Organização de Computadores Digitais - 5954008

2. Lógica Digital

Prof. Luiz Otavio Murta Jr

Local: Depto. de Computação e Matemática (FFCLRP/USP)

Principais Tópicos

2. Revisão de Lógica Digital

2.1. Álgebra Booleana

2.2. Portas Lógicas

2.3. Circuitos Combinatórios

2.4. Circuitos Sequenciais

2.3. Circuitos Combinatórios

- **Interconexão de portas lógicas na qual**
  - Sinal de saída é, em qualquer instante, função apenas dos sinais de entrada do circuito aplicados naquele instante
    - Considerando-se que a propagação do sinal é praticamente instantânea

- **Consiste de**
  - *n* entradas binárias
  - *m* saídas binárias

2.3. Circuitos Combinatórios

- **Pode ser definido de três formas**
  - Tabela Verdade
  - Símbolos Gráficos
  - Equações Booleanas

- **Obtenção a partir da tabela verdade**
  - Uso de equações com mintermos e simplificação lógica
    - Exemplos:
      - Cada mintermo é substituído por uma porta AND
      - Soma de mintermos substituída por portas OR

2.3.1. Multiplexadores (MUX)

- **Circuito que conecta várias entradas em uma única saída**
  - Em qualquer instante, uma única entrada é selecionada para ser passada para a saída
    - Pode ser projetado através da soma de mintermos

- **Utiliza *n* bits para selecionar uma das entradas**
  - Assim, no máximo, 2^n entradas podem ser selecionadas

2.3.1. Multiplexadores (MUX)

- No nível lógico, um **multiplexador** é um circuito com 2^n entradas de dados,
  - uma saída de dados e n entradas de controle que selecionam uma das entradas de dados.
- Essa entrada selecionada é dirigida (isto é, roteada) até a saída.
- A próxima figura é um diagrama esquemático de um multiplexador de oito entradas.

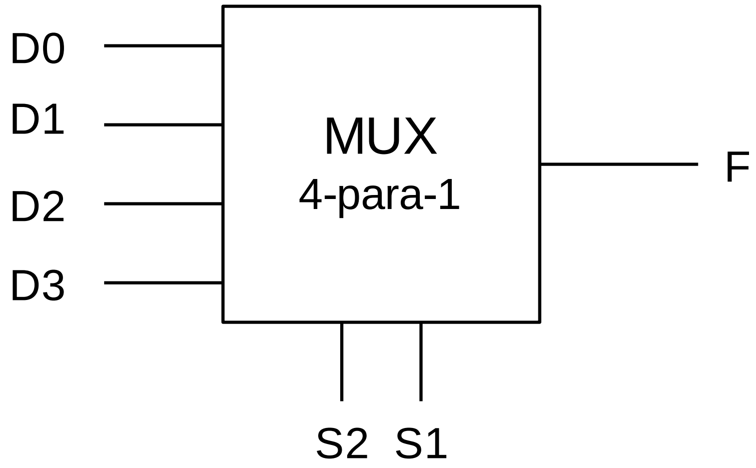
2.3.1. Multiplexadores (MUX)

- As três linhas de controle,
  - A, B e C, codificam um número de 3 bits que especifica qual das oito linhas de entrada é direcionada até a porta or e dali até a saída.
- Não importa qual valor esteja nas linhas de controle,
  - sete das portas and sempre produzirão saída 0;
  - a outra pode produzir ou um 0 ou um 1,
  - dependendo do valor da linha de entrada selecionada.
- Cada porta and é habilitada por uma combinação diferente das entradas de controle.

2.3.1. Multiplexadores

Exemplo 2.3.1. Multiplexador 4-para-1

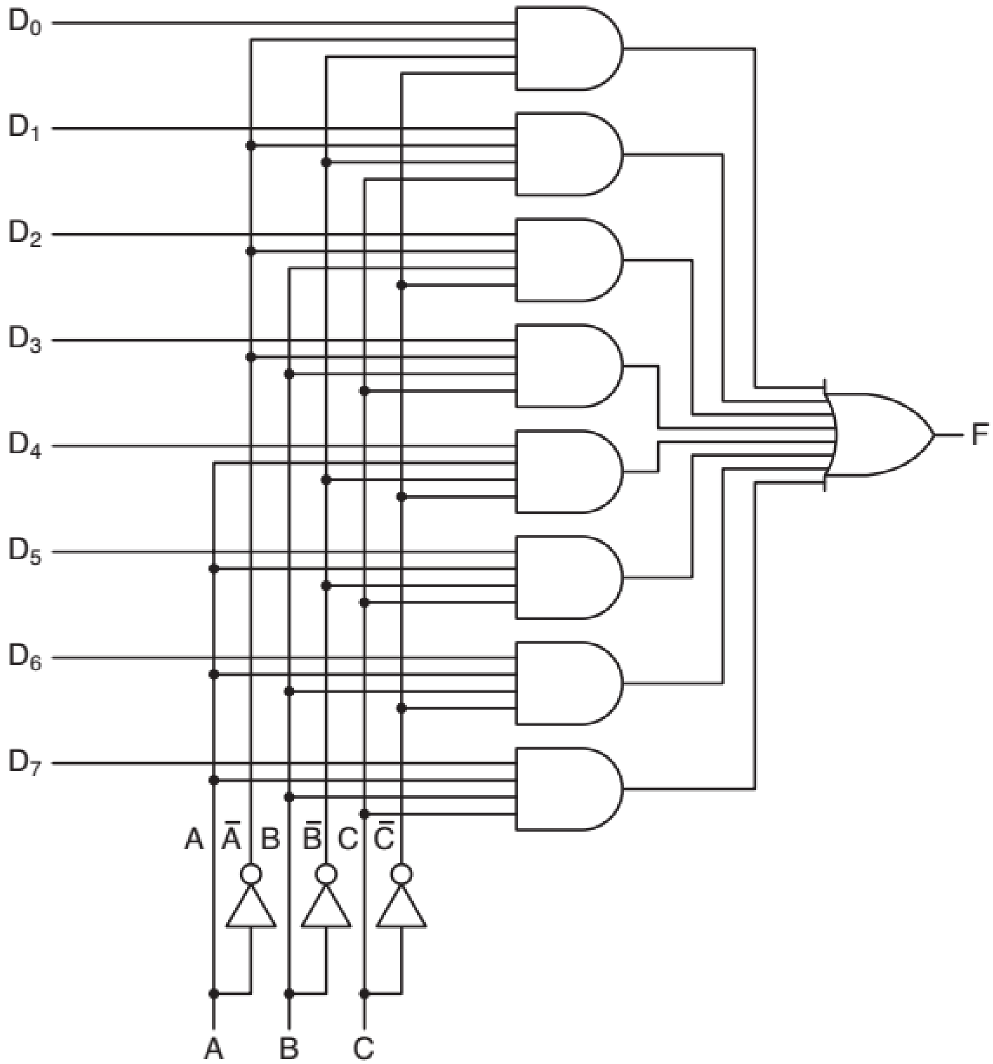
S2	S1	F
0	0	D0
0	1	D1
1	0	D2
1	1	D3



Qual é a equação e o circuito que descreve o multiplexador 4-para-1?

