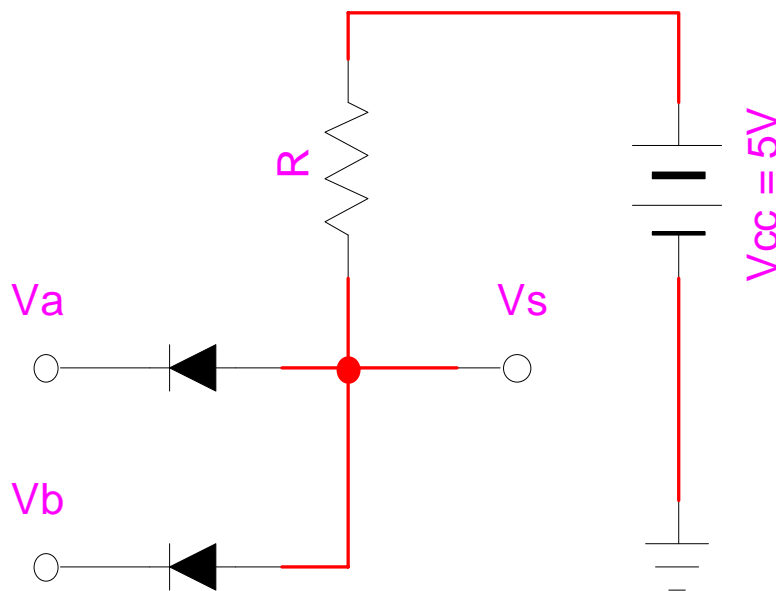


Figura 1 – Circuito digital com diodos

- I) As tensões v_A e v_B só podem assumir dois níveis: cerca de 0 volt ou cerca de 5 volts.
- II) Se uma tensão de 0 volt for aplicada a qualquer das entradas, o diodo correspondente conduzirá, e a tensão v_S valerá praticamente 0 volt.
- III) Se ambas as tensões estiverem em 5 volts (ou acima), nenhuma corrente circulará em R e, portanto v_S será de 5 volts.

Observe que o nível baixo de v_S não é exatamente 0 volt, e certamente varia quando se tem v_A em zero ou v_B em zero, ou ambos, pois os dois diodos não podem ser idênticos. Entretanto, essa pequena indefinição não nos impede de distinguir o nível baixo do nível alto. Num circuito complexo, como o de um computador, diversos fatores causam flutuações, o que igualmente não chega a ser um problema, a menos que as flutuações sejam grandes a ponto de inverterem o nível de algum sinal. Essa característica, denominada **imunidade a ruído**, é uma das maiores vantagens dos circuitos digitais sobre os analógicos.

Devido a sua natureza, os circuitos digitais são interpretados em termos de variáveis e funções lógicas. Existem duas convenções em uso. Na chamada **lógica positiva**, o nível alto de tensão é associado ao nível lógico 1 (verdadeiro), e o nível baixo ao nível lógico 0 (falso). Na **lógica negativa** o ocorre o contrário. No circuito da **Figura 1**, por exemplo, podemos representar as tensões v_A , v_B e v_S pelas variáveis lógicas A, B e S respectivamente. Nesse caso, dentro da convenção de lógica positiva, temos $S = 1$ se e somente se $A = 1$ e $B = 1$. O circuito executa, portanto, a operação **AND**. Caso fosse adotada a lógica negativa, teríamos $S = 0$ se e somente se $A = 0$ e $B = 0$. O circuito executa a operação **OR**.

A convenção de lógica positiva é quase sempre a preferida na prática e será usada neste conjunto de experiências.

