Teoria de Controle e suas aplicações em Sistemas Digitais

Anderson Nakashima (6797459)
Guilherme Eiji Tsuyama (6797417)
Renato Yoshinari (6797362)
William Miyasaka (5749939)

30 de novembro de 2011

Resumo

Sistemas digitais são de vital importância nos dias atuais. Eles estão presentes em praticamente todos os ramos da tecnologia, devido principalmente ao seu baixo custo e alta performace. Dentre algumas de suas aplicações estão o controle, processamento, filtragem e medição de sistemas e sinais analógicos, técnicas muito utilizadas em Ciências da Terra, por exemplo. Sendo assim, este trabalho visa desenvolver a teoria necessária para o estudo e o entendimento desses modelos especiais de sistemas dinâmicos, chamados Sistemas Digitais. Para tal, alguns tópicos adicionais serão necessários, tais como: Álgebra Booleana, Teoria dos Autômatos, "Theory of Switching Networks"e "Logic Design".

Sumário

1 Objetivo **4**

2 Introdução **4**
2.1 Sistemas Digitais 4

3 Álgebra Booleana e Circuitos Lógicos **5**
3.1 Operações básicas da álgebra booleana 5
3.1.1 Operação OU (Adição Lógica) 5
3.1.2 Operação E (Multiplicação Lógica) 5
3.1.3 Complementação (Negação) 5
3.2 Avaliação de Expressões Booleananas 6
3.3 Leis Fundamentais e Propriedades da Álgebra Booleana . . . 6
3.4 Teoremas de De Morgan 7
3.5 Portas Lógicas 7

4 Circuitos Combinacionais **9**
4.1 Circuitos Combinacionais de Interconexão 9
4.1.1 Decodificadores 9

5 Circuitos Sequenciais **14**
5.1 Latches 15
5.1.1 Latch RS 15
5.1.2 Latch RS controlado 19
5.1.3 Latch D 20
5.1.4 Latches com porta de ativação complementar 21
5.2 Flip-flops 22
5.2.1 Flip-flop D mestre-escravo 22
5.2.2 Flip-flop D disparado pela borda ascendente 23

1 Objetivo

Neste presente trabalho o nosso objetivo principal é o de desenvolver a teoria sobre Sistemas de Controle e sua aplicabilidade a Sistemas Digitais.

2 Introdução

Nesta seção, daremos uma breve introdução sobre sistemas digitais. Este artigo será baseado no livro de referência[1] , que trata basicamente da teoria de sistemas em dimensão finita.

2.1 Sistemas Digitais

Do latim *systēma*, sistema é definido como sendo um conjunto de elementos, concretos ou abstratos, organizados de tal forma a se alcançar um determinado objetivo. Sendo assim, um Sistema Digital pode ser definido, fisicamente, como um conjunto de componentes interconectados que processam informações de maneira digital.

A principal diferença entre dispositivos analógicos e digitais está na forma de como os dados, informações ou quantidades físicas, são lidos e processados internamente pelo sistema físico. Essencialmente, no primeiro caso o tratamento das quantidades manipuladas é feito de forma proporcional, podendo variar num determinado intervalo contínuo de valores. No caso digital, as quantidades manipuladas são representadas por símbolos chamados dígitos, que assumem apenas valores discretos.

Muitos dos sistemas digitais são binários, ou seja, sua entrada, saída, e variáveis de estado assumem apenas dois diferentes valores. Matematicamente, eles podem ser representados pelo Sistema Binário $\mathbb{Z}_2$.

Basicamente, há duas classes de sistemas digitais: os circuitos combinacionais e os circuitos sequenciais.

[1] Hinrichsen, Diederich e Pritchard, J. Anthony. Mathematical Systems Theory I, Springer.

3 Álgebra Booleana e Circuitos Lógicos

Utilizaremos aqui uma álgebra booleana de dois valores, ou seja, cada variável pode assumir um dentre dois valores. Neste trabalho utilizaremos o conjunto $B = \{0, 1\}$ como o conjunto de valores para a variável booleana.

3.1 Operações básicas da álgebra booleana

3.1.1 Operação OU (Adição Lógica)

Definição 1. *A operação OU resulta 1 se pelo menos uma das variáveis de entrada vale 1.*

Notação 2. *Definiremos a operação OU por meio do símbolo + .*

A operação OU goza das propriedades associativa e comutativa, como pode ser visto na tabela abaixo:

A	B	C	$A+B+C$	$A+B$	$(A+B)+C$	$B+C$	$(B+C)+A$
0	0	0	0	0	0	0	0
0	0	1	1	0	1	1	1
0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	1	1
0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	1

3.1.2 Operação E (Multiplicação Lógica)

Definição 3. *A operação E resulta 0 se pelo menos uma das variáveis de entrada vale 0.*

Notação 4. *Definiremos a operação E por meio do símbolo . .*

Pela tabela abaixo, podemos notar que, assim como a operação OU, a operação E goza das propriedades associativa e comutativa.

3.1.3 Complementação (Negação)

A operação Negação resulta no complementar do valor de entrada. Usualmente a operação Negação ou Complementação sobre uma variável A é representada pelo símbolo: $\overline{A}$.