2.3.1. Multiplexadores (MUX)

- Circuito multiplexador de oito entradas.



2.3.1. Multiplexadores

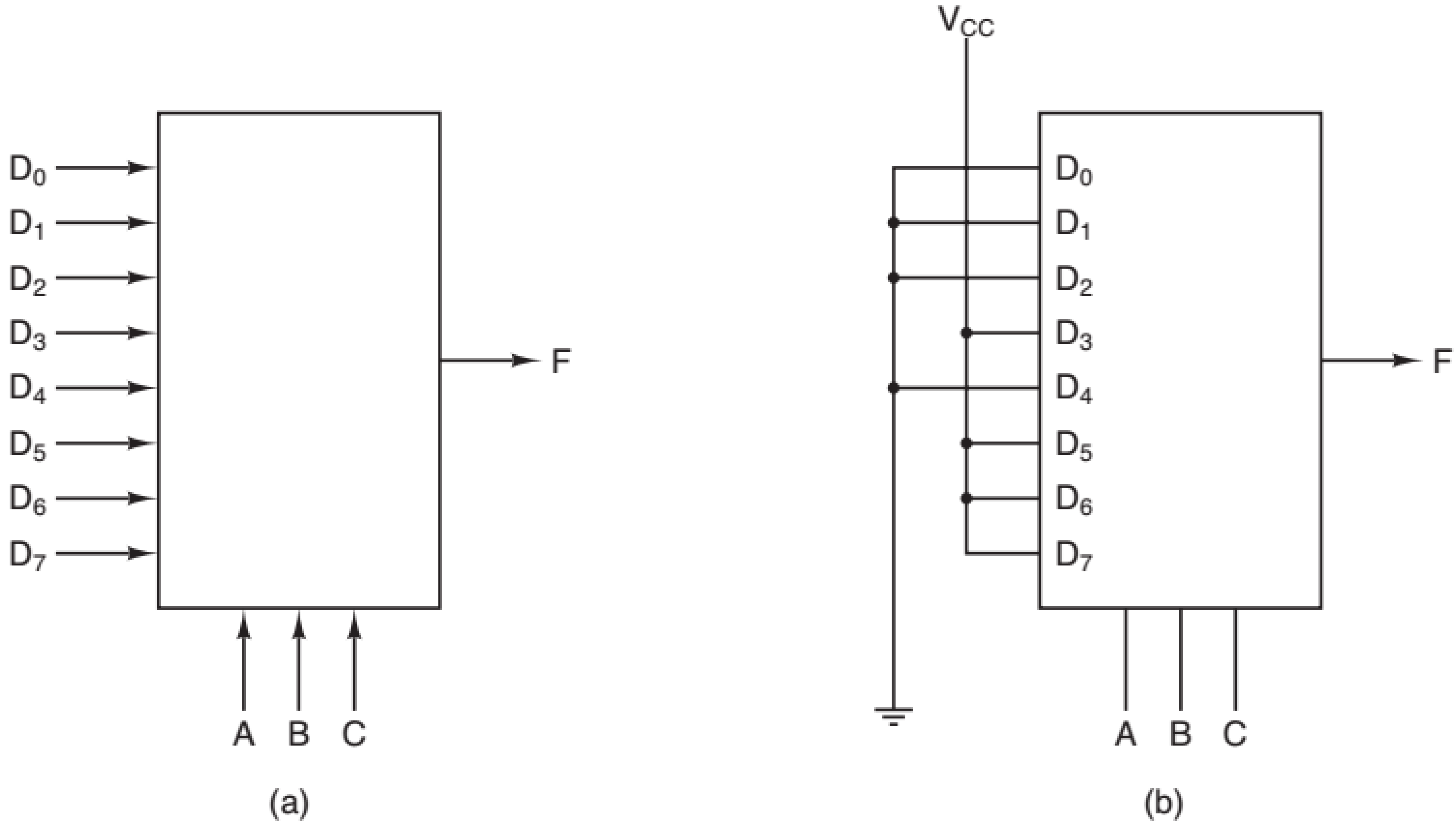
- Usando o multiplexador,
  - podemos executar a função majoritária da figura (a),
  - como mostrado na figura (b).
- Para cada combinação de A, B e C, uma das linhas de dados é selecionada.
- Cada entrada é ligada ou a V_{cc} (1 lógico) ou ao terra (0 lógico).
- O algoritmo para ligar as entradas é simples:
  - a entrada Di é a que tem o mesmo valor da linha i da tabela verdade.

2.3.1. Multiplexadores

- Na figura (a):
  - as linhas 0, 1, 2 e 4 são 0, portanto, as entradas correspondentes estão aterradas;
  - as linhas restantes são 1, portanto,
  - estão ligadas a 1 lógico.
- Dessa maneira, qualquer tabela verdade de três variáveis pode ser executada usando o chip da figura (a).

2.3.1. Multiplexadores

- (a) Multiplexador com oito entradas.
- (b) O mesmo multiplexador ligado para calcular a função majoritária.



2.3.1. Multiplexadores

- Acabamos de ver como um chip multiplexador pode ser usado para selecionar uma das diversas entradas e como ele pode implementar uma tabela verdade.
- Outra de suas muitas aplicações é como um conversor de dados paralelo para serial.
- Colocando 8 bits de dados nas linhas de entrada e então escalonando as linhas em sequência de 000 a 111 (binário), os 8 bits são colocados em série na linha de saída.
- Uma utilização típica da conversão paralela para serial é um teclado, onde cada acionamento de uma tecla define implicitamente um número de 7 ou 8 bits que deve ser enviado por um enlace serial, como USB.

