Universidade Tecnológica Federal do Paraná – UTFPR
Bacharelado em Ciência da Computação

BCC33B – Arquitetura e Organização de Computadores

Prof. Paulo C. Gonçalves
paulogoncalves@utfpr.edu.br

Material desenvolvido pelo
Prof. Rogério A. Gonçalves

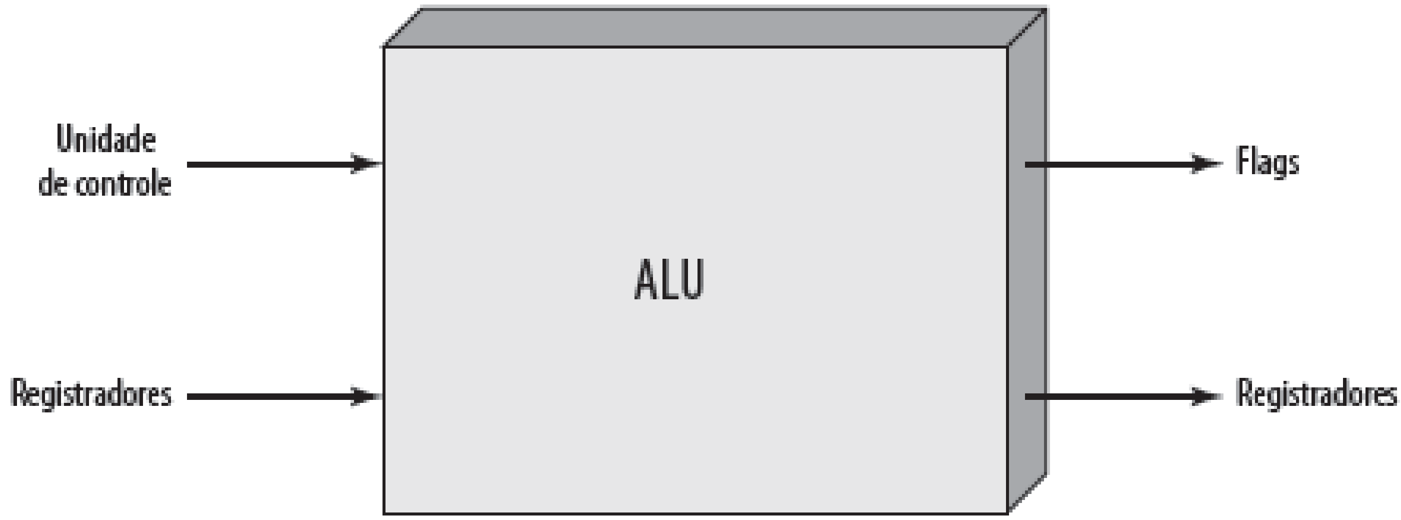
ULA

Unidade Lógica e Aritmética

Unidade Lógica e Aritmética

- **Realiza as operações lógicas e aritméticas**
  - add, sub, and, or...
- **Faz os cálculos.**
- **Trata de inteiros.**
- **Pode tratar de números de ponto flutuante (reais).**
- **Pode ser FPU separada (coprocessador matemático).**
- **Pode estar em chip de FPU separado (486DX +).**

Unidade Lógica e Aritmética



Representação de Dados

Como a informação é representada num processador?

- **A representação de um dado corresponde aos dígitos que escrevemos para simbolizá-lo.**
- **Exemplos:**

Valor (quantidade) 12

Representações:

Hexadecimal ⇒ C
Romano ⇒ XII
Binário ⇒ 1100

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Representação de Dados

Números de Ponto Fixo (Inteiros)

Números de Ponto Fixo Sem Sinal: usam representação binária convencional

Exemplo:

Binário	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

O valor do número é inteiro. Nenhum bit é usado para representar sinal.

Representação de Dados

Números de Ponto Fixo (Inteiros)

Números de Ponto Fixo Com Sinal

Existem 4 Métodos de Representação:

1. **Sinal Magnitude**
2. **Complemento de 1**
3. **Complemento de 2**
4. **Notação em Excesso**

Representação de Dados

Números de Ponto Fixo (Inteiros)

Representação Sinal Magnitude:

- **Em decimal para representarmos as quantias +12 e -12⇒usamos os sinais + e – para indicar se o número é positivo ou negativo**

- **Em Sinal Magnitude: Bit mais significativo (mais à esquerda) indica o sinal do número representado**

0 indica número positivo
1 indica número negativo

Os bits restantes representam a Magnitude (valor do dado)

Representação de Dados

Números de Ponto Fixo (Inteiros)

Exemplo na Representação Sinal Magnitude:

$+12_{10} \Rightarrow 00001100_2$

$-12_{10} \Rightarrow 10001100_2$

Só muda o bit de sinal

Os bits restantes representam a Magnitude (valor do dado)

Representação de Dados

Números de Ponto Fixo (Inteiros)

Observações para a Representação Sinal Magnitude:

1. Há 2 representações para o número 0

$+0_{10} \Rightarrow 00000000_2$

$-0_{10} \Rightarrow 10000000_2$

-Pode gerar erros de programação

-Requer hardware mais complexo para comparar com os dois 0s.

Dificulta testes

Representação de Dados

Números de Ponto Fixo (Inteiros)

Observações para a Representação Sinal Magnitude:

2. Intervalo de representação é menor, isto é, a quantidade de números representáveis é menor

011	+3
010	+2
001	+1
000	+0
100	-0
101	-1
110	-2
111	-3

Exemplo: $2^3=8$

Isso significa que com 3 bits poderíamos representar até 8 valores diferentes, mas devido às duas representações do valor 0 (+0 e -0) podemos representar até 7 valores diferentes

Representação de Dados

Números de Ponto Fixo (Inteiros)

Representação em Complemento de 1:

- Na representação em Complemento de 1 nós complementamos (invertemos) todos os bits 1 por 0 e os bits 0 por 1
- Exemplo:

$+12_{10} \Rightarrow \boxed{0}0001100_2$

$-12_{10} \Rightarrow \boxed{1}1110011_2$

Os números positivos também têm o bit mais significativo em 0 e os números negativos em 1

Representação de Dados

Números de Ponto Fixo (Inteiros)

Observações para a Representação Complemento de 1:

1. Há também 2 representações para o número 0

$+0_{10} \Rightarrow 00000000_2$

$-0_{10} \Rightarrow 11111111_2$

-Pode gerar erros de programação

-Requer hardware mais complexo para comparar com os dois 0s.

Dificulta testes →

A!=00000000

E

A!=11111111

Representação de Dados

Números de Ponto Fixo (Inteiros)

Observações para a Representação Complemento de 1:

2. Intervalo de representação é menor, isto é, a quantidade de números representáveis é menor

011	+3
010	+2
001	+1
000	+0
111	-0
110	-1
101	-2
100	-3

Exemplo: $2^3=8$

Isso significa que com 3 bits poderíamos representar até 8 valores diferentes, mas devido às duas representações do valor 0 (+0 e -0) podemos representar até 7 valores diferentes

Representação de Dados

Números de Ponto Fixo (Inteiros)

Representação em Complemento de 2:

- Na representação em Complemento de 2 nós complementamos (invertemos) todos os bits 1 por 0 e os bits 0 por 1 e somamos 1 ao resultado do Complemento de 1
- Exemplo:

$+12_{10} \Rightarrow 00001100_2$

Em Complemento de 2 os números positivos também têm o bit mais significativo em 0 e os números negativos em 1

$-12_{10} \Rightarrow C1 = 11110011_2$

$+1$

$-12_{10} = \underbrace{11110100_2}_{-12_{10} \text{ em Complemento de 2}}$

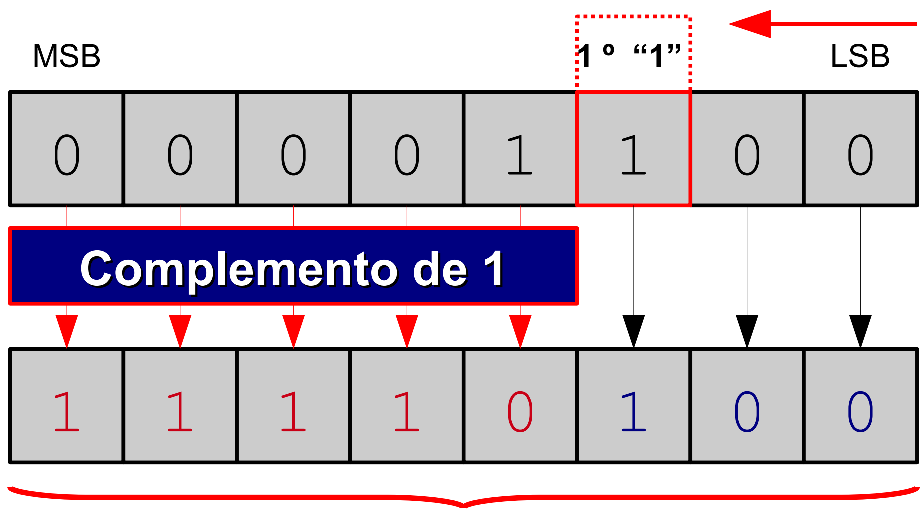
UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Representação de Dados

Representação em Complemento de 2

- **Método Alternativo**
- **Troque todos os bits à esquerda do bit 1 menos significativo.**
- **Passos:**
  1. **Comece à direita com o LSB e escreva os bits como eles aparecem até o primeiro 1 (inclusive ele).**
  2. **Tome o complemento de 1 dos bits restantes.**
- **Exemplo:**

$$+12_{10} \Rightarrow 00001100_2$$



-12_{10} em Complemento de 2

Representação de Dados

Representação em Complemento de 2

- **Método Alternativo**
- **Troque todos os bits à esquerda do bit 1 menos significativo.**
- **Passos:**
  1. **Comece à direita com o LSB e escreva os bits como eles aparecem até o primeiro 1 (inclusive ele).**
  2. **Tome o complemento de 1 dos bits restantes.**
- **Exemplo:**

$+15_{10} \Rightarrow 00001111_2$

-15_{10} em Complemento de 2

Representação de Dados

Representação em Complemento de 2

- **Método Alternativo**
- **Troque todos os bits à esquerda do bit 1 menos significativo.**
- **Passos:**
  1. **Comece à direita com o LSB e escreva os bits como eles aparecem até o primeiro 1 (inclusive ele).**
  2. **Tome o complemento de 1 dos bits restantes.**
- **Exemplo:**

$\boxed{+1_{10} \Rightarrow 00000001_2}$

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Representação de Dados

Números de Ponto Fixo (Inteiros)

Observações para a Representação Complemento de 2:

1. Há somente 1 representação para o número 0

$\boxed{+0_{10} \Rightarrow 00000000_2}$

$$-0_{10} \Rightarrow C1 = 11111111_2$$
$$+1$$
$$-0_{10} = 1\ \underbrace{00000000_2}_{-0_{10} \text{ em Complemento de 2}}$$

Carry é ignorado na conversão do número

Representação de Dados

Números de Ponto Fixo (Inteiros)

Observações para a Representação Complemento de 2:

2. Intervalo de representação é maior que dos outros métodos de representação anteriores porque só há uma representação para o Zero

011	+3
010	+2
001	+1
000	+0
000	-0
111	-1
110	-2
101	-3
100	-4

Intervalo maior: 8 representações diferentes

Representação de Dados

Números de Ponto Fixo (Inteiros)

Representação em Excesso (Bias ou Deslocamento):

- **A representação em Excesso tem o efeito de deslocar o número a ser representado, de forma que, o menor valor (negativo) corresponda à representação com todos os bits em zero e os valores sejam representados em ordem crescente, a partir do menor**
- **Exemplo em Excesso de 128:**

$+12_{10} \Rightarrow +12+128 = 140 = 10001100_2$

$-12_{10} \Rightarrow -12+128 = 116 = 01110100_2$

Representação de Dados

Números de Ponto Fixo (Inteiros)

Observações para a Representação Excesso:

1. Há somente 1 representação para o número 0
2. Intervalo de representação maior

$+127_{10} \Rightarrow +127+128 = 255 = 11111111_2$

...

$0_{10} \Rightarrow +0+128 = 128 = 10000000_2$

...

