

Circuitos Lógicos

Aula 24

Aula passada

- Máquina de estado
- Contadores genéricos

Antepassada

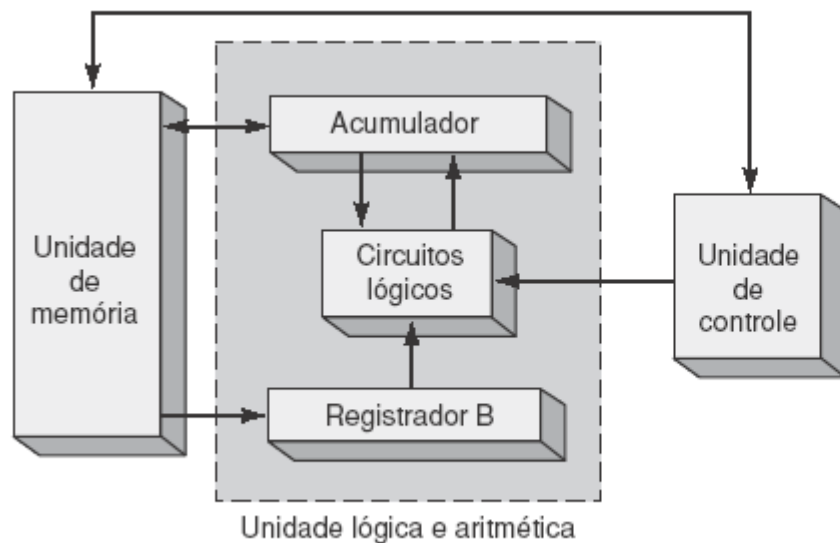
- Representação Complemento a 2
- Adição/Subtração

Aula de hoje

- ULA
- Circuitos para adição
- Propagação
- Complexidade

Unidade Lógica Aritmética

- ULA: módulo fundamental de qualquer microprocessador
- Implementa diversas funções lógicas e aritméticas
- Contém registradores internos
 - acumulador (resultados temporários)



- 1) Unidade de controle determina operação (via memória)
- 2) Número a ser somado é transferido da memória para B
- 3) B é somado com o acumulador
- 4) Resultado é armazenado no acumulador (e poder ser transferido para a memória)

Adição Binária

- Idêntica a aritmética na base 10

- mesmos princípios

- $9 + 5 = ?$

9 = 1001

5 = 0101

1110 = 14



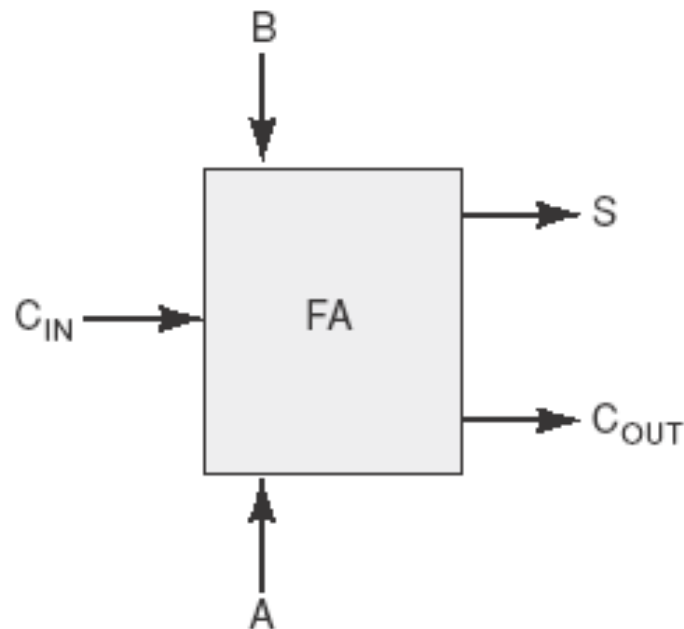
- Circuito que realiza esta operação?