2.3.1. Multiplexadores

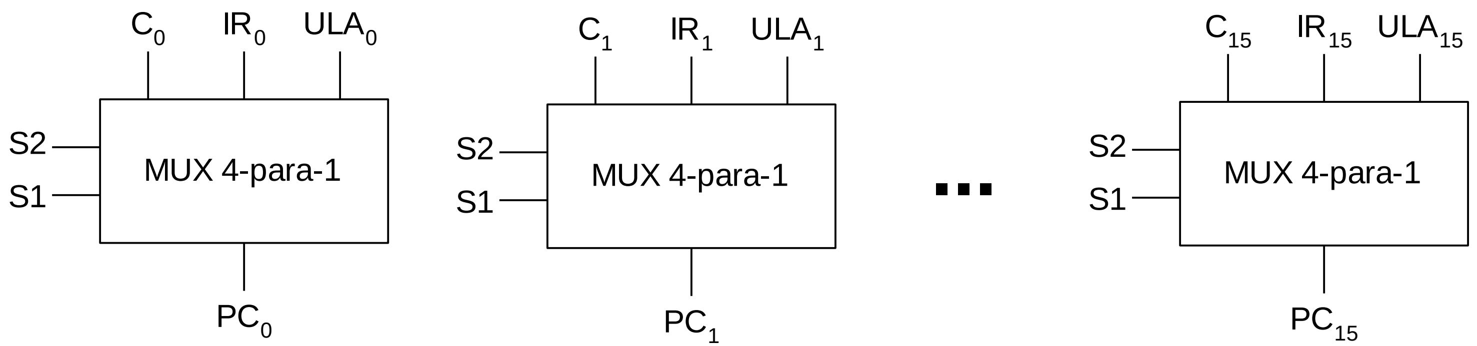
- O inverso de um multiplexador é um demultiplexador, que dirige sua única entrada até uma das 2^n saídas, dependendo dos valores das n linhas de controle.
- Se o valor binário das linhas de controle for k, é selecionada a saída k.

2.3.1. Multiplexadores

- **São usados em circuitos digitais para controle e roteamento de sinais**

Exemplo 2.3.2. Carga do Contador do Programa (PC)

- Pode vir de
  - Contador binário, que fornece o endereço seguinte
  - Registrador de instrução, no caso de desvio com endereçamento imediato
  - Saída da Unidade Lógica e Aritmética (ULA), no caso de desvio usando modo de endereçamento por deslocamento



2.3.1. Multiplexadores

Exercício 2.3.4. Montar a tabela verdade e o circuito combinatório de um multiplexador 8-para-1.

2.3.2. Decodificadores

- **Circuito com várias linhas de saída e que ativa apenas uma a cada instante dependendo do padrão de sinais nas linhas de entrada**
  - Pode ser projetado através da composição de mintermos

- **Utiliza *n* bits para selecionar uma das saídas**
  - Assim, no máximo, 2^n saídas podem ser selecionadas

2.3.2. Decodificadores

- Como um segundo exemplo, agora vamos examinar um circuito que toma um número de n bits como entrada e o usa para selecionar (isto é, definir em 1) exatamente uma das 2^n linhas de saída.
- Tal circuito, ilustrado para n = 3 na próxima figura, é denominado decodificador.
- Para ver como um decodificador pode ser útil, imagine uma pequena memória que consiste em oito chips, cada um contendo 256 MB.
- O chip 0 tem endereços de 0 a 256 MB, o chip 1 tem endereços de 256 MB a 512 MB e assim por diante.

2.3.2. Decodificadores

- Quando um endereço é apresentado à memória, os 3 bits de ordem alta são usados para selecionar um dos oito chips.
- Usando o circuito da figura, esses 3 bits são as três entradas, A, B e C.
- Dependendo das entradas, exatamente uma das oito linhas de saída, D_0,..., D_7, é 1; o resto é 0.
- Cada linha de saída habilita um dos oito chips de memória.
- Como só uma linha de saída é colocada em 1, apenas um chip é habilitado.

2.3.2. Decodificadores

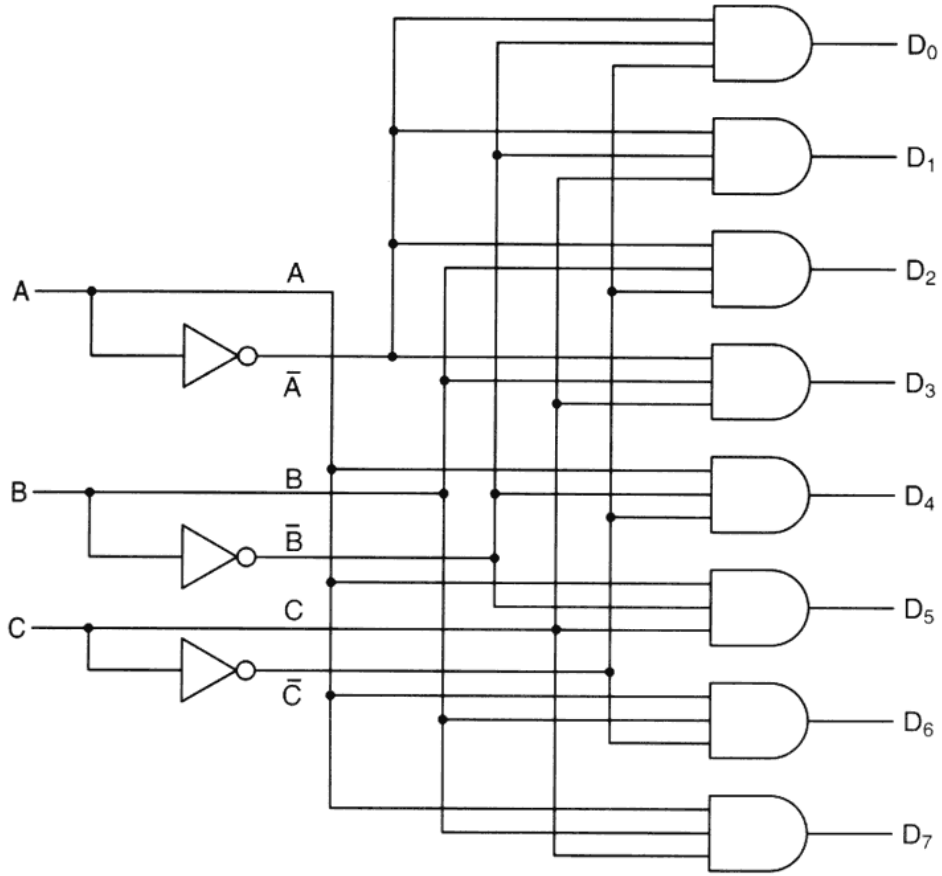
Exemplo 2.3.3. Decodificador com três entradas e oito saídas (2^3).