$-127_{10} \Rightarrow -127+128 = 1 = 00000001_2$

$-128_{10} \Rightarrow -128+128 = 0 = 00000000_2$

Com 8 bits pode-se representar $2^8=256$ números (de 0 a 255)

Ordem crescente facilita comparações entre os números

Representação de Dados

Resumo das Representações de Dados

Decimal	Sem Sinal	Sinal Magnitude	Complemento de 1	Complemento de 2	Excesso de 4
+7	111				
+6	110				
+5	101				
+4	100				
+3	011	011	011	011	111
+2	010	010	010	010	110
+1	001	001	001	001	101
+0	000	000	000	000	100
-0	-	100	111	000	100
-1	-	101	110	111	011
-2	-	110	101	110	010
-3	-	111	100	101	001
-4	-			100	000

Representação de Dados

Números em Ponto Flutuante (Reais)

Problema: Ponto Fixo requer uma quantidade muito grande de dígitos para representar números muito grandes ou muito pequenos

- **Exemplo:** Para representar 1 Trilhão ⇒ Requer 40 bits à esquerda do ponto fixo

- **Exemplo:** Para representar 1 Trilhonésimo no mesmo processador ⇒ Requer 40 bits à direita do ponto fixo

- **No total precisamos de 80 bits por número**

Representação de Dados

Números em Ponto Flutuante (Reais)

- **Exemplo: Número de Avogrado $+6,023 \times 10^{23}$**
- **Intervalo: 10^{23}**
- **Precisão: 6,023 (3 dígitos de precisão)**
- **Representação do Número em Notação Científica:**

ponto decimal

expoente

sinal

$$+6.023 \times 10^{23}$$

Mantissa

raiz (base)

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Representação de Dados

Números em Ponto Flutuante (Reais)

- **Obs: Há várias maneiras de se representar o mesmo número**
- **Exemplos:** $\boxed{3584{,}1 \times 10^{0} = 3{,}5841 \times 10^{3} = 0{,}35841 \times 10^{4}}$

Várias representações dificultam cálculos e comparações

↓

Necessidade de Normalização da representação

Representação de Dados

Números em Ponto Flutuante (Reais)

Normalização: o ponto é deslocado ("flutua") para a esquerda do dígito diferente de 0 mais à esquerda (bit mais significativo), e o expoente é ajustado

- **Exemplo:**

$\boxed{0{,}35841 \times 10^4}$ ← Forma Normalizada

- **Obs: Para representar 0 usa-se a mantissa com todos os valores em 0**

Representação de Dados

Números em Ponto Flutuante (Reais)

Observações para representação em Ponto Flutuante:

- **Para representar 0 usa-se a mantissa com todos os valores em 0**
- **Em binário, não há necessidade de se armazenar o dígito 1 “mais significativo” da mantissa (já se sabe que ele é 1) ⇒ esse bit é chamado de “bit escondido”. Sobra mais espaço para o número ser representado ⇒ aumenta a precisão**

Exemplo

$01101 \ldots \times 2^x$

$0{,}1101 \ldots \times 2^y$

Bit escondido

$101 \ldots \times 2^z$

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Representação de Dados

Números em Ponto Flutuante (Reais)

Padrão IEEE 754

- 1980: Padronização da representação em Ponto Flutuante pela IEEE (*Institute of Electrical and Electronics Engineers*)
- Padronização:
  - -Facilita a troca de dados entre diferentes computadores
  - -Facilita os algoritmos aritméticos de PF, pois tratam os os números sempre no mesmo formato
  - -Melhora a precisão dos números representados devido ao bit escondido

Representação de Dados

Números em Ponto Flutuante (Reais)

Padrão IEEE 754
(ANSI/IEEE 754-1985)

- **São três formas:**
  - **Precisão Simples:** 32 bits {**S** → 1, **E** → 8, **M** → 23}
    - ***E*** é representado em excesso de 127 e ***M*** efetivamente possui 24 bits, um bit escondido.
  - **Precisão Dupla:** 64 bits {**S** → 1, **E** → 11, **M** → 52}
    - ***E*** é representado em excesso de 1023 e ***M*** efetivamente possui 53 bits, um bit escondido.
  - **Precisão estendida:** 80 bits {**S** → 1, **E** → 15, **M** → 64}
    - ***E*** é representado em excesso de 16383 e ***M*** possui 64 bits, não tem bit escondido.

Representação de Dados

Números em Ponto Flutuante (Reais)

Formatos do Padrão IEEE 754

Precisão Simples

1	8	23
S	E	M

bit de sinal

expoente: excesso de 127

mantissa: (bit escondido)

Precisão Dupla

1	11	52
S	E	M

bit de sinal

expoente: excesso de 1023

mantissa: (bit escondido)

Precisão Estendida

1	15	64
S	E	M

bit de sinal

expoente: excesso de 16383

mantissa

Representação de Dados

Números em Ponto Flutuante (Reais)

Padrão IEEE 754

- **Exceções:**
  - O número *0,0* é representado por *0s* em todas as posições.
  - E *infinito* é representado com *1s* em todas as posições do expoente e *0s* em todas as posições da mantissa.

Representação de Dados

Formatos do Padrão IEEE 754

Precisão simples

Exemplo: converter o número decimal para binário

$$3{,}248\,x\,10^4 = 32480 = 111111011100000_2 = \boxed{1},\boxed{11111011100000}\,x\,2^{14}$$

O MSB não ocupa a posição de um bit porque ele é sempre 1

Mantissa

$\boxed{11111011100000000000000}$

Expoente (polarizado → em excesso de 127):

$$14+127=141=\boxed{10001101_2}$$

O número completo representado em ponto flutuante:

S	Expoente	Mantissa
0	1 0 0 0 1 1 0 1	1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Aritmética Computacional