- Ideia: todos bits são tratados igualmente

- Replicar circuito para somar um bit

Somador Completo

- Full Adder (FA): circuito que soma um bit
- Quantas entradas? Quantas saídas?
 - entradas: A , B , C_i . saída: S , C_o
- Circuito síncrono ou assíncrono?



- Organizar FAs em paralelo
- Um FA por bit a ser somado

Exemplo

- Soma do conteúdo dos registradores A e B
 - 5 bits
- Valor de C_0 ?

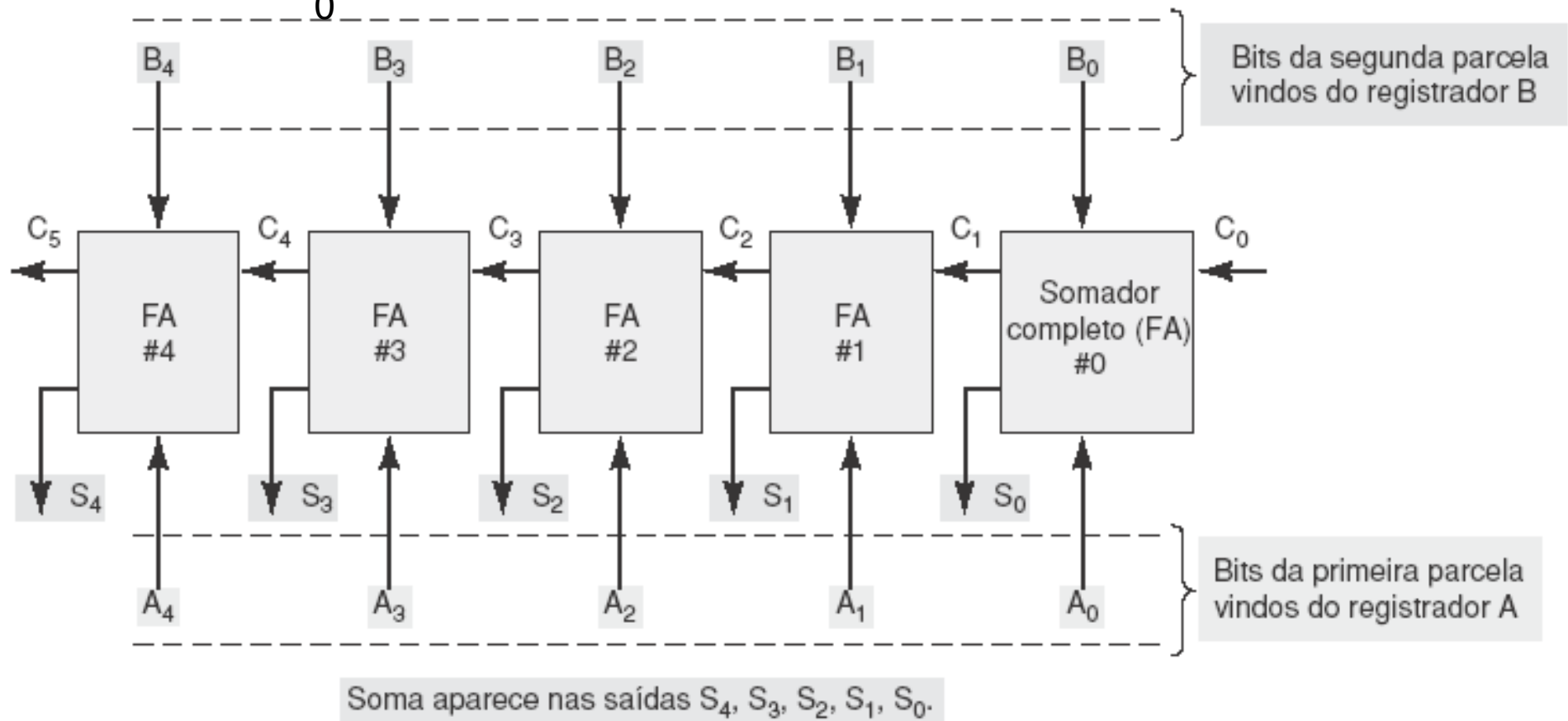


FIGURA 6.6

Diagrama em bloco de um circuito somador paralelo usando somadores completos.

