

# 11-12 TREE AND ARRAY MULTIPLIERS AND VARIATIONS IN MULTIPLIERS

## Chapter Goals

Learn additional methods for synthesizing fast multipliers as well as other types of multipliers (bit-serial, modular, etc.)

## Chapter Highlights

- Building a multiplier from smaller units
- Performing multiply-add as one operation
- Using a multiplier for squaring is wasteful
- Building modular multipliers

Apr. 2012  
Computer Arithmetic, Addition/Subtraction

SLIDE 1

Aula sobre árvores de multiplicação

## TREE AND ARRAY MULTIPLIERS AND VARIATIONS IN MULTIPLIERS: TOPICS

### Topics in This Chapter

11.1. Full-Tree Multipliers

12.1 Divide-and-Conquer Designs

12.2 Additive Multiply Modules

12.5 The Special Case of Squaring

12.6 Modular Multipliers

Apr. 2012  
Computer Arithmetic, Multiplication

SLIDE 2

Vejamos os tópicos que serão apresentados.

## 11.1 FULL-TREE MULTIPLIERS

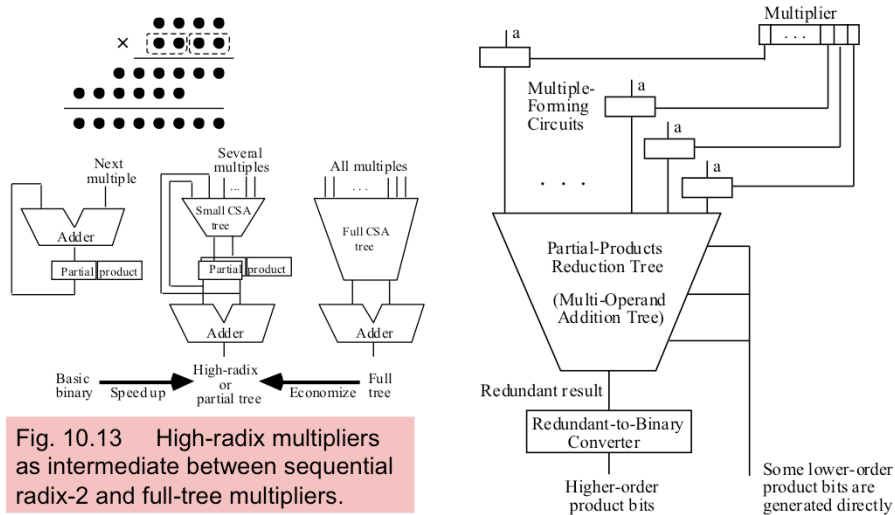


Fig. 10.13 High-radix multipliers as intermediate between sequential radix-2 and full-tree multipliers.

Fig. 11.1 General structure of a full-tree multiplier.

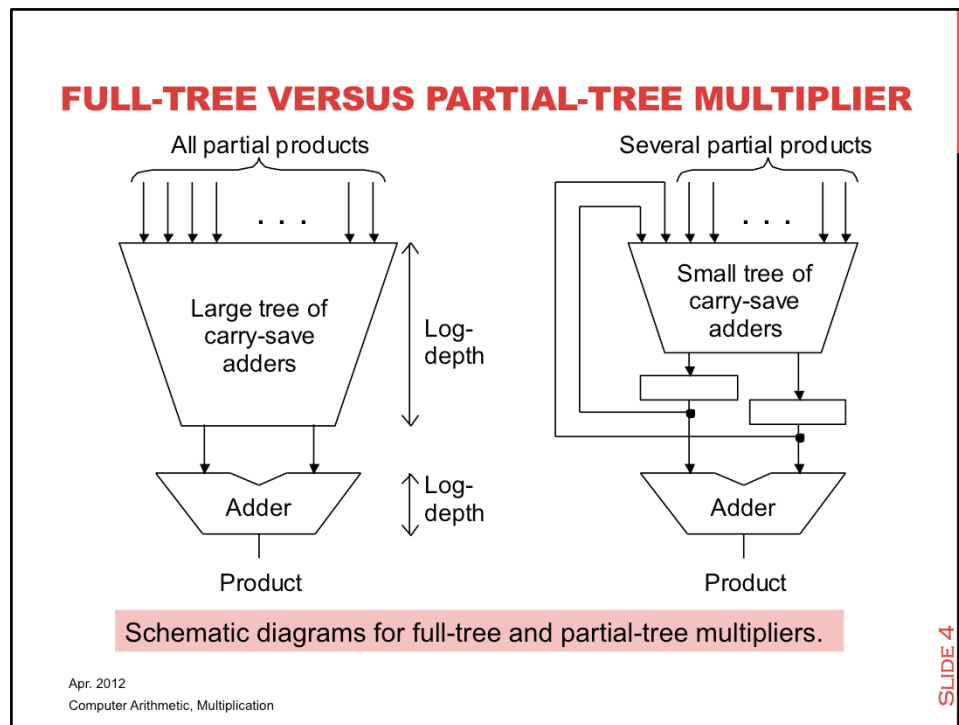
Apr. 2012  
Computer Arithmetic, Multiplication