A	B	C	$A.B.C$	$A.B$	$(A.B).C$	$B.C$	$A.(B.C)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0
1	1	1	1	1	1	1	1

A	$\overline{A}$
0	1
1	0

3.2 Avaliação de Expressões Booleananas

A avaliação do comportamento de uma expressão booleana é descrito pela sua tabela verdade. Uma tabela verdade consiste em um conjunto de colunas, nas quais são listadas todos os possíveis valores de entrada (geralmente à esquerda da tabela) e os valores de saída da função (geralmente à direita), sendo que na maioria das vezes, principalmente em funções mais complexas, é conveniente criar colunas intermediárias, listando resultados de subexpressões. Além disso, na análise de expressões booleanas é imprescindível seguir a ordem de precedência entre as operações OU e E (E tem precedência em relação à OU). Além disso, as expressões entre parênteses tem precedência sobre os operadores E e OU que estejam no mesmo nível.

3.3 Leis Fundamentais e Propriedades da Álgebra Booleana

Seja A, B, C variáveis booleanas. Temos então as seguintes propriedades:
Da adição lógica:

1. $A + 0 = A$
2. $A + 1 = 1$
3. $A + A = A$
4. $A + \overline{A} = 1$

Da multiplicação lógica:

5. $A.0 = 0$
6. $A.1 = A$
7. $A.A = A$
8. $A.\overline{A} = 0$

   Da complementação:

9. $\overline{\overline{A}} = A$

   Comutatividade:

10. $A + B = B + A$
11. $A.B = B.A$

    Associatividade:

12. $A + (B + C) = (A + B) + C = (A + C) + B$
13. $A.(B.C) = (A.B).C = (A.C).B$

    Distributiva:

14. $A.(B + C) = A.B + A.C$

3.4 Teoremas de De Morgan

Theorem 5. $\overline{A.B.C.\ldots} = \overline{A} + \overline{B} + \overline{C} + \ldots$

Theorem 6. $\overline{A + B + C + \ldots} = \overline{A}.\overline{B}.\overline{C}.\ldots$

3.5 Portas Lógicas

Portas lógicas são símbolos que representam cada uma das operações lógicas. Definiremos circuito lógico como um conjunto de portas lógicas com suas respectivas conexões.

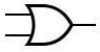

Figura 1: Porta lógica OR associada à operação lógica OU

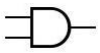

Figura 2: Porta lógica AND associada à operação lógica E

Figura 3: Porta lógica NOT (complementação)

Figura 4: Porta lógica NAND associada à operação lógica: $\overline{A.B}$

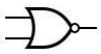

Figura 5: Porta lógica NOR associada à operação lógica: $\overline{A+B}$

Figura 6: Porta lógica XOR associada à operação lógica: $A.\overline{B}+\overline{A}.B$ (A ou B exclusivo.Ou seja: A ou B, mas não ambos simultaneamente)

Figura 7: Porta lógica XNOR associada à operação lógica: $\overline{A.\overline{B}+\overline{A}.B}$ (negação de XOR)

4 Circuitos Combinacionais

Circuitos combinacionais são aqueles constituídos por um conjunto de portas lógicas as quais determinam os valores de saída diretamente a partir dos valores atuais da entrada.

Em termos matemáticos, seja $\sum$ um sistema dinâmico com m entradas e p saídas, $u : \mathbb{Z} \to \mathbb{Z}_2^m$ representando a entrada, e $y : \mathbb{Z}_2^m \to \mathbb{Z}_2^p$ a saída, então existe um operador F que descreve completamente o comportamento do sistema, onde $F : \mathbb{Z}_2^m \to \mathbb{Z}_2^p$, tal que $y(t) = F(u(t))$. Este operador realiza o processamento das informações por meio da Álgebra Booleana.

Sendo assim, cada combinação de valores da entrada $u(t)$ pode ser vista como uma informação diferente, e cada conjunto de valores de saída $y(t)$ representa o resultado da operação booleana. Este é um típico sistema de controle de malha aberta, onde seu controle depende apenas da entrada e do modelo do sistema.

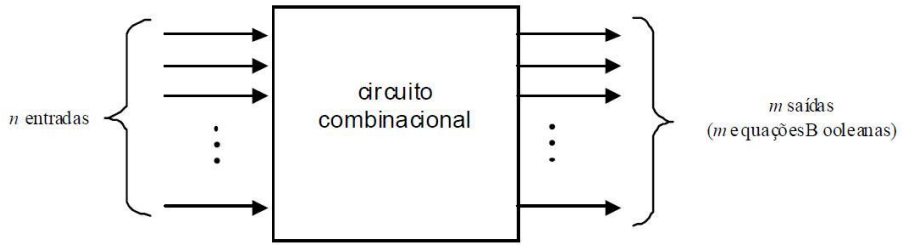

Figura 8: Diagrama genérico de um circuito combinacional

4.1 Circuitos Combinacionais de Interconexão

Os circuitos combinacionais são responsáveis também por funções de conexão entre os diversos operadores. Dentre essas funções está a decodificação. Os elementos que realizam essa operação são denominados decodificadores.