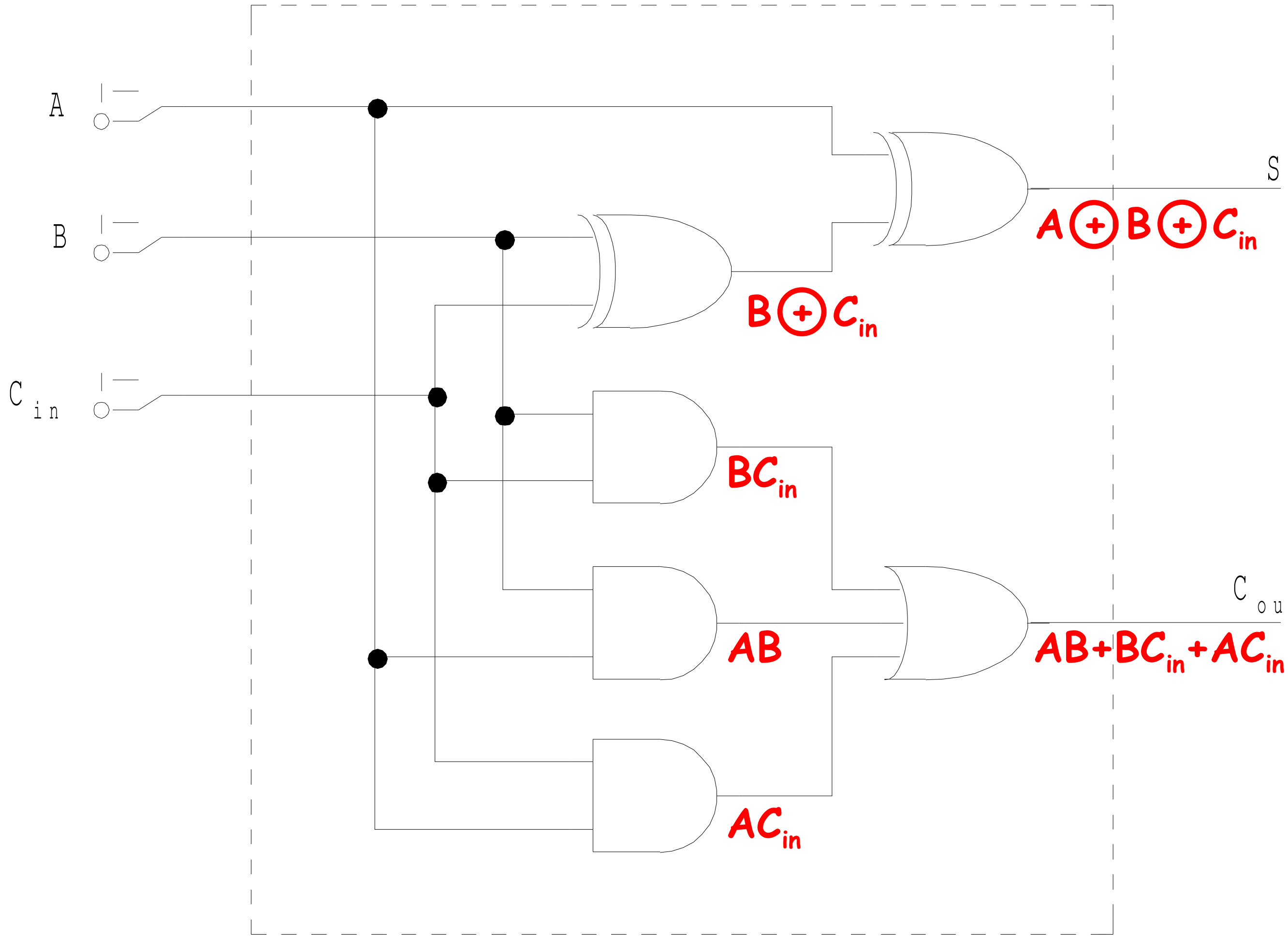
Revisão Adição e Subtração

Aritmética Computacional

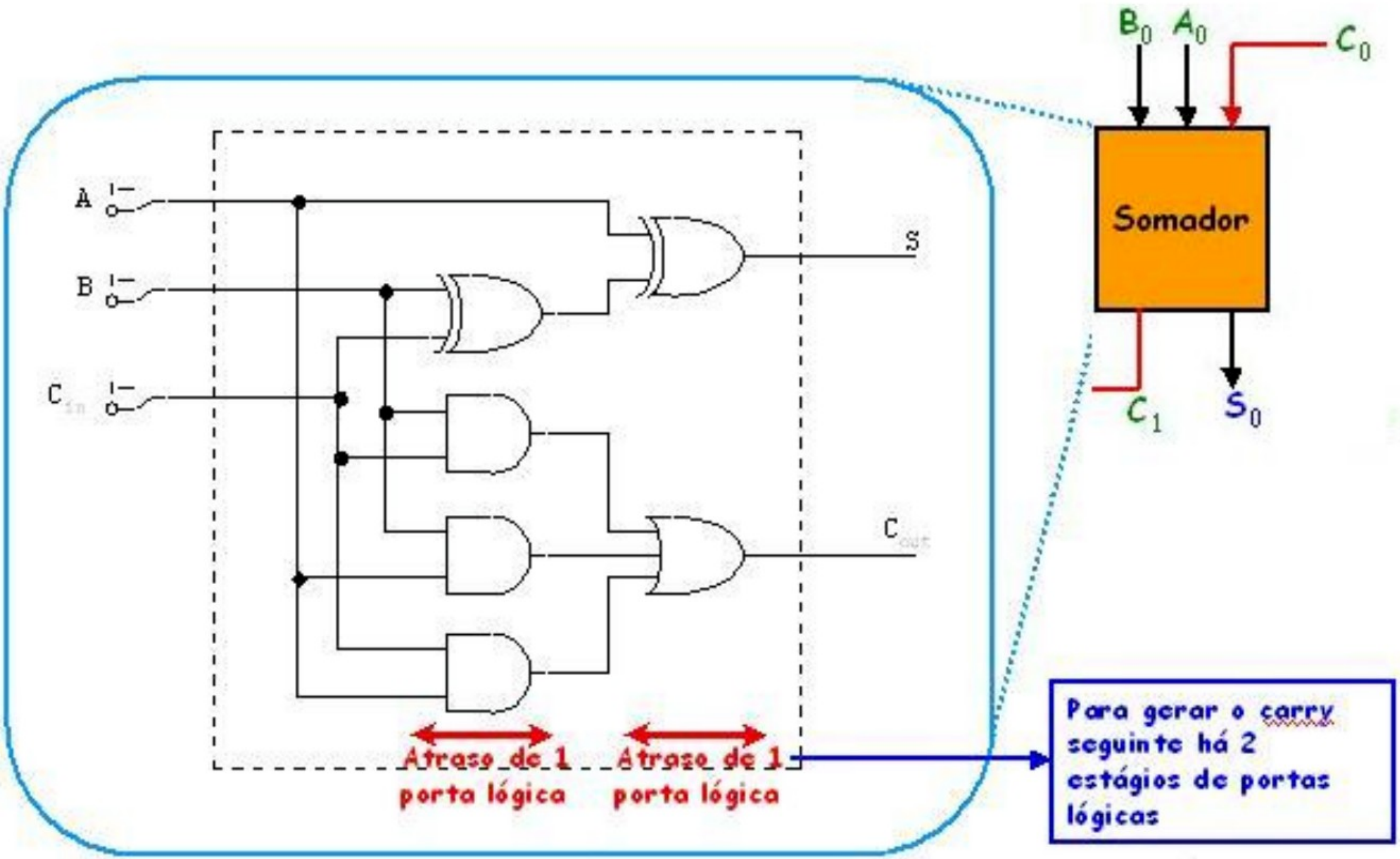
Circuito Somador

$S = A \oplus B \oplus C_{in}$

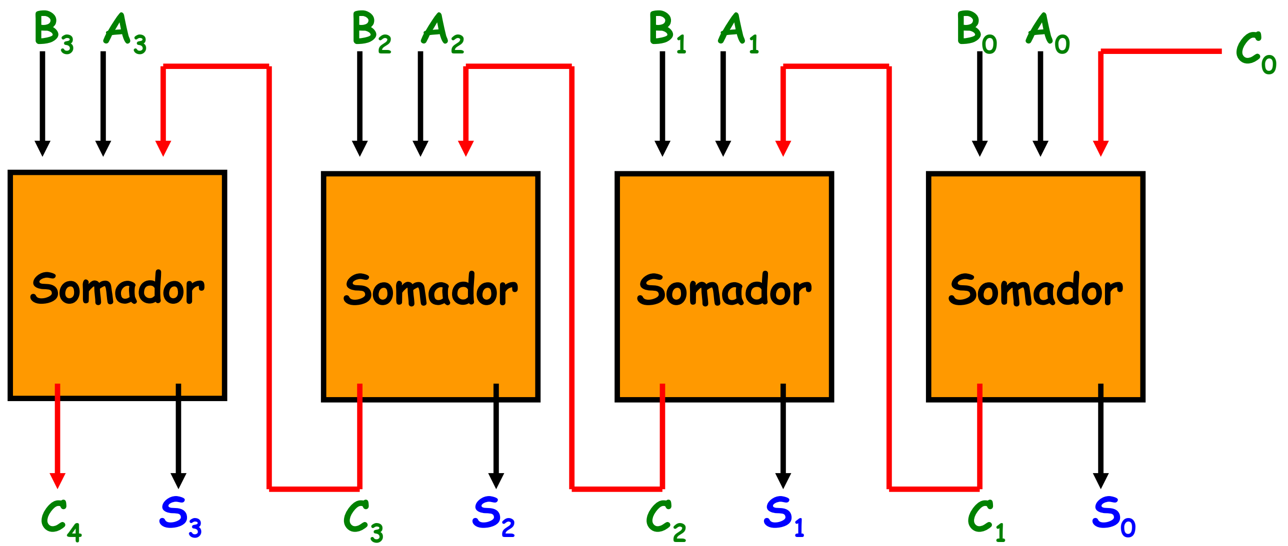
$C_{out} = AB + BC_{in} + AC_{in}$



Aritmética Computacional



Somador de 4 bits



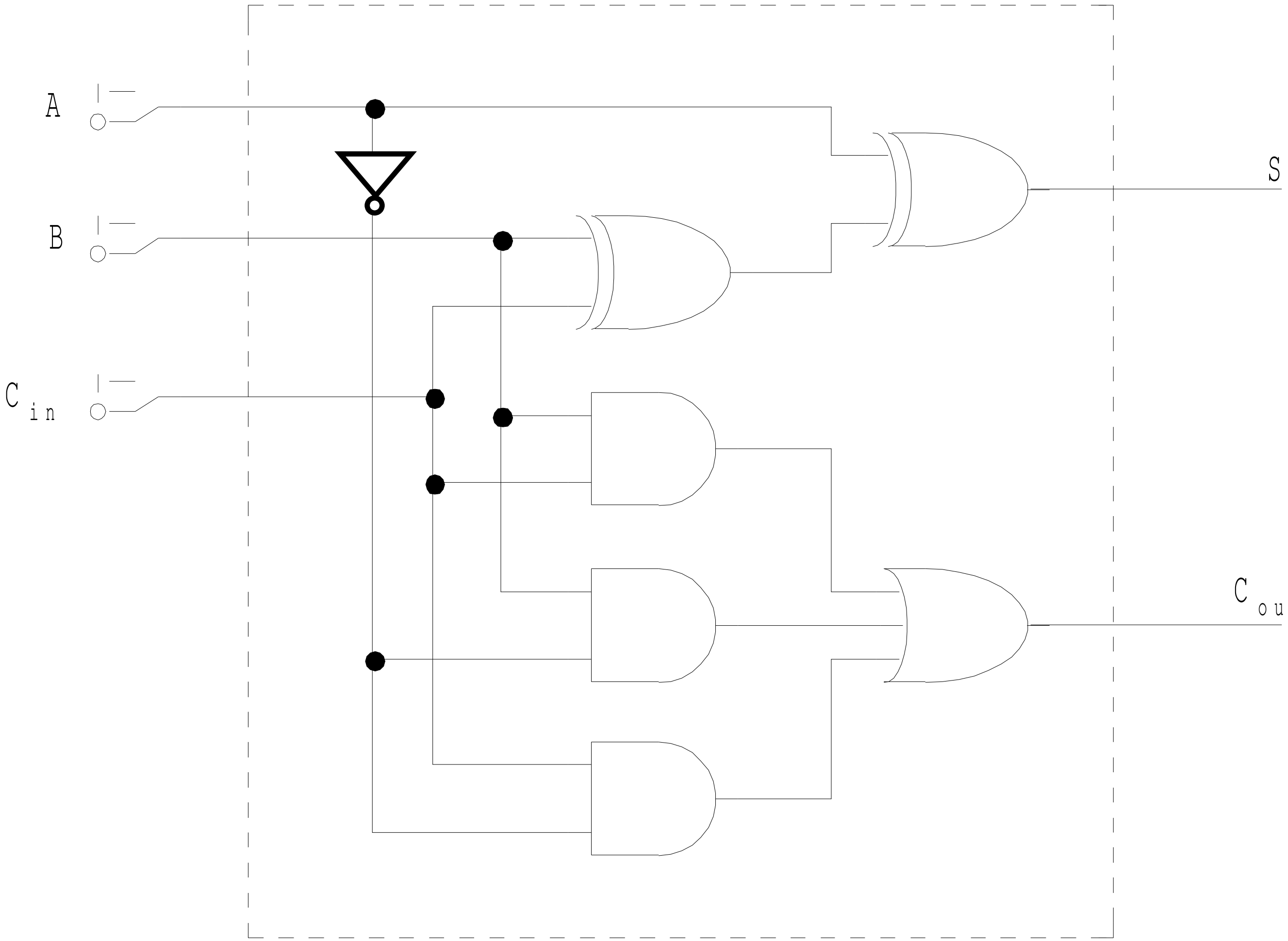
Somador Ripple-Carry

Soluções

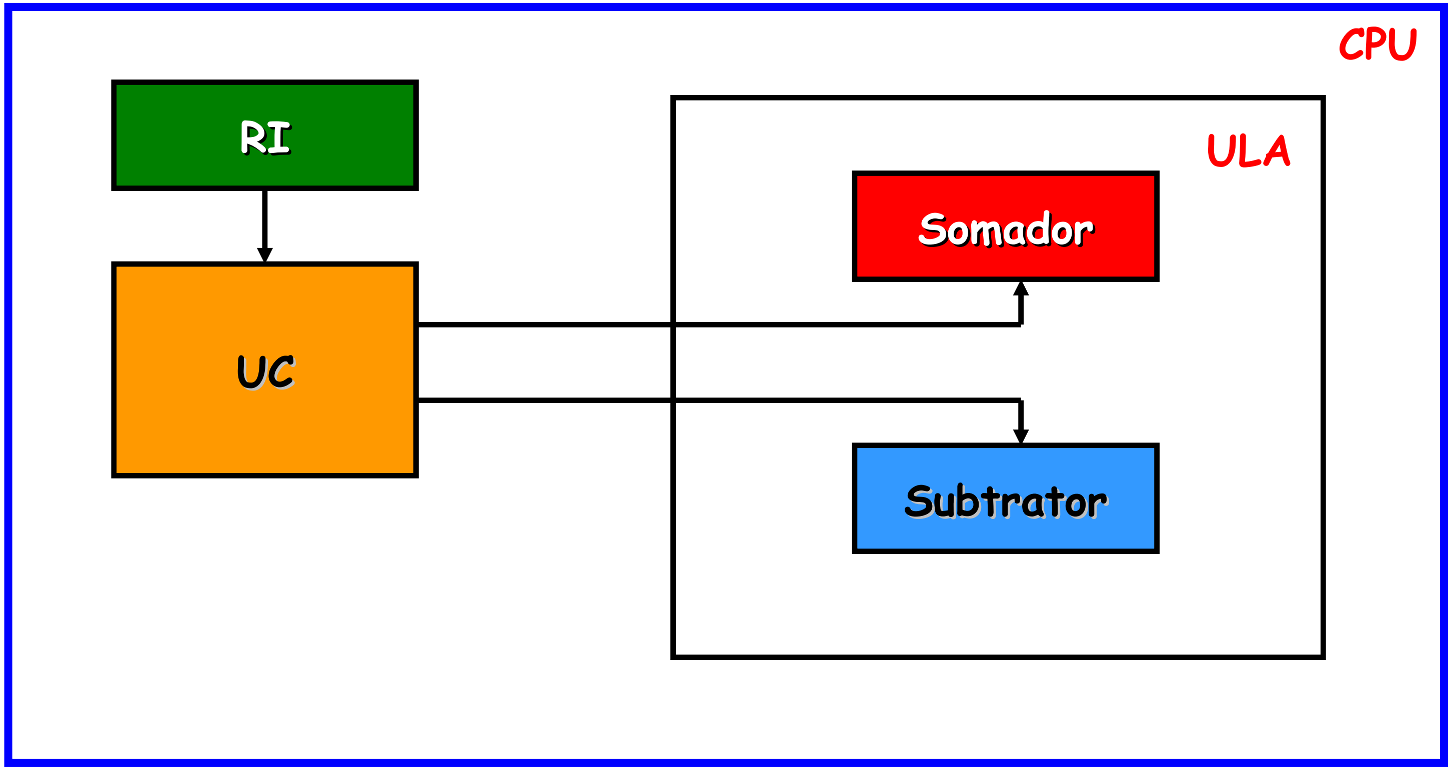
Circuito Subtrator

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = \overline{A}B + BC_{in} + \overline{A}C_{in}$$