SLIDE 3

As árvores de multiplicação consistem em casos em que todos os  $k$  múltiplos do multiplicando são produzidos em uma vez e uma só e aglomerados em uma árvore enquanto que CSAs são utilizados para reduzi-la a apenas dois operandos para a adição final. Como todos os múltiplos são combinados em uma única passagem, a árvore não requer links de feedback, tornando o pipelining uma aplicável e viável.

Vários múltiplos do multiplicando "a" são formados no topo.

Os circuitos de formação de múltiplos podem ser uma coleção de portas AND (multiplicador binário), geradores múltiplos de Booth (multiplicador recodificado) e assim por diante. Esses múltiplos são somados em uma árvore de redução de produtos parciais combinacionais, que produz sua soma de forma redundante. Finalmente, o resultado redundante é convertido para a saída binária padrão na parte inferior.



Se o custo de um multiplicador de árvore completa for muito alto para uma aplicação específica, uma variedade de designs considerada.

Tem-se como uma solução apresentada no slide, realizar a adição de  $k$ -operando necessária para a multiplicação  $k \times k$  via  $k / h$  passando por uma árvore CSA menor, seja  $h$  um número menor que  $k$ . A árvore demonstrada na figura direita possui CSA de entrada  $(h + 2)$  para adicionar o produto parcial cumulativo e “ $h$ ” novos operandos, realimentando a soma e o carry resultantes para serem combinados com o próximo lote de operandos  $h$ .

Quando “ $h$ ” é relativamente pequeno, digamos até 8 bits, tendemos a ver o multiplicador da Figura da direita como um High Radix multiplier. Por outro lado, quando “ $h$ ” é maior, então vemos o projeto como um multiplicador parcial da árvore

## 12.1 DIVIDE-AND-CONQUER DESIGNS

Building wide multiplier from narrower ones

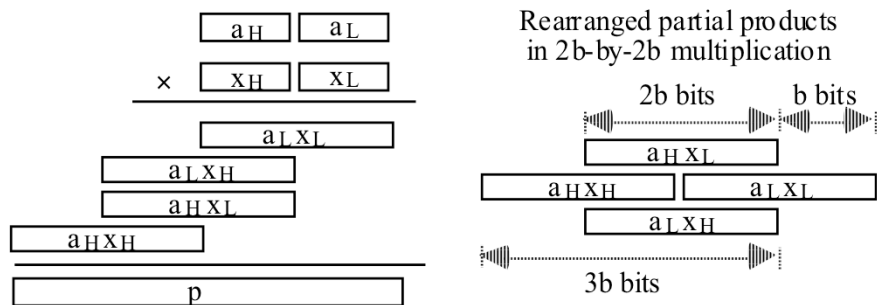


Fig. 12.1 Divide-and-conquer (recursive) strategy for synthesizing a  $2b \times 2b$  multiplier from  $b \times b$  multipliers.