4.1.1 Decodificadores

Um decodificador é um circuito combinacional usado para ativar um dentre m componentes. Um decodificador n : m (lê-se n por m) possui n entradas e m saídas, com m $\leq 2^n$. Cada combinação das variáveis de entrada seleciona um e somente uma dentre as m saídas, de modo que cada

saída somente será selecionada por uma das m combinações. Desta forma, é natural que se associe a cada saída um índice decimal que represente a combinação de entradas responsável pela sua ativação. Assumindo-se ativação em lógica direta, isto é, que uma saída está ativada se ela vale 1, então a tabela verdade e o diagrama para um decodificador 3:8 são os seguintes:

endereço	Entradas (sinais de controle)			saídas							
	A_2	A_1	A_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

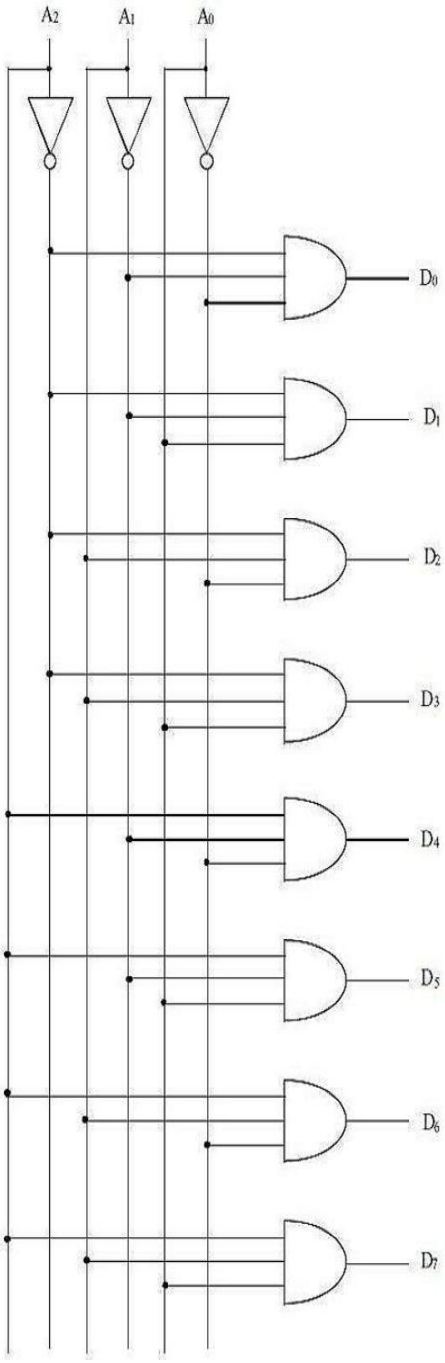

Figura 9: Diagrama de um decodificador 3:8

Um decodificador pode possuir uma entrada de habilitação. Esta entrada tem a função de habilitar ou desabilitar seu funcionamento. Assim, se esta entrada valer 0, nenhuma saída estará ativada, independente dos valores das demais entradas. Por outro lado, se a entrada de habilitação valer 1, o decodificador estará ativando uma das saídas. A tabela verdade e o diagrama de um decodificador 2:4 com ativação e habilitação em lógica direta são os seguinte:

endereço	entradas (sinais de controle)			saídas			
	E	A_1	A_0	D_0	D_1	D_2	D_3
-	0	0	0	0	0	0	0
-	0	0	1	0	0	0	0
-	0	1	0	0	0	0	0
-	0	1	1	0	0	0	0
0	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
2	1	1	0	0	0	1	0
3	1	1	1	0	0	0	1

saídas desabilitadas

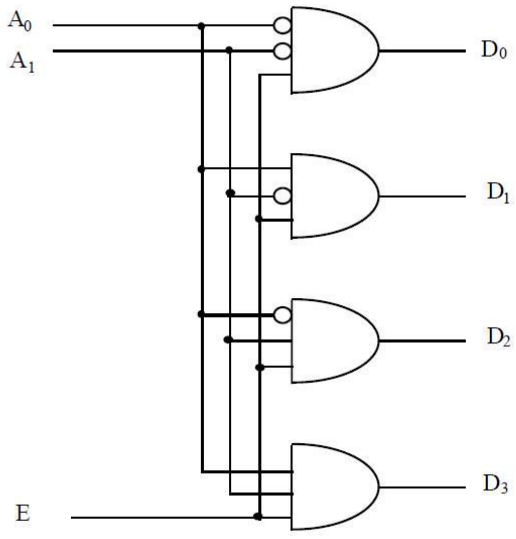

Figura 10: Diagrama de um decodificador 2:4 com entrada de habilitação

Como pode-se verificar, nas primeiras 4 linhas o sinal de habilitação (E) vale zero, o que desativa as saídas, independentemente dos valores das demais entradas (A1 e A0). Desta forma, podemos re-escrever esta tabela de maneira mais compacta:

endereço	entradas (sinais de controle)			saídas			
	E	A_1	A_0	D_0	D_1	D_2	D_3
saídas desabilitadas → -	0	X	X	0	0	0	0
0	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
2	1	1	0	0	0	1	0
3	1	1	1	0	0	0	1

5 Circuitos Sequenciais

Circuitos sequenciais, também conhecidos como autômatos finitos, são aqueles cujas saídas dependem não apenas dos valores atuais de entrada mas também da sequência das entradas anteriores. Em outras palavras, podemos considerar um circuito sequencial como sendo um circuito combinacional que contém um outro componente, chamado de elementos de memória.