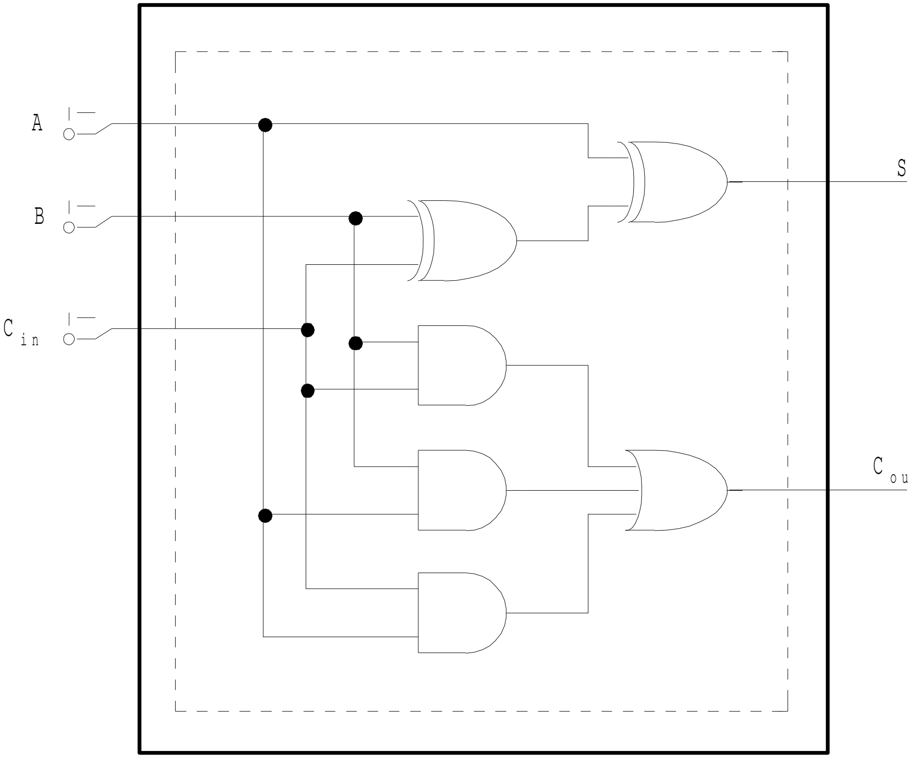
ULA: Somador e Subtrator



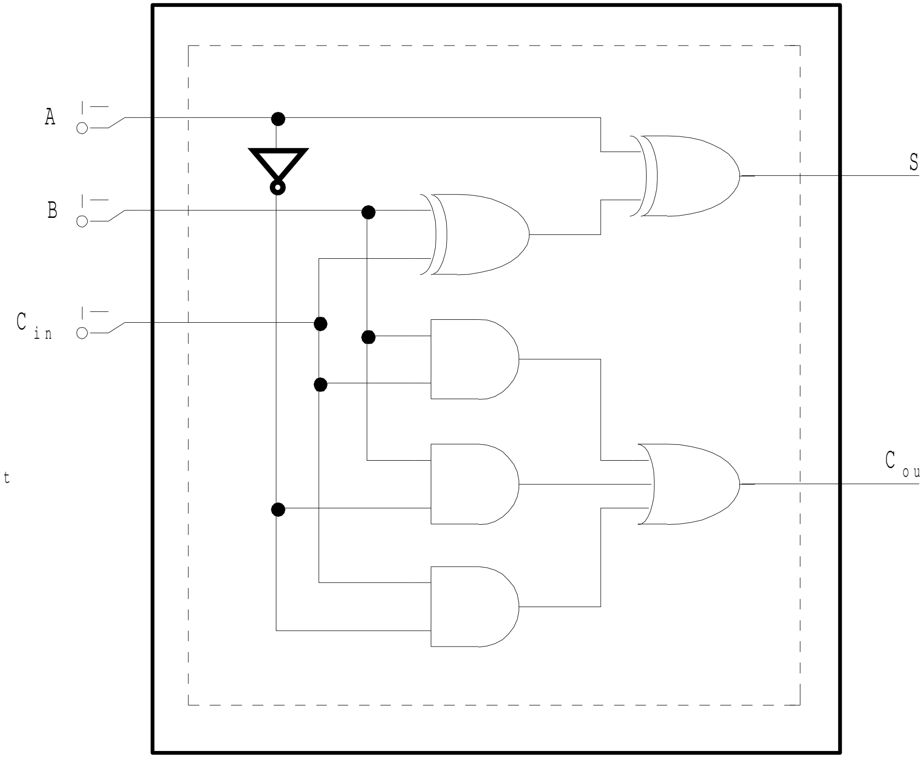
ULA com 2 circuitos para efetuar a adição e a subtração

ULA: Somador e Subtrator

Circuito Somador

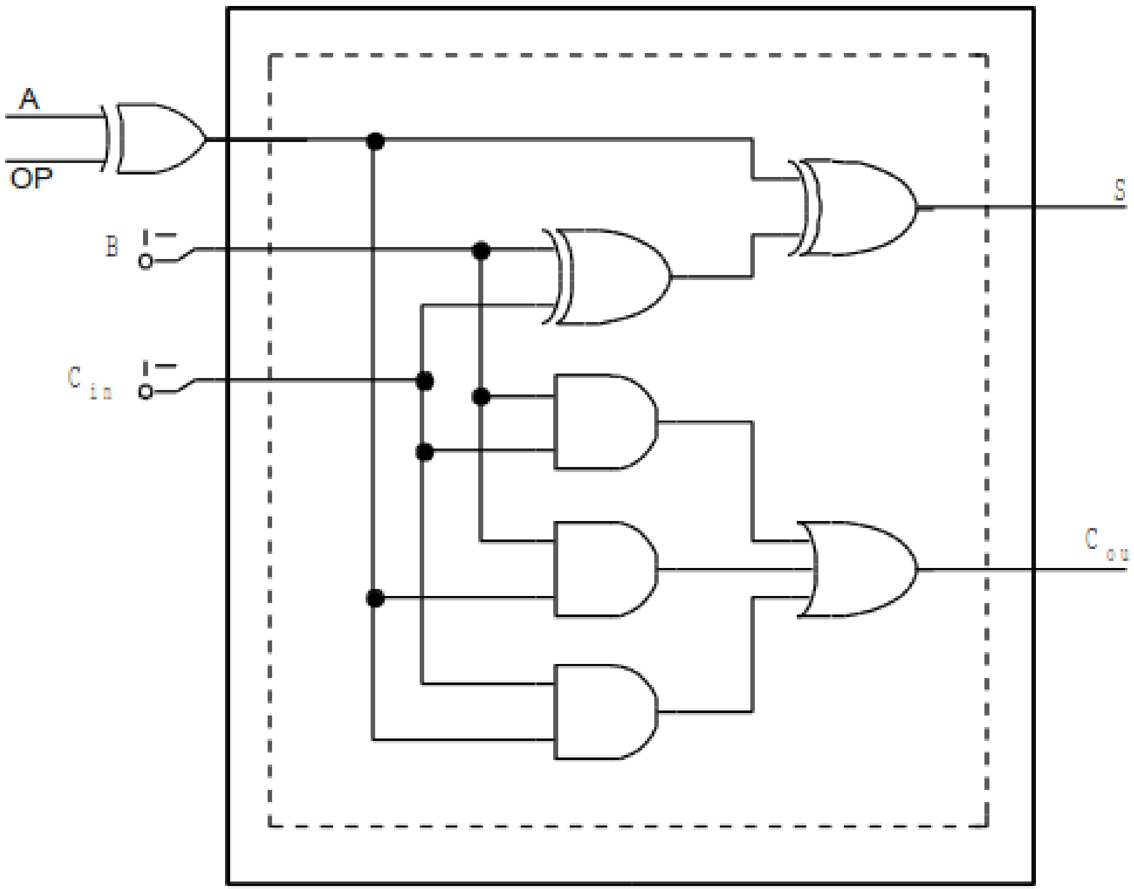


Circuito Subtrator



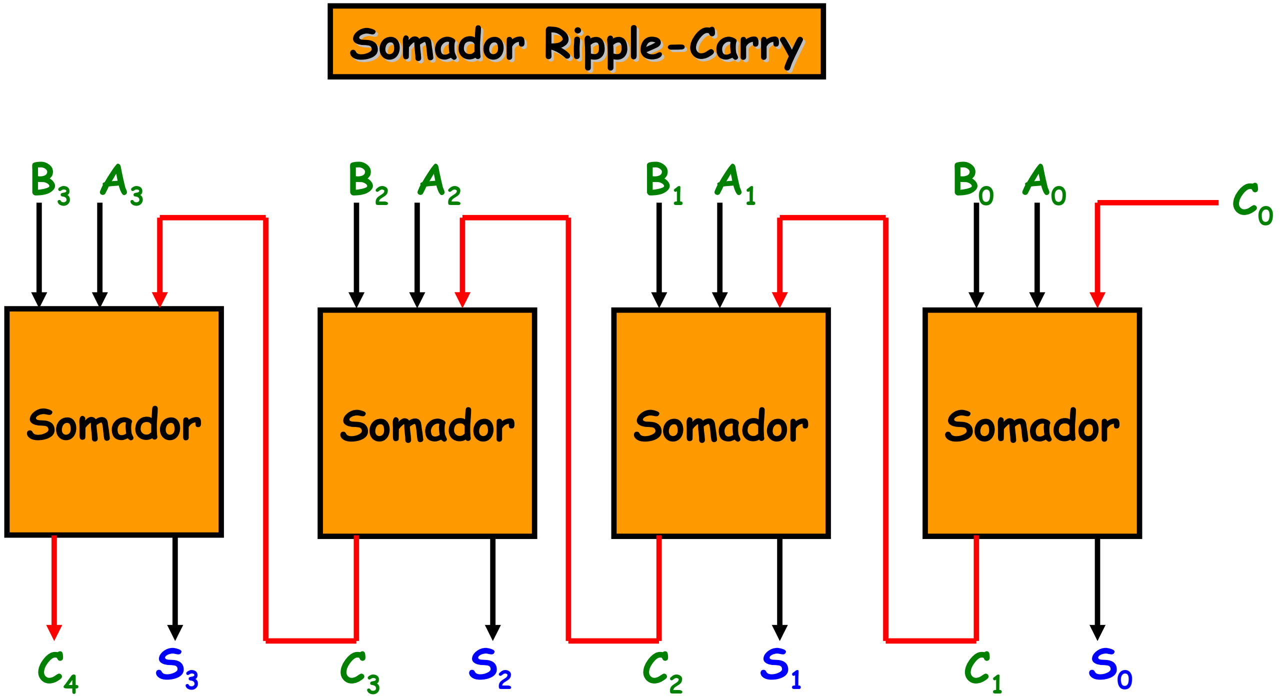
ULA: Somador e Subtrator

Circuito Somador/Subtrator



Somador de Alto Desempenho

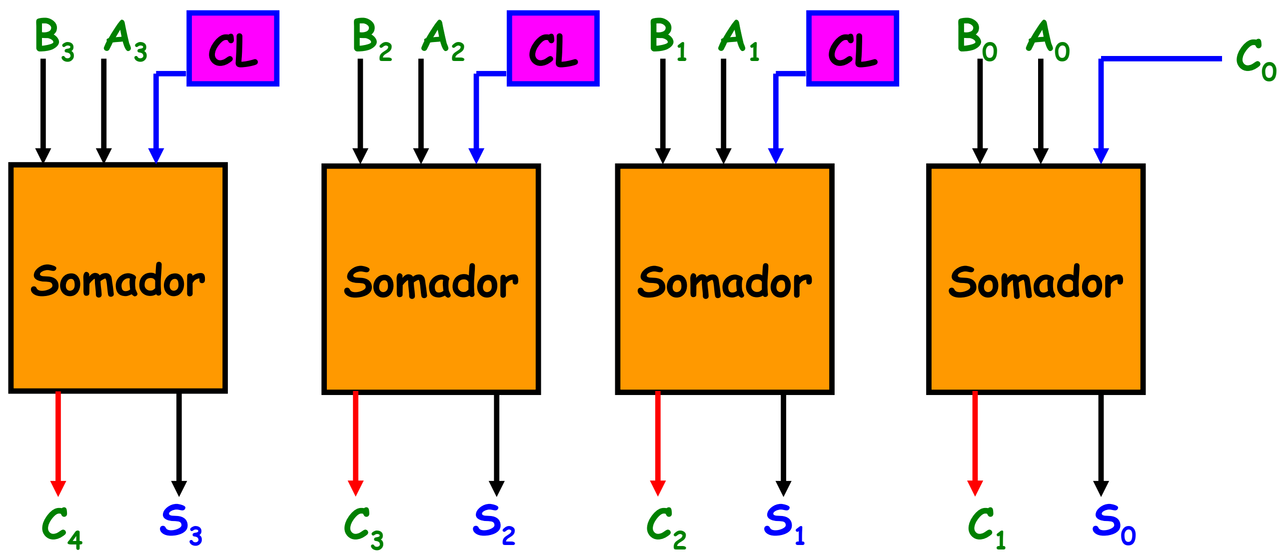
Somador Convencional



Somador Convencional: Atrasos para propagar o carry

Somador de Alto Desempenho

Somador Carry Lookahead



CL: Lógica para antecipar o carry sem passar pelo somador

Somador de Alto Desempenho

A	B	C-out	
0	0	0	"nada"
0	1	C-in	"propaga"
1	0	C-in	"propaga"
1	1	1	"gera"