Tabela Verdade

- Lógica (funcionalidade) por trás do FA
- ou seja, sua tabela verdade

Entradas de bits da primeira parcela	Entradas de bits da segunda parcela	Entradas de bits do carry	Saída de bits da soma	Saída de bits do carry
A	B	C_{IN}	S	C_{OUT}
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

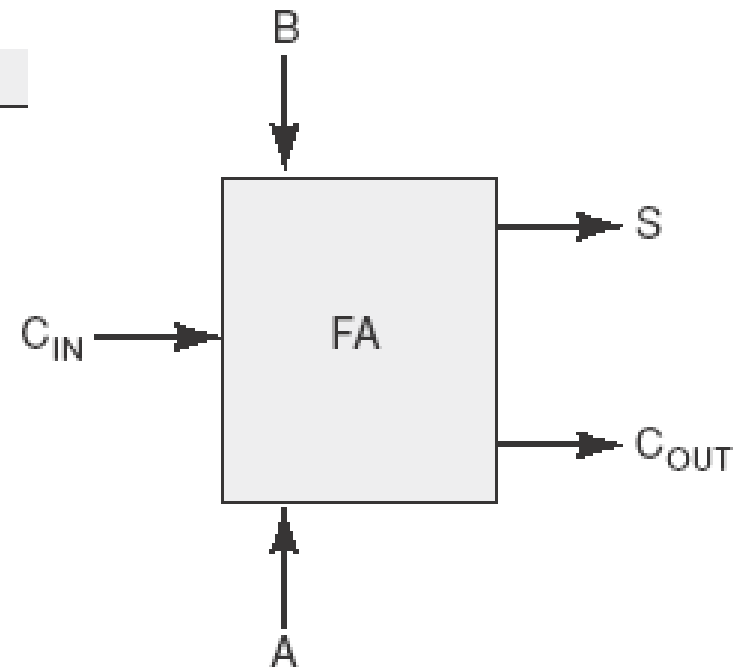


FIGURA 6.7

Tabela-verdade para um circuito somador completo.

Expressão Booleana

- Método da soma dos produtos
- $S = A'B'C_i + A'BC_i' + AB'C_i' + ABC_i$
- $C_o = A'BC_i + AB'C_i + ABC_i' + ABC_i$
- Simplificações?
- Mapa de Karnaugh?