Qual a tabela verdade?

Quais as equações algébricas?

2.3.2. Decodificadores

Exemplo 2.3.3. Decodificador com três entradas e oito saídas (2^3)



2.3.2. Decodificadores

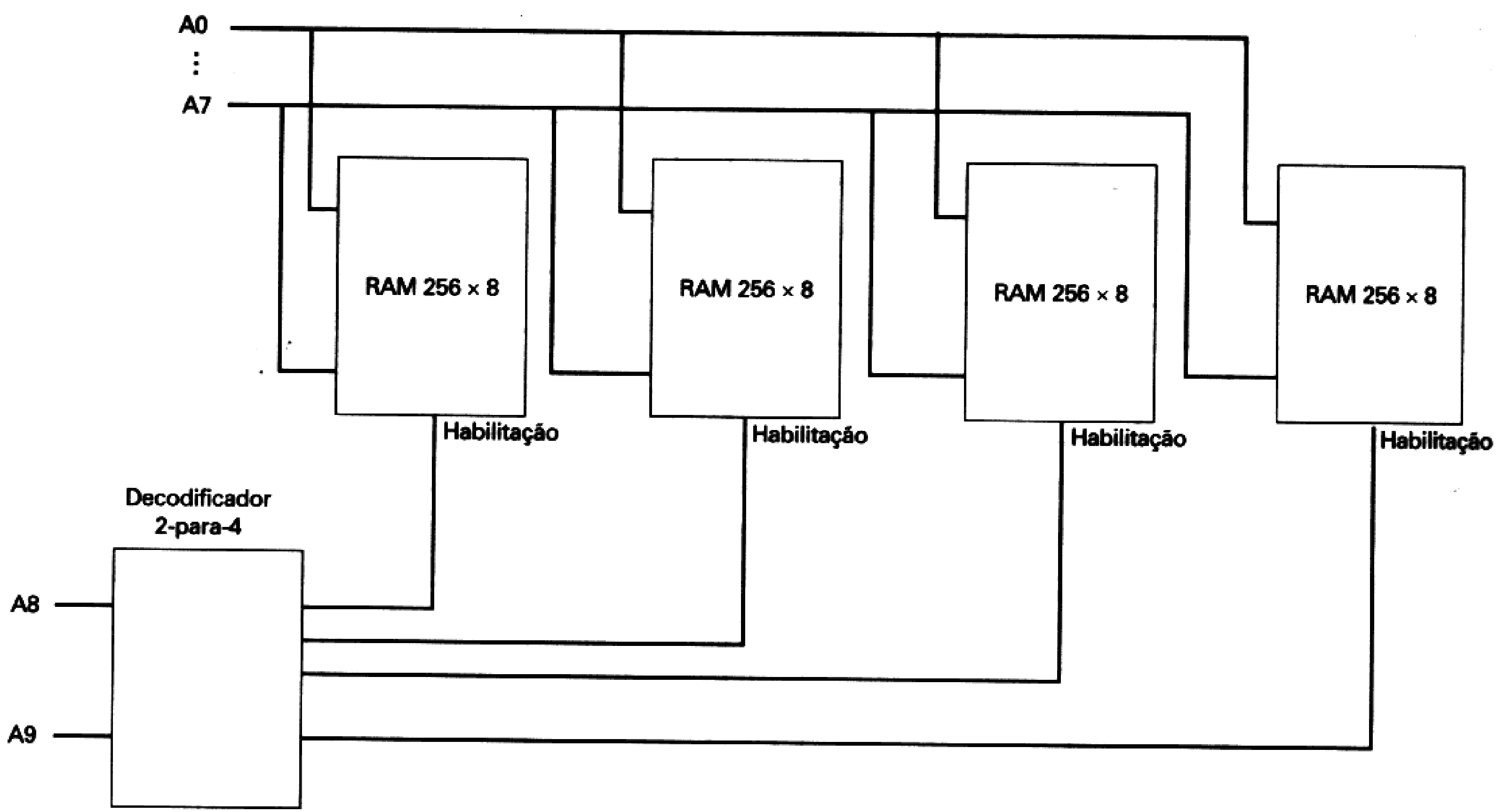
- **São usados para executar diferentes funções em um computador digital**

Exemplo 2.3.4. Decodificação de Endereços de Memória

- Construção de uma memória de 1Kbyte com quatro pastilhas de memória RAM de 256x8 bits. O espaço de endereçamento pode ser

Endereço		Pastilha
Hexadecimal	**Binário**	
0000-00FF	0000000000 - 0011111111	0
0100-01FF	0100000000 - 0111111111	1
0200-02FF	1000000000 - 1011111111	2
0300-03FF	1100000000 - 1111111111	3

Exemplo 2.3.4.


A0
⋮
A7
RAM 256 × 8
RAM 256 × 8
RAM 256 × 8
RAM 256 × 8
Habilitação
Habilitação
Habilitação
Habilitação
Decodificador
2-para-4
A8
A9