Expressão do Carry do Somador

$C_{i+1} = A_iB_i + A_iC_i + B_iC_i$

1. **Fatorando a expressão**

$C_{i+1} = A_iB_i + C_i\,(A_i+B_i)$

2. **Chamando A_iB_i de G_i e A_i+B_i de P_i**

$C_{i+1} = G_i + P_iC_i$

3. **Substituindo os índices para obter os carries para um somador de 4 bits**

$C_1 = G_0 + P_0C_0$

4. **Para simplificar a análise, vamos considerar $C_0=0$ para soma**

$C_1 = G_0$

Somador de Alto Desempenho

Expressão do Carry do Somador

$C_2 = G_1 + P_1C_1$

5. Substituindo $C_1 = G_0$

$C_2 = G_1 + P_1G_0$

6. Obtendo C_3

$C_3 = G_2 + P_2C_2$

7. Substituindo $C_2 = G_1 + P_1G_0$

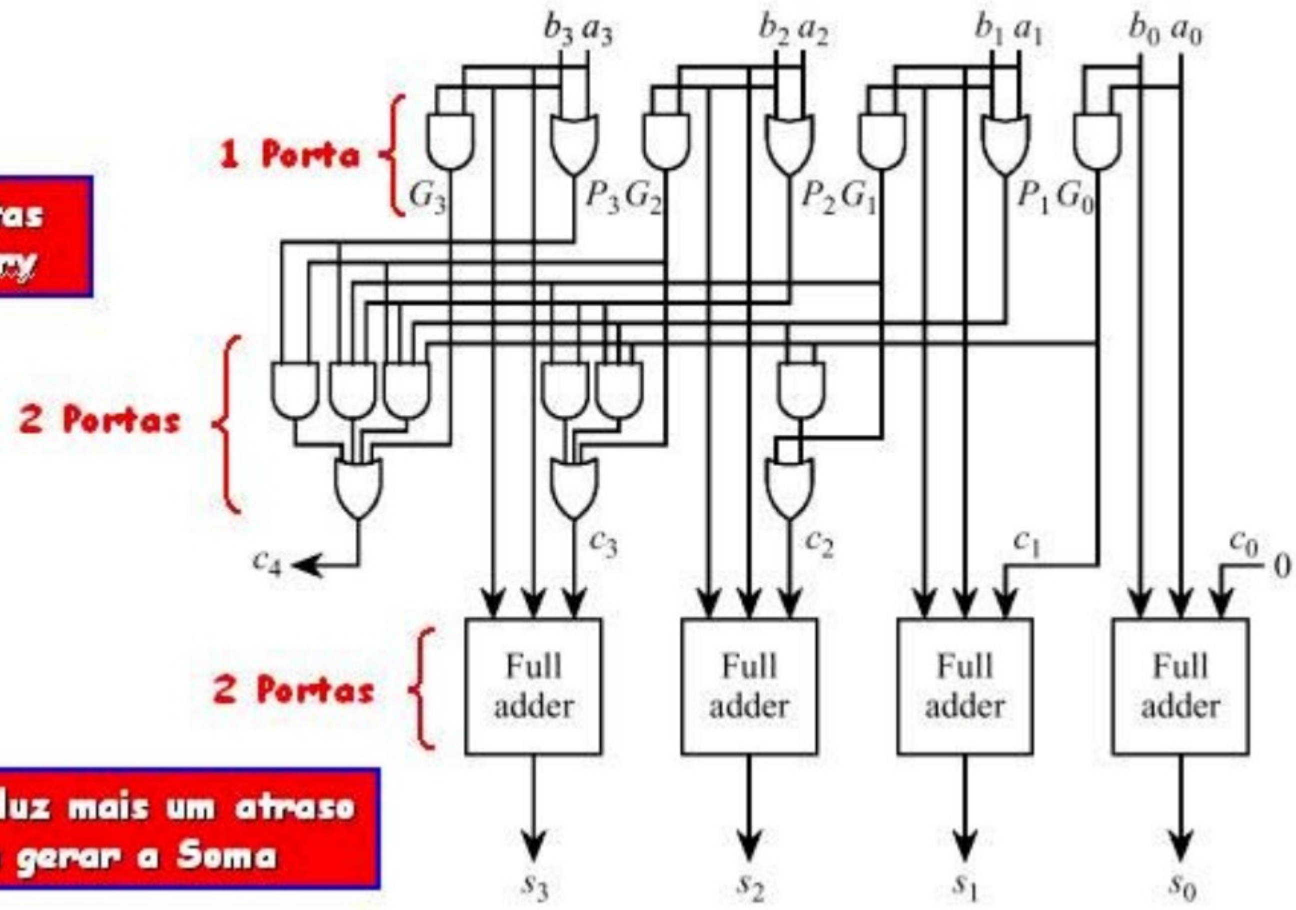
$C_3 = G_2 + P_2(G_1 + P_1G_0) \Rightarrow C_3 = G_2 + P_2G_1 + P_2P_1G_0$

8. Obtendo C_4

$C_4 = G_3 + P_3C_3 \Rightarrow C_4 = G_3 + P_3(G_2 + P_2G_1 + P_2P_1G_0) \Rightarrow C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$

Somador de Alto Desempenho

Atraso de 3 Portas para gerar o carry



O somador introduz mais um atraso de 2 Portas para gerar a Soma

Aritmética Computacional

Multiplicação e Divisão

Aritmética Computacional

Multiplicação em Binário:

Exemplo

a) $\begin{array}{r} 0 \\ 0\ \times \\ \hline 0 \end{array}$ b) $\begin{array}{r} 0 \\ 1\ \times \\ \hline 0 \end{array}$ c) $\begin{array}{r} 1 \\ 0\ \times \\ \hline 0 \end{array}$ d) $\begin{array}{r} 1 \\ 1\ \times \\ \hline 1 \end{array}$

	1 1 0 1	Multiplicando (M)
	× 1 0 1 1	Multiplicador (Q)
n-1 somas parciais	1 1 0 1	Produtos Parciais
	1 1 0 1	
	0 0 0 0	
	+ 1 1 0 1	
Produto final tem 2n bits →	1 0 0 0 1 1 1 1	Produto

Aritmética Computacional

Aritmética Computacional

Multiplicação em Binário:

Cada dígito do Multiplicador, a partir da direita deve multiplicar o Multiplicando gerando um produto parcial

Quando o bit do Multiplicador é 0 ⇒ Produto Parcial é 0

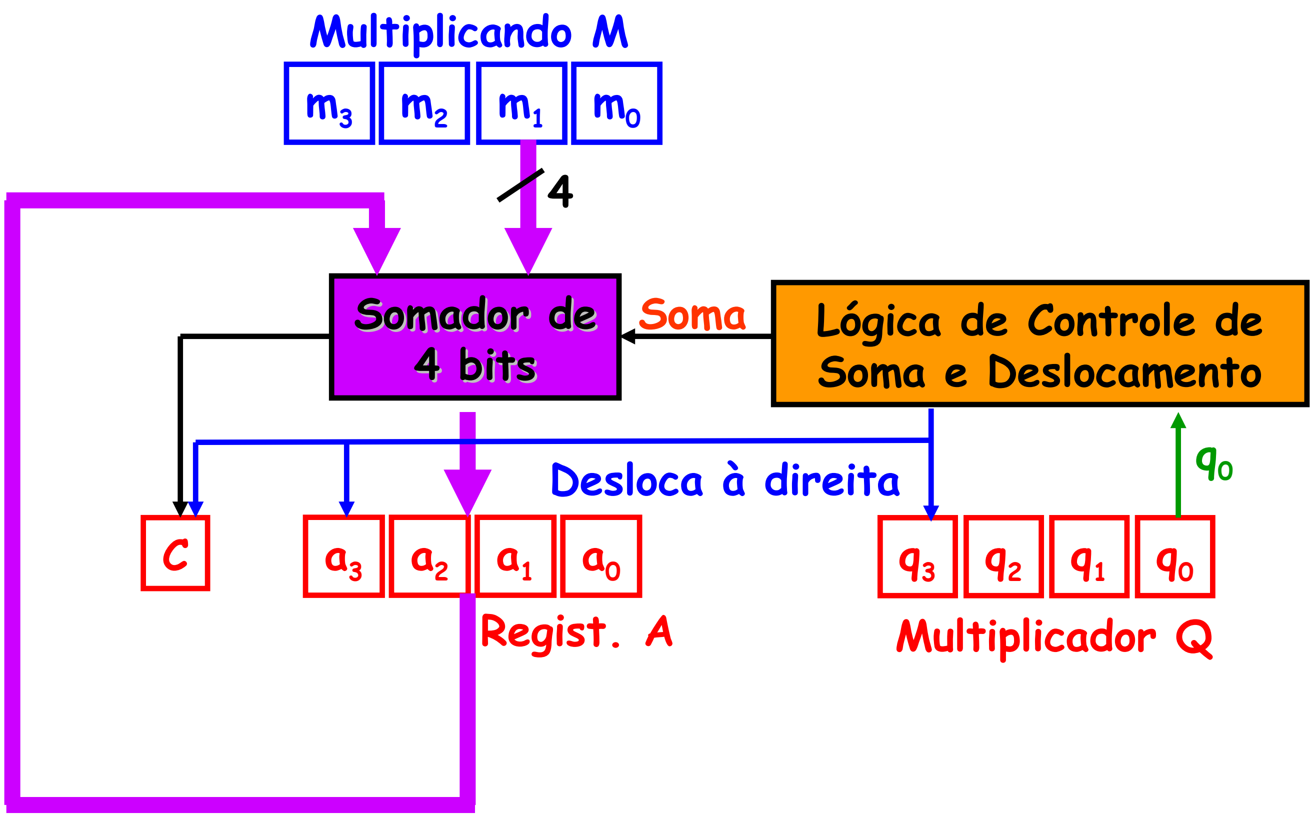
Quando o bit do Multiplicador é 1 ⇒ Produto Parcial é o próprio Multiplicando

Cada Produto Parcial é deslocado 1 bit à esquerda em relação ao produto parcial anterior

$$
\begin{array}{r l}
1\,1\,0\,1 & \text{Multiplicando (M)} \\
\times\,1\,0\,1\,1 & \text{Multiplicador (Q)} \\
\hline
1\,1\,0\,1 & \\
1\,1\,0\,1 & \text{Produtos Parciais} \\
0\,0\,0\,0 & \\
+\ 1\,1\,0\,1 & \\
\hline
1\,0\,0\,0\,1\,1\,1\,1 & \text{Produto}
\end{array}
$$