1.2. CIRCUITOS TTL

Existem diversos tipos de circuitos capazes de executar funções lógicas. Os circuitos integrados que usaremos pertencem à família de circuitos TTL (*Transistor-Transistor-Logic*), que são os mais usados atualmente em sistemas de pequeno e médio porte. Os circuitos TTL são alimentados com uma tensão de 5 volts, e os níveis lógicos são definidos como se mostra a **Figura 2**. Observe a diferença entre os níveis de entrada e de saída. O fabricante garante que a saída de um circuito TTL estará entre 0 e 0,4 volts, quando no nível lógico 0. Por outro lado, ele garante também que qualquer tensão entre 0 e 0,8 volts aplicada a uma entrada será interpretada como nível lógico 0. Conseqüentemente, há um intervalo de 400 mV de **margem de ruído** para o nível lógico 0. Significa que um ruído de até 400 mV pode ser adicionado à saída de um circuito sem perturbar o funcionamento dos circuitos ligados àquela saída. No nível lógico 1 a situação é parecida.

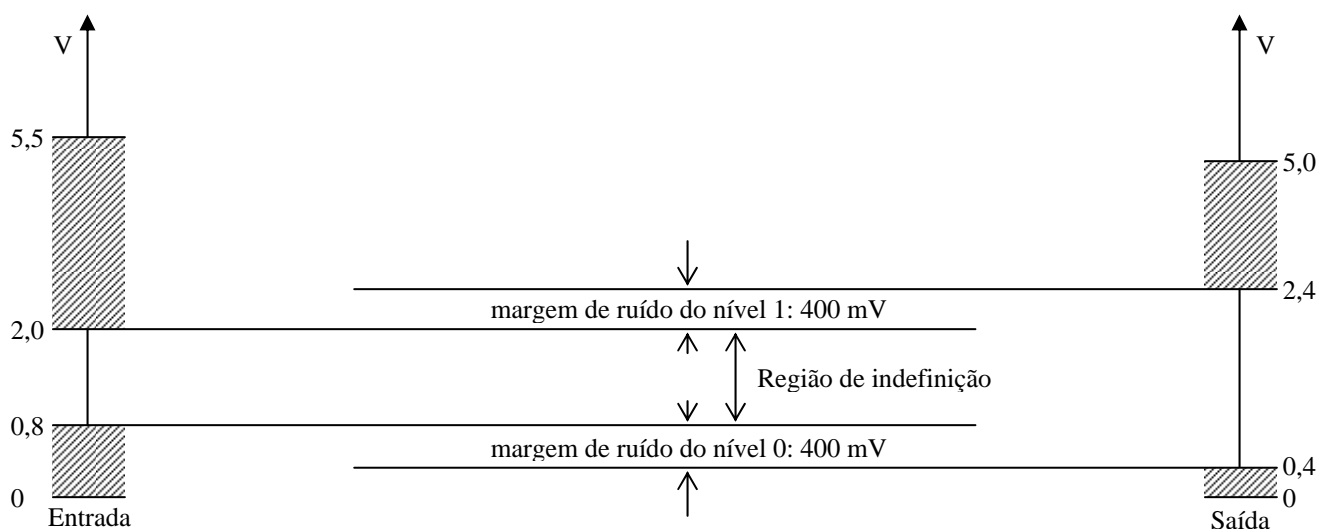


Figura 2 – Níveis lógicos de entrada e saída da família TTL, na convenção de lógica positiva

Veja, no entanto, que existe uma separação entre os níveis alto e baixo na entrada que é uma região de **indefinição**. Para ser corretamente interpretado, um sinal não deve permanecer nesta região a não ser durante uma rápida transição. Sinais negativos, ou excedendo 5,5 volts podem causar destruição do circuito integrado.

1.3. PORTAS AND, OR E NOT

Circuitos destinados a executar operações lógicas são denominados **portas**. As operações lógicas básicas são AND, OR e NOT, definidas conforme as tabelas abaixo. Essas tabelas são chamadas **tabelas da verdade**. Elas simplesmente descrevem o resultado da operação sobre cada combinação possível de operandos.

As tabelas abaixo mostram também as notações algébricas correspondentes.