2.3.3. Comparadores

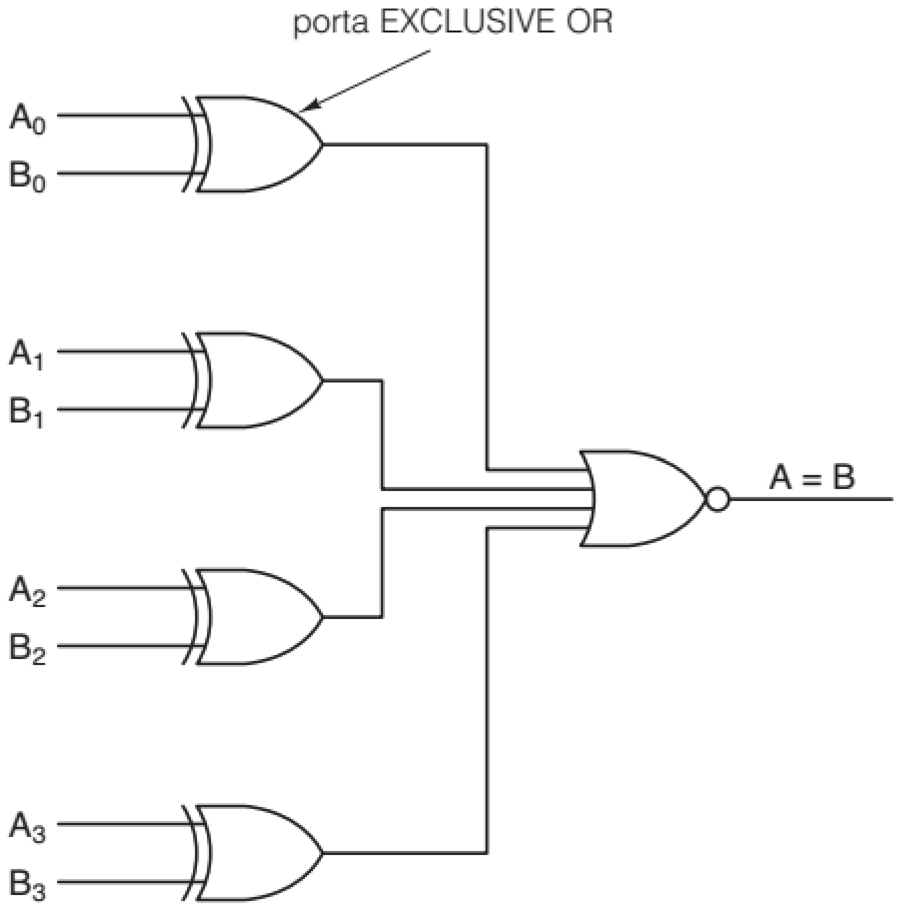
- Outro circuito útil é o **comparador**, que compara duas palavras de entrada.
- O comparador simples da figura toma duas entradas,
  - A e B, cada uma de 4 bits de comprimento,
  - e produz um 1 se elas forem iguais e um 0 se elas não o forem.
- O circuito é baseado na porta XOR (EXCLUSIVE OR), que produz um 0 se suas entradas forem iguais e um 1 se elas forem diferentes.
- Se as duas palavras de entrada forem iguais, todas as quatro portas xor devem produzir 0.

2.3.3. Comparadores

- Então, pode-se efetuar uma operação OR nesses quatro sinais;
  - se o resultado for 0, as palavras de entrada são iguais;
  - caso contrário, não.
- Em nosso exemplo, usamos uma porta nor como o estágio final para reverter o sentido do teste:
  - 1 significa igual, 0 significa diferente.

2.3.3. Comparadores

- Comparador simples de 4 bits.



2.3.4. Circuitos Aritméticos

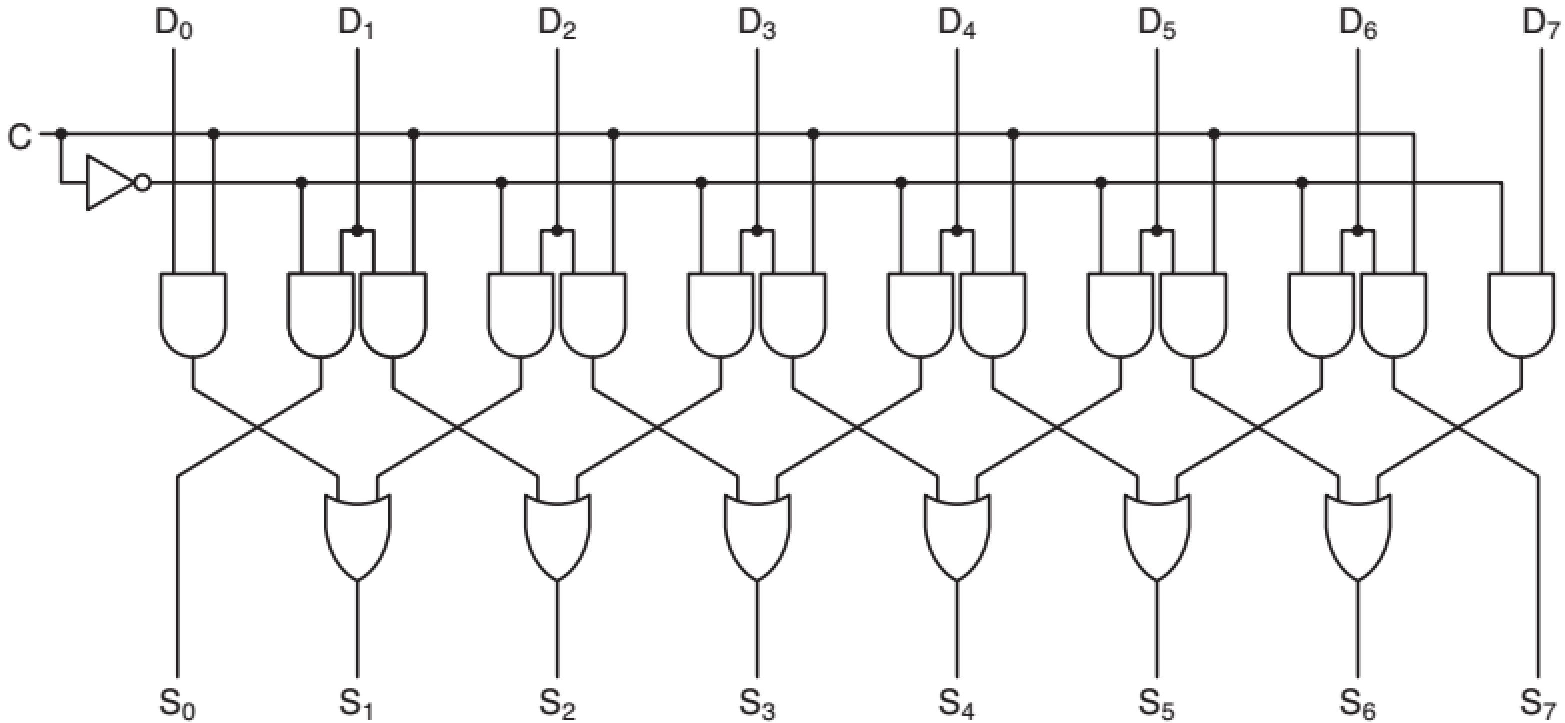
- Agora de passamos aos circuitos de uso geral discutidos anteriormente para circuitos combinatórios usados para operações aritméticas.
- Começaremos com um simples **deslocador** de 8 bits e em seguida veremos como são construídos os somadores e, por fim, estudaremos as unidades de lógica e aritmética, que desempenham um papel fundamental em qualquer computador.
- Nosso primeiro circuito aritmético é um deslocador de oito entradas e oito saídas (na figura).
- Oito bits de entrada são apresentados nas linhas D_0,..., D_7.