Aritmética Computacional

Multiplicação Binária



Aritmética Computacional

Algoritmo de Multiplicação

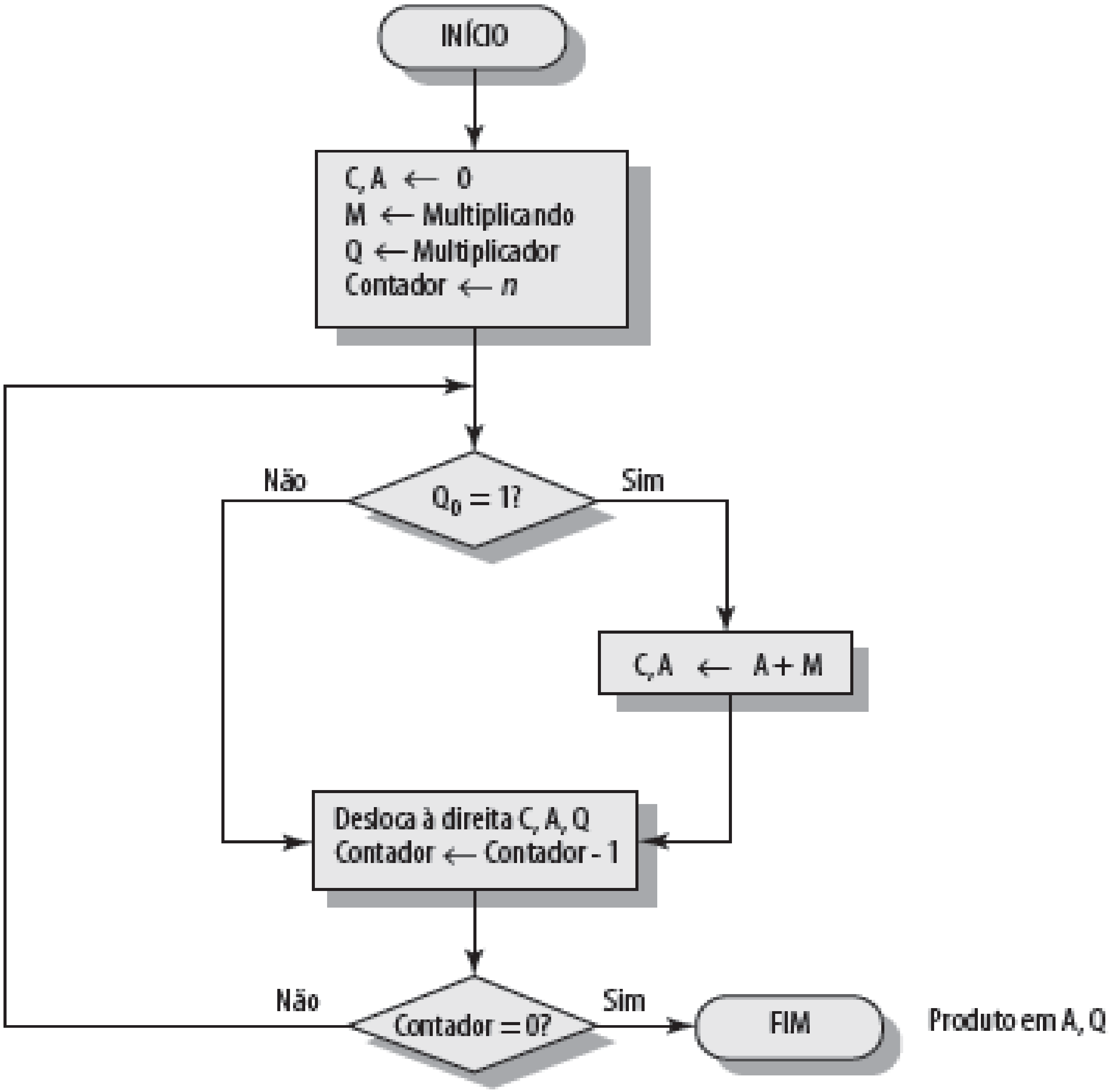
1. Registrador M ⇐ Multiplicando
2. Registrador Q ⇐ Multiplicador
3. Registrador A ⇐ 0
4. Registrador C ⇐ 0
5. Bit q_0 do Multiplicador é testado

   Se q_0=0 {
   - Não soma M e A (o produto parcial não é somado ao Multiplicando)
   - Desloca os registradores C/A/Q para a direita

   }

   Se q_0=1 {
   - Soma M e A resultado fica em A)
   - Desloca os registradores C/A/Q para a direita

   }
6. Repete o passo 5, n vezes (n = nº de bits de M e Q)
7. Produto (resultado final) está armazenado em A e Q


INÍCIO
C, A ← 0
M ← Multiplicando
Q ← Multiplicador
Contador ← n
$Q_0 = 1?$
Não
Sim
C, A ← A + M
Desloca à direita C, A, Q
Contador ← Contador - 1
Contador = 0?
Não
Sim
FIM
Produto em A, Q

Multiplicação Binária

1101
×1011

Exemplo:

1 1 0 1 ← Multiplicando (M)

C	A	Q
0	0 0 0 0	1 0 1 1 ← Multiplicador (Q)

C	A	Q
0	0 0 0 0	1 0 1 $\boxed{1}$ $q_0=1 \Rightarrow A+M$

Aritmética Computacional

Multiplicação Binária

1101
×1011

Exemplo:

1 1 0 1 Multiplicando (M)

	C	A	Q	
n=1	0	1 1 0 1	1 0 1 $\boxed{1}$	$q_0=1 \Rightarrow A+M$

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Multiplicação Binária

1101
×1011

Exemplo:

1 1 0 1 Multiplicando (M)

	C	A	Q	
n=1	0	1 1 0 1	1 0 1 1	
	0	0 1 1 0	1 1 0 1	desloca para a direita

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Multiplicação Binária

Exemplo:

1101
×1011

1 1 0 1 Multiplicando (M)

	C	A	Q	
n=1	0	1 1 0 1	1 0 1 1	
	0	0 1 1 0	1 1 0 $\boxed{1}$	$q_0=1 \Rightarrow A+M$
n=2	1	0 0 1 1	1 1 0 1	

Aritmética Computacional

Multiplicação Binária

1101
×1011

Exemplo:

1 1 0 1 Multiplicando (M)

	C	A	Q	
n=1	0	1 1 0 1	1 0 1 1	
	0	0 1 1 0	1 1 0 1	
n=2	1	0 0 1 1	1 1 0 1	
	0	1 0 0 1	1 1 1 0	desloca para a direita

Multiplicação Binária

1101
×1011

Exemplo:

1 1 0 1 Multiplicando (M)

	C	A	Q	
n=1	0	1 1 0 1	1 0 1 1	
	0	0 1 1 0	1 1 0 1	
n=2	1	0 0 1 1	1 1 0 1	
	0	1 0 0 1	1 1 1 $\boxed{0}$	$q_0=0 \Rightarrow$ Não soma A+M, desloca

Aritmética Computacional

Multiplicação Binária

1101
×1011

Exemplo:

1 1 0 1 Multiplicando (M)

	C	A	Q	
n=1	0	1 1 0 1	1 0 1 1	
	0	0 1 1 0	1 1 0 1	
n=2	1	0 0 1 1	1 1 0 1	
	0	1 0 0 1	1 1 1 0	
n=3	0	0 1 0 0	1 1 1 1	desloca para a direita

Aritmética Computacional

Multiplicação Binária

1101
×1011

Exemplo:

1 1 0 1 Multiplicando (M)

	C	A	Q
n=1	0	1 1 0 1	1 0 1 1
	0	0 1 1 0	1 1 0 1
n=2	1	0 0 1 1	1 1 0 1
	0	1 0 0 1	1 1 1 0
n=3	0	0 1 0 0	1 1 1 $\boxed{1}$ $q_0=1 \Rightarrow A+M$
n=4	1	0 0 0 1	1 1 1 1

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Multiplicação Binária

1101
×1011

Exemplo:

1 1 0 1 Multiplicando (M)

	C	A	Q	
n=1	0	1 1 0 1	1 0 1 1	
	0	0 1 1 0	1 1 0 1	
n=2	1	0 0 1 1	1 1 0 1	
	0	1 0 0 1	1 1 1 0	
n=3	0	0 1 0 0	1 1 1 1	
n=4	1	0 0 0 1	1 1 1 1	
	0	1 0 0 0	1 1 1 1	desloca para a direita

$\underbrace{0 \quad 1\,0\,0\,0 \quad 1\,1\,1\,1}_{\text{Produto}}$

Exercício

Faça a multiplicação de 0010_2 (M) x 0011_2 (Q)

Solução

Algoritmo de Multiplicação

1. Registrador M ⇐ Multiplicando
2. Registrador Q ⇐ Multiplicador
3. Registrador A ⇐ 0
4. Registrador C ⇐ 0
5. Bit q_0 do Multiplicador é testado

   Se q_0=0 {

   - Não soma M e A (o produto parcial não é somado ao Multiplicando)
   - Desloca os registradores C/A/Q para a direita

   }

   Se q_0=1{

   - Soma M e A resultado fica em A)
   - Desloca os registradores C/A/Q para a direita

   }
6. Repete o passo 5, n vezes (n = nº de bits de M e Q)
7. Produto (resultado final) está armazenado em A e Q

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Solução

Multiplicação Binária

0010
×0011

0 0 1 0 ← Multiplicando (M)

0	0 0 0 0	0 0 1 1 ← Multiplicador (Q)
C	A	Q
0	0 0 0 0	0 0 1 $\boxed{1}$ $q_0=1 \Rightarrow A+M$
C	A	Q

Solução

Multiplicação Binária

0 0 1 0 Multiplicando (M)

	C	A	Q	
n=1	0	0 0 1 0	0 0 1 $\boxed{1}$	$q_0=1 \Rightarrow A+M$

Solução

Multiplicação Binária

0 0 1 0 Multiplicando (M)

	C	A	Q	
n=1	0	0 0 1 0	0 0 1 1	
	0	0 0 0 1	0 0 0 1	desloca para a direita

Solução

Multiplicação Binária

0 0 1 0 Multiplicando (M)

		C	A	Q	
n=1		0	0 0 1 0	0 0 1 1	
		0	0 0 0 1	0 0 0 $\boxed{1}$	$q_0=1 \Rightarrow A+M$
n=2		0	0 0 1 1	0 0 0 1	

Solução

Multiplicação Binária

0 0 1 0 Multiplicando (M)

	C	A	Q	
n=1	0	0 0 1 0	0 0 1 1	
	0	0 0 0 1	0 0 0 1	
n=2	0	0 0 1 1	0 0 0 1	
	0	0 0 0 1	1 0 0 0	desloca para a direita

Solução

Multiplicação Binária

0 0 1 0 Multiplicando (M)

	C	A	Q	
n=1	0	0 0 1 0	0 0 1 1	
	0	0 0 0 1	0 0 0 1	
n=2	0	0 0 1 1	0 0 0 1	
	0	0 0 0 1	1 0 0 $\boxed{0}$	$q_0=0 \Rightarrow$ Não soma A+M, desloca

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Solução

Multiplicação Binária

0 0 1 0 Multiplicando (M)

	C	A	Q	
n=1	0	0 0 1 0	0 0 1 1	
	0	0 0 0 1	0 0 0 1	
n=2	0	0 0 1 1	0 0 0 1	
	0	0 0 0 1	1 0 0 0	
n=3	0	0 0 0 0	1 1 0 0	desloca para a direita

Solução

Multiplicação Binária

0 0 1 0 Multiplicando (M)

		C	A	Q
n=1		0	0 0 1 0	0 0 1 1
		0	0 0 0 1	0 0 0 1
n=2		0	0 0 1 1	0 0 0 1
		0	0 0 0 1	1 0 0 0
n=3		0	0 0 0 0	1 1 0 $\boxed{0}$
n=4		0	0 0 0 0	0 1 1 0

$q_0=0 \Rightarrow$ Não soma A+M, desloca

Produto

Aritmética Computacional

Divisão:

Exemplo

Dividendo (Q)	Divisor (M)
1 0 0 1 0 1 0	1 0 0 0
	Quociente
Resto	

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Divisão:

Exemplo

```
   1 0 0 1 0 1 0  | 1 0 0 0
 - 1 0 0 0        |--------
 ---------            1
```

Divisão: - Verifica quantas vezes o divisor pode ser subtraído do dividendo

- Para cada tentativa de subtração insere 1 dígito no quociente

Aritmética Computacional

Aritmética Computacional

Divisão:

Exemplo

```
  1 0 0 1 0 1 0  | 1 0 0 0
- 1 0 0 0        |--------
 ----------            1
        1
```

Aritmética Computacional

Aritmética Computacional

Divisão:

Exemplo

```
  1 0 0 1 0 1 0  | 1 0 0 0
- 1 0 0 0        |--------
  -------              1
        1 0
```

Aritmética Computacional

Aritmética Computacional

Divisão:

Exemplo

```
  1 0 0 1 0 1 0  | 1 0 0 0
- 1 0 0 0        |--------
  -------              1 0
        1 0 1
```

Aritmética Computacional

Divisão:

Exemplo

$$
\begin{array}{r|l}
1\,0\,0\,1\,0\,1\,0 & 1\,0\,0\,0 \\ \cline{2-2}
-\,1\,0\,0\,0 & \,1\,0\,0 \\ \cline{1-1}
1\,0\,1\,0 &
\end{array}
$$

Aritmética Computacional

Aritmética Computacional

Divisão:

Exemplo

```
  1 0 0 1 0 1 0  | 1 0 0 0
– 1 0 0 0        |--------
  -------          1 0 0 1
      1 0 1 0
    – 1 0 0 0
      -------
```

Aritmética Computacional

Divisão:

Exemplo

```
  1 0 0 1 0 1 0  | 1 0 0 0
- 1 0 0 0        |--------
  ---------        1 0 0 1  ← Quociente
      1 0 1 0
    - 1 0 0 0
      -------
          1 0  ← Resto
```

-Para verificar se o divisor "cabe" no dividendo faz-se uma subtração: Dividendo - Divisor

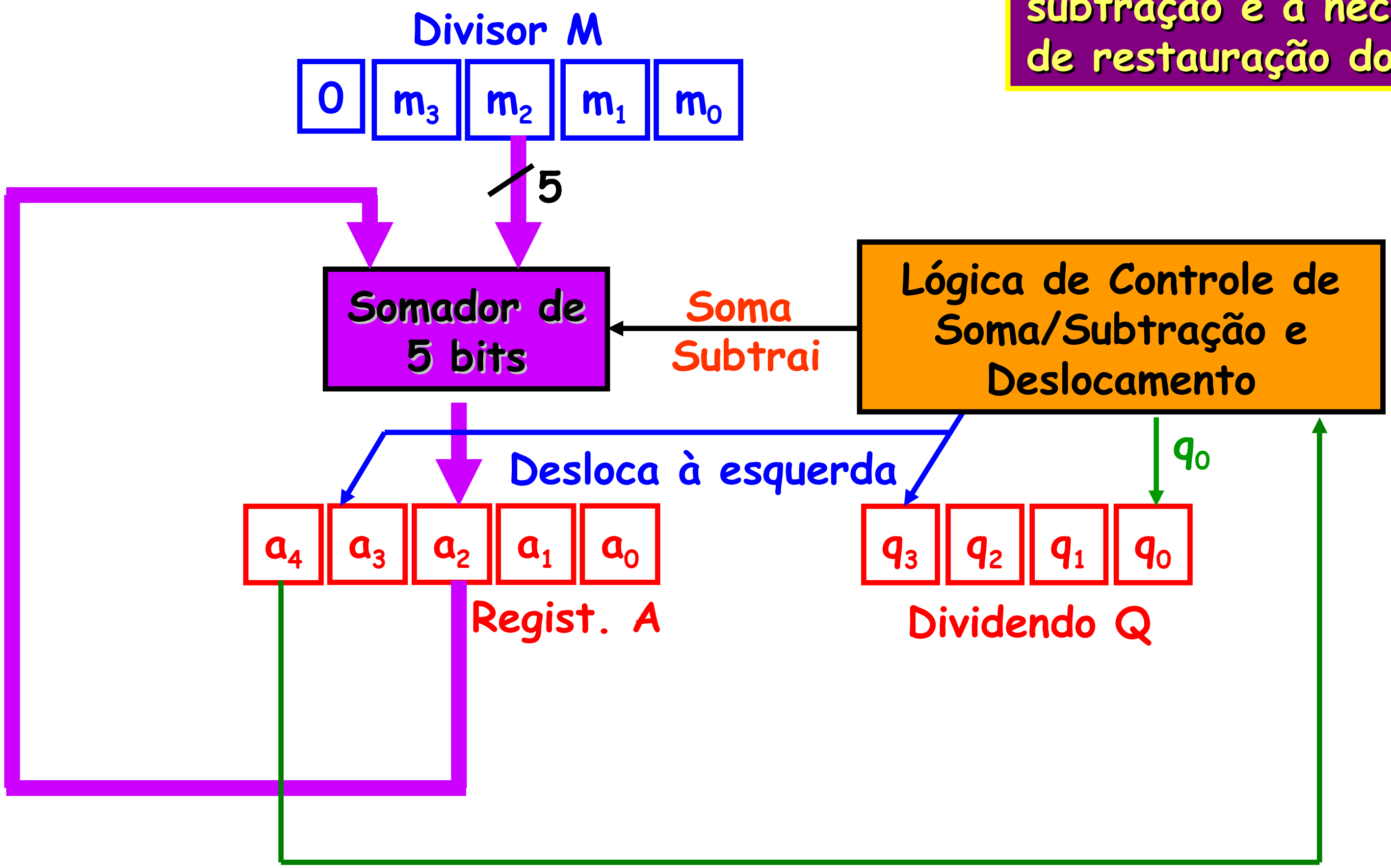
-Se o resultado é negativo significa que ainda não dá para dividir, então restaura-se o valor do dividendo e insere 0 no quociente.

-Se o resultado é positivo significa que dá para dividir, então insere-se 1 no quociente.

Aritmética Computacional

Divisão Binária

Usa registradores de 5 bits para detectar o sinal da subtração e a necessidade de restauração do dividendo



UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Algoritmo de Divisão

1. Registrador Q ⇐ Dividendo
2. Registrador M ⇐ Divisor
3. Registrador A ⇐ 0
4. Bit mais significativo de M (m_4) é zerado
5. Desloca os Registradores A e Q para a esquerda
6. Subtrai A-M para saber se o divisor "cabe" no dividendo
7. Bit a_4 do Dividendo (Registrador A) é testado

   Se a_4=0 {

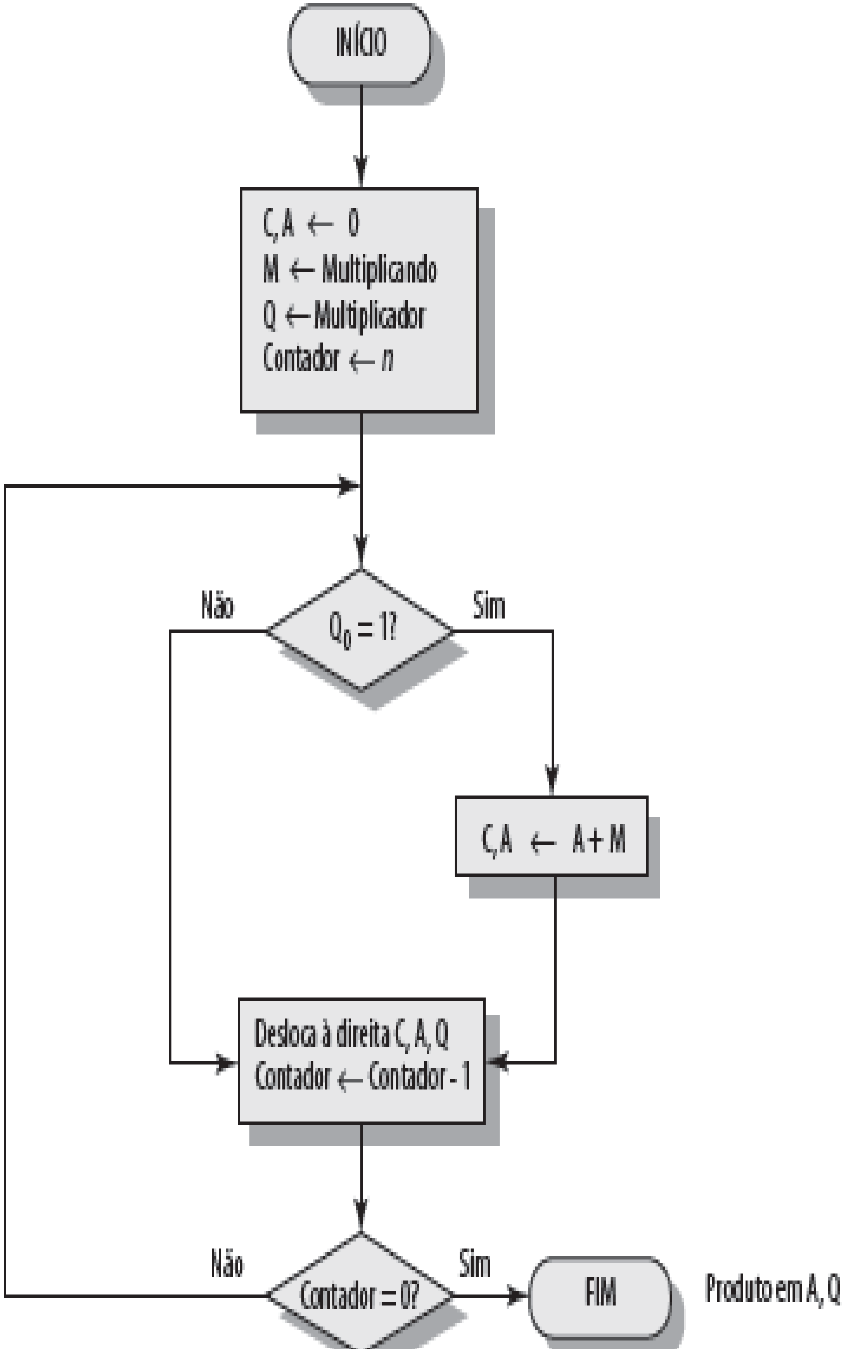
   - Não soma A e M
   - $q_0 \leftarrow 1$ significa que é possível subtrair o divisor do dividendo

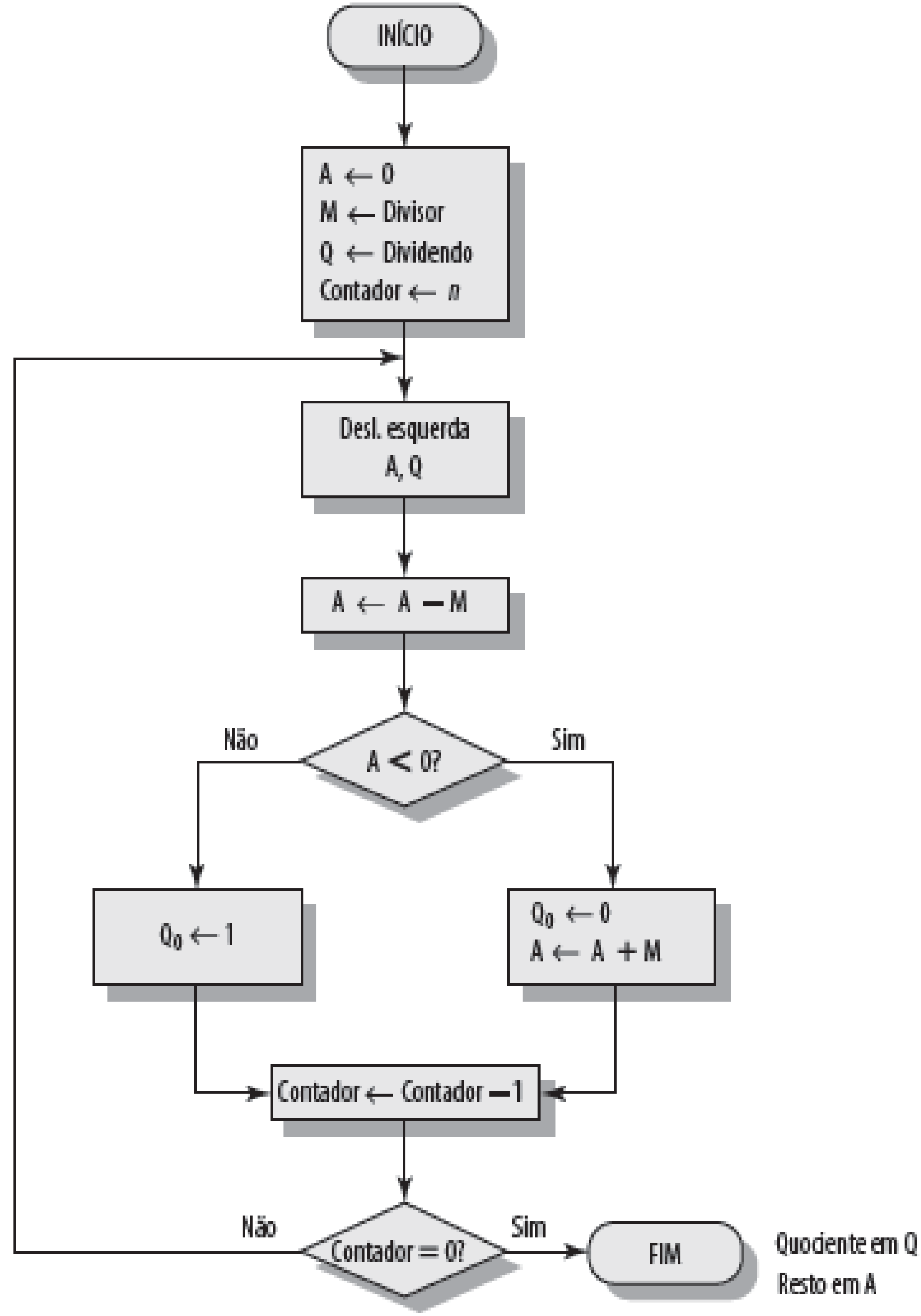
   }

   Se a_4=1 {

   - Soma A e M para restaurar o dividendo
   - $q_0 \leftarrow 0$ (significa que ainda não é possível fazer a divisão)