A	B	$S = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Tabela I – AND

A	B	$S = A + b$
0	0	0
0	1	1
1	0	1
1	1	1

Tabela II – OR

A	$S = \bar{A}$
0	1
1	0

Tabela III – NOT

A expressão $A \cdot B$ lê-se “A e B”, $A + B$ lê-se “A ou B” e \bar{A} lê-se “não A” ou “A barra”. É comum omitir-se o ponto na notação da operação AND; pode-se escrever (e ler) $S = AB$.

As portas que realizam estas operações são respectivamente as portas AND, OR e NOT (E, OU e INVERSORA; sendo a última também chamada de NÃO ou NEGAÇÃO). Os símbolos usados em esquemas estão desenhados na **Figura 3**. Observe que as portas AND e OR podem ter mais do que dois terminais de entrada, sendo que o significado dessa extensão é óbvio. A porta NOT só tem um terminal de entrada.

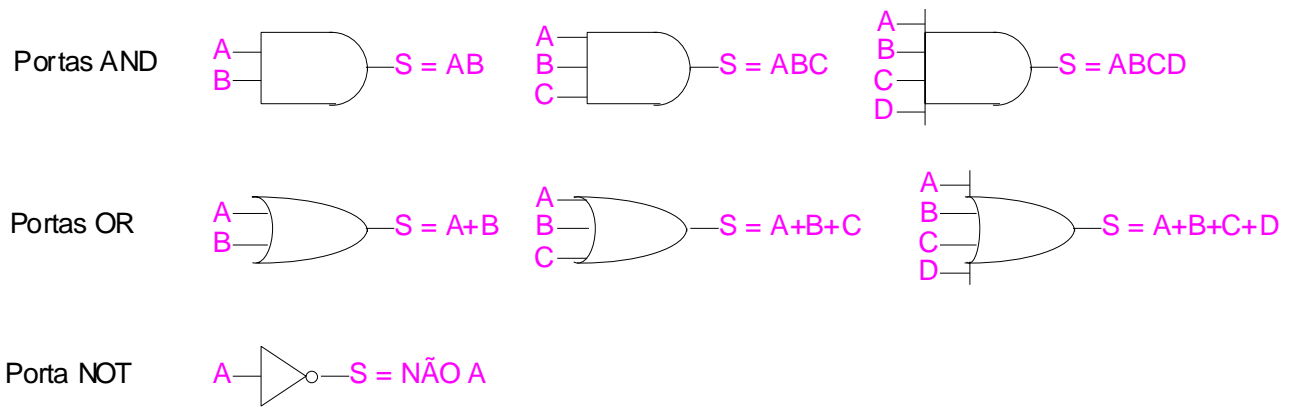


Figura 3 – Símbolos lógicos das portas AND, OR e NOT

As portas são implementadas com circuitos integrados (CI's), e cada CI contém em geral mais de uma porta. O CI número 7408, por exemplo, da família TTL, tem 14 pinos. Dois entre eles destinam-se a alimentação, e os 12 restantes dão acesso a 4 portas AND de 2 entradas, que podem ser usadas independentemente (veja **Figura 4**).

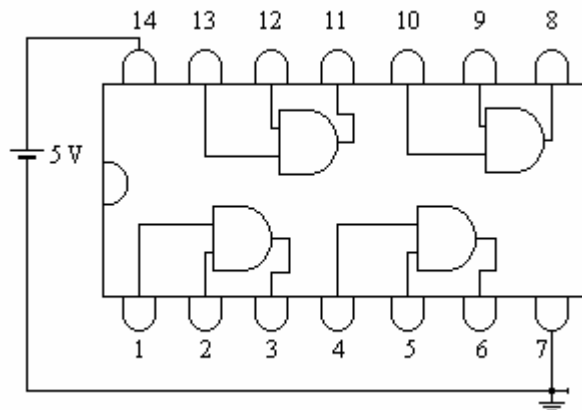


Figura 4 – Identificação dos terminais do CI 7408

As portas são interconectadas para executar as mais diversas operações lógicas. A **Figura 5** mostra como a operação OR pode ser implementada apenas com portas AND e NOT. Similarmente, a operação AND também pode ser implementada apenas com portas OR e NOT. Por outro lado, não é possível implementar uma porta NOT com portas AND e OR.