2.3.4. Circuitos Aritméticos

- A saída, que é apenas a entrada deslocada de 1 bit, está nas linhas S_0,..., S_7.
- A linha de controle, C, determina a direção do deslocamento, 0 para a esquerda e 1 para a direita.
- Quando o deslocamento for para a esquerda, um 0 é inserido no bit 7.
- De modo semelhante, quando o deslocamento for para a direita, um 1 é inserido no bit 0.

2.3.4. Circuitos Aritméticos

- Deslocador esquerda/direita de 1 bit.



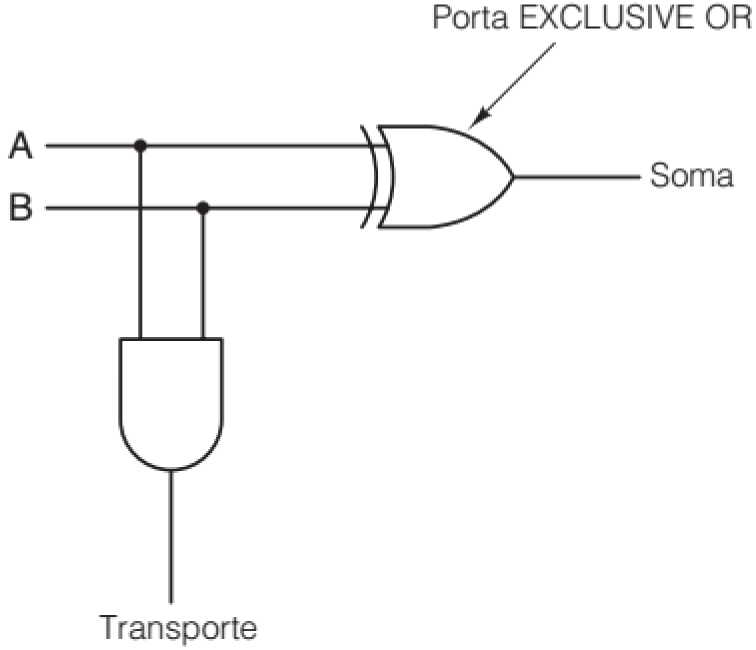
2.3.4. Circuitos Aritméticos

- Um circuito **somador** de hardware para efetuar adição é uma parte essencial de toda CPU.
- A tabela verdade para adição de inteiros de 1 bit é mostrada na figura (a).
- Há duas saídas presentes: a soma das entradas, A e B, e o transporte (vai-um) para a posição seguinte (à esquerda).
- Um circuito para calcular o bit de soma e o de transporte é ilustrado na figura (b).
- Esse circuito simples é conhecido como um meio-somador.

2.3.4. Circuitos Aritméticos

- (a) Tabela verdade para adição de 1 bit. (b) Circuito para um meio-somador.

A	B	Soma	Transporte
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



2.3.4. Circuitos Aritméticos

- Embora um meio-somador seja adequado para somar os bits de ordem baixa de duas palavras de entrada de múltiplos bits, ele não servirá para uma posição de bit no meio da palavra porque não trata o transporte de bit da posição à direita (vem-um).
- Em seu lugar, precisamos do somador completo da figura.
- Pelo circuito, deve ficar claro que um somador completo é composto de dois meios-somadores.
- A linha de saída Soma é 1 se um número ímpar A, B e o vem-um (carry in) forem 1.

2.3.4. Circuitos Aritméticos

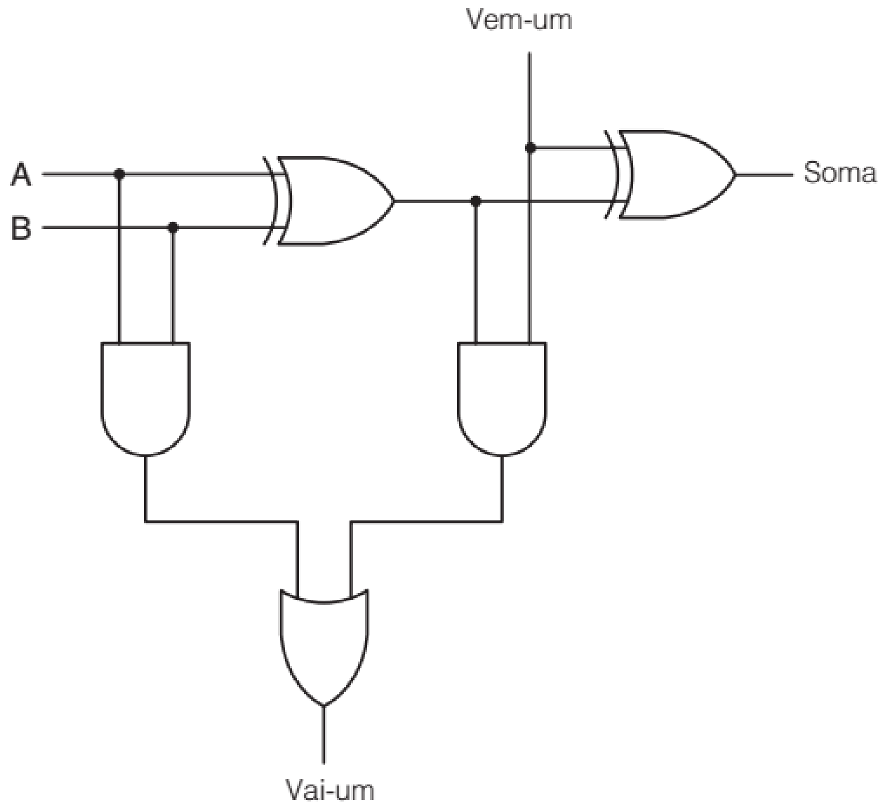
- O “vai-um” (*carry out*) é 1 se A e B forem ambos 1 (entrada esquerda para a porta *or*) ou se exatamente um deles for 1 e o bit de “vem-um” (*carry in*) também é 1.
- Juntos, os dois meios-somadores geram a soma e também os bits de transporte.
- Para construir um somador para palavras de 16 bits, por exemplo, basta repetir o circuito da figura (b) 16 vezes.
- O “vai-um” de um bit é usado como vem-um para seu vizinho da esquerda.
- O “vem-um” do bit da extrema direita está ligado a 0.