   }
8. Repete os passos 5 a 7, n vezes (n = nº de bits de M e Q)
9. Quociente está armazenado em Q e o resto em A


INÍCIO
C, A ← 0
M ← Multiplicando
Q ← Multiplicador
Contador ← n
Não
Q0 = 1?
Sim
C, A ← A + M
Desloca à direita C, A, Q
Contador ← Contador - 1
Não
Contador = 0?
Sim
FIM
Produto em A, Q


INÍCIO
A ← 0
M ← Divisor
Q ← Dividendo
Contador ← n
Desl. esquerda
A, Q
A ← A − M
Não
A < 0?
Sim
Q0 ← 1
Q0 ← 0
A ← A + M
Contador ← Contador − 1
Não
Contador = 0?
Sim
FIM
Quociente em Q
Resto em A

Aritmética Computacional

Divisão Binária

Exemplo:

0 0 0 1 1 ← Divisor (M)

A	Q
0 0 0 0 0	0 1 1 1 ← Dividendo (Q)

Aritmética Computacional

Divisão Binária

Exemplo:

	0 0 0 1 1	← Divisor (M)	
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
	A	Q	
n=1	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=1	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 0 1	1 1 1 0	A-M (Compl. 2 de M e soma)

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	$\boxed{0\ 0\ 0\ 0\ 0}$	$\boxed{0\ 1\ 1\ 1}$	← Dividendo (Q)
n=1	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
a_4=1 →	$\boxed{1}$ 1 1 0 1	1 1 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 1 1 0	A+M (restaura divisor)

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=1	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 0 1	1 1 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
	0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=1	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 0 1	1 1 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
	0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)
n=2	0 0 0 0 1	1 1 0 0	desloca A e Q para a esquerda

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=1	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 0 1	1 1 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
	0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)
n=2	0 0 0 0 1	1 1 0 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 1 0	1 1 0 0	A-M (Compl. 2 de M e soma)

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Divisão Binária

Exemplo:

0 0 0 1 1 ← Divisor (M)

A	Q	
0 0 0 0 0	0 1 1 1	← Dividendo (Q)

n		A	Q	
n=1		0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
	$a_4=1$ →	1 1 1 0 1	1 1 1 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
		0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)
n=2		0 0 0 0 1	1 1 0 0	desloca A e Q para a esquerda
	$a_4=1$ →	1 1 1 1 0	1 1 0 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 1	1 1 0 0	A+M (restaura divisor)

Aritmética Computacional

Divisão Binária

Exemplo:

0 0 0 1 1 ← Divisor (M)

		A	Q	
		0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=1		0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
	$a_4=1$ →	1 1 1 0 1	1 1 1 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
		0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)
n=2		0 0 0 0 1	1 1 0 0	desloca A e Q para a esquerda
	$a_4=1$ →	1 1 1 1 0	1 1 0 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 1	1 1 0 0	A+M (restaura divisor)
		0 0 0 0 1	1 1 0 0	Zera q_0 ($q_0 \leftarrow 0$)

Aritmética Computacional

Divisão Binária

Exemplo:

0 0 0 1 1 ← Divisor (M)

A	Q	
0 0 0 0 0	0 1 1 1	← Dividendo (Q)
0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda

n=3

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Divisão Binária

Exemplo:

	A		Q	
	0 0 0 1 1			← Divisor (M)
	0 0 0 0 0		0 1 1 1	← Dividendo (Q)
n=3	0 0 0 1 1		1 0 0 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 0		1 0 0 0	A-M (Compl. 2 de M e soma)

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=3	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 0	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 0 0 1	$q_0 \leftarrow 1$

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=3	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 0	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 0 0 1	$q_0 \leftarrow 1$
n=4	0 0 0 0 1	0 0 1 0	desloca A e Q para a esquerda

Aritmética Computacional

Divisão Binária

Exemplo:

0 0 0 1 1 ← Divisor (M)

A	Q	
0 0 0 0 0	0 1 1 1	← Dividendo (Q)

n=3

	A	Q	
	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 0	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 0 0 1	$q_0 \leftarrow 1$

n=4

	A	Q	
	0 0 0 0 1	0 0 1 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 1 0	0 0 1 0	A-M (Compl. 2 de M e soma)

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=3	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
a_4=0 →	0 0 0 0 0	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 0 0 1	$q_0 \leftarrow 1$
n=4	0 0 0 0 1	0 0 1 0	desloca A e Q para a esquerda
a_4=1 →	1 1 1 1 0	0 0 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 1	0 0 1 0	A+M (restaura divisor)

Aritmética Computacional

Divisão Binária

Exemplo:

	A	Q	
	0 0 0 1 1		← Divisor (M)
	$\boxed{0\ 0\ 0\ 0\ 0}$	$\boxed{0\ 1\ 1\ 1}$	← Dividendo (Q)
n=3	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
$a_4=0$ →	$\boxed{0}$ 0 0 0 0	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 0 0 $\boxed{1}$	$q_0 \leftarrow 1$
n=4	0 0 0 0 1	0 0 1 0	desloca A e Q para a esquerda
$a_4=1$ →	$\boxed{1}$ 1 1 1 0	0 0 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 1	0 0 1 0	A+M (restaura divisor)
	0 0 0 0 1	0 0 1 $\boxed{0}$	Zera q_0 ($q_0 \leftarrow 0$)
	Resto	Quociente	

Exercício

Faça a divisão 0111_2 (Q) : 0010_2 (M)

Aritmética Computacional

Algoritmo de Divisão

1. Registrador Q ⇐ Dividendo
2. Registrador M ⇐ Divisor
3. Registrador A ⇐ 0
4. Bit mais significativo de M (m_4) é zerado
5. Desloca os Registradores A e Q para a esquerda
6. Subtrai A-M para saber se o divisor "cabe" no dividendo
7. Bit a_4 do Dividendo (Registrador A) é testado

   Se a_4=0 {

   - Não soma A e M
   - $q_0 \leftarrow 1$ significa que é possível subtrair o divisor do dividendo

   }

   Se a_4=1 {

   - Soma A e M para restaurar o dividendo
   - $q_0 \leftarrow 0$ (significa que ainda não é possível fazer a divisão)

   }
8. Repete os passos 5 a 7, n vezes (n = nº de bits de M e Q)
9. Quociente está armazenado em Q e o resto em A

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

0 0 0 0 0   0 1 1 1 ← Dividendo (Q)

A   Q

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

$\boxed{0\ 0\ 0\ 0\ 0}$ $\boxed{0\ 1\ 1\ 1}$ ← Dividendo (Q)

A Q

n=1 { 0 0 0 0 0 1 1 1 0 desloca A e Q para a esquerda

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

A	Q	
$\boxed{0\ 0\ 0\ 0\ 0}$	$\boxed{0\ 1\ 1\ 1}$	← Dividendo (Q)

n=1

	A	Q	
	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
$a_4=1$ →	$\boxed{1}$ 1 1 1 0	1 1 1 0	A-M (Compl. 2 de M e soma)

Solução

Divisão Binária

	0 0 0 1 0		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
	A	Q	
n=1	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 1 0	1 1 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 1 1 0	A+M (restaura divisor)

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

		A	Q	
		0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=1		0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
	a_4=1 →	1 1 1 1 0	1 1 1 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
		0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

		A	Q	
		0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=1		0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
	$a_4=1$ →	1 1 1 1 0	1 1 1 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
		0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)
n=2		0 0 0 0 1	1 1 0 0	desloca A e Q para a esquerda

Solução

Divisão Binária

		A	Q	
		0 0 0 1 0		← Divisor (M)
		0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=1		0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
	$a_4=1$ →	1 1 1 1 0	1 1 1 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
		0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)
n=2		0 0 0 0 1	1 1 0 0	desloca A e Q para a esquerda
	$a_4=1$ →	1 1 1 1 1	1 1 0 0	A-M (Compl. 2 de M e soma)

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

A	Q	
0 0 0 0 0	0 1 1 1	← Dividendo (Q)

		A	Q	
n=1		0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
	a_4=1 →	1 1 1 1 0	1 1 1 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
		0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)
n=2		0 0 0 0 1	1 1 0 0	desloca A e Q para a esquerda
	a_4=1 →	1 1 1 1 1	1 1 0 0	A-M (Compl. 2 de M e soma)
		0 0 0 0 1	1 1 0 0	A+M (restaura divisor)

Solução

Divisão Binária

	0 0 0 1 0		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
	A	Q	
n=1	0 0 0 0 0	1 1 1 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 1 0	1 1 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 0	1 1 1 0	A+M (restaura divisor)
	0 0 0 0 0	1 1 1 0	Zera q_0 ($q_0 \leftarrow 0$)
n=2	0 0 0 0 1	1 1 0 0	desloca A e Q para a esquerda
$a_4=1$ →	1 1 1 1 1	1 1 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 1	1 1 0 0	A+M (restaura divisor)
	0 0 0 0 1	1 1 0 0	Zera q_0 ($q_0 \leftarrow 0$)

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

A	Q	
0 0 0 0 0	0 1 1 1	← Dividendo (Q)
0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda

n=3

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

0 0 0 0 0	0 1 1 1	← Dividendo (Q)
A	Q	
0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
a_4=0 → 0 0 0 0 1	1 0 0 0	A-M (Compl. 2 de M e soma)

n=3

Solução

Divisão Binária

	A	Q	
	0 0 0 1 0		← Divisor (M)
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=3	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 1	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 1	1 0 0 1	$q_0 \leftarrow 1$

Solução

Divisão Binária

	0 0 0 1 0 ← Divisor (M)		
	$\boxed{0\ 0\ 0\ 0\ 0}$	$\boxed{0\ 1\ 1\ 1}$	← Dividendo (Q)
	A	Q	
n=3	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
	$a_4=0$ → $\boxed{0}$ 0 0 0 1	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 1	1 0 0 $\boxed{1}$	$q_0 \leftarrow 1$
n=4	0 0 0 1 1	0 0 1 0	desloca A e Q para a esquerda

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

	A	Q	
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=3	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 1	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 1	1 0 0 1	$q_0 \leftarrow 1$
n=4	0 0 0 1 1	0 0 1 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 1	0 0 1 0	A-M (Compl. 2 de M e soma)

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Solução

Divisão Binária

0 0 0 1 0 ← Divisor (M)

	A	Q	
	0 0 0 0 0	0 1 1 1	← Dividendo (Q)
n=3	0 0 0 1 1	1 0 0 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 1	1 0 0 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 1	1 0 0 1	$q_0 \leftarrow 1$
n=4	0 0 0 1 1	0 0 1 0	desloca A e Q para a esquerda
$a_4=0$ →	0 0 0 0 1	0 0 1 0	A-M (Compl. 2 de M e soma)
	0 0 0 0 1	0 0 1 1	$q_0 \leftarrow 1$
	Resto	Quociente	

Exercícios

Faça a multiplicação de 0110_2 (M) x 0011_2 (Q)

Faça a divisão de 1110_2 (Q) : 0110_2 (M)

Resumo da Aula de Hoje

<u>Tópicos mais importantes:</u>

- **Aritmética Computacional**
- **Circuitos Aritméticos**
  - **Circuito Multiplicador**
  - **Circuito Divisor**
- **<u>Entregar folha com:</u>**
  - **Nome**
  - **RA**
  - **Data de Hoje**
  - **Resumo**

Referências

- **Notas de Aulas do Prof. João Angelo Martini do DIN-UEM.**