Basicamente, a estrutura de um circuito sequencial é representado da seguinte forma: as entradas e saídas do circuito sequencial são ligadas apenas ao circuito combinacional. Parte das saídas do circuito combinacional são ligadas aos elementos de memória, e as saídas dos elementos de memória formam parte da entrada do circuito combinacional. Este é um típico sistema de controle de malha fechada, onde seu controle depende da entrada e da realimentação do sistema. A informação armazenada nos elementos de memória define o estado do circuito sequencial. O processamento da informação da entrada do circuito sequencial juntamente com o estado atual determina os valores de saída e o próximo estado.

Os circuitos sequenciais podem ser divididos em dois grupos, de acordo com seu comportamento temporal: síncronos e assíncronos. Os circuitos sequenciais assíncronos são circuitos nos quais o tempo de propagação de uma mudança nas entradas até a saída pode ser encarado como o tempo real (tempo que cada componente demora para processar a informação). Já os circuitos sequenciais síncronos são circuitos nos quais se utiliza um sinal chamado de clock com o objetivo de cadenciar as trocas de estado, uniformizando o tempo entre duas bordas (descida ou subida). Os elementos de memória utilizados nos circuitos sequenciais são denominados flip-flops.

Um flip-flop é um circuito digital que possui duas entradas e duas saídas sendo também capaz de armazenar um bit de informação. O objetivo é que o estado atual fique armazenado no conjunto de flip-flops até que uma nova borda do relógio chegue, quando o próximo estado passa a ser o estado atual e o próximo estado seja gerado pelo circuito combinacional. Os tipos mais básicos de flip-flop são os latches. Eles operam por níveis dos sinais de entradas e são base para a elaboração de flip-flops mais sofisticados.

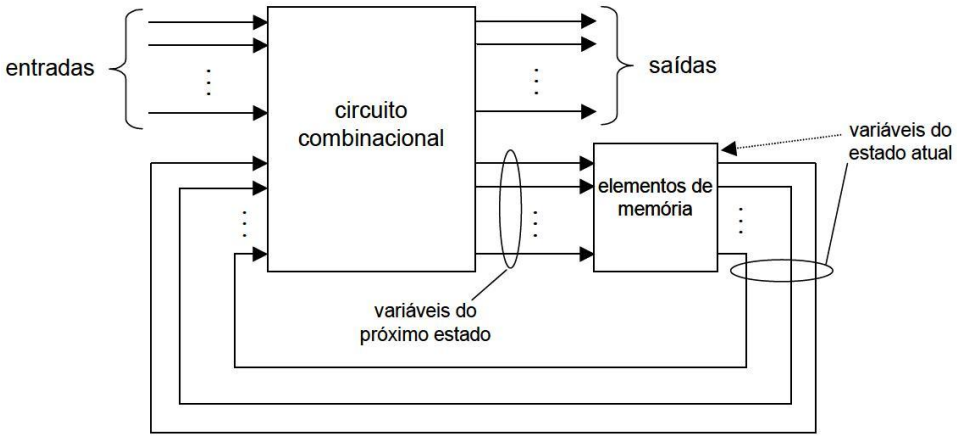

Figura 11: Diagrama de blocos de um circuito sequencial.

5.1 Latches

Os tipos mais simples de flip-flops são denominados latches, que são componentes que operam por níveis de sinais (diz-se que eles são sensíveis a nível) e servem como base para a elaboração de flip-flops mais sofisticados. Estudaremos agora alguns exemplos de latches.

5.1.1 Latch RS

O latch RS é constituído por duas portas nor (n1 e n2) com duas entradas cada conforme pode-se observar na figura.

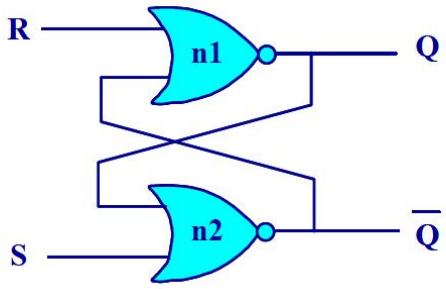

Figura 12: Latch RS com portas nor.

Nesse tipo de circuito temos duas entradas: R e S, e duas saídas: Q e $\overline{Q}$ sendo que há uma conexão entre a saída Q com uma das entradas de n2 e

uma outra conexão entre a saída $\overline{Q}$ e uma das entradas de n1. Essas conexões entre entradas e saída de circuitos digitais são denominadas realimentações e são responsáveis pela propriedade de armazenamento apresentada no circuito. Observamos então que se trata de um circuito sequencial.