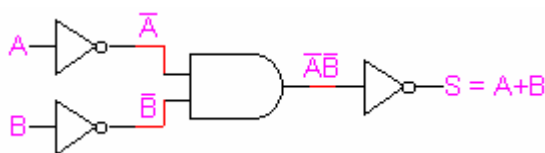


Figura 5 – Implementação da operação OR com portas AND e NOT

A	B	\bar{A}	\bar{B}	$\bar{A} \cdot \bar{B}$	$\overline{\bar{A} \cdot \bar{B}}$	$A + B$
0	0	1	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	0	1	1

Tabela IV – A comprovação de que $\overline{\bar{A} \cdot \bar{B}} = A + B$ é feita comparando-se suas tabelas da verdade

É possível demonstrar que qualquer operação lógica pode ser realizada apenas com portas AND e NOT, ou apenas com portas OR e NOT. Conjuntos de portas com esta propriedade de “auto-suficiência” são ditos universais.

A interpretação puramente lógica dos circuitos digitais é conveniente por sua simplicidade. Entretanto, não devemos nos esquecer completamente da natureza física das portas representadas pelos símbolos lógicos. Uma consideração muito importante é o atraso de propagação das portas, isto é, o tempo necessário para que sua saída mude, depois que uma entrada mudou. Quando diversas portas são ligadas em cascata, o atraso total de propagação é igual à soma dos atrasos em cada porta. Na família TTL, as portas têm um atraso típico da ordem de 10 ns (nano-segundo = 10^{-9} segundo). Desse modo, a porta OR da **Figura 3** e o circuito da **Figura 5** são idênticos do ponto de vista lógico, mas têm atrasos de propagação diferentes: cerca de 10 ns e 30 ns, respectivamente.

Os atrasos de propagação estabelecem um limite superior para a velocidade de operação de qualquer sistema digital.

1.4. MÓDULOS PERIFÉRICOS

Além dos módulos de circuitos integrados, usaremos um painel de experiência que conta com alguns módulos destinados à entrada e saída de dados. Estes módulos periféricos serão usados ao longo de todas as experiências a serem realizadas. Faremos aqui uma breve descrição. (Lembramos que a descrição detalhada de todos os módulos encontra-se no manual técnico do painel).

1.4.1. Chaves de 1 posição

Consistem basicamente de chaves de pressão com contatos de transferência. Destina-se à aplicação manual de pulsos num sistema.