Apr. 2012  
Computer Arithmetic, Multiplication

SLIDE 5

Agora, suponha que tenhamos multiplicadores  $b \times b$  e gostaríamos de usá-los para sintetizar um multiplicador  $2b \times 2b$ . Denotando as metades superior e inferior do multiplicando (multiplicador) por  $a_H$  e  $a_L$  ( $x_H$  e  $x_L$ ), podemos usar quatro multiplicadores  $b \times b$  para calcular os quatro produtos parciais  $a_L x_L$ ,  $a_L x_H$ ,  $a_H x_L$  e  $a_H x_H$  como mostrado na Fig. 12.1 (da esquerda). Esses quatro valores devem então ser somados para obter o produto final. Na verdade, conforme mostrado na Fig. 12.1 (da direita), apenas três valores precisam ser adicionados, uma vez que os produtos parciais não sobrepostos  $a_H x_H$  e  $a_L x_L$  podem ser vistos como um único número de  $4b$  bits.

## GENERAL STRUCTURE OF A RECURSIVE MULTIPLIER

$2b \times 2b$  use (3; 2)-counters  
 $3b \times 3b$  use (5; 2)-counters  
 $4b \times 4b$  use (7; 2)-counters

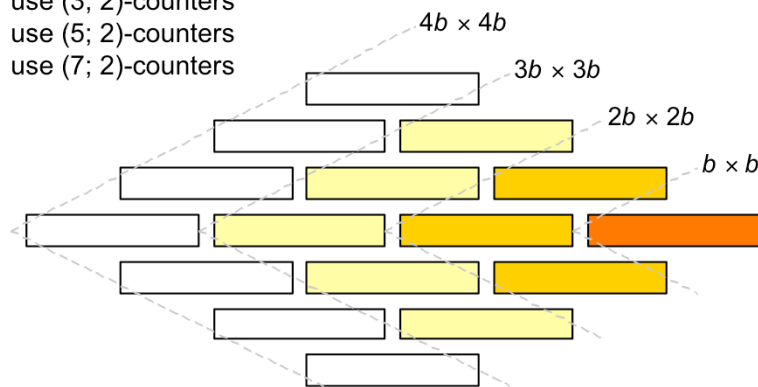


Fig. 12.2 Using  $b \times b$  multipliers to synthesize  $2b \times 2b$ ,  $3b \times 3b$ , and  $4b \times 4b$  multipliers.

Apr. 2012  
 Computer Arithmetic, Multiplication

SLIDE 6

Multiplicadores maiores, como  $3b \times 3b$  ou  $4b \times 4b$ , podem ser sintetizados de forma semelhante a partir dos blocos de construção do multiplicador  $b \times b$ .

A Figura 12.2 mostra que a multiplicação  $3b \times 3b$  leva a cinco números, enquanto a multiplicação  $4b \times 4b$  produz sete números. Portanto, podemos completar o processo de multiplicação nesses dois casos usando uma linha de contadores (5; 2) ou (7; 2), seguido por um somador final de  $5b$  ou  $7b$  bits, respectivamente. Observe que  $b$  bits do produto são obtidos diretamente de um pequeno multiplicador em cada caso.

## PROBLEMAS

**Problema 11.1.** Para uma multiplicação de dois operandos de 24 bits, aplique o método de dividir para conquistar e obtenha o custo e caminho crítico dos blocos considerando  $A_{FA}$  e  $T_{FA}$  como a área e atraso por *Full-Adder*, e  $0,5 \times A_{FA}$  e  $0,5 \times T_{FA}$  para o *Half-Adder*,  $(a/2) \times A_{FA}$  e  $(a/2) \times T_{FA}$  para o  $(2^a:1)$  MUX.

SLIDE 7

Gabarito no Moodle

## 12.2 ADDITIVE MULTIPLY MODULES

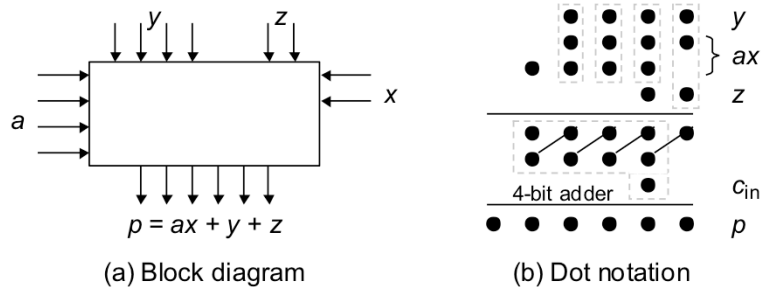


Fig. 12.4 Additive multiply module with  $2 \times 4$  multiplier ( $ax$ ) plus 4-bit and 2-bit additive inputs ( $y$  and  $z$ ).