A fim de compreender o comportamento de um latch elabora-se uma tabela denominada tabela de transição de estados (ou tabela de transição) que descreve o comportamento do circuito. Cada latch (e cada flip-flop) tem uma tabela de transição que lhe é própria. Em uma tabela de transição de estados listam-se todas as combinações dos possíveis valores de entrada associando os seus respectivos valores de saída. O latch pode assumir um entre dois estados (saídas) possíveis: 0 e 1. O estado 0 é denominado reset e o estado 1 é denominado set.

Uma outra forma de compreender o comportamento do circuito é analisar a sua forma de onda. Uma forma de onda pode ser imaginada sobre o plano cartesiano, onde o eixo x representa o tempo e o eixo y representa o valor lógico da variável.

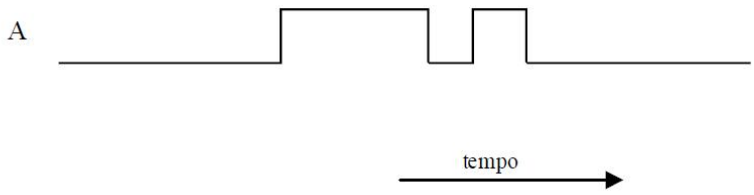

Figura 13: Exemplo de forma de onda.

Infelizmente as portas lógicas não conseguem responder de maneira instantânea às variações em suas entradas. Ao tempo decorrido entra a mudança dos valores de entrada e a mudança dos valores de saída dá-se o nome de atraso da porta lógica. Vamos agora estudar o comportamento do latch RS. Seja t_d(n1) e t_d(n2) os atrasos referentes à portas lógicas nor n1 e n2, respectivamente.

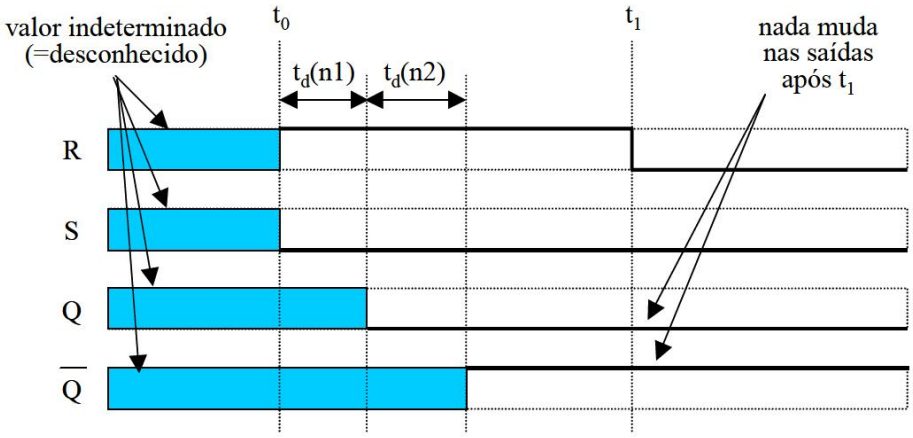

Figura 14: Formas de onda para a aplicação do vetor de entrada (R=1;S=0) seguido do vetor (R=0;S=0) no latch RS.

Suponha que, no instante t_0, seja aplicado simultaneamente os valores de entrada R = 1 e S = 0. Note que R = 1 implica em Q = 0, então temos que a saída Q se estabilizará com valor 0 em $t_0 + t_d(\text{n1})$ e, como a saída de n1 está ligada à entrada de n2, temos que a partir do tempo $t_0 + t_d(\text{n1}) + t_d(\text{n2})$, a saída $\overline{Q}$ estará estabilizada com valor 1.

Aplicando agora que no instante $t_1 > t_0 + t_d(\text{n1}) + t_d(\text{n2})$ foram aplicados os valores R = 0 e S = 0. É facil ver que, mesmo após $t_1 + t_d(\text{n1}) + t_d(\text{n2})$ os valores de Q e $\overline{Q}$ permanecem 0 e 1, respectivamente. Analisando agora se em t_0 fossem aplicados os valores de entrada: R = 0 e S = 1. Veja que, neste caso, $\overline{Q}$ se estabilizará com valor lógico 0 após $t_0 + t_d(\text{n2})$ e, consequentemente Q se estabilizará com valor lógico 1 após $t_0 + t_d(\text{n2}) + t_d(\text{n1})$. Observemos que, também nesse caso, ao aplicarmos os valores de entrada R = 0 e S = 0 em $t_1 > t_0 + t_d(\text{n2}) + t_d(\text{n1})$ não há mudança dos valores lógicos de Q e $\overline{Q}$.