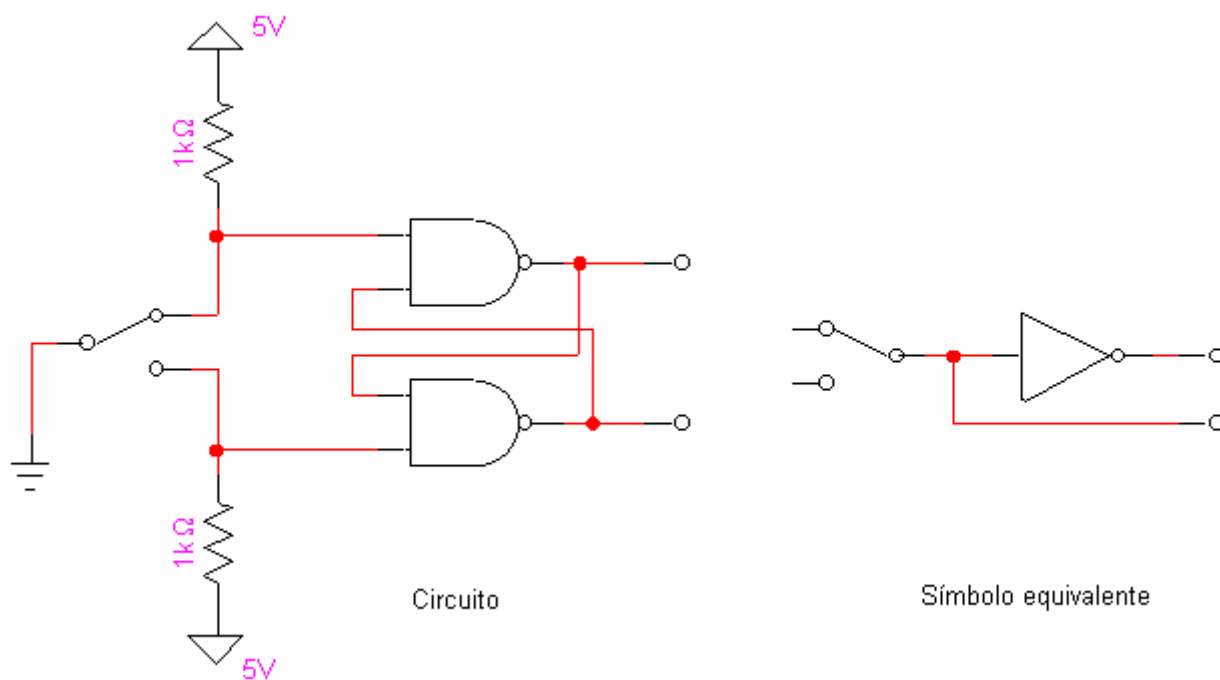


Figura 6 – Chave de 1 posição com 2 saídas e um circuito equivalente

As portas NAND e resistores adicionais servem para eliminar o **ruído de comutação** (*bouncing*). Quando a chave mecânica fecha, o contato não se estabiliza imediatamente. Pode haver comutações microscópicas durante um intervalo de 10 a 50 milissegundos produzindo vários pulsos em forma de ruído. Se a chave fosse ligada diretamente a um circuito digital, estes pulsos poderiam causar funcionamento errôneo. O circuito biestável formado pelas 2 portas NAND elimina esse problema.

1.4.2. Relógio

O módulo de relógio destina-se, por exemplo, à sincronização de operações sequenciais. Tem duas saídas e dois modos de operação. No modo lento, as saídas fornecem ondas quadradas com patamares em 0 e 5 volts, com frequências de 1 Hz e 100 Hz, aproximadamente. No modo rápido, as frequências são de 1 kHz e 100 kHz. Os potenciômetros miniatura permitem variação fina de frequência em torno desses valores. A **Figura 7** mostra o símbolo usado para representar o relógio.



Figura 7 – Símbolo CLOCK

1.4.3. Diodo foto-emissor

Os módulos de diodo foto-emissor servem para indicar o nível lógico de um dado ponto num circuito. O esquema montado é indicado na **Figura 8**.

Se a entrada estiver no nível lógico 1, o diodo acenderá. O transistor é usado para reduzir a corrente que o módulo exige do circuito ao qual está ligado.