Circuito do FA

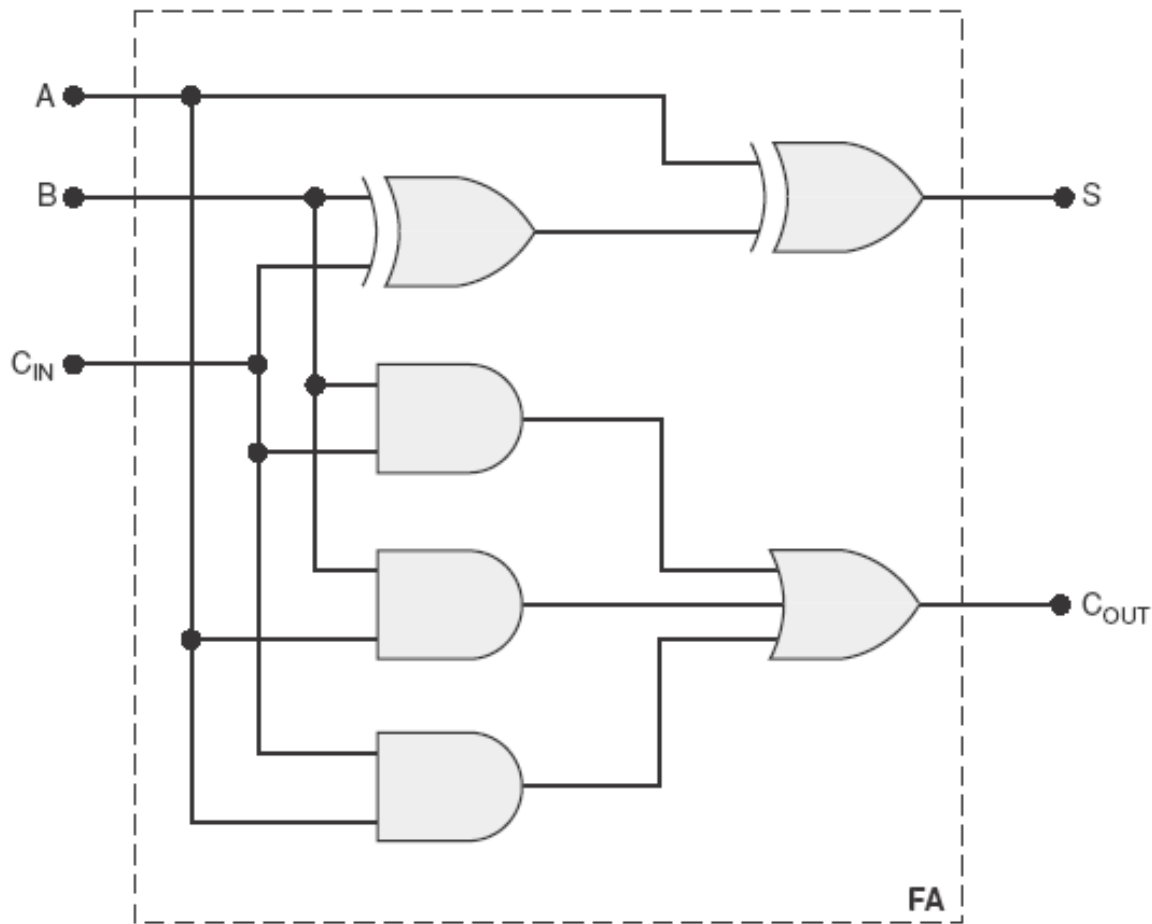
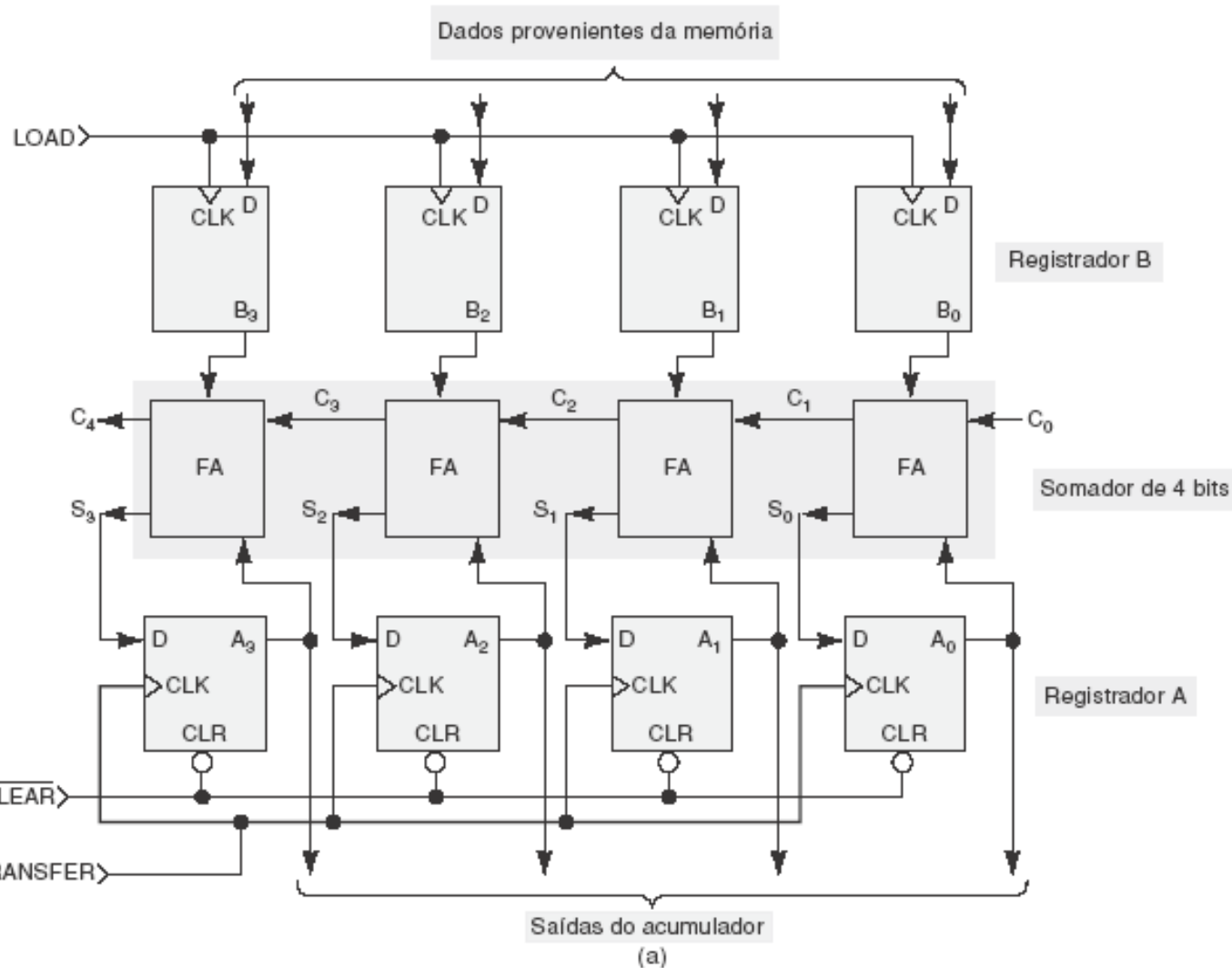