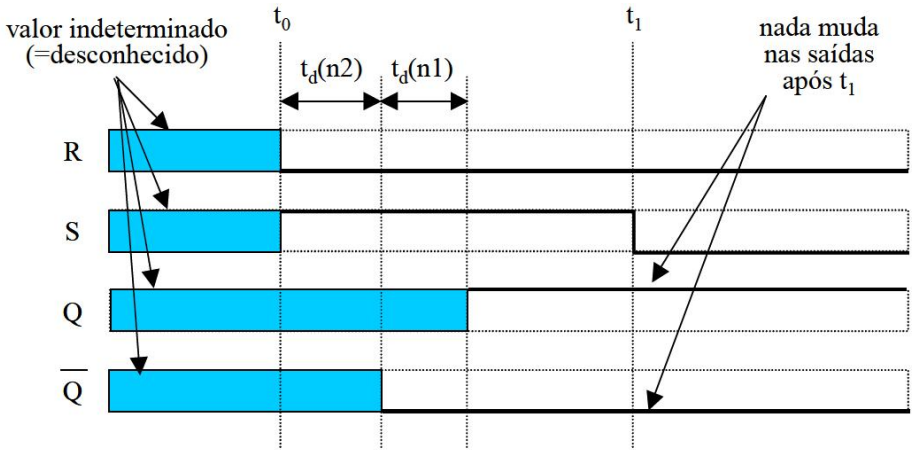

Figura 15: Formas de onda para a aplicação do vetor de entrada (R=0;S=1) seguido do vetor (R=0;S=0) no latch RS.

Notemos que, nos casos estudados até agora, temos que os valores de Q e $\overline{Q}$ são sempre complementares. Entretanto, observemos que se aplicarmos os valores de entrada R = 1 e S = 1 obteremos Q = 1 e $\overline{Q}$ = 1. Então foi convencionado que esse seria um estado proibido (ou indeterminado). O funcionamento sequencial do latch RS pode ser resumido pela tabela:

R	S	Q	$\overline{Q}$	ação
1	0	0	1	vai para **estado reset**
0	0	0	1	mantém **estado reset** (= estado anterior)
0	1	1	0	vai para **estado set**
0	0	1	0	mantém **estado set** (= estado anterior)
1	1	0	0	**estado proibido**

A tabela anterior pode ser escrita de maneira mais compacta com uma tabela de transição de estados para o latch RS, de modo a incorporar a informação da dependência temporal.

R	S	Q_{t+1}	comentário
0	0	Q_t	mantém estado anterior
0	1	1	estado set
1	0	0	estado reset
1	1	-	proibido

Para evitar que se desenhar o circuito completo toda vez que houver um

ocorrência do latch RS, costuma-se adotar o símbolo abaixo:

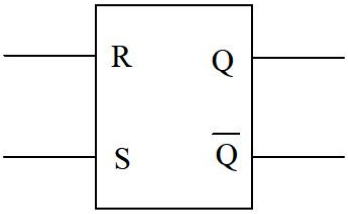

Figura 16: Símbolo do latch RS

5.1.2 Latch RS controlado

O latch RS controlado é um aprimoramento do latch RS. Ele é constituído por um par de portas E e pelo latch RS. O latch RS controlado possui três entradas: além das duas entradas R e S do latch RS ele apresenta uma entrada C.

A entrada C temo objetivo de habilitar ou desabilitar o latch RS, ou seja: se C = 0, o latch mantém seu estado; caso C = 1, o latch funciona normalmente.

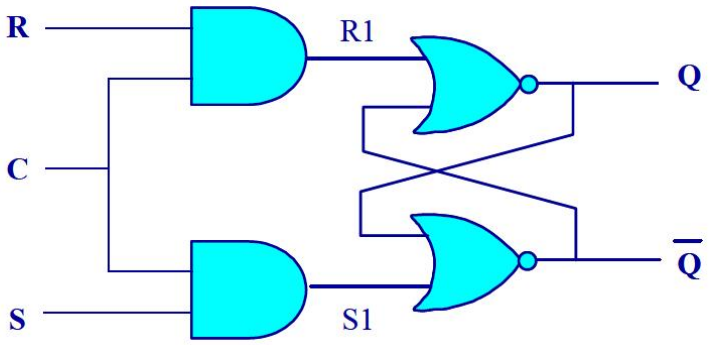

Figura 17: Latch RS controlado.

A tabela de transição de estados e o símbolo para o latch RS controlado são dados abaixo:

C	R	S	Q_{t+1}	comentário
0	X	X	Q_t	mantém estado anterior
1	0	0	Q_t	mantém estado anterior
1	0	1	1	estado set
1	1	0	0	estado reset
1	1	1	-	proibido

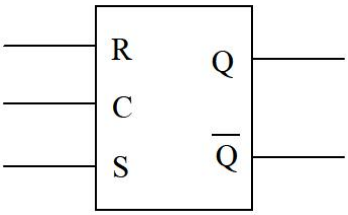

Figura 18: Símbolo do latch RS controlado

5.1.3 Latch D

Nos latches RS temos a necessidade de evitar que o estado proibido ocorra. Para evitar tal problema foi elaborado o latch D. O latch D é constituído de um latch RS e uma porta do tipo inversora. A porta inversora é colocada entre as entradas S e R de forma a assegurar que nunca ocorrerá a situação R = 1 e S = 1 simultaneamente.

Note que, pela disposição das entradas o estado R = 0 e S = 0 (responsável pela manutenção do estado) também desaparecerá.