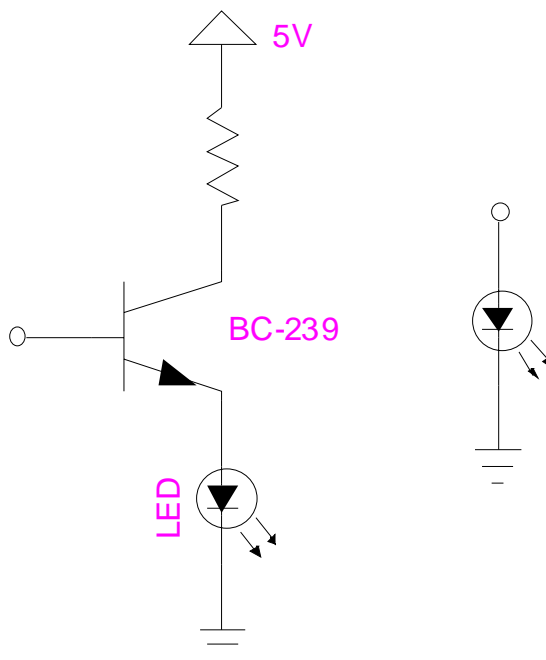


Figura 8 – Diodo emissor de luz e circuito equivalente

1.4.4. Ponta lógica

A ponta lógica é um módulo de teste e depuração de circuitos montados. É semelhante ao módulo de diodo emissor de luz, porém, com um refinamento. Além de indicar níveis estacionários, a ponta detecta e alarga pulsos mais estreitos que 50 ms. Dessa forma, um pulso muito estreito faz o diodo indicador da ponta piscar durante um tempo (50 ms) suficiente para percepção pelo olho humano.

1.4.5. Mostradores numéricos

Reproduzem algarismos decimais segundo um certo código binário. Serão vistos com detalhes posteriores.

1.4.6. Protoboard

A *protoboard* é uma das peças mais importantes e a usaremos em todos os experimentos. Ela é encaixada na parte central do painel de forma a podermos usar todos módulos disponibilizados pelo painel, bastando os conectar a *protoboard* com fios. A **Figura 9** mostra o esquema de uma *protoboard*.

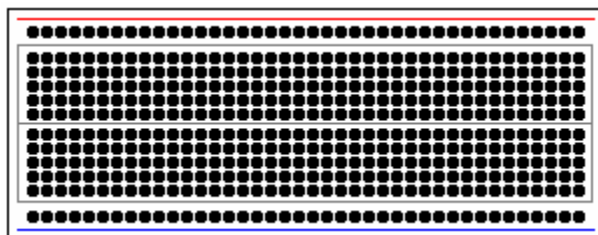


Figura 9 – Esquema de uma *protoboard*

Os pontos negros representam furos que existem na *protoboard* para se conectar fios e encaixar CI's. Estes furos são, internamente, conectados entre si como mostra a **Figura 10**.

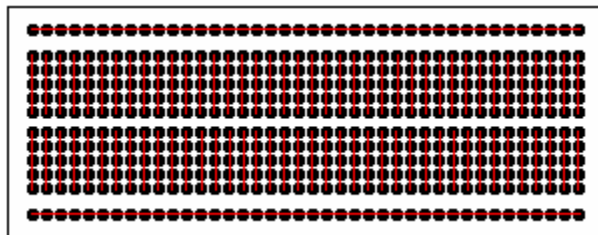
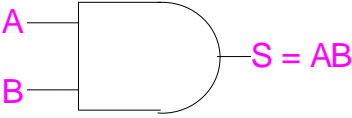


Figura 10 – Ligações da *protoboard*

2. PARTE EXPERIMENTAL

- 2.1. Ligue um diodo emissor de luz à saída de uma chave de uma posição. Verifique se ele acende e apaga corretamente quando a chave é pressionada e solta.
- 2.2. Ligue um diodo emissor de luz à saída de uma porta AND de duas entradas e use as chaves de uma posição para controlar as entradas da porta, conectando-as aos pinos correspondentes. Preencha a tabela da verdade dessa porta aplicando nas entradas as quatro combinações lógicas possíveis.

A	B	S



- 2.3. Implemente uma porta OR usando apenas portas AND e NOT (circuito da **Figura 5**). Preencha a tabela da verdade como no item anterior.

A	B	S

- 2.4. Projete e implemente uma porta AND usando apenas portas OR e NOT. Desenhe o esquema e preencha a tabela da verdade.

A	B	S	Esquema

Obs.: A operação AND pode ser escrita como $A \cdot B = \overline{\overline{A} + \overline{B}}$

- 2.5. A finalidade deste item é investigar a existência dos atrasos de propagação em portas. Monte o circuito da **Figura 11**.