FIGURA 6.8
Circuito para um somador completo.

FA com Registradores

- Registradores alimentam o FA
- Resultado retorna ao registrador



- “Load” no clock: seta valor do regs B
- Clear assíncrono: zera valor do regs A
- “Load” no clock: set valor de A como soma de A com B

Tempo e Propagação



- Quanto tempo demora para realizar uma adição?
- Depende do número de bits!
Por que?

- *Propagação do carry!*
- FA de mais alta ordem precisa do carry do FA anterior, que precisa do anterior, etc.
- Seja M o tempo de propagação por um FA
 - ex. $M = 120\text{ns}$
- Tempo para somar números de 32 bits? E n bits?

Complexidade da Adição

- Observação fundamental: tempo para somar dois números depende da magnitude dos números!
- Sejam K e L dois números quaisquer. Quantos bits eles tem? Quanto tempo demora para somá-los?
- Pior caso: todos os bits do maior número trocam e temos overflow!

Aspecto central da Computação

- Tempo depende do tamanho da entrada
- Como projetar algoritmos eficientes?

Complexidade da Adição na Realidade

- Na realidade, ULA tem tamanho fixo
 - 8 bits, 16 bits, 32 bits ou 64 bits
- Tempo de adição é constante neste caso
 - Sacadas para reduzir tempo de propagação
- ULA pode trabalhar com mais de um tamanho fixo
 - tempo constante para cada tamanho
- E se quisermos somar números de 1024 bits?
 - Fazer operação em software, usando a ULA para somar tamanhos fixos (ex. 32 bits)
 - Tempo depende do tamanho (1024 bits)