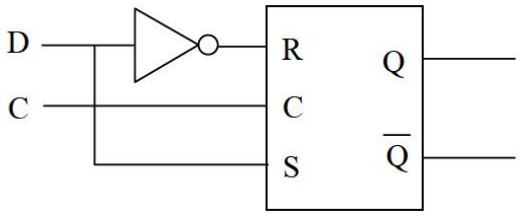

Figura 19: Latch D.

A tabela de transição de estados e o símbolo para o latch D são dados abaixo:

C	D	Q_{t+1}	comentário
0	X	Q_t	mantém estado anterior
1	0	0	estado reset
1	1	1	estado set

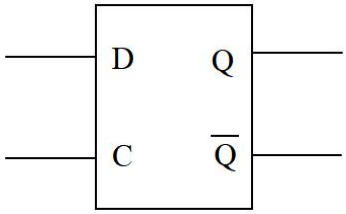

Figura 20: Símbolo do latch D.

5.1.4 Latches com porta de ativação complementar

Até aqui os latches são ativados quando o controle C = 1 e desativados quando C = 0. Estes são denominados latches com lógica de ativação direta.

Apresentamos agora o latch com lógica de ativação complementar. Esse tipo de latch é ativado quando o controle C = 0 e desativado quando C = 1 e é construído através da simples inserção de uma porta inversora antes da entrada de controle.

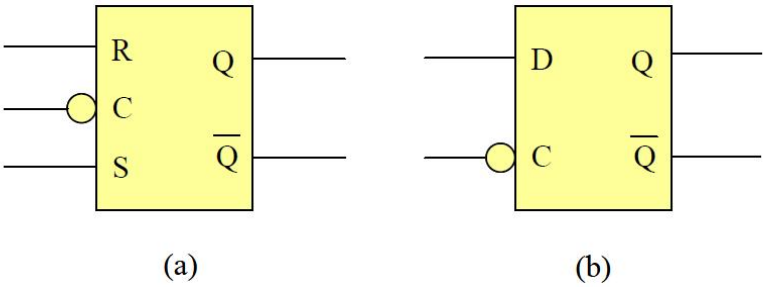

Figura 21: Símbolo do latch RS controlado (a) e do latch D (b), ambos com lógica de ativação complementar.

As tabelas abaixo mostram o funcionamento destes latches com lógica de ativação negada. Comparando-se com as tabelas de transição dos latches correspondentes com lógica de ativação direta, nota-se que as ações são as mesmas; apenas o que muda é o nível do sinal de controle necessário para

ativá-los.

C	R	S	Q_{t+1}	comentário
1	X	X	Q_t	mantém estado anterior
0	0	0	Q_t	mantém estado anterior
0	0	1	1	estado set
0	1	0	0	estado reset
0	1	1	-	proibido

C	D	Q_{t+1}	comentário
1	X	Q_t	mantém estado anterior
0	0	0	estado reset
0	1	1	estado set

5.2 Flip-flops

Flip-flops são circuitos derivados dos latches e são amplamente utilizados em circuitos sequenciasis síncronos. Como visto anteriormente, os latches são ativados pelo nível de sinal de controle, já os flip-flops são ativados pela transição do sinal de controle, o que faz com que um flip-flop permaneça ativado apenas durante um intervalo de tempo muito pequeno após a ocorrência da transição do sinal de controle.

Dependendo da sua construção, um flip-flop pode ser disparado pela transição de subida ou pela transição de descida do sinal de controle.

5.2.1 Flip-flop D mestre-escravo

O flip-flop D mestre-escravo e composto por dois latches D conectados conforme a figura.

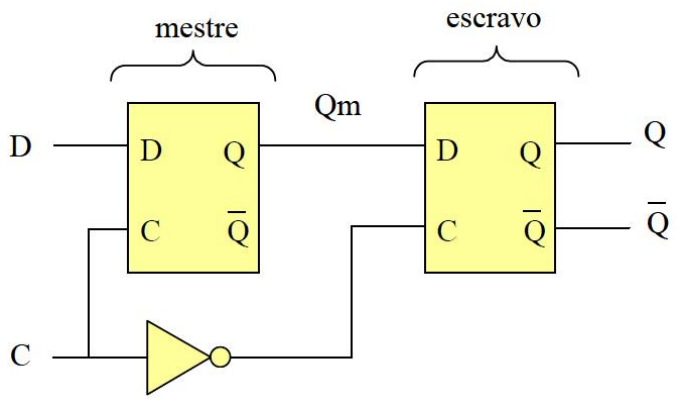

Figura 22: Flip-flop D mestre-escravo.

O primeiro latch é denominado mestre e o segundo é denominado escravo. Note que o sinal de controle está conectado à entrada C do latch

mestre e ao inversor cuja saída está contectada à entrada C do latch escravo por esse motivo dizemos que o mestre e o escravo funcionam de forma complementar ao sinal de controle. O diagrama de forma de onda exemplifica esse comportamento.