2.3.4. Circuitos Aritméticos

- (a) Tabela verdade para somador completo. (b) Circuito para um somador completo.

A	B	Vem- -um	Soma	Vai- -um
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a)



(b)

2.3.4. Circuitos Aritméticos

Esse tipo de somador é denominado somador de transporte encadeado porque,

- na pior das hipóteses, somando 1 a 111...111 (binário), a adição não pode ser concluída até que o vai-um tenha percorrido todo o caminho do bit da extrema direita até o da extrema esquerda.

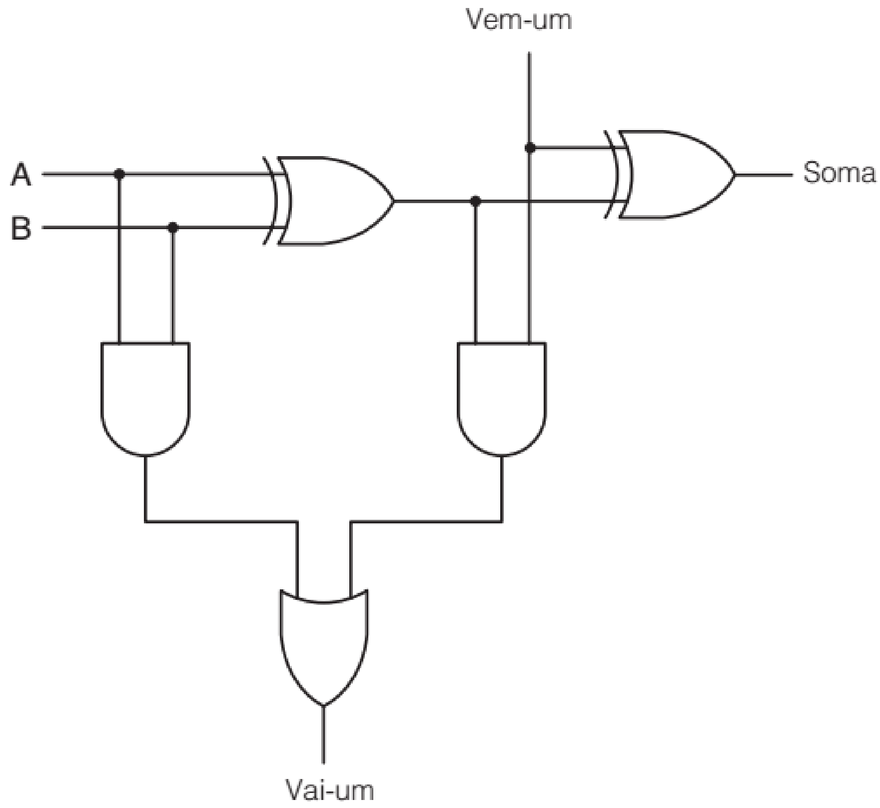
- Também existem somadores que não têm esse atraso e, portanto, são mais rápidos – em geral, são os preferidos.
- Como exemplo simples de um somador mais rápido, considere subdividir um somador de 32 bits em uma metade inferior e uma metade superior de 16 bits cada.
- Quando a adição começa, o somador superior ainda não pode trabalhar porque não sabe qual é o vem-um por 16 tempos de adição.

2.3.4. Circuitos Aritméticos

- (a) Tabela verdade para somador completo. (b) Circuito para um somador completo.

A	B	Vem- -um	Soma	Vai- -um
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a)



(b)

2.3.4. Circuitos Aritméticos

- Contudo, considere essa modificação no circuito.
- Em vez de uma única metade superior,
  - vamos dar ao somador duas metades superiores em paralelo duplicando o hardware da metade superior.
- Desse modo, agora o circuito consiste em três somadores de 16 bits:
  - uma metade inferior e duas metades superiores,
  - U0 e U1 que funcionam em paralelo.
- Um 0 é alimentado em U0 como vai-um; um 1 é alimentado em U1 como vai-um.
- Agora, ambos podem iniciar ao mesmo tempo do que a metade inferior, mas somente um estará correto.

2.3.4. Circuitos Aritméticos

- Após 16 tempos de adição de bits, já se saberá qual é o vem-um que deve ir para a metade superior,
  - portanto, agora já se pode selecionar a metade superior correta com base em duas respostas disponíveis.
- Esse estratagema reduz o tempo de adição por um fator de dois.
- Um somador como esse é denominado somador de seleção de transporte.
- Então, o estratagema pode ser repetido para construir cada somador de 16 bits com base em somadores de 8 bits repetidos.

2.3.4. Circuitos Aritméticos

- Grande parte dos computadores contém um único circuito para efetuar and, or e soma de duas palavras de máquina.
- No caso típico, tal circuito para palavras de n bits é composto de n circuitos idênticos para as posições individuais de bits.
- A figura próxima é um exemplo simples de um circuito desses, denominado **unidade lógica e aritmética (ULA)** (**Arithmetic Logic Unit – ALU**).
- Ela pode calcular qualquer uma das quatro funções – a saber, A and B, A or B, B ou A + B, dependendo de as linhas de entrada de seleção de função F_0 e F_1 conterem 00, 01, 10 ou 11 (binário).