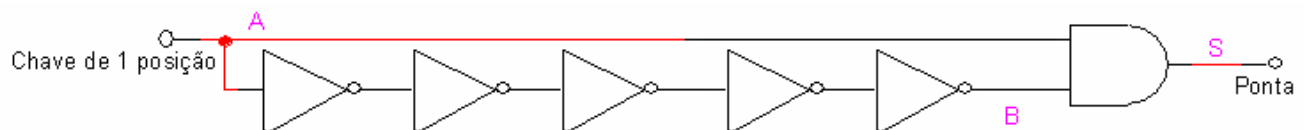


Figura 11 – Verificação do atraso de propagação

Enquanto se tiver $A = 0$, tem-se $B = 1$ e $S = 0$. Suponha agora que a chave é pressionada, de forma a fazer $A = 1$. Devido ao atraso nas 5 portas NOT's, a entrada B ainda permanecerá no

nível lógico 1 durante cerca de 50 ns. Após decorrerem os primeiros 10 ns, tempo necessário para a porta AND responder, a saída S irá, também, para o nível 1 e permanecerá aí até 10 ns após a entrada B finalmente passar para 0. Conseqüentemente, um pulso com largura de aproximadamente 50 ns terá aparecido na saída.

Com a ponta lógica ligada, este pulso será detectado e o diodo piscará de forma perceptível. É fácil ver que no retorno de A para o nível 0 **não** será produzido nenhum pulso em S.

Verifique o funcionamento deste circuito. Se a ponta lógica não for capaz de detectar pulsos de 50 ns, aumente-os para 70 ns, usando 7 portas NOT's no lugar de 5.

Será que algum pulso seria produzido na saída se fosse usado um número par de NOT's?

3. SUMÁRIO

São apresentados os circuitos digitais da família TTL. As portas AND, OR e NOT são usadas em montagens elementares, com o objetivo de verificar o funcionamento lógico e os atrasos de propagação. A familiarização com o painel é feita mediante a apresentação dos módulos periféricos para entrada ou saída de sinais de controle e depuração, e sua aplicação em situações diversas.

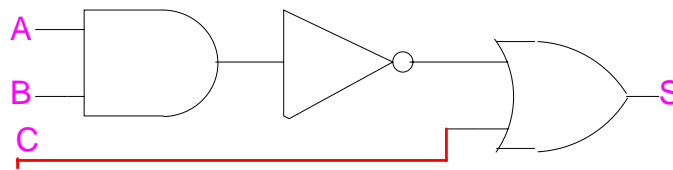
4. EQUIPAMENTOS E MATERIAIS

- Painel digital;
- *Protoboard*;
- Ponta lógica;
- Fios conectores;
- Portas AND, OR e NOT.

5. TESTE DE AUTO-AVALIAÇÃO

1. Com relação aos níveis lógicos TTL de entrada e saída, assinale a alternativa correta:
 - a) ENTRADA: 0 a 0,4 V e 2,0 a 5,5 V
SAÍDA: 0 a 0,8 V e 2,4 a 5,0 V
 - b) ENTRADA: 0 a 0,8 V e 2,0 a 5,5 V
SAÍDA: 0 a 0,4 V e 2,4 a 5,0 V
 - c) ENTRADA: 0 a 0,4 V e 2,4 a 5,0 V
SAÍDA: 0 a 0,8 V e 2,0 a 5,5 V
 - d) ENTRADA: 0 a 0,8 V e 2,4 a 5,0 V
SAÍDA: 0 a 0,4 V e 2,0 a 5,5 V
2. Assinale os conjuntos universais dentre os conjuntos abaixo:
 - a) AND, OR e NOT
 - b) AND e OR
 - c) AND e NOT
 - d) OR e NOT
 - e) NOT

3. Preencha a tabela da verdade do circuito abaixo:



A	B	C	S
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

4. Com relação a pulsos em S de correntes de atrasos de propagação, estabeleça uma associação um a um entre as colunas da esquerda e da direita:

- () Produz um pulso \sqcap quando A passa de 0 para 1.
- () Não produz pulso em nenhuma transição.
- () Produz um pulso \sqcup quando A passa de 1 para 0.
- () Produz um pulso \sqcap quando A passa de 0 para 1.