$$b \times c \text{ AMM} \begin{cases} b\text{-bit and } c\text{-bit multiplicative inputs} \\ b\text{-bit and } c\text{-bit additive inputs} \\ (b + c)\text{-bit output} \end{cases}$$

$$(2^b - 1) \times (2^c - 1) + (2^b - 1) + (2^c - 1) = 2^{b+c} - 1$$

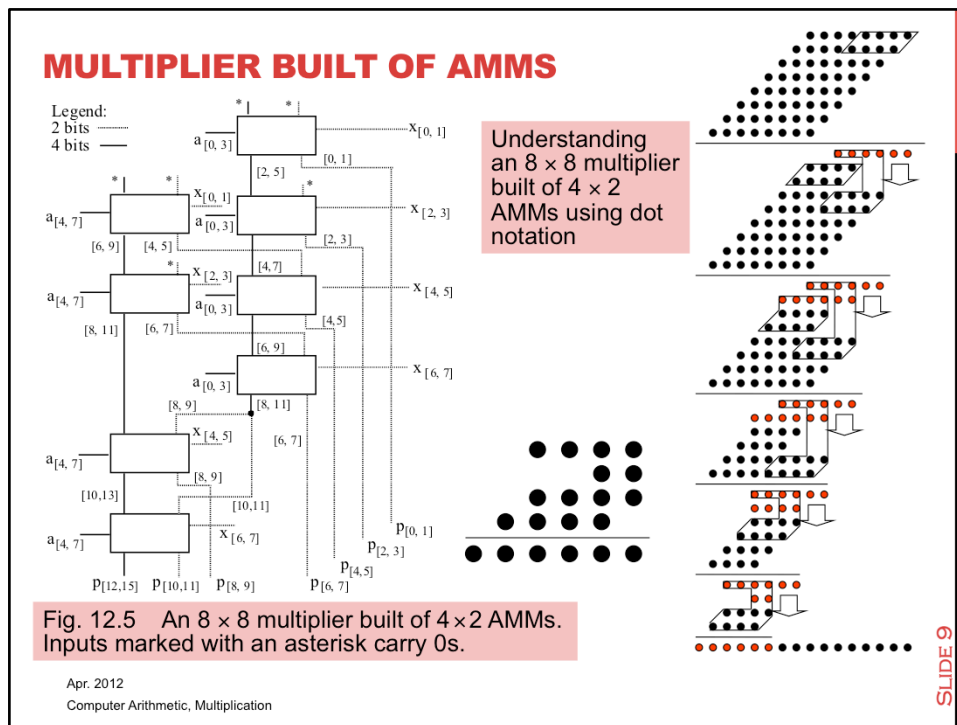
Apr. 2012  
Computer Arithmetic, Multiplication

SLIDE 8

Para otimizar a solução ainda mais, se pudermos combinar as funções de multiplicação e adição em uma unidade, talvez um único tipo de módulo seja suficiente para implementar esses multiplicadores. Essa seria a ideia por trás dos módulos de multiplicação aditivos (AMMs).

O AMM na Fig. 12.4 (a) realiza o cálculo  $p = ax + y + z$ , onde “a” e “y” são números de 4 bits e “x” e “z” são números de 2 bits. O valor máximo do resultado “p” é  $(15 \times 3) + 15 + 3 = 63$ , que pode ser representado com 6 bits. A Figura 12.4b mostra uma implementação deste AMM usando quatro somadores completos (FAs), representados como caixas envolvendo três pontos e um somador de 4 bits.





A Figura 12.5 mostra como um multiplicador  $8 \times 8$  pode ser construído a partir de oito AMMs do tipo representado na Figura 12.4. Observe que oito multiplicadores  $4 \times 2$  teriam sido necessários para este projeto; portanto, o número de módulos é mínimo. Cada AMM é mais lento do que um multiplicador  $4 \times 2$  em, no máximo, um nível FA. Portanto, o atraso na Figura 12.5 que é atribuível à função de adição não é maior que seis atrasos FA (o caminho crítico passa por seis AMMs). Assim, dado que o custo de um AMM  $4 \times 2$  é menor do que os custos combinados de um multiplicador  $4 \times 2$  e um somador de 4 bits, o projeto mostrado na Fig. 12.5 é muito econômico.