2.3.4. Circuitos Aritméticos

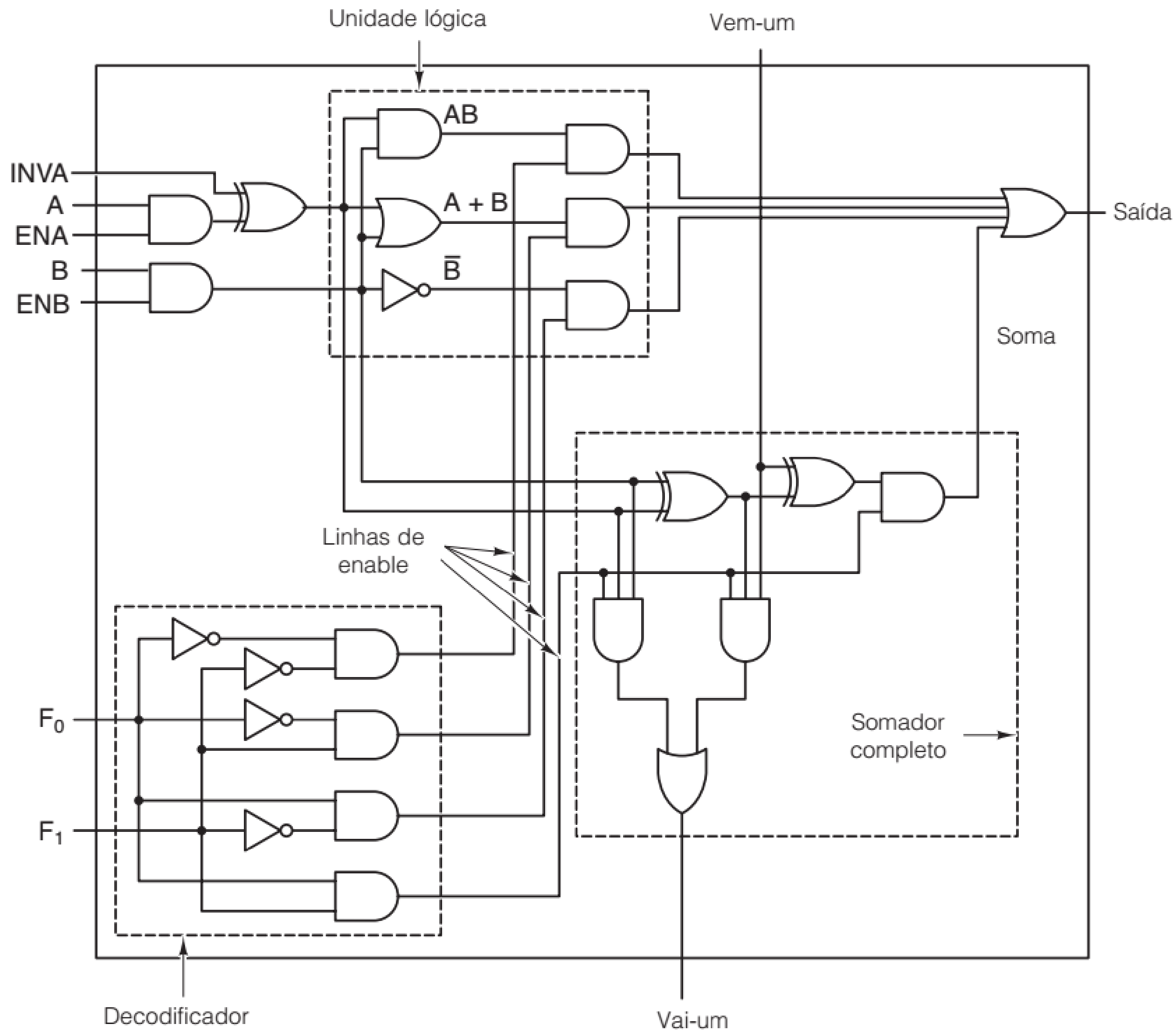
- Note que, aqui, A + B significa a soma aritmética de A e B, e não a operação booleana or.
- O canto inferior esquerdo de nossa ULA contém um decodificador de 2 bits para gerar sinais de enable (habilitação) para as quatro operações, com base nos sinais de controle F_0 e F_1.
- Dependendo dos valores de F_0 e F_1, exatamente uma das quatro linhas de habilitação é selecionada.
- Ativar essa linha permite que a saída para a função selecionada passe por ela até a porta OR final, para saída.

2.3.4. Circuitos Aritméticos

- O canto superior esquerdo contém a lógica para calcular A and B, A or, B e B, mas no máximo um desses resultados é passado para a porta or final,
  - dependendo das linhas de habilitação que saem do decodificador.
- Como exatamente uma das saídas do decodificador será 1, exatamente uma das quatro portas and que comandam a porta or será habilitada;
  - as outras três resultarão em 0, independente de A e B.

2.3.4. Circuitos Aritméticos

- ULA de 1 bit



2.3.4. Circuitos Aritméticos

- Além de poder usar A e B como entradas para operações lógicas ou aritméticas, também é possível forçar quaisquer delas para 0 negando ena ou enb, respectivamente.
- Também é possível obter A ativando inva.
- Em condições normais, ena e enb são ambas 1 para habilitar ambas as entradas e inva é 0.
- Nesse caso, A e B são apenas alimentados na unidade lógica, sem modificação.

2.3.4. Circuitos Aritméticos

- O canto direito inferior da ULA contém um somador completo para calcular a soma de A e B,
  - incluindo manipulação de transportes (vai-um e vem-um),
  - porque é provável que, em seu devido tempo,
  - vários desses circuitos serão ligados juntos para efetuar operações de palavra inteira.
- Na verdade, existem circuitos como o da figura próxima que são conhecidos como segmentos de bits (bit slices).
- Eles permitem que o projetista do computador monte uma ULA da largura que quiser.
- A figura mostra uma ULA de 8 bits montada com 8 segmentos (slices) de ULA de 1 bit.
- O sinal inc só é útil para operações de adição.

2.3.4. Circuitos Aritméticos

- Quando presente, aumenta o resultado (isto é, soma 1 a ele), possibilitando o cálculo de somas como A + 1 e A + B + 1.
- Anos atrás, um segmento de bit era na verdade um chip que você podia comprar.
- Hoje, é mais como uma biblioteca que um projetista de chip pode replicar quantas vezes quiser em um programa projeto-auxiliado-por-computador produzindo um arquivo de saída que direciona as máquinas de produção de chips.
- Mas a ideia, na essência, é a mesma.

2.3.4. Circuitos Aritméticos

- Oito segmentos (slices) de ULA de 1 bit conectados para formar uma ULA de 8 bits. Os sinais de habilitação e inversão não são mostrados por simplicidade.