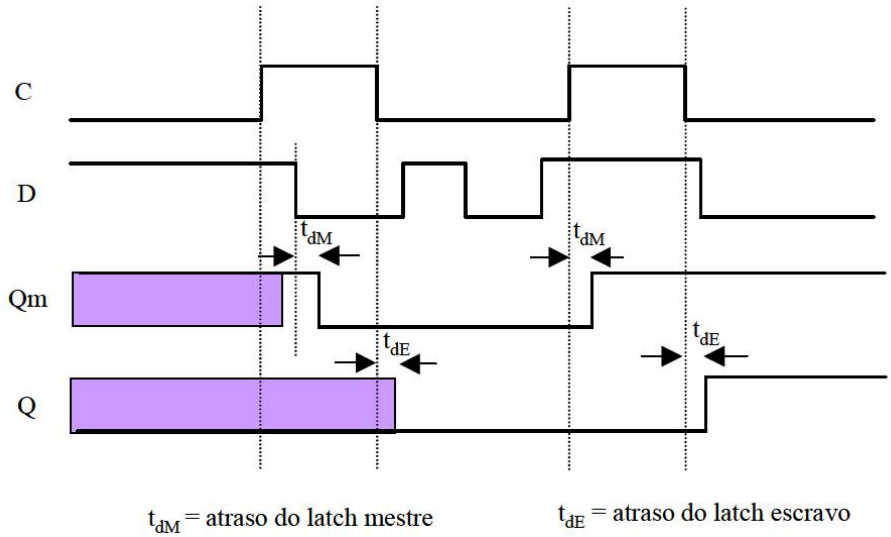

Figura 23: Exemplo do funcionamento do flip-flop D mestre-escravo.

O flip-flop D mestre-escravo funciona como se fosse disparado pela borda descendente do sinal de controle: o último valor de D amostrado pelo latch mestre antes da borda descendente fica armazenado, aparecendo na saída Q do latch escravo logo após a mesma borda descendente.

5.2.2 Flip-flop D disparado pela borda ascendente

Dependendo da maneira de como é contruído, o flip-flop pode ser disparado somente pela borda ascendente ou somente pela borda descendente do sinal de controle.

A figura mostra um flip-flop D disparado pela borda ascendente.

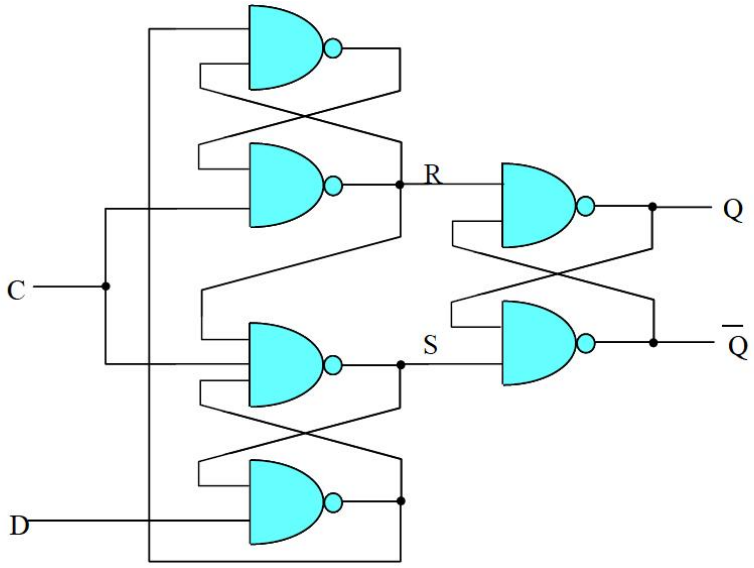

Figura 24: Flip-flop D disparado pela borda ascendente.

Veja na tabela de transição de estados, o comportamento de um flip-flop D disparado pela borda ascendente.

C	D	Q_{t+1}	comentário
$\neq\uparrow$	X	Q_t	mantém estado anterior
$\uparrow$	0	0	estado reset
$\uparrow$	1	1	estado set

O símbolo $\uparrow$ indica que a ativação do flip-flop é instantânea e ocorre somente durante as bordas ascendentes do sinal de controle.

Veja na figura o símbolo do flip-flop D disparado pela borda ascendente.

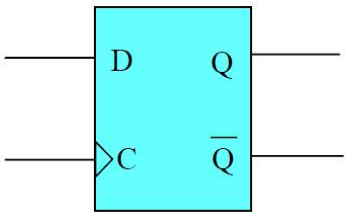

Figura 25: Símbolo do flip-flop D disparado pela borda ascendente.

O triângulo colocado na entrada C indica que a ativação ocorre na borda ascendente.

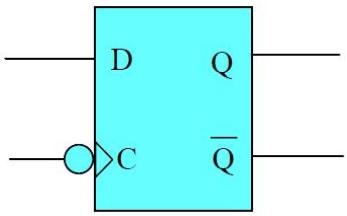

Figura 26: Símbolo do flip-flop D disparado pela borda descendente.

O círculo colocadoantes da entrada de controle indica que o flip-flop é disparado pela borda descendente.

Referências

[1] HINRICHSEN, Diederich, PRITCHARD, Anthony J. **Mathematical Systems Theory I:** Modelling, State Space Analysis, Stability and Robustness, Volume 1, Springer, 2010 (ISBN 3642039405, 9783642039409)

[2]http://minerva.ufpel.edu.br/ guntzel/isd/isd.html, acesso: 23/10/2011