## PROBLEMAS

**Problema 12.1.** Projete um AMM  $2 \times 2$ , com duas entradas de soma de dois bits usando unicamente 4 full adders e 4 portas AND.

- a) Mostre como conectar os AMMs projetados para projetar um multiplicador  $4 \times 4$ .
- b) Determine o caminho critico usando o Full adder como unidade de atraso.
- c) Pode ser usado o multiplicador do apartado a como um AMM  $4 \times 4$ ?

**Problema 12.2.** Projete os seguintes AMMs usando unicamente  $2 \times 4$  AMMs:

- a)  $4 \times 4$  AMM;
- b)  $2 \times 8$  AMM;
- c)  $6 \times 6$  AMM
- d)  $4 \times 8$  AMM
- e)  $4 \times 8$  AMM (usando  $4 \times 4$  AMMs).
- f) Compare a eficiência de d) e e) em área e atraso considerando  $A_{FA}$  e  $T_{FA}$  como a área e atraso por *Full-Adder*, e  $0,5 \times A_{FA}$  e  $0,5 \times T_{FA}$  para o *Half-Adder*.

**Problema 12.3.** Projete o circuito AMM da seguinte expressão:  $A \times B \times C + 2^b D + 2^c E + 2^a F$ , onde A, D tem a=4 bits, B, E tem b=3 bits e C, F tem c=2 bits.

SLIDE 10

Gabarito no Moodle

## 12.5 THE SPECIAL CASE OF SQUARING

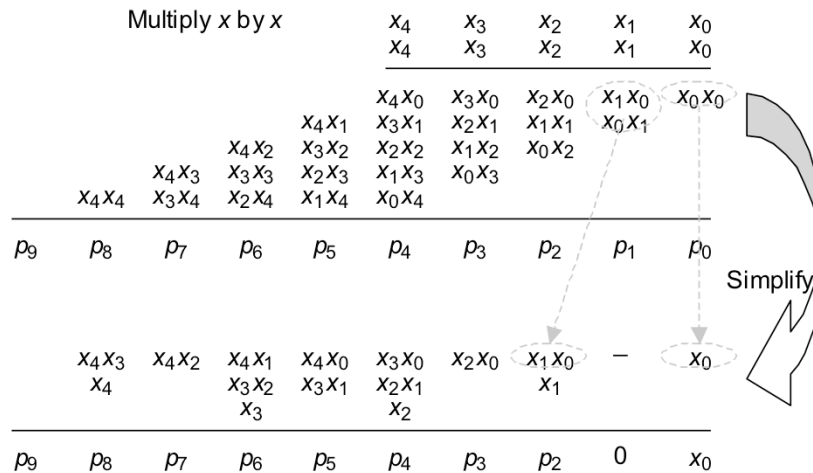


Fig. 12.18 Design of a 5-bit squarer.

Apr. 2012  
Computer Arithmetic, Multiplication

SLIDE 11

Considere o problema de elevação ao quadrado de um inteiro binário sem sinal de 5 bits ( $x_4 x_3 x_2 x_1 x_0$ ) (base 2). Conforme mostrado na Fig. 12.18, a matriz de produtos parciais pode ser consideravelmente simplificada antes de realizar a adição de multioperandos. Um termo  $x_i x_j$  se reduz para  $x_i$  e um par de termos  $x_i x_j$  e  $x_j x_i$ , em qualquer coluna, pode ser substituído por  $x_i x_j$  na próxima coluna superior. A matriz de produtos parciais simplificada resultante, para o exemplo de 5 bits, é mostrada na parte inferior da figura. Vemos que os dois LSBs do quadrado são obtidos sem esforço e que o cálculo dos bits restantes envolve uma adição de três operandos em oposição a uma adição de cinco operandos necessária para a multiplicação  $5 \times 5$ .

## PROBLEMAS

**Problema 12.4.** Projete a estrutura do multiplicador quadrático RNS para os seguintes módulos:

- a) 29;
- b) 31;
- c) 13.

SLIDE 12

Gabarito no Moodle